An Event-Centric Provenance Model for Digital Libraries

Donatella Castelli, Leonardo Candela, Paolo Manghi,
Pasquale Pagano, Cristina Tang, and Costantino Thanos

Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" – CNR, Pisa, Italy
{name.surname}@isti.cnr.it

Abstract. Provenance is intended as the description of the origin and/or of the descendant line of data. In the last decade, keeping track of provenance has become crucial for the correct exploitation of data in a wide variety of application domains. The rapid evolution of digital libraries, which have become today advanced systems for the integration and management of cross-domain digital objects, recently called for models capturing the aspects of data provenance in this application field. However, there is no common definition of digital library provenance and existing solutions address the problem only from the perspective of specific application scenarios. In this paper we propose a provenance model for digital libraries, inspired by approaches and experiences in the e-Science and cultural heritage worlds and based on the notion of *event occurred to an object*. The model aims at capturing the specificities of provenance for digital libraries objects in order to provide practitioners and researchers in the field with common DL-specific provenance description languages.

1 Introduction

To use a computer science definition, "*provenance*, also called *lineage*, is the term used to describe the source and derivation of data" (ref. Peter Buneman [8]). Nowadays, keeping track of provenance is becoming crucial for the correct exploitation of data in a wide variety of application domains. For example, physics and medical science communities are not only interested in the data resulting from their experiments, but also in: (*i*) the *origin* of the data, i.e. the physical or virtual location where it was originally produced, or (*ii*) in the *descendant line* of the data, i.e. the sequence of actions that followed its production and determined its transformation in a sequence of intermediate digital manifestations.

Lately, provenance has become increasingly important for Digital Libraries (DLs). DLs, which started out as a digital replicate of traditional libraries, have witnessed a rapid evolution in the last two decades. In particular, content is no longer limited to digital text documents described by bibliographic *metadata records* and functionality not restricted to ingestion and metadata-based search of such objects. DLs today handle so-called *information objects* [9], intended as digital objects of any file format associated to metadata records of different kinds (e.g. geo-spatial information, licensing) and to other objects, to form graphs of arbitrary complexity. Similarly, functionality is richer and much more complex than in the past, in order to cope with such graph-oriented data

M. Agosti, F. Esposito, and C. Thanos (Eds.): IRCDL 2010, CCIS 91, pp. 79–88, 2010.

models and variety of application domains. Due to their cross-domain nature, DLs numbered among the challenges that of introducing ways for keeping track of provenance of information objects and offering functionality to exploit at best such information.

Provenance has been studied extensively by the e-Science community where several models have been proposed [18,20] to capture the notion of provenance of objects generated by *scientific workflows*. A scientific workflow consists of a set of orchestrated processes generating output objects from a set of input objects. In these applicative scenarios provenance is typically recorded at the time at which the workflow is executed and output objects are typically "versioned" and not "modified" by the workflows. However, the life-cycle of DL information objects differs from e-Science's and such models can hardly be adopted in this context. Here, objects are processed (i.e. deleted, updated, generated) by actions fired by independent actors within different jurisdictions, and based on input objects possibly originated in multiple environments and not necessarily obeying to the same management rules (e.g. versioning).

DLs require a provenance model in-sync with these peculiar processing patterns. A number of application specific solutions can be found in the literature, solving provenance representation from the perspective of peculiar DL domains. However, none of them aims at capturing the notion of provenance in a broader sense, that is beyond the boundaries of individual DL solutions and capturing the commonalities of DL information objects on that respect. In this paper we intend to address this issue and propose a novel model for DL provenance based on the notion of *event*. Unlike other provenance models [18,16], which capture the notion of *causality* between different objects of interest, our model claims that the provenance of an object is described by its history, expressed as the sequence of events happened to the object since its birth. The objective of the model is two-fold: providing guidelines and best practices for DL designers and developers to deal with provenance for DLs and, at the same time, facilitating DL systems interoperability on that respect. Moreover, the model mitigates the long-standing *granularity problem* [7], in which the expressiveness of a provenance model is constrained by the granularity level of objects.

