# Probe-It! Visualization Support for Provenance

Nicholas Del Rio and Paulo Pinheiro da Silva

University of Texas at El Paso
500 W. University Ave, El Paso, Texas, USA

**Abstract.** Visualization is a technique used to facilitate the understanding of scientific results such as large data sets and maps. Provenance techniques can also aid in increasing the understanding and thus acceptance of scientific results by providing access to information about the sources and methods which were used to derive them. Visualization and provenance techniques, although rarely used in combination, may further increase scientists' understanding of results since the scientists may be able to use a single tool to see and evaluate result derivation processes including any final or partial result. In this paper we introduce Probe-It!: a visualization tool of scientific provenance information that enables scientists to move the visualization focus from intermediate and final results to provenance back and forth. To evaluate the benefits of Probe-It!, in the context of maps, this paper presents a quantitative user study on how the tool was used by scientists to discriminate between quality results and results with known imperfections. The study demonstrates that only a very small percentage of the scientists tested can identify imperfections using maps without the help of knowledge provenance and that most scientists, whether GIS experts, subject matter experts (i.e., experts on gravity data maps) or not, can identify and explain several kinds of map imperfections when using maps together with knowledge provenance visualization.

## 1 Introduction

In complex virtual environments like cyber-infrastructures, scientists rely on visualization tools to help them understand large amounts of data that are generated from experiments, measurements obtained by sensors, or a combination of measurements and applied derivations. Instead of tediously tracing through datasets, scientists view results condensed as a graph or map, and draw conclusions from these projected views. Data visualization, however, may not be enough for scientists to fully understand and accept these results because they may need to know which data sources and data processing services were used to derive the results and which intermediate datasets were produced during the derivation process. In fact, scientists may need to have access to provenance information, which in this paper is described as meta-information about the final results and how they were generated. Provenance information includes both *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 an answer.

Provenance visualization capabilities are expected to be more sophisticated than the ones required for the visualization of results. For example, in addition to the visualization of the final results, provenance visualization should include capabilities for visualizing intermediate results, the derivation process, and any related meta-information available including meta-information about sources and services. The development of provenance visualization tools is relatively new. We observe that most provenance systems focus on capturing and managing provenance information, while most visualization systems focus on providing a robust and accurate rendering of a dataset. For the few systems that try to bridge these concerns, many research concerns arise including the need to allow scientists to keep focused on their provenance inspection tasks while they navigate through a collection of integrated views based on multiple visualization techniques.

The coupling between results and their associated provenance is inherent, thus justifying the development of a framework that can facilitate easy viewing of both. In this paper, we report our progress on Probe-It!, a general-purpose, provenance visualization prototype that has been used to visualize both logical proofs generated by inference engines and workflow execution traces generated by Kepler, a scientific workflow environment [5]. Additionally, the paper reports on a user study that confirms the need for knowledge provenance information in identifying imperfections of complex products such as maps.

## 2 Scientific Knowledge Provenance Visualization

Probe-It! is a browser suited to graphically rendering provenance information associated with results coming from inference engines and workflows. In this sense, Probe-It! does not actually generate content (i.e. logging or capturing provenance information); instead it is assumed that users will provide Probe-It! with end-points of existing provenance resources to be viewed. The task of presenting provenance in a useful manner is difficult in comparison to the task of collecting provenance. Because provenance associated with results from small workflows can become large and incomprehensible as a whole, Probe-It! consists of a multitude of viewers, each suited to different elements of provenance. Decomposing provenance into smaller more comprehensible chunks, however, raises the following questions:

1. How do scientists navigate back and forth between the visualizations of final and intermediate results (i.e., datasets and scientific artifacts such as maps) and information about the generation of such results (i.e., meta-data about the applied sources, methods, and sequencing regarding the execution of those methods).
2. How do scientists define relevance criteria for distinct provenance information and how can tools use relevance criteria to improve scientist experiences during the visualization of scientific provenance?
3. How can scientists instruct tools to present scientific provenance by defining and selecting preferences?

