Converting Scripts into Reproducible Workflow Research Objects

Lucas A. M. C. Carvalho, Khalid Belhajjame, Claudia Bauzer Medeiros

lucas.carvalho@ic.unicamp.br

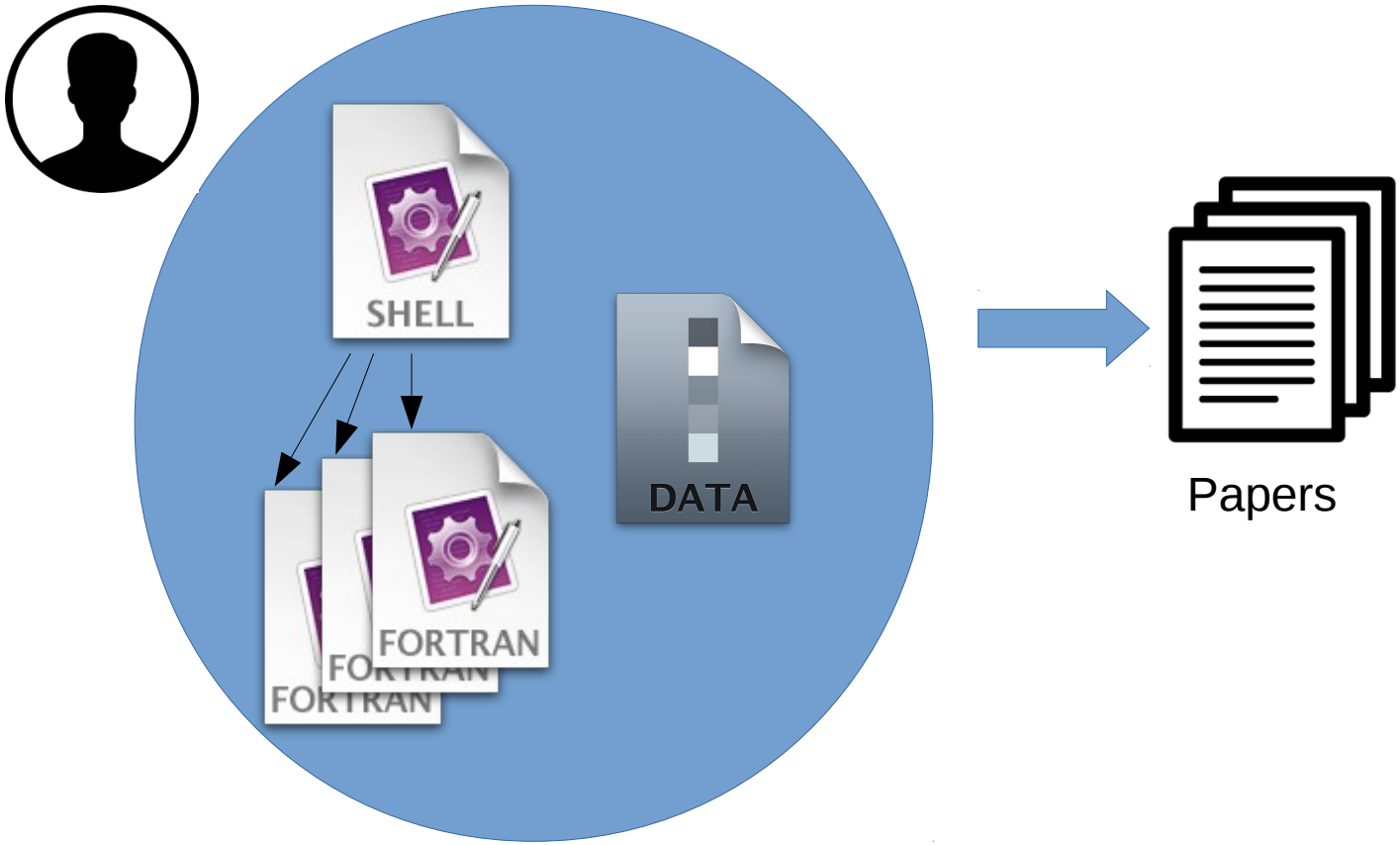
12th eScience 2016
IEEE INTERNATIONAL CONFERENCE

