Converting Scripts into Reproducible Workflow Research Objects

L. A. M. C. Carvalho*, K. Belhajjame**, C. B. Medeiros*
* University of Campinas (UNICAMP) , ** University of Dauphine
lucas.carvalho@ic.unicamp.br

Problem

Understand the processing units and data dependencies of script-based experiments, reproduce these experiments and reuse their model and data.

Objectives of this Research

To support the process of conversion of script-based experiments into a reusable and reproducible workflow-based representation.

Results

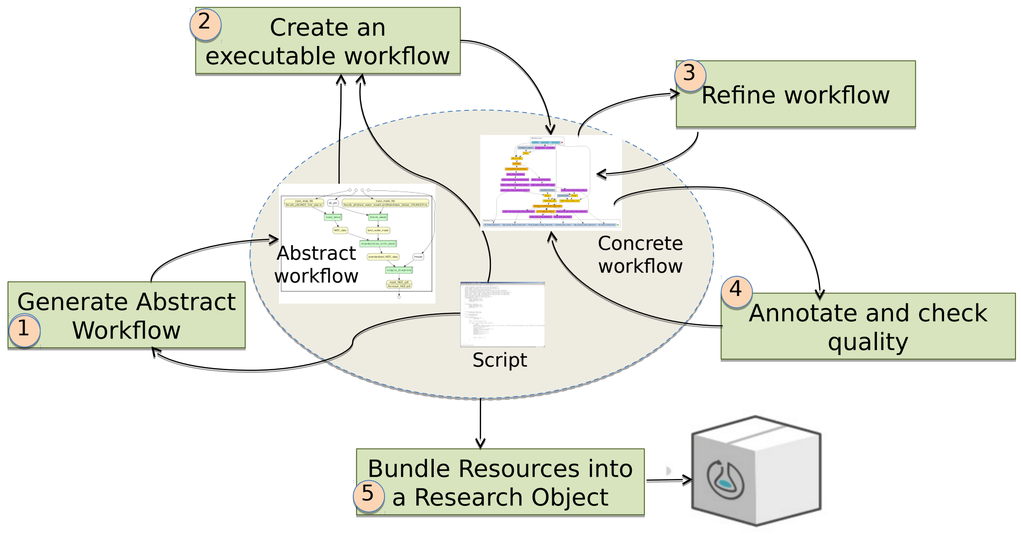
Methodology to guide curators through (a) conversion from script code to an executable workflow, and (b) construction of a Workflow Research Object that bundles the workflow plus resources needed to reproduce the experiment.

Requirements

The methodology was based on requirements elicited given our experience and collaboration with scientists who use script-based experiments.

1. Produce workflow-like view of the script.
2. Create executable workflow and compare execution of workflow and script.
3. Modify of the workflow resources.
4. Record provenance data.
5. Aggregate all resources to support Reproducibility and Reuse.

Methodology



Running Example: Molecular Dynamics

Step 1 Manually annotate the script code using YesWorkflow tags to generate the abstract workflow.

```
14  # @BEGIN split
15  # @IN initial_structure @URI file:structure.pdb
16  # @IN directory_path @AS directory
17  # @OUT protein_pdb  @URI file:{directory}/protein.pdb
18  # @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
19  # @OUT water_pdb  @URI file:{directory}/water.pdb
20  structure = $directory_path"/structure.pdb"
21  protein = $directory_path"/protein.pdb"
22  water = $directory_path"/water.pdb"
23  bglc = $directory_path"/bglc.pdb"
24  egrep -v '(TIP3|BGLC)' $structure > $protein
25  grep TIP3 $structure > $water
26  grep BGLC $structure > $bglc
27  # @END split
```

YesWorkflow tags

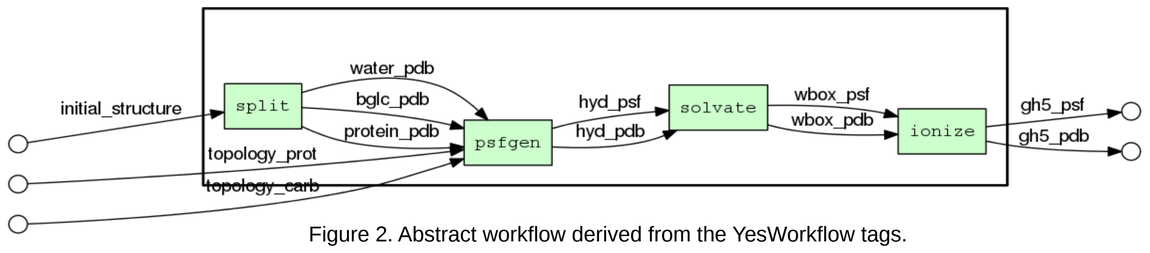


Figure 2. Abstract workflow derived from the YesWorkflow tags.

Step 2 Manually implement the executable workflow based on the abstract workflow and reusing the corresponding script block for each activity.

