

# Capturing interoperable reproducible workflows with Common Workflow Language

## Identifier

<https://doi.org/10.5281/zenodo.1312623>

## Date created


2018-07-16

## Submitted to

Workshop on Research Objects (RO2018)

## Authors


**Stian Soiland-Reyes** The University of Manchester, UK; Common Workflow Language project; Apache Taverna (incubating)

 <https://orcid.org/0000-0001-9842-9718>

**Farah Zaib Khan** The University of Melbourne, Australia; Common Workflow Language project

 <https://orcid.org/0000-0002-6337-3037>

**Richard O. Sinnott** The University of Melbourne, Australia

 <http://orcid.org/0000-0001-5998-222X>

**Andrew Lonie** The University of Melbourne, Australia

 <https://orcid.org/0000-0002-2006-3856>

**Michael R Crusoe** Common Workflow Language project

 <https://orcid.org/0000-0002-2961-9670>

**Carole Goble** The University of Manchester, UK

 <https://orcid.org/0000-0003-1219-2137>

## Abstract

We present our ongoing work on integrating Research Object practices with Common Workflow Language, capturing and describing prospective and retrospective provenance.

## Background

Performing computational data analysis using scientific workflows is common in many domains, including bioinformatics, physics, earth sciences and cheminformatics [1]; and it is now considered best practice to use workflows or automated scripts to improve reproducibility of analyses [2]. Recent advances in software containers and packaging help remove previous challenges in capturing software tools needed by an analysis [3,4], but other interoperability factors remain important, such as ensuring a pipeline definition can execute in multiple workflow systems [5], and that provenance of workflow runs can be reliably captured and used across heterogeneous execution environments [6]. While robust reproducibility of results is important, the outcomes and methods also need to be *replicable* and *generalizable* [7].

[Common Workflow Language](#) (CWL) [8] is a multivendor community-led standardization effort to define a dataflow language that is [implemented](#) by multiple workflow systems capable of executing on a wide range of compute platforms. CWL focus primarily on what has been determined as a common denominator across bioinformatics pipeline systems; coordinating and parallelizing command line tools that exchange files. While CWL started in the bioinformatics community, usage and adoption has spread beyond the life sciences to radio astronomy, hydrology, digital humanities, and more.

CWL has seen considerable interest over recent years [9], in particular primed for the role of computational interoperability, as FAIR sharing of data [10] should also include the workflows that produced it [11].

## Prospective provenance — capturing the workflow template

We created [CWL Viewer](#) [12] for visualizing CWL workflow definitions, and it has now more than 2500 [CWL workflows registered](#). Most of these are developed as open source in GitHub repositories. While continual active development of workflows is good for reuse and stability, it adds a challenge for reproducibility and reliable workflow citation. CWL workflows are typically split in multiple files (e.g. one per command line tool), and may contain internal references used during execution, such as secondary data files. Although CWL encourages interoperability, it has not yet defined a standard mechanism to reliably capture or transfer the complete definition required for starting a workflow run. A given CWL workflow and its components can be [richly annotated](#) using vocabularies like [EDAM](#) [13] and [schema.org](#) to indicate formats, contributors, licenses and references, but, while this reuse follows Linked Data principles, the metadata is embedded within the CWL definitions, requiring [Schema Salad](#) preprocessing before further use.

To this end we have extended the CWL Viewer to produce [permalinks](#) based on the corresponding git commit, adding [content-negotiation](#) to other formats (SVG, RDF Turtle, JSON-LD) with archiving as Research Objects [14] that include snapshots of the CWL files, their dependent resources and extracted metadata augmented with authorship information extracted from the *git* log. This work aims to capture the *prospective provenance* of a workflow definition; the metadata that should be true for every execution.

To complete this picture we are working with [Open AIRE](#) to make CWL workflows Findable, and plan to calculate prospective execution details such as linking Docker images [15] and BioConda packages [16] with [SciCrunch RRDs](#) [17] and [ELIXIR's Tools and Data Services](#) registry [18] using CWL [SoftwarePackage annotations](#).

