Towards the Preservation of Scientific Workflows

David De Roure
Oxford e-Research Centre
University of Oxford
Oxford, UK
david.deroure@oerc.ox.ac.uk

Khalid Belhajjame
School of Computer Science
University of Manchester
Manchester, UK
khalidb@cs.man.ac.uk

Paolo Missier
School of Computing Science
Newcastle University
Newcastle upon Tyne, UK
paolo.missier@ncl.ac.uk

José Manuel Gómez-Pérez
iSOCO
Madrid, Spain
jmgomez@isoco.com

Raúl Palma
Poznań Supercomputing and Networking Center
Poznań, Poland
palma@man.poznan.pl

José Enrique Ruiz
Instituto de Astrofísica de Andalucía
Granada, Spain
jer@iaa.es

Kristina Hettne & Marco Roos
Leiden University Medical Center, Leiden, NL
{k.m.hettne,m.roos}@lumc.nl

Graham Klyne
Department of Zoology
University of Oxford
Oxford, UK
graham.klyne@zoo.ox.ac.uk

Carole Goble
School of Computer Science
University of Manchester
Manchester, UK
carole.goble@manchester.ac.uk

ABSTRACT

Some of the shared digital artefacts of digital research are *executable* in the sense that they describe an automated process which generates results. One example is the computational *scientific workflow* which is used to conduct automated data analysis, predictions and validations. We describe preservation challenges of scientific workflows, and suggest a framework to discuss the reproducibility of workflow results. We describe curation techniques that can be used to avoid the 'workflow decay' that occurs when steps of the workflow are vulnerable to external change. Our approach makes extensive use of provenance information and also considers aggregate structures called *Research Objects* as a means for promoting workflow preservation.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Data sharing; H.5.3 [**Group and Organization Interfaces**]: Collaborative computing

1. INTRODUCTION

Research is being conducted in an increasingly digital and online environment. Consequently we are seeing the emergence of new digital artefacts. In some respects these objects can be regarded as data; however some warrant particular attention, such as when the object includes a description of some part of the research method that is captured as a computational process. Processes encapsulate the knowledge related to the generation, (re)use and general transformation of data in experimental sciences. For example, an object might contain raw data, the description of a computational analysis process and the results of executing that process, thus offering the capability to reproduce and reuse the research process. Processes are key to the understanding and evolution of science; consequently as the scientific community need to curate and preserve data, so we should preserve and curate associated processes [5]. The problem, as observed by Donoho et al, is that "current computational science practice does not generate routinely verifiable knowledge" [3].

In this paper we focus on computational *scientific workflows* which are increasingly becoming part of the scholarly knowledge cycle [11]. A computational scientific workflow is a precise, executable description of a scientific procedure – a multi-step process to coordinate multiple components and tasks, like a script. Each task represents the execution of a computational process, such as running a program, submitting a query to a database, submitting a job to a computational facility, or invoking a service over the Web to use a remote resource. Data output from one task is consumed by subsequent tasks according to a predefined graph topology that orchestrates the flow of data. The components (the dataset, service, facility or code) might be local and hosted along with the workflow, or remote (public repositories hosted by third parties) [9].

Workflows have become an important tool in many areas, notably in the Life Sciences where tools like Taverna [7] are popular. From a researcher's standpoint, workflows are a transparent means for encoding an *in silico* scientific method that supports reproducible science and the sharing and replicating of best-of-practice and know-how through reuse.

However, the preservation of scientific methods in the form of computational workflows faces challenges which deal precisely with their executable aspects and their vulnerability to the volatility of the resources – data and services – required for their execution. Changes made by third parties to the workflow components may lead to a *decay* of the abil-

ity of the workflow to be executed and consequently hinder the repeatability and reproducibility of their results.

This paper highlights such challenges and states the prominent role of information quality evaluation and curation in order to diagnose and react to workflow decay. Although we draw on our specific experience with workflows, the framework in this paper is designed for a more generalised notion of executable objects which we refer to as *Research Objects* [1].

