*Label*Flow: Exploiting Workflow Provenance to Surface Scientific Data Provenance

Pinar Alper[1], Khalid Belhajjame[2] Carole A. Goble[1], and Pinar Karagoz[3]

[1] School of Computer Science, University of Manchester, Manchester, UK
alperp@cs.manchester.ac.uk
[2] Université Paris Dauphine, Paris, FR
[3] Dept. of Computer Engineering, Middle East Technical University, Ankara, TR

Abstract. Provenance traces captured by scientific workflows can be useful for designing, debugging and maintenance. However, our experience suggests that they are of limited use for reporting results, in part because traces do not comprise domain-specific annotations needed for explaining results, and the black-box nature of some workflow activities. We show that by basic mark-up of the data processing within activities and using a set of domain specific label generation functions, standard workflow provenance can be utilised as a platform for the labelling of data artefacts. These labels can in turn aid selection of data subsets and proxy for data descriptors for shared datasets.

Keywords: provenance, annotation, scientific workflows

1 Introduction

Many fields of science are experiencing a proliferation in the sharing and re-use of scientific datasets [TA+11]. Widespread data-oriented science and data sharing necessitates principled data reporting regimes [TF+08] and richer metadata. In this context **"scientific data provenance"** is considered to be essential metadata that describes 1) **the experimental context**, in which data is generated, such as the scope of study, assumptions, experimental settings and descriptions of specialist resources or techniques adopted [TF+08], and 2) **the data's origins** in terms of primary datasets or source databases [TA+11].

Scientists go through a phase of **experiment reporting** prior to sharing datasets. During reporting they **select relevant data subsets** among the pool of all results obtained and **annotate data** to denote its scientific provenance using domain-specific vocabularies [TF+08]. A recent survey [TA+11] has shown that even though there is significant tool support for the collection and analysis of data, similar support does not exist for the organisation of results. Consequently scientists welcome any tool support for it.

Increasingly, scientific datasets are produced from entirely computational experiments. In many domains, Scientific Workflows have become a widespread mechanism for specifying experiments as systematic and (re)runnable compositions of datasets and analysis tools [DF08]. Experiments organised as workflows are advantageous over adhoc analyses as they provide repeatability of computation and traceability among results. Wide adoption of scientific workflows has fostered research on workflow provenance [DF08] with several provenance models and query mechanisms developed [Ge12] [BC+12][MD+13] [MLA+08]. Given their extensive provenance traces, at first glance one expects workflow-based experiments to be advantageous during **experiment reporting**. However, there is little use of workflow provenance during experiment reporting. This is due to: (1) workflow provenance being generic, implementation-oriented metadata [SSH08] that cannot stand-in for domain-specific descriptions expected during scientific data publishing; and (2) the established means of querying workflow provenance i.e. lineage traversal, can be an imprecise selection mechanism for scoping data subsets to be reported.

To this date, the approach to acquiring domain-specific annotations over workflow generated data has been either entirely manual [ZW+04] or partially-automated [MSZ+10]. Certain fixed characteristics at workflow description level are collected and then propagated to data generated by executions. This fixed metadata is useful for reporting but insufficient. Often experiments are reported based-on parametric information that is supplied at runtime via inputs. When one workflow execution is configured with multiple values of one parameter, results need to be annotated accordingly. This category of *dynamic information* offers significant utility in reporting yet it has received limited research attention. On the other hand, while manual annotation can be feasible for capturing fixed metadata, it is hard to scale for dynamic metadata.

Scientists invest significant time and effort into organising experiments as workflows. While this brings benefits when running the experiment, it has limited benefits for reporting. **We propose to bridge this gap and exploit workflow provenance to its full potential by treating it as a medium on which an automated data annotation (labelling) framework can be weaved.** The benefit of labels are twofold: (1) they have the potential to stand-in as data descriptors during publishing; and (2) they can be used for more precise scoping of data subsets to be reported.