The remainder of the paper is organized as follows: Section 2 describes our provenance model, while Section 3 introduces a case study to exemplify the usage of the model and show how it can tackle the granularity problem. Finally, Section 4 illustrates the contributions of this work in the context of related works and Section 5 ends the paper.

2 Description of the Model

In this section we propose an event-centric model, whose structure is illustrated in Figure 1.

According to the model *events* are happenings that occur to *objects*, here intended as instances of the *reference objects*, which in turn represent the entities of interest. More in detail,

Definition. *Reference objects* are conceptual objects classified as such by an extrinsic authority. Reference Objects can be *abstract*, such as concepts and ideas, or *concrete*, such as people and places.

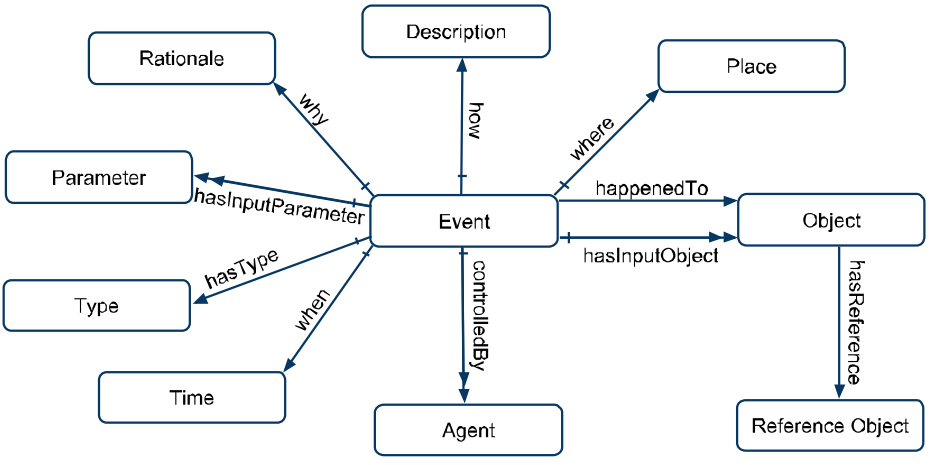

Fig. 1. Relationships and entities of the model

A reference object can have multiple instances over time, called *objects*.

Definition. An *Object* represents an instance of a reference object and is uniquely identifiable. An object can be a physical materialization, a digital surrogate or a digital instantiation, of a reference object.

All the relationships and entities of the model are shown In this model, we are concerned only with the identity of the entities. The instances of these entities may be expressed in the form of free text, in a machine-parsable ontology or in a grammar-defined language. The modeling of these entities is outside the scope of this model.

In this model, objects are created through an *event* related with one reference object. Such a relationship is represented by the relationship *hasReference*. Different objects relating to the same reference object can be considered as distinct versions of it.

Definition. The relationship *hasReference* maps every object onto exactly one reference object.

Definition. An *Event* is a happening that has an effect on a reference object involving at least one object relating to that reference object.

The provenance of an object is a list of all the events happened to the reference object related to that object. Therefore, the provenance of an object is not established by the chain of all the intermediate versions of the object since its birth, but is established by tracing all the objects relating to the same reference object. This approach enables to partially reconstruct provenance even when some versions of a reference object are missing. This feature is useful when the life cycle of the objects is not confined to a controlled environment.

Sometimes we are only interested in a particular type of events happened to an object. For example, we may be interested in “transfer of custody” events of an artwork or only

in its "restoration" events. Therefore, we introduce the concept of *type* into the model to enable this kind of event filtering.

Definition. The entity *Type* describes the type of an event.

Definition. The relationship *hasType* maps an event onto zero or one type.

An event may be intentionally triggered by an *agent* or may be the result of coincident actions of multiple players. Example events are "the fall of the Berlin Wall" and "update of census statistics". An event is related to objects through the two relationships *hasInputObject* and *happenedTo*.

Definition. The relationship *hasInputObject* maps an event onto zero or more objects. It identifies all the objects necessary for the happening of an event.

Definition. The relationship *happenedTo* maps an event onto exactly one object. This property identifies the object subjected to an event.

