Abstract, link, publish, exploit: An end to end framework for workflow sharing

Daniel Garijo [a,*], Yolanda Gil [a], Oscar Corcho [b]

[a] *Information Sciences Institute and Department of Computer Science, University of Southern California, United States*
[b] *Facultad de Informática, Universidad Politécnica de Madrid, Spain*

HIGHLIGHTS

- Approach for publishing scientific workflows as Linked Data using standards.
- A suite of tools for consuming, browsing, updating and mining workflow data.
- Definition of scenarios for workflow exploitation based on the state of the art.
- Conformance to requirements and coverage of the suite of presented tools.

ARTICLE INFO

ABSTRACT

Scientific workflows are increasingly used to manage and share scientific computations and methods to analyze data. A variety of systems have been developed that store the workflows executed and make them part of public repositories However, workflows are published in the idiosyncratic format of the workflow system used for the creation and execution of the workflows. Browsing, linking and using the stored workflows and their results often becomes a challenge for scientists who may only be familiar with one system. In this paper we present an approach for addressing this issue by publishing and exploiting workflows as data on the Web with a representation that is independent from the workflow system used to create them. In order to achieve our goal, we follow the Linked Data Principles to publish workflow inputs, intermediate results, outputs and codes; and we reuse and extend well established standards like W3C PROV. We illustrate our approach by publishing workflows and consuming them with different tools designed to address common scenarios for workflow exploitation.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Scientific workflows define the "set of tasks needed to manage a computational science process" [1]. They have been used successfully in several domains [2–4] in order to represent, execute, re-run, debug, document and reproduce scientific methods. Scientific workflows are important products of scientific research which should be treated as first-class citizens in cyber-infrastructure [5].

There has been great interest in the publication of workflows, particularly to enable reproducibility and shareability of scientific experiments. There are a number of frameworks that allow the inclusion of workflows and codes in scientific articles [6–8]. Work-

flow repositories like myExperiment [9] and CrowdLabs [10] provide mechanisms to publish and search workflows. These repositories support the publication of workflows in their original language specifications. Given the proliferation of workflow systems both domain-independent [11–13,4,14–16] and domain specific [17,3, 18], this is an impediment for reuse.

An important research challenge is to represent workflows and their associated resources (i.e., inputs, outputs, codes and intermediate results) to facilitate their exploration and adoption by others. Here we tackle this challenge by addressing two main problems: (1) publishing and sharing workflows in a way that can be used by both humans and machines at a low cost; and (2) finding ways to exploit the published data to facilitate reuse and understandability of workflows by other domain scientists.

In order to achieve our goals, this paper describes a framework to publish and exploit computational workflows in a manner that is platform independent. The framework also provides the means to help users understand the published workflows and

* Corresponding author.
E-mail addresses: dgarijo@isi.edu (D. Garijo), gil@isi.edu (Y. Gil),
ocorcho@fi.upm.es (O. Corcho).

their associated resources. Our approach makes the following contributions:

- A **collection of requirements and use case scenarios** for workflow reuse and exploitation, based on the state of the art, that we use to assess the features and limitations of our approach.
- A **methodology for publishing workflows as open data on the Web**, in a representation that is independent of the platform used to create them. The methodology publishes workflows according to existing standards and makes them and their associated resources (inputs, intermediate resources, outputs, configurations and codes) available as web objects using the Linked Data principles [19].
- A suite of **tools for exploiting and consuming workflow data**, helping end users to overcome the workflow language barrier and address workflow understanding. The suite includes tools for workflow template and execution trace visualization, workflow results browsing, automated documentation of workflows and workflow mining for reuse and revision.

We also present an implementation of this framework for publishing workflows, based on standards like OWL, RDF and PROV [20], and illustrate its use for several workflow domains, ranging from text analytics to ecology. For demonstration purposes, we use the WINGS workflow system[1] [16], which has an expressive language to represent reusable abstract workflow templates using semantic constraints in OWL and RDF. Our approach can be applied to other workflow systems, and we illustrate this by publishing workflows from the LONI Pipeline [3].

The rest of the paper is organized as follows. Section 2 introduces prior research on workflow publication and exploitation. Section 3 describes the main scenarios we are interested in addressing, as well as the requirements derived from the scenarios and the state of the art. Section 4 introduces the framework for workflow publication, based on an existing methodology for publishing data on the Web. Section 5 shows how we exploit the data published to facilitate workflow understanding and reuse, while Section 6 discusses the conformance of our framework against the requirements defined on Section 3. Finally, we present our conclusions and future lines of work.

2. Related work

Prior related work can be grouped under three major topics: workflow publication, workflow provenance capture and workflow exploitation.

2.1. Workflow publishing and sharing

The ability to share workflows among scientists is key for proper reuse and adaptation of existing workflows, as well as a means to provide attribution and feedback from other scientists. The simplest way in which a scientific workflow (or any of its associated resources) may be shared is through data repositories and digital archives. In fact, popular data repositories like Zenodo[2] and Dryad[3] are starting to be heavily used by the community to share datasets and results of their experiments, since they can assign them a Digital Object Identifier[4] (DOI) for referencing them in their work. DOIs are an important feature for sharing workflow resources, as they ensure their persistence and proper credit through

citation. However, in these repositories every description of a resource has to be added manually, and the links and relationships between the rest of the components of the workflow are often missing (e.g., an intermediate result would not be linked to the step that produced it or used it unless it is manually added by the user) even when they have their own entries.

Workflow repositories such as myExperiment [9] and Crowd-Labs [10] can be used for sharing scientific workflows created with different systems. These repositories store workflows in their native languages, that is, without requiring their conversion to a common language. Although they are great resources for sharing workflows in the community, they do not usually include links to their executions and have to be manually documented. In order to address this issue, some repositories have started to adopt Research Objects [21] which bundle together all the resources used within a scientific experiment. However, these are currently in early stages of adoption, relying heavily on manual curation by users.

Another approach for publishing workflows is science gateways, which include applications, libraries of components and workflows that are integrated through a portal. Some examples are described in [22] for volcanology, in [23] for hydro-meteorology, or in [3] for neuro-image analysis. Unlike workflow repositories, science gateways usually provide the infrastructure to execute some of the shared components through such community portals.

Finally, prior work [24,25] has explored the publication of workflows as Linked Data. In this paradigm, the published workflow resources (e.g., inputs, outputs, intermediate results, etc.) are associated with a URI (Uniform Resource Identifier) to be accessible as web objects. An interesting contribution of that work is to illustrate how the workflow inputs and outputs can be linked to other resources in the Linked Data cloud. However, the workflows (or their resources) were not published using any standard, so they are only reusable by the workflow system used to create them. This damages interoperability and affects the potential for understanding and reusability by others.

2.2. Provenance of workflow executions

Most of the existing workflow systems capture workflow execution provenance records at some degree in order to help scientists debugging their experiments. Some examples include Galaxy [17], Taverna [4], WINGS [16] or Vistrails [27]. Other tools have also been created to help scientists who use scripting languages like the Jupiter Notebooks[5] and Apache Zeppelin,[6] which aim to resemble a scientist's lab notebook.

After the consolidation of the W3C standard model for representing provenance in the web (W3C PROV) [20], many systems have started to implement it [28], including scientific workflow systems. Different models have been developed to adapt PROV to scientific workflow executions[7] [29,21,30], but at the moment there is not a standard for scientific workflow representation.

Provenance repositories have started to emerge in order to store workflow executions [31,32]. However, the provenance stored in these repositories is difficult to use at the moment without being familiar with the underlying provenance model.

2.3. Exploitation of workflow resources

While a growing number of workflow systems capture workflow executions as provenance records, very few make the

---

associated workflow resources (i.e., inputs, outputs, intermediate resources and codes) accessible and browseable by scientists. The most relevant effort in this regard is the PBase provenance repository [31], which includes predefined types of queries for the users to edit when exploring the repository. Although PBase is a step forward towards addressing the exploitation of workflow resources, users still need to upload their traces and edit their queries using a graph database language.

General workflow repositories like myExperiment [9] and CrowdLabs [10] provide the means to enable workflow accessibility, browsing and search. These repositories have often been exploited by workflow mining researchers to perform different operations with a real workflow corpus for purposes ranging to workflow search enhancement [33] to workflow clustering [34]. However, they do not contain links to the execution traces.