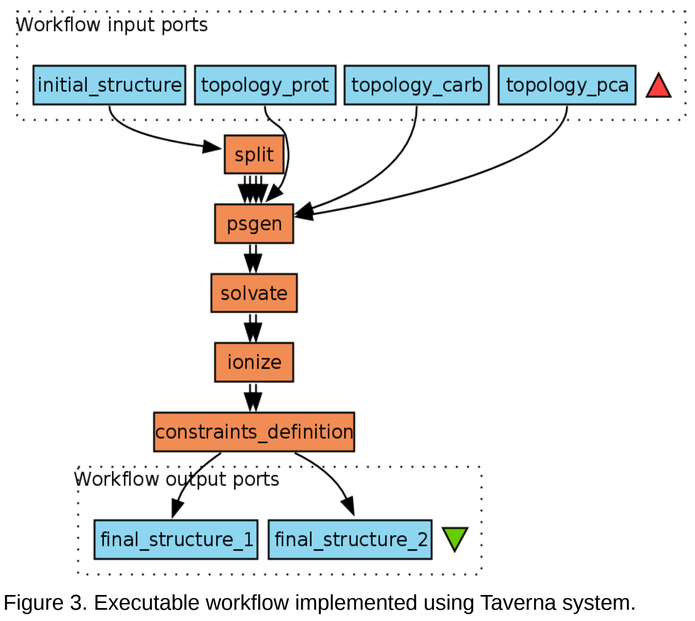


Figure 3. Executable workflow implemented using Taverna system.

Step 3 Change some resources (e.g. data set, algorithms) from the initial executable workflow to reproduce the experiment.

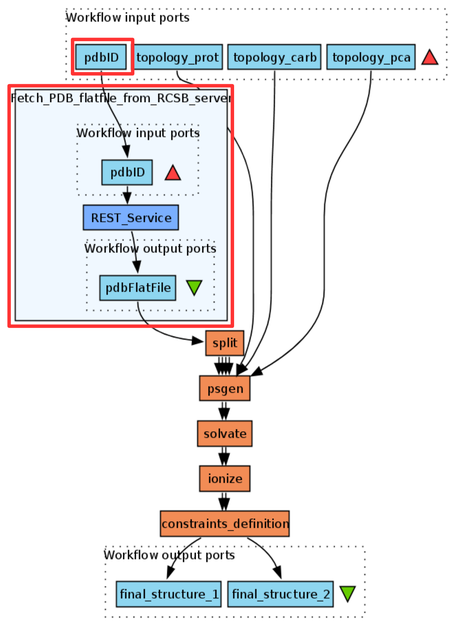


Figure 4. New version of the executable workflow.

In steps 2 and 3, provenance data is recorded wrt the execution traces and the conversion process applied to the script during these steps.

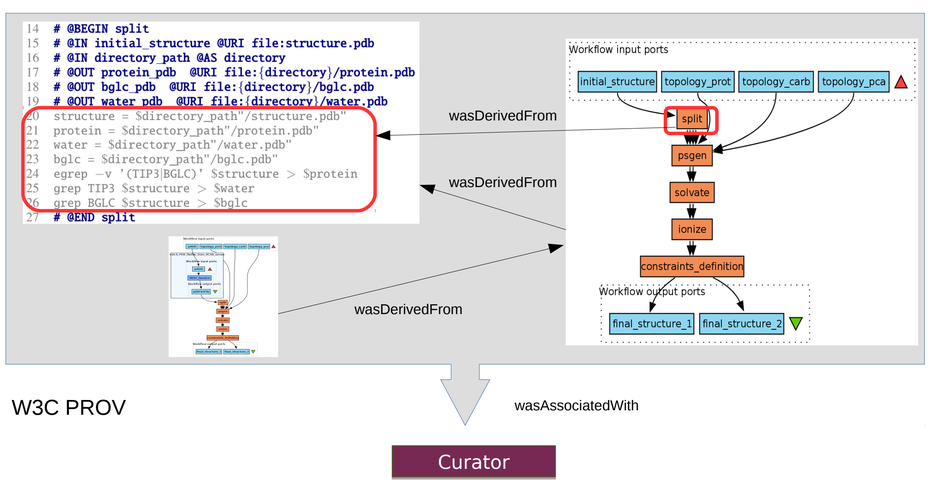


Figure 5. Provenance data describing the conversion process of the script.

Step 4 In steps 1, 2 and 3 check the quality of the conversion and provide annotations to describe the workflow.

- Use provenance data
  - Check the quality of the conversion process.
  - Check the Reproducibility of the conversion.
- Run checks to verify the soundness of the workflow.

Step 5 Aggregate all resources required to reproduce the experiment into a bundle called Workflow Research Objects.

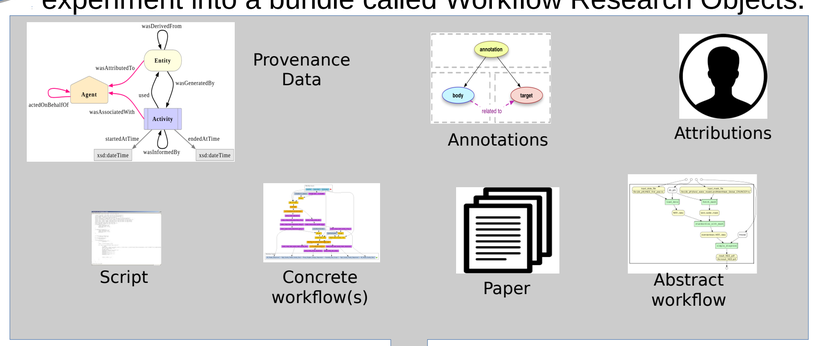


Figure 6. Workflow Research Objects

WRO available at http://w3id.orgw2share/s2wro/

Acknowledgments: Work partially financed by FAPESP (2014/23861-4) and FAPESP/CEPID CCES (2013/08293-7).