The object subjected to an event may represent an initial version or a new version of a reference object. In such cases, the event would be a creation event or a modification event, which takes a previous version of the reference object as input.

An event has a number of attributes. *How* an event happened is described by a *description*. Example descriptions are "change all letters to lowercase" and "compute by exponential smoothing the updated value of the temperature reported by a sensor".

Definition. *Description* is an entity that describes how an event happened.

Definition. The relationship *how* maps an event onto zero or one description.

An event happens in a particular *place* at a particular *time*. Understanding *where* an event happened can be useful in establishing the legal aspect of an event. For example, the use of a particular cipher may be legal in one country while illegal in another one. Understanding *when* an event happened enables us to sort events cronologically. Whether an event is instantaneous or has a duration depends on the time with which it is associated.

Definition. *Place* represents any referenceable physical location.

Definition. The relationship *where* establishes where an event happened. It maps an event onto zero or one place.

Definition. *Time* comprises all temporal notions. We assume that instances of this entity are comparable.

Definition. The relationship *when* establishes when an event happened. It maps an event onto zero or one time.

An event can be influenced by a number of factors. When an event is artificial, it may be *controlled by agents*. In this case, the agents may have *rationales why* they triggered the event. For example, the "Apollo 11 mission is controlled by NASA". The outcome of the event may be affected by some *parameters*.

Definition. The entity *Agent* comprises people, groups or organizations capable of controlling an event. An agent controls an event when he/she can initiate and terminate the event at will.

Definition. The relationship *controlledBy* maps an event onto zero or more agents. This relationship describes which agent controls an event.

Definition. The entity *Rationale* represents the motivation of why an agent has triggered an event.

Definition. The relationship *why* describes the rationale behind an agent's decision to trigger an event and is only defined by events controlled by some agents. It maps an event onto zero or one rationale.

Definition. The entity *Parameter* represents is a value, not an entity.

Definition. The relationship *hasInputParameter* maps an event onto zero or more parameters. It identifies parameters that affect the outcome of an Event.

We have seen all the entities and relationships of the model. We assume that the entities are expressed in models defined elsewhere in a digital library and that instances of these entities can be uniquely referenced.

3 Case Study: AquaMaps

In this section, we illustrate through an example how, by using the proposed model, provenance can be queried at an arbitrary granularity level even when events are only captured at their native granularity.

The example is taken by an application, called AquaMaps, implemented in the framework of the D4Science project [3]. D4Science (DIstributed colLaboratories Infrastructure on Grid ENabled Technology 4 Science - Jan 2008-Dec 2009) is a project co-funded by European Commissions Seventh Framework Programme for Research and Technological Development. Its major outcome is a production e-Infrastructure enabling on-demand resource sharing across organizations boundaries. Through its capabilities this e-Infrastructure accelerates multidisciplinary research by overcoming barriers related to heterogeneity (especially related to access to shared content), sustainability and scalability. In the context of D4Science, content can be of very heterogeneous nature, ranging from textual and multimedia documents to experimental and sensor data, to images and other products generated by elaborating existing data, to compound objects made of heterogeneous parts. The D4Science e-infrastructure also supports the notion of Virtual Research Environments (VREs), i.e. integrated environments providing seamless access to the needed resources as well as facilities for communication, collaboration and any kind of interaction among scientists and researchers. VREs are built by dynamically aggregating the needed constituents, i.e data collection, services and computing resources, after on-demand hiring them through the e-Infrastructure [3]. A D4Science VRE offers a virtual view of an information space populated by objects that

can be composed by different parts, each of which can be derived through specialized elaborations from different heterogeneous sources. If the D4Science supported flexibility for sharing and re-using provides a great potentiality to VRE users, it also largely increases the importance of all the requirements that motivate the need for provenance information, i.e. authenticity and data quality, reproducibility, policies management, etc. Moreover, it also makes particularly challenging the association of provenance information with the information objects and their parts.