We begin by discussing the difficulties underlying scientific workflow preservation (in Sec. 2). We go on to highlight the role that Research Objects, as artefacts that bundle workflows together with other resources, can play in ensuring the preservation of scientific workflows (in Sec. 3). We close the paper by discussing our ongoing work (in Sec. 4).

2. PRESERVATION CHALLENGES

To illustrate preservation needs in scientific workflows, we use an example workflow from the field of astronomy, which is used to extract a list of companion galaxies. The workflow is illustrated in Figure 1. It starts by running two activities in parallel, the first extracts a list of companion galaxies by querying the public Virtual Observatory (VO) database, and the second activity extracts a second list of companion galaxies by invoking a web service. The results of the two activities are then cross matched to obtain an improved list of companion galaxies.

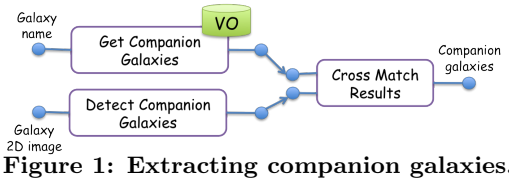

Figure 1: Extracting companion galaxies.

The content of the VO database is subject to update, and the implementation of the web service responsible for detecting companion galaxies is also subject to modifications. Thus it is possible, and likely, that the workflow produces different lists of companion galaxies when run at different times. It is important therefore to record the provenance of workflow outputs; i.e. the sources of information and processes involved in producing a particular list [12].

Should the VO database become unavailable or alter its interface so that the workflow can no longer access it, the workflow will become inoperable. This **workflow decay** is a fundamental challenge for the preservation of scientific workflows. Even though a workflow description remains unchanged, and may still have value in helping interpret results, the execution of that workflow may fail or yield different results. This is due to dependencies on resources outside the immediate context of the object which are subject to independent change. Further use cases can be found in [13] for bioinformatics and [14] for astronomy.

Gil *et al* observe that "It must be possible to re-execute workflows many years later and obtain the same results. This requirement poses challenges in terms of creating a stable layer of abstraction over a rapidly evolving infrastructure while providing the flexibility needed to address evolving requirements and applications and to support new capabilities" [4].

This abstraction approach insulates from some change, but we will still experience decay when the execution is dependent on resources and services that use independently controlled resources. For example, service providers such as the European Bioinformatics Institute (EMBL-EBI) routinely update their service offerings, and must do in response to developments in the field of life science. Resources become obsolete or are no longer sustained. Even workflows that depend on local components are still vulnerable to changes in operating systems, data management sustainability and access to computational infrastructure. We note that workflows have many of the properties of software, such as the composition of components with external dependencies, and hence some aspects of software preservation [10] are applicable. We also observe that the above requirement is actually quite stringent: it must be possible to reproduce an experiment, it is helpful if rerunning a workflow produces the same results but this is not the only way.

We need a means to i) evaluate the current status of the resources upon which the workflow depends and ii) react to any signs of diagnosed decay in order to ensure workflow execution. In the Wf4Ever project[1] we are addressing this twofold goal through the combination of techniques for computing **information quality** and, more specifically, the integrity and authenticity of the associated resources, and **curation** techniques. Foreseeing the case where actual reproducibility cannot be achieved despite such efforts, we propose **partial reproducibility** as the means required to *play back* workflow execution based on the provenance of previous executions.

2.1 Reproducibility in scientific workflows

To provide a framework for this discussion we briefly analyse, in abstract terms, the key scenarios that arise when attempting to reproduce a workflow execution. The short formalism that follows identifies four cases for consideration here and can readily be used to discuss other cases.

Let $W_{S,D}$ denote a workflow W with dependencies on a set of services S and on a data state D. A typical example would be a bioinformatics workflow that depends on a set S of EBI services, some of which provide query capabilities into some of the EBI databases. Here D represents the content of those databases. Let $exec(W_{S,D}, d, t)$ denote the execution of W on input dataset d at time t.

