Semantic Annotation of Maps Through Knowledge Provenance

Nicholas Del Rio[1], Paulo Pinheiro da Silva[1], Ann Q. Gates[1], Leonardo Salayandia[1]

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

Abstract. Maps are artifacts often derived from multiple sources of data, e.g., sensors, and processed by multiple methods, e.g., gridding and smoothing algorithms. As a result, complex metadata may be required to describe maps semantically. This paper presents an approach to describe maps by annotating associated provenance. Knowledge provenance can represent a semantic annotation mechanism that is more scalable than direct annotation of map. Semantic annotation of maps through knowledge provenance provides several benefits to end users. For example, a user study is presented showing that scientists with different levels of expertise and background are able to evaluate the quality of maps by analyzing their knowledge provenance information.

1 Introduction

Maps are expected to be generated, understood, accepted, shared, and reused by scientists like many other scientific products, e.g., reports and graphs. Semantic annotation of maps is often necessary to assure that scientists are able to understand and evaluate information represented by maps. For example, map annotation can be used by scientists not involved in a map generation process to understand the properties of the map, e.g., recency, geospatial coverage, and data sources used, and evaluate the map against some established criteria, e.g., that the data used in the map generation of the map came from a reliable source. Once a scientist understands and accepts a given map, the scientist can confidently reuse and share the map to save time and resources of other collaborators that would otherwise be required to be regenerated.

There are different methods for annotating maps and images in general, each with their respective benefits. For instance, semantic annotation of maps may be achieved by defining map artifacts as instances of semantic concepts comprising an ontology and may involve the annotation of the resources used to generate maps (e.g., source data types, intermediate data types, and transformation methods). However, a small variation in the generation process of a map, e.g., the use of a different filtering algorithm, would require the introduction of at least a new class in the ontology, along with new semantic annotations. Another challenge of this approach is that it becomes difficult to reuse existing domain ontologies to annotate semantic information. For example, suppose there existed an ontology developed by a third party that contained semantic annotations for general-purpose filtering algorithms; the annotations provided by such ontology might not be rich enough to capture the relationship between a filtering algorithm and its particular application to generate a map artifact.

Provenance information in general is meta-information that can be used to document how products such as maps are generated. Provenance often includes meta-information about the following: original datasets used to derive products; executions of processes, i.e., traces of workflow executions and composite services execution; methods called by workflows and composite services, i.e., services, tools, and applications; intermediate datasets generated during process executions; and any other information sources used. This paper refers to the term *Knowledge provenance* (KP) [14], to account for the above meta-information that includes *provenance meta-information*, which is a description of the origin of a piece of knowledge, and *process meta-information*, which is a description of the reasoning process used to generate the answer, which may include intermediate datasets referred to as *intermediate results*. We have used the phrase "knowledge provenance" instead of data provenance intentionally. Data provenance [3, 4] may be viewed as the analog to knowledge provenance aimed at the database community. That community's definition typically includes both a description of the origin of the information and the process by which it arrived in the database. Knowledge provenance is essentially the same except that it includes proof-like information about the process by which knowledge arrives in the knowledge base. In this sense, knowledge provenance broadens the notion of data derivation that can be performed before data is inserted into a database or after data is retrieved from a database. Nevertheless, data provenance and knowledge provenance have the same concerns and motivations. In this paper we describe how KP can be used to semantically annotate maps and how this semantic information can help scientists to understand and evaluate map products.

The rest of this paper is organized as follows. Section 2 introduces a scenario where a map is generated through a workflow executing over cyberinfrastructure services. Section 3 describes how these services are instrumented to log KP about the workflow execution. Section 4 describes how KP annotation can be used by scientists to better understand how maps are generated. Section 5 describes a user study that demonstrates the need of scientists to have access to KP associated with maps. Section 6 discusses the pros and cons of annotating maps while Section 7 concludes the paper

2 Gravity Map Annotation: An Example

2.1 Gravity Map Scenario

Contour maps generated from gravity data readings serve as models from which geophysicists can identify subterranean features. In particular, geophysicists are often concerned with data anomalies, e.g., spikes and dips, because these are usually indicative of the presence of some subterranean resource such as a water table or an oil reserve. The Gravity Map scenario described in this section is based on a cyberinfrastructure application that generates such gravity contour maps from the Gravity and Magnetic Dataset Repository[1] hosted at the Regional Geospatial Service Center at the University of Texas at El Paso. In this scenario, scientists request the generation of contour maps by providing a footprint defined by a pair latitude and longitude coordinates; this footprint specifies the 2D spatial region of the map to be created. The following sequence of tasks generate gravity data contour maps in this scenario:

1. *Gather Task*: Gather the raw gravity dataset readings for the specified region of interest
2. *Filter Task*: Filter the raw gravity dataset readings (remove unlikely point values)
3. *Grid Task*: Create a uniformly distributed dataset by applying a gridding algorithm
4. *Contour Task*: Create a contoured rendering of the uniformly distributed dataset

Each of the tasks involved in this scenario are realized by a web service, thus emphasizing the use of a loosely coupled, distributed environment comparable to that of a cyberinfrastructure, where semantic annotation information is particularly critical. Furthermore, this particular scenario can be viewed as a pipeline, where the output of a task is used as input in the subsequent task. The specification stating that these tasks must be sequentially executed in the order described above can be viewed as an executable workflow and it is further described in Section 3.2. Of course it is possible to implement the required functionalities as a single autonomous application, however, the availability of these services over the Web as smaller cohesive tasks allows for greater possibility of reuse especially in other domains; tasks 3 and 4 are not specific to gravity data.