The following sections describe how Probe-It! addresses these concerns.

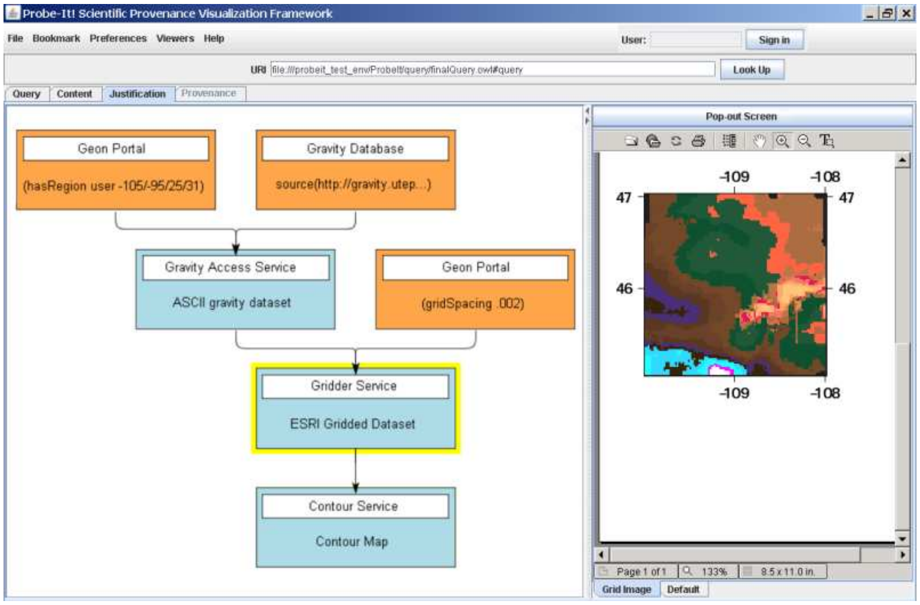
### 2.1 Results, Justifications, and Provenance

Probe-It! consists of three primary views to accommodate the different kinds of provenance information: results, justifications, and provenance, which refer to final and intermediate data, descriptions of the generation process (i.e., execution traces), and information about the sources respectively.

The *results view* provides graphical renderings of the final and intermediate results associated with scientific workflows. This view is captured on the right hand side of Figure 1, which presents a visualization of a gridded dataset. Because there are many visualizations for data sets the *results view* is dynamic and determined by the preference of a particular user. The framework supporting this capability is described in more detail in Section 3.

The *justification view*, on the other hand, is a complimentary view that contains all the process meta-information associated with the execution trace, such as the requests, the functions invoked by the workflow, and the sequencing associated with these invocations. Probe-It! renders this information as a directed acyclic graph (DAG). An example of a workflow execution DAG can be found on the left hand side of Figure 1, which presents the *justification view* of Probe-It! This view is supposed to provide a global view of the execution by graphically rendering the execution trace. In the justification view, data flow is represented by edges of the DAG; the representation is such that data flows from the leaf nodes towards the root node of the DAG. For the sake of keeping the DAG description simple, in the case of Web services there exist two types of nodes, inputs labeled as "direct assertions" and information transformation services labeled as "Generic Web Service." Workflow inputs may have been provided by a user, software agent, or data sink, and have no incoming edges into their nodes. Information transformation services are represented by the internal nodes of the DAG, and thus have one or more incoming edge representing data input. These services may have outgoing edges representing output data that may be consumed by other services (i.e., this is indicated by an arrow being fed into another node). Each node contains a label indicating the name of the invoked service. The DAG root node represents the final service executed by the workflow, generating thus the workflow results. The type of each Web service result is made explicit by a label appended to each node; the data is described at a semantic level by labels such as "gravity dataset" rather than at a syntactic level by labels such as "ASCII table". Nevertheless, provenance encoding informs when gravity datasets are encoded as ASCII tables. For example, in Figure 1, the "GEON Gridder Service" has the label "ESRI Gridded File" to indicate that the output of that service was an ESRI gridded dataset.

