A New Approach for Publishing Workflows: Abstractions, Standards, and Linked Data

Daniel Garijo
OEG-DIA
Facultad de Informática
Universidad Politécnica de Madrid
dgarijo@delicias.dia.fi.upm.es

Yolanda Gil
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
gil@isi.edu

ABSTRACT

In recent years, a variety of systems have been developed that export the workflows used to analyze data and make them part of published articles. We argue that the workflows that are published in current approaches are dependent on the specific codes used for execution, the specific workflow system used, and the specific workflow catalogs where they are published. In this paper, we describe a new approach that addresses these shortcomings and makes workflows more reusable through: 1) the use of *abstract workflows* to complement executable workflows to make them reusable when the execution environment is different, 2) the publication of both abstract and executable workflows using *standards* such as the Open Provenance Model that can be imported by other workflow systems, 3) the publication of workflows as *Linked Data* that results in open web accessible workflow repositories. We illustrate this approach using a complex workflow that we re-created from an influential publication that describes the generation of 'drugomes'.

Categories and Subject Descriptors

C. Computer systems organization, D.2 Software engineering, D.2.10 Design.

General Terms

Documentation, Performance, Design, Standardization.

Keywords

Workflows, provenance, OPM, Wings, reproducibility.

1. INTRODUCTION

Scientific workflows are products of research and should be treated as first-class citizens in cyberinfrastructure [9]. Workflows represent computations carried out to obtain scientific results, but these computations are only described in the narrative of published scientific articles and only at a very high level. Scientific articles describe computational methods informally, often requiring a significant effort from others to reproduce and to reuse. The reproducibility process can be so costly that it has been referred to as "forensic" research [1]. Studies have shown that reproducibility is not achievable from the article itself, even when datasets are published [1], [15]. Retractions of publications do occur, more often than is desirable [26]. A recent editorial proposed tracking the "retraction index" of scientific journals to indicate the proportion of published articles that are later found problematic [7]. Publishers themselves are asking the community to end "black box" science that cannot be easily reproduced [24]. The impact of this issue is well beyond scientific research circles. Clinical trials based on erroneous results pose significant threats to patients [14]. The validity of scientific research methods has been put in question [17]. The public has neutral to low trust on scientists for important topics such as flu pandemics, depression drugs, and autism causes [25].

To facilitate reproducibility, the idea of enhancing scientific publications with explicit workflows has been proposed [5]. Workflows could be incorporated as supplementary material of scientific publications, much like datasets are included today. This would make scientific results more easily reproducible because articles would have not just a textual description of the computational process used but also a workflow that, as a computational artifact, could be inspected and automatically re-executed. Some systems exist that augment publications with scripts or workflows, such as Weaver and GenePattern [6] [18] [19]. Repositories of shared workflows enable scientists to reuse workflows published by others and reproduce their results [27]. Some research addresses the publication of workflows using semantic web technologies [21]. Many workflow researchers have developed and adopted the Open Provenance Model (OPM) as a shared model for workflow publication that is independent of particular workflow systems [23]. OPM is an important step towards reusability in workflow publication.

However none of the approaches to workflow publication to date supports reproducing workflows across different execution infrastructures. Each lab has an execution infrastructure that includes software libraries and codes for computations that are different from other labs. Therefore, even if a workflow is published in OPM and can be run in other workflow systems, it remains tied to the particular executable components used in the original workflow and therefore its reusability is severely limited.

We see a workflow as a "digital instrument" that enables scientists to analyze data through the lens of the method that the workflow represents. A technological challenge is how to make such instruments reusable across labs and institutions, since each has a diverse software and hardware infrastructure. Publishing and sharing workflows that can only be run using a particular software platform is useful, but their reusability is severely limited to labs that have the same software platform. What is needed is a mechanism to publish workflows that would give scientists access to such digital instruments at very low cost, and that would facilitate the reuse of the method, i.e., the workflow, in the desired execution infrastructure which may be different from the original one used by the workflow publishers.

This paper describes a framework to publish computational workflows used in a research article in a manner that is both platform independent and easily reusable in different platforms. Our work has three major contributions:

1. **Publishing an abstract representation of the executed workflow.** This abstract workflow captures a conceptual and execution-independent view of the data analysis method. It makes the workflow more reusable, providing a better understanding of its methods and making every workflow step a separate reusable unit. We use the Wings workflow system [8] [10], which has an expressive language to represent reusable abstract workflow templates using semantic constraints in OWL and RDF.
2. **Publishing both the abstract workflow and the executed workflow in OPM**. Although other systems publish the executed workflow in OPM, our work is novel in that the abstract workflow is published as well. As a result, the abstract method is no longer dependent on the particular execution environment used to run the original workflow. We extended OPM with a profile called OPMW that includes terms appropriate to describe abstract workflows.
3. **Publishing the workflows as web objects.** We used the Linked Data principles [4], [13] to enable direct access to workflows, their components, and the datasets used as web objects with a unique URI and represented in RDF. This would enable other scientists to inspect a workflow without having to ask the investigators for details and without having to reproduce it. We offer an RDF repository of published workflows, as accessible from a SPARQL Endpoint. Other applications can import these workflows. An additional advantage is that the workflows could be linked to life sciences entities that are already published as web resources, including OBO[4], PDB[5], and UniProt[6]. Published workflows can become Research Objects (ROs) [2] or nanopublications [11].

We applied our framework to reproduce the method of an influential publication that describes how to derive the drug-target network of an organism, called its "drugome" [16]. Wings is used to create the workflows (abstract and executable). We extended Wings to publish the workflow in OPM as Linked Data.

The rest of the paper is organized as follows. The next section briefly reviews the drugome workflow that we reproduced. Section 3 introduces our approach, defining abstract workflows and modeling decisions made. Section 4 explains the architecture of the conversion and publication process. Section 5 discusses the advantages of publishing workflows as Linked Data along with query examples. Finally, we present conclusions and future work.

[4] http://www.obofoundry.org/
[5] http://www.pdb.org
[6] http://www.uniprot.org/
[8] http://funsite.sdsc.edu/scb/smap/smap.html

2. INITIAL FOCUS: THE DRUGOME WORKFLOW

Our initial focus is a method to derive the drug-target network of an organism (i.e., its drugome) described in [16]. The article describes a computational pipeline that accesses data from the Protein Data Bank (PDB) and carries out a systematic analysis of the proteome of Mycobacterium tuberculosis (TB) against all FDA-approved drugs. The process uncovers protein receptors in the organism that could be targeted by drugs currently in use for other purposes. The result is a drug-target network (a "drugome") that includes all known approved drugs. Although the article focuses on a particular organism (TB), the authors state that "the methodology may be applied to other pathogens of interest with results improving as more of their structural proteomes are determined through the continued efforts of structural biology/genomics". That is, the expectation is that others should be able to reuse this method to create other drugomes, and to do so periodically as new proteins and drugs are discovered. The original work did not use a workflow system. Instead, the computational steps were run separately and manually.