2.2 Gravity WDO

Services, datasets, and workflow specifications in the scenario need to be semantically described by an ontology if one wants to understand contour maps about gravity data. In this paper, we rely on the Gravity Workflow Driven Ontology as a source of gravity map concepts and relationships.

Dr. Randy Keller, a leading expert on gravity data, worked with Flor Salcedo to encode his knowledge in the gravity field as an ontology. The development of the ontology was part of the NSF-funded GEON Cyberinfrastructure project [1], and it is part of a concentrated effort to capture essential knowledge about the Gravity domain as it is applied to Geophysical studies. The initial motivation for the effort was to document and share gravity terminology and resources within the GEON community. At the time of this writing, the Gravity ontology contains more than 90 classes fully documented .

[1] http://irpsrvgis00.utep.edu/repositorywebsite/

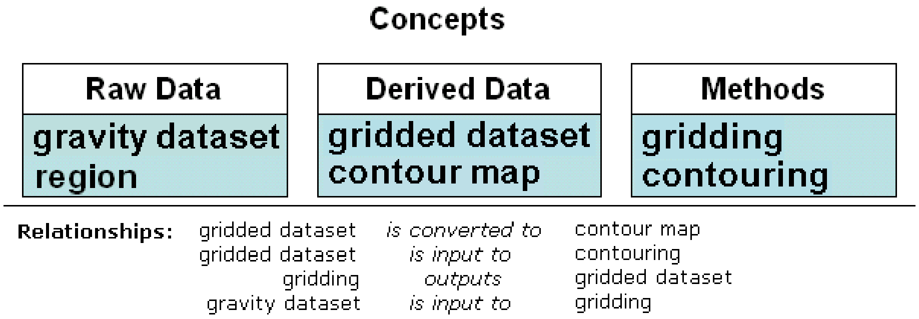

Concepts
Raw Data
gravity dataset
region
Derived Data
gridded dataset
contour map
Methods
gridding
contouring
Relationships:
gridded dataset is converted to contour map
gridded dataset is input to contouring
gridding outputs gridded dataset
gravity dataset is input to gridding

Fig. 1. Gravity Ontology.

Figure 1 presents a visual representation of three upper level classes in the ontology class hierarchy from the Gravity ontology and some of the subclasses related to producing a gravity contour map, e.g., *region* and *gridding*. The Gravity ontology specifies multiple relationships between classes across the three hierarchies; for clarity the relations that are associated with the classes are listed in the sidebar of the figure rather than shown graphically. In the case of workflow-driven ontologies, it is expected that the different types of services published on the cyberinfrastructure are represented as classes defined in an ontology used to create workflows. Consequently, services that correspond to classes under the *Raw Data* and *Derived Data* hierarchy of the ontology are services that provide access to data repositories; services that correspond to classes under the Method hierarchy are services that take data as input, provide some functionality that can transform the data, and outputs the transformed data; and services that correspond to classes under the Product hierarchy are services that provide access to an artifact library.

The relationships between classes provide the basic roadmap to specify complex functionality through composition of services. As an example, consider the second row of the relationship sidebar in Figure 1 that shows the *outputs* between the classes *gridding* and *gridded dataset*. This relationship suggests that, given a service that corresponds to the *gridding* class, a service composition is viable that would result in *derived data* corresponding to a *gridded dataset* class.

2.3 Semantic Annotation of the Gravity Map and Related Work

Semantics are associated with artifacts, such as maps, through appended meta-information known as annotations. Annotations serve as the link between concepts defined in ontologies and artifacts; annotations are simply tags that refer to some concept. For instance, the gravity contour map resulting from the gravity map scenario, can be associated with the *contour map* concept defined in the gravity ontology as shown in Figure 2 without provenance. Scientists or agents would be able to unambiguously identify this artifact as a *contour map*.

