Geophysical Research Abstracts
Vol. 18, EGU2016-1558, 2016
EGU General Assembly 2016

Persistent identifiers for web service requests relying on a provenance ontology design pattern

Nicholas Car (1), Jingbo Wang (2), Lesley Wyborn (2), and Wei Si (2)

(1) Geoscience Australia, Canberra, Australia (nicholas.car@ga.gov.au), (2) National Computing Infrastructure, Canberra, Australia

Delivering provenance information for datasets produced from static inputs is relatively straightforward: we represent the processing actions and data flow using provenance ontologies and link to stored copies of the inputs stored in repositories. If appropriate detail is given, the provenance information can then describe what actions have occurred (transparency) and enable reproducibility.

When web service-generated data is used by a process to create a dataset instead of a static inputs, we need to use sophisticated provenance representations of the web service request as we can no longer just link to data stored in a repository. A graph-based provenance representation, such as the W3C's PROV standard, can be used to model the web service request as a single conceptual dataset and also as a small workflow with a number of components within the same provenance report. This dual representation does more than just allow simplified or detailed views of a dataset's production to be used where appropriate. It also allow persistent identifiers to be assigned to instances of a web service requests, thus enabling one form of dynamic data citation, and for those identifiers to resolve to whatever level of detail implementers think appropriate in order for that web service request to be reproduced.

In this presentation we detail our reasoning in representing web service requests as small workflows. In outline, this stems from the idea that web service requests are perdurant things and in order to most easily persist knowledge of them for provenance, we should represent them as a nexus of relationships between endurant things, such as datasets and knowledge of particular system types, as these endurant things are far easier to persist.

We also describe the ontology design pattern that we use to represent workflows in general and how we apply it to different types of web service requests. We give examples of specific web service requests instances that were made by systems at Australia's National Computing Infrastructure and show how one can 'click' through provenance interfaces to see the dual representations of the requests using provenance management tooling we have built.