We describe *Label*Flow, a semi-automated infrastructure for tracking domain specific provenance with *Data Labels*. We introduce a domain-independent process model comprised of four operators for the **in-situ generation** and **propagation** of labels, predicated on basic information given in the form of semantic workflow annotations, called *Motifs*, that describe the data processing characteristic of workflow steps. We provide a practical algorithm for the generation of *Labelling Pipelines* out of motif-annotated scientific workflows, and provide an implementation where labelling pipelines are realised as functional programs. In prior work [AGB13] we proposed requirements and a preliminary approach; here we present a fully implemented architecture and report results on the impact of availability of labels to provenance queries. We start by introducing a sample real-world workflow and outline the provenance categories and queries for experiment reporting (Section 2). We outline the *Label*Flow architecture in Section 3 followed by details of the proposed solution, including Motif annotations (3.1), the core model for labelling pipelines (3.2), labels (3.3) and labelling operators (3.4). We review related work in Section 4, and conclude in Section 5.

2 Motivation

Figure 1 illustrates a workflow from astronomy[4] that takes as input a set of galaxy names ("list_cig_name"), and outputs extinction/reddening calculations per galaxy ("data_internal_extinction"), and galaxy details such as coordinates and morphology ("ra" "dec" "sesame" "logr25", and "leda_output"). The workflow starts by retrieving data, including coordinates, for each galaxy through a service based lookup from the Sesame astro-repository (Step-1- "SesameXML"). Coordinates are used to query the Visier Database to retrieve further data regarding galaxies (Step-2- "VII_237"). Galaxy morphology information is extracted from the Visier results, which is input together with coordinates into a local tool that computes galaxy extinction values (Step-3-"calculate_internal _extinction"). The scientifically significant activities in this workflow are the data retrievals and the local extinction calculation. The remaining activities are data adapters [GAB+14], a.k.a. shims, which are dedicated to the extraction of data, format transformation or moving data between the workflow environment and the file system. An important adapter in our example is the "Flatten_List" step, which bundles all input coordinates for all galaxies from Step-1 into a single output list for Step-2.

Workflow execution results in a set of intermediary and final data artefacts. For a single galaxy (e.g. M31, the Andromeda Galaxy) a total of 17 final results are generated at 6 output ports. The number of outputs increases linearly with the number of inputs. For a list of 6 galaxy names supplied as input, we get 20+ values for extinction and 100+ values for all results. This illustrates how workflows as automation tools proliferate data generation and makes apparent that manual annotation of data artefacts would quickly become a challenge for users.

The provenance landscape for workflow-generated data contains two categories of information

(i) *Generic*: Standard (Workflow) Provenance vocabularies make-up this category. They capture activities, input/output ports, activity instantiations, and data artefacts appearing at ports. Data influence and activity causality relations are also represented at this layer [Ge12] [BC+12].
(ii) `Domain Specific`: Field-specific vocabularies for describing the scientific context and characteristics of data and experiments make up this category. The importance of domain-specific metadata has been acknowledged early-on in provenance research; 5 out of 9 of the Provenance Challenge queries [MLA+08] are based on restrictions on either data values or "annotations", which are "assumed" to exist. Domain specific annotations can further the categorised as containing **`Static`** or `Dynamic` metadata. The

[4] http://www.myexperiment.org/workflows/2920.html

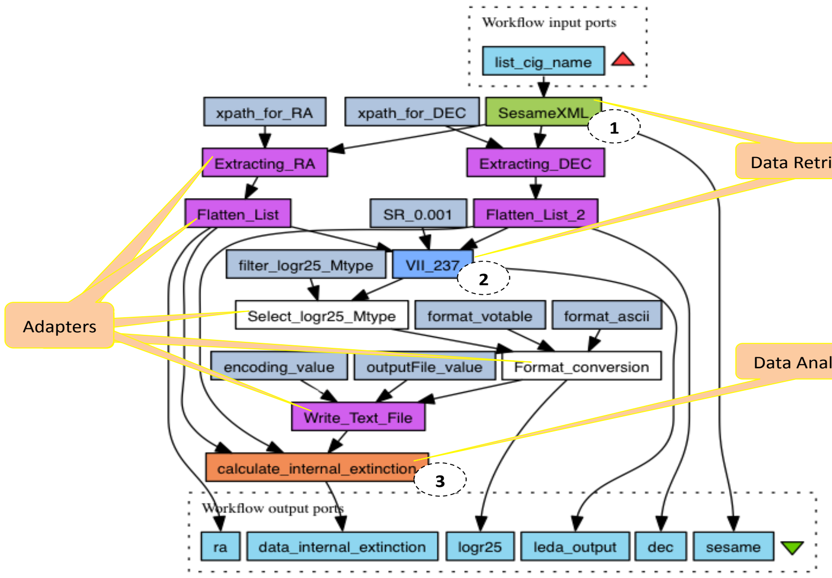

Fig. 1. Sample workflow from Astronomy developed by the Wf4Ever project.