One of the VREs that are currently supported by the D4Science Infrastructure is AquaMaps [2]. This VRE inherits its name from a homonymous service which implements an approach to generate model-based, large-scale predictions of currently known natural occurrence of marine species. The AquaMaps service allows the biodiversity community to establish/predict species geographic distribution based on so-called "species ecological envelopes". In order to enhance this service, a dedicated VRE has been deployed in D4Science to provide biodiversity scientists with an experimentation environment, providing seamless access to a potentially large array of data sources and facilities.

Information objects in AquaMaps can be elementary objects or compound objects, e.g. ,a datasets expressed in form of a relational table, whose records are elementary objects interrelated with *part-of* relationships with the compound object of the table. Since the modification of one object may lead to the implicit modification of another object, e.g. , the one that contains the former, a provenance model for AquaMaps must be able to capture these implicit modifications.

One of the main functionality of AquaMaps is to generate fish occurrence prediction maps for marine species. Fish occurrence data are collected from a number of sources such as OBIS [5] and GBIF [4], these sources may in turn be aggregators that harvest data from other sources. The collected data is then fed into Fishbase [1] and may be curated. Prediction maps are generated from data in Fishbase based on prediction formulae provided by experts. Each prediction map is composed of half-degree cells. The prediction map of a half- degree cell indicates the probability of occurrence of a certain species in that location. One provenance query is to find out why we get a certain value for a particular half-degree cell. The main difficulty of such query arises from the so-called "granularity problem". Previous research efforts in provenance have recognized this problem and attempted [11] to enforce a single granularity level. However, the end result was either a too coarse grain model that does not capture enough information or a too fine grain model that is too laborious to maintain. Most importantly, in order to match the granularity of processes to that of objects, we would need to introduce counterintuitive fictitious processes. Instead, in our model, modifications to tables and tuples can be modeled as events at their native granularity, i.e. , if a process is applied to a table, then it is modeled as an event that takes a table as input.

The following example queries are written in set notation.

The computation process is shown in Listing 1. In this example, we are interested in computing all the events happened to the information object Salmon. Line 2 computes all the information objects that contain the object Salmon. Line 3 computes all the reference objects of these information objects. Line 4 computes the time at which the information object Salmon came to existence. Under the assumption that time is

Listing 1. Example showing how to retrieve all the events that modified the information object Salmon. We assume that the data model and the provenance model are interoperable by the function 'find_objects_that_contain(r)' and that time is comparable.

1. r = record of Salmon
2. containing_objects = find_objects_that_contain(r)
3. reference_objects = {o.hasReference : o$\in$ { r } $\cup$ containing_objects }
4. t = e.when, where e $\in$ all_events $\wedge$ e.happenedTo = r
5. events = { e : e $\in$ all_events $\wedge$ e.happenedTo.hasReference $\in$ reference_objects $\wedge$ e.when $\leq$ t}
6. provenance(r) = sort(events)

comparable, line 5 computes all the events happened to the reference objects before that time. Line 6 sorts the events in cronological order.

The second query is to find out contributors along the data collection chain. The difficulty lies in understanding when to attribute credits because not all events happened to an object are significant and the criteria of establishing which events are significant should be domain specific. In Aquamaps, all the data source providers and aggregators should be credited. We assume that all the data objects are brought into the system by an event of type 'create' and that event is *controlledBy* its data provider. Listing 2 shows how the contributors to the information object Salmon can be computed. The first 5 lines are identical as those in Listing 1. Line 6 defines a set 'contributing_events' of event types whose values are assumed to come from a controlled vocabulary. An agent of an event whose type is in 'contributing_events' is considered to be a contributor. Therefore by extending the model with a controlled vocabulary and defining a 'contributing_events' set, we can appropriately credit the contributors. The same approach can be applied to find out the copyright holders of an information object in a DL.

