Re-thinking Workflow Provenance against Data-Oriented Investigation Lifecycle

Technical Report, 2014-05-06

Version:
1.0

Author(s):
Pinar Alper, University of Manchester

Reviewers:
Carole A. Goble, University of Manchester
Khalid Belhajjame, University Paris-Dauphine

Overview

This report is the first installment of a two-part knowledge transfer. In this first part we introduce the status quo and issues, (with underlying reasons and remedies) in exploiting provenance information. The second report will provide details on approaches and techniques that can help towards increased use of provenance information.

We start by revisiting the workflow provenance capabilities of the Taverna system, and the provenance-related information needs during stages of workflow-based scientific investigations. We look at the characteristics of workflow provenance that obstruct its pervasive use throughout the investigation lifecycle, and the current mechanisms that are devised as remedies.

We briefly introduce a framework, in which we investigate techniques for abstraction and annotation of workflow provenance. We introduce Workflow Motifs, a domain-independent categorization workflow activities, a pillar upon which our provenance abstraction techniques are built.

We conclude the report with a set of recommendations on provenance capabilities of Taverna tooling that would provide food for thought for the development team.

Table of Contents

1. Workflow Provenance: Status Quo in the Taverna System

Scientific data provenance is defined as "information that helps determine the derivation history of a data product, starting from its original sources" [9]. Scientific Workflows are an established method for scientists to design data-oriented computational investigations as systematic compositions of datasets and analyses [10].

The stepping-stone of provenance for data artifacts generated from workflow-based analyses is the workflow description itself. By outlining the process followed, as of a network of activities connected by dataflow links, the workflow description makes up an important part of the provenance of (any) results generated from the workflow's execution. Workflow descriptions can get overlooked in the context of engineering provenance modeling, capture and query solutions. Meanwhile scientists consider the **workflow description** itself the most important part of the provenance of their results.

In Figure 1 we give an example Taverna workflow from Heliophysics domain [11]. This is a data-chaining pipeline that gathers and integrates data from multiple scientific databases. The workflow starts by building up a query to be submitted to the "Helio Feature Catalog" (HFC) to retrieve the active regions detected on the solar surface within a specified time frame. Once the query is executed the result string is parsed and each record is further used as a parameter for generating a query to be submitted to the "Helio Events Catalog" (HEC). The event catalog is queried to retrieve solar flares observed within the locality of the active regions identified in the previous query. This flare data is then aggregated and appended to the Active Region dataset as a flare count in the "MergeHessiField" step. The workflow also joins the HFC and non-empty HEC results to obtain an overall view of flare events and their regional location.

Workflow descriptions are represented in languages that are often specific to the scientific workflow tooling. Meanwhile, their importance as provenance artefacts has recently prompted platform-independent abstract workflow models to be devised as part of provenance models. Examples are OPMW [37], P-Plan [38], D-

PROV [39] and Wfdesc [15]. The Taverna system allows for representation of Scufl workflows with the Wfdesc model in its current provenance capability.

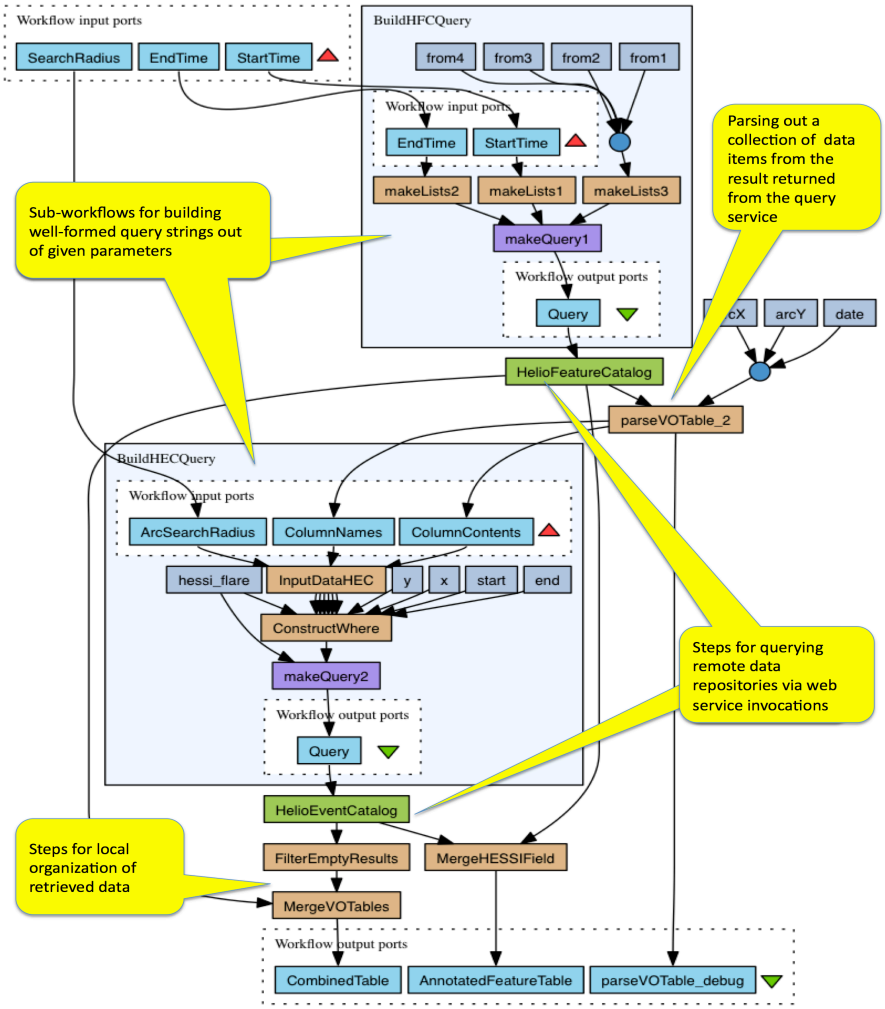

Figure 1 Sample Taverna Workflow from Heliophysics Domain

The workflow descriptions as the one described above is only half the picture for provenance of results. When workflow descriptions are run with input data, this results in the generation of output data and execution provenance. Execution provenance contains 1) **process-oriented information** reporting the execution statuses of activities, their causal ordering and the consumption/production relations among activities and data 2) **data-lineage information** interconnecting outputs to intermediary results through to inputs using "influence" or "derivation" links

Taverna provides two ways of accessing results and provenance:

1) During execution Taverna allows the user to view execution status, to display the intermediate and final results generated via the Taverna Results Panel in the Workbench, or the tabbed results page in the Portal. Through this interface, information on execution statuses of activities,

values of data artifacts can be seen. Moreover, if there are custom viewers/visualizers for data, Taverna allows plugging of such utilities into the results panel.

