Best practices for workflow design: how to prevent workflow decay

Kristina Hettne[1*], Katy Wolstencroft[2], Khalid Belhajjame[2], Carole Goble[2], Eleni Mina[1], Harish Dharuri[1], Lourdes Verdes-Montenegro[3], Julian Garrido[3], David de Roure[4], Marco Roos[1]

[1]Leiden University Medical Center, Leiden, The Netherlands
{k.m.hettne, e.mina, h.k.dharuri, m.roos}@lumc.nl
[2]University of Manchester, Manchester, United Kingdom
{katherine.wolstencroft, carole.goble,}@manchester.ac.uk
khalid.belhajjame@cs.man.ac.uk
[3]Instituto de Astrofísica de Andalucía, Granada, Spain
{Lourdes, jgarrido }@iaa.es
[4]Oxford e-Research Centre, University of Oxford, Oxford, United Kingdom
david.deroure@oerc.ox.ac.uk

Abstract. In this position paper we present a set of best practices for workflow design to prevent workflow decay and increase reuse and re-purposing of scientific workflows. MyExperiment provides access to a large number of scientific workflows. However, scientists find it difficult to reuse or re-purpose these workflows for mainly two reasons: workflows suffer from decay over time and lack sufficient metadata to understand their purpose. We argue that good workflow design is a prerequisite for repairing a workflow, or redesigning an equivalent workflow pattern with new components. We present a set of best practices for workflow design and the semantic tooling that is being developed in the Workflow4Ever (Wf4Ever) project to support these best practices.

Keywords: Scientific workflows, Decay, Best practices

1 Scientific Workflow Decay

Workflows are increasingly used in life sciences as a means to capture, share, and publish the steps of a computational analysis. Tools such as Taverna [1] and Galaxy [2] are widely adopted tools for creating and executing workflows, which can be published and shared using myExperiment [3], the largest public repository of scientific workflows. While myExperiment provides access to a large number of

workflows from different domains such as genomics, biodiversity and astronomy, scientists find it often difficult (or impossible) to exploit those workflows by reusing or re-purposing them for their analyses [4, 19]. Deciding factors in re-purposing or not would be the functionality of the workflow that the user is after, and if the existing workflow provides a fnctionality that can be used as a building block (i.e., as a sub-workflow) in the new workflow. Regarding workflow decay, in a recent study [5] of Taverna workflows on myExperiment, the authors found that as much as 80% failed to be either executed, or to produce the same results. The main reasons for the workflows to break were the following: volatile third-party resources, missing example data that can help the scientist understand the main purpose of the workflow, missing execution environment, and insufficient metadata. We believe that all of these issues, with a possible exception of the first one since it is not under the direct control of the designer, can be prevented by following a minimum set of guidelines at the workflow design stage. Unfortunately, to our knowledge no such guidelines exist, this hence leading us to define here 10 Best Practices for designing workflows.

2 Proposed Best Practices for Workflow Design

The proposed best practices for workflow design are based on combining the principles of the scientific method and the best practices for software development and data management [6]. Therefore, the next 10 steps allow the creation of higher quality workflows, as required in the scientific discourse.

1. **Make an abstract workflow:** A workflow sketch provides a reference to the main task(s) of the workflow through its implementation process. A workflow can be compared to a scientific protocol, so sketching out the method helps when designing the experiment. We also anticipate that a workflow sketch will help in communication with for example supervisors and colleagues, while at the same time promoting sharing between computer and human generated systems due to its non-explicit nature.
2. **Use modules:** One of the main strengths of workflows is the possibility of plugging in and reusing parts and also swapping broken parts, plugging in different methods and comparing them. Implementing all the executable components of a workflow in such a way that they can be run as separate subworkflows would facilitate the understanding, maintenance, re-use and separate testing and validation of the workflow.
3. **Think about the output:** What is the output intended for? Is it supposed to be used as input to another workflow, stored in a database, or be presented to the end user? Should it be a graph, a table or text? Thinking about the output of the workflow at the design stage is easier than trying to adjust a finished workflow and will drive the design of the next steps. Also, a workflow has the potential to produce masses of data that need to be visualized and managed properly.
4. **Provide input and output examples:** Inputs and output examples are crucial for: the understanding of the workflow, validation, maintenance

purposes, as well as to be able to use them as tools for training or tutorials.