Listing 2. Example showing the retrieval of all the contributors to the information object 'Salmon'

1. r = record of Salmon
2. containing_objects = find_objects_that_contain(r)
3. reference_objects = {o.hasReference : o$\in$ { r } $\cup$ containing_objects }
4. t = e.when, where e $\in$ all_events $\wedge$ e.happenedTo = r
5. events = { e : e $\in$ all_events $\wedge$ e.happenedTo.hasReference $\in$ reference_objects $\wedge$ e.when $\leq$ t}
6. contributing_events = {'creation', 'harvest'}
7. creditors(r) = {e.controlledBy : e $\in$ events $\cup$ e.type $\in$ contributing_events}

The third query is to find an explanation of the presence of a certain information object. The difficulty of this query is similar to that of the first one, i.e., that of being capable of returning the result independently of the data processing granularity. An example is shown in Listing 3. The function contains(object1, object2) in line 5 returns true if object2 is part of object1 or if object2 equals to object1. Line 4 computes the events that created the information object Salmon, which may have been created in a creation event at the granularity level of tuple or in a modification event at the granularity level of table. In both cases, the creation event can be correctly retrieved.

Listing 3. Example showing how we can explain the presence of the information object 'Salmon'

1. r = record of Salmon
2. containing_objects = find_objects_that_contain(r)
3. reference_objects = {o.hasReference : o∈ { r } ∪ containing_objects }
4. events = { e : e ∈ all_events ∧ e.happenedTo.hasReference ∈ reference_objects ∧ !contains(e.hasInput,r) ∧ contains(e.happenedTo, r)}
5. rationale = {e.why : e ∈ events}

We have seen that under the reasonable assumption that the *part-of* relationship is captured by the data model, we can mitigate the long-standing granularity problem with a relatively simple provenance model.

4 Related Work and Contributions

Provenance has extensively been studied by the e-Science community [21]. The objects of concern are data generated by scientific workflows and object provenance is described by a static description of the workflow as "the set of actions executed over the object". Workflow-based provenance models assume that each object has its own provenance trail, which can be represented by the versions of the object generated at different steps of the workflow or by the processes of the workflow that contributed to the creation of the object. Some provenance models focus on the trail of versions approach [13]. Moreau et al. [17] focused instead on the trail of processes approach. The authors surveyed provenance requirements of a number of use-cases and, through the organization of a series of provenance challenges [6], defined a directed acyclic graph model suitable for describing *causal relationships* between objects of interest.

Complementary to the workflow-based approach taken by the e-Science community lays the event-centric approach. In the event-centric approach, the provenance of an object is described by the chains of events that affected its status. This approach is typically taken by the cultural and heritage community to describe historical events [15], typically intended as meetings between physical and/or abstract historical entities in some space-time context. Such models focus on the interweaving of the entities rather than on the reason and outcome of the meeting. Ram and Liu [19] proposed a generic full-blown event-based model to describe the semantics of provenance. The event-based provenance model proposed in this paper differs from this one in two main aspects.

First of all, the model reflects the importance, typical of digital libraries, of keeping track of the agents firing an event. Understanding how an event happened and what parameters directly influenced an event are important in this context. This is not the case for historical events, which are identified a posteriori for their symbolic significance and are generally caused by the coincidence of actions of independent users.

Secondly, Ram and Liu's model views provenance as information that enriches information objects while our model views provenance as the glue that links information objects. In this sense, differently from process provenance approaches [14], in our model events are orthogonal to objects. Events regarding an object do not belong to the provenance briefcase of the object but are shared among all objects involved in the

event. Decoupling objects from events in the model has two immediate implementative benefits: (*i*) the approach is well-applicable to distributed/parallel computing scenarios, where participating services can collaboratively operate over shared objects; (*ii*) since the size of provenance data can easily exceed that of the objects [10], keeping the two apart reduces redundancy phenomena and contributes to system scalability.

In an optimized implementation, our work lies in between lazy provenance and eager provenance [12] in that events are captured prior to the provenance request, while the list of all the events happened to an information object is generated upon provenance request. By leaving the decision on when to generate provenance trails exogenous to the model, it is possible to make time-space trade-offs on a per-application basis, for example, we can pre-compute and cache provenance trails for a certain set of objects.

