On assisting scientific data curation in collection-based dataflows using labels

DOI:
10.1145/2534248.2534249

Citation for published version (APA):
Alper, P., Belhajjame, K., & Goble, C. (2013). On assisting scientific data curation in collection-based dataflows using labels. In *Proceedings of the 8th Workshop on Workflows in Support of Large-Scale Science* (pp. 7-16). Association for Computing Machinery. https://doi.org/10.1145/2534248.2534249

Published in:
Proceedings of the 8th Workshop on Workflows in Support of Large-Scale Science

On Assisting Scientific Data Curation in Collection-Based Dataflows Using Labels

Pinar Alper
School of Computer Science
University of Manchester
Oxford Road, Manchester, UK
alperp@cs.man.ac.uk

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

Carole A. Goble
School of Computer Science
University of Manchester
Oxford Road, Manchester, UK
carole.goble@manchester.ac.uk

ABSTRACT

Thanks to the proliferation and adoption of computational tools and analysis, scientists are nowadays producing large amounts of datasets. Sharing and publishing such datasets is key to scientific progress, e.g., scientists can analyze datasets produced by their peers to investigate a new hypothesis. Genuine reuse of such datasets can however only be achieved if the are curated using metadata that describe, among other aspects, the context in which they were produced, the datasets from which they were derived and the people involved in their generation. By and large, the curation process is manual, tedious, repetitive and time consuming.

In this paper, we investigate the problem of curating data artifacts resulting from workflow-based analyses. Scientific workflows have gained momentum in the last decade as a means for specifying and automating the repetitive execution of experiments. Most workflow systems have been instrumented to automatically gather provenance information about the data artifacts generated as a result of the workflow execution. While such *raw* provenance traces provide useful information on the lineage of the data artifacts, our interactions with scientists from modern sciences, in particular bioinformatics and biodiversity, suggests that they are not sufficient for curating data artifacts from the data publication point of view. To assist scientists in the curation of such data artifacts, we propose in this paper a novel approach that semi-automates the curation process by exploiting the specification of the workflow incarnating the experiment, the raw provenance traces resulting from its execution as well as motif annotations that describe the data manipulation carried out by the workflow steps. We semi-formally describe the elements of our solution, and showcase its usefulness using a real use case from the biodiversity field.

Keywords

Data curation, Scientific workflows, Workflow Provenance

1. INTRODUCTION

In this age of "Data-Intensive Science" [15] researchers are ever more relying on computational tools and datasets to gain new scientific insights. The sharing and the re-use of valuable scientific data is of paramount importance, as these datasets are often captured through costly instrumentation, complex experiments or labour intensive methods [23].

Data re-use is facilitated through publishing datasets in community databases or archives. A crucial enabler of sharing is *metadata*. It is expected that data is accompanied with at least basic metadata describing 1) its *Provenance* i.e. origins, scientific methodology and 2) its *Context* i.e. assumptions, relations other datasets and scope [7]. Such metadata is not only useful for scientists to discover 3rd party datasets but also useful for them to preserve, recall and understand their own past results, or for peers to easily review data submissions [23]. In order to promote metadata creation there is a recent proliferation in the number of community vocabularies targeted for describing, for instance, derivative relations among datasets [1][10], or, representing citations to source datasets [8].

Metadata for data publishing is created, in most cases, through a manual curation process, which is often performed after the completion of the computational analysis and just prior to sharing of the results. At this stage scientists are faced with the task of recollecting details of significant experimental configurations/parameters and resources and datasets consulted so that this information can be expressed as metadata. Unfortunately scientists often have little time to spare for such curation, a recent survey [23] has shown that scientists are calling for methods and tools to support this process.

In various domains of research "scientific workflows" have become an established mechanism for weaving data-processing activities into structured computational analysis pipelines [11]. Workflows provide 1) an automation function as they are (re)executable pipelines of analysis activities 2) a methodological documentation function as they capture the analysis process followed for the investigation. These benefits encourage scientists to invest significant effort to design their analysis as workflows, modularize and share them [12]. Scientific workflows represent computational how-tos or best-practices, consequently they are designed once and executed several times, by different input data or configurations. Workflow executions result in the generation of several intermediary and final data artifacts. In addition to data, most workflow systems allow the collection of rich metadata, called workflow provenance through instrumenting the execution of workflows. Workflow provenance provides information on the process followed through activity instantiations, their causal relationships, data artifacts consumed and produced by activities and the implicit derivation relations among data artifacts.

When we look at the publishing practices for data artifacts resulting from workflow-based analyses we see that they are also manually curated [3]. Raw workflow execution provenance has limited or almost no contribution to the curation process. We observe that there is a gap between workflow provenance and the *Provenance* and *Context* metadata needs of scientific data publishing. The former provides an implementation-oriented view of the data derivation method and lineage relations among local data artifacts encountered by the workflow execution engine, whereas the latter requires information on the scientific methodology, significant experimental settings, the context/scope of datasets, and their relations to (external) datasets.

Often the metadata required for publishing is to be found implicitly; either in the data itself (e.g. data values, file headers, file names), or it manifests in the workflow design elements such as names of input/output parameters and activities. During manual curation the burden lies with the scientist to sift through workflow descriptions, numerous result files (from multiple runs) and provenance traces to recollect the experimental context.

In this paper we propose a new approach for semi-automating data curation processes by exploiting the description of scientific workflows, which is a useful source that documents the data analysis pipelines. Specifically, our proposal has the following characteristics: 1) We argue that the metadata required for publishing datasets is different than *raw* workflow provenance captured by workflow execution engines. 2) We adopt **labels** as a means to describe the origins and context of the data artifacts generated by the workflow execution. This idea of using annotations to denote origins has been previously put forward, particularly in the areas of "Where-Provenance" in database queries [4] [24]. 3) We automatically generate and propagate labels via curation processes called **Labeling Workflows**. Labels are generated using the following sources of information: the workflow description and the provenance traces captured as a result of its executions 4) We adopt and take inspiration from the idea of the tracking of "value-copying" from database provenance research and adapt it to the context of scientific workflows. In particular, we use a characterization of common activities [13] in workflows to track value-copying and propagate labels through certain activities in workflows.

The paper is organized as follows, we first provide a real-life workflow (from Biodiversity) as a running example (Sec 2.1), this is followed by elaborating on the capabilities of state of the art in workflow provenance (Sec 2.2) and we illustrate metadata requirements of data publishing (with an example again from Biodiversity) (Sec 2.3). The second half of the paper introduces our approach from an architectural point of view and outlines our contributions (Sec 3). This is followed by sections elaborating on each contribution, namely the model of Data Labels (Sec 5), a Process-Based curation model containing four Labeling Operators (Sec 6) and an case based illustration of how Labeling Workflows are generated from annotated scientific workflows (Sec 7). We review related work (Sec 8) and conclude (Sec 9).