former identifies fixed/general domain types for activities or their inputs and outputs. E.g. Specifying that an activity is a SesameDB lookup, a parameter is a galaxy name. Dynamic metadata corresponds to attributes of data that can change from run to run. This information is often to be found innately but implicitly within data values, e.g. the galaxy name input parameter such as M31 or M33.

Let's now look at the state of the art in reporting with the *Lineage-Based Approach*, and compare with our proposed *Label-Based Approach*. In the former we only have generic workflow provenance to query, in the latter we employ *Label*Flow to obtain domain-specific annotations, which we later query.

Lineage-Based Data Selection: One can use workflow provenance to select data subsets by using lineage as a scoping mechanism. For instance, querying for results that are on the derivation path of a particular input artefact, or those whose derivation includes a particular activity. Table 1 presents three traditional lineage queries; Q1a, Q2a are adapted from [ZS+11], and Q3a is an adaptation of Provenance Challenge Query #6 [MLA+08]. Queries are font-highlighted to denote the different layers of provenance metadata needed to support them.

We analyse queries with respect to their **Contextual-Precision**, which we define as $\frac{\#of\textbf{Contextually-Accurate}results}{Total\#ofresults}$. We define **Contextual Accuracy** as the results actually belonging to the scope implied by the query (e.g. for Q1a the results that actually contain data that is retrieved from the Sesame database, or for Q2a the results that actually contain data belonging to galaxy M31).

Lineage	Q(1a) Find all *outputs* whose *derivation path* includes a **SesameDB lookup**.
	Q(2a) Find all *outputs* whose *derivation path* includes *input* with value M31.$^{\text{see caption}}$
	Q(3a) Find all **extinction values** that is *output from* **extinction calculation** where the **galaxy coordinates** *taken as input* have been *directly/indirectly outputted from* a **SesameDB lookup** with a **galaxy name** *input* with value M31.
Label	Q(1b) Find all *outputs* who has referenceURI http://cds.u-strasbg.fr.
	Q(2b) Find all *outputs* who has subject M31.
	Q(3b) Find all **extinction values** that is *output from* **extinction calculation** where the **galaxy coordinates** *taken as input* has referenceURI http://cds.u-strasbg.fr and has subject M31.

Table 1. Provenance queries to select results of interest from the execution traces of workflow in Figure 1. In Q(2a) we locate the specific data artefact with value M31 prior to formulating the query.

Q1a queries for the origin of data by expressing it as a path-based linkage to the "Sesame XML" activity in the workflow description. This way of designating the origin proves to be a weakly precise yet robust filter (see Fig 2 (left)). Only one third of the results whose derivation path includes a Sesame DB lookup actually contain data that is retrieved from the Sesame DB. Increasing the number of galaxies in a workflow run does not diminish the precision of Q1a. **Q2a** defines a filter for results belonging to the Andromeda Galaxy by expressing it as a path-based linkage to the data artefact at the galaxy name input port with value "M31". While Q1a puts constraints on workflow description level entities, Q2a puts restrictions on run-time provenance-level entities. As depicted in Fig 2 (right) the precision for Q2a quickly deteriorates. **Q3a** is a more elaborate query that combines the metadata requirements of Q1a and Q2a. Q3a is not robust against input data increase either. The fragility of queries that make use of dynamic elements (Q2a, Q3a) is due to the well-known Black-Box nature of workflow activities. For our case specifically, the "Flatten_List" step, which bundles all input coordinates for all galaxies into a single output list. At this point we lose fine-grained traceability between a specific galaxy name and the relevant data generated downstream in the workflow. As our example demonstrates, in the face of loss of fine-grained traceability, path-based querying of provenance becomes an ineffective index for reporting.