Baltimore, Maryland, USA

October 23-26, 2016

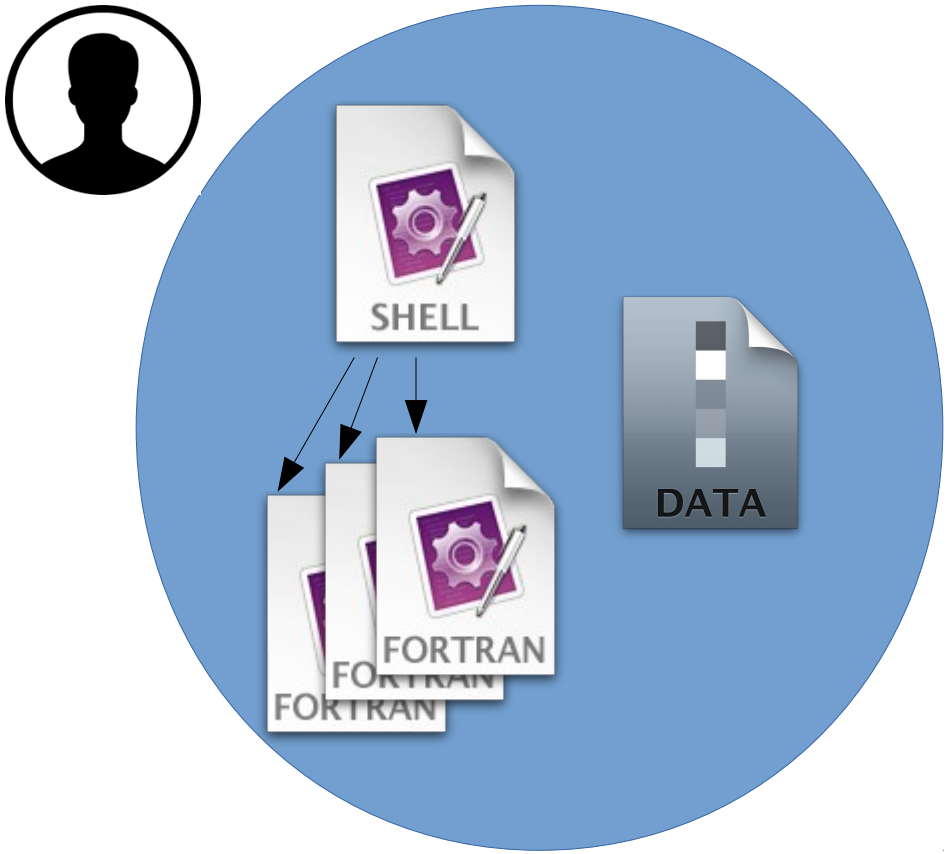
Background and Motivation

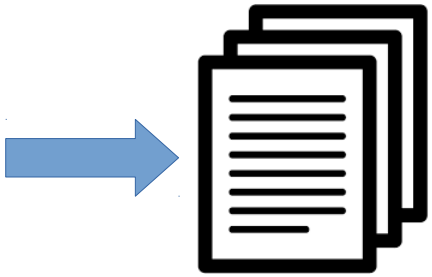
- Data-Intensive Experiments
  - Collection of scripts, programs and (big) data



Background and Motivation

- Data-Intensive Experiments
  - Collection of scripts, programs and (big) data



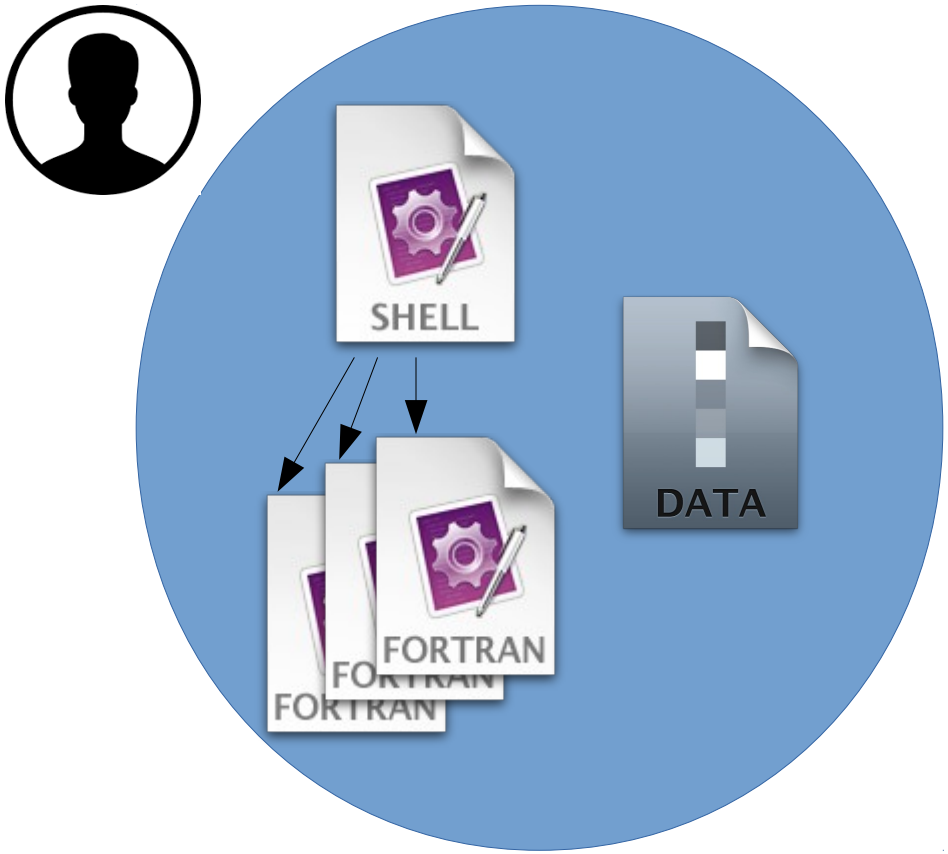


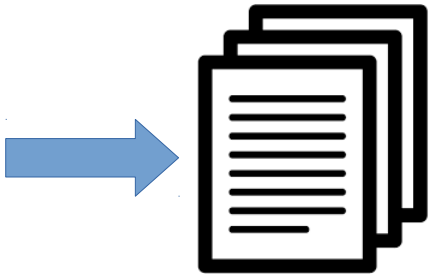
Papers

How to **understand, reproduce or reuse** data and models of experiments?

