ProvONE: extending PROV to support the DataONE scientific community

Yang Cao*[1], Christopher Jones[2], Víctor Cuevas-Vicenttín[3], Matthew B. Jones[2], Bertram Ludäscher[1], Timothy McPhillips[1], Paolo Missier[4], Christopher Schwalm[5], Peter Slaughter[2], Dave Vieglais[6], Lauren Walker[2], Yaxing Wei[7]

Abstract. The DataONE federated data network has adopted and extended the PROV model to support the collection, storage, indexing, and user browsing of the provenance of data packages stored in its member nodes. The PROV extension, ProvONE, adds provenance elements (entity and relationships types) for describing process structure alongside the data dependencies that originate from process execution. The ProvONE model was defined in 2015 and its specification is available online [1], while provenance support based on the model is now (2016) in its pre-production phase.

1 DataONE

DataONE (Data Observation Network for Earth) is a large, federated data network for open, persistent, robust, and secure access to Earth observational data [2]. DataONE's primary goals include support for: data discovery, access, integration, and synthesis; education, training, and building community; and data sharing. The DataONE infrastructure consists of three principal components: *Member Nodes* (MN) represent existing or new data repositories that support the DataONE Member Node API; *Coordinating Nodes* (CN) serve the coordination and discovery needs of the network; and the *Investigator Toolkit* which contains tools that enable programmatic interaction with DataONE infrastructure through a REST service API exposed by the CNs and MNs. *DataONE Search* is a web-based application that lets users search across space (geographical region), time, and using a set of keywords, and discover publicly accessible data packages that hare hosted on any of its MN.

2 Provenance in DataONE

DataONE collects and manages the provenance of its datasets, in order to facilitate the reproducibility of those datasets and provide attribution of scientific results to community members, as well as a way to document scientific processes and their executions. To achieve these goals, DataONE introduces *ProvONE* [1], a proper extension of the PROV model that allows for descriptions of process structure and their association to

* [1]University of Illinois, Urbana-Champaign, [2]National Center for Ecological Analysis and Synthesis, UCSB, [3]Universidad Popular Autónoma del Estado de Puebla, Mexico, [4]School of Computing Science, Newcastle University, UK, [5]Woods Hole Research Center, Falmouth, MA, [6]University of Kansas, Lawrence, [7]Environmental Sciences Division, ORNL, TN.

runtime provenance (data dependencies). Building upon this extended model, DataONE supports the semi-automated recording of provenance from its Investigators' Toolkit, specifically the R and MatLab clients, by tracing these clients' interactions with the DataONE API, i.e., data read and write requests. These provenance documents, as well as other ProvONE-compliant documents that may have been generated independently by client applications, are then automatically indexed into the CNs and exposed through the DataONE Web front-end, where users may visually move along the dependency graph as they explore data packages of interest.

3 ProvONE

The ProvONE OWL specification [1] builds upon a previous incarnation, D-PROV [5], to extend PROV with process structure elements and with specific Entity types. A UML depiction of ProvONE is shown in Fig. 1.