With the help of the authors of the article, we were able to create the executable workflow that reflects the steps described in the original article. We are able to run it with data used in the original experiments. As is usual in computational biology, the paper has a "methods" section that describes conceptually what computations were carried out, but we needed clarifications from the authors in order to reproduce the computations. Moreover, although the article had just been published we found that some of the software originally used in the experiments was no longer available in the lab, so some of the steps already needed to be done differently.

Figure 1 shows the dataflow diagram of the core steps of the drugome workflow represented in Wings. Datasets are represented as ovals, while computations (codes) are shown as rectangles. The main inputs to the workflow are: 1) a list of binding sites of approved drugs that can be associated with protein crystal structures in PDB, 2) a list of proteins of the TB proteome that have solved structures in PDB, and 3) homology models of annotated comparative protein structure models for TB. First, the SMAP[8] tool is used to compare both the binding sites of protein structures and the homology models against the drug binding sites. The results are sorted and merged. Next, the FATCAT[9] tool is used to compare the overall similarity of the global protein structures, and only significant pairs are retained. A graph of the resulting interaction network is generated, which can be visualized in tools such as Cytoscape[10]. Finally, the Autodock Vina[11] tool is used to perform molecular docking, to predict the affinity of drug molecules with the proteins.

However, in order for this drugome workflow to be widely reusable, we need to be able to publish not just what was executed, but the abstract method in a standard language that many groups can reuse. In addition, having the workflow and supporting data accessible for queries to a public endpoint would allow anyone gain a better understanding of the experiment even without having to reproduce it.

[9] http://fatcat.burnham.org
[10] http://www.cytoscape.org/
[11] http://vina.scripps.edu/

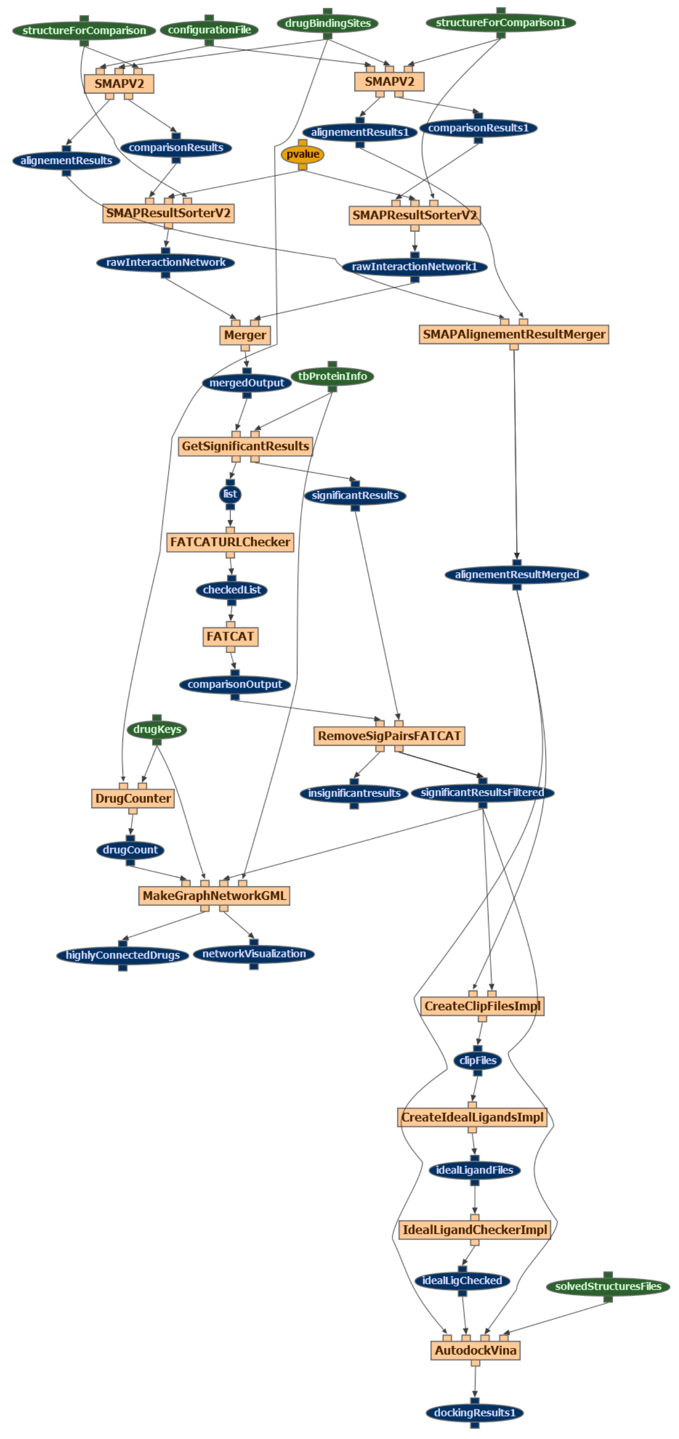


Figure 1: The TB Drugome executable workflow, where the software component is specified for each computational step.

3. APPROACH

This section describes three key features of our approach. First, publishing an abstract workflow provides the means to separate a method from the current implementation as an executable workflow. Second, by transforming both the abstract template and the workflow instance results to OPM we separated the workflow from any workflow system representation. Third, publishing workflows as Linked Data provides the added value of allowing sharing and reusing the templates and results from other workflow systems, as well as being able to link resources from datasets already published as Linked Data.

3.1 Abstract workflows

A key feature of our approach is the creation of an abstract workflow in addition to the executable workflow. This addresses several limitations of executable workflows regarding reusability.

First, the executable workflow runs codes that may not be available to other researchers. In our case, one of the codes was no longer available in the UCSD lab. In the paper, there is a step that obtains docking results from a tool called eHits[13]. However, this tool is proprietary. For our workflow we used alternative tool, Autodock Vina, which obtains docking results too, and it is open source. Another tool, SMAP, had been revised and a new version was available that had a few differences with the one originally used. Note that these changes in the execution environment occurred within the same lab that published the original method just in a few months time. The execution environment in other labs that could reproduce the method would be likely have many more differences. Therefore, publishing the executable workflow has very limited use. This can be addressed if, in addition to publishing the executable workflow that mentions the software that was used, the authors publish an abstract workflow description. In our case, such abstract workflow would include an abstract "docking" step with the same input and output datasets as eHits but that can be easily mapped to Autodock Vina as an alternative tool for that abstract step.