5. **Annotate:** Careful annotation of a workflow helps to record all steps and assumptions hidden in the workflow, what is not only needed for a publication later on but also crucial for the scientific method. It also facilitates use and re-use of workflows. There is no accepted standard for annotating a workflow. We propose to choose meaningful names for the workflow title, inputs, outputs, and for the processes that constitute the workflow as well as for the interconnections between the components, so that annotations are not only a collection of static tags but capture the dynamics of the workflow. A high-level functional annotation should be included (for example similar to the functional units suggested in [7]), as well as a description of the resource, keeping in mind a scenario where it may disappear or change at some time in the future.
6. **Make it executable from outside the local environment:** This best practice leads to portability of the workflow which potentially increases its reproducibility and reuse. It can for example be realized either by using remote Web services, or platform independent code/plugins. However, if there is need to use local services, library or tools, then the workflow should be annotated in order to define its dependencies i.e. which local tool, version or operating system is required, where to find it, if it is licensed or any other particular restriction e.g. the application has to be called with a particular name.
7. **Choose services carefully:** One of the major reasons that cause workflows to break are volatile third-party services. The status, reliability and stability of a Web Service as well as the reputation of the service provider are often the deciding factors for choosing a service.
8. **Reuse existing workflows:** Reuse is important for many reasons. It fights redundancy, and perpetuates "tried and tested" and published methods conveying good scientific practice. It will also help the workflow developer get ideas on methods and workflow patterns. It is also beneficial when repairing workflows: repairing a given workflow may entail repairing the workflows in which it is used as a subworkflow.
9. **Test and validate:** Defining test cases and implementing validation mechanisms facilitates maintenance, decay identification and guarantee the correctness of the results by, for example including components in the workflow whose function is checking assertions that must be true.
10. **Advertise and maintain:** It is a duty of science to share your results. It also helps progress by letting others build on your work without reinventing it. Workflow maintenance is expected to increase the longevity of the workflows. Frequent testing, monitoring services used, communication with other users all represent ways to maintain a workflow.

2.1 Technology Supporting the Best Practices

Research Objects (ROs) are semantically rich aggregations of resources that bring together data, methods and people in scientific investigations [8]. Their goal is to

create a class of artifacts that can encapsulate digital knowledge and provide a mechanism for sharing and discovering assets of reusable research and scientific knowledge. In the EU Wf4Ever project [9] the focus is on those ROs whose methods are implemented as scientific workflows. A (workflow-centric) RO is an artifact that bundles one or several workflows, the provenance of the results obtained by their enactments, other digital objects that are relevant to the experiment (papers, datasets, etc.), and annotations that semantically describe all these objects. A model that can be used to describe these workflow-centric research objects is proposed in [10]. This model is implemented as a suite of lightweight ontologies or vocabularies, building upon existing work from related communities. The RO manager tool [11] was designed to prepare a workflow-centric RO and its related data and metadata.

3 Use case: Next-generation text mining for functional annotation of Single Nucleotide Polymorphisms (SNPs)

We followed the 10 best practices for workflow design when implementing a set of workflows in Taverna that perform a sophisticated text mining procedure referred to as concept profile matching [12] to functionally annotate SNPs. The set of workflows were published on myExperiment as a pack [13]. We describe point by point below how the RO model supported the design and implementation process.

1) Make an abstract workflow: We made workflow sketch using flowchart symbols, since there is no accepted standard for creating such a sketch. The sketch tries to capture the whole experiment at a higher level. It is aggregated in the RO as a wf4ever:Document. A file description was added using the rdfs:comment property.
2) Use modules: All executable components were implemented as separate, runnable workflows. These were described using the following properties of the RO model: class: wfdesc:Workflow and relation: wfdesc:hasSubWorkflow. This indeed facilitated independent testing and validation of the execution of each of the individual components as well as the main nested workflow.
3) Think about the output: The outputs of the workflows were implemented as output boxes in Taverna. They can be saved to disc from Taverna, for example using the save to Excel option. Although the output can be saved from Taverna in this way, the limited export options give a scattered impression and it can be difficult to relate the different outputs to each other. The outputs after a workflow run are annotated with the wfprov:wasOutputFrom relation.
4) Provide example inputs and outputs: All workflows have example inputs and outputs. The example outputs match the example inputs. However, there is no standard for how to do this if the example is a large data file that does not fit in the example window of Taverna. We solved this by providing the example files in the RO and use the wfdesc:Input and wfdesc:Output properties to annotate the example files.
5) Annotate: All elements of the workflows including inputs and outputs, processes (for example Web services) and subworkflows (nested components) were annotated using the description and example fields in Taverna. We used the Taverna description fields and example fields for workflow-related annotation since a workflow developer

is expected to provide the annotations continuously while developing the workflow. We used the Wf4Ever tool scufl2-wfdesc [14] to extract a wfdesc-compliant workflow description from a Taverna workflow.