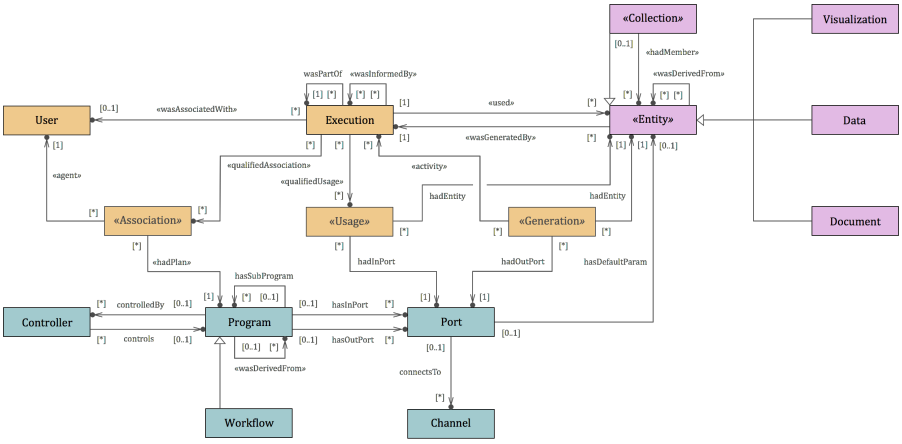

Fig. 1: UML depiction of ProvONE

ProvONE introduces three commonly used `prov:Entity` types, namely `provone:Visualization`, `provone:Data`, and `provone:Document`; it also introduces `provone:User` as a specific type of `prov:Agent`. Crucially, ProvONE accounts for a description of static process structure by extending `prov:Plan` with `provone:Program` as well as its own extension `provone:Workflow`, and by introducing new Entity types `provone:Port` and `provone:Channel`. The model is designed to accommodate in particular dataflow-type structures, where Programs have input and output Ports, which are connected by Channels, and Program nesting is expressed using the `provone:hasSubProgram` relationship. ProvONE extensions connect to PROV through the `prov:WasAssociatedWith` relationship, where `provone:Program` participate as a type of `prov:Plan`.

Process execution is described by `provone:Execution`, an extension of `prov:Activity` that can participate in `prov:WasAssociatedWith`,

`prov:Used` and `prov:WasGeneratedBy` relationships, and also admits a new `provone:wasPartOf` relationship.

By connecting the `prov:Plan` extensions with `provone:Execution`, ProvONE supports rich provenance descriptions where data usage, derivation, and generation are explicitly linked with the programs that produce and consume the data, through observation of data bindings on Program ports and of data flow along Channels.

4 DataONE support for ProvONE: infrastructure and tooling

The ProvONE model features in DataONE infrastructure and tooling, in three main ways. Firstly, ProvONE provides R and MatLab developers with user libraries [6,3] to enable provenance recording during program execution. The provenance recorder captures all interactions with the DataONE REST API, which include reading and writing data packages from and to a MN. The resulting provenance document at the end of a script execution includes references to all data packages consumed and produced by the script. Importantly, these references use the DataONE-assigned data identifiers. Thus, future reuse of the same data packages that occurs through DataONE API clients, will again reference the data packages using the same identifiers, thus enabling tracking through multiple generations of data derivation.

Secondly, developers may use a program markup tool, called *YesWorkflow* [4], to annotate their scripts (regardless of the language) and obtain a visual rendering of their structure, similar to that of a workflow (hence the name of the tool). The structural description that is compiled from the annotations is a ProvONE document. An example of annotated script and of its visual rendering are shown in Fig. 2 and 3, respectively.

```
%% @begin initialize_Grass_Matrix
%  @out Grass @as Grass_variable

%% Initialize Grass Matrix
Grass=zeros(ncols,nrows);
for i=1:ncols
    for j=1:nrows
            Grass(i,j)=sum(frac(i,j,20:28))*0.5+sum(frac(i,j,4
    end
end
%% @end initialize_Grass_Matrix


%% @begin examine_pixels_for_grass
%  @in Tair @as Tair_Matrix
%  @in Rain @as Rain_Matrix
%  @out C3 @as C3_Data
%  @out C4 @as C4_Data

%% Algorithm 1: method used in MstMIP
% examine the type of each pixel to see if it includes grass
C3=ones(ncols, nrows)*(-999.0);
C4=ones(ncols, nrows)*(-999.0);
for i=1:ncols
    for j=1:nrows
        frac_c3=0.0;
        frac_c4=0.0;
        if (Grass(i,j)>0)
```

Fig. 2: A fragment of YW-annotated MatLab script.

Finally, the DataONE search facility provides simple visualisation of the data derivations through the scripts, enabling step-by-step user navigation of potentially complex provenance graphs. Fig. 4 shows a screenshot from the DataONE UI.