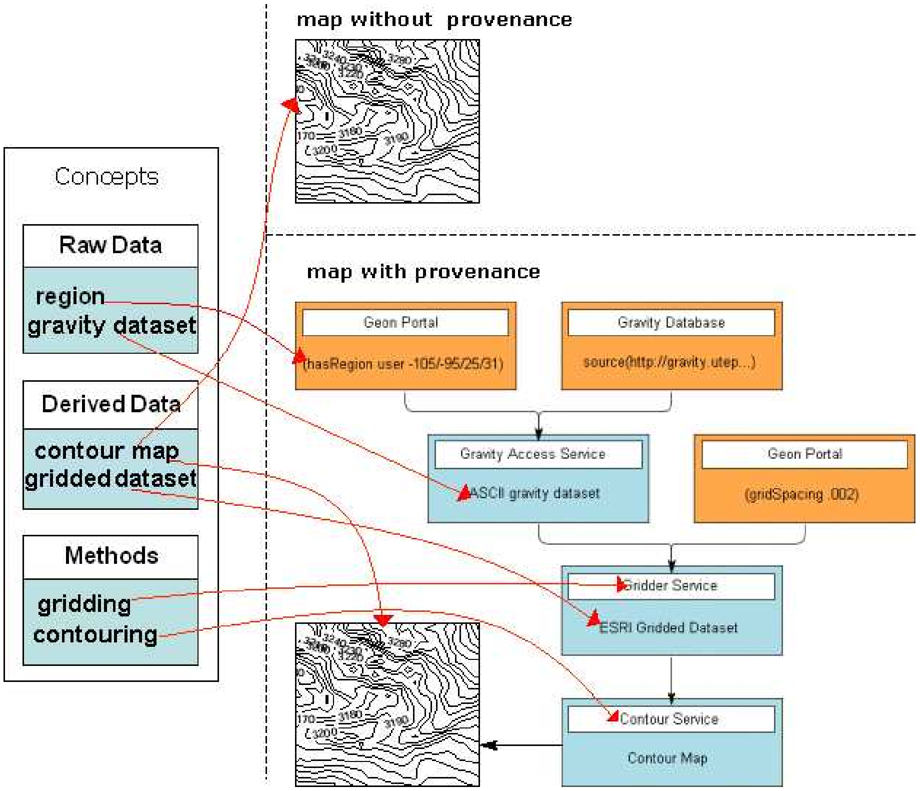


Fig. 2. Annotations with and without provenance.

The current practice is to associate each object in some domain with only one concept in an ontology. Usually, only a single artifact itself (i.e., a map) is annotated by either concepts in some ontology or with arbitrary terms or captions as in Google Maps [8], ArcGis [7], and XML for Image and Map Annotations (XIMA) [6]. In Google Maps, annotations are limited to the map as a whole or for particular latitudes/longitudes (i.e., single points) in the map. In contrast, ArcGis and XIMA allow users to annotate whole maps, points on a map and sub-regions (i.e., subsections defined by polygons) of a map using text based captions. In all of these cases however, only the final map or image is annotated, where as the approach presented in this paper aims to annotate both the map and associated knowledge provenance.

For many cases however, a single concept or annotation may provide adequate semantic information for both human and software agents to correctly manage the artifact. In the gravity map example, the geospatial region provided by the scientist is associated with a *region* concept in the gravity ontology. This provides enough information to the service represented by task 1 to know that the input contains both upper and lower latitudes and longitudes in some particular format and thus facilitate correct parsing; the format of *region* would also be defined in the gravity ontology. A single concept annotation however may not always be enough to define a complex artifact such as a map. Referring to the gravity scenario, the gravity ontology defines a concept *contour map*, which can be used to semantically define the resultant gravity contour map. However, this concept, by itself, says nothing about what kind of map it is (i.e. what kind of data was used to generate this map). If an ontology is very rich, then perhaps complex reasoning might provide answers to the questions posed. Even so, in the case of the gravity ontology, there are many methods defined which generate contour maps (i.e. they all have an *outputs* relationship with the *contour* concept). Reasoning alone could not indicate which methods were used to generate a particular instance of a contour map. A more explicit way to semantically define the map may be to associate both the map and its knowledge provenance to concepts in the ontology as shown in Figure 2 with provenance. In this case, most of the KP can be associated with some concept in an ontology providing better utilization of the knowledge and a much richer description of the artifact. KP already contains the process by which the map was generated including all intermediate data such as the raw gravity dataset. If the gravity dataset, contained in the KP, was defined as an instance of *gravity data*, then any scientist or software agent could quickly realize that the contour map was generated by gravity data and is thus a gravity contour map. In this sense, KP is the medium through which additional semantics, that might otherwise have to be deduced by reasoning, can be appended to the artifact. Adding semantics to KP associated with some artifact in turn adds richer descriptions of the artifact itself. A few systems including PSW, described in Section 3, and MyGrid [16], from the e-science initiative, provide provenance associated with complex artifacts while leveraging ontologies to further enrich the provenance descriptions.

Once KP has been annotated with concepts in the ontology, tools can be used to view this semantically defined provenance. Section 4 further explores such a tool and potential uses.

