Comparing Provenance Data Models for Scientific Workflows: an Analysis of PROV-Wf and ProvOne

Wellington Oliveira[1, 2], Paolo Missier[3], Daniel de Oliveira[1], Vanessa Braganholo[1]

[1]Instituto de Computação, Universidade Federal Fluminense (UFF), Brazil
[2]DACC, Instituto Federal do Sudeste de Minas Gerais – Rio Pomba Campus, Brazil
[3]School of Computing Science, Newcastle University, UK

{wellmor, danielcmo, vanessa}@ic.uff.br, paolo.missier@ncl.ac.uk

Abstract. *Scientific workflows rely on provenance to be understandable, reproducible and trustworthy. Nowadays, there is a growing demand for interoperability between provenance data generated from heterogeneous workflow management systems. To address this issue, some provenance models have been proposed by extending PROV to support specific requirements of scientific workflows. In this paper, we present two prominent provenance models for scientific workflows, PROV-Wf and ProvOne, which are specializations of PROV, and compare their elements and relationships. Our goal is to provide an overview of each one and to support the choice for the most suitable for a specific context.*

1. Introduction

With the popularization of the Workflow Management Systems (WfMS), many *ad-hoc* provenance models emerged. Their goal was to represent provenance that was captured during the workflow execution, allowing scientists to compare different workflow executions, their parameters, and input data. Since different WfMS capture provenance using different perspectives, the provenance models of two different WfMS are never the same. To make matters worse, different WfMS use different storage models such as relational databases, RDF, XML, and specific models such as Virtual Data Model (VDL) [Foster *et al.* 2002].

Well aware of this scenario, the scientific community organized the *Second Provenance Challenge*[1] to discuss a standard provenance model. The result was the *Open Provenance Model* (OPM) [Moreau *et al.* 2008] that was implemented for many WfMS aiming at reaching provenance interoperability. Later, covering some gaps of OPM and adding new relationships, the W3C Provenance Working Group[2] proposed the PROV[3] model.

PROV provides a generic data model (*i.e.*, PROV-DM) to outline provenance. However, it is not able to represent information about the workflow structure itself in the scientific experiment domain (prospective provenance [Freire *et al.* 2008]). Its focus is solely on retrospective provenance [Freire *et al.* 2008]). Thus, new specializations

[1] http://twiki.ipaw.info/bin/view/Challenge/SecondProvenanceChallenge
[2] https://www.w3.org/2011/prov/wiki/Main_Page
[3] https://www.w3.org/TR/prov-dm/

were created to bridge this gap. Initiatives like PROV-Wf [Costa *et al.* 2013] and ProvONE [Missier *et al.* 2013] introduced new elements and relations to PROV to represent the prospective provenance and their links to retrospective provenance, much needed in this context. However, each model has approached prospective provenance in different ways. This may pose some difficulties to integrate provenance from WfMS that use different models to represent provenance. For example, one could ask "*Can I map PROV-Wf Program entity directly to ProvONE Program entity?"* If no, "*is there another entity or relationship that represents the same element in both models?"*

Since both prospective and retrospective provenance are fundamental for analyzing workflow evolution and results, scientists should be able to identify a model that better represents their provenance dataset or map their elements for exchanging purposes. In this paper, we describe these provenance models, map and compare their elements. We chose ProvONE and PROV-Wf models because they are intended for representing and exchanging provenance. Our goal is to show the similarities and differences among them and raise some issues about their representations that can help scientists to make a decision about which one is more suitable to represent their provenance dataset or guide a provenance integration process.

The paper is structured as follows. Section 2 provides a background about provenance data models. The PROV-Wf and ProvONE models are described in Section 3 and Section 4, respectively. Section 5 presents a mapping between those models. Finally, Section 6 provides final remarks about the mapping and discusses future work.