Annotation with *Label*Flow and Label-Based Selection: In order to employ *Label*Flow, as a pre-requisite we developed two simple functions that extract attributes (labels) for astronomical datasets from their XML based representation. We associated these functions with the "SesameXML" and "VII_237" activities, so that whenever these two data retrieval activities are used in a workflow they would have an associated labelling capability denoting the data's origin using an endpoint and its context i.e. the astronomical object it belongs to. We also semantically annotated data adaptation steps in our astro-workflow to give them basic transparency to denote whether inputs are carried-forward to (copied-to) outputs. Using this information *Label*Flow creates a labelling pipeline, which

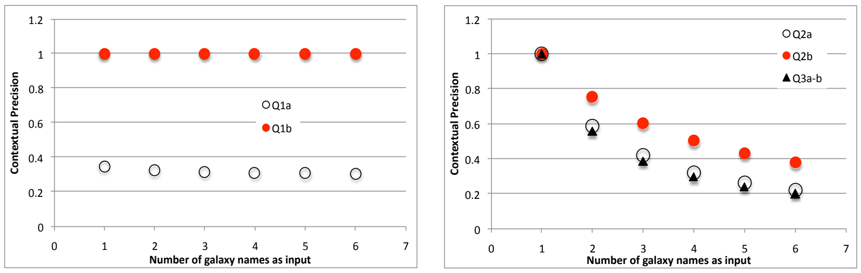

Fig. 2. Precision values for Q1 (left) and for Q2 &Q3 (right) with respect to input size.

we use to decorate the runs of our workflow with labels. Labels have two potential uses, as descriptors during publishing and as data selection aides. In this work we explore the latter use of labels.

Table 1 also presents label-based data selections queries Q1b, Q2b and Q3b. In these we directly refer to the asserted origin (`has referenceURI`) and the asserted context (`has Subject`). Label-based queries Q1b and Q2b have higher precision then their lineage based counterparts (see Fig 2), which can be explained as follows. First, lineage-based association is by-definition only a pseudo mechanism for denoting origin/context. By replacing lineage-based association with explicitly asserted attributes we gain in precision, as now only the data items that originate from the Sesame DB, and their local copies are returned to **Q1b**. Secondly, loss of fine-grained traceability also affects label-based query precision, see **Q2b** in Fig 2 (right). While each item output from "SesameXML" bears the correct label denoting the associated galaxy, all items in the output of "Flatten_List" would bear a set of labels (for all galaxies), even though each contains the data of one. This time, however, *Label*Flow offers the possibility of asserting/recovering context in other data minting steps ("VII_237"); the labelling function associated with this step would exploit the raw data returned from the Visier DB and associate each result item with its context using a common attribute (`has subject`). In precision **Q3b** and Q3a are of equal capability in filtering (Fig 2 (right)). This shows us that even though Q3b makes use of labels, it queries workflow results with reference to a particular blindspot (i.e. output of "Flatten_List") and therefore has precision performance equivalent to lineage-based queries. Thus, lineage-based queries represent the bottom-line (worst-case) precision for data scoping, where availability of labels offers the possibility of increased precision (at varying levels depending on existence and frequency of activities where fine-grain traceability is lost). In the remainder of the paper we describe the *Label*Flow infrastructure.

3 The *Label*Flow System

Figure 3 provides the overall architecture of our approach. We undertake labelling as an offline process, where we do not interfere with the established process of scientific workflow design (Step A1) and execution (Step A2). Workflow runs result in the generation of data artefacts and generic workflow provenance. These two make up our primary sources of information for obtaining and propagating domain-specific *Data Labels*. We perform labelling through latent processes informed by scientific workflow descriptions themselves enriched with semantic **Motif** annotations and associated **Labelling Functions**.