3 Capturing Gravity Map Knowledge Provenance

3.1 The Inference Web and the Proof Markup Language (PML)

The Inference Web [9, 10] is a knowledge provenance infrastructure for conclusions derived from inference engines which supports interoperable explanations of sources (i.e. sources published on the Web), assumptions, learned information, and answers associated with derived conclusions, that can provide users with a level of trust regarding those conclusions. The goal of the Inference Web is the same as the goal of this work which is to provide users with an understanding of how results are derived by providing them with an accurate account of the derivation process (i.e. knowledge provenance), except that this work deals with workflows rather than inference engines; workflow knowledge provenance encompasses a range of complex artifacts such as datasets and corresponding visualizations while inference Web provenance always consists of logical statements leading to some final conclusions and can thus be regarded as a justification.

Inference Web provides the Proof Markup Language (PML) to encode KP. PML is an RDF based language defined by a rich ontology of provenance and justification concepts which describe the various elements of automatically generated proofs. The main concept defined in PML is *node set*, which contains both a conclusion (i.e., a logical expression) and a collection of inference steps each of which provide a different justification of the conclusion; in its simplest composition, a single PML node set simply represents a single proof step. Inference steps themselves contain a number of elements including antecedents, rule, and inference engine, which correspond to the rule antecedents, the name of the rule applied to the antecedents, and the name inference engine responsible for the derivation respectively. In PML, antecedents are simply references to other node sets comprising the rest of a justification. Thus PML justifications are graphs with node sets as nodes and antecedents acting as edges. This graph is directed and acyclic, with the edges always pointing towards the direction of root, the conclusion of the entire proof. In this sense, node sets always contribute to the final conclusion.

PML justifications can also be used to store KP information associated with scientific workflow execution. From this perspective, node sets represent the execution of a particular web service; the node set conclusion serves as the output of the service (i.e., and intermediate result) while the inference step represents provenance associated with the service's function. For example, elements antecedent, rule, and the inference engine can be used to describe the service's inputs, function, and name or hosting organization respectively. Additionally, the links between nodesets can be viewed as an execution sequence of a workflow.

PML itself is defined in OWL [5, 12] thus supporting the distribution of proofs throughout the Web. Each PML node set comprising a particular justification can reside in a uniquely identified document published on the Web separately from the others. The workflows considered in this research are service oriented and thus distributed. The support provided by PML is so well suited for scientific workflows that it is used as the provenance interlingua for out KP browser Probe-It! briefly described in Section 4. It is also relevant to mention that PML addresses only the encoding issues related to provenance but prescribes no specific method for collecting it.

3.2 Workflows and the PML Service Wrapper (PSW)

The gravity map scenario is realized by a service-oriented workflow composed of four Simple Object Access Protocol (SOAP) services, which gather, filter, grid and contour gravity datasets respectively. These Web services are piped or chained together; the output of one service is forwarded as the input to the next service specified in the workflow. A workflow director is responsible for managing the inputs/outputs of each service as well as coordinating their execution. KP associated with scientific workflows of this nature might include the services execution sequence as well as each of their respective outputs, which we refer to as *intermediate results*.

PML Service Wrapper (PSW) is a general-purpose Web service wrapper that logs knowledge provenance associated with workflow execution as PML documents. In order to capture knowledge provenance associated with workflows execution, each service composing the workflow has an associated PSW wrapper that is configured to accept and generate PML documents specific to it. Since PML node sets include the *conclusion* element, which is used to store the result of an inference step or Web service, the provenance returned by the wrappers also includes the service output thus workflows can be composed only of these PSWs; this configuration introduces a level of indirection between service consumers (i.e. workflow engine) and the target services that performs the required function. In this sense, PSW can be seen as a server side provenance logger.

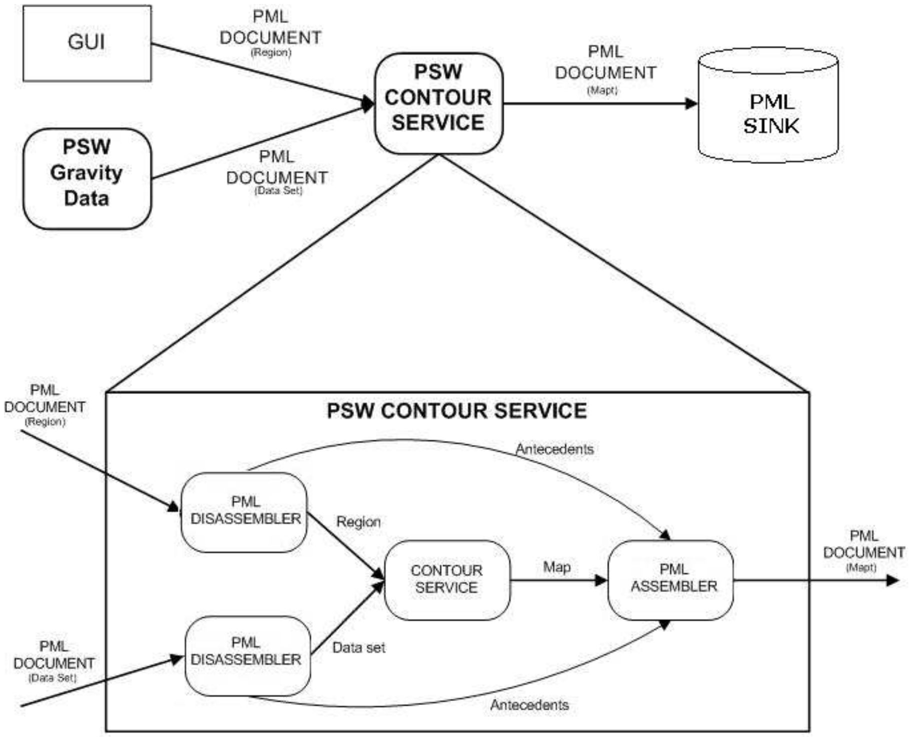


Fig. 3. Example of PSW configured for a contouring service.