Second, different labs prefer to use different tools for data analysis. In our case, there is a visualization step that can be done using Cytoscape, a known and well-integrated tool, but the lab preferred using yEd[14], which is also very popular. Publishing an abstract workflow that has a more general visualization step and does not mention particular tools facilitates the customization to each lab's software environment.

Third, an investigator may not be familiar with the particular implementations used by others. Many investigators prefer to use Matlab[15] because it is a commercial product that has been debugged and verified and do not want to use R[16] because it has not been as thoroughly tested. Other investigators strongly favor R because of its wide availability to other researchers. So having abstract descriptions of steps that are independent of the implementation makes the workflow more understandable and therefore more reusable.

We define abstract workflows as reusable templates whose steps are abstract classes of components that are implemented by several executable components. These classes describe the inputs and outputs of each step as well as any constraints they have.

[13] http://www.simbiosys.ca/ehits/index.html

[14] http://www.yworks.com/en/products_yed_about.html

[15] http://www.mathworks.com/products/matlab/index.html

[16] http://www.r-project.org/

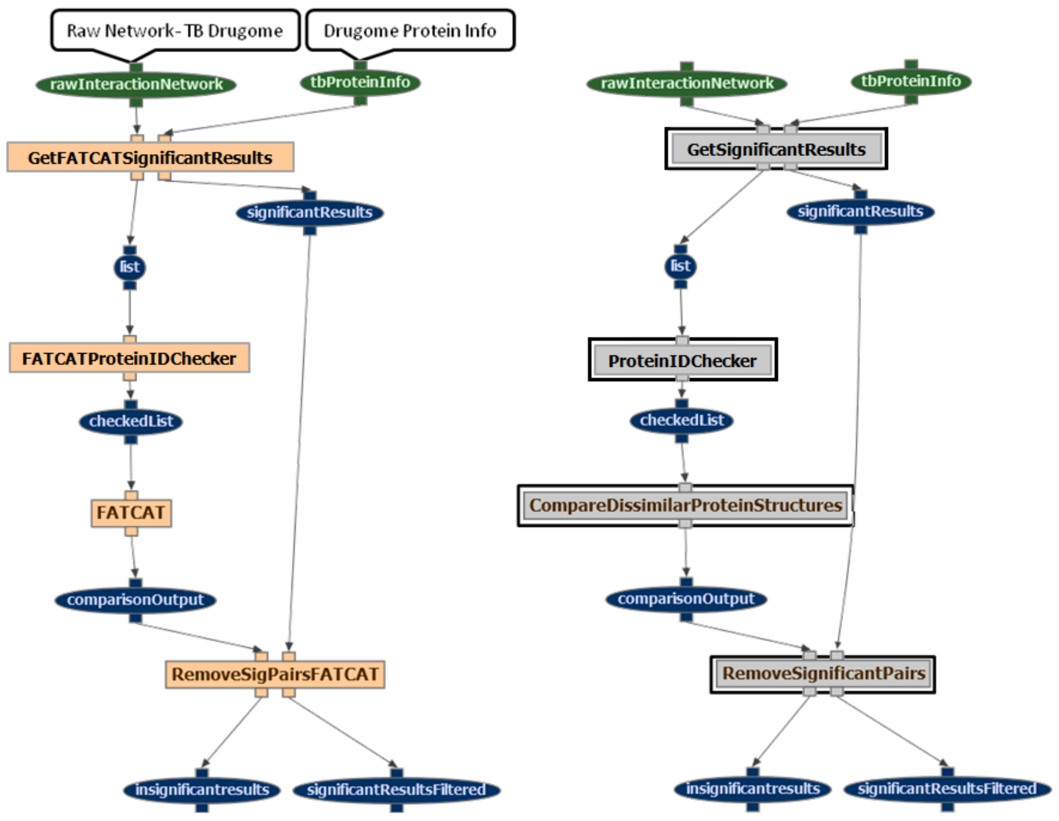


Figure 2: Executable workflow (left) and abstract workflow (right) for the protein structure comparison portion of the drugome workflow.

Thus, the purpose of the abstract workflow is twofold: a) to separate a component description from its actual implementation, making possible to have different instantiations of the same component for the same abstract workflow, b) to make the workflow more human readable by providing a general view of the steps executed in the workflow in a tool-independent way.

Clearly abstract workflows are useful in themselves. However, they should not replace the publication of the executable workflow. The executable workflow provides data products and details of the code invocations that may be useful to other investigators. That is, the abstract workflow complements the executable workflow, but should not replace it.

Wings models the aforementioned separation through three main ontologies: one describes workflows (abstract or executable), another one describes components (abstract or executable), and the last one describes the data catalog, which describes data files.

Figure 2 shows on the left side the dataflow diagram of one of the subworkflows from the drugome workflow. The subworkflow consists on a comparison of dissimilar protein structures, with a formatting step followed by a checking step, and then a filtering step (post processing) after the main comparison.

In Wings, users can define workflow templates that include abstract components as shown in Figure 2 on the right. Wings has a workflow generation algorithm to create valid executable workflows from an abstract workflow template. This algorithm infers which of the tool implementations must be used for each of the abstract steps in the template, and if many are available, asks the user to choose one or else the system chooses one automatically. What we want to do is export the Wings abstract workflow used to create the executable workflow, as well as the executable workflow itself.

The Wings executable workflow on the left of Figure 2 is submitted to an execution engine. Typically the execution engine is Pegasus[19], which makes many transformations to the workflow and executes it in the Condor infrastructure[20]. That is, the Wings executable workflow is a high-level plan for the execution (which, in turn, Pegasus calls an "abstract" workflow because it does not contain execution details), and represents a view of what was executed. This is the execution provenance that we export in this work. The detailed execution provenance records for Pegasus and Condor is exported by Pegasus [20] and is not addressed in this paper.