The *provenance view*, provides information about sources and some usage information e.g., access time, during the execution of an application or workflow. For example, upon accessing the input node labeled *gravity database*, meta-information about the database, such as the responsible organization, is displayed in another panel. Similarly, users can access information transformation nodes, and view information about used algorithms, or the hosting organization.



URI file:///probeit_test_env/ProbeIt/query/finalQuery.owl#query

**Fig. 1.** Probe-It! justification view.

### 2.2 Navigating Between Views

**From Query View to Justification View:** In a highly collaborative environment such as the cyberinfrastructure, there are often multiple applications published that provide the same kind of service. A thorough integrative application may consider all the different ways it can generate and present results to users, placing the burden on users to discriminate between high quality and low quality results. This is no different from any question/answer application, including a typical search engine on the Web, which often uses multiple sources and presents thousands of answers back to users. Probe-It!'s query view visually shows the links between application requests and results of that particular request. Each result is visualized as a node (similar to the nodes in the justification view). The request rendering is labeled with a textual representation of the request. Upon accessing one of the answer nodes, Probe-It! switches over to the justification view associated with that particular result.

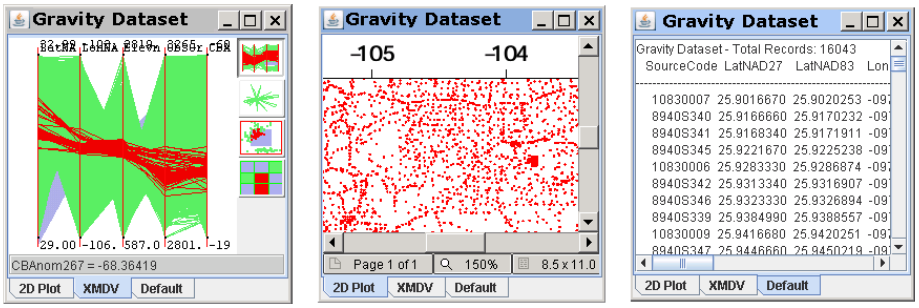
**From Justification View to Results View:** Once the justification view has been accessed, provenance associated with the generation of the selected answer can be visually accessed. As discussed earlier, information associated with the execution trace is rendered as a DAG that serves as a medium between provenance meta-information and process meta-information. Upon accessing a *Generic Web Service* node, the results view presents the output data associated with that service in a viewer, as discussed in Section 2.3.

A typical visualization scenario requiring usage of provenance views involves a scientist using a Web portal to request an artifact such as a map. Upon completion, the portal invokes Probe-It! which shows the available results, i.e. maps, that could be generated satisfying the constraints provided in the request. Upon selecting a particular map, the justification is presented to the scientist. The scientist can now access the intermediate results and understand their contribution to the final map.

**Comparing Knowledge Provenance:** It was anticipated that scientists might not quickly realize what particular map is best suited for their needs. Just as there may be many answers from search engines, each based on different source data, there may be many ways to derive a map in a cyber-infrastructure. In many cases, scientists may need to compare the provenance associated with each map in order to decide which map best fits their needs. To facilitate such comparisons, results of a workflow can be popped out in separate windows. The pop-up capability provided by the tool is useful when comparing both final and intermediate results of different maps. Users can pop-up a visualization of intermediate results associated with one map, navigate to the justification of a different map and pop-up a window for the corresponding results, i.e., results of the same type, for comparison purposes. In addition to the result that is being viewed, pop-up windows contain the ID of the artifact from which it is associated. This allows users to pop-up several windows without loosing track of what artifact the pop-up window belongs to.

