Metadata Management in the Taverna Workflow System

Khalid Belhajjame , Katy Wolstencroft , Oscar Corcho , Tom Oinn ,
Franck Tanoh , Alan William and Carole Goble

School of Computer Science
University of Manchester, Oxford Road
Manchester, M13 9PL, UK
Khalid.Belhajjame@cs.man.ac.uk

EMBL European Bioinformatics Institute,
Hinxton, Cambridge CB10 1SD, UK
tmo@ebi.ac.uk

Abstract

There seems to be a general consensus on the crucial role metadata can play for enhancing the functionalities of scientific workflows systems, e.g., workflow and service discovery, composition and provenance browsing, among others. However, in most cases their management is under-specified, if not left unaddressed at all. A step in this direction, the main contribution of the work presented in this paper is an overview of metadata and their management in the Taverna workflow system. In Taverna, we consider metadata to be a first class citizen in the system, in the sense that we fully cover their life cycle from their creation, through their use and curation until their eventual removal. We present the main steps of this cycle and present the models used for metadata specification. In doing so, we distinguish two classes of metadata: metadata that describe workflow related entities, such as services, workflows and sub-workflows, and metadata that describe workflow executions, also known as workflow provenance.

1 Introduction

Key to the realisation of the semantic web vision are metadata that describe available resources. Metadata are generally defined as *structured data about an object that supports functions associated with the designated object* [6]. In our case, metadata are used to describe workflow related entities with the objective to enhance the potential of the applications that make use of them either internally, that is within the workflow system, or externally, i.e., by third party applications. For example, using metadata that describe a workflow, its constituent processors, i.e., the steps that compose the workflow, the services invoked as a result of processors' enactment and processors' dependencies users may be able to know the scientific value of the experiment implemented by the workflow, the tasks performed by each step in the workflow as well as debugging mismatches by analysing processors' dependencies.

Commonly, metadata are specified using annotations which associate resources to their respective descriptions. In its simplistic form, annotations can be textual descriptions or lists of keywords. However, to enable their use by machines, as well as humans, a more controlled annotation mechanism should be employed for their specification. For example, annotations can be encoded in the form of associations that relate the annotated resources to concepts and properties defined in ontologies. An ontology is described as an explicit specification of a shared conceptualisation [7]. An example of a domain ontology that can be used for semantically describing workflow entities is that built within the myGrid project[1] and which contains concepts that can be used for annotating workflows and services from the domain of bioinformatics [11].

There is a general consensus as to the crucial role metadata can play for enhancing the functionalities of scientific workflows systems [2, 3]. In most cases, however, their management is underspecified, if not unaddressed at all [10].

A step in this direction, the main contribution of the work presented in this paper is an overview of metadata and their management in the Taverna workflow system [5]. In Taverna, we consider metadata to be a first class citizen in the sense that we attempt to fully cover their life cycle from their creation, through their use until their eventual removal. Metadata in Taverna fall into two categories: metadata that describe the workflow entities, i.e., workflows, the processors composing the workflow, the services invoked as a result of processors enactment, and metadata that describe workflow executions, also known as workflow provenance [1, 8]. The remainder of the paper is structured as follows. We begin by presenting the models used for specifying metadata in Taverna (in Sections 2 and 3). Then we describe those aspects related to the life cycle of metadata management (Section 4), with an emphasis on metadata curation, which is the process whereby metadata are curated. Finally, we conclude the paper by discussing our ongoing work, which mainly aims to investigate how the functionalities provided by the Taverna workflow system can be enhanced using collected metadata (Section 5).

2 Metadata for Describing Workflow Entities

The data model used for describing workflow entities extends the Feta data model [9] in which the unit of publication is a service implementing a processor task to cater for the description of *any* processing unit, be is a service, a workflow processor or a workflow. The following introduces the concepts supported by this model focusing on those that are not supported in the Feta data model.