2. Provenance Types and Models

The term *data provenance* can be defined as the source or lineage of the data and it can be used to interpret and reproduce the results of scientific experiments [Freire *et al.* 2008]. Especially for experiments modeled as scientific workflows, provenance can be classified as *prospective* and *retrospective* [Freire *et al.* 2008]. *Prospective* provenance (henceforth called just *p-prov*) represents the specification of computational tasks. It corresponds to the steps to be followed to achieve a result. *Retrospective* provenance (henceforth called just *r-prov*) consists in a structured and detailed history of the execution of computational tasks (metadata associated to the execution of activities and environment characteristics).

In 2006, a discussion about standard provenance representations at the *International Provenance and Annotation Workshop* (IPAW) culminated with the creation of the *Provenance Challenge*, aiming at verifying and comparing existing provenance representations. Since the *Second Provenance Challenge*, the community began to investigate interoperability issues. That culminated with the OPM 1.0, later extended to OPM 1.1 after the *Third Provenance Challenge*[4]. Most participants of the OPM initiative joined an effort at W3C to develop another general provenance model to serve as a W3C standard. It was called PROV, and it became a W3C recommendation in April 2013. A comparison of these models was conducted a couple of years ago [Bivar *et al.* 2013].

[4] http://twiki.ipaw.info/bin/view/Challenge/ThirdProvenanceChallenge

PROV-DM is the data model of PROV. It is an agnostic conceptual model and it can be applied to model different domains. It incorporates a core structure, composed by *entity, agent,* and *activity* elements and their relationships *WasDerivedFrom, WasInformedBy, Used*, *WasGeneratedBy*, *WasAssociatedWith, WasAttributedTo*, *ActedOnBehalfOf*. These elements and relationships are shown in Figure 1. PROV has also extended elements such as (i) *subtyping*, including software agent (from agent element) and *revision* (from entity element); (ii) *expanded relations*, including activity association and other *n-ary* relationships; (iii) *optional identification* that can identify an instance of an association of two or more elements; and (iv) *new relations*, consisting of subtypes or expanded versions of existing ones. The complete PROV-DM structure is composed of six components: (i) entities or activities; (ii) derivations of entities; (iii) agents; (iv) bundles; (v) properties; and (vi) collections.

As stated before, in the scientific workflows context provenance can be classified as p-prov and r-prov. PROV model is able to represent r-prov elements and their relationships, but there is just a *Plan* entity to represent p-prov aspects. In this way, new representation models such as PROV-Wf and ProvONE emerged aiming to extend PROV, adding elements and relationships to represent p-prov and r-prov in the context of scientific workflows. In the next sections, we describe these two conceptual models, PROV-Wf and ProvONE that have been proposed to extend PROV adding new domain elements and cover the p-prov representation gap.

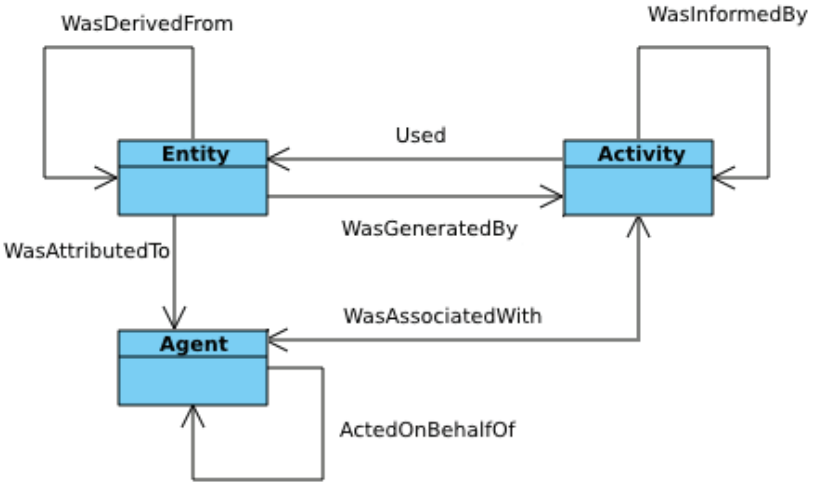

Figure 1. PROV core structures[5]