2. DATA-INTENSIVE BIODIVERSITY RESEARCH

2.1 Sample Scientific Dataflow

Biodiversity research includes all investigations on understanding biological diversity, its evolution and preservation. Datasets containing species taxonomies, occurrence records or genomic data, which is contributed by various institutions, are pooled in community repositories. Data repositories are made accessible through the web. Alongside data, analysis tools are exposed for the use of the larger community through services. BioVEL[1] is a project that is pioneering the adoption of scientific workflows for biodiversity by building golden-exemplar workflows that bring together datasets and analysis tools for different research scenarios.

Often, prior to performing any particular analysis there occurs a data collection phase. The workflow we give in Figure 1 is designed for such purpose. A set of species names that are of interest in the scope of the analysis is input to this workflow. For each name in the list the workflow retrieves occurrence records using the search services of community repositories. The workflow branch on the left hand side queries the GBIF[2] data repository through a REST service invocation (activity named "gbifOccurence"). For each species the retrieval operation is repeated. Moreover, as the access service provides some kind of pagination functionality, for datasets that exceed 1000 occurrence records, the retrieval operation is performed repetitively until all pages of information regarding one species is collected. For each retrieval the results are returned in a community agreed XML format. [darwin-core]. The data is stripped of its XML tagging and converted to a CSV format using an XSL transformation step ("Transform_XML"). Each page of occurrence data regarding a species makes up an item in a list of strings that contain CSV formatted data. These items in the list are flattened to a single CSV ("Merge_String_List_to_a_String"). The branch on the right hand side perform a similar retrieval operation from the SLW repository ("slwOccurence"), only difference being the lack of pagination, hence the entire occurrence information from SLW [3] repository is retrieved in one call per species. The lists originating from the two branches are joined up ("Flatten_List") and the resulting list of lists (depth 2) is flattened into a single list (depth 1) and finally flattened into a single CSV value.

In a typical scenario, this workflow, which is designed as a golden exemplar, is used as a utility by several biodiversity scientists to retrieve occurrence records of interest for their investigation. The results are input to follow-on workflows that run population modeling simulations. In the remainder of this paper we refer to this workflow as the Data Retrieval (DR) Workflow.

2.2 Workflow Provenance

Figure 2 illustrates a sample execution of the DR workflow, which is run with a collection of two species names, "Cercopagis Pangoi" and "Branciura Sowerby", as input. We illustrate activity invocations with boxes and actual data

[1] http://www.biovel.eu/

[2] http://www.gbif.org/

[3] http://www.slu.se

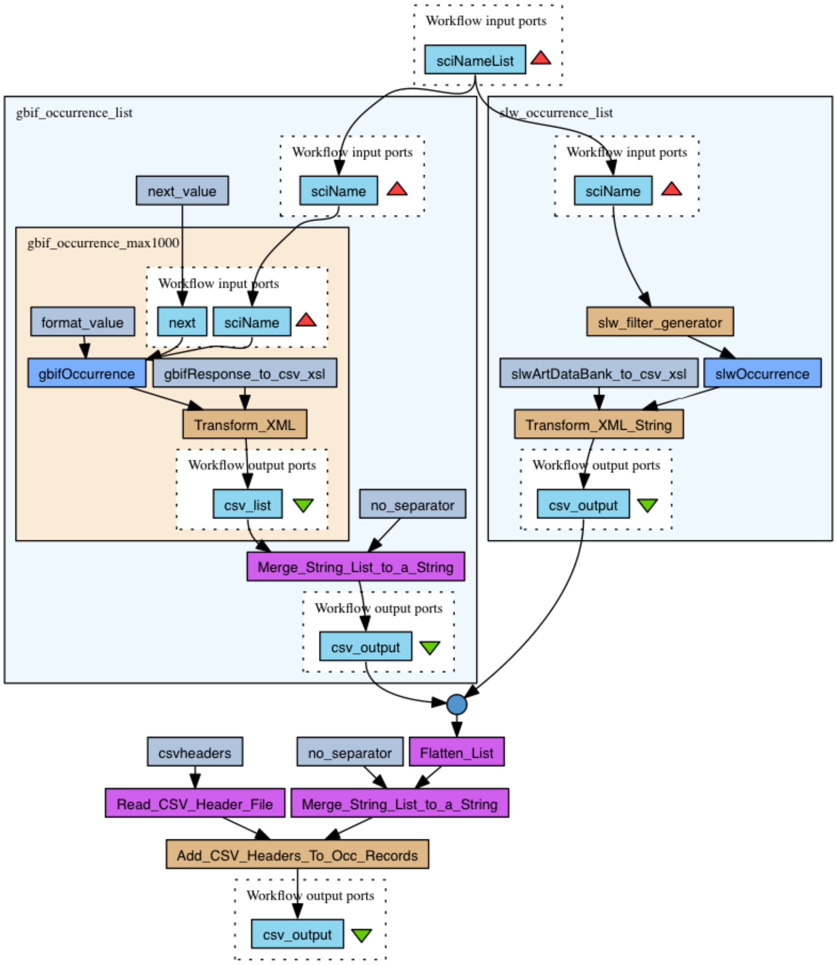


Figure 1: An example workflow from Biodiversity domain. This workflow is part of a larger set of workflows developed for species population modeling.

artifacts consumed and produced by activities are displayed alongside datalinks

We can see that some of the activities, including the data retrieval operations are iterated (i.e. repeatedly invoked). This is either because there are explicit iteration configurations on the activity (e.g. pagination) or because there is a cardinality mismatch between the input expected by the activity by-design and the input encountered at run time. Certain workflow systems, in particular Taverna [20] overcomes cardinality mismatches in two ways.

- In cases where activities are designed to operate over single items and encounter collections, then Taverna repeatedly invokes the target activity with each item in the incoming collection. (This behavior is denoted with a split icon in Figure 2)

- In cases where activities expect collections rather than singletons, Taverna performs cardinality adjustment by converting all accumulated input artifacts into a list. (This behavior is denoted with square-brackets icon in Figure 2)

There is an extensive body of research on provenance in general [22] [21], and workflow provenance in particular [11]. There exist several models for representing workflow provenance [17] [3], and dedicated query languages, query apis and browsers. Workflow provenance allows us to represent and infer activity instantiations, their causality relations among each other and the lineage relations among intermediary and final data artifacts consumed and produced by activities. This viewpoint on origin is particularly useful 1) for workflow debugging, by allowing scientists to pose path queries over derivation relations (e.g. do these two runs with different inputs produce the same values along an execution path) or 2) for smart re-runs , by allowing the re-use of result from previous runs (e.g. re-run the SLW branch of the DR workflow with an updated SLW service end-point).