[19] http://pegasus.isi.edu/

[20] http://www.cs.wisc.edu/condor/

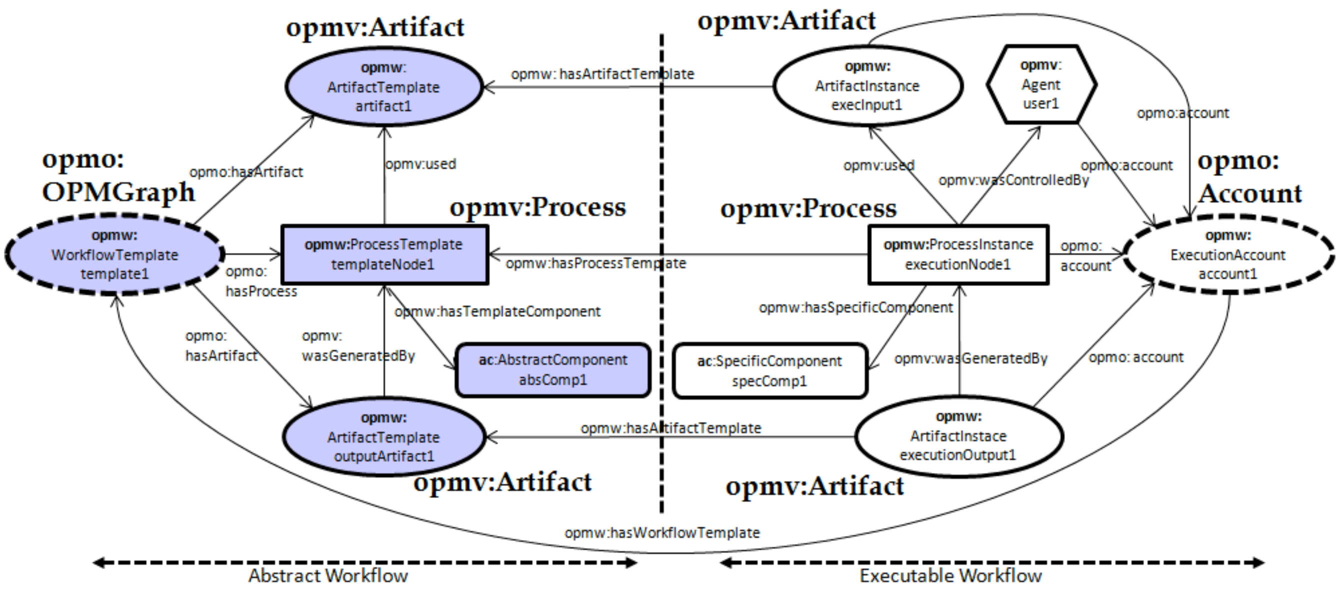

opmv:Artifact
opmw: ArtifactTemplate artifact1
opmw: hasArtifactTemplate
opmv:Artifact
opmw: ArtifactInstance execInput1
opmw: Agent user1
opmo:account
opmo: OPMGraph
opmo:hasArtifact
opmv:used
opmv:used
opmv:wasControlledBy
opmo:account
opmo: Account
opmv:Process
opmv:Process
opmw: WorkflowTemplate template1
opmw:ProcessTemplate templateNode1
opmw:hasProcessTemplate
opmw:ProcessInstance executionNode1
opmo: account
opmw: ExecutionAccount account1
opmo: hasProcess
opmw:hasTemplateComponent
opmw:hasSpecificComponent
opmo: hasArtifact
opmv: wasGeneratedBy
ac:AbstractComponent absComp1
ac:SpecificComponent specComp1
opmv:wasGeneratedBy
opmo: account
opmw: ArtifactTemplate outputArtifact1
opmw:hasArtifactTemplate
opmw: ArtifactInstace executionOutput1
opmv:Artifact
opmv:Artifact
opmw:hasWorkflowTemplate
Abstract Workflow
Executable Workflow

Figure 3: Modeling of an example abstract and executable workflow that has only step (executionNode1), which runs the workflow component (specComp1) that has one input (execInput1) and one output (executionOutput1). References to OPMV and OPMO are represented with their prefixes. The extended classes for our new OPMW profile use the opmw prefix.

3.2 OPMW: Modeling abstract workflows and executions with OPM

To export the abstract workflows and the executable workflows we use OPM, a widely-used domain-independent provenance model result of the Provenance Challenge Series[21] and years of workflow provenance exchange and standardization in the scientific workflow community.

There are several reasons to use OPM. First, OPM has been already used successfully in many scientific workflow systems, thus making our published workflows more reusable [22]. Another advantage is that the core definitions in OPM are domain independent and extensible to accommodate other purposes, in our case workflow representations. In addition, OPM can be considered the basis of the emerging W3C Provenance Interchange Language (PROV), which is currently being developed by the W3C Provenance Working Group[23] as a standard for representing and publishing provenance on the Web.

OPM offers several core concepts and relationships to represent provenance. OPM models the resources (datasets) as *artifacts* (immutable pieces of state), *processes* (action or series of actions performed on artifacts), and *agents* (controllers of processes). Their relationships are modeled in a provenance graph with five causal edges: *used* (a process used some artifact), *wasControlledBy* (an agent controlled some process), *wasGeneratedBy* (a process generated an artifact), *wasDerivedFrom* (an artifact was derived from another artifact) and *wasTriggeredBy* (a process was triggered by another process). It also introduces the concept of *roles* to assign the type of activity that artifacts, processes or agents played when interacting with each other, and the notion of *accounts* and *provenance graphs* to group sets of OPM assertions into different subgraphs. An account represents a particular view on the provenance of an artifact based on what was executed.

We mapped Wings ontologies to the OPM core model, extending OPM core concepts and relationships according to our needs in a new profile called OPMW.

We use two OPM ontologies for our mapping. OPMV[24] is a lightweight RDF vocabulary implementation of the OPM model that only has a subset of the concepts in OPM but it facilitates modeling and query formulation. OPMO[25] covers the full functionality of the OPM model, and we use it for mapping to OPM concepts that are not in OPMV, such as Account or OPM Graph.

Figure 3 shows a high level diagram of the mappings to OPM of an abstract workflow on the left and a specific execution on the right. The workflow shown here has one step (executionNode1), which runs the workflow component (specComp1) that has one input (execInput1) and one output (executionOutput1). For some of the concepts there is a straightforward mapping: datasets (ovals represented in Figure 2) are a subtype of Artifacts, while workflow steps (rectangles in Figure 2), also called nodes, map to OPM Processes. Notice that each node has a link to the component that is run in that step, for example the workflow in Figure 1 has two nodes that run the same component SMAPV2. There is no OPM term that can be mapped to components, so we used our own terms (represented with the ac prefix in the Figure 3).