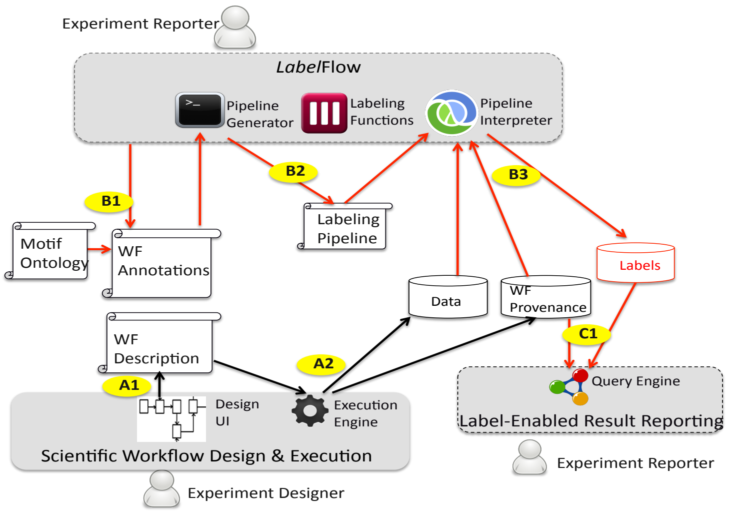

Fig. 3. Labelling System Architecture.

We operationalize the process model with **Labelling Pipelines**. Labels are opaque to the process model, as it out-sources their creation to external **Labelling Functions**. Using motif annotations (Step B1 in Figure 3) and a repository of labelling functions we compile (Step B2) a labelling pipeline for a given scientific workflow. This pipeline is in-turn used to annotate the desired execution traces of that workflow with labels (Step B3). Once labels are generated they can be used in conjunction with generic workflow provenance metadata for the reporting of experimental results (Step C1).

3.1 Annotation of Workflow Activities with Motifs

In a previous empirical study [GAB+14] we inspected a corpus of 240 workflows from 4 systems and 10 domains in order to understand the nature of data processing in them. This resulted in a catalog of *Motifs*, a set of high-level abstractions for describing activity functionality. The analysis showed that a certain minority (30%) group of activities perform the scientific heavy lifting in a workflow by minting data through analysis or retrievals. The remainder majority (70%) are dedicated to data adaptation. A common characteristic of adapters, is that their computation is based on *value-copying* from inputs to the outputs. It follows then that we should seek labels for data artefacts that are generated by Data Minting activities, and grab hold of labels as data passes through (i.e. copied through) Data Preparation activities. These two categories of behaviour form the backbone of our labelling system. In Table 2 we list a subset of motifs with examples (including those from our astro-workflow as applicable) and corresponding labelling behaviour. Motifs are captured in an ontology, which we use to manually annotate activities. This basic annotation is in turn used to infer the data handling behaviour of each step. Annotation is finalised by collecting the particulars from the user; for value-copying, the source and sink ports, and for data minting the associated Labelling Function (if any) and the sink port to receive labels. Note that we scope our approach to scientific **dataflows**, i.e. those without any explicit control construct such as looping or branching. The pure dataflow model underpins several systems such as Taverna [MSRO+10], Galaxy [5] or Wings [GRK+11]. In others like Kepler [LAB+06] and Vistrails [MSFS11] pure dataflow model is widely adopted, while control-constructs are add-on modules or supplied in alternative design modes. We also assume that data is structured as Collections-Items, which is a ubiquitous structure for scientific workflow systems.

Motif	**src→snk**	**Example**	**Labelling**
`Data_Minting`	$I\xrightarrow{m-1}O$	“SesameXML”, “VII_237”,“calculate_int_extinction”	mint
`Augmentation`	$I\xrightarrow{m-1}O$	Adding a header to a CSV dataset.	propagate
`Extraction`	$I\xrightarrow{1-m}O$	“Select_logr25_Mtype”, “’Extract_DEC&RA”	propagate
`Splitting`	$I\xrightarrow{1-1}O$	Splitting a dataset by newline char.	propagate
`Flattening`	$I\xrightarrow{1-1}O$	“Flatten_List”	propagate
`Filter`	$I\xrightarrow{1-1}O$	Filtering empty rows from a CSV.	propagate
`Join`	$I\xrightarrow{m-1}O$	Row by row dot product of two CSV tables.	propagate
`Union`	$I\xrightarrow{m-1}O$	Concatenating two CSV tables.	propagate

Table 2. Workflow Motifs, Value Copying and Corresponding Labelling Behavior

3.2 Labelling Pipelines

We provide a tool which takes as input a motif annotated workflow description w and produces a labelling pipeline Π_w for this workflow. Π_w could in turn be used to annotate data artefacts generated from all runs of w. A pipeline generator implements an algorithm based on the traversal of all dataflow paths in w. For each workflow element (i.e. activity or dataflow link) the tool checks the availability of motif annotations and label-flow continuity and accordingly places