The logging capability provided by PSW can be decomposed into three basic tasks: decompose, consume, and compose as illustrated in Figure 3. Upon invocation, the wrapper decomposes the conclusion of an incoming PML document, i.e., extracts the data resident in the PML conclusion using Inference Web's PML API. PSW then consumes the target service, forwarding the extracted data as an input to the target service. The result and associated provenance of the target service is then composed to produce the resultant PML document, the PSW output. For example, a contouring service requires 2D spatial data to map and the region to consider in the mapping therefore a PSW wrapper for this contouring service would require two PML documents, one containing 2D spatial data, coming from some data retrieval service, and the other containing a region, (e.g. specified by latitude and longitude) specified by some user. The output of the contour service is a map, from which a new PML document is created, referencing the two input PML node sets as antecedents.

PSW has been developed in support of scientific workflows able to execute in a distributed environment such as the cyberinfrastructure. In traditional Inference Web applications [13, 10], inference engines are instrumented to generate PML. However in a cyberinfrastructure setting, reasoning is not necessarily deductive and is often supported by Web services that can be considered "black boxes" hard to be instrumented at source-code level to generate PML. This is the primary reason why PSW, a sort of external logger, must be deployed to intercept transactions and record events generated by services instead of modifying the services themselves to support logging. Despite this apparent limitation, PSW is still able to record provenance associated with various target systems' important functions. For example, PSW configured for database systems and service oriented workflows can easily record provenance associated with queries and Web service invocations respectively in order to provide a thorough recording of the KP associated with cyberinfrastructure applications.

3.3 IW-Base

For querying and maintaining large quantities of KP, the parsing of PML files has shown to be too expensive. Therefore, to increase scalability, certain generic provenance elements are also stored in a database known as IW- Base [11]. The result are PML documents that can reference KP elements stored in IW-Base rather than including their defintion in the PML document itself. This also alleviates PML provenance loggers (i.e., PSW) from always re-generating certain meta-data that could otherwise be shared. For example, PML documents associated with conclusions from the Java Theorem Prover might reference the Knowledge Interchange Format (KIF) provnenace element stored in IW-Base, to indicate that their resulting logical statement are encoded in KIF. Otherwise, each PML document would have to contain the redundent definition describing the KIF format. Additionally, having a centralized defintion of some elements supports interoperability when sharing KP among Inference Web tools and between Inference Web tools and other Semantic Web tools in general. Thus, IW-Base can serve as standard of defintions, for provenance elements that are commonly used. This paper proposes that an ontology can supplement the information contained in IW-Base, by providing additional semantic defintions of certain PML elements. For example, traditional PML documents associated with services that retrieve gravity data might reference the *ASCII dataset* definition in IW-Base to indicate that the dataset is in ASCII tabular format. This paper proposes that PML documents should also reference concepts in an ontology, such as *gravity data*, in order to provide a richer description of the services' outputs. In an inference Web scenario, inference engines mainly output logical statements, which semantics are provided within the statement itself, thus only the format of the statement is an issue. In a cyberinfrastructure scenario, conclusions range from datasets and reports to complex visualizations, thus associated semantic defintions of these different data becomes more necessary.

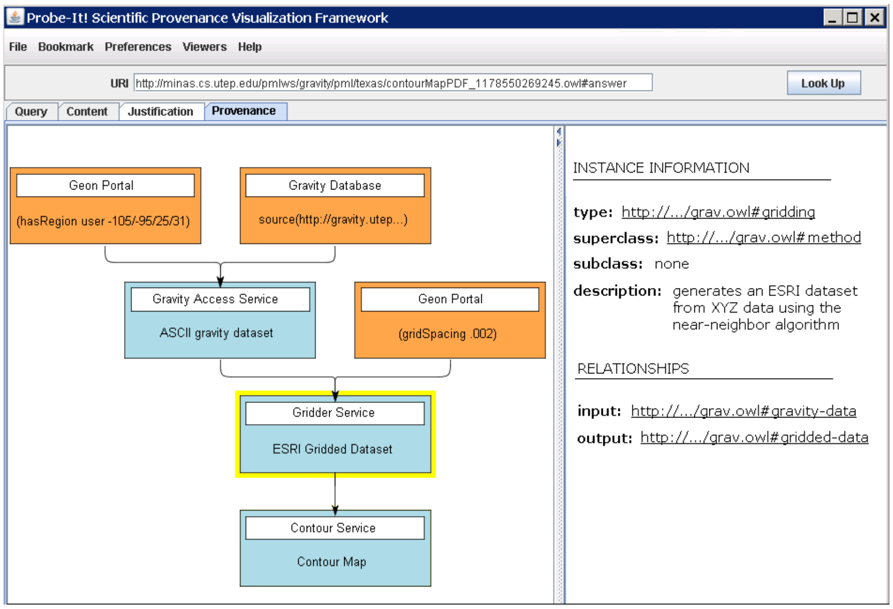
IW-Base critically depends on the IW-Base registry and IW-Base registrar. An IW-Base registrar is a collection of applications used for maintaining an IW-Base registry. From a human user point of view, the registrar is an interactive application where the user can add, update, and browse the registry contents. From a software agent point of view, the registrar is a collection of services for querying and updating the registry. The registrar is also responsible for keeping the synchronization between the registry database or provenance elements and the OWL files representing those elements.