[21] http://twiki.ipaw.info/bin/view/Challenge/OPM
[23] http://www.w3.org/2011/prov/wiki/

[24] http://open-biomed.sourceforge.net/opmv/ns.html
[25] http://openprovenance.org/model/opmo

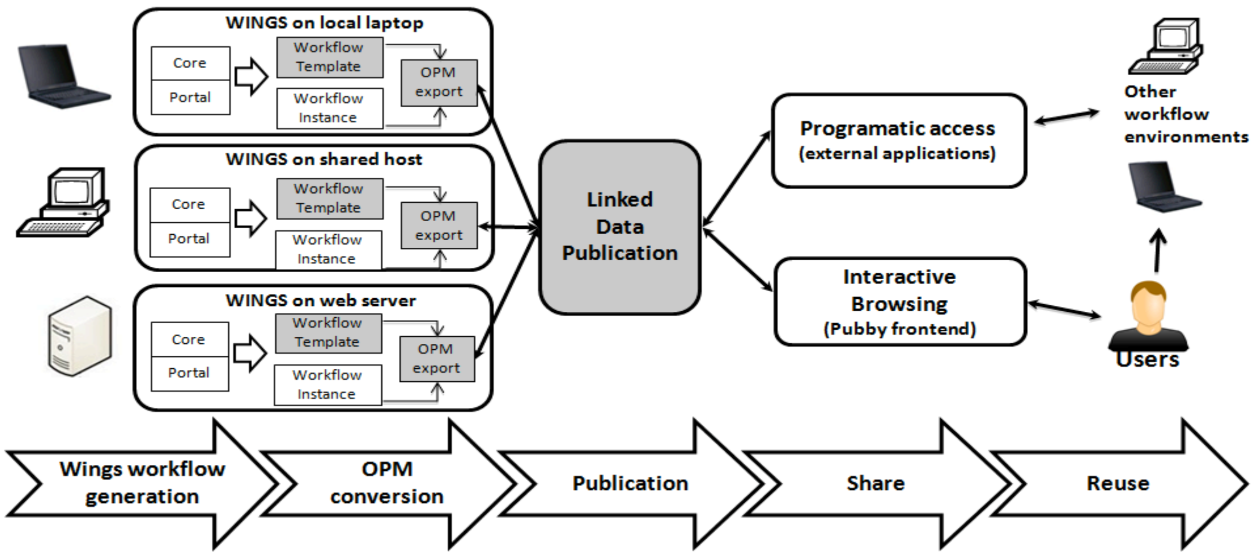


Figure 4: Architecture overview and conversion process

In the figure, the terms taken from OPMO and OPMV are indicated using their namespaces. The new terms that we defined in our extension profile use the OPMW prefix.

In Wings, abstract workflows are considered a special case of workflow templates where all the steps are abstract classes of components. That is, a Wings workflow template can include a mixture of abstract and specific steps. This is why the concepts in OPMW refer to templates instead of abstract workflows, so that we can use OPMW to export any type of workflow.

The executable workflows and the abstract workflows are mapped to different concepts in OPM. Executed workflows are mapped to OPM Accounts, reflecting the fact that they capture a Wings view on the execution (recall that Pegasus and Condor each produce their own views on the execution, each at different levels of detail). Workflow templates are not considered accounts since they can be defined in absence of any execution. Therefore, we represent them as a subclass of OPMGraphs.

Since Wings does not capture the exact time execution of each of the nodes (Pegasus and Condor do) but only captures the starting and ending time of the whole execution, we have linked this information to the execution account along with additional metadata like if it has been a successful execution.

To make the distinction between datasets used in the abstract workflows and the workflow executions explicit, we have extended OPM Artifacts with ArtifactTemplates (the general artifacts used in the abstract workflow) and ArtifactInstances (which are bound to an ArtifactTemplate). Likewise, we defined two subclasses of OPM Process as ProcessTemplates (the abstract steps used in the abstract workflow) and ProcessInstances (the steps in the executable workflow).

The template process in Figure 3 (templateNode1) uses one input artifact (artifact1), has one abstract component (absComp1) and generates an output artifact (outputArtifact1). All template artifacts and processes are linked to a WorkflowTemplate through hasArtifactTemplate and hasArtifactProcess respectively. On the right side of the figure we can see how the processInstance (executionNode1), controlled by a user (user1) which is of type Agent, used a particular input bound to its corresponding ArtifactTemplate and generates the ouputArtifact. All artifacts, processes and agents are linked to the execution account, which has as template workflow the one in the left side of the figure. Each instance is also bound to its template by the explicit relationship hasWorkflowTemplate.

Both the executable workflow and the abstract workflow shown in Figure 2 are published. Other workflow systems (e.g., [21], [22]), only have the former available for publication.

3.3 Exporting workflows as Linked Data

Publishing the OPM abstract and execution workflows is a very important step for reproducibility and reuse. In order to be able to reference all the resources properly, we have decided to follow the Linked Data principles. According to them, we should *use URIs as names for things*, (fully compatible with the expression of OPM in RDF), *use HTTP URIs so that people can look up those names* (making those URIs dereferenceable and available in any browser), *provide useful information when someone looks up a URI* (by showing the resources that are related to the URI) and *include links to other URIs,* so they can discover more things.

There are several important advantages of publishing workflows as Linked Data: a) link to web resources available, for instance refer to proteins in the Protein Data Bank by using their published URI; b) get linked from other applications by pointing to the URIs that we publish, which include both the workflows and the data generated by them; and c) produce interoperable results within different systems without having to define particular catalog structures and access interfaces.

The Wings workflows published by any user as Linked Data become publicly accessible. In some domains privacy is a concern (e.g., if the workflow processes genomic data), in those cases the publication as Linked Data would not be appropriate. However, there are many areas of science where privacy is not an issue and that would benefit tremendously of a more open architecture for sharing both data and workflows as Linked Data.

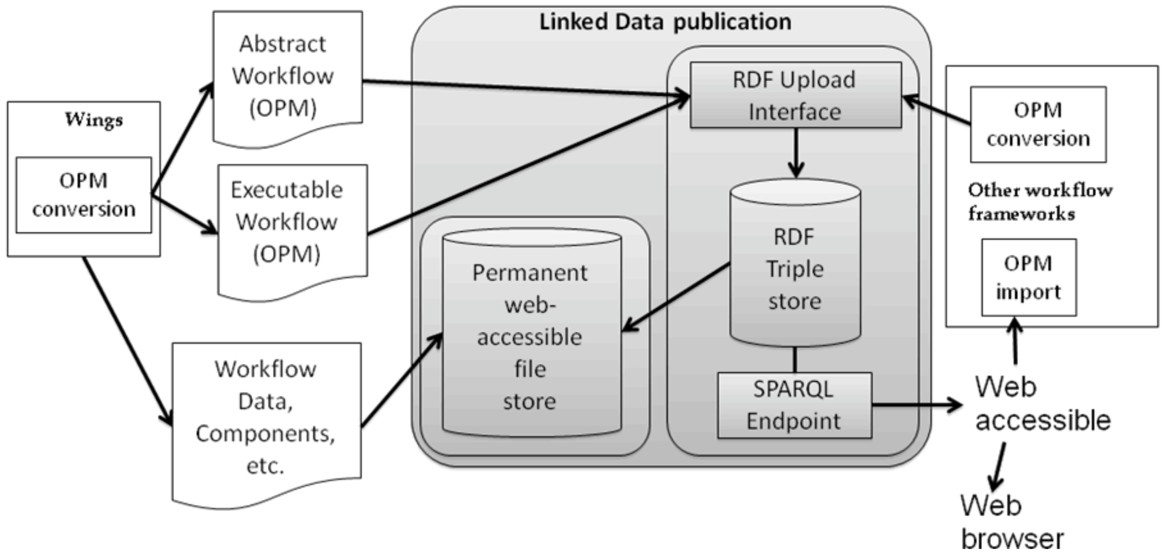


Figure 5: Linked Data publication architecture

All the URIs published have been designed as "Cool URIs"[26], which mean that we have produced them under a controlled domain, they are unique, and we can be sure that they are not going to be changed. Each of these URIs identifies a different resource in our system that can be individually accessed.