### 2.3 Result Viewers and Framework Support for Visualization Techniques

Upon selecting a node in a DAG, a viewer associated with the result of the selected node will be presented in the results view window. However, there are many kinds of results associated with scientific workflows that must be appropriately rendered by Probe-It!'s results view. In a Web browser scenario, incoming data is tagged with a MIME-TYPE, which is associated with a particular plug-in. Probe-It! 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 as well. For example, the XMDV visualization tool as presented on the left hand side of Figure 2, is a visualization suited to any N dimensional data. The data rendered by XMDV need only be in a basic tabular format, as shown on the right hand side of Figure 2, with a few additional headers indicating the min/max values of each column. The provenance associated with data of this kind of would indicate that it in



```
Gravity Dataset - Total Records: 16043
  SourceCode  LatNAD27    LatNAD83    Lon
--------------------------------------------
   10830007  25.9016670  25.9020253  -09
   8940S340  25.9166660  25.9170232  -09
   8940S341  25.9168340  25.9171911  -09
   8940S345  25.9221670  25.9225238  -09
   10830006  25.9283330  25.9286874  -09
   8940S342  25.9313340  25.9316907  -09
   8940S346  25.9323330  25.9326894  -09
   8940S339  25.9384990  25.9388557  -09
   10830009  25.9416680  25.9420251  -09
   8940S347  25.9446660  25.9450219  -09
```

**Fig. 2.** Three different viewers for gravity data sets.

an *ascii tabular* format. However, this kind of data is also semantically defined as being *gravity point data*, in which case Probe-It! would invoke the more appropriate 2D spatial viewer, as shown in the center of Figure 2. The semantic capabilities provided by the Probe-It! viewer framework compliments the MIME tables used in typical Web browsers, which only indicate the format or syntax of the data.

In many cases, Probe-It! will have to rely on existing applications to provide a rendering, much like how a typical Web browser relies on plug- ins to render non-HTML content. In order to manage the many relationships between a kind of data and the appropriate renderer, Probe-It relies on a MIME-like table to store these mappings. This table contains all the known data types, their semantic descriptions, and their respective renderers. Thus, an appropriate renderer for a particular kind of data is based on both the data's format and semantic description. The property that makes this MIME-like table so desirable for the Probe-It is its extendibility; scientists can register new mappings on request, keeping Probe-It! up-to-date with the scientists' needs.

Different visualizations model the data from different perspectives thus it is important to provide the scientists with as many views as possible. Although scientists have their preferred views, Probe-It! still allows the scientists to access the other available viewers. For example, gravity datasets have three associated visualizations: default textual view, location plot view, and XMDV view. The default textual view is essentially a data table, the raw result from gravity database. Figure 2 shows a pop-up of both the location plot view and the XMDV view. The location plot visualization provides a 2D plot of the gravity reading in terms of latitude and longitude. XMDV provides a parallel coordinates view, a technique pioneered in the 1970's, which has been applied to a diverse set of multidimensional problems. Additionally, the ESRI gridded dataset has associated colored image visualization, also presented in Figure 2 as well as a textual default view, another tabular structure.

## 3 Underlying Technologies

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

Inference Web [6, 7] provides the Proof Markup Language (PML) [11] for encoding and publishing provenance information on the web as well as tools and services for handling PML documents. PML is an OWL [1, 9] based language that can be used for encoding provenance. PML consists of a specification of terms for encoding collections of justifications for computationally derived results. Depending upon the application domain, users may view each justification as an informal execution trace or as a proof describing the inference steps used by an inference engine, e.g., theorem prover or web service, to derive some conclusion. Regardless of the application domain, PML is composed of many different elements which describe the various elements of automatically generated proofs. The highest level element defined in PML is the node set, which contains both a conclusion (i.e., a logical expression) and a collection of inference steps each of which provide a different derivation 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 antecedent, 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 proof. Thus PML proofs are graphs with node sets as nodes and antecedents acting as edges. This proof 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.

Alternatively, the structure of PML proofs can be used to store provenance information associated with scientific applications such as workflows. From this perspective, node sets represent the execution of a particular web service; the node set conclusion serves as the output of the service while the inference step represents the provenance associated with the function provided by a service or application. For example, the inference step proof elements antecedent, rule, and the inference engine can be used to describe the applications inputs, function, and name respectively. The relationship between PML proofs and provenance associated with workflows is so strong, it is suggested that provenance be used as a proof for the result of scientific workflows.