4 Using Annotated Gravity Map

Users who store their provenance as a collection of PML documents can use Probe-It!, a KP visualization tool, to view their information. Probe-It! is capable of graphically rendering every aspect of KP associated with map generation on the cyberinfrastructure. Figure 4 illustrates the renderings provided by Probe-It! in visualizing the KP associated with a gravity contour map. The left side of the screen presents the KP associated with the execution trace visualized as a DAG. In this representation, data flow is represented by edges; the representation is such that data flows from the leaf nodes towards the root node of the DAG, which represents the final service invoked in the workflow. The DAG essentially contains two types of nodes, workflow inputs and information transformation services corresponding to the workflow inputs and invoked Web services respectively. Upon clicking on the nodes (i.e. KP elements) comprising the execution DAG, the associated semantic information is displayed on the right pane. For example, a highlighted border surrounding the gridding service node denotes that this KP item is selected, and thus the semantic information is presented. According to the gravity ontology, this service is an instance of type *gridding* inheriting from the *method* concept. Additionally, this service requires *gravity-data* as input and outputs *gridded-data*. Scientists can use this rich information to get a very good understanding of the how the map was generated in their own terminology.

5 Evaluation

The premise of our work is that KP is a valuable resource that will soon become an integral aspect of all cyberinfrastructure applications. The use of ontologies is



URI http://minas.cs.utep.edu/pmlws/gravity/pml/texas/contourMapPDF_1178550269245.owl#answer

INSTANCE INFORMATION

type: http://.../grav.owl#gridding

superclass: http://.../grav.owl#method

subclass: none

description: generates an ESRI dataset from XYZ data using the near-neighbor algorithm

RELATIONSHIPS

input: http://.../grav.owl#gravity-data

output: http://.../grav.owl#gridded-data

Fig. 4. Probe-It! Provenance Viewer.

becoming more pervasive in the sciences, however the use of KP is still being researched and its various applications are still being explored, thus a widespread adoption of KP has yet to take place. A previous study has indicated however that providing scientists with visualizations of KP helps them to identify and explain map imperfections [15]. This study was composed of seven evaluation cases all derived from the different possible errors that can arise in the gravity map scenario; each case was based on a gravity contour map that was generated incorrectly. The subjects were each asked to identify the map as either correct or with imperfections. Additionally, they were asked to explain why they identified the map as such, usually by indicating the source of error. Table 1 shows the subjects accuracy in completing the identifying and explaining tasks with a contour map that was generated using a grid spacing parameter that was too large with respect to the density of data being mapped; this causes a loss of resolution hiding many features present in the data. Without KP, the majority of scientists were not able to recognize that the map was incorrect, due to the surprisingly smooth contours resulting from the course grids. With KP and corresponding visualizations provided by Probe-It!, the scientists were able to either see the gridding parameter in the process trace or access the intermediate result associated with gridding and see the pixelated image. In either case, every category of scientists: subject matter experts (SME), Geographic Information Systems Experts (GISE), an non experts (NE), performed better collectively. This study motivates the usage of provenance information to understand complex artifacts, such as maps, generated in a distributed and heterogeneous environment such as the cyberinfrastructure. The study did not include the concept of leveraging ontologies to further annotate KP, as is discussed in this paper. We strongly believe however that adding formal semantics to the provenance will only increase the accuracy of the scientist in understanding scientific results.

Table 1. Percentage of correct identifications and explanations of map imperfections introduced by the inappropriate gridding parameter. [No Provenance (NP), Provenance (P)]

	(%) Correct Identifications		(%) Correct Explanations	
Experience	NP	P	NP	P
SME	50	100	25	100
GISE	11	78	11	78
NE	0	75	0	75
all users	13	80	6	80

6 Discussion

6.1 PML support for semantic annotations

PML node sets contain the conclusion derived as a result of applying a particular inference step. Additionally, the node set contains an element *language*, which is used to indicate the language the conclusion is encoded in; this makes more sense in a theorem proving scenario, where the result of each proof step is some logical statement encoded in a first order language such as Knowledge Interchange Format (KIF). The possible entries for this particular element are any of the languages registered in IW-Base. Similarly, the elements comprising the inference step: rule and inference engine, can be annotated with any registered entries for rules and theorem provers.