## Retrospective provenance — how was the workflow executed?

In terms of *retrospective provenance*, capturing what happened in a particular execution of a CWL workflow, we have been extending our earlier work on capturing workflow runs as research objects [14]. Previous work with [TavernaProv](#) [19] and [Wings](#) [20] both used the distinction of *prospective* vocabularies with [wfdesc](#) [14] or [P-Plan](#) [21] of the workflow template, and *retrospective* PROV statements with [wfprov](#) [14] or [OPMW](#) [22] of the instantiated workflow execution; thus allowing queries and views of the abstracted workflow independent of its execution, however computational re-execution or reuse of the workflows were effectively still restricted to their original workflow engines.

CWL is implemented by multiple workflow engines, with different execution characteristics (e.g. distributed on cloud), but with an explicit and interoperable definition of what is to be run. Therefore the aim was to define a more specific Research Object profile for workflow execution that should enable more precise replication of the workflow run, but at the same time being general enough to be used by multiple engines.

For this we created [CWLProv](#) [23], developed by modifying the CWL reference implementation [cwltool](#) to add a new [--provenance flag](#). The resulting profile describes how the existing Linked Data standards for Research Object, [wfdesc](#), [wfprov](#) [14], [PROV](#) [24] and [Web Annotation Data Model](#) [25] are combined to track detailed workflow execution logs.

Unlike TavernaProv, which used [Research Object Bundle](#) ZIP files [26], CWLProv uses *BagIt* for archiving the files of the RO. BagIt [27] is a Library of Congress supported digital preservation format, that can exist either as a flat file hierarchy or packaged as tar/ZIP archives. BagIt support external references and checksums of all files, but its metadata provisions are fairly minimal.

Complementing the two approaches we use the [BDBag](#) profile for [Research Object in BagIt](#), as it has previously been shown to support data from large-scale workflows [28,29] and is also of consideration by [NIH](#)

[Data Commons](#) and [Research Data Alliance](#) [30].

The [PROV files captured by CWLProv](#) [23] reuse the wfdesc and wfprov vocabularies, but compared to TavernaProv stay closer to plain PROV statements. In cwltool we used the [Prov Python](#) library and could thus save in multiple serializations, including PROV-N, PROV-XML, PROV-JSON as well as PROV-O based RDF as RDF Turtle, N-Triples and JSON-LD. It is still under investigation which of these formats will be most beneficial for consumers and producers, thus CWLProv only mandate PROV-N for now. CWLProv uses different *provenance levels* to avoid complicating coarse-grained provenance with the details of fine-grained execution traces.

The CWLProv approach is also the basis for Research Object support [being developed](#) for the workflow system Nextflow [31], but there without requiring an executable CWL workflow. One interesting collaboration aspect we are exploring is to identify common approaches not previously described for Workflow Research Objects, such as capturing execution of Docker images or referencing large data files.