Domain classification Workflow processing units (i.e., services, processors and workflows) can belong to different application domains, e.g., bioinformatics, biomedical informatics, cheminformatics. Also, domains can be specialised into more specific domains, for example, we can distinguish within the domain of bioinformatics processing units that belong to transcriptomics subdomain. A workflow processing unit can belong to several domains, for example, a workflow can belong to both phylogenetics and proteomics domains. The domain annotation of a workflow processing unit is specified by relating it to concepts from the domain classification ontology currently being developed as an extension of the myGrid ontology.

The benefit from supporting domain classification annotations is two-fold. First, they allow a focused and straightforward browsing of available processing units [4]. Second, and more importantly, their specification can be useful in designating the domain experts that are knowledgeable of the domain a processing unit belongs to and, thus, are able to provide accurate annotations regarding more specific aspects such as the task the processing unit implements and the semantic type of its parameters.

Task It captures information about the action carried out by a workflow processing unit within a domain of interest. In bioinformatics, for instance, an operation is annotated using a term that describes the in silico analysis it performs. Examples of bioinformatics analyses include *sequence alignment* and *protein identification*.

Semantic Type A workflow processing unit is associated with a set of input and output parameters. A parameter is described using free text, its mime type, e.g., html, xml and its data type (called transport type in the model). Additionally, it is described using a semantic type that captures information about the application domain covered by the parameter by relating it to the real world concept to which it corresponds. Example of concept in bioinformatics that can be used for describing operation parameters are *biological sequence* and *alignment report*.

Parameter instances In addition to the semantic type, in the myGrid data model, parameters are associated with sample instances. These are used to provide users with an idea on the kind of data required or delivered by a processing unit. More importantly, these sample instances can be used as inputs for performing (regression) testing of the processing unit thereby providing a means to verify it availability and reliability.

Third party annotations A processing unit can be described by a third party using a model other than myGrid data model. Those descriptions are made accessible through URLs that associate them to their corresponding service description in the myGrid service registry.

3 Metadata for Describing Workflow Provenance

Workflow provenance can be thought of as the information necessary for reconstructing the execution of a given workflow. In Taverna, we use a graph model to capture provenance information: a graph model allows for capturing the dependencies between processor instances (i.e. the enactment of processors) and data products. Accordingly, we distinguish two classes of provenance information: process-related and data-related. The former describes the behaviour of the processors that compose a workflow during the workflow enactment, e.g., it specifies whether a processor execution succeeded or failed, whereas the second describes intermediate data products used as input or delivered as a result of processors' executions. We will now describe them in more detail:

Process-related provenance: This captures information about the status of workflow components, i.e., the workflow itself, its sub-workflows and constituent processors, by specifying whether their enactment succeeded or failed. It also captures temporal dependencies between processor instances.

In Taverna2 (the upcoming version of Taverna) developers will be able to customise the behaviour of processors to meet their specific needs. This can be useful for implementing advanced functionalities, such as credential acquisition, dynamic selection of the service to perform the processor's task and so forth. For this purpose, every processor will be associated with a customisable dispatch stack responsible for its enactment. A dispatch stack is composed of ordered dispatch layers, each of which is responsible for processing the jobs delegated to it by the layer above, as a result of the process enactment, and returns the processing results or errors message. Examples of dispatch layers include *invocation layer* that is used for calling a service and the *logging layer* which is used for logging the jobs and messages exchanged. A dispatch layer is configured using a set of properties each of which contains a property name and a property value. For example, the dispatch layer *Parallelize* has a property called *maxJobs* which specifies the maximum number of parallel jobs that can be processed by the dispatch layer. At run time, the enactment of a processor is performed by submitting jobs to its associated dispatch stack which in turn issues the jobs to the first (top) dispatch stack.

The provenance log will capture the behaviour of a processor by logging the exchange of jobs, results and error messages between the dispatch layers of the dispatch stack responsible for the enactment of the processor.