3. Requirements for workflow reuse and exploitation

In this section we review, summarize and expand on the existing requirements defined in the literature to make use of the different aspects of scientific workflows and their executions from an exploitation perspective. The existing infrastructure for the publishing and consumption of provenance traces (and their links to their respective workflow specifications) remains limited to date and does not address these requirements.

3.1. Workflow reuse and exploitation scenarios

We consider here three major purposes for exploiting a workflow repository. One purpose is for gathering information about workflows in the repository in order to understand, evaluate or compare existing results with new work. A second purpose is for updating and adapting existing workflows and their associated resources in order to reuse them to do new research. Both cases are related to workflow reuse, either by the original creators of the workflows, by researchers in their lab or by other colleagues in their institutions.

A third purpose for exploiting a workflow repository is automated workflow mining. Analysis of the workflows stored in the repository can lead to identify groups of results or other salient features of workflows. For example, different efforts have focused on workflow similarity browsing [35], clustering [34], searching [33] and reuse [36].

We illustrate each of these purposes in the scenarios presented below.

3.1.1. Workflow information gathering scenario

Our first scenario is composed by three user stories. Alice is a proteomics researcher who is working on a new version of a workflow she created previously. Alice wants to evaluate if her new results are consistent with those obtained in her original publication. Luckily, back in the day Alice executed the original workflow several times with different parameter values and datasets, selecting the appropriate results for a publication and storing them in a repository. By looking for the original workflow, Alice is able to recover the successful executions she stored, along with the input datasets and parameter values used to produce them. Alice then uses those input datasets and parameter values with her new version of the workflow and compares the outputs to the originals in order to validate her new workflow. She then publishes her new workflow and its executions in the repository, as a new version.

She is also interested in knowing how her research products (i.e., workflows, software, and data) are used by others within the workflow repository. In particular, she developed a nearest-neighbor clustering algorithm and several workflows to analyze proteomics data and visualize the clustering results. Alice wants

this information to report to her funders by pointing to examples where the code, datasets or workflows are used and to collect the list of labs and organizations benefiting from her work. By searching for the name of the software and workflows that she has developed, Alice is able to find a list of the workflow executions where the software and workflows are used. Alice notices that some of those executions failed due to problems with her software, and recovers the names of their creators and other details that will enable her to track and resolve bugs.

Alice now tries to find similar software that others have developed. She searches for workflows that use hierarchical clustering algorithms, which is the general class of algorithms that include nearest neighbor and other clustering methods. Alice finds hierarchical clustering in abstract descriptions of workflows, and through them she finds several implementations of similar software. Therefore, in order to understand each of the implementations, Alice retrieves the executions corresponding to the abstract workflow and studies their intermediate and final results and the parameter values used as input.

This scenario motivates the following requirements:

- Record a workflow execution and the execution of its steps.
- Retrieval of workflows and associated codes based on their name or identifier.
- Given a workflow, retrieval of its executions.
- Associate inputs, intermediate results, codes and outputs of a given workflow execution.
- Document creators/contributors of a workflow and its executions.
- Document versions of a workflow.
- Document categories of workflow steps.

3.1.2. Workflow updating scenario

In our second scenario, Bob is trying to run a workflow for deriving the drug-target network that a student of his lab ran some years ago. Unfortunately, the software that the student used for a molecular docking step is no longer maintained and requires a license that the research lab no longer has. Bob searches the repository for the workflow and finds it, but cannot re-run it with the original settings due to the lack of a license. Bob looks up the workflow template associated with the student's execution specifies abstract steps, and he finds other more recent workflows that execute components of the same abstract class. Thanks to those components, he finds similar software that was used in the last month by another colleague, has the same functionality and is open source. Bob checks the constraints of the input data (e.g., its format and characteristics) and results produced by the new software, and sees that an additional following data conversion step will be needed. Bob updates his workflow with the new step and adds the data conversion step.

The requirements derived from this scenario are the following:

- Retrieval of workflows based on a name or identifier.
- Retrieval of the executions of a workflow.
- Record of the abstract steps of a workflow.
- Retrieval of workflows with a given abstract step.
- Retrieval of Implementations of an abstract workflow step in other workflows.
- Record the types and constraints of data consumed and produced by a step in a workflow.
- Record steps that precede or follow a given step or workflow.

3.1.3. Workflow mining scenario

In our final scenario, Clarence is a proteomics scientist that aims to test a hypothesis that a certain protein is present in patient

Table 1

List of requirements for workflow template and provenance consumption, grouped by their category and pointing to the scenario where they appear. When an item is in brackets, it indicates that can be referred to any combination of resources included in it.

N°	Category	Requirement	Scenario
R1		[templates, steps, inputs, outputs] of the repository with a certain name	Information gathering, updating, mining
R2		Steps that precede or go after a certain step	Information gathering, updating
R3		Steps that [use, generate] a certain variable	Information gathering
R4	Template specification	[inputs, outputs, intermediate variables, parameters, steps] that belong to a template	Information gathering, updating, mining
R5		[documentation, instructions] defined by users on a template or its components	Information gathering, updating
R6		Current [version, creator, license, contributor] of the template	Information gathering
R7		[creation date, modification date] of a template	Information gathering
R8		[Inputs, codes, parameters intermediate results] involved in the [execution, generation of a result]	Information gathering, updating, mining
R9		Workflow executions that [used, generated] some [code, inputs, outputs, intermediate results, parameter values]	Information gathering, updating
R10	Workflow execution	[Outputs, intermediate results] of workflows with a certain [input dataset, parameter value]	Information gathering
R11		Status of the processes in the run (and the run)	Information gathering, mining
R12		Execution runs that share common configuration parameters, inputs and results	Mining
R13		Creator of the workflow execution	Information gathering
R14		[Start date, end date] of the execution run	Information gathering, mining
R15	Linkage	All executions associated with [a template specification, workflow step, variable]	Information gathering, updating, mining
R16		All templates that are associated with a [input dataset, intermediate result, output]	Information gathering
R17		Implementations of an abstract [workflow, component]	Information gathering, updating, mining
R18	Semantics	Components that have the same function as a given code.	Updating
R19		Workflows that [use, generate] an [input, output, code] type with specific semantic constraints	Updating
R20		All the [datasets, code] of a given type	Updating, mining

samples for a type of cancer. Clarence knows that his colleagues have published a set of execution runs of different workflows for this kind of analysis, and he wants to exploit those results to test his hypothesis. Thanks to the repository, Clarence is able to cluster all the execution runs based on their common types, parameter values and inputs datasets. After this categorization, he is able to analyze the results in an automated way and assign a confidence value to his hypothesis.

The requirements derived from the scenario are the following:

- Retrieval of the inputs and outputs of a given execution of a workflow.
- Shared inputs and parameter values among groups workflow execution traces.
- Record the types of input data used by a workflow.

3.2. Requirements summary

The requirements of the previous scenarios can be grouped into four main categories:

- **Workflow template specification requirements**, which tackle the representation and metadata of workflow steps and their dependencies, including how they are connected to each other or any restriction that a given input may have. These requirements also include the metadata of the workflow itself, necessary for tracking authors, contributors, version, license, creation date, etc.
- **Workflow execution (provenance) requirements**, which refer to the processes by which the results of the workflow have been obtained. These set of requirements also refer to the description of the execution as a whole (i.e., its inputs, outputs, intermediate results and codes), including its metadata (authors, contributors, license, etc.). In the literature, this set of requirements has been well defined in the provenance challenge series [37].
- **Linkage requirements**, which model the relationships among templates and executions. Some approaches that link both of them together have started to emerge in the last few years[8] [29,21,30], capturing crucial connections between executions and the methods defining them.

- **Semantic requirements**, which address the types and semantic constraints of data and steps being handled by the workflows. This category of requirements also involves the type of method used in the workflows. Although related work introduces the benefits of this type of semantic information for scientific workflows and tackles the problem from a modeling perspective [24,30], there are not many workflow systems that are capable of handling these set of requirements for workflow templates and executions.

Table 1 summarizes and generalizes the requirements according to these categories and scenario they belong to. Requirements R1–R14 may be similar to those tackled in other related work, as they are partially addressed by workflow systems when enacting and executing workflows (requirements R1–R7) [4,14,15] or representing provenance traces (requirements R8–R14).[9] Requirements R15–R20 are the ones where we focus our novel contributions on.