2) Users are given the ability to export the resulting data artifacts and the execution provenance in standard vocabularies, specifically a combination of PROV-O [12] (from the W3C) and WfProv [15] (from the Wf4Ever project). Earlier versions of Taverna also supported OPM [40] compliant export.

In Figure 2 we give a subset of the execution provenance captured per invocation of the "ConstructWhere" processor in our example workflow. This process-oriented part of provenance describes the activities that took place, their start and end times, the agents involved with the activity, and the qualified data consumption and productions of activities.
The Data-lineage part of execution provenance is expected to report on "derivation" or "influence" relations among data artefacts[1]. In Figure 3 we depict this using the "wasInfluencedBy" relation from the PROV-DM model that interconnects output artefacts of the "ConstructWhere" pocessor invocation to the input aretefacts. Note that as of recent developments in provenance modeling [12] the term "derivation" has been given very specific semantics referring to "the generation of new data artefacts through updates or revisions to existing data artefacts". In the current Taverna provenance capability data-lineage is (rightfully) not represented with "derivation" links. (We foresee future versions of this utility may adopt the "influence" relation to assert a vague lineage among data artefacts).

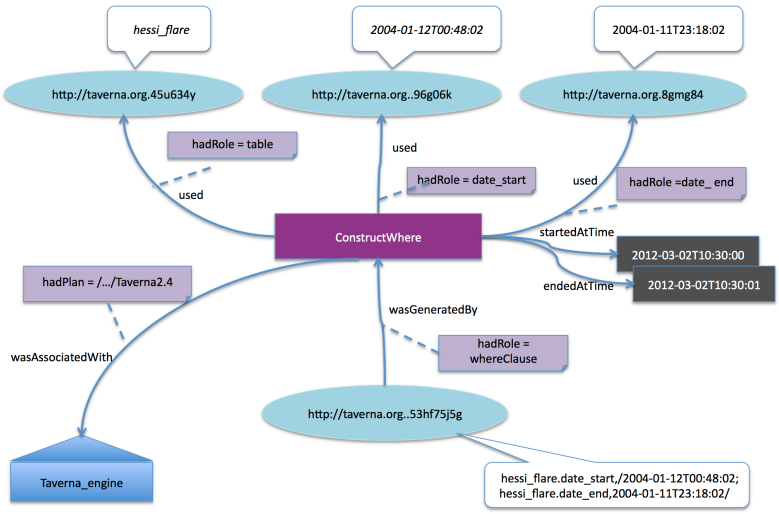

Figure 2 A subset of the process-oriented provenance trace captured for a single invocation of the "ConstructWhere" processor. This trace adopts the PROV-DM conceptual model and the graphical illustration follows PROV Graph Layout conventions[2]

[2] **http://www.w3.org/2011/prov/wiki/Diagrams.**

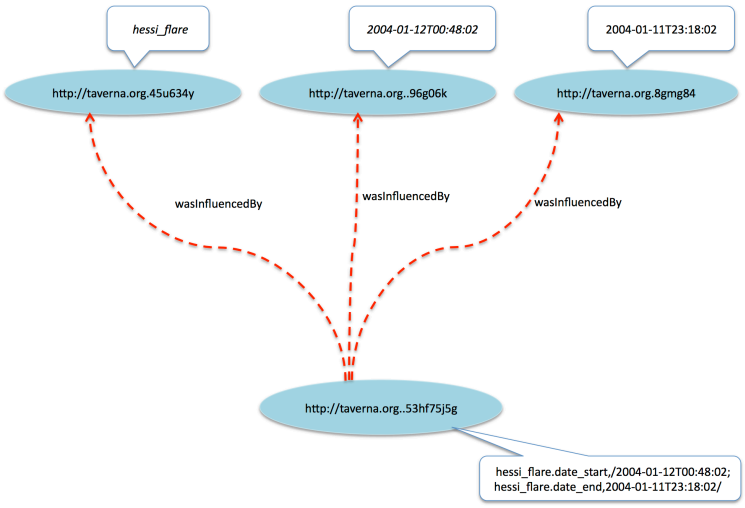

Figure 3 An adoption of the PROV "wasInfluencedBy" relation to denote vague data lineage among outputs and inputs of the "ConstructWhere" processor invocation.

The provenance trace for the execution of the entire workflow is in turn a chain of such activity execution and data influence traces.

2. Workflow Based Experiment Lifecycle

There are several lifecycle models for the emerging paradigm of data-oriented scientific investigations [34] [35] [16]. For those investigations that are realized with workflows in [16] Goble et.al. give a 4 staged lifecycle (this is in the context of life sciences, meanwhile our review did not identify a contradicting case from other domains):

Workflow Design (I) and Execution (II): Scientists start their investigations by designing the intended workflow, identifying the services, tools and resources to be used, their data and control dependencies expected inputs and configuration parameters. Design is followed by execution. Often scientists iterate through a fast-paced design-execution cycle until they obtain the finalized design of the investigation. Finalized designs act as best-practices or protocols for data-oriented investigations. Therefore often workflows, once the design is settled, are executed multiple times with different input data/configurations. As we shall see later provenance plays different roles for the design and execution stages.

Result Analysis (III) and Publishing (IV): The results generated from executions are retrieved, inspected, and visualized by scientists during the analysis stage. This is the stage where (if any) scientific insights, are made hypotheses are validated/revised and study scopes are (re)set. Findings at this stage might necessitate updating the workflow design, changing run parameters, or even prompt further physical data collection/labwork. Once the scientists are content with the results the final phase is the publishing of experimental work products

and the publishing of (selected) results to public or private repositories for future reference/reuse.

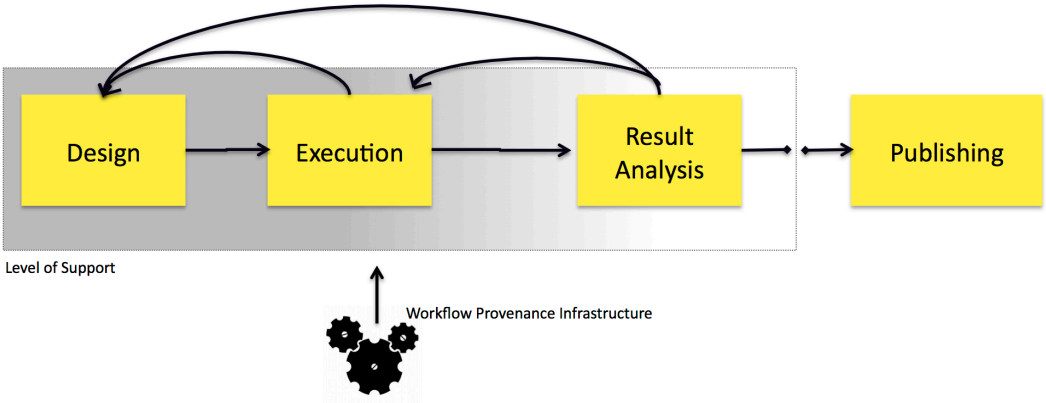

Figure 4 Data-Oriented Scientific Investigation Lifecycle and the Level of Provenance Support

We observe that the existing models for representing workflow provenance and the mechanisms to query it have found intensive application in the design and execution stages of this lifecycle. Use for result analysis is sporadic, whereas use for publishing is minimal (almost none). Let us elaborate on each phase with examples.