Other work [21] links inputs of the workflows to other datasets in Linked Data.

4. ARCHITECTURE

Figure 4 shows an overview of the architecture. Different users produce their own abstract workflows and execution results, either in their own local installation or in central installations of Wings accessible as web portals. These workflows are RDF files, and are converted through a new Wings module to OPM RDF files. The OPM files are then published as Linked Data (center square of Figure 4). Once the files are published on the cloud, they are ready to be queried through external applications or visualized with Linked Data browsers. Users can import workflows to their own installations of Wings, where they can run or change the workflows. Workflows can also be imported to other systems if they are OPM compatible.

Figure 5 gives more details about Linked Data publication. The RDF files are loaded into a Triple Store through its interface, and made public through a public endpoint. We have selected Allegro[28] as our triple store and Pubby[29] for browsing and visualizing the RDF. An additional file store is needed to store the files referred to in the links available in the triple store. The file store is in our local servers (http://wings.isi.edu). The endpoint can be browsed through generic visualizing tools like Pubby, but it can also be accessed programmatically from other applications. For example, other workflow systems could access the workflows and import them into their framework. The access point for a workflow is simply a URI (of a workflow template or an execution), and all the components and datasets in the workflow can be accessed from it. Additionally, other workflows systems could publish their own workflows on the public endpoint too. For doing so, the only requirement is to support the OPM export in a compatible way and make a secure connection to the triple store.

5. ACCESSING WORKFLOWS AS LINKED DATA

The workflow repository is open and accessible over the web[30]. The repository will grow as users publish more workflows using our framework.

We show the broad accessibility of the published workflows by illustrating the queries that we can issue to the repository. Recall that both abstract workflows and workflow executions coexist in the same repository. Thus, we can query either of them or a mixture of both representations. The latter is very useful, as it enables cross-indexing of methods (the abstract workflows) and runs (executable workflows).

We illustrate this with three queries to exemplify how to extract different kinds of information from the repository. To make the text readable, we have included the following prefix declarations:

```
@prefix exec: <http://wings.isi.edu/opmexport/resource/
    ArtifactInstance/> .
@prefix abst: <http://wings.isi.edu/opmexport/resource/
    WorkflowTemplate/> .
@prefix opmw: <http://wings.isi.edu/ontology/opmv/> .
@prefix opmv: <http://purl.org/net/opmv/ns#>.
```

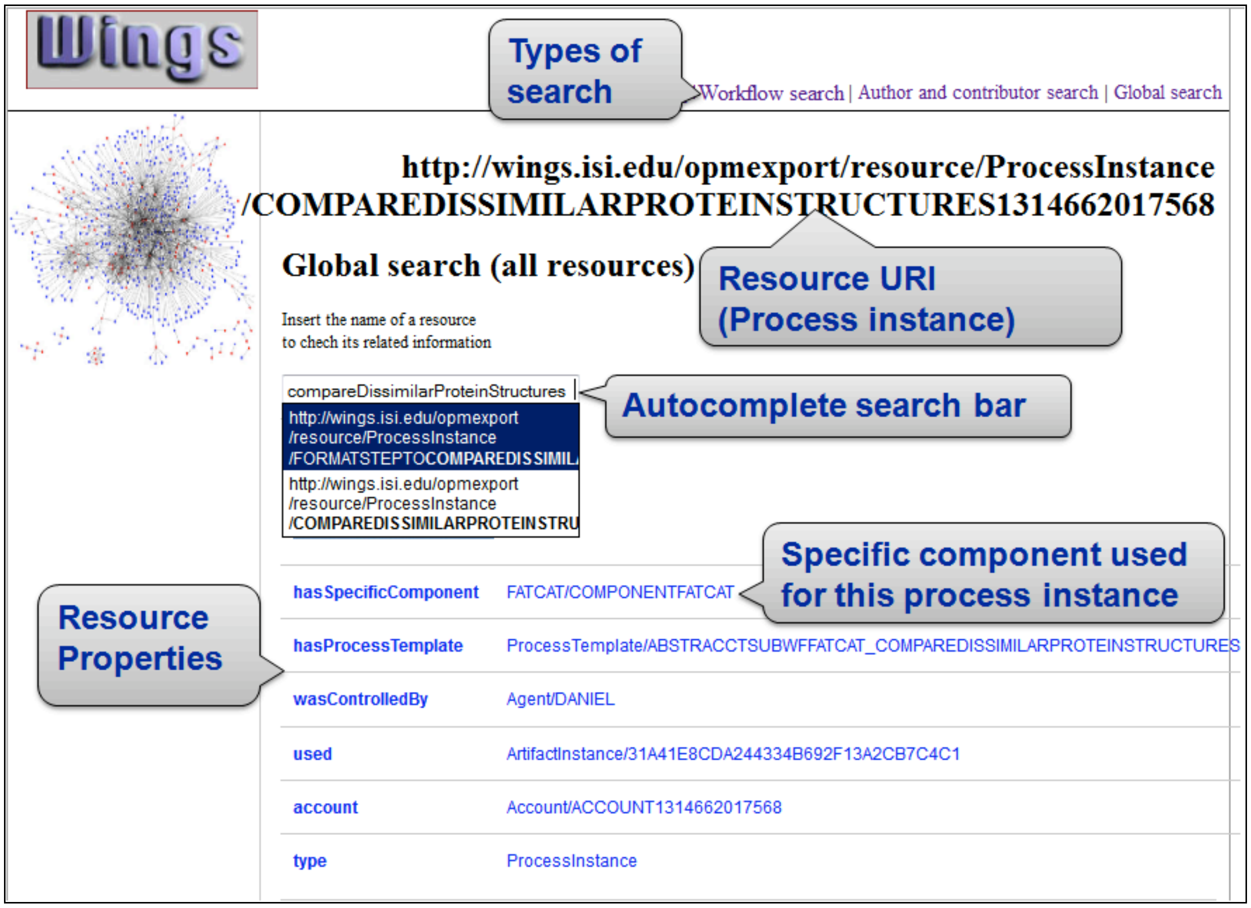
The first example query is designed to retrieve the executable workflow step that generated a given artifact and the corresponding abstract workflow step. The query starts with the name of an artifact (artifactName) and finds its type (?type), its