By combining some of the requirement categories (specially the linkage and semantics with the rest), we can create further requirements, such as retrieving the executions of a template that uses or generates certain type of data, comparison of results of runs that used different codes for the same template specification, etc. In Table 1 we have tried to simplify by focusing on those requirements that address the main goals covered by the use case scenarios. An additional list of requirements is available online.[10]

4. An end to end framework for workflow publication and exploitation

In this section we describe how to publish all workflows and workflow-related products (i.e., the workflow template specifications, execution related resources and metadata) in an automated manner. Several options exist for such publication, as described in Section 2, from workflow repositories to digital archives. In order to be able to reference all the resources properly, we have decided to follow the *Linked Data principles* [19]. According to those principles, we should use URIs as names (identifiers) for things, use HTTP URIs so that people can look up those names (making those URIs

---

8 http://purl.org/provone.

9 http://twiki.ipaw.info/bin/view/Challenge.
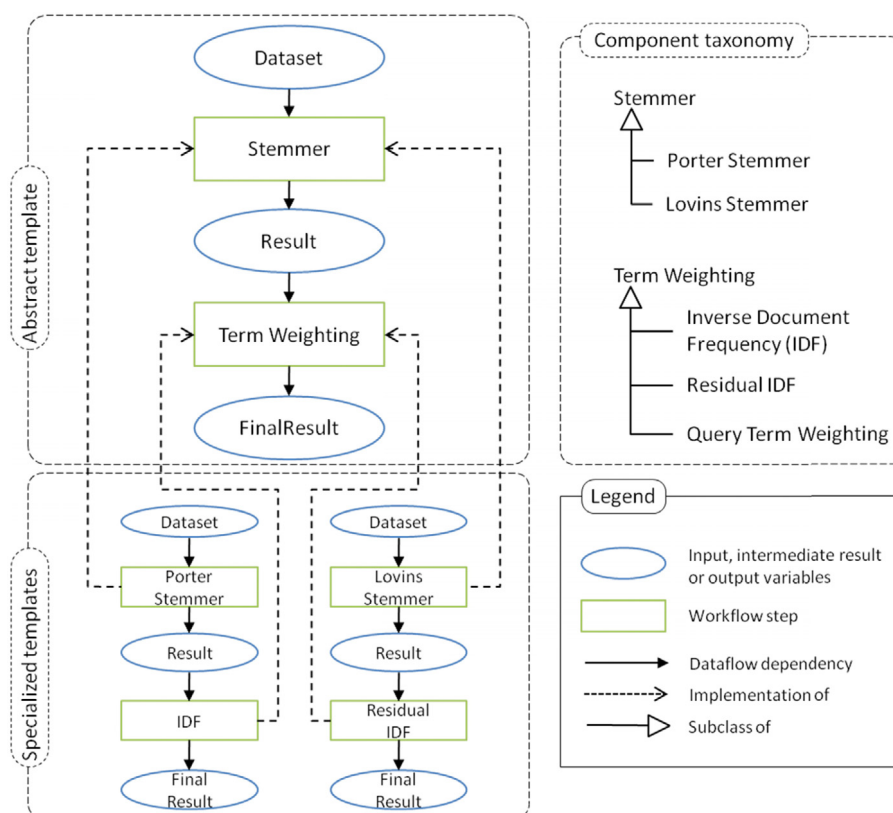10 https://dx.doi.org/10.6084/m9.figshare.3971994.v1.

Fig. 1. Example of abstract and specialized templates in the WINGS workflow system.

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 anyone can discover additional information.

There are three important advantages of publishing workflows and their resources as Linked Data. The first one is to get linked from other applications by pointing to the URIs that we publish, which include both the workflows and the data generated by them. The second advantage is the ability to produce interoperable results within different systems without having to define particular catalog structures and access interfaces. Just by using standard HTTP operations and formats (like JSON) anyone can use the information published in the repository. Finally, the third advantage is the ability to link to available web resources, for instance referring to proteins in the Protein Data Bank by using their published URI.

This section describes our generic methodology for workflow publication in an end to end scenario. We use WINGS [16] as the workflow system of our choice to exemplify our approach. The remainder of the section introduces the key type of abstraction we use to represent workflows, describes the model we have chosen to represent scientific workflows in an infrastructure independent manner (the OPMW model), illustrates the architecture and methodological steps that need to be carried out to publish the workflow template specifications and their executions as Linked Data on the Web and summarizes the main features of WINGS.

4.1. Representing semantic abstractions

We represent workflows with skeletal plan abstractions [38], where every step in the abstract plan always corresponds to a class of steps in the concrete plan. This type of abstraction allows a simple generalization of the steps in the workflow. An example can be seen in Fig. 1, showing an *abstract* template on the top and two *specialized* (concrete) templates on the bottom. By looking at

the taxonomy of workflow steps shown in the top right of the figure, we can see that both of the specialized templates shown are implementations of the same abstract method (with the Stemmer and Term Weighting steps).

This way, an abstract workflow template may have different levels of abstraction, depending on the generality aimed at by the designer on the workflow. This approach also supports mixing specific step implementations within an abstract template (e.g., if a particular step of the workflow will always use the same implementation).

4.2. Workflow template and execution representation: OPMW

As we mentioned in Section 2, W3C PROV is a standard for representing provenance on the Web. PROV uses three main concepts to represent provenance: *Entities*, i.e., the things we want to describe; *Activities*, i.e., the processes that consumed or produced entities; and *Agents*, who are the entities responsible for carrying out the activities. As shown in Fig. 2, these concepts capture the provenance of an entity by using seven main relationships: *used* (an activity used some entity), *wasAssociatedWith* (an agent participated in some activity), wasGeneratedBy (an activity generated an entity), *wasDerivedFrom* (an entity was derived from another entity), *wasAttributedTo* (an entity was attributed to an agent), *actedOnBehalfOf* (an agent acted on behalf of another agent) and *wasInformedBy* (an activity uses results from another activity).

PROV was designed to be generic and domain independent, and needs to be extended to address the requirements to represent workflow templates and executions. In order to achieve this goal, we have developed the OPMW model[11] [30]. Since PROV only has
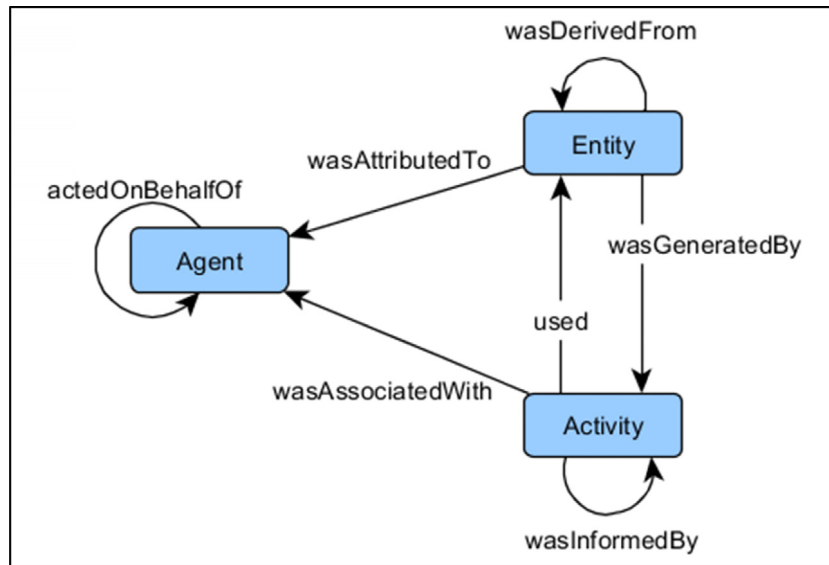
---

[11] http://www.opmw.org/ontology/.

Fig. 2. Overview of PROV.

one term to refer to the plan associated with the execution of an activity (called *Plan*), OPMW adopts the P-Plan[12] model [39], a model designed to capture the steps and variables of scientific processes, for addressing the template representation requirements. In addition, OPMW also extends the Open Provenance Model (OPM) [40], a legacy provenance model developed by the workflow community that was used as reference to create PROV.

OPMW supports the representations of workflows at a fine granularity with a lot of details pertaining to workflows that are not covered in those more generic languages. OPMW also allows the representation of links between a workflow template and a workflow execution that resulted from it. Finally, OPMW also supports the representation of roles and attribution metadata about a workflow.