I. **For design:** An application area not explored in the Taverna system but exists in Kepler and Vistrails [22] [20] is the tracking of the provenance of the workflow design itself. Capturing of workflow versions, revisions and keeping track of the use/inclusion of activities across workflow descriptions fall into this capability. The requirements for these largely stem from domains, where workflow design is a particularly complex, lengthy, and exploratory process. The following can be answered by tracking and using provenance at this state:

I.a. Which are the workflows that have been derived from this workflow?
I.b. At which version has this activity been added?
I.c. In which other workflows is this activity included?
I.d. Get me the latest revision of this workflow.
I.e. Get me other workflows that contain data treatment steps with effect similar to this example (transition of raw image into a smoothed image obtained from previous runs given as example).

Vistrails provides the epitome of design-time provenance support, it even integrates execution provenance for tracking and finding workflows during design (queries like (I.e)).

II. **For execution:** Most workflow systems including Taverna, capture and query provenance to achieve seamless workflow execution [8] [17] [19] [20]. Process-oriented execution traces are used for monitoring and debugging of runs, or for workflow steering (i.e. partial re-runs) by reusing

results of previous executions [21] [20] [23]. Almost all workflow systems provide some support in their tooling to view this information (e.g. Run/Execution panes). Using such information the user can find answers to below:

II.a. What are the runs with failed invocations of this service?
II.b. Which service(s) is/are responsible for the failed execution of this workflow?
II.c. What are the maximum and minimum execution times for runs of this workflow?
II.d. Re-run the workflow using cached results from yesterday's execution.
II.e. Are there any freshly generated intermediary results in this re-run?

Another critical use of provenance at a post-execution stage is for auditing or reviewing of previous runs. Certain workflow systems such as Vistrails and KNIME have tight integration of result data and the workflow descriptions (i.e. whenever one opens up a workflow, cached results from the most recent (or a designated prior) run, will also be opened up). Emerging tooling like Taverna 3 Alpha also allow runs to be saved and opened up for review at a later time.

The current mechanisms for users to perform reviews are to open up runs and perform a manual inspection. By combining the execution trace information together with utilities at the presentation layer such as node selection, traversals, highlighting it can be possible to answer below queries using execution provenance:

II.f. Which result files are influenced by this input calibration parameter?
II.g. Which inputs have influenced this erroneous output?
II.h. Among a set of runs of multiple workflows which consecutive workflow run have consumed the output of this run as its input?

III. **For Result Analysis:** Science is exploratory, and so are workflow-based investigations. Scientists will run workflows several times with differing parameters. At the result analysis stage scientists may use provenance to select data subsets of interest among the larger pool of workflow results. Scientists may ask for results that fall into a particular context, a particular time frame, or those obtained from a particular external source. Such selections necessitate queries that transcend process-oriented workflow provenance and data content. Selected results are then inspected, compared or analyzed to gain scientific insights, validate hypotheses or (re)define study scopes. Examples inquiries are:

III.a. List all VOTable results containing Helio Event data outputted from workflow X, where SearchRadius was 100 and temporal coverage falls within May 2013. (Workflow X could be the one in Figure 1)

III.b. Find all Galaxy Extinction results output from Workflow Y for the M31 galaxy and where the galaxy coordinates used was obtained from the Sesame DB. (Workflow Y could be the one in Figure 4)

III.c. Have these N runs generated the same output (same with an ε boundary). i.e. pinpoint the difference between multiple runs.

III.d. Launch the results of the last N runs with domain specific tooling (e.g. a visualizer).

No workflow system fully supports all the inquiries exemplified above. Taverna, KNIME and Vistrails supports (III.d) by integration with domain specific data visualizers. While Taverna and KNIME support visual inspection of selected results one at a time, Vistrails provides more provenance-aware integration by parameter exploration, multi-view and comparative visualizations.

Queries of kind (III.a) and (III.b) are based on examples from the Provenance Challenge, and they transcend provenance information and data content[3]. No system provides out of the box capabilities for such queries. System like Wings and Galaxy should particularly be mentioned as exceptions, these systems have a strong focus for domain specific typing of data, where types identify the attributes a dataset can have. Wings types are semantic descriptions, whereas Galaxy supports a set of predefined types for bioinformatics. Having rich typing and run-time populated attributes enable queries (III.a) and (III.b) without the need for preprocessing.

IV. For Result Publishing: Experimental work-products, i.e. workflows, the data and the execution provenance combined, make up a machine-actionable substrate [33] that documents the assumptions, methodology and results of the investigation. Often this substrate is published alongside research articles as a machine actionable form of supplementary material.

And recently, in domains such as Biodiversity, Astronomy or Systems Biology, scientists are encouraged (or sometimes mandated) to submit their backing data into community repositories [1] [24] [25]. To ensure such shared datasets can be preserved and can later be re-used by others, three kinds of metadata needs to accompany data [6]: 1) A **Reference** to unambiguously identify a dataset 2) **Provenance** that specifies datasets derivation history, origins and ownership, and 3) **Context** that outlines the dataset's relationships to other data, its dependencies and scope. Below inquiries exemplify the kind of Provenance and Context metadata required:

IV.a. Is this derivative or primary data?
IV.b. If it is derivative, what is the origin dataset/database?
IV.c. What data citations should accompany this data?
IV.d. Which species/taxa does this dataset contain records for? (or similarly, Which galaxies does this dataset contain records for?)
IV.e. What is the spatial/temporal coverage of the dataset?

Clearly, standard workflow execution provenance cannot answer these, as it does not explicitly capture any domain-specific information (with the exception of Wings and Galaxy for queries (IV.d) & (IV.e)). Either the scientist or the repository curator creates rich metadata, including above information through manual annotation at data publishing time. In current practice, in order to ensure

[3] Although all teams participating in the challenge have answered queries (a) and (b), most have preprocessed provenance to annotate it with data content and hand-crafted the queries over the provenance store.

that a controlled/agreed set of attributes are reported, each domain has devised their own vocabularies and annotation tooling. The SysMO project's Rightfield tool, the GBIF Initiative's Integrated Publishing Toolkit or DataONE project's DataUP tool are examples [28] [29] [27] [26].