IW-Base is a repository of provenance elements that can be reference by PML documents. [8]. In order to support interoperability when sharing provenance among Inference Web tools and between Inference Web tools and other Semantic Web tools in general, elements in the registry are stored as PML files. Thus, with the use of semantic web tools, one can retrieve, parse, and use proof-related meta-data. For querying and maintaining large quantities of meta-data, the parsing of PML files has shown to be too expensive. Therefore, to increase scalability, provenance elements are also stored in a database. The result is PML documents that contain many references to provenance elements stored in the database, alleviating PML provenance loggers from always generating and re-generating

PML files that can be shared. For example, a PML document describing the provenance associated with task one in the Gravity Map scenario would reference the provenance element *ASCII DATASET* stored in IW-Base, to indicate that the resulting dataset is in an ASCII tabular format. Many services published on the cyber-infrastructure that also return ASCII datasets are relieved of having to generate the PML file that defines ASCII dataset.

### 3.2 PML Service Wrapper (PSW)

PML Service Wrapper (PSW) is a general-purpose wrapper that logs knowledge provenance associated with workflow executions as a set of PML documents. Since workflows can be composed entirely of Web services, PSW logs workflows at the level of service invocations and transactions. Thus, information such as the input/output of each service and meta-information regarding the used algorithm are all logged by PSW.

In traditional Inference Web applications [10, 7], inference engines are instrumented to generate PML. However in a cyber-infrastructure setting, reasoning 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 service and workflows themselves to support logging.

## 4 Evaluation

The effectiveness of provenance visualization in the task of understanding complex artifacts was verified by a user study described below. The context of the user study is presented first, following with a brief discussion of how provenance and visualization aided scientists in the evaluation tasks.

### 4.1 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 cyber-infrastructure 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 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] http://irpsrvgis00.utep.edu/repositorywebsite/

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 one of the four tasks above is realized by a web service. This set of web services would be piped or chained together; the output of one service would be forwarded as the input to the next service specified in the sequence or workflow. This sequence of piped services or *workflow* serves as the foundation for the Gravity Map Scenario. The following Section describes how this scenario served as a test-bed to evaluate the effectiveness of Probe-It! in aiding scientists to both identify and explain imperfect maps generated by the gravity map workflow.

### 4.2 Results

The premise of our work is that scientific provenance is a valuable resource that will soon be become an integral aspect of all cyber-infrastructure applications. The use of provenance is still being researched and its various applications are still being explored, thus a widespread adoption of provenance has yet to take place. A previous study of ours has indicated that providing scientists with visualizations of provenance helps them to both identify and explain map imperfections [12]. 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 incorrectly generated. 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 provenance, 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 provenance 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), both SME and GISE (SME+GISE), 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 cyber-infrastructure.

**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 |
| SME+GISE | 50 | 100 | 50 | 100 |
| NE | 0 | 75 | 0 | 75 |
| all users | 13 | 80 | 6 | 80 |

## 5 Related Work

VisTrails, a provenance and data exploration system provides an infrastructure for systematically capturing provenance related to the evolution of a workflow [2]. Provenance information managed by Vistrails refers to the modifications or history of changes made to particular workflow in order to derive a new workflow; modifications include, adding, deleting or replacing workflow processes. VisTrails renders this *history of modifications* as a treelike structure where nodes represent a version of some workflow and edges represent the modification applied to a workflow in order to derive a new workflow. Upon accessing a particular node of the provenance tree, users of VisTrails are provided with a rendering of the scientific product which was generated as a result of a particular workflow associated with the node.

Only workflows that generate visualizations are targeted by Vistrails, however the authors describe how this system could be transformed to handle the general case. This notion is the foundation of Probe-It; to provide a framework that can manage and graphically render any scientific result ranging from processed datasets to complex visualizations.

