Introduction

W3C-PROV (PROV) helps to standardize the encoding of data life cycle provenance. PROV statements capture the entities, agents, and actions, along with persistent identifiers, involved in data production, dissemination, and post-processing. A so-called *provenance trace* can give us powerful ways of accessing the history and lineage of data. Yet, data and information management efforts such as the Global Change Information System[1] and recent efforts in the USGS to produce a Biogeographic Information System are often working retrospectively to curate provenance[2] information and capture annotations about scientific data and information assets. Once the provenance traces are retrospectively assembled there still exist challenges in creating a human-readable presentation.

This project looked at these provenance challenges from both sides. First, we created a prototype system for capturing data life cycle provenance as it happens. Second, we created a generic visualization service that can graphically display complete provenance traces irrespective of dataset. The combination of provenance capture and visualization services demonstrates a complete end-to-end system over the entire spectrum of provenance usage - from submission and capture to end-user exploration and discovery. ***The goal of this project is not to create a production ready end-to-end system. Rather, by working with a prototype implementation of existing standards, we aim to identify impedance points. An impedance point is any point along the data lifecycle in which provenance generation/capture is ill-defined or ambiguous due to lack of standards and/or loosely defined specifications.***

Groups such as the DataONE, CSIRO, and Geoscience Australia have existing work and prototypes in both provenance capture and visualization. Work from these groups is also used highlight the multiple impedance points along the provenance spectrum. A key deliverable of this project is the use our end-to-end prototype to explore all of the areas in which provenance can be described, submitted, searched, and visualized using varying techniques. To the best of our knowledge, previous provenance research has chosen practical implementations based off requirements of specific projects and organizations. In this project, we want to take a step back and look at the provenance workflow from a distributed and generic perspective. Using our prototype end-to-end system, we highlight all of the places in which the provenance specifications are open to interpretation, in flux, or provide multiple implementation suggestions. From this examination we provide suggested recommendations and identify key areas in which the Earth science community needs to discuss best practices for practical cross-organization implementations of PROV.

Provenance Submission and Querying

The W3C Provenance Access and Query (PROV-AQ) Working Group is developing specifications that define how to locate, retrieve, and query provenance records. The key features of PROV-AQ include:

1. Simple mechanisms for retrieving and discovering provenance records.
2. Provenance query mechanisms that may be used for more demanding deployments
3. A simple "ping-back" mechanism allowing for discovery of additional provenance that would otherwise be unknown to the publisher of the resource (e.g. provenance about future entities that are based upon or influenced by a resource)

Feature 3 is key to this project. The “ping-back” mechanism can be used to register related provenance information that the data creator does not otherwise know about; e.g. provenance describing

[1] https://data.globalchange.gov/
[2] https://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance#A_Working_Definition_of_Provenance

how it is used after it has been created. A “ping-back” feature extends the capabilities of PROV by addressing questions such as:

- What new resources are based on this resource?
- What has this resource been used for?
- Who has used it?
- What other resources are derived from the same sources as this resource?

We note that the PROV-AQ specification is ambiguous in places regarding how such a ping-back can be implemented. Similar variance is included in the PROV specification as well. While this ambiguity is intentional and allows provenance to be extended into a number of domains, it makes generic cross-organizational provenance systems difficult to implement. It is completely feasible that PROV documents, submission, and query systems could all be developed that conform to the W3C specs, but can not be used together across the Earth sciences.

In this document we look at provenance creation, submission, and querying from both the data provider and end-user perspectives. We highlight the multitude of places where ambiguity challenges interoperability. At times we offer suggested paths forward. Other times we highlight discussion points for the Earth science community.

Data Provider Perspective

1. Advertising a PROV-AQ Pingback service

PROV-AQ is based on Linked Data and Semantic Web principles. In the Linked Data world, all entities have a unique identifier (a URI). This identifier is resolvable - meaning that it can be put into a web browser and it will return semantic data. The PROV-AQ spec stipulates that when a user resolves a URI (for example a dataset identifier) the returned information should be the semantic data on the resource (e.g. the dataset) and also additional links stating that the resource has provenance and a service exists for submitting additional provenance[3].

This works very well within the Semantic Web context. However, we note that many data systems within the Earth sciences are not yet semantically enabled. Moreover, an individual researcher may not be familiar with URIs and Semantic Web constructs. We propose an additional means of advertising PROV-AQ services based on identifiers more familiar to Earth science researchers.