As noted earlier, both service specification and implementation will evolve over time (and some services may be retired), and the state of the databases will change as well. Let S' and D' denote the new service and data dependencies at some later time t' (possibly months, or years). At this time, an investigator may be interested in using W with the following goals, and corresponding concrete options:

1. *Updated workflow on original data.* To update the old outcomes using the current, updated state of services and databases (possibly, to compare with the original outcomes): $exec(W_{S',D'}, d, t')$.
2. *Updated workflow on new data.* To test the workflow in its current state on a new dataset: $exec(W_{S',D'}, d', t')$.
3. *Original workflow on new data.* To replicate the original experiment on a new dataset d': $exec(W_{S,D}, d', t')$.
4. *Original workflow on original data.* To confirm earlier claims on the original outcomes. This translates into $exec(W_{S,D}, d, t')$, i.e., the same input d is used on W's original configuration;

[1] http://www.wf4ever-project.org

Different issues arise in each of these four cases. Cases (1) and (2) highlight workflow decay, primarily due to the evolution $S \rightarrow S'$. This is a difficult problem, which involves the evaluation of the integrity and authenticity of S and D as they evolve into S' and D' respectively and some form of ongoing curation of W in order to make it compatible with S'. We describe three approaches to curation below (Sec. 3.1), amongst which the first two have been investigated in the context of the myExperiment workflow repository [2]. For information quality, we propose provenance as an important type of evidence that can support the detection of workflow decay with respect to external resources S and D (Sec. 2.3). By providing scientists with such provenance-enabled diagnosis, we aim at feeding curation systems with accurate information of what is causing workflow decay, how and why.

Cases (3) and (4) are increasingly relevant in e-Science, as they are paradigmatic of the emerging *executable publications* [8] scenario. In an executable paper, some of the quantitative results (tables, charts) that appear in the publication are not statically part of the text, but dynamically linked to the process that produced them. In our case, the results $exec(W_{S,D}, d, t')$ are published in the paper, but they are also linked to $W_{S,D}$ as well as the input d. The intent of this emerging form of "active publication" is precisely to let readers replicate, entirely or in part, the computational portion of an experiment in order to reproduce its results. For example, Koop *et al.* [8] proposed a method that automatically captures provenance information of the experiments in order to assist authors by integrating and updating experiment results into the paper as they write it.

Supporting this scenario is not simple as it requires the entire set of original resources S and D, to be available at time t', along with the guarantee that a suitable runtime environment can be provided for the services, as well as any other software component in S. Although approaches based on Virtual Machines (VM) are common in this case [6], the high volume of state data, along with third party services that cannot be replicated locally, and the potentially high cost of execution for computationally expensive workflows, may make this approach infeasible. For example, modeling 3D data of galaxies [14] involves the manipulation of large data cubes, the size of which may reach tens of TB. *Partial reproducibility* alleviates the problem in practical cases.

2.2 Partial Reproducibility

Consider the astronomy workflow presented in Figure 1. For (3) and (4), insisting on executing W in its original environment is not always feasible and may not be needed. An executable paper may for example provide limited workflow execution capability to readers, permitting only execution of lightweight tasks, such as analysis and charting of tabular data, as opposed to compute-intensive simulations, for example. This corresponds to splitting the workflow into two portions (top/bottom), where only the latter is made available for readers to experiment with, while they will still have to rely upon the usual peer-review guarantees regarding the correctness of the top portion of the workflow.