3. PROV-Wf

PROV-Wf is a conceptual model for the representation of p-prov and r-prov generated from scientific workflows [Costa *et al.* 2013]. PROV-Wf is a specialization of the W3C PROV model, which is designed to be a generic model for representing provenance in a diversity of domains different from scientific experiments (arts, industry, *etc*.). Because PROV is a generic model, it is far from trivial for several users to map its elements to the scenario of scientific experiments. PROV-Wf was thus proposed aiming at specializing PROV to the scientific experimental scenario by providing specific

[5] https://www.w3.org/TR/prov-dm/#core-structures

elements for this context. PROV-Wf can explicitly define what information is captured and how it is stored.

PROV-Wf is agnostic about the environment and the WfMS and works with a set of elements that, according to Costa *et al.* [2013], can be classified into three main types: (i) Structure of the Experiment; (ii) Execution of the Experiment; and (iii) Environment Configuration. The Structure of the Experiment element is formed by the set of planning objects (*Workflow* and *Activity*) and entities (*Program, Field, Relation, Value,* and *Value Type*). Figure 2 shows this structure as blue rectangles. The Execution of the Experiment element consists of entities (*File*, and *Domain Data*) and activities (*Execute Workflow, Execute Activity, Execute Extractor,* and *Program Invocation*) shown as dark yellow rectangles in Figure 2. Finally, the Environment Configuration includes the *Machine* and *Scientist* agents, shown in Figure 2 as light grey rectangles. The elements *Domain Data* and *Execute Extractor* were added in an extension of PROV-Wf by de Oliveira *et al.* [2015] to represent domain-specific data.

In the PROV-Wf model, a *Workflow* class is composed by *Activities* that have one or more *Programs* and *Fields* defined by a *Value Type*. The *Execute Workflow* class represents an execution of a *Workflow* and it is performed in a *Machine* that can have many *Execute Activities*. The *Scientist* class is responsible for controlling a *Program Invocation* and *Execute Activity* that can have many *Execute Extractor* classes to capture and store *Domain Data* from *Files* (specializations of *Value Type*).

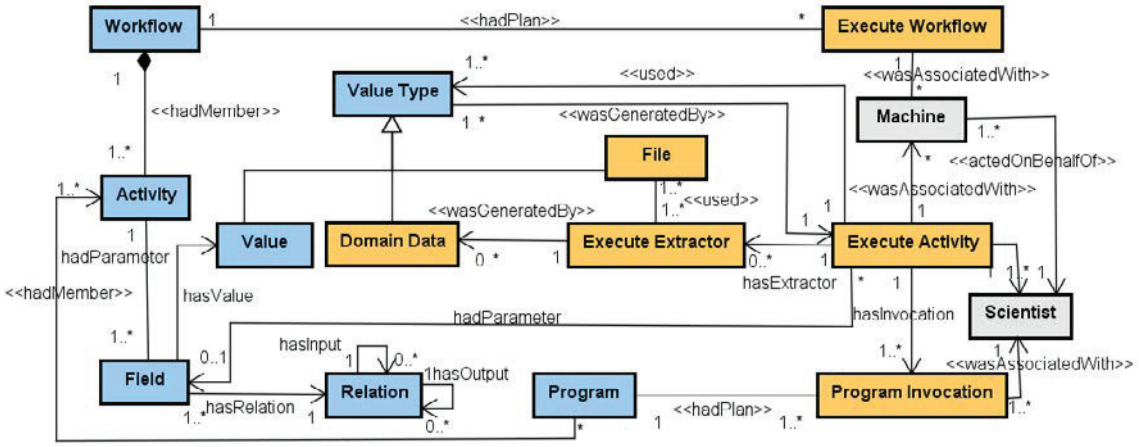

Figure 2. PROV-Wf conceptual model. Adapted from [De Oliveira *et al.* 2015]

4. ProvONE

ProvONE also extends the PROV model with an explicit representation of p-prov, thus capturing the most relevant information on scientific workflow processes, and is designed to accommodate extensions for specific scientific workflow systems [Missier *et al.* 2013]. It is fairly comprehensive including both p-prov and r-prov and allows for easy integration of terms from external vocabularies, including Dublin Core or WfMS. ProvONE is also fairly stable, and supported by a large data conservation project,

DataONE[6]. Figure 3 shows the classes and relationships that compose the ProvONE conceptual model.