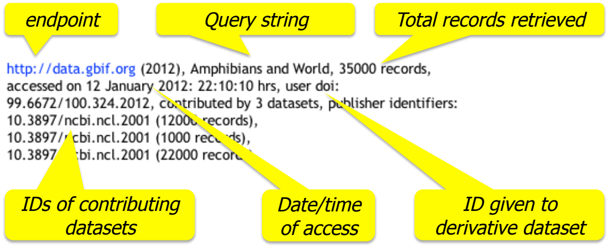
That said, when judged against the metadata needs of data publishing we observe that:

- While workflow provenance captures some form of "origin" information it is an implementation-oriented viewpoint of data lineage that is local to the workflow execution environment (i.e. the trail of all data artifacts that the workflow engine has encountered on the computation path of activities leading to a particular result).

- Existing provenance querying or browsing capabilities, is often "path-oriented" (focused on traversing activity causality and data derivation paths). Scientists often have a "result oriented" viewpoint of the data products of workflows. If, for instance, an intermediary data artifact has significance, it will be promoted to be a workflow output .

2.3 Metadata Required for Data Publishing in Biodiversity

When reporting their findings scientists are expected to make available the datasets used and produced during their analysis and provide information on the context and source of data. If, for instance, the datasets are derivatives of existing ones, this relation needs to specified with a data citation. Note that citations are a very specific kind of metadata for data publishing, it is not our intention here to provide an exhaustive review of metadata requirements in data publishing. Nevertheless, in order to illustrate a special case of metadata and to discuss its coverage with provenance, we provide a sample data citation represented in one of several styles given by the popular Biodiversity data repository GBIF [8]. GBIF states that this citation string can be provided to data consumers by the provider during data delivery or it could be built up by the consumer. In order to build up this citation the scientist needs to recollect the endpoint from which data is retrieved, the query string used, the identifiers of datasets that contribute to the retrieved records, and the identifier that she has assigned to the derivative based on source datasets.



http://data.gbif.org (2012), Amphibians and World, 35000 records, accessed on 12 January 2012: 22:10:10 hrs, user doi: 99.6672/100.324.2012, contributed by 3 datasets, publisher identifiers: 10.3897/ncbi.ncl.2001 (12000 records), 10.3897/ncbi.ncl.2001 (1000 records), 10.3897/ncbi.ncl.2001 (22000 records)

Figure 3: An example data citation in GBIF style.

The way scientists recollect experimental details is by scanning through several result files. Some part of the information (e.g. IDs, query string and record counts) required to build a citation is implicitly available as embedded in data artifacts rather than in workflow provenance assertions relating data artifacts and activities to each other. Moreover info on the nature of required data (e.g. the fact that it is a query string) may serendipitously be available in workflow port names or file names.

Against this setting a desired capability would be to have annotations over workflow results that makes explicit the above illustrated information. Note that this information would need to be sourced from the data itself, the workflow description and its execution provenance.

3. APPROACH OVERVIEW

We outline our approach through the architecture illustrated in Figure 4. We propose the automated generation of annotations over data, which we call *Data Labels*, by building on products of scientific workflow design and execution. Our approach does not interfere with the conventional process (A), where workflow designers create workflows (Step A.1), these workflows are shared with multiple workflow users, they are executed (Step A.2), resulting in the creation of several data artifacts and workflow provenance traces that are stored (Step A.3).

The need for creating metadata on the experimental data products arises when the results are to be reported/published. The reporting process (B) requires the annotation of an existing workflow definition by *Motifs* and a *Label Model* that is to be supported by the workflow (B.1). Before proceeding to the description of our approach, we introduce Motifs and the Label Model.

Motifs denote the data processing characteristics of each activity from a domain independent perspective (such as Data Retrieval, Merging, Filtering). Each motif outlines the data processing behavior, but also defines the labeling behavior, and associated labeling function expected of an activity. Depending on their motif, each activity is associated with either a **label generation** or **label propagation** function. Functions could be chosen from a set of generic ones, or could be domain specific. The simplest and most frequent case a label propagation function is a label copying operation from the data at the input port of an activity to the designated output port.

Label Model is intended as a schema for labels which will carry explicit metadata to be utilized while reporting workflow results. A label model is comprised of *Label* definitions and *Label Vector* definitions. These are intended as a basic metadata schema, a basic set of attributes to be tracked for the data that can be generated during the execution of a workflow. Note that label and label vector definitions can be shared among several workflows, in fact we anticipate that label model definitions are to be made at the investigation level, which spans multiple workflows. In Figure 2 we have illustrate actual label values for data resulting from"gbifOccurence" and "slwOccurrence" activities (illustrated labels carry information on origin, scope and model of data). These label will be generated using the activity definitions and configurations and actual data output of the aforementioned data retrieval activities. Following this retrieval step labels are to be propagated through to input and output data artifacts of succeeding activities based on value-copying relationships.

Given the supported label model for a workflow description and motif annotations on its activities we then generate a *Labeling Workflow* (step B.2). This workflow contains activities each of which is a label generation and propagation operators. This labeling workflow can be applied to the provenance log of a selected execution workflow, which will result in generation of labels for data artifacts.

Following from the outlined approach, in this paper we make the following contributions that we elaborate in the rest of the paper:

- A model of Labels for carrying metadata regarding data artifacts generated in workflow based scientific data analysis.
- A process model for automating curation of intermediary and final data results of workflows. The model is comprised of Label Operators that fall into two major categories 1) label generation 2) label propagation.
- A case study illustrating the generation of Labeling