Proposed additional mechanism for advertising PROV-AQ

Data providers should be free to set up their own URLs based on familiar identifier schemes. For instance, the example service URL may be: http://my.org/pingback/ and it may accept identifiers according to REST principles such as: http://my.org/pingback/Identifier where *Identifier* could be a DOI, an ARK, or similar resource identifier familiar to the community. An example execution of this service would be: http://my.org/pingback/doi:10.1037/rmh0000008, which would request provenance on the dataset with DOI **doi**:10.1037/rmh0000008. The

[3] See Examples 1 and 2 at https://www.w3.org/TR/prov-aq/

science community would interact with the service using HTTP, REST, and familiar identifier schemes (e.g. DOI, ARK, etc.) and it would be up to the service provider to hide the semantic complexity by translating between DOI (or other identifier) and URI.

We also recommend two approaches for advertising such services - one human readable and one machine readable. We suggest data providers advertise the ping-back service in dataset landing pages for human consumption while simultaneously providing a JSON-LD machine consumable markup within the same page. Specifically, we recommend the UpdateAction[4] concept from Schema.org to be compatible with existing landing page markup efforts.
An example of JSON-LD markup from the Open Core project follows with the PROV-AQ advertisement highlighted in red. Note the use of the “identifier” property to indicate which identifier types this service understands.

```
{
        "@context": "http://schema.org",
        "@type": "WebSite",
        "@id": "http://opencoredata.org",
        "url": "https://www.opencoredata.org/",
        "description": "opencoredata is a data ...",
        "potentialAction": {
                "@type": "SearchAction",
                "@id": "http://opencoredata.org#searchAction",
                "target": "https://www.opencoredata.org/search?q={search_term_string}",
                "query-input": "required name=search_term_string"
        },
        "potentialAction": {
                "@type": "UpdateAction",
                "@id": "http://opencoredata.org#updateAction",
                "target": "http://my.org/pingback/{identifier}",
                "query-input": "required name=identifier",
                “identifier”: “DOI”,
                “identifier”: “ARK”
        },
```

[4] http://schema.org/UpdateAction

```
        "@context": {
                "@vocab": "http://schema.org/",
                "re3data": "http://example.org/re3data/0.1/"
        },
        "@type": "Organization",
        "@id": "http://opencoredata.org/id/facilityinfo",
        "name": "Open Core Data",
        "re3data:datalicense": "http://opencoredata.org/datapolicy.html",
        "contactPoint": {
                "@type": "ContactPoint",
                "@id": "http://opencoredata.org/id/facilityinfo#contactPoint",
                "name": "Douglas Fils",
                "email": "dfilsATоceanleadershipDOTorg",
                "url": "http://orcid.org/0000-0002-2257-9127",
                "contactType": "technical support"
        },
        "url": "http://www.opencoredata.org",
        "sameAs": "http://www.re3data.org/repository/r3d100012071"
}
```


2. Suffix URL issue for linked data

For those that are working with Linked Data, the PROV-AQ spec mandates that every resource URI have a ping-back service. For example, given two datasets with URIs http://acme.example.org/dataset1 and http://acme.example.org/dataset2, respectively, the PROV-AQ spec mandates that the data provider support http://acme.example.org/dataset1/pingback to receive provenance data for dataset 1 as well as http://acme.example.org/dataset2/pingback to receive provenance data for dataset 2. Architecturally, this makes sense and allows for the provenance to be easily assigned to the appropriate resource. However, it places a burden on system developers and administrators to maintain such a scheme. This may not be much of an issue at present as not many fully compliant Linked Data PROV-AQ systems are expected to exist. Yet, moving forward, it is a potential impedance point and should be revisited. The Earth science community may be better served to accept, as a community, a simple service URL scheme like the one depicted above for JSON-LD.