6) Make it executable from outside the local environment: Since our workflows only use public Web services we experienced no problems when testing them outside the local environment. We also successfully executed them using t2web (e.g. http://workflow.biosemantics.org/t2web/workflow/2972), a web tool that uses the Taverna server [15].

7) Choose services carefully: The workflows use public Web services listed on BioCatalogue [16]. Service availability in BioCatalogue is indicated using a simple 'traffic light' mechanism, whereby green means the service is active, yellow means it has one or more unresolved issues, and red means it is currently unavailable. This green light, together with very limited history on BioCatalogue, is the only available trust-metric for a Web service. More effort to develop and implement reliability statistics for Web services is needed.

8) Reuse existing workflows: We made our workflows modular and noticed that the modularity made it possible to use one subworkflow twice in a nested workflow. This relates to ongoing work on Taverna workflow components, which we expect will make these types of implementations easier in the future.

9) Test and validate: We did notice that Taverna does not provide test mechanisms for the workflows and the nested workflows. For this reason, we used a well-known and small sample for testing (it matches with the sample data provided in the annotations).

10) Advertise and maintain: The workflows were put on myExperiment, both as separate workflows and as a pack [13]. We propose to use these links when referring to them from scientific publications. Other ways to advertise workflows include making them available through the Galaxy and Genome Space [17] environments. One major advantage of myExperiment is that it connects users to the creators of the workflow. We respond to e-mails from users regarding Web services being down and causing the workflow to break. Our maintenance plan is to perform monthly tests of the workflows by running them with their example values. A schedule for this in myExperiment with built-in reminders (for example, automatically generated e-mails alerting the developer that the workflow needs to be run) would help the maintenance.

4 Concluding remarks

When following the 10 best practices for workflow design we were helped by the available models and tooling, but we also noticed several technology gaps where specific guidelines or tooling would be of great help (see section 2). We suggest that something similar to unit testing would be useful for testing workflows and support workflow maintenance. The lack of testing support makes it even more important to integrate validation mechanisms in the workflow. We propose that myExperiment and Taverna draw from the same linked sources (the ROs), showing appropriate fields at appropriate times. The Wf4Ever project is working towards the release of an RO-enabled myExperiment. The pack functionality in myExperiment will be extended according to the RO model, which will constitute the interpretation layer between

Taverna and myExperiment. Ongoing work on nanopublications [18] might provide means to refer to different parts of a workflow or an RO in a more fine-grained way. The question on how to convince users that the best practices are beneficial in the long run remains.

Acknowledgements. The research reported in this paper is supported by the EU Wf4Ever project (270129) funded under EU FP7 (ICT-2009.4.1).

References

1. Taverna. http://www.taverna.org.uk.
2. Galaxy. http://galaxy.psu.edu.
3. MyExperiment. http://www.myexperiment.org.
4. Goderis, A., et al.: Seven bottlenecks to workflow reuse and repurposing. In: Proceedings of the 4th International Semantic Web Conference, Galway, Ireland, 6-10 November 2005
5. Zhao, J., et al.: Why workflows break - Understanding and combating decay in Taverna workflows, accepted for the 8th IEEE International Conference on eScience 2012
6. Aruliah, D. A., et al.: Best Practices for Scientific Computing. (Submitted on 1 Oct 2012)
7. Missier P., et al.: Functional Units: Abstractions for Web Service Annotations. In: SERVICES 2010; 05 Jul 2010-05 Jul 2010; Miami, USA. (2010)
8. Bechhofer, S., et al.: Why linked data is not enough for scientists. Future Generation Computer Systems. In Press.
9. Wf4Ever project. http://www.wf4ever-project.org.
10. Belhajjame, K., et al.: Workflow-Centric Research Objects: First Class Citizens in Scholarly Discourse. In: SEPUBLICA: Future of scholarly communication in the Semantic Web. 2012.
11. RO manager. https://github.com/wf4ever/ro-manager.
12. Jelier, R., et al.: Text-derived concept profiles support assessment of DNA microarray data for acute myeloid leukemia and for androgen receptor stimulation. BMC Bioinformatics. 8, 14 (2007)
13. Workflow pack on myExperiment. http://www.myexperiment.org/packs/282.
14. Scufl2-wfdesc. https://github.com/wf4ever/scufl2-wfdesc.
15. Taverna server. http://www.taverna.org.uk/download/server/.
16. BioCatalogue. http://www.biocatalogue.org.
17. GenomeSpace. http://www.genomespace.org.
18. Nanopub. http://nanopub.org.
19. Hettne, K., et al.: Best practices for workflows. Presented at 2nd BioVeL Workshop on taxonomic and phylogenetic workflows. Gothenburg, Sweden. 2012.
    http://www.biovel.eu/images/events/MS6WorkshopPix/presentations/hettne.ppt