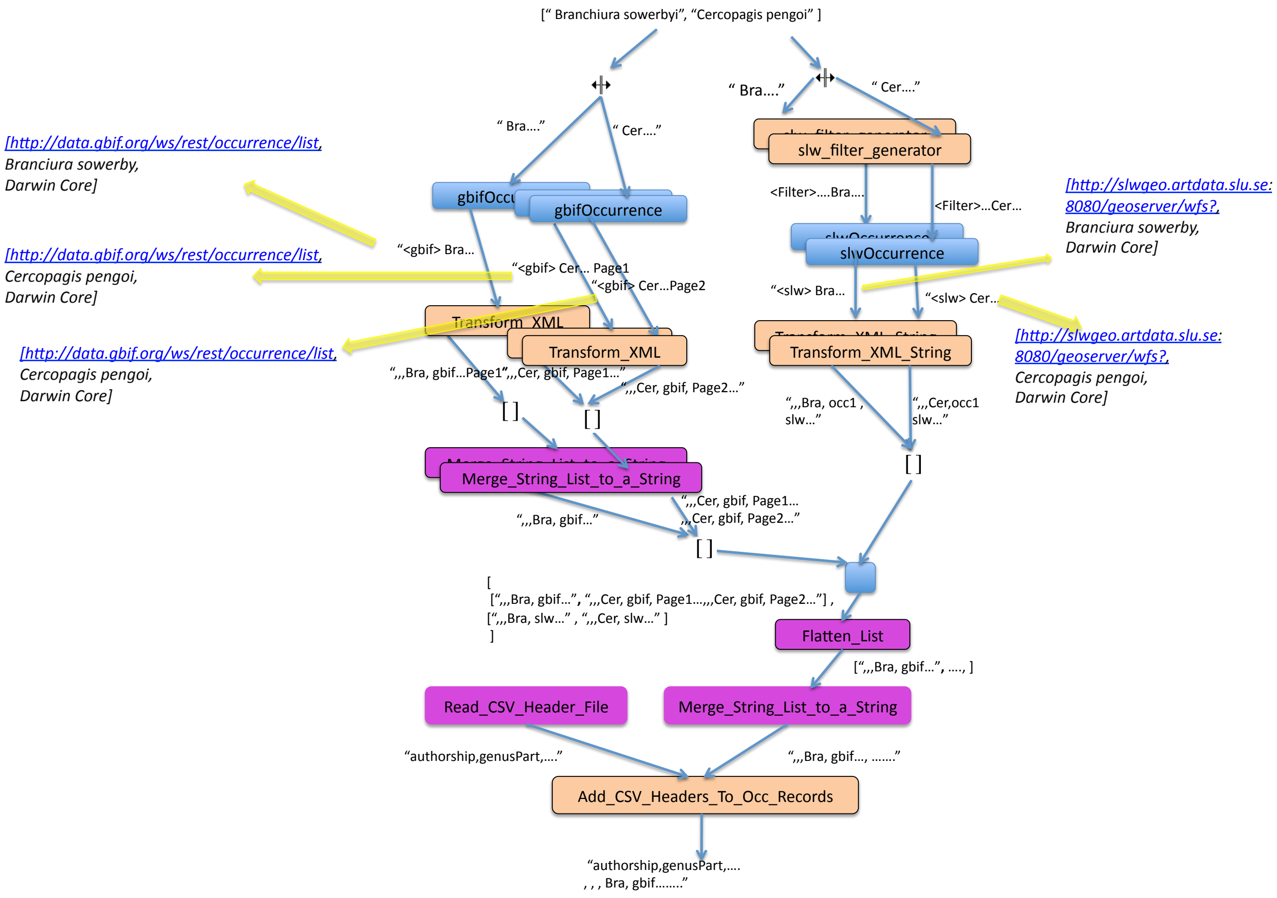

[“ Branchiura sowerbyi”, “Cercopagis pengoi”]
“ Bra….”
“ Cer….”
“ Bra….”
“ Cer….”
[http://data.gbif.org/ws/rest/occurrence/list, Branciura sowerby, Darwin Core]
[http://data.gbif.org/ws/rest/occurrence/list, Cercopagis pengoi, Darwin Core]
[http://data.gbif.org/ws/rest/occurrence/list, Cercopagis pengoi, Darwin Core]
[http://slwgeo.artdata.slu.se:8080/geoserver/wfs?, Branciura sowerby, Darwin Core]
[http://slwgeo.artdata.slu.se:8080/geoserver/wfs?, Cercopagis pengoi, Darwin Core]
gbifOccurrence
slw_filter_generator
<Filter>….Bra….
<Filter>…Cer…
slwOccurrence
“<gbif> Bra…
“<gbif> Cer… Page1
“<gbif> Cer…Page2
“<slw> Bra…
“<slw> Cer…
Transform_XML
Transform_XML_String
“,,,Bra, gbif…Page1”
“,,,Cer, gbif, Page1…”
“,,,Cer, gbif, Page2…”
“,,,Bra, occ1 , slw…”
“,,,Cer,occ1 slw…”
[ ]
Merge_String_List_to_a_String
“,,,Bra, gbif…”
“,,,Cer, gbif, Page1… ,,,Cer, gbif, Page2…”
[[“,,,Bra, gbif…”, “,,,Cer, gbif, Page1…,,,Cer, gbif, Page2…”] , [“,,,Bra, slw…” , “,,,Cer, slw…”]]
Flatten_List
[“,,,Bra, gbif…”, ….,]
Read_CSV_Header_File
Merge_String_List_to_a_String
“authorship,genusPart,….”
“,,,Bra, gbif…, …….”
Add_CSV_Headers_To_Occ_Records
“authorship,genusPart,…. , , , Bra, gbif……..”

Figure 2: Activity Instantiations and data artifacts generated during a run of the Biodiversity Workflow.

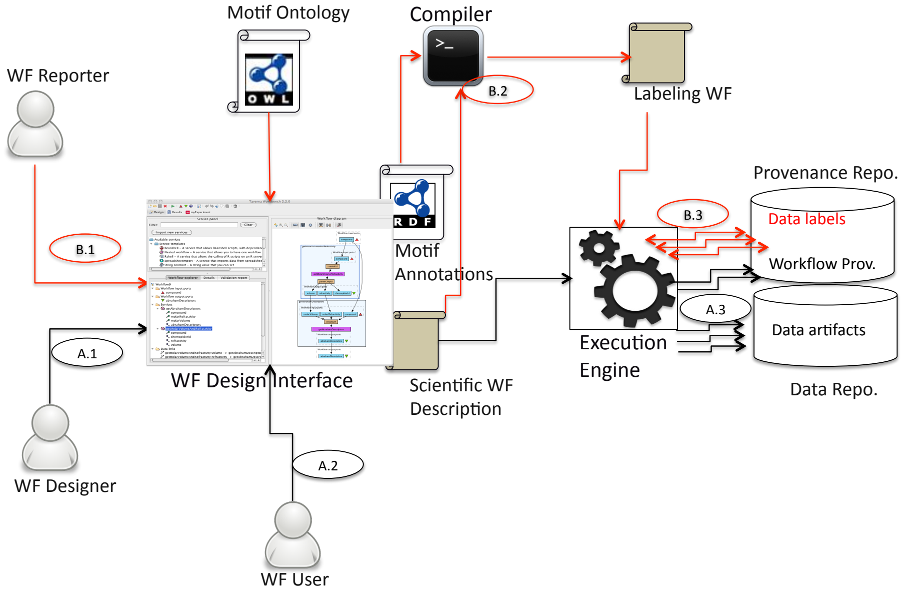


Figure 4: Architecture Overview of Proposed Approach

Workflows containing operators from the process model.