While this kind of domain-specific metadata is not tracked by most workflows systems, it is this type of metadata that is considered to be the scientific provenance of datasets. And here lies the provenance gap. Given that the majority of scientific workflow systems provide significant capabilities for collecting workflow execution provenance [10], one would expect such traces to have some utility in stages (III) and (IV) to:

- help select datasets for analysis or publishing
- obtain some of the publishable "**Provenance**" and "**Context**" metadata.

That is, however, not the case. While raw provenance traces can be published **as-is** within supplementary material packages[4], they are hardly ever exploited for data publishing.

3. Closer Look at the Provenance Gap

The reason for the lack of usage of workflow provenance can be attributed to its following characteristics:

1. **Lineage Genericity:** In cross-domain workflow systems (like Taverna) the activities executed via the workflow engine appear as a black-box to the provenance collection framework. Consequently provenance is rich and informative on the process-oriented aspects of execution, but it is vague and uninformative on the data aspects. Data lineage can only be specified with an opaque "derivedFrom" [5] or "influencedBy" [12] relationship. As the nature of relationships between data items are not made explicit, one can not tell whether an input is a query responsible for the retrieval of an output, or whether an output is a cleaned version of the input [31]. Provenance researchers have proposed to qualify lineage either through manual annotation of traces (early research using Taverna [36]) or through intensive adoption of semantic annotations and rules at the workflow design level, which are used at execution time to generate domain specific lineage relations (as in the Wings system [19]).

2. **Lineage Coarseness:** Due to black-boxes, workflow execution provenance informs us superficially on the dependencies between granules of data generated during the run. In most systems (excluding paper prototypes) provenance reports that for each invocation of an activity, all output data at all ports are "influencedBy" all input data at all ports. There are two ways to report more fine-grained lineage 1) at the port level specifying if some output ports depend on a subset of the input ports and 2) at the data artefact level, where we specify lineage between

[4] Often workflow execution provenance is treated as an esoteric form of metadata by scientists; treated as a core dump, which they include in experimental reporting packages just in case the exact experimental steps require audit or inspection.

items within collection-type input and output artefacts (Figure 5 gives a depiction).

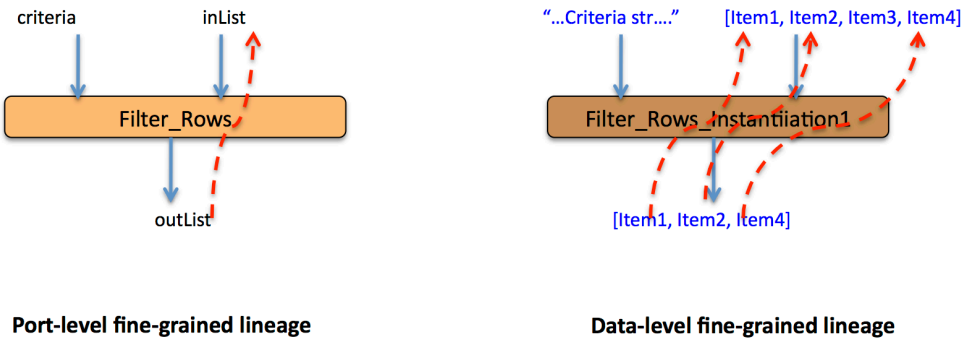

Figure 5 Different Types of Fine-Grained Lineage for Workflow Activities

.

Superficial all-to-all type lineage reporting affects Stage (III) and Stage (IV) inquiries as it causes large number of low precision results to be returned to queries such as (III-a) (III-b). Provenance research has explored how to track and query fine-grained provenance when one uses database query operators (Select-Project-Join) [23] or Pig-Latin [41] programming for the realization of workflows [30]. Given the transparency of SPJ/Pig operators one can report at a fine-grain the data derivations and showcase its utility in use cases like debugging or what-if analyses. Both [23] and [30] are research prototypes and have not found applicability in actively used workflow systems.

3. **Data Opaqueness:** Queries such as (III-a) (III-b) and (IV-d) contain references to the domain types of data artefacts (i.e. galaxy, species, votable). In the absence of domain specific information the only option to represent these is to write "mindful" queries that refer to the correct port names, which (one hopes) are named mindfully to be informative of their runtime content. The solution that research has offered to this problem is through domain-type annotations stated at workflow design stage, which are propagated to data artefacts during run time [32] [19.] That said, these annotations could only capture static characteristics of data that are unchanging across all invocations of the workflow. For instance using annotations one would be able to specify that a particular output is an Image**.** This information is useful but insufficient for answering (III-a) (III-b) type queries, an important part of context metadata is related to dynamic characteristic of results that could change from run to run, such as, stating that the output is an Image of Galaxy M31 or M33 depending on (run-time) parameters.

4. **Storage schemes that separate data and metadata**: Almost all workflow systems support different storage schemes for provenance metadata and data artefacts. And for most systems provenance records mention data by reference. This is mindfully so, as data artefacts may come in all sizes and shape (binary etc.) and it is natural to have a separation. On the other hand, in order to support queries that transcend data and provenance such as (III-a) and (III-b), highly-controlled systems like Wings and Galaxy have the capability to instantiate domain-specific

descriptions at run time and populate design-time identified properties with run-time data.

5. **Implementation Realities of Workflows:** As can be seen in our example Heliophysics workflow, Scientific Workflows can be complex with many scientifically significant steps, such as analysis, visualizations, combined with many mundane steps, such as format transformers and data IO. This complexity arises due to realities of implementing workflows in open-world environments where such adapters are a necessity to incorporate 3rd party resources into the workflow. Complex workflows lead to deep lineage traces that often contain redundant copies of the same data content (e.g. line-wise split versus merged version of a CSV file).

   Moreover Execution provenance documents the generation of all data artifacts thoroughly, regardless of data's scientific significance. We do not have an indication whether a data item is a significant one such as a result of visualization, or whether it is just a side-product such as a status message, or a temporary file address. While this indiscriminating approach to provenance collection is necessary for scientific audit and review, it becomes overwhelming for the scientist when trying to report results. Provenance Research has offered abstracting workflow provenance:

   1) Through pinpointing of significant activities at the workflow description layer, and then using this information to compact deep lineage traces at the execution layer. The ZOOM approach [42] and an earlier Provenance API in the Taverna tooling[5] follows this principle.

   2) Through gradual exposure of the user to the provenance through an interactive provenance browser [43].

The above outlined characteristics call for 1) Domain-Specific Annotation of data artefacts and 2) Abstraction of provenance. Abstractions should allow the scientists **to more easily differentiate the report-worthy from the detail** and Annotations should allow the scientist **to more effectively select data subsets of interest and will explicitly inform her on the context of data.** Such annotations can help reducing the need to manually sift through result files to recall and compile configuration parameters, or result characteristics that needs to be reported.

4. Existing Remedies for Provenance Abstraction

There are partial remedies to the above shortcomings. These are recent developments in workflow tooling and user-crafted ways of abstracting workflows.