4.2.1. OPMW overview

In OPMW, a workflow template specification is represented as an *opmw:WorkflowTemplate*. Workflow templates are formed by *opmw:WorkflowTemplateProcesses*, which use or produce *opmw: WorkflowTemplateArtifacts* hence capturing the dataflow dependencies.

On the execution side, each *opmw:WorkflowExecutionProcess* represents the execution of a workflow template process, and is bound to it via the *opmw:correspondsToTemplateProcess* relationship. Similarly, each *opmw:WorkflowExecutionArtifact* that is used or generated by a workflow execution process is linked to its corresponding workflow template artifact with the *opmw: correspondsToTemplateArtifact* relationship. Finally, the *opmw: WorkflowExecutionAccount* containing all the provenance statements of the execution is linked to the workflow template that contains all the assertions of the template with the *opmw: correspondsToTemplate* relationship.

Fig. 3 shows an example of the OPMW vocabulary extending OPM, PROV and P-Plan. Each vocabulary concept and relationship is represented with a prefix to help understand its source (OPMW, P-Plan, PROV or OPM (with opmo and opmv)). In the figure, an abstract workflow template with one sorting step, an input and an output (on the top right of the figure, represented using P-Plan) is linked to its provenance trace on the bottom right of the

figure (depicted with PROV and OPM). Each activity and artifact of the execution is linked to its respective step and variable on the template. Additional metadata of the variables (e.g., constraints), steps (e.g., conditions for execution), activities (e.g., used code), artifacts (e.g., size, encoding), account (e.g., status) and template (e.g., associated dataflow graph) is modeled with OPMW, but has been omitted from the figure for simplicity.

Attribution is crucial for scientists who create and publish workflows, as well as for those who provide the data or workflow execution infrastructure. Hence, OPMW reuses terms from existing vocabularies to represent them. For example, the Dublin Core (DC) Metadata Vocabulary is reused to represent the author, contributor, rights and license of an input dataset, a code used in a workflow or the workflow itself. OPMW also defines additional terms for referring to the start and end of the whole execution of the workflow, the size of the produced files, the status of the final execution, the tool used to design the workflow, the tool used to execute the workflow, etc.

4.3. Publishing workflows as data on the web

There are methodologies for Linked Data generation and publication [19]. However, there are no methodologies for publishing workflows and their constituents online. Therefore, we have developed a methodology that adapts an existing one used in the publication of government data [41] and other domains like energy consumption [42] to achieve our purpose.

The methodology consists of five main steps:

(1) **Specification**, where the of data sources to use are identified and a URI naming convention is designed (the license for the resulting dataset is often agreed during this step as well);
(2) **Modeling**, where users decide which vocabularies should be used to represent the data properly according to the existing requirements and scenarios;
(3) **Generation**, i.e., the process of transforming the data from their heterogeneous formats to a structured format, typically RDF,[13] with the help of existing tools; cleaning the data and linking it with other existent sources
(4) **Publication**, where the resulting dataset and its metadata is made available by using an online accessible triple store; and
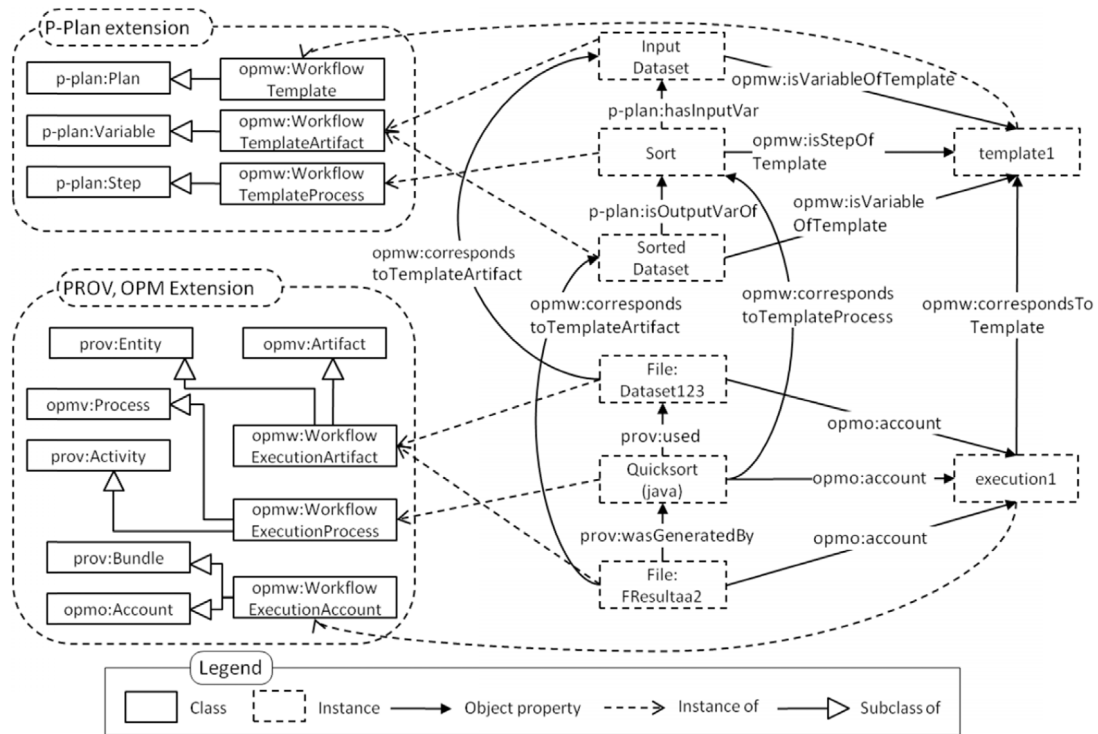

Fig. 3. OPMW example of an execution (bottom right of the figure) and the links to its corresponding workflow template (top right of the figure). The extensions to P-Plan, OPM and PROV can be seen in the left part of the figure.

(5) **Exploitation**, where the benefits of the dataset is made clear through applications or queries that consume it.

An overview of the architecture for our implementation of the methodology can be seen in Fig. 4. For the specification step, we decided that all the URIs generated by our system would become *cool URIs*,[14] following the W3C recommendations. This means that they are produced under a domain under our control, they are unique, and they are not going to change. Each URI identifies a different resource that can be individually accessed and dereferenced with content negotiation (i.e., the same URI can handle requests for users (returning HTML) and machines (returning an RDF serialization)). The URIs adopt the following naming scheme:

Base URI = http://www.opmw.org/
Ontology URI = http://www.opmw.org/ontology/
Assertion URI = http://www.opmw.org/export/resource/ClassName/instanceID.

As for the license, we choose the Creative Commons Attribution-Share Alike 3.0,[15] which allows anyone to reuse the published data if proper attribution is provided. However, the license might be changed if the authors prefer another one.

For the modeling step, we chose the OPMW vocabulary. Other approaches like the Research Object Model [21] or the PROV-One vocabulary[16] may be considered when addressing the modeling step.

For the generation step, the workflow template specifications and execution traces are converted automatically to OPMW with a transformation script.[17] The script takes as input a workflow template or an execution of the target workflow system, and produces an RDF file using the naming convention established

in the first step. Camel case notation is used for composing the identifiers of classes and instances, and an MD5 encoding is used to generate a unique identifier for each resource when necessary (e.g., a used file, an execution step, etc.).

The **publication step** takes as input the RDF files produced by the generation step. As shown in Fig. 4, an upload manager loads the files into a triple store and makes them available through a public endpoint (i.e., an access point for both human users and machines). We have selected Openlink Virtuoso[18] as our triple store because of its robustness, support from the community and mechanisms to create a public access. Whenever an execution run is published, all the referenced resources (inputs, intermediate results and outputs of the workflows) are versioned and saved in an additional permanent file store. If the file size is too big (size can be configured), then the pointers to the original location are maintained. The upload interface makes sure the consistency with the links used in the execution traces in the triple store, so they can be accessed appropriately. The file store is available in our local servers (http://www.opmw.org). A Linked Data Frontend is set up using Pubby[19] for facilitating the access to each individual resource of the endpoint. The public endpoint is available online,[20] along with a set of sample queries to retrieve basic data from workflows and demonstrate its main functionality.[21]

The last step of methodology, the exploitation phase, is described in Section 5.