4. MOTIFS IN SCIENTIFIC WORKFLOWS

When to generate new labels and when to relay them is mainly dependent on when data is generated and relayed during the execution of a workflow. In previous work [13] we performed an empirical analysis of 200+ scientific workflows from various workflow systems and diverse domains. The analysis was intended to obtain a categorization of data-processing activities in workflows from a domain independent perspective. The categorization resulted in a catalog of *Motifs* for Scientific Workflows [4]. The analysis has shown that a certain and minority group of activities in workflows perform the scientific heavy lifting in a workflow. These steps are responsible for creating new data either from an analysis/visualization or by retrieving data from external sources. For example, the "gbifOccurence" and "slwOccurence" data retrieval activities in our biodiversity workflow are such steps. The remainder majority activities can be broadly categorized as Data Preparation steps. These helper steps mainly act as adapters that glue together significant activities, or they are dedicated for the local organization of data. (Some of these motifs can be likened to relational query operators, such as Join or Select. The difference is: motifs are high level classifications that would give a rough idea of the data processing rather than an explicit traceable behavior). In Figure 5 we denote the motifs of each activity in call-outs.

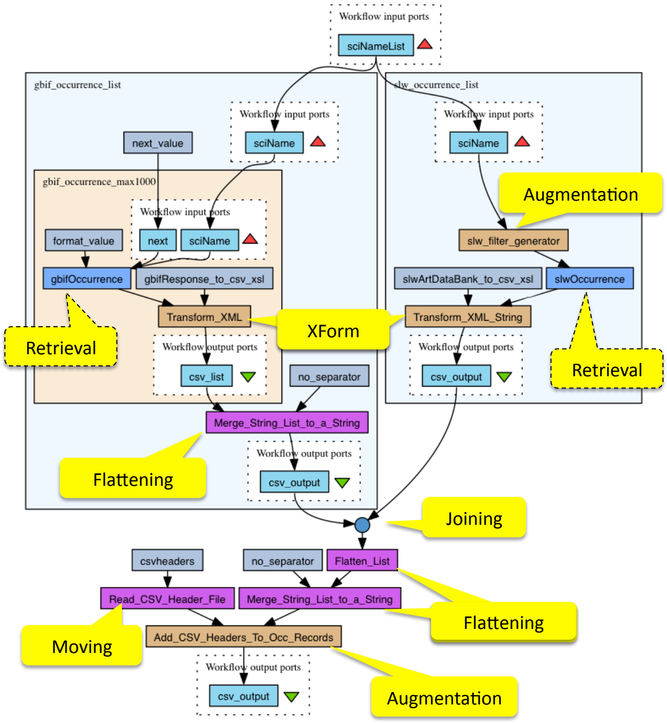


Figure 5: Biodiversity Workflow annotated with its motifs

A common characteristic of all Data Preparation steps is that the nature of their data processing is largely comprised of **value-copying** from the inputs of an activity to its outputs. Note that thos copying could be inexact aswell. Following from Figure 5 the "Transform_XML" steps transforms the XML formatted occurrence data to a CSV format, the "Merge_String_List_To_a_String" and "Flatten_List" steps consume a list of strings and produce a single string by concatenating all input strings with a separator.

We postulate that in cases of such value-copying, the labels of the input data can be transferred to the output data. Our analysis has shown that a very large percentage of operations in workflows (90+%) is categorizable with a motif, i.e. as a data minting or relaying step. This also serves as a justification for introducing a metadata management model based on label generation and propagation.

4.1 Specifying Motifs through Workflow Activity Annotations

We specify the motif of an activity and the corresponding labeling behavior in the curation process by semantic annotations over the workflow. Annotations refer to the Motif Ontology. In Figure 6 we provide a fragment that partially depicts the markup of two sample activities in the DR Workflow. The fragment tells us that the "gbifOccurence" activity in the DR workflow has the "DataRetrieval" motif , whereas the "Merge_String_List_To_a_String" activity has the *Merging* motif. The labeling behavior that the curation process should exhibit over the input/output data artifacts of these activities is inferred based on the motif of the activity. To exemplify, the motif ontology states that the "Merging" motif by-default corresponnds to label propagation behavior. This stated by the "hasDefaultLabelingFunction" property of the "Merging " class. In the case of merging, the default way propagation is realized is with a copy function (Denoted with the "Copy" class, which is defined to a subclass of the "Propagate" operator, in Figure 6, we shall elaborate label operators in the next section). The operational behavior implied by the "Propagate' operator is: applying the designated function (in this case a copy) to the labels of the data artifact of the source port of an activity and associating the resulting labels to the data artifacts at the sink port of the activity. As seen in Figure the source and the sinks of propagation are also specified during motif annotations.

In scientific workflows each activity has a technical grounding with which it is implemented. For the majority of activities, i.e. above-mentioned Data Preparation steps, in which value-copying occurs, we have access to the computing instructions in the form of scripts (e.g. Phyton, R, Beanshell). By using techniques such as static code analysis or programme slicing one could learn these value-copying relations. Similarly using information sources like activity and port names it could be possible to semi-automate the classification of each workflow activity with its motif. This area of investigation is out of the scope of our work.

5. DATA LABELS IN SCIENTIFIC WORKFLOWS

We provide a semi-formal representation for data labels. A label definition is a tuple of the form:

$L_{DEF} = \langle n, t \rangle$

n denotes the name of the label, t denotes its type. Labels are of sets of primitive types such as "xsd:string" or "xsd:int". (From an technical grounding perspective a label definition could, in an RDF based implementation [**?**], correspond to an RDF property definition, label values would

[4] Represented in a light-weight ontology http://purl.org/net/wf-motifs)

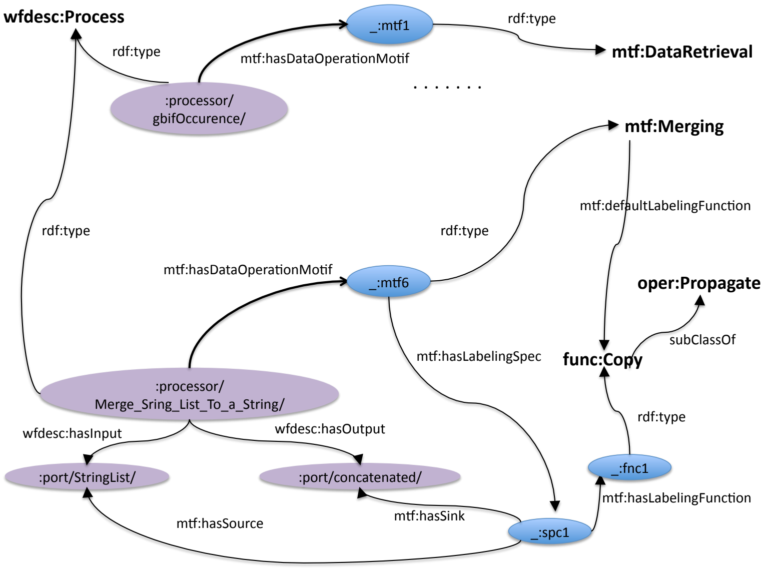


Figure 6: A fragment of workflow annotations depicting motifs and value copying specifications

be represented with RDF literals which the property points at).

A label vector is a set of distinct label definitions : $LV_{DEF} = \{L_{DEF}\}$.