Executing W is unnecessary provided that a complete and reliable provenance trace has been recorded at time t. By combining provenance traces with partitioned executable workflow fragments, provenance can be used to "play back" the original execution and be queried to inspect all data dependencies that resulted from that execution: *(i)* the provenance is recorded from the execution of segments that are heavily dependent on S and D, which are then omitted, and *(ii)* a VM approach is used for the remaining segments, which are executable. Partitioning requires that the executable segments be found downstream (in terms of the directed graph that represents the workflow structure) from the omitted segments. This places a requirement on workflow design. Minimising the associated *cost* to reproducibility of a workflow, which involves S, D, and the actual cost of execution (which may well be a monetary cost, for example in the case of cloud-based computations) presents the challenge of finding the optimal partitioning. These are just two of the challenges arising from taking this pragmatic approach.

2.3 Information quality evaluation

In order to detect workflow decay with respect to the evolution of the tuple (S,D) of services and data needed for workflow execution, we focus on two main aspects relevant for information quality: integrity and authenticity. *Integrity* refers to the quality or condition of being whole, complete and unaltered while *authenticity* aims at the lineage of data.

One of the main sources required to evaluate information quality is provenance information (in our case about S and D) which offers the means to verify the evolution of data and services, to analyse the processes that led to their current status, and to decide whether they are still consistent with a given workflow. We build on provenance to compute the integrity and authenticity of workflows with respect to (S,D), thus providing scientists with accurate information about what is causing the workflow decay due to changes in such resources, how and why.

We can use and extend existing provenance vocabularies like the Open Provenance Model[2] to record and reason about provenance metadata relevant to the diagnosis of workflow decay. Additional challenges include providing scientists with the means to interpret easily the results of such analysis and to assist them in the early diagnosis of workflow decay and the selection of the most appropriate curation techniques.

3. PRESERVING WORKFLOWS USING RESEARCH OBJECTS

3.1 Preservation in Practice

The myExperiment[3] social website for finding, storing and sharing workflows has been in operation since 2007 and holds the largest public collection of scientific workflows [2]. As such it provides a useful case study in workflow decay and preservation, supporting two main mechanisms.

First, the continual downloading and uploading of workflows provides a *community curation* mechanism for workflows that are reused, and these in turn can act as examples to inform people updating other workflows. Expert curators, e.g. scientists, are involved in annotating workflows, by tagging and providing exemplars.

The second mechanism is *assistive curation* using semi-automated processes to perform 'housekeeping' on the corpus of workflows. For example, when a service provider announces that a service is deprecated and will be removed or

[2]http://openprovenance.org
[3]http://www.myexperiment.org

replaced on a certain date, the workflows affected by this can be tagged accordingly and replacement advice propagated to the appropriate users. Potentially this could progress to *autonomic curation* where workflows could be executed and repaired automatically, for example when services change.

The assistive approach keeps the 'human in the loop' and the wf4ever project is pursuing this approach by focusing on *recommendations* for curation and repair; for example a replacement for a service can be confirmed using provenance logs.

3.2 Research Objects

Workflow specifications are insufficient for guaranteeing the preservation of scientific workflows. The reproducibility strategies listed in Sec. 2.1 show that, in addition to workflow specification, we need information about the components that implement workflow steps, the data used and produced as a result of workflow enactment.

In practice myExperiment users sometimes choose to aggregate workflows with associated data (in 'packs') and this provides a powerful means to track S and D. Building on packs, to cater for workflow preservation we use the notion of a *Research Object*, which can be viewed as an aggregation of resources that bundles workflow specification and additional auxiliary resources. These may include input and output data which enables workflows to be validated.

The elements that compose a Research Object may differ from one to another, and this difference may have consequences on the *level* of reproducibility that can be guaranteed. At one end of the spectrum, the Research Object is represented by a paper. As we progress to the other end the Research Object is enriched to include elements such as the workflow implementing the computation, annotations describing the experiment implemented and the hypothesis investigated, and provenance traces of past executions of the workflow. Assessing the reproducibility of computations described using electronic papers can be tedious: a paper may just sketch the method implemented by the computation in question, without delving into details that are necessary to check that the results obtained, or claimed, in the paper can be reproduced. Verifying the reproducibility of Research Objects at the other end of the spectrum is less difficult. The provenance trace provides data examples to re-enact the workflow and a means to verify that the results of workflow executions are comparable with prior results.