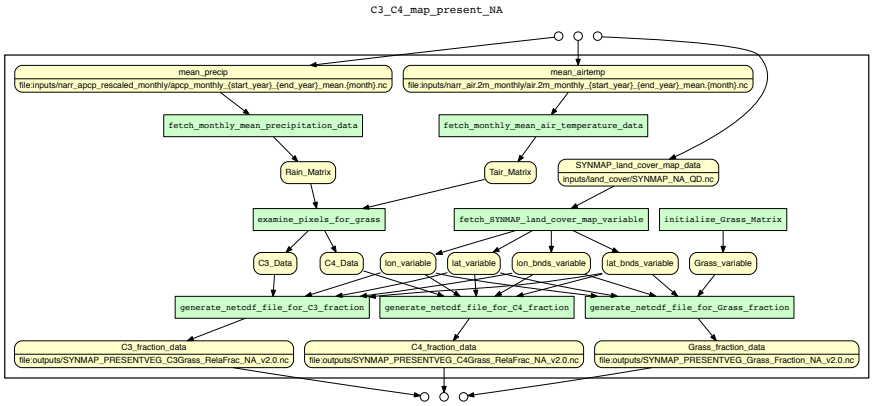

Fig. 3: A rendering of the complete script from Fig. 2.

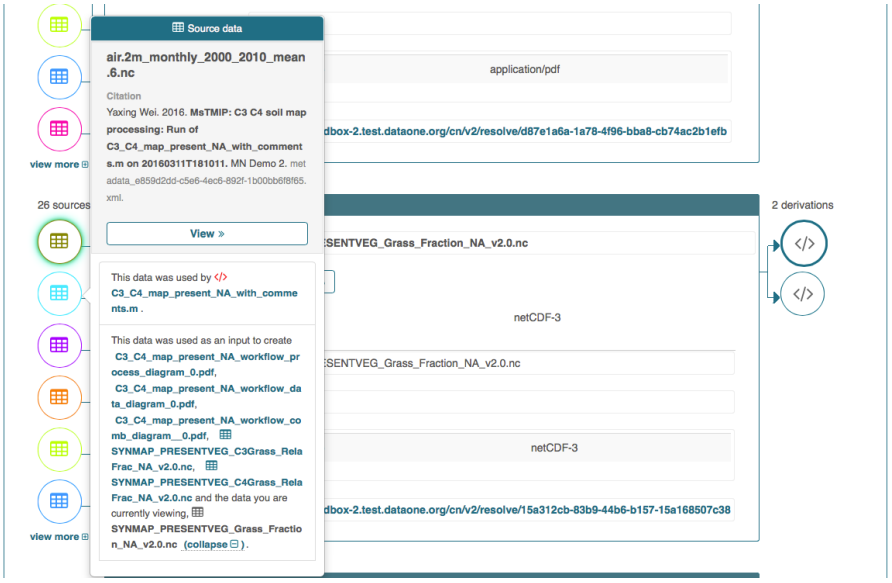

Fig. 4: Rendering in the DataONE UI of data derivations through scripts

Acknowledgements. This work was supported in part by NSF awards ACI-1430508 and ABI-1262458 in the US.

References

1. Cuevas-Vicenttín, V., et al.: ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance (2015)
2. Data Observation Network for Earth (DataONE). www.dataone.org and search.dataone.org
3. Jones, C., Cao, Y., Slaughter, P., Jones, M.B.: MATLAB DataONE Toolbox. https://github.com/DataONEorg/matlab-dataone (2016)
4. McPhillips, T., Song, T., Kolisnik, T., Aulenbach, S., Belhajjame, K., Bocinsky, K., Cao, Y., Chirigati, F., Dey, S., Freire, J., Huntzinger, D., Jones, C., Koop, D., Missier, P., Schildhauer, M., Schwalm, C., Wei, Y., Cheney, J., Bieda, M., Ludäscher, B.: YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts. International Journal of Digital Curation 10, 298–313 (2015)
5. Missier, P., Dey, S., Belhajjame, K., Cuevas, V., Ludaescher, B.: D-PROV: extending the PROV provenance model with workflow structure. In: Procs. TAPP'13. Lombard, IL (2013)
6. Slaughter, P., Jones, M.B., Jones, C.: recordr: Provenance tracking for R. https://github.com/NCEAS/recordr (2016)