In an analogy to metadata kept regarding files in a file system (e.g. file name, file size, date of creation etc.), when workflows are associated with a given label vector definition, it means that the data artifacts generated from the execution will be auto-curated with these labels. Let us assume that the species population modeling investigation, which, amongst others, incorporates the DR workflow, is associated with a label vector of the following form.

$LV_{DR-WF} = [L_{origin}, L_{scope}, L_{model}]$, where L_{origin}, L_{scope} and L_{model} are labels defined as follows:

$$L_{origin} = \langle \text{“}origin'', \{\text{“}xsd : string''\}\rangle$$
$$L_{scope} = \langle \text{“}scope'', \{\text{“}xsd : string''\}\rangle$$
$$L_{model} = \langle \text{“}model'', \{\text{“}xsd : string''\}\rangle$$

Label definitions act as a schema for label instances. Actual labels are generated in conformance to the definitions by using the actual data artifacts obtained from the execution trace of workflows. According to this schema the sample label vectors in Figure 2 contains information on 1) *origin* that is the service endpoint from which the data is retrieved, 2) *scope* information, which is the species name input parameter to the retrieval operation, 3) *model* information, which describe the retrieved data using the Darwin Core vocabulary (a standard vocabulary in Biodiversity)[5].

In our model there is a one to one correspondence between a data artifact and its label vector. So a data artifact can have zero or one label vector, and a label vector can belong to only one data artifact. In scientific data flows, as with the case of Taverna, data is carried around in very basic data structures; namely, collections and single items. We adopt this **collection-oriented data structuring** approach of dataflows. Consequently our framework caters for labels of singletons and of collections.

In the previous section we described Motifs, these annotations are used to specify the labeling behavior expected of each activity in a workflow. As part of the automated curation process that we introduce, there are cases where new labels need to be inferred from existing ones, outside the scope of specific activities. These are the cases of *datalinks* where data that is outputted by one activity becomes an input to another. The need to infer labels arise in the case of cardinality mismatches between the two ends of a datalink. Consequently in our model, each label L definition is associated with a tuple of functions $\langle fn^{L}_{gen}, fn^{L}_{dis}\rangle$, which correspond to generalization and distribution respectively. To illustrate the semantics of these two functions, consider our previous example. Each of the three labels in our example would be associated with function tuples of the form:

$\langle \text{“}motifs : functions : union'', \text{“}motifs : functions : copy''\rangle$

These functions are identified by function $URIs$. Specifically

- The generalization function allows us to infer a label for the collection when multiple items are aggregated into a collection. So for the each of the three labels in our example, a union function will be used to infer a label for a collection from the individual labels of items forming the collection.

- The distribution allows the transfer of a collection's labels to each individual item in the collection. Consequently the association of the above tuple with the three labels in our model would specify that each label of the collection will be copied over to individual items in the collection (should need arise due to cardinality mismatches aong data links)

Association of functions with label definitions was a design decision we took to cater for domain specific label propagation and inference capabilities. Instead of functions, we could have opted for fixed operators. As in the case of annotation propagation in databases [4], fixed algebraic operators

[5] http://rs.tdwg.org/dwc/dwctype/

(such as Union) can identify how to infer labels in cases of generalization. We are aware that this approach would allow for a less complicated label model, but label inference behavior would be fixed in nature. An example of domain specific label inference can be given from Biodiversity. Biodiversity datasets contain usage licenses, this information is often provided within data records themselves. When data is aggregated from multiple sources and providers, the license of the aggregate dataset corresponds to the most restrictive license of the data in the set. Considering license information is represented with a label, such an inference capability can only be achieved with having functions coupled with labels.

We foresee that for a majority cases of metadata attributes a common generic set of functions for inferring new labels will be sufficient. We capture this with support for **default functions**, which we exemplified in Section 4.1.

6. DATA CURATION PROCESS MODEL

As introduced in our approach, we automate curation of data artifacts through Labeling Workflows. These workflows are underpinned by a Process Model that contain operators for label creation and propagation. Labeling Workflows are not authored directly by users. Instead these workflows are generated/compiled automatically by exploiting 1) the Motif annotations on the scientific workflow description and 2) the Label Model designed to be adopted for the curation. When generating a Labeling Workflow from these two inputs, the generator includes a labeling operator for each annotated data processing activity in the workflow definition, the choice of the operator is informed by the motif annotation of that activity. The generator also includes a labeling operator for the datalinks in the workflow definition, for those which appear to have cardinality mismatches between data structures expected at their two ends. The choice of operator for datalinks is informed by the difference of depths in the data structures expected at the two ends of a workflow data link.

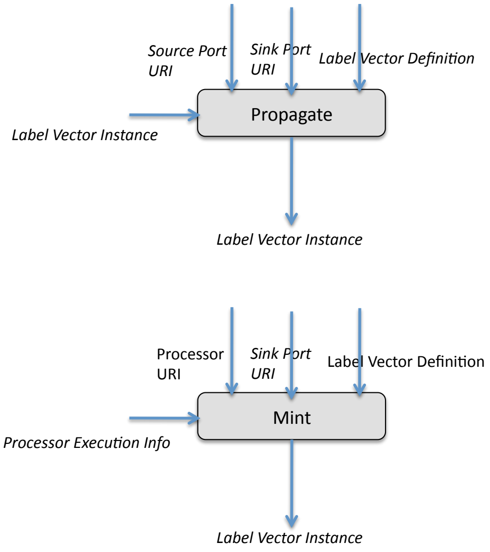


Figure 7: Label Operators Corresponding to Activities in Workflow Descriptions

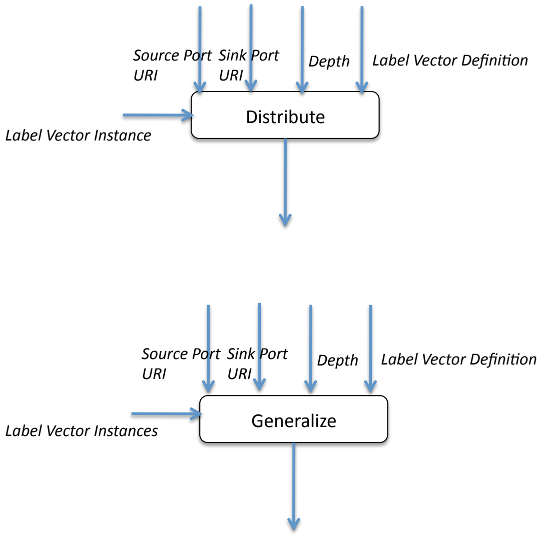


Figure 8: Label Operators Corresponding to Data Links in Workflow Descriptions

Figure 7 illustrates operators included in the Labeling Workflow in response to *activities* in the Workflow, and Figure 8 illustrates operators that are included in the Labeling Workflow in response to *datalinks* in the Scientific Workflow. The vertically drawn inputs to operators are those that are obtained from the annotated scientific workflow description. The horizontally drawn inputs are those that are obtained from the provenance repository, which is also the destination where generated labels are forwarded.