4.4. Designing and executing workflows: WINGS

WINGS is a workflow system that uses semantic representations to describe the constraints of the data and computational

---

14 http://www.w3.org/TR/cooluris/.
15 http://creativecommons.org/licenses/by-sa/3.0.
16 http://purl.org/provone.
17 https://github.com/dgarijo/WingsProvenanceExport.

18 http://virtuoso.openlinksw.com/.
19 http://wifo5-03.informatik.uni-mannheim.de/pubby/.
20 www.opmw.org/sparql.
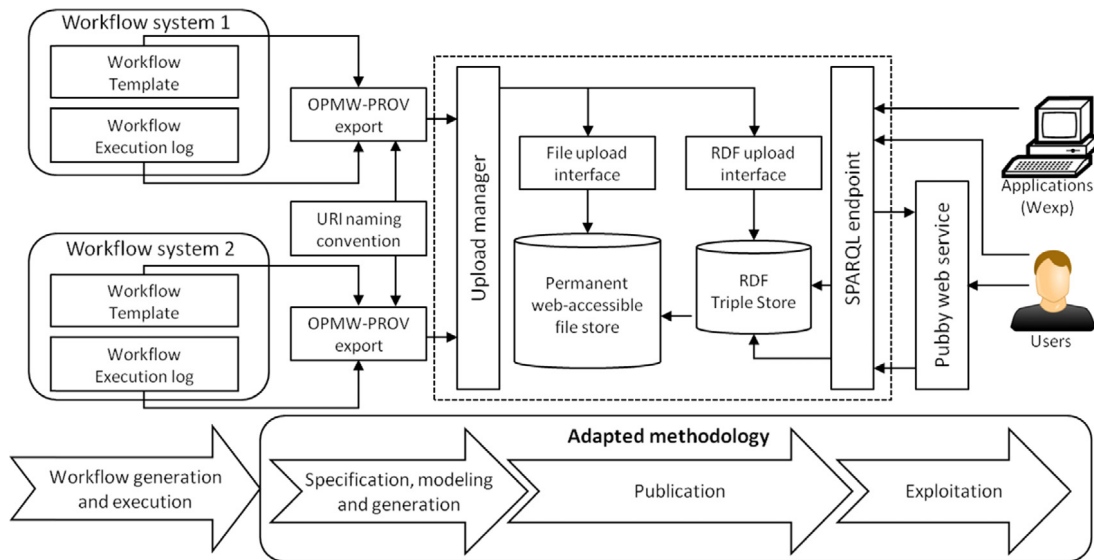21 http://www.opmw.org/node/6.

Fig. 4. Overview of our architecture for publishing workflow as Linked Data.

steps in the workflow [16,26]. WINGS can reason about these constraints, propagating them through the workflow structure and using them to validate workflow templates. It has been used in different domains, ranging from life sciences to multimedia analysis and geosciences. WINGS provides web-based access and can run workflows locally, or submit them to execution environments such as Pegasus/Condor [11] or Apache OODT [15], so as to handle large-scale distributed data and computations, optimization and data movement.

There are two main reasons why we have selected WINGS to exemplify our methodology for publishing workflows and their resources on the Web. The first one is that WINGS separates types of data and types of components when designing the workflow. This results in a data catalog and a component catalog organized in a taxonomy, which we can exploit for addressing the semantic requirements depicted in our scenarios. The second reason is that WINGS allows creating abstract workflow templates [16], similar to those presented in Section 4.1, which can then be instanced differently on each execution.

5. Workflow data exploitation

Once the workflow related data are available as web resources, they can each be independently accessed with their metadata, or through queries to the endpoint to retrieve more complex relationships. The data consumption can be done both from a human perspective and a machine point of view, since all the contents are dereferenceable and offer serializations in multiple formats on demand.

However, for "true users" [43] like Alice, Bob or Clarence this is often not enough, as they lack the skills necessary to be familiar with query languages like SPARQL, SQL or similar and they do not know the models or schemas the workflow data is represented in.

In this section we describe our approach for addressing this issue. We present a suite of tools (developed by ourselves or third parties) that we use for exploiting the contents of a repository of workflow templates and their execution traces published according to the principles discussed in Section 4.

5.1. Workflow information gathering and adaptation

We help end users browse, inspect, documentation and visualize the contents of the repository in an easy manner, for that does not require that they issue any queries against the workflow repository.

5.1.1. Workflow browsing

The Workflow Explorer (WExp)[22] allows navigating over different workflow templates, their metadata and their workflow execution results. WExp loads dynamically the workflow information stored in a repository, and allows the user to search it on demand.

The entry point for the application is a workflow template name. Whenever a user like Alice or Bob selects the one they are looking for, a series of lists with information appear on the screen as shown in Fig. 5. Each list shows resources related to the template, grouped by their common type and retrieving the data asynchronously from the server. First, the metadata list of the template is shown. It includes the creators and contributors (such as the ones Alice was searching to include in her report), the template license, version number, and the workflow system used to create it, as well as a picture of its workflow template specification (if available).

The rest of the lists show the data variables of the workflow template (i.e., inputs, outputs and intermediate variables) and their constraints, the steps of the workflow (so users like Bob can look for their configuration and results on the execution traces) and the available execution traces (which Alice wanted to explore for testing the new version of her workflow). Each resource is a link that can be resolved in the browser for more information, making it easy to know if an execution run has been successful or not, find the available implementations of a workflow step or how a variable in the workflow template has been instanced in the available executions. A live demo of WExp can be found online.[23]

5.1.2. Workflow documentation

The Organic Data Science Wiki[24] is an extension of semantic wikis designed to develop meta-workflows that result in many workflow explorations and runs. A snapshot of the interface is shown in Fig. 6 depicting a workflow template on the left, one of its executions on the top and the detail for one of the generated outputs on the bottom. By using this wiki, end users can import workflow templates and workflow executions into this framework to generate persistent documentation pages that link to data

---

22 https://github.com/dgarijo/WorkflowExplorer.

23 http://purl.org/net/wexp.

24 http://www.organicdatacuration.org/sandbox/index.php/Main_Page.

Fig. 5. Snapshot of WExp showing the information of a workflow.

Fig. 6. Snapshot of the Organic data publishing wiki, showing a page of a workflow template (left), its execution (top) and metadata of one of the used values (bottom right).
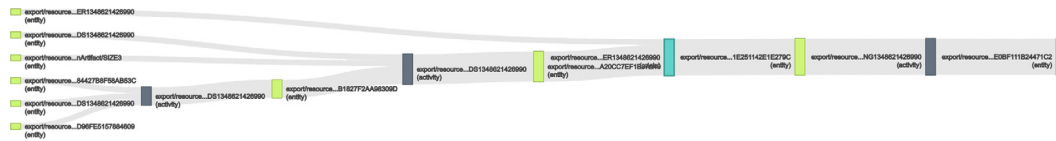
Fig. 7. PROV-O-Viz sample visualization diagram of a workflow execution.

and algorithm descriptions in the wiki. End users can augment this documentation as they relate to the overall meta-workflow. This may help not only improve the quality of the workflows and results they use for their experiments, but also foster collaboration with other scientists in other domains. The pages look similar to what WExp presents, as they include inputs and outputs, metadata of the workflow, steps and codes used in the executions, etc. The advantage of this approach is that the initial page for the workflow documentation is populated automatically from the contents of the repository, making it easy to navigate through the workflow and explore its main resources and executions. While WExp is for searching and retrieving particular types of data, the Organic Data Science Wiki focus on high quality descriptions and automated documentation of workflows, their execution traces and results.

Finally, the components used for the workflow execution can be further browsed and documented in the OntoSoft portal[25] [44]. This is especially useful for users who want to find and replace old software with other compatible versions equal in functionality. The OntoSoft portal is a platform created to provide a community repository for software, allowing contributors to specify structured metadata about software, assisting users to make their software more reusable (e.g., by suggesting which information is missing to be able to rerun it) and maintaining compatibility with existing software catalogues such as Github or BitBucket. The OntoSoft ontology [44][26] is used by the portal to provide a vocabulary for allowing users to publish and describe their software in terms of accessibility, execution, citation and understanding.