Background and Motivation

- Data-Intensive Experiments
  - Collection of scripts, programs and (big) data





Papers

How to **understand, reproduce or reuse** data and models of experiments?

Manual collection and organization of data provenance

Background and Motivation

- Script-based experiments

```
# Split input pdb into segments
grep -v HOH pccel45a.pdb > protein.pdb
grep HOH  pccel45a.pdb > water.pdb

psfgen << ENDMOL

# Read topology file
topology ../toppar/top_all22_prot.rtf

# Build protein segment
pdbalias atom ILE CD1 CD
segment GH45 {
  pdb protein.pdb
}
patch GLUP GH45:124
patch GLUP GH45:146
patch GLUP GH45:169
patch ASPP GH45:121
patch DISU GH45:165 GH45:179
patch DISU GH45:61  GH45:33
patch DISU GH45:28  GH45:123
patch DISU GH45:149 GH45:64
patch DISU GH45:103 GH45:94
regenerate angles dihedrals
coordpdb protein.pdb GH45

# Build structural waters segment
pdbalias residue HOH TIP3
pdbalias atom HOH O OH2
segment H2O {
        auto none
        pdb water.pdb}
coordpdb water.pdb H2O

# Guess missing coordinates
guesscoord

# Write structure and coordinate files
```

Example of script code.

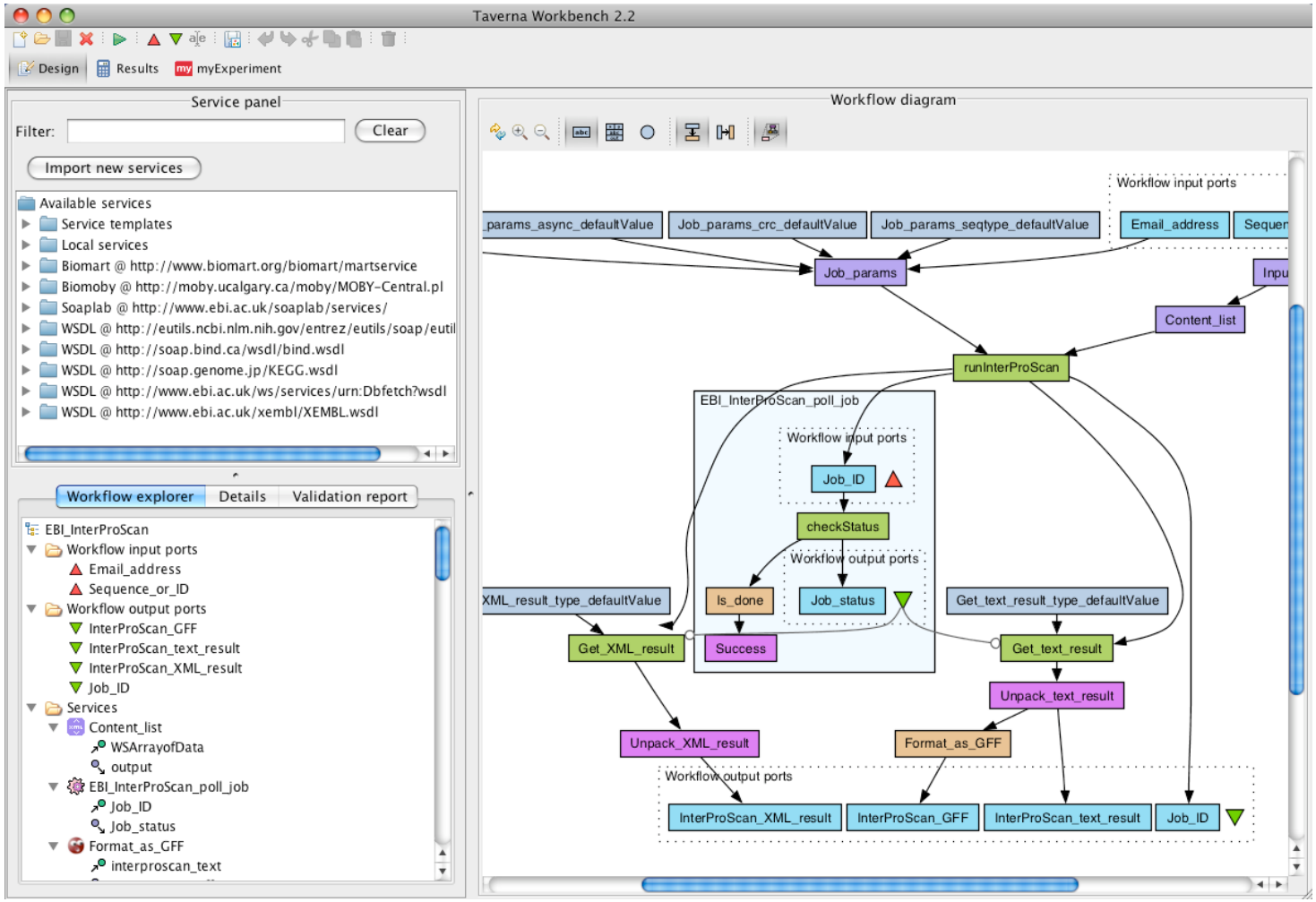
What are the inputs and outputs?