The $Mint$ operator underpins the label generation, whereas, *Propagate*, *Generalize* and *Distribute* underpin label propagation. The major difference between generation and propagation is that the former relies solely on the existence of data artifacts within the execution log of a workflow, whereas the latter require the existence of labels to be previously associated with certain data artifacts. Let us visit each operator

- *The mint operator* generates labels that conform to the label model associated with the scientific workflow. These are domain/activity specific functions that extract metadata from input and output data artifacts (e.g. species name, occurrence details, usage restrictions) of the activity and activity execution configurations (e.g. endpoint, activity type etc). The labeling behavior corresponding to the data retrieval activities in the DR workflow would be represented with the *mint* operator.
- *The propagate operator* corresponds to propagation of the labels of the data artifacts that appear at the designated input port of an activity to the designated output of the activity. The labeling behavior of the activities, “Transform_XML_String”, “Merge_String_List_To_a_String”, “Flatten_List“, “Transform_XML”, and “Add_CSV_Headers_To_Occ _Records” correspond to the *propagate* operator.
- *The distribute operator* corresponds to generating labels for individual items from the label of a collection.
- *The generalize operator* corresponds to obtaining a label for the entire collection from the label set of the individual items.

7. LABEL GENERATION WORKFLOWS

In order to illustrate the compilation/generation process of Labeling Workflows in Figure 9 we provide a fragment of the DR Workflow on the left hand side, and we provide the corresponding fragment of the Labeling workflow on the right hand side. The inputs values to each label operator in the labeling workflow are static configuration parameter. The variable part of the inputs (not depicted for brevity) are the actual data artifacts, data labels and the provenance information that are looked up from the relevant repositories. This parameterized lookup capability allows us to execute the Labeling workflow over different execution traces of the same workflow and be able to generate labels for each.

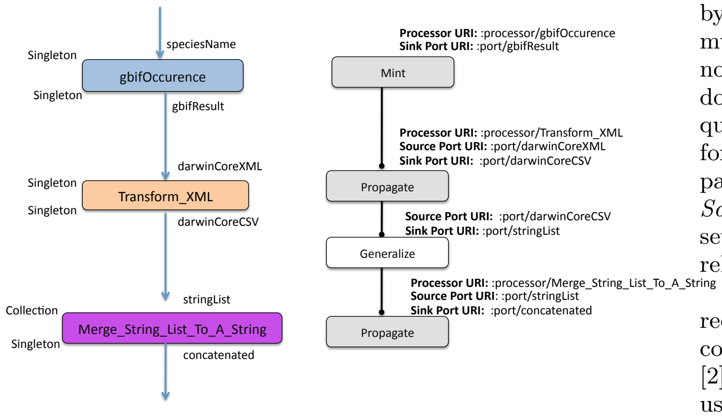

speciesName
Singleton
gbifOccurence
Singleton
gbifResult
darwinCoreXML
Singleton
Transform_XML
Singleton
darwinCoreCSV
stringList
Collection
Merge_String_List_To_A_String
Singleton
concatenated
Processor URI: :processor/gbifOccurence
Sink Port URI: :port/gbifResult
Mint
Processor URI: :processor/Transform_XML
Source Port URI: :port/darwinCoreXML
Sink Port URI: :port/darwinCoreCSV
Propagate
Source Port URI: :port/darwinCoreCSV
Sink Port URI: :port/stringList
Generalize
Processor URI: :processor/Merge_String_List_To_A_String
Source Port URI: :port/stringList
Sink Port URI: :port/concatenated
Propagate

Figure 9: A fragment of the DR workflow (left hand side) and the corresponding Labeling Workflow (right hand side)

As the communication of data is achieved through a repository lookup process and as the label propagation operators depend on labels of source ports to be able to infer labels of sink ports, there are simple control flow dependencies (runs-after type of dependencies) exist between them. Please note that mint operators do not depend on labels, therefore the compilation/generation could result in multiple small Labeling Workflowlets. The algorithm for this generation process is mainly based on a traversal of the Scientific Workflow definition and generating the labeling workflow by picking up the corresponding operators depending on the motifs of activities or the cardinality statuses of datalinks.

8. RELATED WORK

Semantic annotation of workflows and propagation of these annotations to actual data artifacts generated during workflow runs has been proposed by authors in [19]. Here, annotation occurs at the design time and consequently annotations do not exploit information in the values of data artifacts (i.e. they are static such as a set values selected from a domain vocabulary). Therefore annotations in these approaches tend to describe the general nature/characteristics of data (e.g. it is a query string or occurrence record). Annotated data traces allow for querying lineage paths by using domain ontology terms. These works adopt a black-box view of activities and they do not propagate annotations among data artifacts.

Another closely related field is the study of provenance of database queries. Research in this area is categorized [9] as "Why","How" and "Where" provenance focusing (respectively) on 1) tracking which source tuple(s) cause a particular record to appear in a query result, 2) through which operations are the source tuples combined and 3) from which source cells are the data values copied to the result. **Where-Provenance** is particularly relevant to our work as it tracks **value-copying**. Where provenance, has been applied to annotation propagation in relational data integration systems. DBNotes [4] is one such system that propagates annotations on source cells to results of Select-Project-Join-Union queries. Where-provenance is sensitive to query re-writes, consequently, DBNotes provides the option to do annotation propagation by computing all equivalent formulations of a query and propagating annotations from cells addressed by all equivalent queries. DBNotes performs a simple accumulation of all annotations of source cell values and does not provide an algebra for annotations themselves, but it does provide a storage scheme for annotations and means to query them alongside data. Polygen [24] is another system for querying multiple databases and aims to track a very particular kind of annotation, which is the designator of the *Source Database* that a value comes from. Polygen outlines a set of operational rules for propagating annotations through relational operators.

Why and How provenance approaches of databases have recently been applied to dataflows with *white-box* activities corresponding to query operators [16] or PigLatin programs [2], Such fine-grained tracking of provenance finds particular usage in workflow debugging, or change impact analysis. In the context of where-provenance for dataflows in [5] authors adopt a logic-based approach to propagation of schema-level semantic annotations through relational query based activities. Rules for propagation of annotations through each relational operator is represented as a logic constraint. A query is represented as a tree of operators, consequently propagation is cast as an application of logical inference with a forward or backward read of the operator tree. Authors speculate that such an approach can find applicability in semi-automated annotation of workflows. Another work from the same authors [6], propose the use of declarative rules for inferring certain classes of dependencies among data artifacts in the execution trace of a workflow. Among the class of dependencies are *value* and id dependency, which correspond to value or identifier copying from the input of an activity to the output. Similar to our approach they expect rules to be specified on top of workflow descriptions, and later fired over the execution traces to generate dependencies among actual data artifacts. Unlike all other reviewed works, authors have not identified how the resulting dependencies will be utilized.

