On the Use of Semantic Abstract Workflows Rooted on Provenance Concepts

Leonardo Salayandia and Paulo Pinheiro da Silva

University of Texas at El Paso, Computer Science Department,
El Paso, Texas 79968, USA
{leonardo,paulo}@utep.edu
http://www.cs.utep.edu

Abstract. Two challenges related to capturing provenance about scientific data are: 1) determining an adequate level of granularity to encode provenance, and 2) encoding provenance in a way that facilitates end-user interpretation and analysis. A solution to address these challenges consists in integrating two technologies: Semantic Abstract Workflows (SAWs), which are used to capture a domain expert's understanding of a scientific process, and PML, an extensible language used to encode provenance. This paper describes relevant features of these technologies for addressing the granularity and interpretation challenges of provenance encoding and presents a discussion about their integration.

Keywords: Process, Provenance, PML, Semantic Abstract Workflows.

1 Introduction

Semantic Abstract Workflows (SAWs) are useful to encode process knowledge from the perspective of domain experts [1] and the Proof Markup Language (PML) is useful to encode justifications about how information is produced [2]. This paper describes the integration of SAWs and PML, which results in two benefits: 1) Given that determining an adequate level of granularity to encode provenance is challenging [3], i.e, provenance at a very fine level may not be scalable and provenance at a very coarse level may not be useful, process knowledge captured from the perspective of domain experts serves as a guide to determine an adequate level of granularity; 2) Provenance languages such as PML utilize specialized terminology that may be unfamiliar to end users. The integration of these technologies has the benefit of having domain-specific terminology used to refer to a domain expert's understanding of a process that can be propagated to refer to provenance knowledge as well.

This paper is organized as follows: Section 2 presents SAWs and PML, Section 3 presents how these technologies are integrated, Section 4 presents a discussion about the integration, and Section 5 concludes the paper.

D.L. McGuinness, J.R. Michaelis, and L. Moreau (Eds.): IPAW 2010, LNCS 6378, pp. 216–220, 2010.

2 Background

2.1 Semantic Abstract Workflows

SAWs capture flows of information from information sources of a process, through information transformation activities, and finally to information sinks at the end of that process. An initial phase in creating a SAW is to have domain experts identify and name the types of information and types of activities involved in their processes. For example, the information type *Digital Elevation Map* is different from the information type *Gravity Map* for a geophysicist because each type of map models different properties of interest about a region of Earth. In contrast, both maps could be represented as PDF files, and therefore, be considered of the same information type from a programmer's point of view; however, this type classification would not yield a SAW that captures the point of view of the process' domain expert, i.e., the geophysicist. Activity types are similarly identified from the point of view of the domain expert, where `Method` is the preferred term used to refer to discrete activities included in processes that transform information, i.e., transform information from one type to another. Methods can be software driven, such as the type of application used to transform a dataset to a map, or human driven, such as the type of activity performed to analyze a model to obtain an interpretation.

SAWs do not contain constructs to represent control flow such as order of execution, selection, and iteration. As a result, one SAW can model different implementations of a process. Another characteristic of SAWs is that path traversals are suggestive in nature. Specifying an information type flowing from one method type to another means that it is conceivable, in the view of the domain expert, that such flow can occur. In executing a process implementation, however, that information flow may or may not happen.

2.2 Proof Markup Language

PML defines primitive concepts and relations for representing provenance about data. Two essensial modules of PML are: 1) The *provenance ontology* (PML-P) that defines concepts to represent identifiable things from the real world that are useful to determine data lineage; and 2) the *justification ontology* (PML-J) that defines concepts and relations to represent dependencies between identifiable things.

The foundational concept in PML-P is `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. Two key subclasses of `IdentifiedThing` motivated by provenance representational concerns are `Information` and `Source`. `Information` supports references to information at various levels of granularity and structure. `Source` refers to an information container, and it is often used to refer to all the information from the container. For example, things such as organization, person, agent, and service can be a `Source`. PML-P provides a simple but extensible taxonomy of sources.

PML-J provides concepts and relations used to encode information manipulation steps used to derive a conclusion. A justification requires concepts for representing conclusions, conclusion antecedents, and information manipulation steps used to transform or derive conclusions from antecedents. The justification vocabulary has two main concepts: `NodeSet` and `InferenceStep`. A `NodeSet` includes structure for representing a conclusion and a set of alternative `InferenceSteps` each of which can provide an alternative justification for a conclusion. Every `NodeSet` has a unique web-addressable identifier, i.e., a URI. Web-addressable `NodeSets` make it possible to construct justification trees in a distributed environment.