[5] http://dev.mygrid.org.uk/wiki/display/provenance/Provenance+Query+Language

- **Using libraries of components engineered to work together**: Taverna allows for the augmentation of specialist component families onto its default configuration. Components are an abstraction mechanism that help conceal boiler plate activities required for the invocation of specialist resources (e.g. accessing grids), or for handling of domain specific data formats (e.g. VOTable). By hiding away detail, they simplify the workflow and consequently lead to simpler/compact lineage traces.
  Also, components allow for domain specific semantic annotation of input and output ports. So far these annotations have been used for generating valid component compositions, but it also offers the possibility of propagating those port annotations to actual data artifacts generated from runs (previous Taverna provenance research on this is reported here [18]).

- **User Design Practices:** There are two established practices of abstraction:
  - **Grouping operations into sub-workflows:** Sub-workflows are a design construct for modularity, which is also an established best practice in workflow development. The major driver for sub-workflows is to create modules of significant or re-usable function within or across workflows. In practice, however, sub-workflows are also be used for purposes other than just modularity, such as configuring nested iterations or organizational or even aesthetic concerns for large workflows (e.g. a sub-workflow denoting a phase in the experiment)

  - **Self-provenance collection:** Rather than querying workflow provenance traces to locate data subsets of interest, users often encode provenance collection into the workflow design itself. They do this by 1) Promoting intermediary results that are report-worthy to become workflow outputs. See an example of this behavior in Figure 6. This approach has the side-effect of cluttering the workflow design. 2) They embed domain specific metadata into file names or workflow port names or file names, which is useful for when reporting.

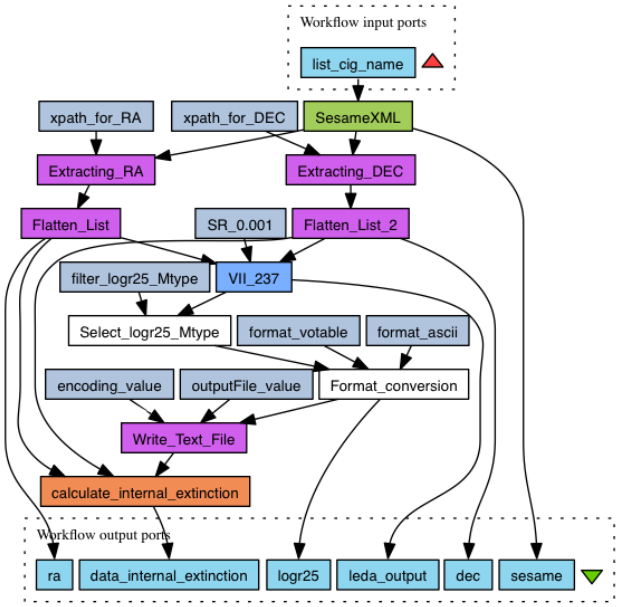

Figure 6 Astronomy Workflow where Intermediary results promoted to become workflow outputs

5. Bridging the Gap with Experiment Reporting Stage

We observe that raw workflow execution provenance and the provenance requirements for data publishing stand at the two ends of a spectrum, which we have depicted as the Provenance Pyramid in Figure 7. The Provenance Pyramid takes inspiration from the Data Pyramid [14], which argues that the amount of data that is of value for preservation is inversely proportional to the number of stakeholders interested in the data. As data moves from a local zone (from the desktop of a single workflow developer) to a community zone (data repositories), only the significant data items are promoted to the next level.

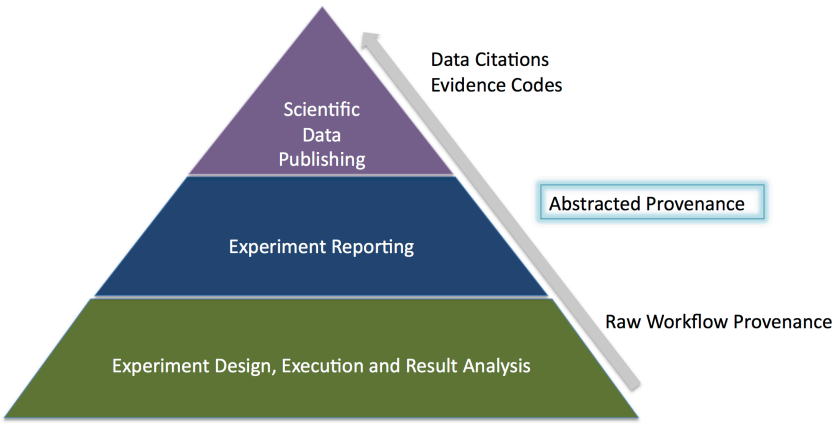

Figure 7 The Pyramid of Provenance Data

Consequently, we place raw workflow provenance at the bottom of the pyramid as it contains indiscriminately collected information about every activity invocation and data item during an execution. This form of provenance is useful for scientists directly involved in the workflow-based experiment for local execution-time activities, such as, design [20], debugging [23], [30] and steering [21]. At the top we have the provenance information that is of community value. These are small nuggets of distilled information currently specified by manual annotation. This community-level provenance is manifested as high-level attributes regarding the derivation method of data, or its origins.

We argue that a middle-layer of Abstracted and Annotated Provenance information is needed. This is to be generated during a transitional **Experiment Reporting** stage, which lies between workflow design/execution phases and result analysis/publishing phases. **We do not claim that provenance abstractions generated at the reporting phase can immediately stand-in as publishable provenance. We claim, however, that these abstractions can help the publishing scientist/curator to organize experimental results and make possible the development of (semi)automated tools for generating community-level provenance indicators (such as data citations).**

6. Research Framework

Currently experiment reporting is predominantly manual, where the scientist selects subsets of experimental data products for publishing, and further annotate them to denote domain-specific provenance and context.

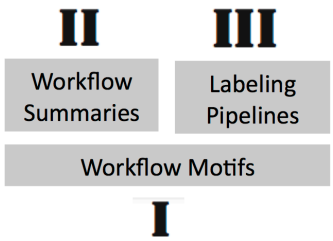

Figure 8 Research Framework

We postulate that with the addition of minimal design time information regarding the characteristic of activities inside workflows, provenance can be exploited semi-automatically 1) to generate configurable filters over experimental data products and 2) to act as a substrate to create and propagate domain-specific provenance. Our research is framed with three investigations (depicted in Figure 8) [3]:

I. **Workflow Motifs** are the minimum information that can be identified by workflow designers to designate the high-level data-processing characteristic of activities inside workflows [7]. The novelty of Motifs lies in their high-level approach to categorizing activities, which is rooted in empirical observation over of a large number of real-world scientific workflows. Motifs can be specified through semantic annotation of workflows, and they provide partial transparency into the inner workings activities Motifs guide the generation of summaries and labeling pipelines.