[5] http://galaxyproject.org

an operator into Π_w as a labelling proxy for that element. We note that this algorithm can operate with partial/missing annotations; in the case of missing motif annotations, the generator simply registers the current stack of connected labelling operators as a labelling sub-pipeline and resets. The algorithm initiates a new thread in the labelling pipeline whenever it encounters an activity that mints new data. To coordinate inter-operator communication among labelling operators we use simple *runs-after* type control tokens. The output of the *generator* tool is an intermediate representation for a labelling pipeline which is further expanded into a runnable form using the syntactic/macro expansion capabilities of a functional programming language.

The inputs to a particular execution of the labelling pipeline Π_w is the 6-tuple $\langle d, p, l, v, F_L, F_P \rangle$, where p, denotes the provenance trace of one run of workflow w, and d denotes the set of data artefacts generated during that run. The domain specific provenance represented with labels is accumulated in the label space l. v is the labelling vector that the system will take into account for label propagation. The system relies on sets of predefined functions, F_L for provisioning labels and for management of the label space (read-write) and F_P for querying generic workflow provenance.

3.3 Labels

A label is in effect a Label Instance that is defined with the triple $L_{ins} = \langle def, target, value \rangle$. def refers to the label's type, $target$ is the id of the data artefact, which the label describes, and *value* is the actual annotation content carried by the label. Label definitions are triples of the form $L_{def} = \langle name, datatype, f_{agg} \rangle$. They have a unique *name* and a *datatype* designator. Labels can contain primitively typed information such as $Integer$ or $String$. f_{agg} is the identifier for a function to be used when the system needs to aggregate multiple labels of this type. For the majority of labels, this element is nil, in which case the default aggregation function, i.e. $Union$, is used. A non-default case is, for example, the *spatialaggregation* function which computes the convex hull representing the overall spatial coverage of multiple datasets. Label definitions are grouped together in Label Vectors, $v = \langle name, \{L_{def}\} \rangle$. When used to configure the run of a pipeline Π_w, the vector sensitizes Π_w to the label types that it contains. Label and label vector definitions are to be made at the scientific investigation level, which spans multiple workflow descriptions.

3.4 Labelling Operators

Labelling pipelines are compositions of four labelling operators, namely $Mint$, $Propagate$, $Distribute$ and $Generalize$ (Figure 4). In addition to input parameters, each operator accesses the provenance space, and depending on the labelling behaviour, accesses either the data artefacts (in case of *mint*) or the label space (others). Each operator has the side-effect of populating the label space. Operators return a boolean control token that is used for composing multiple operators into a labelling pipeline:

- *Mint* is a labelling proxy for those scientifically significant steps in the workflow. *Mint* obtains labels by invoking the designated external labelling function; the labels are then associated with the data artefacts that fulfil the sink port and submitted to the label space. Minting is iterated for all invocations of the designated activity found in the provenance trace.
- *Propagate* is a labelling proxy for the value-copying Data Preparation steps in the workflow. Similar to mint, it is iterated for all invocations of the designated activity. *Propagate* clones labels describing the inputs at the source port and associates these clones with the outputs at the sink port.
- *Distribute* and *Generalize* are variants of propagation. While the former two are labelling proxies for activities, these are labelling proxies for dataflow links in the workflow, specifically those links with data structure depth mismatches between the two ends. In cases where the activity at one end of a dataflow link produces a collection, and the other end consumes an item, *Distribute* is responsible for propagating labels from the top-level collection to each item at specified depth. And vice-versa for *Generalize*.

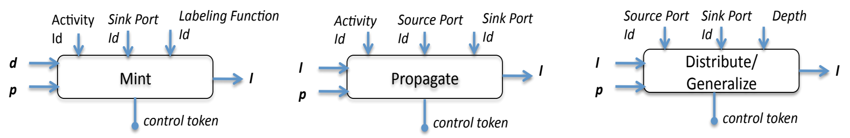

Fig. 4. Labelling Operator Signatures.