In the same way PML documents reference entries in IW-Base, they could also be adapted to reference concepts defined in an ontology, as suggested in this work. For example, the third task in the gravity map scenario outputs an ESRI gridded-dataset, thus a PML node set associated with this task would contain a dataset as it's conclusion. The corresponding language element of this PML node set could be annotated with the URI of the *gridded-data* concept contained in the gravity ontology, instead of a language registered in IW-Base; similarly, the corresponding inference engine element could be annotated with the *gridding* concept. The result is a PML document describing the gridding service of the gravity map scenario as outputting a *gridded-dataset* generated from a*gridding* service using only standard PML elements and the gravity ontology.

6.2 Pre vs. Post processing annotation

Knowledge provenance can be annotated with semantic information during workflow execution or after as a post-processing step. KP annotation during workflow execution implies that the PML service wrappers be equipped with the capability to semantically annotate PML node sets, prior to execution. As the wrappers generate the PML provenance, it can incorporate the semantic annotations. This entails that PSW be coupled with a particular ontology of some domain. At the cost of a more complex wrapper, this configuration may be the most straight forward way to annotate KP.

On the other hand, annotating the PML documents after execution of workflow provides greater flexibility; instead of PSW annotating the KP with concepts of some fixed domain, the KP can be semantically annotated by concepts of any domain, provided an ontology. Of course it would be up to a scientist to correctly associate the KP elements to concepts of some ontology. In order to automate the post annotation process, the program would require a mapping of provenance elements stored in IW-Base and instances of a some ontology. This is because standard PML only references provenance elements stored in IW-Base. In order to compliment the IW-Base entries with concepts of some ontology, a mechanism is needed to ensure that the IW-Base entries and concepts are congruent.

6.3 Annotation granularity

PSW is capable of logging most aspects of KP associated with scientific workflows including the execution trace (i.e. the sequence of services that were invoked), intermediate results, and information describing the functionality provided by each service composing the workflow. Because the gravity ontology is very detailed, every aspect of KP associated with the gravity map scenario can be semantically annotated. The gravity ontology defines concepts for all the inputs/outputs and services that comprise the gravity map workflow. Additionally, the gravity ontology defines the relationships between the data and different methods that operate on that data. Semantically annotating KP elements would not be possible if the gravity ontology were not defined with scientific processes in mind. Therefore, the level of KP that can be annotated depends upon the granularity of the ontology. If an ontology is defined at such a high level, that relationships between data and methods are not explicit, then annotation of KP elements regarding the output of each service may not be possible.

6.4 Distributed provenance (PML) vs. Workflow-level provenance

Service oriented workflows, such as the gravity map workflow can be segmented into two parts: the workflow engine or director and the services comprising the workflow activities. The workflow director is responsible for forwarding the output of each service to the next service specified in the sequence, therefore the director must know details about the services such as where they are located (i.e. what are the services endpoint URI) and what the data type of there respective input/output parameters. The services, on the other hand are not aware of the workflows they belong to; they simply execute upon request and return their results to the calling application, which may or may not be a workflow.

Just as there are two main segments composing a service-oriented workflow, there are two points from which to collect knowledge provenance. Knowledge provenance can be collected from either the workflow engine side or on the service side such as is done with PSW. Typically, systems that record KP on the workflow engine side are tightly coupled to the workflow engine itself, thus only aspects of KP visible to the workflow engine can be recorded. Kepler [2], a workflow engine, records KP on the engine side, thus information regarding the input/outputs of each service and the sequence of their execution can be logged. However, from the workflow engine side, the services composing the workflow are simply "black boxes", only their location, input and outputs are known. On the other hand, PSW and other service side KP loggers have the benefit of being closely coupled with the service they are logging and can usually provide more detailed KP regarding their functionality. Additionally, with these types of configurations, the responsibility of logging KP is removed from the workflow engine and placed on the service side.

A side-effect of service side logging however is that a layer of indirection is added between the workflow engine and the target service that performs the desired function. This overhead may be a small price to pay in order to obtain rich KP associated with a service's functionality. If PSW is wrapping a service from the "black box" perspective then the wrapper can only log very basic provenance, such as the services end-point URI. Despite this limitation, the wrapper is still able to log process meta-information and intermediate results, which at the level of single service correspond to name of the service and its output data respectively. If PSW or other service side loggers have intimate details about the services they are wrapping (i.e., the source code of the services is available) then the wrapper may be configured to capture richer provenance such as the employed algorithm or the hosting organization. In contrast, provenance captured by Kepler does not include any description or indication of the organization hosting the invoked services or their supporting algorithm because provenance is captured on the workflow side; from the point of view of the Kepler workflow engine, services are "black boxes" located at some end-point address.