II. **Workflow Summaries** are configurable filters over workflow provenance [2]. We explore the use and effectiveness of graph-rewriting as a mechanism for reducing workflow design graphs. We combine motifs with workflow re-write primitives in re-write rules, and we explore the possibility of capturing user's reporting preferences as configured reduction rules.

III. **Labeling Pipelines** are processes for the creation and propagation of domain-specific metadata [4]. We observe that workflow descriptions and their execution provenance and data artefacts themselves are readily available sources for obtaining domain-specific metadata annotations. We devise a process model for labeling and outline default labeling behavior for each workflow motif. Given a motif-annotated scientific workflow, and a library of annotation functions, we generate a corresponding labeling pipeline. The novelty lies in our emphasis on domain-specific and dynamic markup. Labeling pipelines can be (re)used to decorate traces form multiple runs of the same workflow with different inputs, all resulting in different annotations.

In the rest of this report we will briefly describe Workflow Motifs and share our experiences in annotating workflows with their motifs.

7. Workflow Motifs

Central to the abstraction of provenance is the notion of motifs, which we outlined in previous work [7] based on an empirical analysis of 260 scientific workflows from 4 systems (including Taverna workflows that do not utilize components) and 10 scientific domains. The motifs are captured in an OWL ontology[6]. The Heliophysics examples presented earlier is highlighted with its motifs in Figure 7.

The motif ontology characterizes activities with respect to their Data-Oriented functional nature and the Resource-Oriented implementation nature. We refer the reader to [7] for details. Here we briefly introduce some of the motifs in light of our running example.

- Data-Oriented nature. Certain activities in workflows, such as **Data Retrieval**, **Analysis** or **Visualizations**, perform the scientific heavy lifting in a workflow. The Heliphysics pipeline in Figure 9 collects data from various external databases through retrieval steps (see "HelioFeatureCatalog", "HelioEventCatalog"). The data retrieval steps are pipelined to each other through use of adapter steps, which are categorized with the **Data Preparation** motif. **Augmentation** is a sub-category of data preparation motif. **Augmentation** decorates data with resource/protocol specific padding or formatting (the sub-workflows "BuildHFCQuery" and "BuildHECQuery" in our example are augmenters that build up a well-formed query request out of given parameters.) **Extraction** steps perform the inverse of **Augmentation**; they extract data from raw results returned from analyses or retrievals (e.g. SOAP XML messages). Another general category is **Data Organization** activities, which perform querying and organization functions over data such as, **Filtering, Joining** and **Grouping**. The "FilterEmptyResults" method in

[6] http://purl.org/net/wf-motifs

our example is an instance of the filtering motif. Another frequent motif is **Data Moving**. Though not exemplified in our workflow, a very common activity occurring in workflows is the movement of data in and out of the workflow environment (e.g. download from URL, write to file etc).

- Implementation-Oriented nature, which outlines the implementation aspects and reflects the characteristics of resources that underpin the operation. Classifications in this category are: **Composite Workflows**, which indicate usage of sub-workflows (as in Figure 9), **Human−Interactions** vs entirely **Computational** steps, **Stateful** vs **Stateless Invocations** and using **Internal** (e.g. local scripts, sub-workflows) vs **External** (e.g. web service, or external command line tools) computational artifacts. For example in Figure 9, all data retrieval operations are realized by external, stateless resources (web services).

During our analysis we were able to categorize more than 90% of all activities in our entire corpus with a Motif using manual inspection. Per workflow, on average, up to 70% of activities are either Data Preparation or Data Organization steps and the remaining 30 percent make up the scientifically significant report-worthy activities.

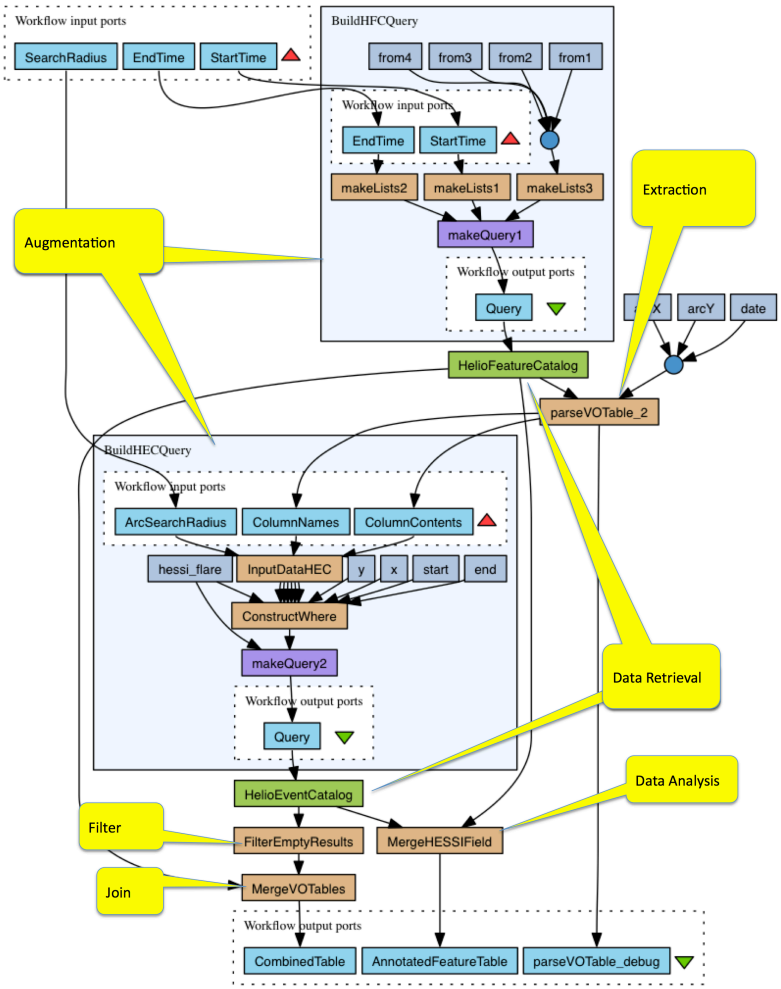

Figure 9 Heliphysics workflow activities annotated with motifs. (For brevity we omitted annotations of activities within sub-workflows.

8. Recommendations for New Provenance Capabilities

In light of the information laid out so far, we compiled an initial set of suggestions s input to future Taverna provenance capabilities:

- Provenance has until recently been a "write once read never" type of artefact. Recent capabilities in Taverna 3 are an important step towards viewing provenance information for audit/review scenarios. As we tried to illustrate, the lack of usage may be due to complexity or genericity of this information, or lack of requirements on the end-project side. Whichever the case, the lifecycle and the layers of provenance can be used to organize any provenance related requirements that might come from specific projects. We shall note that provenance queries we give in Section-3 are the result of a research review, and might not be applicable to all projects.