MyGrid, from the e-science initiative, tracks data and process provenance of some workflow execution. Authors of MyGrid draw an analogy between the type of provenance they record for cyber-infrastructure type applications and the kind of information that a scientist records in a notebook describing where, how and why results were experimental results were generated [14]. From these recordings, scientists can achieve three primary goals: (i) debugging, (ii) validity checking, and (iii) updating, which refer to situations when, a result is unexpected, when a result is novel, or a workflow component is changed respectively. The Haystack application displays the provenance as a labeled directed graph, tailored to a specific user; only relevant provenance elements related to the role of a user are rendered in order to reduce data overloading on the screen. In this scenario, links between resources are rendered allowing users to realize the relationships between provenance elements such as inputs/outputs and applied processes thus realizing the execution trace.

The Earth Science System Workbench (ESSW) is another effort at capturing and presenting scientific results to users [3]. ESSW is a client/server architecture in which a workflow (client) transmits provenance information to services which manage the provenance, similar to the PSW described in Section 3 ref¿¿. Recorded provenance is rendered by [4] in the form of a directed graph, where nodes are data objects and edges define relationships between objects.

In contrast to graphically displaying scientific provenance, the Kepler workflow design and execution tool provides an interface for querying recorded provenance associated with workflow execution via a set of predefined operators. In this case, provenance is queried, with the result of the query being some relation. Similarly, Trio, a management system for tracking data resident in databases, tracks data as it is projected and transformed by queries and operations respectively [13]. Because of the controlled and well understood nature of a database, lineage of some result can many times be derived from the result itself by applying an inversion of the operation that derived it. These inverse transformations of the data are stored in special table and made available via querying capabilities.

## 6 Conclusions and Future Work

The research presented in this paper is based on the fundamental problem of developing tools and methods that can help scientists understand complex scientific products (e.g., datasets, reports, graphs, maps) derived from complex software systems (e.g., applications and services) deployed on a distributed and heterogeneous environments such as cyber-infrastructures. The work has been developed in the context of a realistic scenario based on ongoing cyber-infrastructure efforts in the fields of Earth Sciences. A user study driven by this scenario verified the effectiveness of provenance visualization in helping scientists understand complex artifacts, strengthening the notion that provenance should be maintained by all cyber-infrastructure applications, and available on demand in some useful representation.

Since the effectiveness of provenance has been demonstrated, the strategy will be to present scientists with the most effective ways of browsing such information. The current evaluation approach was based on the suitability of provenance in decision making scenarios, rather than the usability of the tool itself. Usability is based on the evaluation of many dimensions including learnability, understandability, and handling ability. Each of the aforementioned aspects refer to the amount of time of necessary training before independent use of a system is possible, ability of users to correctly draw conclusion from display, and the speed of a trained user respectively. The next step is to develop a more formal model of how users interact with provenance visualizations in order to improve the usability of Probe-It!

## References

1. M. Dean and G. Schreiber. OWL web ontology language reference. Technical report, W3C, 2004.

2. Juliana Freire, Claudio T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo. Managing Rapidly-Evolving Scientific Workflows. In *Proceedings of the International Provenance and Annotation Workshop (IPAW)*, 2006. (to appear).
3. J. Frew and R. Bose. Earth System Science Workbench: A Data Management Infrastructure for Earth Science Products. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management*, pages 180–189, Fairfax, VA, July 2001.
4. AT&T Research Labs. AT&T Graphiz. http://www.graphviz.com.
5. B. Ludašcher and et al. Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience*, 2005. Special Issue on Scientific Workflows.
6. 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.
7. Deborah L. McGuinness and Paulo Pinheiro da Silva. Explaining Answers from the Semantic Web. *Journal of Web Semantics*, 1(4):397–413, October 2004.
8. 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.
9. Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. Technical report, World Wide Web Consortium (W3C), February 10 2004. Recommendation.
10. 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.
11. Paulo Pinheiro da Silva, Deborah L. McGuinness, and Richard Fikes. A Proof Markup Language for Semantic Web Services. *Information Systems*, 31(4-5):381–395, 2006.
12. 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.
13. Jennifer Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research*, pages 262–276, Asilomar, CA, January 2005.
14. 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