Data-related provenance: In the upcoming version of Taverna, processors will be able to exchange data by value but also by reference. Exchanging data by reference allows, among other things, to reduce the amount of data the workflow enactor has to convey between services, thereby (and hopefully) supporting data-intensive processes.

Figure 1 illustrates that there are four kinds of data entities that can be input/output by a workflow processor: namely, literals, data documents, error documents and list of data entities. Data and error documents are composed of references schemes, which can be seen as keys that identify the data that compose the document in question. Data entity are related to each other using the association *derivedFrom* and which is used to specify for a given data entity, the data entities that were used in its derivation, e.g., the data output by a processor is generally derived from the data that were used to feed the execution of that processor. The derivation relationship can be specialised further to specify, for example, whether the derived data is identical to the source data.

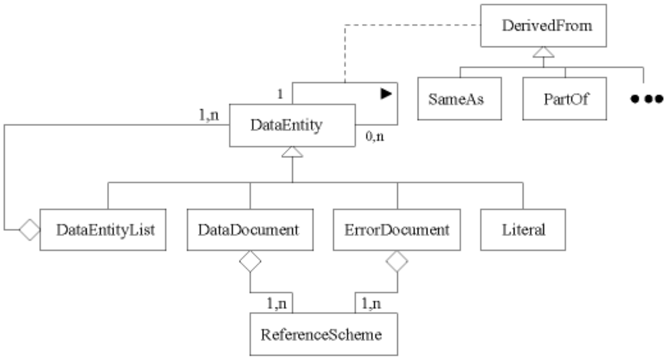

Figure 1. Data-related provenance

4 Metadata Life Cycle

The life cycle of annotations (metadata) is the process whereby they are created, retrieved, used, stored, updated and eventually archived or removed. In this section we describe each of these operations, which are grouped in three main sets, as shown in figure 2.

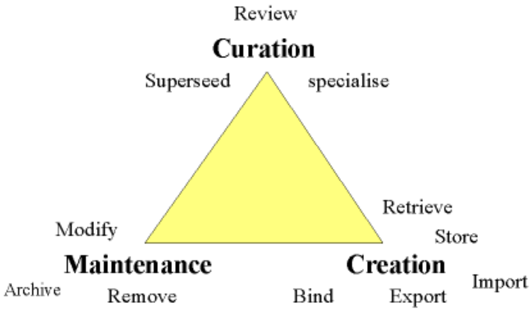

Figure 2. Metadata life cycle

4.1 Metadata Creation Operations

Metadata Creation refers to the group of operations of instantiating the objects that describe workflows entities. Here, by workflow entity we mean any workflow-related *object*, be it a web service, a sub-workflow, a data link, the workflow itself or any other provenance information. The creation of an annotation can take place at different points in time depending on the described entity. For example, workflows and their processors can be described at design time, i.e., while specifying the workflow, whereas annotations describing workflow provenance are generally created during workflow enactment or after its completion.

Besides, metadata may be created directly by users or created outside the workflow system and then imported (from existing entities like metadata repositories, publications, etc.). A typical example of metadata import is that of web service annotations created using standalone annotation tools [9]. Also, some of the metadata describing workflows as well as their provenance can be imported from third party repositories. An example is that of the workflow repository being developed under the aegis of the myExperiment project[2].

Once created, metadata has to be stored, maintaining the links (aka bindings) between the workflow entity that they describe and the vocabulary used for their description. The type of storage support used will depend on the nature of specified/collected metadata. For example, annotations describing workflows, processors, as well as processors behaviour can be defined as part of the workflow specification. On the other hand, annotations describing services as well as workflow provenance can be stored in separate RDF/XML stores, thereby to facilitate their retrieval and exploitation by parties other than the workflow system.

4.2 Metadata Maintenance Operations

Workflow entities are not immutable objects, rather they are subject to changes, e.g., a workflow can be modified by removing/adding processors.

Also, it is possible that the implementation of a web service operation is modified to cater for new tasks. In these cases, the annotator may want to update the annotations accordingly keeping them conform to the entities they describe.