- The Self-Provenance collection behavior is clearly calling for a **Bookmarking capability**. Bookmarking can be done for activities or for ports, and it should be performed at the workflow description level. These bookmarks could be used to report more compact lineage or be used to reduce the amount of information displayed in the result pane (e.g. bookmarks only).

- In case Taverna tooling intends to support queries like (III-a) and (III-b) it would be necessary to adopt uniform storage schemes for values of certain data and provenance. This would require capability to allow the user to pinpoint at the design-time the data ports, the run-time values of which are handy in querying provenance. During execution Taverna could in turn adopt a homogeneous storage scheme for those designated ports' values This capability essentially gives the user the option to promote data values of certain selected ports (e.g. search radius, start time, end time) to the metadata level to support transcending queries.

- We know that Taverna Components are well-behaved. Documentation on components also mention "agreed provenance" [7], but they do not further elaborate on it. Given the significant need for domain specific information, Components could be the anchor point for generating such domain specific provenance

9. Implementation Experiences with Tooling

We specify motifs with semantic annotations over the workflow. Alongside the motif ontology we adopted an implementation-independent abstract model for representing workflow descriptions, namely the Wfdesc [15] model developed in

[7] http://www.slideshare.net/anpawlik/stfc-workshopworkflows2013

the Wf4Ever project In Figure 8 we provide a fragment that partially depicts the markup of two sample activities in our running example. The fragment specifies that the "HelioEventCatalog" activity has the **Data Retrieval** motif, whereas the "ParseVOTable_2" activity has the **Extraction** motif.

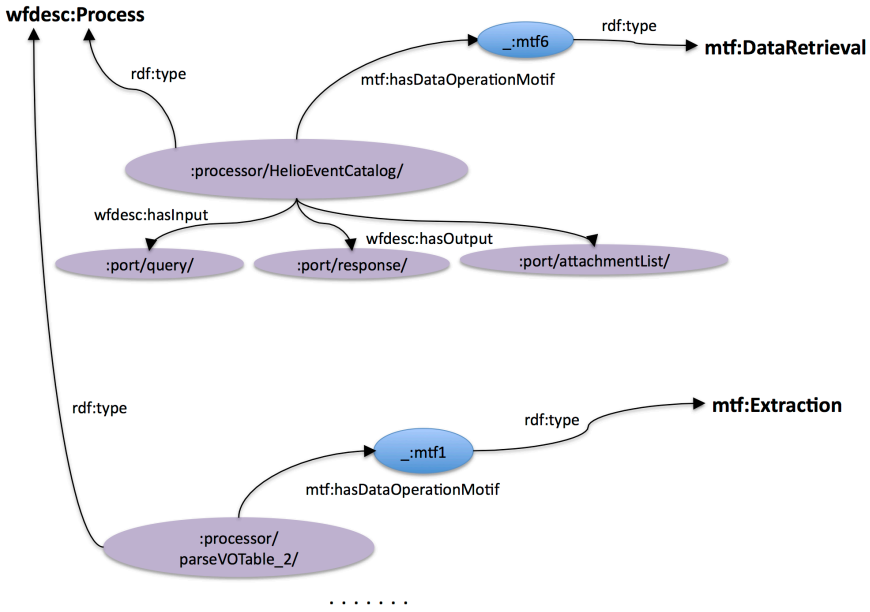

Figure 10 Sample Motif Annotations

Exporting abstract representations of workflows: We consider the workflow description as an important and self-standing part of provenance. For Taverna workflows there already exist capabilities to map the Taverna specific Scufl representation into the abstract Wfdesc representations [15], which we have used extensively for our implementation. One feature we required was integration of this capability to the workbench. While it is possible to export the Wfdesc based workflow representations when bundled with execution provenance, it is not possible to export just the workflow definition (without executing it).

Annotating workflow activities: We struggled with the lack of tooling for annotating Taverna workflows. Taverna workbench provides a general free text annotation capability, and the Component plug-in provides semantic annotations for components. What we needed was a capability in between, some form of structured annotation capability (e.g. key-value) for vanilla workflows (i.e. those which do not use components).

10. Conclusion

This report was intended to familiarize the Taverna development team with demands on provenance and the issues in meeting those demands in light of the experimental lifecycle. Existing coping mechanisms, and proposed research solutions all gravitate toward **Abstraction** and **Annotation** as general solution

methods. We made a brief introduction to our research framework and how we're approaching abstraction and annotation. The details of our techniques and their applicability to the tooling will be provided in the second part of this report.

Finally we tried to give initial hints toward development of new provenance capabilities in the tooling. We also include feedback from our own use Taverna tooling.

11. References

[1] K. Abazajian and J. K. A.-M. et. al. The first data release of the sloan digital sky survey. The Astronomical Journal, 126(4):2081, 2003.

[2] P. Alper, K. Belhajjame, C. Goble, and P. Karagoz. Small is beautiful: Summarizing scientific workflows using semantic annotations. In Proceedings of the IEEE 2nd International Congress on Big Data (BigData 2013), Santa Clara, CA, USA, June 2013.

[3] P. Alper, K. Belhajjame, C. A. Goble, and P. Karagoz. Enhancing and abstracting scientific workflow provenance for data publishing. In Proceedings of the Joint EDBT/ICDT 2013 Workshops, EDBT '13, pages 313–318, New York, NY, USA, 2013. ACM.

[4] P. Alper, C. A. Goble, and K. Belhajjame. 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, WORKS '13, pages 7–16, New York, NY, USA, 2013. ACM.

[5] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche. The open provenance model core specification (v1.1). Future Gener. Comput. Syst., 27(6):743–756, June 2011.

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

[7] D. Garijo, P. Alper, K. Belhajjame, O. Corcho, Y. Gil, and C. Goble. Common motifs in scientific workflows: An empirical analysis. Future Generation Computer Systems, 2013.

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

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

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

[11] R. Bentley, J. M. Brooke, A. Csillaghy, D. Fellows, A. L. Blanc, M. Messerotti, D. Perez-Suarez, G. Pierantoni, and M. Soldati. Helio: Discovery and analysis of data in heliophysics. In eScience, pages 248–255. IEEE Computer Society, 2011.

[12] Yolanda Gil, Simon Miles, Khalid Belhajjame, Henela Deus, Daniel Garijo, Graham Klyne, Paolo Missier, Stian Soiland-Reyes, and Stephan Zednik. A primer for the prov provenance model, 2012. World Wide Web Consortium (W3C).

[13] Pinar Alper, Khalid Belhajjame, Carole A. Goble, and Pinar Senkul. Enhanc- ing and abstracting scientific workflow provenance for data publishing. BIGProv 2013 International Workshop on Managing and Querying Provenance Data at Scale, DEC 2012.