Without provenance related to a service's function however, scientists may not be able to identify what algorithm was employed leading to a weaker understanding of what function the service provides and thus a weaker understanding of the quality of the final result. Although from a computer science perspective, the "black box" nature of service-oriented architecture is very beneficial, especially in terms of designing highly scalable systems, it makes it difficult to analyze the output of systems designed as such. From the study discussed in Section 5, it was determined that scientists need rich KP associated with all aspects of the workflow execution, including the algorithm supported by each service in order to fully understand complex results. Additionally, measurements such as trust that are derived from provenance can not easily be obtained through the use of workflow-side captured provenance such as provided by Kepler. For the provenance use cases outlined by Kepler developers however, the detail of provenance recorded is more than adequate. Additionally, tracing provenance in Kepler only inflicts minimal processing time penalties, because there is no level of indirection introduced between workflows and target services, as is the case in PSW.

7 Conclusions

Ontologies provide a formal definition of concepts in some domain, essentially establishing a standard vocabulary, from which both scientists and software agents can use to better understand artifacts. Knowledge provenance provides a detailed description of the origins of some artifact generated by complex processes such as scientific workflows. When used in conjunction as described in this paper, scientists are provided with very rich knowledge about some artifact, including a description of its origins defined by an ontology. This paper demonstrates how knowledge provenance is leveraged as a medium, from which rich semantics can be associated with complex artifacts such as a maps. Semantically annotating KP associated with maps, such as gravity contour maps, provides a richer description than is available when annotating only the artifact itself. Scientists need detailed information regarding the generation of artifacts in order to accurately reuse them. From the positive results achieved in the user study evaluating the need of KP, we believe that further annotating KP with semantics will only further aid scientists in better understanding and thus better utilizing complex artifacts.

Acknowledgements

This work has been partially supported by the University of Texas at El Paso GIS Center.

References

1. R. Aldouri, G.R. Keller, A. Gates, J. Rasillo, L. Salayandia, V. Kreinovich, J. Seeley, P. Taylor, and S. Holloway. GEON: Geophysical data add the 3rd dimension in geospatial studies. In *Proceedings of the ESRI International User Conference 2004*, page 1898, San Diego, CA, August 2004.
2. S. Bowers, T. McPhillips, B. Ludascher, S. Cohen, and S. B. Davidson. A Model for User-Oriented Data Provenance in Pipelined Scientific Workflows. In *International Provenance and Annotation Workshop (IPAW)*, LNCS. Springer, 2006.
3. Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and Where: A Characterization of Data Provenance. In *Proceedings of 8th International Conference on Database Theory*, pages 316–330, January 2001.
4. Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the Lineage of View Data in a Warehousing Environment. *ACM Trans. on Database Systems*, 25(2):179–227, June 2000.

5. M. Dean and G. Schreiber. OWL web ontology language reference. Technical report, W3C, 2004.
6. J. Evans. Discussion Paper: XML for Image and Map Annotations (XIMA) Draft Candidate Inferface Specificatio. http://portal.opengeospatial.org.
7. ESRI: GIS and Mapping Software. Annotation Features. http://www.esri.com.
8. Google. Google Map Features. http://maps.google.com/.
9. Deborah L. McGuinness and Paulo Pinheiro da Silva. Infrastructure for Web Explanations. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, LNCS-2870, pages 113–129, Sanibel, FL, USA, October 2003. Springer.
10. Deborah L. McGuinness and Paulo Pinheiro da Silva. Explaining Answers from the Semantic Web. *Journal of Web Semantics*, 1(4):397–413, October 2004.
11. Deborah L. McGuinness, Paulo Pinheiro da Silva, and Cynthia Chang. IW-Base: Provenance Metadata Infrastructure for Explaining and Trusting Answers from the Web. Technical Report KSL-04-07, Knowledge Systems Laboratory, Stanford University, 2004.
12. Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. Technical report, World Wide Web Consortium (W3C), February 10 2004. Recommendation.
13. J. Willian Murdock, Deborah L. McGuinness, Paulo Pinheiro da Silva, Christopher Welty, and David Ferrucci. Explaining Conclusions from Diverse Knowledge Sources. In *Proceedings of the 5th International Semantic Web Conference (ISWC2006)*, pages 861–872, Athens, GA, November 2006. Springer.
14. Paulo Pinheiro da Silva, Deborah L. McGuinness, and Rob McCool. Knowledge Provenance Infrastructure. *IEEE Data Engineering Bulletin*, 25(2):179–227, December 2003.
15. N. Del Rio and P. Pinheiro da Silva. Identifying and Explaining Map Imperfections Through Knowledge Provenance Visualization. Technical report, The University of Texas at El Paso, June 2007.
16. J. Zhao, C. Wroe, C. Goble, R. Stevens andq D. Quan, and M. Greenweed. Using Semantic Web Technologies for Representing E-science Provenance. In *Proceedings of the 3rd International Semantic Web Conference*, pages 92–106, November 2004.

This article was processed using the LaTeX macro package with LLNCS style