5.1.3. Workflow execution trace visualization

We use the PROV-O-Viz tool [45] for visualizing workflow executions stored in the repository, so users like Alice can get a quick overview of an execution trace. PROV-O-Viz creates a Sankey[27] diagram out of the execution trace of the workflow, which helps provide a general insight into how the inputs influence the final results. A screenshot is shown in Fig. 7, showing a sample of a simple workflow execution. PROV-O-Viz was designed to work in a generic domain with the W3C PROV standard, but we consider a useful asset for creating overviews of workflow traces. Thanks to the OPMW compatibility with PROV, we are able to benefit from this and other applications that may be developed in the future.

5.2. Mining workflow data

The efforts shown in the previous section focus on presenting and facilitating the access to data to users so they can consume it without having to perform complex queries to a repository. Next, we detail our own efforts to consume the data for exploiting its value.

5.2.1. Meta-workflow analysis

An important aspect of publishing workflows is being able to consume them afterwards to review and use their findings for other experiments. We do so in the DISK system [46],[28] introducing a novel approach to automate the hypothesize-test-evaluate discovery cycle with an intelligent system that a scientist can task to test hypotheses of interest in a data repository.

The DISK approach captures three types of data analytics knowledge: (1) common data analytic methods represented as semantic workflows; (2) meta-analysis methods that aggregate those results, represented as meta-workflows; and (3) data analysis strategies that specify for a type of hypothesis what data and methods to use, represented as lines of inquiry.

Given a hypothesis specified by a scientist, appropriate lines of inquiry are triggered, which lead to retrieving relevant datasets, running relevant workflows on that data, and finally running meta-workflows on these and previous workflow results (retrieved from the repository). The scientist is then presented with a level of confidence on the initial hypothesis (or a revised hypothesis) based on the data and methods applied.

5.2.2. Finding commonly occurring fragments in workflows

We aim to facilitate workflow reuse and understanding by mining the templates of our repository to find commonly occurring fragments of workflows. Our approach, FragFlow [36], collects all the workflow templates from an existing repository and performs a series of analyses by using state of the art frequent sub-graph mining techniques. As a result, we obtain a series of *workflow fragments*, which relate workflows to each other and indicate the parts of the repository that are more likely to be reused. FragFlow also exploits the abstract methods available in the repository, verifying whether two templates might share common fragments at a higher level of abstraction. When applied to multi-disciplinary workflow repositories, FragFlow may be used to discover common methods used by different users, groups or even disciplines.

6. Results and discussion

Having introduced our suite of tools, in this section we aim to assess (1) the feasibility of our methodology for publishing workflows in the web in an automated manner, (2) whether our suite of tools fully addresses the requirements identified in Section 3 for workflow exploitation and (3) additional exploitation opportunities that we have not addressed in this work.

6.1. Methodological feasibility

Following the steps introduced in Section 4, we have published on the public endpoint 73 workflow templates and 106 workflow executions traces of the WINGS workflow system. In addition, in order to test the feasibility of our approach with other workflow systems, we have also published 91 workflows from the LONI Pipeline public library.[29] The workflows belong to different

---

domains, ranging from text analytics to pharmacology or water quality analysis. The size of each workflow is also variable. For example, we have workflows with only a few steps, while others have executions with more than 170. Each template includes all its original metadata and links to its corresponding executions (if any). For each execution, all the metadata and inputs, results, intermediate outputs and used codes have been made available as well. This includes files ranging from a few bytes to almost 20 GB. All the workflows and executions, as well as the tools presented in previous sections, have been made part of the *WEST workflow ecosystem* [47], which integrates them through OPMW and P-Plan to interoperate at different levels of abstraction.

From the modeling perspective, we have tested the feasibility of our approach by answering each of the requirements listed in Table 1 with the OPMW model and examples from the repository. Each requirement has been assigned a SPARQL query that addresses it. The resultant table from this effort is available online.[30] Since each of the requirements can be answered successfully, we conclude that our model is feasible to represent the workflow templates and executions we want to exploit.

Regarding the validation of the resources required to support publication according to the Linked Data principles, we have used the Vapour system,[31] which launches a set of tests retrieving the exposed data in different formats typically consumed by humans (e.g., html) and machines (e.g., RDF/XML). In addition, we have performed a set of unitary tests to check any inconsistencies on the data.[32]

Given that we are able to answer the requirements for workflow template and execution exploitation, we have published successfully a corpus of workflows and executions and we have validated that they are accessible under the Linked Data principles, we can conclude that our proposed methodology is sufficient for publishing workflows as data on the web.

That said, keeping track and storing all the data resultant from an experiment may cause scalability issues in certain scenarios. For example, external repositories may be used as an input of a workflow. In other cases, thousands of execution runs may have to be explored before achieving the desired results. In our approach, the total amount of triples required by the system to represent the 270 workflow templates and executions in the repository is around 136,000. Triple stores are nowadays capable of handling billions of triples, which would ensure the storage of tens of thousands of workflow executions and templates in a single instance. Regarding the accessibility to external repositories with big data, we do not consider a big issue not storing the whole dataset each time an experiment is run. Instead, the author can provide enough metadata by, for example, linking to the source repository and to the queries used to retrieve the workflow data.

6.2. Workflow exploitation: conformance to requirements

In this section we analyze whether each of the requirements defined in Section 3 are addressed by the tools we propose to facilitate the exploitation of workflow data. The first category is the template specification requirements. Thanks to the Workflow Explorer and the Organic Data Science Wiki, template metadata like documentation (R5), version, creator (R6) and creation dates (R7) are shown to the user when accessing a workflow by its name (R1). Similarly, the inputs, outputs and other resources (R4) are also displayed when a workflow template is accessed in any of

the applications. The steps that use or generate a certain variable within the workflow template (R3) can be solved by resolving the URI of the desired template in the browser. Finally, the list of steps that precede or go after a certain step (R2) is not directly shown in the applications, but can be inferred easily from the list of processes and the available visualization of the workflow.

The workflow execution requirements are addressed in a similar way. The metadata of the execution e.g., status (R11), creator (R13) and start and end dates (R14) are shown by default whenever accessing an execution of the Organic Data Science Wiki, along with the codes, inputs and intermediate values that are involved in the generation of a result (R8). The workflow executions that used or generated a certain dataset (R9) and the outputs of workflows with a certain input (R10) can be found out by searching in the wiki for the result or by simply resolving the result itself to obtain an HTML description of its metadata. However, the current support for finding groups of execution runs that share parameter values or inputs (R12) is limited (not directly shown on our current suite of tools), and part of our future work.

The remaining two categories of requirements are addressed by WExp, the Wiki and our mining applications. For each template searched in the explorer, a link to each execution is provided (R15), and whenever a resource URI is resolved against the server, the templates that instantiates it are specified with the *correspondsToTemplateArtifact* relationship (R16). The implementations of an abstract component (R17) can be seen by dereferencing the URI of that component, by looking for the workflow in WExp or by running FragFlow in the repository to find abstract workflow fragments. Abstract workflow fragments link together the different parts of workflows which share the same functionality, being able to know which components share the same function at a higher level of abstraction (R18). Finally, by looking for a certain semantic type on the Wiki, one can retrieve files associated with it (R20), which can be further explored to retrieve the workflow they belong to (R19).

In summary, 18–20 the requirements are fully supported by our suite of tools, and when they are not supported directly, the applications provide the information necessary to address them.

6.3. Additional exploitation opportunities

There are two main additional aspects that have not been addressed in our work, but that may benefit from our current approach. The first one is the automatic re-execution of the published workflows. This is facilitated because the provenance traces have all the required information and code used in the workflow, while the workflow templates represent the order and data dependencies needed to link the different steps. In fact, we have successfully used OPMW and P-Plan to execute published workflows in different execution engines [48]. However, executing workflows in a different environment would require us to extend our current approach: important metadata would need to be captured, such as the software dependencies that have to be installed for each of the workflow steps or the execution environment requirements (memory, OS, CPU, etc.). We believe that by incorporating models that capture the missing metadata (e.g., WICUS [49]) or by using virtual execution environments (e.g., Docker containers[33]) we may be able to address most of the challenges associated with workflow re-execution.

The second main aspect that would benefit from our work is the automatic generation of explanations for different workflow results. We believe that despite all the tools presented here for

---

[30] https://dx.doi.org/10.6084/m9.figshare.3971994.v1.

[31] http://linkeddata.uriburner.com:8000/vapour.