The work of [18] shall be mentioned here as it has been influential in our work in terms of methodology, specifically for the choice of having a process model for automating the curation. In this work authors describe a scientific workflow re-writing approach, where the workflow is re-written to embed into it a data quality view, which computes quality annotations and filters data based on those annotations.

Metadata propagation is also explored in digital library research. Based on concerns that are similar to ours outlined in the introduction, in [14] authors attempt to ease curation of shared research work products through propagation of basic metadata, such as authorship, subject, or publication date. Propagation is from the research articles to their supplementary material (such as data artifacts, visualizations, charts). The authors acknowledge that propagation of metadata may result in incorrect annotations (e.g. not all charts of a paper may have been authored by the same person), which can be edited during a manual curation step.

9. CONCLUSION

The sharing and re-use of scientific datasets has various benefits to data-intensive science, e.g., acceleration of investigations, improved transparency and reproducibility. One of the cost of these benefits to scientists is the curation effort required prior to publishing data resulting from their analysis. Results of scientific analysis implemented as workflows also require curation. Even though workflow engines collect extensive provenance metadata during the execution this information is not fit for the metadata needs of data publishing. However, by adopting a systematic and structured approach to the analysis process, we observe that workflow based analysis have a big advantage over ad-hoc analyses and bring-about a substrate on which a metadata generation and propagation framework can be weaved.

In this paper we proposed an architecture for assisting in the curation of data artifacts generated through workflow-based analyses. We proposed a model of Labels for carrying metadata, and a process model based on label generation and propagation operators. The process model formally underpins Labeling Workflows which are generated from scientific workflow definitions with markup denoting the Motif of each activity. Motifs characterize data processing, which allows for inferring associated label/metadata processing operator in the Labeling workflow.

The development of the proposed architecture is on-going at the moment. Upon completion of the algorithm for generating the Labeling workflow, we intend to investigate how much these Labeling Workflows lend themselves to concurrent execution. Evaluations with both a synthetic dataset, and an empirical dataset will be performed. Synthetic tests will be used to demonstrate the practicality of the execution of Labeling Workflows. Real-life workflows will be used to understand how much coverage does the proposed model have in real life workflows.

Acknowledgment

The authors would like to thank Francisco Quevedo from the BioVEL project for his support on the DR workflow (BioVEL: EU FP7 283359 BioVel BioDiversity eLaboratory). This work was supported by the myGrid Platform Grant (EPSRC EP/G026238/1, “myGrid: A platform for e-Biology Renewal”).

10. REFERENCES

[1] K. Alexander, R. Cyganiak, et al. Describing linked datasets. In *Linked Data on the Web Workshop in the International World Wide Web Conference*, 2009.

[2] Y. Amsterdamer, S. B. Davidson, et al. Putting lipstick on pig: Enabling database-style workflow provenance. *PVLDB*, 5(4):346–357, 2011.

[3] K. Belhajjame, O. Corcho, et al. Workflow-centric research objects: First class citizens in scholarly discourse. In *Proc. Workshop on the Semantic Publishing (SePublica)*, Crete, Greece, 2012.

[4] D. Bhagwat, L. Chiticariu, et al. An annotation management system for relational databases. In M. A. Nascimento, M. T. Ězsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, editors, *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 900–911, 2004.

[5] S. Bowers and B. LudŁscher. A calculus for propagating semantic annotations through scientific workflow queries. In *In Query Languages and Query Processing (QLQP): 11th Intl. Workshop on Foundations of Models and Languages for Data and Objects, LNCS*, 2006.

[6] S. Bowers, T. McPhillips, and B. LudÃd’scher. Declarative rules for inferring fine-grained data provenance from scientific workflow execution traces. In P. Groth and J. Frew, editors, *Provenance and Annotation of Data and Processes*, volume 7525 of *Lecture Notes in Computer Science*, pages 82–96. Springer Berlin Heidelberg, 2012.

[7] Ccsds. Reference Model for an Open Archival Information System (OAIS). Blue book. Technical Report 1, January 2002.

[8] V. Chavan. Recommended practices for citation of data published through the GBIF network. (May), 2012.

[9] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1(4):379–474, 2007.

[10] P. Ciccarese et al. Pav ontology: Provenance, authoring and versioning. *CoRR*, abs/1304.7224, 2013.

[11] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD Conference*, pages 1345–1350, 2008.

[12] D. De Roure, C. Goble, and R. Stevens. The design and realisation of the myexperiment virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25:561–567, May 2008.

[13] D. Garijo, P. Alper, K. Belhajjame, et al. Common motifs in scientific workflows: An empirical analysis. In *In the proceedings of the IEEE eScience Conference*. IEEE CS, 2012.

[14] J. Greenberg. Theoretical considerations of lifecycle modeling: An analysis of the dryad repository demonstrating automatic metadata propagation, inheritance, and value system adoption. *Cataloging and Classification Quarterly*, 47(3-4):380–402, 2009.

[15] T. Hey, S. Tansley, and K. M. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

[16] R. Ikeda, J. Cho, et al. Provenance-based debugging and drill-down in data-oriented workflows. In *ICDE 2012*. Stanford InfoLab.

[17] P. Missier, S. Dey, et al. D-prov: extending the prov provenance model with workflow structure. In *Proceedings of the 5th USENIX Workshop on the Theory and Practice of Provenance*, TaPP ’13, pages 9:1–9:7, 2013.

[18] P. Missier, S. M. Embury, et al. Quality Views: Capturing and Exploiting the User Perspective on Data Quality. In *Procs. VLDB*, pages 977–988, Seoul, Korea, Sept. 2006.

[19] P. Missier, S. S. Sahoo, J. Zhao, et al. *Janus*: From workflows to semantic provenance and linked open data. In *IPAW*, pages 129–141, 2010.

[20] P. Missier, S. Soiland-Reyes, S. Owen, et al. Taverna, reloaded. In *SSDBM*, pages 471–481, 2010.

[21] L. Moreau et al. The Provenance of Electronic Data. *Communications of the ACM*, 51:52–58, 2008.

[22] Y. L. Simmhan et al. A survey of data provenance in e-science. *SIGMOD Rec.*, 34(3):31–36, Sept. 2005.

[23] C. Tenopir, S. Allard, et al. Data sharing by scientists: Practices and perceptions. *PLoS ONE*, 6(6):e21101, 06 2011.

[24] Y. R. Wang and S. E. Madnick. A polygen model for heterogeneous database systems: The source tagging perspective. In D. McLeod, R. Sacks-Davis, and H.-J. Schek, editors, *16th Int. Conf. on Very Large Data Bases, Proceedings*, pages 519–538. Morgan Kaufmann, 1990.