[14] B. Francine. Got Data? A Guide to Data Preservation in the Information Age. Communications of the ACM, 51(12):50–56, 2008.

[15] Workflow-Centric Research Objects: First Class Citizens in Scholarly Discourse Khalid Belhajjame, Oscar Corcho, Daniel Garijo et al. Proceedings of Workshop on the Semantic Publishing, (SePublica 2012) 9 th Extended Semantic Web Conference Hersonissos, Crete, Greece, May 28, 2012. 2012;

[16] Carole Goble, Katy Wolstencroft, Antoon Goderis, Duncan Hull, Jun Zhao, Pinar Alper, Phillip Lord, Chris Wroe, Khalid Belhajjame, Daniele Turi, Robert Stevens, Tom Oinn, and David De Roure. Chapter in Knowledge Discovery for Biology with Taverna: Producing and Consuming Semantics on the Web of Science. 2006.

[17] Bertram Ludaescher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew B. Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific work- flow management and the kepler system. Concurrency and Computation: Practice and Experience, 18(10):1039–1065, 2006.

[18] Paolo Missier, Satya Sanket Sahoo, Jun Zhao, Carole A. Goble, and Amit P. Sheth. Janus: From workflows to semantic provenance and linked open data. In IPAW, pages 129–141, 2010.

[19] YolandaGil, Varun Ratnakar,Jihie Kim,Pedro A.González-Calero,Paul . Groth, Joshua Moody, and Ewa Deelman. Wings: Intelligent workflow-based design of computational experiments. IEEE Intelligent Systems, 26(1):62–72, 2011.

[20] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cludio T. Silva, and Huy T. Vo. Managing the evolution of dataflows with vistrails. In IEEE Workshop on Workflow and Data Flow for Scientific Ap- plications (SciFlow, 2006.

[21] I Altintas, O Barney, and E Jaeger-Frank. Provenance Collection Support in the Kepler Scientific Workflow System. In IPAW, pages 118–132, 2006.

[22] Jihie Kim, Ewa Deelman, Yolanda Gil, Gaurang Mehta, and Varun Ratnakar. Provenance trails in the Wings-Pegasus system. Concurr. Comput. : Pract. Exper., 20(5):587–597, April 2008.

[23] Robert Ikeda, Junsang Cho, Charlie Fang, Semih Salihoglu, Satoshi Torikai, and Jennifer Widom. Provenance-based debugging and drill-down in data-oriented workflows. In ICDE 2012. Stanford InfoLab.

[24] Wolstencroft K, Owen S, du Preez F, Krebs O, Mueller W, Goble CA, Snoep JL (2011) The SEEK: A Platform for Sharing Data and Models in Systems Biology, Methods in Enzymology, Volume 500: 629-655 PUBMED: 21943917

[25] Site for the global biodiversity information facility (gbif). http://www.gbif.org. Accessed: 10.Dec.2012.

[26] Katy Wolstencroft, Stuart Owen, Matthew Horridge, Wolfgang Mueller, Finn Bacall, Jacky L. Snoep, Franco du Preez, Quyen Nguyen, Olga Krebs, and Carole A. Goble. Rightfield: Scientific knowledge acquisition by stealth through ontology-enabled spreadsheets. In EKAW'12, pages 438–441, 2012.

[27] Dataup. describe, manage and share your data. http://dataup.cdlib.org/. Accessed: 10.Dec.2012.

[28] Wolstencroft K.J., Owen S., Krebs O., Mueller W. Nguyen Q., Snoep J.L. & Goble C. (2013), Semantic Data and Models Sharing in Systems Biology: The Just Enough Results Model and the SEEK Platform. In: The Semantic Web-ISWC 2013: Springer-Verlag. 212-227.

[29] Suggested citation: GBIF (2011). GBIF Metadata Profile, Reference Guide, Feb 2011, (contributed by O Tuama, E., Braak, K., Copenhagen: Global Biodiversity Information Facility,19 pp. Accessible at http://links.gbif.org/gbif_metadata_profile_how-to_en_v1

[30] Yael Amsterdamer, Susan B. Davidson, Daniel Deutch, Tova Milo, Julia Stoy- anovich, and Val Tannen. Putting lipstick on pig: Enabling database-style workflow provenance. PVLDB, 5(4):346–357, 2011.

[31] Adriane Chapman and H. V. Jagadish. Understanding provenance black boxes. Distributed and Parallel Databases, 27(2):139–167, January 2010.

[32] Paolo Missier, Khalid Belhajjame, Jun Zhao, Marco Roos, and Carole Goble. Provenance and annotation of data and processes. chapter Data Lineage Model for Taverna Workflows with Lightweight Annotation Requirements, pages 17– 30. Springer-Verlag, Berlin, Heidelberg, 2008.

[33] Herbert Van de Sompel and Carl Lagoze. All aboard: toward a machine-friendly scholarly communication system. In Hey et al. editors, pages 193– 199.

[34] Alberto Pepe, Matthew Mayernik, Christine L. Borgman, and Herbert Van de Sompel. From artifacts to aggregations: Modeling scientific life cycles on the semantic web. J. Am. Soc. Inf. Sci. Technol., 61(3):567–582, March 2010.

[35] S. Grosvenor J. Jones A. Koratkar C. Li J. Mackey K. Neher K. Wolf. Linking Science Analysis with Observation Planning: A Full Circle Data Lifecycle. 2001, NASA Technical Report.

[36] 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.

[37] Daniel Garijo, Yolanda Gil, A new approach for publishing workflows: abstractions, standards, and linked data, in: Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science, ACM, Seattle, 2011, pp. 47–56.

[38] Daniel Garijo, Yolanda Gil, Augmenting prov with plans in p-plan: scientific processes as linked data. in: Second International Workshop on Linked Science: Tackling Big Data (LISC), Held in Conjunction with the International Semantic Web Conference, ISWC, Boston, MA, 2012.

[39] Paolo Missier, Saumen Dey, Khalid Belhajjame, Victor Cuevas, Bertram Ludaescher, D-PROV: extending the PROV provenance model with workflow structure, in: Procs. TAPP'13, Lombard, IL, 2013.

[40] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche. The open provenance model core specification (v1.1). Future Gener. Comput. Syst., 27(6):743–756, June 2011

[41] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. pages 1099–1110, 2008.

[42] Olivier Biton, Sarah Cohen-Boulakia, Susan B. Davidson, and Carmem S. Hara. Querying and Managing Provenance through User Views in Scientific Workflows. 2008 IEEE 24th International Conference on Data Engineering, pages 1072–1081, April 2008.

[43] Manish Kumar Anand, Shawn Bowers, and Bertram Luda¨scher. Provenance browser: Displaying and querying scientific workflow provenance graphs. In ICDE, pages 1201–1204, 2010.