In the ProvONE model, a *Workflow* class is a specialization of a *Program* class that can have sub-programs (*hasSubProgram*), versions (*wasDerivedFrom*), and can be manage by (*controlledBy*) a *Controller*. Each *Program* can also have a plan (*Association* with *User* and *Execution*) and *Ports* that may be linked to (*connectTo*) *Channels*. The *Port* class can have parameters (*hasDefaultParam*) represented by *Entities* (*Documents, Data,* and *Visualizations*) that may be consumed (*Used*) or produced by (*wasGeneratedBy*) an *Execution*. Finally, the *Execution* class can be linked to (*wasAssociatedWith*) a *User* and an input (*Usage – hadInPort*) or output (*Generation - hadOutPort*) *Port*. Figure 3 shows plan entities as light blue rectangles, execution entities as dark yellow rectangles, and data entities as dark blue rectangles.

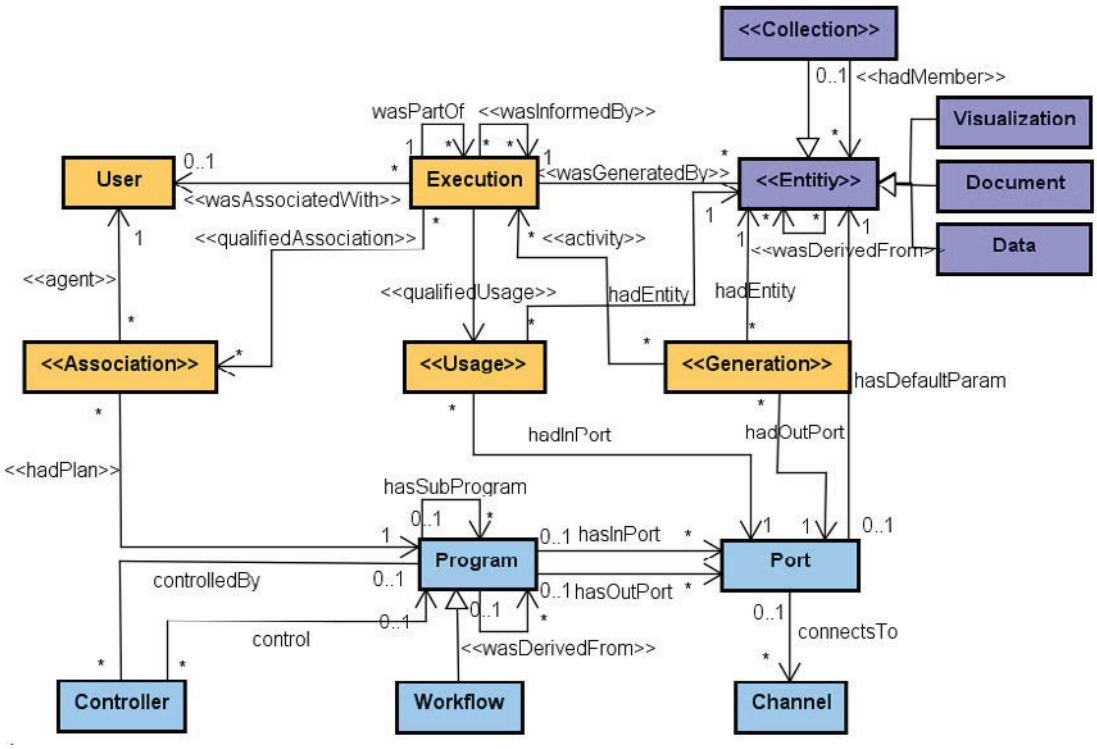

Figure 3. ProvONE conceptual model, from the DataONE documentation[7]