## References

- [1] M. Atkinson, S. Gesing, J. Montagnat, I. Taylor, Scientific workflows: Past, present and future, *Future Generation Computer Systems*. 75 (2017) 216–227. <https://doi.org/10.1016/j.future.2017.05.041>
- [2] G.K. Sandve, A. Nekrutenko, J. Taylor, E. Hovig, Ten simple rules for reproducible computational research., *PLoS Comput. Biol.* 9 (2013) e1003285. <https://doi.org/10.1371/journal.pcbi.1003285>
- [3] S. Möller, S.W. Prescott, L. Wirzenius, P. Reinholdtsen, B. Chapman, P. Prins, et al., Robust Cross-Platform Workflows: How Technical and Scientific Communities Collaborate to Develop, Test and Share Best Practices for Data Analysis, *Data Sci. Eng.* 2 (2017) 232–244. <https://doi.org/10.1007/s41019-017-0050-4>
- [4] B. Grüning, J. Chilton, J. Köster, R. Dale, N. Soranzo, M. van den Beek, et al., Practical computational reproducibility in the life sciences., *Cell Syst.* 6 (2018) 631–635. <https://doi.org/10.1016/j.cels.2018.03.014>
- [5] S.D.I. Fernando, D.A. Creager, A.C. Simpson, Towards Build-Time Interoperability of Workflow Definition Languages, in: Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2007), IEEE, 2007: pp. 525–532. <https://doi.org/10.1109/SYNASC.2007.18>
- [6] D. Garijo, Y. Gil, O. Corcho, Abstract, link, publish, exploit: An end to end framework for workflow sharing, *Future Generation Computer Systems*. 75 (2017) 271–283. <https://doi.org/10.1016/j.future.2017.01.008>
- [7] P.D. Schloss, Identifying and overcoming threats to reproducibility, replicability, robustness, and generalizability in microbiome research., *MBio.* 9 (2018). <https://doi.org/10.1128/mBio.00525-18>
- [8] P. Amstutz, M.R. Crusoe, Nebojša Tijanić, B. Chapman, J. Chilton, M. Heuer, et al., Common Workflow Language, v1.0, (2016). <https://w3id.org/cwl/v1.0> <https://doi.org/10.6084/m9.figshare.3115156.v2>
- [9] J. Leipzig, A review of bioinformatic pipeline frameworks., *Brief. Bioinformatics.* 18 (2017) 530–536. <https://doi.org/10.1093/bib/bbw020>
- [10] M.D. Wilkinson, M. Dumontier, I.J.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, et al., The FAIR Guiding Principles for scientific data management and stewardship., *Sci. Data.* 3 (2016) 160018. <https://doi.org/10.1038/sdata.2016.18>
- [11] S. Cohen-Boulakia, K. Belhajjame, O. Collin, J. Chopard, C. Froidevaux, A. Gaignard, et al., Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities, *Future Generation Computer Systems*. 75 (2017) 284–298. <https://doi.org/10.1016/j.future.2017.01.012>
- [12] M. Robinson, S. Soiland-Reyes, M.R. Crusoe, C. Goble, CWL Viewer: the Common Workflow Language viewer, *F1000Research*. 6 (2017) 1075 (poster). <https://doi.org/10.7490/f1000research.1114375.1>
- [13] J. Ison, M. Kalas, I. Jonassen, D. Bolser, M. Uludag, H. McWilliam, et al., EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats., *Bioinformatics.* 29 (2013) 1325–1332. <https://doi.org/10.1093/bioinformatics/btt113>
- [14] K. Belhajjame, J. Zhao, D. Garijo, M. Gamble, K. Hettne, R. Palma, et al., Using a suite of ontologies for preserving workflow-centric research objects, *Web Semantics: Science, Services and Agents on the World Wide Web.* 32 (2015) 16–42. <https://doi.org/10.1016/j.websem.2015.01.003>
- [15] B.D. O'Connor, D. Yuen, V. Chung, A.G. Duncan, X.K. Liu, J. Patricia, et al., The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows. [version 1; referees: 2 approved], *F1000Res.* 6 (2017) 52. <https://doi.org/10.12688/f1000research.10137.1>
- [16] The Bioconda Team, B. Grüning, R. Dale, A. Sjödin, B.A. Chapman, J. Rowe, et al., Bioconda: sustainable and comprehensive software distribution for the life sciences, *Nat. Methods.* (2018). <https://doi.org/10.1038/s41592-018-0046-7>