[26] http://www.w3.org/Provider/Style/URI.html.en

[28] http://www.franz.com/agraph/allegrograph/

[29] http://www4.wiwiss.fu-berlin.de/pubby/

[30] http://wind.isi.edu:10035/catalogs/java-catalog/repositories/WINGSTemplatesAndResults


Wings
Types of search
Workflow search | Author and contributor search | Global search
Resource URI (Process instance)
Autocomplete search bar
Specific component used for this process instance
Resource Properties

http://wings.isi.edu/opmexport/resource/ProcessInstance/COMPAREDISSIMILARPROTEINSTRUCTURES1314662017568

Global search (all resources)

Insert the name of a resource
to chech its related information

compareDissimilarProteinStructures

http://wings.isi.edu/opmexport/resource/ProcessInstance/FORMATSTEPTOCOMPAREDISSIMIL

http://wings.isi.edu/opmexport/resource/ProcessInstance/COMPAREDISSIMILARPROTEINSTRU

hasSpecificComponent	FATCAT/COMPONENTFATCAT
hasProcessTemplate	ProcessTemplate/ABSTRACCTSUBWFFATCAT_COMPAREDISSIMILARPROTEINSTRUCTURES
wasControlledBy	Agent/DANIEL
used	ArtifactInstance/31A41E8CDA244334B692F13A2CB7C4C1
account	Account/ACCOUNT1314662017568
type	ProcessInstance

Figure 6. GUI snapshot of the Linked Data application. The information shown refers to part of the workflow instance in Figure 2.

artifact template (?aTempl), the process which generated that artifact (?process) and then its process template (?templP). The query is:

```
SELECT DISTINCT ?process ?type ?aTempl ?templP
WHERE {
<exec:artifactName> a ?type .
<exec:artifactName> <opmw:hasArtifactTemplate>
      ?aTempl .
< exec:artifactName> <opmv:wasGeneratedBy> ?process .
?process <opmw:hasProcessTemplate> ?templP.
}
```

The second example query retrieves all workflows that have used a given dataset. The query also starts with an artifact (artifactName), but we just ask for the accounts that used such artifact, along with their corresponding workflowTemplate (which is the abstract workflow). The query is:

```
SELECT ?account ? templ
WHERE {
<exec:artifactName> <opmo:account> ?account.
?account <opmw:hasWorkflowTemplate> ?templ
}
```

For the last query example we change the perspective of the query to the abstract workflow, and we ask how many executions were run of a given abstract workflow. For each execution we also query the start time (?startT), end time (?endT) and the status (?stat), which specifies whether the execution failed. The query is:

```
SELECT ?acc ?startT ?endT ?stat
WHERE {
?acc <opmw: hasWorkflowTemplate>
      <abst:templateName>.
?acc <opmw:hasStartTime> ?startT.
?acc <opmw:hasEndTime> ?endT.
?acc <opmw:hasStatus> ?stat.
}
```

As we have demonstrated with these queries, the workflows can be accessed with basic knowledge of the OPM ontologies. However, complex queries would require understanding of OPMW. For this reason, and since navigating through the RDF with Linked Data browsers (such as Pubby) might be tedious, we have designed a small Linked Data application[31] for helping users to browse, search and retrieve the data available in the repository.

Figure 6 shows an overview of this application, retrieving one of the steps of the executable workflow of Figure 2. On top of the figure, the user can select what kind of search he is aiming for (workflow search, author search or resource search). The user can enter the word terms of the search, which will be auto completed immediately suggesting any available resources. By selecting one of the resources, all its relations will be displayed on the same page. In the case of authors, all their contributions will be

[31] http://wind.isi.edu/DemoWFLinkedData/wf.html

displayed, allowing users browsing their published workflows in detail. This kind of application can be easily built to allow end users to access any published Linked Data, making all the Linked Data management transparent to them. We are currently developing an application customized for browsing workflow results.

6. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach to publishing scientific workflows that makes the methods of a scientific article more explicit and reusable than previous approaches. The key contributions of our work are: 1) the publication of an abstract workflow that represents the computational method in an execution-independent manner, 2) the publication of the abstract workflow and the executed workflow using the OPM standard that is independent of the execution environment used, and 3) the publication of the workflows, components, and datasets as Linked Data on the web. Our initial work is focused on publishing the drugome workflow, which represents a recently proposed approach to drug discovery that is both comprehensive and systematic.

In future work, we plan to develop web applications that will illustrate the utility of the repository. We plan to develop an application that enables users to browse the contents of the repository with workflow visualizations, such as dataflow graphs and constraints grouped around datasets, and with data visualizations, showing all data that are related across workflow runs or within workflow types. Other applications that could be developed include applications to import the contents of the workflow repository into other workflow systems, as well as into other Wings installations in different labs with different execution infrastructure.

A limitation of our abstract workflows is that they include data conversion steps that are not appropriate in a high-level conceptual description. In other work, we have extended Wings to reason about incomplete workflows and add steps where data conversions are needed [10]. We plan to extend our work to make the abstract workflows correspond more closely to how methods are described in an article, so we can describe a computational experiment at a conceptual level that makes it even more understandable, more reproducible, and more reusable. Finally, we are studying how to merge the abstract workflows with other publication approaches [2], [11] to avoid workflow execution decay and increase interoperability.

7. ACKNOWLEDGMENTS

This work was done while the first author was at the Information Sciences Institute. We gratefully acknowledge the support of Elsevier Labs in sponsoring this project. This research was supported in part by the US National Science Foundation (NSF) with grant number CCF-0725332, the Air Force Office of Scientific Research (AFOSR) with grant number FA9550-11-1-0104, the Wf4Ever European project (FP7-270192) and MyBigData project from the Spanish Science and Innovation Ministry (TIN2010-17060). We would also like to thank Phil Bourne, Li Xie, Sarah Kinnings, and Lei Xie for their help in understanding and reproducing the drugome workflow.

8. REFERENCES

[1] Baggerly, K. A. and Coombes, K. R. "Deriving Chemosensitivity from Cell Lines: Forensic Bioinformatics and Reproducible Research in High-Throughput Biology." Annals of Applied Statistics, 3(4), 2009. Available from http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.aoas/1267453942

[2] Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D., Owen, S., Newman, D., Sufi, S., Goble, C. "Why Linked Data is not enough for scientists". Future Generation Computer Systems (2011). Available from: http://www.sciencedirect.com/science/article/pii/S0167739X11001439.