5. Mapping PROV-Wf and ProvONE

Both PROV-Wf and ProvONE extend PROV to represent provenance generated from scientific workflows. They are also able to represent p-prov and r-prov in the same model. However, each one of them has advantages and disadvantages. To analyze the similarities and differences among them we create a mapping of entities (Table 1) and relationships (Table 2). To generate the mapping, we took each entity in one model and try to find in the other model an entity that has the same semantics. We did the same for relationships. In this section, we also explain the intersections and gaps of each model.

[6] https://www.dataone.org/

[7] http://jenkins-1.dataone.org/jenkins/view/Documentation%20Projects/job/ProvONE-Documentation-trunk/ws/provenance/ProvONE/v1/provone.html

Table 1 lists and maps the entities from PROV-Wf and ProvONE models and classifies them as p-prov, r-prov, and domain data. As can been seen, some entities do not match. This is because some of them were represented as entities in one model, and as relationships in the other. For example, the entity *Program* in PROV-Wf can be represented by the *Program's* auto-relationship *hasSubProgram* in ProvONE. Similarly, the entity *ExecuteWorkflow* in PROV-Wf can be represented by the *Execution's* auto-relationship *wasPartOf* in ProvONE.

Table 1. Mapping PROV-Wf and ProvONE entities

PROV-Wf	ProvONE	Provenance Type
Workflow	Workflow	p-prov
Activity	Program	p-prov
Program	-	p-prov
Field	-	p-prov
ValueType	Entity	p-prov
ExecuteWorkflow	-	r-prov
ExecuteActivity	Execution	r-prov
ProgramInvocation	-	r-prov
ExecuteExtractor	-	domain data
DomainData		domain data
File	Document	r-prov
Value	Data	r-prov/p-prov
Relation	Port	r-prov/p-prov
Machine	-	r-prov
Scientist	User	r-prov
-	Controller	p-prov
-	Visualization	r-prov/p-prov
-	Collection	r-prov/p-prov
-	Channel	p-prov
-	Association	r-prov/p-prov
-	Usage	r-prov/p-prov
-	Generation	r-prov/p-prov

PROV-Wf provides specific elements to represent domain data such as *Execute Extractor* and *Domain Data*. On the other hand, ProvONE does not have an element to represent that kind of object, but it has a different specialization (*Visualization*) and grouping element (*Collection*) for the *Entity* object. ProvONE can also represent provenance from WfMS that use *channel* concept (*i.e.,* Kepler [Ludäscher *et al.* 2006]). A *Channel* element can connect output to input ports. PROV-Wf, in turn, identifies a computer or a virtual *Machine* as an element where some *trial* (workflow execution) may run. ProvONE cannot represent that element, but it represents a *Controller* to a specific *Program* that does not exist in PROV-Wf.

ProvONE has also three entities that connect p-prov to r-prov using a ternary relationship: *Association*, *Usage*, and *Generation*. These kinds of entities are not present in Prov-Wf, but some similar features are encountered in the *hadPlan, wasAssociatedWith, used,* and *wasGeneratedBy* relationships. ProvONE allows for easy integration of terms from external vocabularies, including Dublin Core or WfMS and it is supported by a large data conservation project, DataONE[8].

[8] https://www.dataone.org/

The mapping outlined by Table 2 relates PROV-Wf and ProvONE relationships. Most relationships of both models come from the PROV model. On the other hand, PROV-Wf and ProvONE models added new elements to represent the relationship among p-prov entities. Here we also have some gaps and relationships that were represented by entities. For example, the relationship *wasInformedBy* in ProvONE model is represented by the entity *Relation* in the PROV-Wf model. This later also represents the same information of the relationship *wasDerivedFrom* (for the entity *Data*) in the ProvONE model.