To ensure the preservation of a workflow and the reproducibility of its results, the Research Object needs to be managed and curated throughout the lifecycle of the associated workflow. The provenance of the Research Object elements (i.e., workflow, data sets and web services) is key to understanding, comparing and debugging scientific workflows and to verifying the validity of a claim made within the context of a Research Object by revealing the data inputs used to yield a given workflow result. We need to support the logging, browsing and querying of the provenance linking components of Research Objects and the traces of workflow executions.

4. CONCLUSIONS

As research practice evolves we anticipate a growing quantity and diversity of executable objects, in particular computational scientific workflows. We outlined the challenges underlying the preservation of scientific workflows and sketched preliminary solutions that can be adopted for that purpose. We used the concept of Research Object as an abstraction for the management of executable objects throughout their life-cycle. We anticipate that this work will give rise to recommendations and best practices for authors and curators of scientific workflows to meet preservation requirements.

We are investigating the reproducibility and curation strategies reported in this paper and developing a software architecture and reference implementation for workflow preservation. The development of the reference implementation will rest on existing developments in scientific workflow repositories, digital libraries and preservation systems. In particular, we will build on well-established digital libraries, such as dLibra[4], to extend the myExperiment workflow repository with further preservation capabilities.

5. ACKNOWLEDGMENTS

Wf4Ever is funded by the Seventh Framework Programme of the European Commission (Digital Libraries and Digital Preservation area ICT-2009.4.1 project reference 270192). myExperiment is funded by UK JISC. The dLibra Digital Library Framework has been produced by the Poznań Supercomputing and Networking Center since 1999. We are grateful to all our collaborators in these projects.

6. REFERENCES

[1] S. Bechhofer, J. Ainsworth, et al. Why linked data is not enough for scientists. In *IEEE Sixth International Conference on e-Science*, pages 300–307, 2010.

[2] D. De Roure, C. Goble, and R. Stevens. The design and realisation of the myExperiment virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561–567, 2009.

[3] D. L. Donoho, A. Maleki, I. Rahman, et al. Reproducible research in computational harmonic analysis. *Computing in Science and Engg.*, 11:8–18, January 2009.

[4] Y. Gil, E. Deelman, et al. Examining the challenges of scientific workflows. *IEEE Computer*, 40:24–32, Dec. 2007.

[5] C. Goble and D. De Roure. Curating scientific web services and workflows. *Educause Review*, 43(5), 2008.

[6] P. J. Guo and D. Engler. CDE: Using System Call Interposition to Automatically Create Portable Software Packages. In *Proc. USENIX Annual Tech. Conf.*, 2011.

[7] D. Hull, K. Wolstencroft, R. Stevens, et al. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(suppl 2):W729–W732, 1 July 2006.

[8] D. Koop et al. A provenance-based infrastructure to support the life cycle of executable papers. *Procedia Computer Science*, 4:648 – 657, 2011. Proceedings of the International Conference on Computational Science.

[9] B. Ludäscher et al. Scientific process automation and workflow management. In *Scientific Data Management*, Computational Science Series. Chapman & Hall, 2009.

[10] B. Matthews et al. A framework for software preservation. *International Journal of Digital Curation*, 5(1), 2010.

[11] J. P. Mesirov. Accessible reproducible research. *Science*, 327(5964):415–416, 2010.

[12] P. Missier. *Modelling and computing the quality of information in e-science*. PhD thesis, University of Manchester, 2008.

[13] M. Roos. Genomics Workflow Preservation Requirements. Technical report, Deliverable D6.1, Wf4Ever project, 2011.

[14] L. Verdes-Montenegro. Astronomy Workflow Preservation Requirements. Technical report, Deliverable 5.1, Wf4Ever project, 2011.

[4] http://dlibra.psnc.pl