- [17] G. Jeffrey, B. Anita, B. Davis, C. Christopher, G. Amarnath, L. Stephen, et al., SciCrunch: A cooperative and collaborative data and resource discovery platform for scientific communities, *Front. Neuroinformatics*. 8 (2014). <https://doi.org/10.3389/conf.fninf.2014.18.00069>
- [18] K.-H. Hillion, I. Kuzmin, A. Khodak, E. Rasche, M. Crusoe, H. Peterson, et al., Using bio.tools to generate and annotate workbench tool descriptions. [version 1; referees: 4 approved], *F1000Res*. 6 (2017). <https://doi.org/10.12688/f1000research.12974.1>
- [19] S. Soiland-Reyes, P. Alper, C. Goble, Tracking workflow execution with TavernaProv, at PROV: Three Years Later, Provenance Week 2016. <https://doi.org/10.5281/zenodo.51314>
- [20] D. Garijo, Y. Gil, O. Corcho, Towards Workflow Ecosystems through Semantic and Standard Representations, in: 2014 9th Workshop on Workflows in Support of Large-Scale Science, IEEE, 2014: pp. 94–104. <https://10.1109/WORKS.2014.13>
- [21] D. Garijo, Y. Gil, Augmenting PROV with Plans in P-PLAN: Scientific Processes as Linked Data, in: Proceedings of the Second International Workshop on Linked Science 2012 - Tackling Big Data, 2012. CEUR Workshop Proceedings 951. <http://ceur-ws.org/Vol-951/paper6.pdf>
- [22] 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 - WORKS '11, ACM Press, New York, New York, USA, 2011: p. 47. <https://doi.org/10.1145/2110497.2110504>
- [23] F.Z. Khan, S. Soiland-Reyes, M.R. Crusoe, A. Lonie, R. Sinnott, CWLProv - Interoperable Retrospective Provenance capture and its challenges, Zenodo preprint, 2018. <https://doi.org/10.5281/zenodo.1215611>
- [24] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, et al., PROV-O: The PROV Ontology., W3C Recommendation 30 April 2013, World Wide Web Consortium. <http://www.w3.org/TR/2013/REC-prov-o-20130430/>
- [25] R. Sanderson, P. Ciccarese, B. Young, Web Annotation Data Model, W3C Recommendation 23 February 2017, World Wide Web Consortium. <https://www.w3.org/TR/2017/REC-annotation-model-20170223/>
- [26] S. Soiland-Reyes, M. Gamble, R. Haines, Research Object Bundle 1.0, researchobject.org specification, 2014. <https://w3id.org/bundle/2014-11-05/> <https://doi.org/10.5281/zenodo.12586>
- [27] J.A. Kunze, J. Littman, L. Madden, J. Scancelli, C. Adams, The BagIt File Packaging Format (V1.0), Internet Engineering Task Force, 2018. <https://tools.ietf.org/html/draft-kunze-bagit-16>
- [28] K. Chard, M. D'Arcy, B. Heavner, I. Foster, C. Kesselman, R. Madduri, et al., I'll take that to go: Big data bags and minimal identifiers for exchange of large, complex datasets, in: 2016 IEEE International Conference on Big Data (Big Data), IEEE, 2016: pp. 319–328. <https://doi.org/10.1109/BigData.2016.7840618>
- [29] R.K. Madduri, K. Chard, M. D'Arcy, S.C. Jung, A. Rodriguez, D. Sulakhe, et al., Reproducible big data science: A case study in continuous FAIRness, *BioRxiv preprint*. (2018). <https://doi.org/10.1101/268755>
- [30] Research Data Repository Interoperability WG, Research Data Repository Interoperability WG Final Recommendations, Research Data Alliance, 2018. <https://doi.org/10.15497/RDA00025>
- [31] P. Di Tommaso, M. Chatzou, E.W. Floden, P.P. Barja, E. Palumbo, C. Notredame, Nextflow enables reproducible computational workflows., *Nat. Biotechnol.* 35 (2017) 316–319. <https://doi.org/10.1038/nbt.3820>

## Acknowledgements

FZK funded by Melbourne International Research Scholarship (MIRS) and Melbourne International Fee Remission Scholarship (MIFRS). SSR funded by [BioExcel CoE](#), a project funded by the European Union contract [H2020-EINFRA-2015-1-675728](#).