[32] https://github.com/dgarijo/OPMWValidator.

[33] https://www.docker.com/what-docker.

workflow visualization and browsing, there is still a need to summarize, compare and explain the results of the workflow in a way that is human readable, depending on the users' needs (e.g., an abstract method for understanding the main steps, the specific software used, etc.). Our approach creates the knowledge base needed to support these kinds of explanations and interactions between different levels of abstraction.

7. Conclusions and future work

An important movement in science is open science, increasingly facilitating access to scientific products by anyone. By analogy with open publications and open data, the open availability of workflows would enable greater reuse of workflows as scientific artifacts. This often involves the publication of the workflow associated resources, in order to provide a provenance trace with evidence of how a particular result was obtained.

In this paper we have described our approach to publish abstract workflow templates and their respective workflow execution traces as data on the web. We believe that publishing workflows and execution traces without providing the means to end users to consume these resources may be a barrier for their reuse. Our contributions to address this issue include a collection of requirements and use cases for consuming and exploiting workflow resources (derived from our own experience and the state of the art), a methodology for publishing workflow as web objects and a suite of tools that can consume the published data in order to facilitate its understanding to users.

The benefits of our approach promote the ability to share and consume each of the published resources by using standard HTTP operations, allow the definition of an evaluation framework made from the requirements collected from the use case scenarios and enable the interoperability between the applications exploiting the data and third party tools. In addition, thanks to our representation based on standards, the published workflows and their traces are agnostic about the workflow system used for their design.

We have tested the feasibility of our methodology by publishing a corpus of workflows from two workflow systems and we have evaluated our approach against the requirements driven by the common scenarios. As a result, we have verified that the majority of the requirements are satisfied by our current suite of applications. Those requirements only partially addressed (e.g., comparison among workflow execution traces) are part of our ongoing work.

Regarding future work, our current approach has focused on the publication of all the resources involved in the workflow as Linked Data on the Web. However, some of these resources may contain sensitive information that should not be published along with the rest of the experiment. Although the endpoint and server storing the resources can be modified to limit the access to certain published files, we are currently studying the alignment of our approach with a conditional access Linked Data framework[34] that allows users accessing on the data based on their roles and privileges. On a related point, we also aim to explore how our publishing framework could be used in the early stages of workflow development (i.e., sharing the preliminary contents of the research only with the working group).

Another area of related work relies on the interoperability of the represented data. As we have shown in our previous work [47] the OPMW model we have used to represent the workflow data can be mapped to other existing approaches. Since there are emerging efforts that publish workflow data according to these other approaches [31] we aim to be able to exploit these data as well. In addition, we plan to explore the possibility of having orchestrating services that help the communication and set up of the workflow ecosystem comprising all the tools consuming the workflow data (e.g., those used for DevOps in software engineering projects).

Finally, another challenge to be addressed is regarding the persistence of the published resources. At the moment we assign them URIs that are not expected to be changed, but we would like to associate each of the resources with a DOI so as to be able to cite them in publications.

References

[1] E. Deelman, D. Gannon, M. Shields, I. Taylor, Workflows and e-science: An overview of workflow system features and capabilities, Future Gener. Comput. Syst. 25 (5) (2009) 528–540.

[2] J. Ruiz, J. Garrido, J. Santander-Vela, S. Sánchez-Expósito, L. Verdes-Montenegro, AstroTaverna: Building workflows with Virtual Observatory services, Astron. Comput. 7–8 (2014) 3–11. Special Issue on The Virtual Observatory: I.

[3] I.D. Dinov, J.D.V. Horn, K.M. Lozev, R. Magsipoc, P. Petrosyan, Z. Liu, A. MacKenzie-Graham, P. Eggert, D.S. Parker, A.W. Toga, Efficient, distributed and interactive neuroimaging data analysis using the LONI Pipeline, in: Frontiers in Neuroinformatics, Volume 3, 2009.

[4] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A.N. de la Hidalga, M.P.B. Vargas, S. Su, C. Goble, The Taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud, Nucleic Acids Res. (2013).

[5] Y. Gil, E. Deelman, M.H. Ellisman, T. Fahringer, G. Fox, D. Gannon, C.A. Goble, M. Livny, L. Moreau, J. Myers, Examining the challenges of scientific workflows, IEEE Comput. 40 (12) (2007).

[6] F. Leisch, Sweave: Dynamic generation of statistical reports using literate data analysis, in: Proceedings of Computational Statistics, 2002.

[7] J.P. Mesirov, Accessible reproducible research, Science 327 (2010) 415.

[8] S. Falcon, Caching code chunks in dynamic documents: The weaver package, Comput. Statist. 24 (2) (2007).

[9] D. De Roure, C. Goble, R. Stevens, The design and realizations of the myExperiment Virtual Research Environment for social sharing of workflows, Future Gener. Comput. Syst. 25 (561–567) (2009).

[10] P. Mates, E. Santos, J. Freire, C.T. Silva, CrowdLabs: Social analysis and visualization for the sciences, in: 23rd International Conference on Scientific and Statistical Database Management, (SSDBM), Springer, 2011, pp. 555–564.

[11] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.H. Su, K. Vahi, M. Livny, Pegasus: Mapping scientific workflows onto the grid, in: M. Dikaiakos (Ed.), Grid Computing, in: Lecture Notes in Computer Science, vol. 3165, Springer, Berlin/Heidelberg, 2004, pp. 11–20.

[12] I.J. Taylor, Triana generations, in: Presented at: Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), Amsterdam, Netherlands, 4-6 December 2006. E-Science 2006: Second IEEE International Conference on e-Science and Grid Computing: 4-6 December 2006, Amsterdam, Netherlands. Los Alamitos, CA: IEEE, p. 143.

[13] C.E. Scheidegger, H.T. Vo, D. Koop, J. Freire, C.T. Silva, Querying and re-using workflows with VisTrails, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08, ACM, New York, USA, 2008, pp. 1251–1254.

[14] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E.A. Lee, J. Tao, Y. Zhao, Scientific workflow management and the Kepler system, Concurr. Comput.: Pract. Exper. 18 (10) (2006) 1039–1065.

[15] C.A. Mattmann, D.J. Crichton, N. Medvidovic, S. Hughes, A software architecture-based framework for highly distributed and data intensive scientific applications, in: Proceedings of the 28th International Conference on Software Engineering, ICSE'06, New York, NY, USA, 2006, pp. 721–730.

[16] Y. Gil, V. Ratnakar, J. Kim, P.A. Gonzalez-Calero, P.T. Groth, J. Moody, E. Deelman, WINGS: Intelligent workflow-based design of computational experiments, IEEE Intell. Syst. 26 (1) (2011) 62–72.

[17] J. Goecks, A. Nekrutenko, J. Taylor, Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, Genome Biol. 11 (8) (2010).

[18] M. Reich, T. Liefeld, J. Gould, J. Lerner, P. Tamayo, J.P. Mesirov, Genepattern 2.0, Nature Genet. 38 (5) (2006) 500–501.

[19] T. Heath, C. Bizer, Linked data: Evolving the web into a global data. Space, in: Synthesis Lectures on the Semantic Web, Morgan & Claypool Publishers, 2011.

---

34 http://conditional.linkeddata.es/.

[20] T. Lebo, D. McGuiness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, J. Zhao, The PROV ontology, W3C Recommendation. Technical report, WWW Consortium, 2013.

[21] K. Belhajjame, J. Zhao, D. Garijo, M. Gamble, K. Hettne, R. Palma, E. Mina, O. Corcho, J.M. Gómez-Pérez, S. Bechhofer, G. Klyne, C. Goble, Using a suite of ontologies for preserving workflow-centric Research Objects, Web Semant. Sci. Serv. Agents World Wide Web (2015).

[22] R. Filgueira, M. Atkinson, A. Bell, I. Main, S. Boon, C. Kilburn, P. Meredith, eScience gateway stimulating collaboration in rock physics and volcanology, in: e-Science (e-Science), 2014 IEEE 10th International Conference on, Vol. 1, IEEE, 2014, pp. 187–195.