How to change this local program for a similar web service?

Difficult to understand, to reuse, and to reproduce.

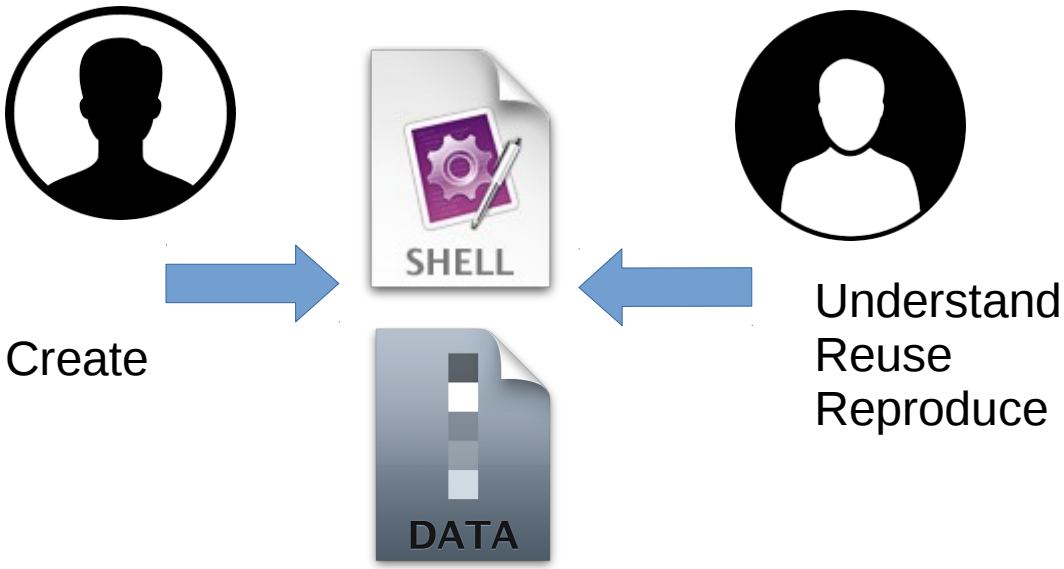
Background and Motivation

- Scientific Workflows



Example of Scientific Workflow Management System.

