On the use of Abstract Workflows to Capture Scientific Process Provenance

Paulo Pinheiro da Silva, Leonardo Salayandia, Nicholas Del Rio, Ann Q. Gates
Overview

- Ontologies and Abstract Workflow to document scientific processes
- The Proof Markup Language (PML) to encode data provenance
- Capturing provenance about scientific processes
- Other efforts
- Conclusions
Documenting Scientific Processes with Ontologies and Abstract Workflows

- **Purpose**
  - Identify appropriate vocabulary for a scientific community
  - Model a scientist’s understanding of a process
  - Identify the parts of a process that are of interest to scientists

- **Benefits**
  - Share scientist’s understanding of a process with others
  - Guide the development of systems that implement scientist’s understanding of a process
  - Enhance existing systems to provide functionality aligned to scientist’s understanding of a process
Phase 1: Capture the vocabulary of the process in a Workflow-Driven Ontology (WDO)

- WDOs have two main classes:
  - **Data**, e.g., Gridded Dataset, Elevation Map
  - **Method**, e.g., Nearest-neighbor extrapolation

- Tool support to construct WDOs
  - Encoded in OWL
  - Reuse vocabulary from other OWL ontologies
  - Generate HTML reports
Phase 2: Model the process as a Semantic Abstract Workflow (SAW)

- Dataflow modeling
- Graphical representation
- Multiple levels of abstraction supported
- Tool support to create SAWs
  - Encoded in OWL
  - Generate HTML reports
  - Generate provenance-capturing modules
Documenting Scientific Processes with Ontologies and Abstract Workflows

- WDOs and SAWs are intended to be authored by Scientists
  - Scientist-centered level of abstraction
  - Dataflow modeling intended to facilitate process modeling
Documenting Scientific Processes with Ontologies and Abstract Workflows

- Some efforts where WDOs and SAWs are being used
  - Environmental data collection at
    - La Jornada Experimental Range
    - The arctic region (Barrow, Alaska)
  - Seismic refraction experiments at Potrillo mountains
Encoding Provenance with PML

- **Proof Markup Language (PML)**
  - Derived from the theorem proving community
  - Divided into three parts:
    - PML-Provenance
    - PML-Justification
    - PML-Trust

With respect to provenance
Encoding Provenance with PML

- Distributed provenance
  - NodeSets generated by distributed components
  - NodeSets linked through Web conventions

Diagram:

- NodeSet 1: Encoded by software at Laboratory
  - URI: http://...

- NodeSet 2: Encoded by software at Data Center
  - URI: http://...

- NodeSet 3: Encoded by field instrumentation
  - URI: http://...

- NodeSet 1 hasAntecendent NodeSet 2
- NodeSet 2 hasAntecendent NodeSet 3
- NodeSet 2 hasAntecendent NodeSet 1
Capturing Scientific Process Provenance

- The framework:
  - Process and Provenance ontology alignment
    - WDO: Identify things that can be used to document how things **can happen** (i.e., process)
    - PML-P: Identify things that can be used to document how things **happened** (i.e., provenance)
Capturing Scientific Process Provenance

The framework:
- WDO reuses concepts from the PML-P ontology
- WDO adds properties to the concepts from PML-P
- WDO vocabulary can be used for Provenance queries!

Vocabulary identified by scientist to document process:

Used to query provenance:
Select NodeSets that have an antecedent of type GravityDataset
Capturing Scientific Process Provenance

- The process of capturing provenance:

1. Encode vocabulary for scientific process
2. Model scientific process as an abstract workflow
3. Capture Provenance about Data that is generated using the scientific process from above; there are two ways:
   - (a) Create Data Annotators (wrapper modules) and use them to enhance a scientific system to capture provenance
   - (b) Use the scientific process as a template to manually link artifacts to capture provenance

Goal: Facilitate provenance encoding in PML
Capturing Scientific Process

**Provenance**

- **Automated scientific systems**
  - Use process knowledge to generate data annotator modules
    - Instrument system to call data annotators to record provenance during execution
      - E.g., C-shell scripts
    - Use data annotators after system execution to construct provenance from logs/temp files generated by the system
      - E.g., field data-gathering instruments with proprietary software and extensive logging features
Capturing Scientific Process Provenance

- Manual scientific systems
  - Tool support to encode PML using process knowledge as template:

Technical Report
Manually entered parameters
Other Efforts

- Provenance Query
  - Build RDF triple stores from PML encodings
  - SPARQL queries
- Provenance Visualization
  - Probe-It!
Conclusions

- Abstraction is used to comprehensively document scientific processes
- Encoding provenance in PML is not straight-forward, but tools can help
- Not all scientific processes are implemented as software systems
- This approach to document provenance may not be scalable for all systems, but it is useful for some:
  - Scientists building custom systems to gather data
Thank you!
Encoding Provenance with PML

- More details about PML
  - Divided into three parts:
    - PML-Provenance
    - PML-Justification
    - PML-Trust