[23] E. Danovaro, L. Roverelli, G. Zereik, A. Galizia, D. DAgostino, G. Paschina, A. Quarati, A. Clematis, F. Delogu, E. Fiori, A. Parodi, C. Straube, N. Felde, Q. Harpham, B. Jagers, L. Garrote, L. Dekic, M. Ivkovic, O. Caumont, E. Richard, Setting up an hydro-meteo experiment in minutes: The DRIHM e-infrastructure for HM research, in: e-Science (e-Science), 2014 IEEE 10th International Conference on, Vol. 1, 2014, pp. 47–54.

[24] P. Missier, S.S. Sahoo, J. Zhao, C. Goble, A. Sheth, Janus: from Workflows to semantic provenance and linked open data, in: Provenance and Annotation of Data and Processes Third International Provenance and Annotation Workshop IPAW 2010 Troy NY USA June, 2010 Revised Selected Papers 6378, 2010, pp. 129–141.

[25] A. Shaon, S. Callaghan, B. Lawrence, B. Matthews, A. Woolf, T. Osborn, C. Harpham, A linked data approach to publishing complex scientific workflows, in: Proceedings of the IEEE 7th International Conference on eScience, Stockholm, 2011, pp. 303–310.

[26] Y. Gil, P.A. Gonzalez-Calero, J. Kim, J. Moody, V. Ratnakar, A semantic framework for automatic generation of computational workflows using distributed data and component catalogs, J. Exp. Theor. Artif. Intell. 23 (4) (2011).

[27] F. Chirigati, J. Freire, D. Koop, C. Silva, VisTrails provenance traces for benchmarking, in: Proceedings of the Joint SDBT/ICDT 2013 Workshops, 2013, pp. 323–324.

[28] T. Dong-Huynh, P. Groth, S. Zednik, PROV Implementation Report. W3C Working Group Note. WWW Consortium. 30 April 2013.

[29] P. Missier, S. Dey, K. Belhajjame, V. Cuevas-Vicenttín, B. Ludäscher, D-PROV: Extending the PROV provenance model with workflow structure, in: Proceedings of the 5th USENIX Workshop on the Theory and Practice of Provenance, TaPP'13, USENIX Association, Berkeley, CA, USA, 2013, pp. 9:1–9:7.

[30] D. Garijo, Y. 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.

[31] Víctor Cuevas-Vicenttín, Parisa Kianmajd, Bertram Ludäscher, Paolo Missier, Fernando Chirigati, Yaxing Wei, David Koop, Saumen Dey, The PBase scientific workflow provenance repository, Int. J. Digit. Curation 9 (2) (2014) 28–38.

[32] Khalid Belhajjame, Jun Zhao, Daniel Garijo, Aleix Garrido, Stian Soiland-Reyes, Pinar Alper, Oscar Corcho, A workflow PROV-corpus based on taverna and WINGS, in: Proceedings of the Joint EDBT/ICDT 2013 Workshops, Genova, Italy, 2013, pp. 331–332.

[33] J. Starlinger, S. Cohen-Boulakia, U. Leser, Layer decomposition: An effective structure-based approach for scientific workflow similarity, in: e-Science (e-Science), 2014 IEEE 10th International Conference on, Vol. 1, 2014, pp. 169–176.

[34] J. Stoyanovich, B. Taskar, S. Davidson, Exploring repositories of scientific workflows, in: Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science, Wands'10, ACM, New York, NY, USA, 2010, pp. 7:1–7:10.

[35] R. Bergmann, Y. Gil, Similarity assessment and efficient retrieval of semantic workflows, Inf. Syst. 40 (2014) 115–127.

[36] Detecting common scientific workflow fragments using templates and execution provenance, in: Proceedings of the Seventh International Conference on Knowledge Capture, K-CAP'13, 2013, pp. 33–40.

[37] L. Moreau, B. Ludäscher, I. Altintas, R.S. Barga, S. Bowers, S. Callahan, G. Chin, B. Clifford, S. Cohen, S. Cohen-Boulakia, S. Davidson, E. Deelman, L. Digiampietri, I. Foster, J. Freire, J. Frew, J. Futrelle, T. Gibson, Y. Gil, C. Goble, J. Golbeck, P. Groth, D.A. Holland, S. Jiang, J. Kim, D. Koop, A. Krenek, T. McPhillips, G. Mehta, S. Miles, D. Metzger, S. Munroe, J. Myers, B. Plale, N. Podhorszki, V. Ratnakar, E. Santos, C. Scheidegger, K. Schuchardt, M. Seltzer, Y.L. Simmhan, C. Silva, P. Slaughter, E. Stephan, R. Stevens, D. Turi, H. Vo, M. Wilde, J. Zhao, Y. Zhao, Special issue: The first provenance challenge, Concurr. Comput.: Pract. Exper. 20 (2008) 409–418.

[38] P.E. Friedland, Y.J. Iwasaki, The concept and implementation of skeletal plans, Automat. Reason. 1 (1985) 161.

[39] D. Garijo, Y. 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.

[40] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, J.V. den Bussche, The open provenance model core specification (v1.1), Future Gener. Comput. Syst. 27 (6) (2011).

[41] B. Villazon-Terrazas, L. Vilches-Blazquez, O. Corcho, A. Gomez-Perez, Methodological guidelines for publishing government Linked Data, in: D. Wood (Ed.), Linking Government Data, Springer, New York, 2011, pp. 27–49.

[42] F. Radulovic, M. Poveda-Villal_on, D. Vila-Suero, V. Rodriguez-Doncel, R. Garcia-Castro, A. Gomez-Perez, Guidelines for Linked Data Generation and publication: An example in building energy consumption, Autom. Constr. 57 (2015) 178–187.

[43] S. Cohen, U. Leser, Search, adapt and reuse: the future of scientific workflows, ACM SIGMOD Rec. 40 (2) (2011) 6–16.

[44] Y. Gil, D. Garijo, V. Ratnakar, OntoSoft: Capturing scientific software metadata, in: Proceedings of the 8th International Conference on Knowledge Capture. Palisades, NY, 2015.

[45] R. Hoekstra, P. Groth, PROV-O-Viz - understanding the role of activities in provenance, in: Proceedings of the International Provenance and Annotation Workshop, Cologne, 2014.

[46] Y. Gil, D. Garijo, V. Ratnakar, R. Mayani, R. Adusumilli, H. Boyce, P. Mallick, Automated hypothesis testing with large scientific data repositories, in: Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems, Evanston, IL, June 2016.

[47] D. Garijo, Y. Gil, O. Corcho, Towards workflow ecosystems through semantic and standard representations, in: Proceedings of the 9th Workshop on Workflows in Support of Large-Scale Science, WORKS'14, Piscataway, NJ, 2014, pp. 94–104.

[48] Y. Gil, Mapping semantic workflows to alternative workflow execution engines, in: Gil, Y. Seventh IEEE International Conference on Semantic Computing, ICSC, Irvine, CA, 2013.

[49] I. Santana-Pérez, M. Pérez-Hernandez, Towards reproducibility in scientific workflows: An infrastructure-based approach, Sci. Program. 2015 (2015) 11.

Daniel Garijo is a postdoc researcher at the Information Sciences Institute of the University of Southern California. His research activities focus on e-Science and the Semantic web, specifically on how to increase the understandability of scientific workflows using provenance, metadata, intermediate results and Linked Data.

Yolanda Gil is Principal Investigator and Project Leader of the Interactive Knowledge Capture research group at USC's Information Sciences Institute (ISI). Her research focuses on intelligent interfaces for knowledge capture, which is a central topic in our projects concerning knowledge-based planning and problem solving, information analysis and assessment of trust, semantic annotation tools, agent and software choreography, and community-wide development of knowledge bases. A recent focus is assisting scientists with large-scale applications throughout the design of workflows and their distributed execution.

Oscar Corcho is a Full Professor at Departamento de Inteligencia Artificial (Facultad de Informática, Universidad Politécnica de Madrid), and he belongs to the Ontology Engineering Group. His research activities are focused on Semantic e-Science and Real World Internet, although he also works in the more general areas of Semantic Web and Ontological Engineering. In these areas, he has participated in a number of EU projects (Wf4Ever, PlanetData, Semsor-Grid4Env, ADMIRE, OntoGrid, Esperonto, Knowledge Web and OntoWeb), and Spanish R&D projects (CEN-ITS mIO!, España Virtual and Buscamedia, myBigData, GeoBuddies), and has also participated in privately-funded projects like ICPS (International Classification of Patient Safety), funded by the World Health Organisation, and HALO, funded by Vulcan Inc.