3.5 Implementation

The provenance and the label spaces are underpinned by RDF based metadata. *Label*Flow can operate over standard PROV [Ge12] + Wfprov[BC+12] compliant provenance traces. Our provenance inquiry functions in the p space are implemented as Java methods. We implemented labelling operators as Java methods and labelling pipelines as Clojure programs that adhere to the dataflow paradigm [6], though in our case we *flow* control tokens among operators and the inter-operator communication regarding labels is done over the shared label space. The Label*Flow* system is agnostic to the inner workings of labelling functions. For our example from astronomy we had a simple local registry of labelling functions, which are Java classes adhering to a label generation interface.

4 Related Work

As mentioned previously, provenance annotation has so far been either entirely manual, or semi-automated with particular focus on static metadata [MSZ+10].

[6] http://clojuredocs.org/clojure_core/clojure.core/future

In [SSH08] authors describe the SPADE system where they highlight dynamic metadata, and they too address data artefacts as the source of this information. The authors propose "semantic provenance modules" to supply this metadata and claim modules can be integrated into workflows on-demand, though details of the integration are omitted. When compared to our work, this work is focused on devising an elaborate provenance ontology for one particular scientific domain, whereas ours is a domain-independent mechanism. Moreover the SPADE system requires altering the original scientific workflow to denote integration points, while ours is non-intrusive to the workflow design and execution process Finally SPADE does not address metadata propagation.

There is a large body of work on the provenance of database queries, which is recently revisited for its applicability to workflow provenance [AD+11][IC+][BL06]. These approaches propose white-box workflow activities that correspond to relational query operators. The benefit of white-box steps is that they allow full-transparency and enable fine-grained lineage, also making way for the tracking of cell-level value-copying and annotation propagation [BC+04]. Similarly, work in the area of dependency analysis from programming languages has recently found applicability as a formal foundation for tracking of Nested Relational Calculus query provenance [CAA07]. Such white-box transparency could be instrumental in developing workflow debugging or change tracking aids. On the other hand, these approaches expect data to be specified in relations and tuples, and reduce data-processing to data-querying; both which can be restrictive assumptions for developing scientific workflows. In contrast, we focus on the unexplored area of grey-box steps, and denote value-copying through a rough-cut semantic annotation.

5 Conclusion

We described a semi-automated approach and an implemented architecture for the generation of Labels over data artefacts generated from runs of workflow based experiments. Labelling is performed through labelling pipelines, which use data artefacts as the main source of information for extracting domain-specific metadata and workflow provenance as a roadmap for association and propagation of labels with data. Pipelines are built up using four domain-independent labelling operators, which are agnostic to the contents of the domain-specific labels they carry around.

We argue that experiments organised as workflows make-up an ideal medium to capture and carry domain-specific provenance. Labels, i.e. carriers of this information, stand as a light-weight but controlled representation mechanism for metadata, which is a middle-ground between having no explicit metadata and having fully-fledged models that can represent complex/structured metadata. The benefit of labelling is two-fold: not only does it make implicit information explicit, but it also enables provenance queries that directly refer to scientific provenance/context rather than expressing context indirectly it in terms of derivation paths.

The cost involved in adapting our system is the manual annotation of workflow activities with motifs and developing labelling functions for the focal data generation points in workflows. These are one-time costs. Both motif annotations and labelling functions are highly reusable as most workflows are built by re-using building blocks pooled in module libraries or service registries. Consequently an annotation or a labelling proxy for a building block propagates to all workflows that the block is involved. When compared to workflow design, the cost of annotation is modest(as it amounts to single attribute setup per activity). Moreover motif annotation can be (semi)automated through the application of mining techniques to workflows and activity scripts [GCP13]. The re-usability of labelling functions can be maximised by developing metadata extraction utilities that operate over standardised scientific data formats.

*Label*Flow is currently being evaluated to understand the trade-off between label accuracy vs coverage of labelling. Results are made available at [7] as they become available.

Acknowledgements The work has been supported in part by the award EP/G026238/1 myGrid: A Platform for e-Biology Renewal, and enabled by collaborations with the EU FP7 STREP 270192 Wf4Ever Advanced Workflow Preservation Technologies for Enhanced Science and EU FP7 283359 BioVel BioDiversity eLaboratory.