It is worthy noting that certain annotations are immutable even when the entities they describe are modified. A concrete example is that of annotations describing workflow provenance. In principle, such metadata should not be modified even when the workflow for which provenance data was gathered is modified: from a provenance point of view, users are interested in information describing the workflow at the time of its execution.

Finally, annotations are rarely deleted since they can be of value even when the workflow entities they are describing do not exist any longer. In certain cases, however, a choice can be made to delete them. For example, a service may be no longer available: services providers are not completed to continuously supply a specific service. In this case, the annotator may decide to remove the annotations describing that web service from the service registry, or at least archive them so that they could be accessed from a backup repository.

4.3 Metadata Curation Operations

Annotation is to a large extent a subjective task in the sense that it reflects the personal opinion of a human annotator as to which text or semantic concept accurately describes a given resource. Therefore, it is likely that different annotators disagree on what the accurate annotation for a given resource is, or on the maintenance, archival and removal operations performed over a piece of metadata. This is the reason why metadata curation operations may be applied at any point in time during the metadata lifecycle.

Instead of favouring the annotations of certain annotators to the detriment of others, in Taverna multiple annotations can coexist and be associated with the same resource. Most importantly, it is possible for an annotator to express his/her opinion on existing annotations or even act on them by, e.g., superseding, under-specifying, over-generalising them without their removal. We call this process the annotation curation process and we strive in Taverna to capture it in details as it may be useful retrospectively in a variety of ways, e.g., to review the quality of annotations made by a given annotator.

Figure 3 illustrates the model we use for capturing the annotation curation process. A resource can be associated with multiple annotations created eventually by different annotators using a textual description or a term from a specification, e.g., an ontology identified by *SpecificationURI*. Annotations are also associated with their creation date, *timestamp* and can be stored in a variety of storage supports, e.g., a registry or a workflow specification. An annotator can be either a human that is affiliated to an institution, e.g., *Franck*, an institution or a project, e.g., $^{\mathrm{my}}$Grid, or a creating system, e.g., Taverna. The annotations of a given resource are related to each other using the association *curated* which is used by annotator to specify why the previous annotation is not appropriate in their opinion, using the attribute *observation*, and what kind of relationship between the previous annotation and the annotation they created, using the attribute *effect*. The annotator may additionally specify a resource, e.g., a publication, s/he thinks present evidence that justify his/her curation action.

5 Conclusions and Future Work

In this document, we overviewed metadata management in the Taverna workflow system by underlying the main steps in the metadata life cycle. We also introduced the models used for describing workflow entities and workflow executions as well as the model used for capturing the metadata curation process. In our ongoing work, we are enhancing and validating the presented models against users requirements. We are investigating potential uses of metadata, e.g., discovering workflow entities and guiding their composition, and querying and browsing workflow provenance. We are also investigating the way these applications can be embedded within the Taverna workflow systems taking into consideration the interaction with and expectations of end users.