3. Generating provenance records

Provenance can exist in many encodings. The W3C PROV Data Model (PROV-DM) provides the generic structure of provenance and offers multiple encodings for this structure such as XML, XML-RDF, and Turtle. We recommend the Earth science community adhere to semantic documents conforming to the PROV-O spec, even if underlying systems are not fully Linked Data compliant. We also recommend that all data providers and organizations supporting PROV-AQ adhere to

1. Providing a PROV-O description of how each of their datasets were generated.
2. Using rdfs:label throughout their PROV documents. The W3C Provenance Data Model (PROV-DM) stipulates that each provenance item have a label. PROV-O suggests that rdfs:label be used to implement such a label; however, the use of rdfs:label is not required. We suggest that it should be required and that the Earth science community accept rdfs:label as a required best practice
3. Every provenance document returned from a PROV-AQ system should utilize the notion of prov:Bundle. In PROV-O terminology, a Bundle is a collection of related provenance statements, the grouping of which may have provenance itself. PROV-AQ systems should return all provenance as a prov:Bundle providing return date, time, and associated service information as provenance of the bundle.

** For an example of what this looks like in practice, see the /prov directory in our github repository.

The PROV-O model contains the concepts of Person, SoftwareAgent, and Organization to begin to standardize descriptions of who is acting on/with a dataset. However, these concepts do not have standardized properties. For example, some people use the Friend-of-a-Friend (foaf) vocabulary to specify name, email address, etc. while others use Dublin Core concepts to do the same thing. This is an area in the Earth science community needs to have more discussion to avoid multiple dialects of provenance that are only understandable within an organization and via specialized tools.

An additional challenge exists in what we are referring to as the many dialects of provenance. PROV-O provides base concepts that are intended to be extended within domains of usage. This provides for a lot of flexibility. It also limits interoperability when multiple sub-communities within the Earth sciences create their own extensions. The exchange of provenance across organizations is no longer feasible. We are aware of at least four such “dialects” of provenance within the Earth sciences: PROV-M, PROV-ONE, OGC, and PROV-ES.

Fortunately, these four dialects serve distinct purposes. PROV-ES is a generic extension of PROV-O with Earth science specific Entities and Actions, such as Dataset, Instrument, and ProcessStep. The OGC model (more formally defined in the Closa et al. paper[5]) adds geospatial concepts such as Feature, Point, and Polygon. PROV-ONE, originating from the DataONE community, is primarily aimed at scientific workflows and focuses on how outputs of one activity become the inputs of another activity. PROV-ONE provides more specificity for inputs, outputs, and workflow configuration. PROV-M is focused on system reporting and specializes terms such as Document and Report.

We have two recommendations in this area.

1. All provenance documents within the Earth science ping-back environment should adhere to one of these dialects. In other words, data providers should not just use the PROV-O base concepts of Entity and Activity. Rather, data providers should provide additional subclasses to indicate which dialect they are using

   For example, use

   ```
   :dataset
           a prov:Entity, eos:product;
           rdfs:label "Some dataset from USGS"^^xsd:string;
   ```

   Instead of

   ```
   :dataset
           a prov:Entity;
           rdfs:label "Some dataset from USGS"^^xsd:string;
   ```

   Where eos:product refers to the Product concept in PROV-ES, which is a subclass of prov:Entity

We believe the W3C Shapes Constraint Language[6] (SHACL) can be used to detect which dialect is being used (and validate its conformity), although we have not tested this yet.

2. All attempts should be made to conform to existing dialects of PROV-O. Any new implementations should first be vetted via open forum - potentially via ESIP.

4. Receiving subsequent provenance records via pingback

[5] http://www.sciencedirect.com/science/article/pii/S0198971517300558
[6] https://www.w3.org/TR/shacl/

The PROV-AQ spec stipulates that provenance of dataset usage and derivative datasets should be sent (ping-backed) as URIs. In other words, one does not send the entire provenance record. Instead, one simply sends a URI, which at a future point can be dereferenced to retrieve the full provenance description. This raises two important issues.

1. Submitting only a URI requires that the full provenance document be stored online indefinitely awaiting deference. We find this infeasible within the Earth science domain. Individual researchers who create derivative datasets, or use existing datasets, likely do not have the resources to host the provenance records they create. We see two options
    a. We modify the PROV-AQ spec such that data users can submit full provenance documents and not have to host them
    b. The Earth science community commits to building and maintaining a provenance hosting service. This service would serve as an intermediary between data users and providers. A user would generate provenance on how a dataset was used. This provenance would then be uploaded to the provenance hosting service, which would return a dereferencable URI to that provenance. The user would then send the received URI as a pingback to the PROV-AQ service.