Overview

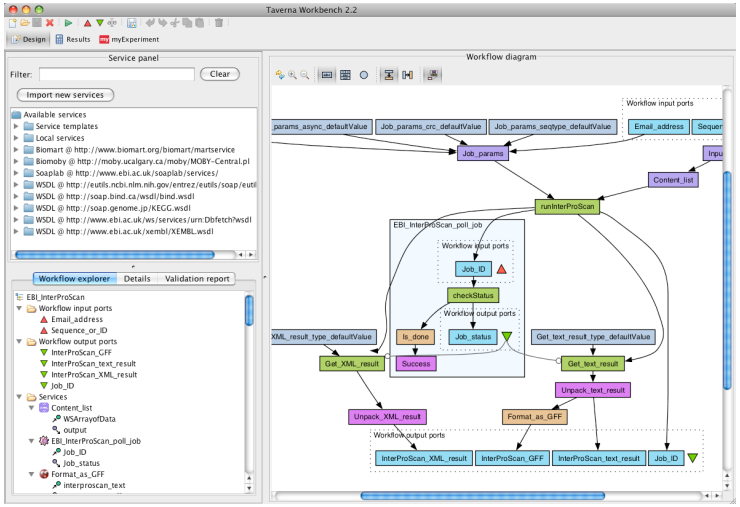


Overview

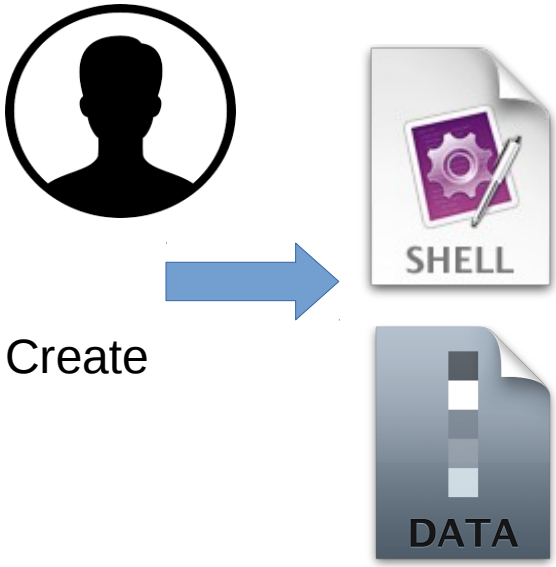
Create

Understand
Reuse
Reproduce

+



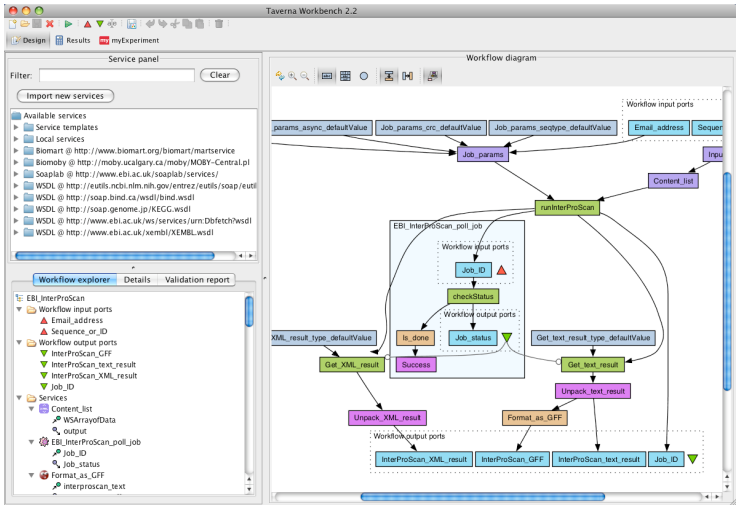
Overview



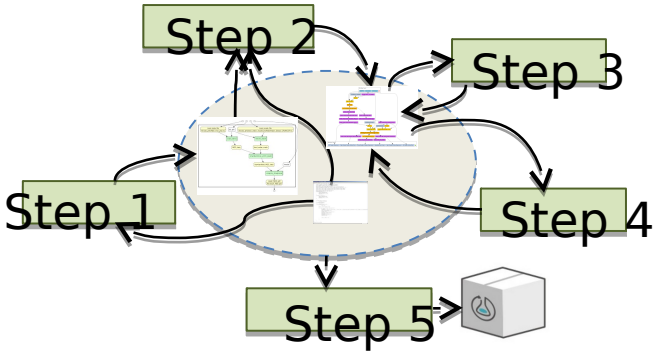
Create

Understand
Reuse
Reproduce

+



Methodology



Related Work

- Script-language specific.
- Workflow-engine specific.
- A new language is needed.
- Outcome is **not** an executable workflow.
- Do not collect provenance data of the conversion process.

Two Kind of Experts

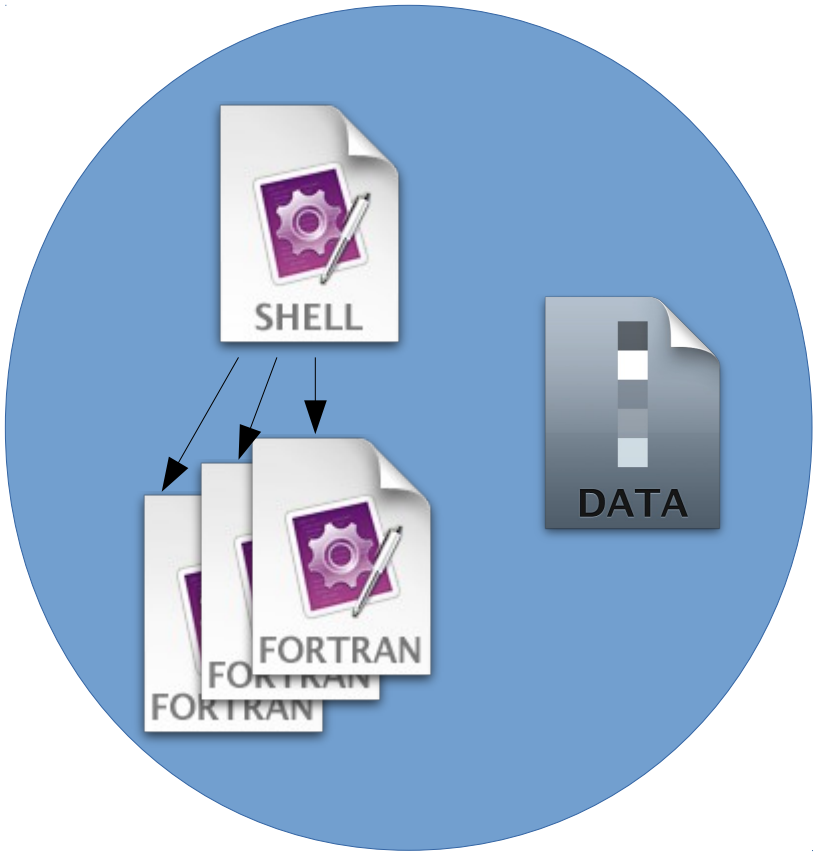
- Scientists
  - Domain experts who understand the experiment, and the script (sometimes called *user*);
- Curators:
  - Scientists who are also familiar with workflow and script programming or;
  - Computer scientists who are familiar enough with the domain to be able to implement our methodology;
  - Responsible for authoring, documenting and publishing workflows and associated resources.