3 Integrating Process and Provenance Concepts

The ontology behind the encoding of SAWs is called Workflow-Driven Ontology (WDO) [4]. This ontology defines the generic concepts of `Information` and `Method` that domain experts specialize to capture the terminology about their processes. The ontology of provenance-related concepts used in PML is the PML-P ontology. PML-P includes the concepts of `Information`, `MethodRule`, and `Source` that are specializations of the more generic concept `Identified Thing`. The WDO and PML-P ontologies are aligned to take advantage of the process knowledge that domain experts capture through SAWs to encode provenance from a process implementation. The main alignment involves the merger of the WDO concept of `Information` with the PML-P concept of `Information`, and the substitution of the WDO concept of `Method` for the PML-P concept of `MethodRule`. The ontology alignment also includes the use of the PML-P concept `Source` as the sources and sinks used in SAWs. Sources and sinks as used in SAWs are conceptually equivalent, except for the flow of information, i.e., sources produce information and sinks receive information. Sources in PML, however, are used through another PML-P concept denominated `SourceUsage`, which records information about the date and time of source access. This is important in provenance encodings because sources, e.g., websites and documents, may change over time and the provenance encodings may lose validity. Date/time source access is not necessary for process knowledge.

4 Discussion

SAWs are designed to model a domain expert's understanding of a process. As such, SAWs can be used to identify provenance use cases that can be obtained from the implementation that the domain expert uses to carry out a process. This approach is offered by the WDO-It! tool, a Java-based editor for SAWs [5]. WDO-It! includes functionality to generate PML-encoding modules based on the activities identified in the process. These modules are used to instrument a process implementation to intercept and interlink intermediate results as a process is being executed, effectively capturing provenance in PML about the resulting artifact.

Abstract process knowledge encoded in SAWs is useful to present provenance at a manageable level of detail to the end user. For example, the execution of a process may involve many iterations of a cycle, and hence, the resulting provenance tree may be cumbersome to interpret and analyze. Given that SAWs do not contain control-flow information, SAWs are an effective canonical representation of a process with respect to the methods involved. What is more, since processes encoded in SAWs use the same ontological concepts used to encode provenance in PML, method and data types included in SAWs can be used to filter provenance trees with respect to specific parts of a process.

The integration of SAWs and PML results in controlled vocabularies created by domain experts to encode process knowledge that are also useful to formulate provenance-related queries and to present provenance for browsing by users familiar with the domain of discourse.

With respect to related work, [6] presents an approach that consists of inferring a schema-level summary of the possible concrete provenance graphs that could be generated from an executable workflow specification. The result is an abstract provenance graph that could be used to facilitate the analysis of data flow as it relates to a workflow specification. In this sense, abstract provenance graphs are similar in nature to SAWs. However, levels of abstraction provided by the two approaches differ. With SAWs, the approach consists in having the scientist model their understanding of a process as a graph and using the terminology that is specific to the problem domain. With abstract provenance graphs, the approach consists on creating an executable specification of the scientist's process first and then generating the abstraction from it. On one hand, abstract provenance graphs will result in a level of abstraction that is less close to the problem domain but that is tightly integrated to a specific execution environment. On the other hand, SAWs will result in a level of abstraction that closely relates to the problem domain, however, additional manual work is needed to map that level of abstraction to specific implementations of the process. An additional benefit of SAWs is that they can be mapped to multiple implementations of the same process, or even be mapped to manual systems where processes are human driven instead of software driven.

5 Conclusion

This paper presented an integration of technologies used to capture process and provenance knowledge through the alignment of their underlying ontologies. Two main benefits are that provenance is encoded at a granularity that suits the level of detail documented by domain experts, and that provenance is encoded using domain expert's defined concepts, which facilitates subsequent querying and analysis of provenance. The integration of these technologies is implemented through the WDO-It! tool [5], and an approach named CI-Miner [7] has been developed to guide domain experts to document their processes and to construct provenance-capturing modules that can be used to instrument process implementations. The latest version of the aligned ontology can be found at http://trust.utep.edu/2.0/wdo.owl.

This work was funded in part by the National Science Foundation (HRD-0734825) and the Department of Homeland Security (2008-ST-062-000007).

References

1. Pinheiro da Silva, P., Salayandia, L., Del Rio, N., Gates, A.Q.: On the Use of Abstract Workflows to Capture Scientific Process Provenance. In: 2nd Workshop on the Theory and Practice of Provenance (TaPP 2010), San Jose, CA (2010)
2. McGuinness, D., Ding, L., Pinheiro da Silva, P., Chang, C.: PML2: A Modular Explanation Interlingua. In: AAAI 2007 Workshop on Explanation-aware Computing, Vancouver, British Columbia, Canada (2007)
3. Stephan, E., Halter, T., Critchlow, T., Pinheiro da Silva, P., Salayandia, L.: Using Domain Requirements to Achieve Science-Oriented Provenance. Late Breaking Contribution Poster, to appear in IPAW (2010)
4. Salayandia, L., Pinheiro da Silva, P., Gates, A.Q., Salcedo, F.: Workflow-Driven Ontologies: An Earth Sciences Case Study. In: 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, Netherlands (2006)
5. WDO-It!: An editor for Worflow-Driven Ontologies, `http://trust.utep.edu/wdo`
6. Zinn, D., Ludaescher, B.: Abstract Provenance Graphs: Anticipating and Exploiting Schema-Level Data Provenance. In: 3rd International Provenance and Annotation Workshop, Troy, NY (2010)
7. Pinheiro da Silva, P., Salayandia, L., Gandara, A., Gates, A.Q.: CI-Miner: Semantically Enhancing Scientific Processes. Earth Science Informatics 2(4), 249–269 (2009)