References

AD$^+$11. Y. Amsterdamer, S. B. Davidson, et al. Putting lipstick on pig: Enabling database-style workflow provenance. *PVLDB*, 5(4):346–357, 2011.

AGB13. P. Alper, C.A. Goble, and K. Belhajjame. On assisting scientific data curation in collection-based dataflows using labels. WORKS '13, pages 7–16, New York, NY, USA, 2013. ACM.

BC$^+$04. D. Bhagwat, L. Chiticariu, et al. An annotation management system for relational databases. In *(e)Proceedings of the 13th VLDB Conference*, pages 900–911, 2004.

BC$^+$12. 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.

BL06. S. Bowers and B. Ludscher. A calculus for propagating semantic annotations through scientific workflow queries. In *In 11th Intl. Workshop on Foundations of Models and Languages for Data and Objects, LNCS*, 2006.

CAA07. James Cheney, Amal Ahmed, and Umut A. Acar. Provenance as dependency analysis. In *Proceedings of the 11th International Symposium on Database Programming Languages (DBPL 2007), number 4797 in LNCS*, pages 138–152. Springer-Verlag, 2007.

DF08. S. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD Conference*, pages 1345–1350, 2008.

GAB$^+$14. Daniel Garijo, Pinar Alper, Khalid Belhajjame, Oscar Corcho, Yolanda Gil, and Carole Goble. Common motifs in scientific workflows: An empirical analysis. *Future Generation Computer Systems*, 36(0):338 – 351, 2014.

[7] http://www.myexperiment.org/packs/617.html

GCP13. Devarshi Ghoshal, Arun Chauhan, and Beth Plale. Static compiler analysis for workflow provenance. In *Proceedings of the 8th Workshop on Workflows in Support of Large-Scale Science*, WORKS '13, pages 17–27, New York, NY, USA, 2013. ACM.

Ge12. Y. Gil and S. Miles editors. A primer for the prov provenance model, 2012. World Wide Web Consortium (W3C).

GRK$^+$11. Yolanda Gil, Varun Ratnakar, Jihie Kim, Pedro A. González-Calero, Paul T. Groth, Joshua Moody, and Ewa Deelman. Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems*, 26(1):62–72, 2011.

IC$^+$. R. Ikeda, J. Cho, et al. Provenance-based debugging and drill-down in data-oriented workflows. In *ICDE 2012*. Stanford InfoLab.

LAB$^+$06. Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006.

MD$^+$13. 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.

MLA$^+$08. L. Moreau, B. Ludäscher, I. Altintas, et al. The first provenance challenge. *CCPE*, 20(5):409–418, April 2008.

MSFS11. Phillip Mates, Emanuele Santos, Juliana Freire, and Cláudio T. Silva. Crowdlabs: Social analysis and visualization for the sciences. In *23rd International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 555–564. Springer, 2011.

MSRO$^+$10. Paolo Missier, Stian Soiland-Reyes, Stuart Owen, Wei Tan, Alex Nenadic, Ian Dunlop, Alan Williams, Tom Oinn, and Carole Goble. Taverna, reloaded. In M Gertz, T Hey, and B Ludaescher, editors, *Procs. SSDBM 2010*, Heidelberg, Germany, 2010.

MSZ$^+$10. 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.

SSH08. S S Sahoo, A. Sheth, and C. Henson. Semantic provenance for escience: Managing the deluge of scientific data. *Internet Computing, IEEE*, 12(4):46–54, 2008.

TA$^+$11. C. Tenopir, S. Allard, et al. Data sharing by scientists: Practices and perceptions. *PLoS ONE*, 6(6):e21101, 06 2011.

TF$^+$08. Chris F Taylor, D. Field, et al. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nature Biotechnology*, 26(8):889–896, 2008.

ZS$^+$11. J. Zhao, S.S. Sahoo, et al. Extending semantic provenance into the web of data. *Internet Computing, IEEE*, 15(1):40–48, Jan 2011.

ZW$^+$04. J. Zhao, C. Wroe, et al. Using semantic web technologies for representing e-science provenance. In *Proc. of the 3rd International Semantic Web Conference*, volume 3298, pages 92–106, Hiroshima, Japan, 2004.