Requirements

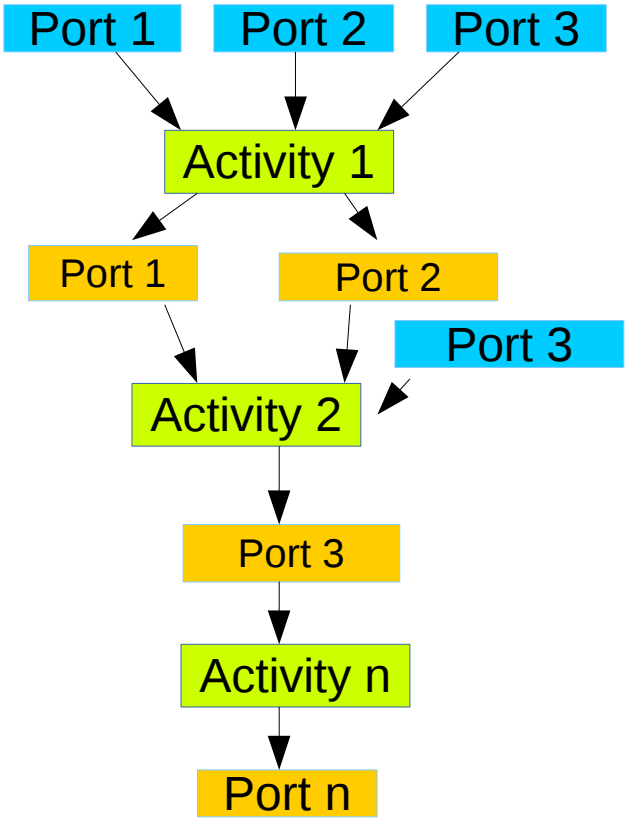
1. Produce workflow-like view of the script.
2. Create an executable workflow and compare execution of workflow and script.
3. Modify the workflow resources.
4. Record provenance data.
5. Aggregate all resources to support Reproducibility and Reuse.

Requirements

1 Produce workflow-like view of the script.



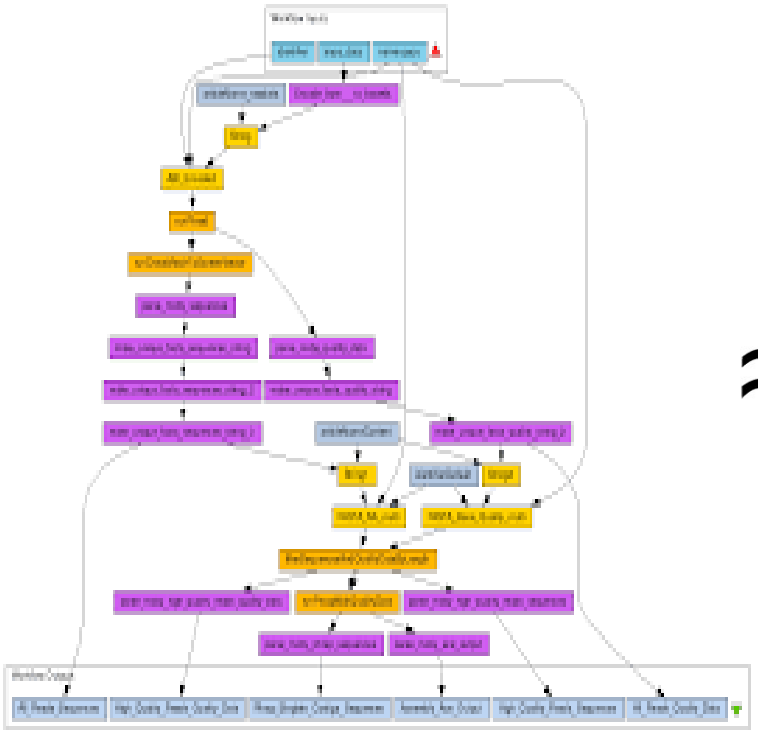
Script-based experiment.



Abstract workflow.

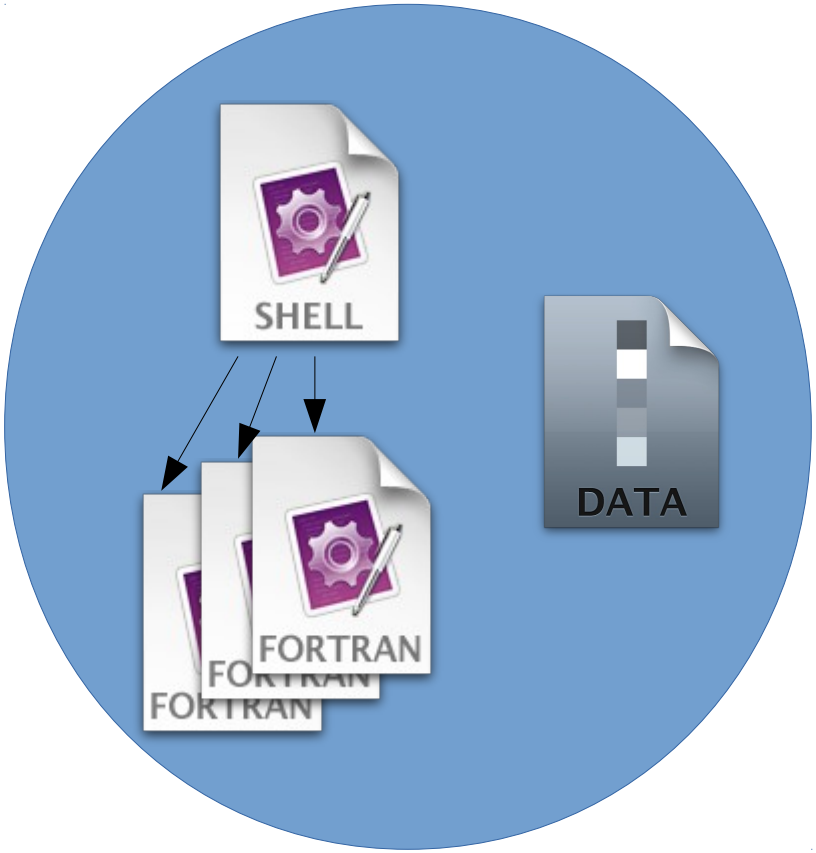
Requirements

2 Create executable workflow and compare execution of workflow and script.



Executable workflow.