Table 2. Mapping PROV-Wf and ProvONE relationships

PROV-Wf	ProvONE
hadMember	hasSubProgram
hasInput	hasInPort
hasOutput	hasOutPort
hadParameter	hasDefaultParam
used	used
wasGeneratedBy	wasGeneratedBy
wasAssociatedWith	wasAssociatedWith
hadPlan	hadPlan
hasValue	hadEntity
hasInvocation	wasPartOf
hasExtractor	-
actedOnBehalfOf	-
hasRelation	-
-	wasInformedBy
-	wasDerivedFrom (*Data*)
-	wasDerivedFrom (*Program*)
-	controledBy/control
-	hadInPort
-	hadOutPort
-	connectsTo
-	hadMember
-	agent
-	qualifiedAssociation
-	qualifiedUsage
-	activity

ProvONE has a special relationship *wasDerivedFrom* (for the entity *Program*) to represent different versions of *programs*. There is no such element in PROV-Wf, but different from ProvONE, it may define relationships to inform links between agents such as *Scientist* and *Machine*. The relationships *hadInPort* and *hadOutPort* link p-prov to r-prov entities in the ProvONE model, but there are no representation in PROV-Wf. Finally, *controlledBy/control* relationships relates *Controller* to *Programs* and the relationships *qualifiedAssociation, qualifiedUsage,* and *activity* relationships connect the aforementioned entities *Association, Usage,* and *Generation* in a ternary relationship among p-prov and r-prov elements. Those later are only represented in ProvONE.

6. Final Remarks

In this paper, we expose PROV-Wf and ProvONE features, their similarities and differences. Both models can represent p-prov and r-prov by extending the PROV data model, but do it using different entities and relationships. We do not compare them to

indicate the best one. Rather we describe their features intending to support the best choice for a specific context of provenance design and exchange.

As future work, we plan to develop a mechanism to convert PROV-Wf to ProvONE model and *vice-versa*. This kind of transformation may help scientists to exchange provenance among WfMS that use heterogeneous models to represent p-prov and r-prov at same time.

7. Acknowledgments

The authors would like to thank CNPq and FAPERJ for partially supporting this work.

References

Bivar, B., Santos, L., Kohwalter, T., *et al.* (jul 2013). Uma Comparação entre os Modelos de Proveniência OPM e PROV.

Costa, F., Silva, V., De Oliveira, D., *et al.* (2013). Capturing and querying workflow runtime provenance with PROV: a practical approach. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. . ACM.

De Oliveira, D., Silva, V. and Mattoso, M. (2015). How Much Domain Data Should Be in Provenance Databases? In *Proceeding of the 7th USENIX Workshop on the Theory and Practice of Provenance (TaPP 15)*. . USENIX Association.

Foster, I., Vöckler, J., Wilde, M. and Zhao, Y. (2002). Chimera: a virtual data system for representing, querying, and automating data derivation. In *14th International Conference on Scientific and Statistical Database Management, 2002. Proceedings*.

Freire, J., Koop, D., Santos, E. and Silva, C. T. (2008). Provenance for Computational Tasks: A Survey. *Computing in Science Engineering*, v. 10, n. 3, p. 11–21.

Ludäscher, B., Altintas, I., Berkley, C., *et al.* (2006). Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, v. 18, n. 10, p. 1039–1065.

Missier, P., Dey, S., Belhajjame, K., Cuevas-Vicenttín, V. and Ludäscher, B. (2013). D-PROV: Extending the PROV Provenance Model with Workflow Structure. In *TaPP 13*. , TaPP '13. USENIX Association. http://dl.acm.org/citation.cfm?id=2482949.2482961, [accessed on Apr 30].

Moreau, L., Freire, J., Futrelle, J., *et al.* (2008). The Open Provenance Model: An Overview. In *IPAW*, Lecture Notes in Computer Science. Springer. http://link.springer.com/chapter/10.1007/978-3-540-89965-5_31.