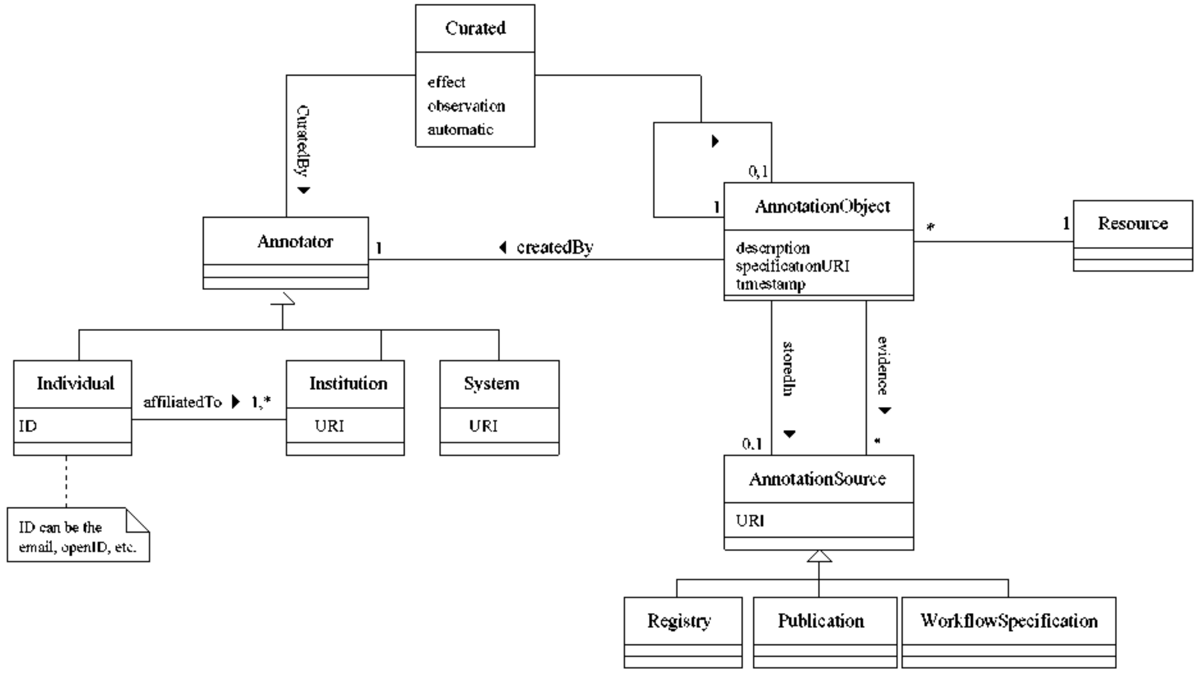

Figure 3. Information model for the metadata curation process

References

[1] I. Altintas, O. Barney, and E. Jaeger-Frank. Provenance collection support in the kepler scientific workflow system. In L. Moreau and I. T. Foster, editors, *IPAW*, volume 4145 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2006.

[2] K. Belhajjame, S. M. Embury, N. W. Paton, R. Stevens, and C. Goble. Automatic annotation of web services based on workflow definitions. *ACM Transactions on the Web (TWEB)*, 2008. To appear.

[3] Sh. Bowers and B. Ludäscher. Actor-oriented design of scientific workflows. In *ER*, pages 369–384. Springer, 2005.

[4] M. Bruno, G. Canfora, M. D. Penta, and R. Scognamiglio. An approach to support web service classification and annotation. In *EEE*, pages 138–143. IEEE Computer Society, 2005.

[5] C. Goble, K. Wolstencroft, A. Goderis, D. Hull, J. Zhao, P. Alper, P. Lord, C. Wroe, K. Belhajjame, D. Turi, R. Stevens, T. Oinn, and D. D. Roure. Knowledge discovery for biology with taverna: Producing and consuming semantics in the web of science. In Christopher J.O. Baker and Kei-Hoi Cheung, editors, *Revolutionizing Knowledge Discovery in the Life Sciences*. Springer-Verlag, 2007.

[6] J Greenberg. Metadata and the world-wide-web. In *Encyclopedia of Library and Information Science*, pages 1876–1888, 2003.

[7] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[8] J. Kim, Y. Gil, and V. Ratnakar. Semantic metadata generation for large scientific workflows. In *ISWC*, pages 357–370. Springer, 2006.

[9] P. Lord, P. Alper, Ch. Wroe, and C. Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In *ESWC*, pages 17–31, 2005.

[10] P. Missier, P. Alper, O. Corcho, I. Dunlop, and C. Goble. Requirements and services for metadata management. *Internet Computing,*, 11(5):17–25, 2007.

[11] K. Wolstencroft, P. Alper, D. Hull, C. Wroe, P. W. Lord, R. D. Stevens, and C. A. Goble. The myGrid ontology: bioinformatics service discovery. *IJBRA*, 3(3):303–325, 2007.