≈



Script-based experiment.

Requirements

3 Modify the workflow resources.

(a)

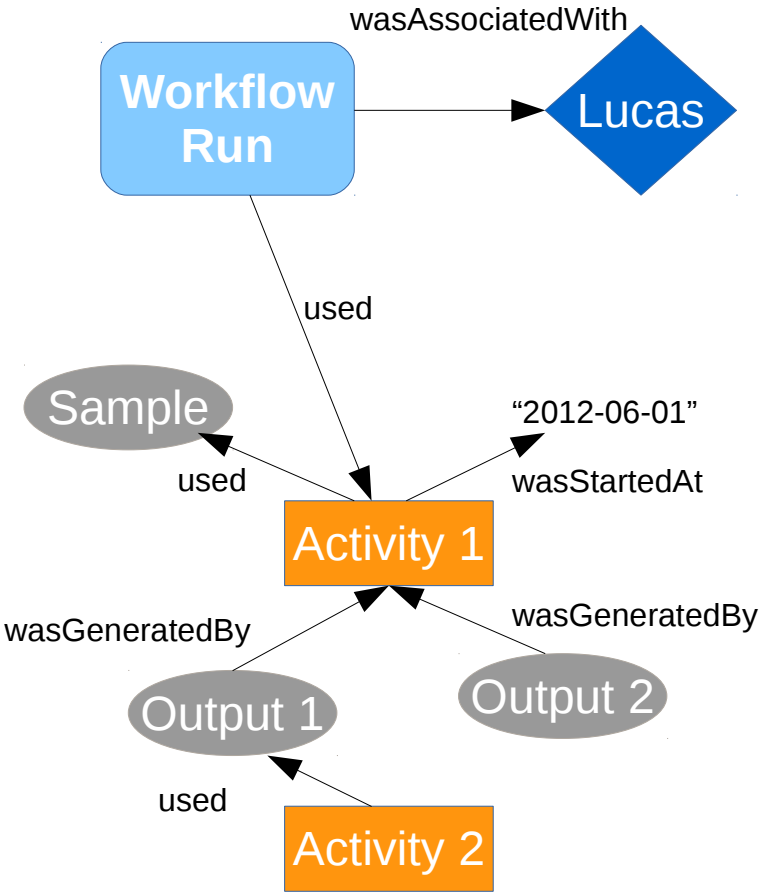
Local

(b)