[3] Bell A.W., Deutsch E.W., Au CE., Kearney R.E., Beavis R., Sechi S., Nilsson T., Bergeron J.J., and the Human Proteome Organization (HUPO) Test Sample Working Group. "A HUPO test sample study reveals common problems in mass spectrometry–based proteomics." Nature Methods, 6(6), 2009. Available from http://www.nature.com/nmeth/journal/v6/n6/full/nmeth.1333.html

[4] Bizer, C., Heath, T. and Berners-Lee, T. "Linked Data - The Story So Far". International Journal on Semantic Web and Information Systems (IJSWIS) (2009).

[5] Bourne, P. "What Do I Want from the Publisher of the Future?" PLoS Computational Biology, 2010. Available from http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1000787

[6] Falcon, S. "Caching code chunks in dynamic documents: The weaver package." Computational Statistics, (24)2, 2007. Available from http://www.springerlink.com/content/55411257n1473414/

[7] Fang, C.F., and Casadevall, A. "Retracted Science and the retracted index". Infection and Immunity. 2011. doi:10.1128/IAI.05661-11

[8] Gil, Y., Groth, P., Ratnakar, V., and C. Fritz. *Expressive Reusable Workflow Templates*, Proceedings of the IEEE e-Science Conference, Oxford, UK, pages 244–351. 2009.

[9] Gil, Y.; Deelman, E.; Ellisman, M. H.; Fahringer, T.; Fox, G.; Gannon, D.; Goble, C. A.; Livny, M.; Moreau, L.; and Myers, J. "Examining the Challenges of Scientific Workflows." IEEE Computer, 40(12), 2007. Preprint available from http://www.bibbase.org/cache/www.isi.edu__7Egil_publications.bib/computer-NSFworkflows07.html

[10] Gil, Y.; Gonzalez-Calero, P. A.; Kim, J.; Moody, J.; and Ratnakar, V. "A Semantic Framework for Automatic Generation of Computational Workflows Using Distributed Data and Component Catalogs." To appear in the Journal of Experimental and Theoretical Artificial Intelligence, 2011. Preprint available from http://www.bibbase.org/cache/www.isi.edu__7Egil_publications.bib/gil-etal-jetai10.html

[11] Groth, P., Gibson, A., and Velterop, J. "The Anatomy of a Nanopublication." Information Services and Use, 30(1-2), 2010. Available from http://iospress.metapress.com/content/ftkh21q50t521wm2/

[12] Hauder, M., Gil, Y. and Liu, Y. "A Framework for Efficient Text Analytics through Automatic Configuration and Customization of Scientific Workflows". Proceedings of the

Seventh IEEE International Conference on e-Science, Stockholm, Sweden, December 5-8, 2011.

[13] Heath, T. and Bizer, C. “Linked Data: Evolving the Web into a Global Data Space” (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool. (2011)

[14] Hutson, S. “Data Handling Errors Spur Debate Over Clinical Trial,” Nature Medicine, 16(6), 2010. Available from http://www.nature.com/nm/journal/v16/n6/full/nm0610-618a.html

[15] Ioannidis J.P., Allison D.B., Ball C.A., Coulibaly I, Cui X., Culhane A.C., Falchi M, Furlanello C., Game L., Jurman G., Mangion J., Mehta T., Nitzberg M., Page G.P., Petretto E., van Noort V. ”Repeatability of Published Microarray Gene Expression Analyses.” Nature Genetics, 41(2), 2009. Available from http://www.nature.com/ng/journal/v41/n2/full/ng.295.html

[16] Kinnings, S. L.; Xie, L.; Fung, K. H.; Jackson, R. M.; Xie, L.; and Bourne, P. E. “The Mycobacterium tuberculosis Drugome and Its Polypharmacological Implications.” To appear in PLoS Computational Biology, 2011. Preprint available from http://sites.google.com/site/beyondthepdf/file-cabinet/FinalPaper.pdf?attredirects=0&d=1

[17] Lehrer, J. “The Truth Wears Off: Is There Something Wrong with the Scientific Method?” The New Yorker, December 13, 2010. Available from http://www.newyorker.com/reporting/2010/12/13/101213fa_fact_lehrer

[18] Leisch, F. “Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis”, Proceedings of Computational Statistics, 2002. Preprint available from http://www.statistik.lmu.de/~leisch/Sweave/Sweave-compstat2002.pdf

[19] Mesirov, J. P. “Accessible Reproducible Research.” Science, 327:415, 2010. Available from http://www.sciencemag.org/cgi/rapidpdf/327/5964/415?ijkey=WzYHd6g6IBNeQ&keytype=ref&siteid=sci

[20] Miles, S., Deelman, E., Groth, P., Vahi, K. Mehta, G., Moreau, L."Connecting Scientific Data to Scientific Experiments with Provenance" Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007) 10-13 December 2007 in Bangalore, India.

[21] Missier, P., Sahoo, S. S., Zhao, J., Goble, C., and Sheth, A. (2010). Janus: from Workflows to Semantic Provenance and Linked Open Data. Provenance and Annotation of Data and Processes Third International Provenance and Annotation Workshop IPAW 2010 Troy NY USA June 1516 2010 Revised Selected Papers *6378*, 129-141. Available at: http://www.mygrid.org.uk/files/presentations/SP-IPAW10.pdf.

[22] Moreau, L. and B. Ludaescher, editors. *Special Issue on the First Provenance Challenge*, volume 20. Wiley, April 2007.

[23] Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., and denBussche, J. V. “The Open Provenance Model Core Specification (v1.1).” To appear in Future Generation Computer Systems, 2011. Preprint available from http://www.bibbase.org/cache/www.isi.edu__7Egil_publications.bib/moreau-etal-fgcs11.html

[24] Nature Editorial. “Illuminating the Black Box.” Nature, 442(7098), 2006. Available from http://www.nature.com/nature/journal/v442/n7098/full/442001a.html

[25] Scientific American. “In Science We Trust: Poll Results on How you Feel about Science” Scientific American, October 2010. Available from http://www.scientificamerican.com/article.cfm?id=in-science-we-trust-poll

[26] The Scientist. “Top Retractions of 2010.” The Scientist, December 16, 2010. Available from http://www.the-scientist.com/news/display/57864/

[27] De Roure, D; Goble, C.;Stevens, R. “The design and realizations of the myExperiment Virtual Research Environment for social sharing of workflows”. Future Generation Computer Systems, 25 (561-567), 2009