5 Conclusions and Future Work

Digital Libraries rapidly evolved in the last decades to become today advanced systems for the integration and management of cross-domain digital objects. As such, keeping track of provenance information of the objects of a digital library, possibly operated over, imported and controlled by different agents through differently trustable processes, becomes a crucial issue for the users of such systems. On the other hand, existing solutions typically address problem-specific provenance requirements and do not follow a general-purpose modeling approach. In this paper we proposed an event-based provenance model for digital libraries, inspired by well-known provenance models in the field of e-Science and cultural heritage. Driven by acquired experiences in such fields, we considered a number of digital library real-case scenarios and defined a provenance model which captures what we believe are the essential aspects for describing the provenance of digital library information objects.

In the future we plan to implement a provenance service, devised to be easily integrated in any digital library system to offer support for provenance management based on our event model. The aim is to be able to use the service in the existing production systems of D4Science, to endow the AquaMaps VRE with provenance support.

Acknowledgments. The work reported has been partially supported by the DL.org Coordination and Support Action, within FP7 of the European Commission, ICT-2007.4.3, Contract No. 231551).

References

1. A Global Information System on Fishes, `http://www.fishbase.org/`
2. Aquamaps, `http://www.aquamaps.org/`
3. D4Science, `http://www.d4science.eu/`
4. Global Biodiversity Information Facility, `http://www.gbif.org/`
5. Ocean Biogeographic Information System, `http://www.iobis.org/`
6. Provenance Challenges, `http://twiki.ipaw.info/bin/view/Challenge/`
7. Braun, U., Garfinkel, S.L., Holland, D.A., Muniswamy-Reddy, K.-K., Seltzer, M.I.: Issues in automatic provenance collection. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 171–183. Springer, Heidelberg (2006)

8. Buneman, P., Khanna, S., Tan, W.-C.: Computing provenance and annotations for views (2002)
9. Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., Pagano, P., Ross, S., Soergel, D., Agosti, M., Dobreva, M., Katifori, V., Schuldt, H.: The DELOS Digital Library Reference Model - Foundations for Digital Libraries. In: DELOS: a Network of Excellence on Digital Libraries (February 2008) ISSN 1818-8044, ISBN 2-912335-37-X
10. Chapman, A., Jagadish, H.V.: Issues in building practical provenance systems
11. Cheng, X., Pizarro, R., Tong, Y., Zoltick, B., Luo, Q., Weinberger, D.R., Mattay, V.S.: Bioswarm-pipeline: a light-weight, extensible batch processing system for efficient biomedical data processing. Frontiers in neuroinformatics 3 (2009)
12. Chiew Tan, W.: Research problems in data provenance. IEEE Data Engineering Bulletin 27, 45–52 (2004)
13. Cui, Y., Widom, J., Wiener, J.L.: Tracing the lineage of view data in a warehousing environment. ACM Transactions on Database Systems 25, 2000 (1997)
14. Davidson, S.B., Freire, J.: Provenance and scientific workflows: challenges and opportunities. In: Proceedings of ACM SIGMOD, pp. 1345–1350 (2008)
15. Doerr, M., Ore, C.-E., Stead, S.: The CIDOC Conceptual Reference Model - A New Standard for Knowledge Sharing. In: ER (Tutorials, Posters, Panels & Industrial Contributions), pp. 51–56 (2007)
16. Foster, I., Vöckler, J., Wilde, M., Zhao, Y.: Chimera: A virtual data system for representing, querying, and automating data derivation. In: Proceedings of the 14th Conference on Scientific and Statistical Database Management, pp. 37–46 (2002)
17. Miles, S., Groth, P., Branco, M., Moreau, L.: The requirements of recording and using provenance in e-science experiments. Technical report, Journal of Grid Computing (2005)
18. Moreau, L., Freire, J., Futrelle, J., Mcgrath, R.E., Myers, J., Paulson, P.: The open provenance model: An overview. In: Freire, J., Koop, D., Moreau, L. (eds.) IPAW 2008. LNCS, vol. 5272, pp. 323–326. Springer, Heidelberg (2008)
19. Ram, S., Liu, J.: Understanding the semantics of data provenance to support active conceptual modeling. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 1–12. Springer, Heidelberg (2006)
20. Sahoo, S., Barga, R., Goldstein, J., Sheth, A.: Provenance algebra and materialized view-based provenance management. Technical report, Microsoft Research (2008)
21. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Record 34, 31–36 (2005)