2. PROV-AQ stipulates that a service is not required to do anything with the URIs it receives. Further, the spec does not indicate in which form a PROV-AQ system should return provenance. It simply says that the user will get back “provenance”. We suggest a standardization of this for the Earth sciences. When queried for a dataset a PROV-AQ service should
    a. Return the provenance of how the dataset was created along with each URI it received via pingback. The URIs should be enclosed within a prov:Collection. Adhering to 3.3 above, all of this provenance should be wrapped in a prov:Bundle to indicate its origin and date of creation.

** For an example of what this looks like in practice, see the /prov directory in our github repository.

User’s Perspective

A major area of concern is how to incentivize provenance generation. It’s immediately obvious why data providers and agencies would want to implement PROV-AQ to track dataset usage. It’s less clear what would motivate dataset users to send pingbacks to data providers. The Earth science community needs to begin thinking about a concerted effort to make provenance generation seamless and transparent. Provenance pingback should be embedded in common tools and research practices.

We also believe the Earth science community should come to a consensus on the scope of a PROV-AQ service. Technically, any action involving an Earth science dataset should generate a provenance record. Yet, we don’t believe all of these provenance records should be

sent back to data providers via ping-backs. For example, using a dataset to generate a figure for an AGU poster qualifies as "usage"; however, we find it unlikely that data providers want this level of granularity in recording downstream usage of their data. Moreover, we think other best practices such as dataset citation are better suited for this type of use case. This is not to say other types of provenance should not be captured. We fully support workflow provenance and efforts such as the Global Change Information System in linking figures and tables in reports back to their source datasets. These types of provenance capture systems are important. Yet, they should exist independent of PROV-AQ services. In order to maximize uptake, we suggest limiting the initial scope of PROV-AQ services to

1. Notifying a data provider regarding the creation of a new derivative dataset that is publicly available to the community
2. Notifying a data provider regarding errors and usage issues in a dataset

We believe 1. can be standardized using existing provenance dialects such as PROV-ES and PROV-ONE. We suggest standardizing 2 by using the Annotation Ontology[7]. See the /prov directory in our github repository for an example of an Annotation ping-back.

Visualization Service

Our prototype visualization service leverages **PROV-O-VIZ (**http://provoviz.org/) a tool by Paul Groth, one of the main PROV authors. It generates Sankey diagrams from PROV-O Turtle documents. Our implementation is available at http://gator.ndm.edu/narock/provisium/prov_viz.php and takes one input parameter "turtle", which is the URL of the PROV-O Turtle file to be visualized. For example:

http://gator.ndm.edu/narock/provisium/prov_viz.php?turtle=http://gator.ndm.edu/narock/provisium/provenance_response_example.ttl

Our service extends the PROV-O-VIZ visualization by also parsing the Turtle file, extracting the ping-back URIs and displaying these as an HTML table under the visualization. The source code for the visualization service can be found in the viz_service/ directory of our github repository.

Additional Recommendations

In order to completely describe the ping-back process we needed to create additional PROV-O terms. Rather than create yet another ontology, we propose these terms as extensions to PROV-ES. Specifically, we defined (in Turtle notation)

[7] https://www.w3.org/TR/annotation-vocab/

```
eos:pingBackService
   a prov:Activity, eos:PingBackService;
   rdfs:label "An instance of a Ping-Back service"^^xsd:string
.
```

Where eos is the PROV-ES namespace. In plain English, sending a ping-back is an instance of a PROV-O Activity, which itself should be captured. To accommodate this, we define eos:PingBackService, which is a subclass of prov:Activity and we are able to record the act of receiving ping-backs in our provenance records.

The eos:PingBackService can be attributed to an organization via a statement such as prov:wasAttributedTo :usgs;

We also define a special prov:Collection to indicate the collection of ping-backs our service has received.

```
:pingBacks
        rdfs:label "URIs submitted to the pingback service"^^xsd:string;
        a prov:Collection, eos:PingBackCollection;
        prov:wasGeneratedBy :pingBackService;
        prov:wasDerivedFrom :dataset;
```

As highlighted in red, we state that the ping-back URIs are a prov:Collection as well as an eos:PingBackCollection (which is a subclass of prov:Collection).

Finally, we would also recommend that ESIP take over the development of PROV-ES. PROV-ES began, in part, within the Semantic Web Cluster of ESIP. To the best of our knowledge, the PROV-ES ontology is no longer under active development and does not have a permanent namespace. We believe PROV-ES should reside within ESIP via the ontology portal and have active community development much like SWEET.