Algorithm A → Algorithm B

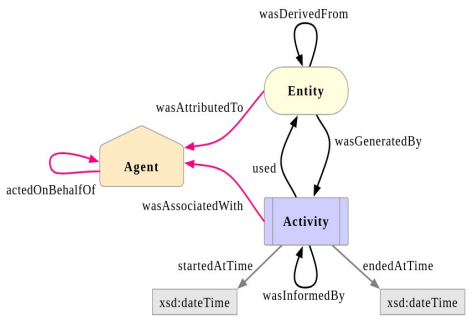
Requirements

4 Record provenance data



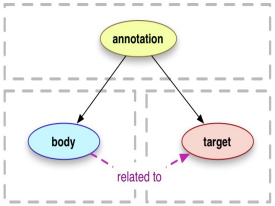
Requirements

5 Aggregate all resources to support Reproducibility and Reuse.



Provenance

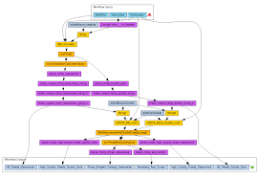
Data



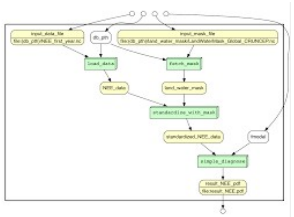
Annotations

Authors

Scripts



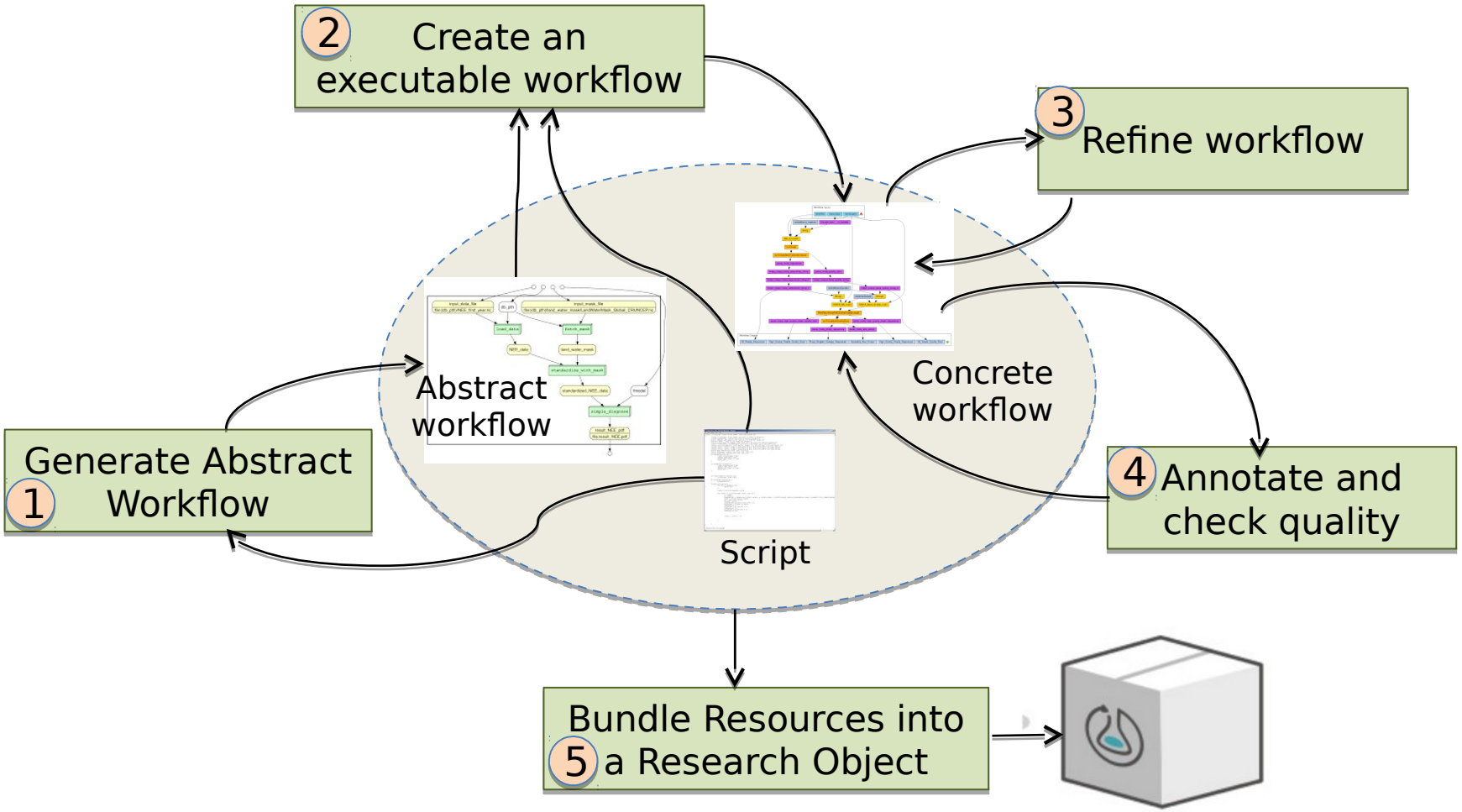
Concrete workflows



Abstract workflows

Papers and Reports

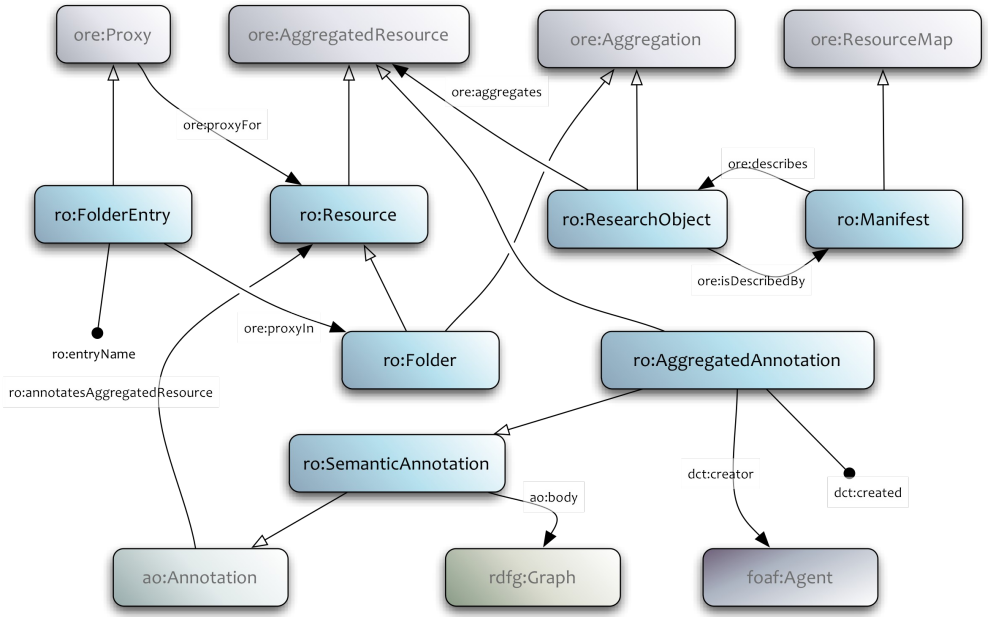
Methodology



Workflow Research Object (WRO)

- Research Objects are semantically rich aggregations of resources that bring together data, methods and people in scientific investigations.

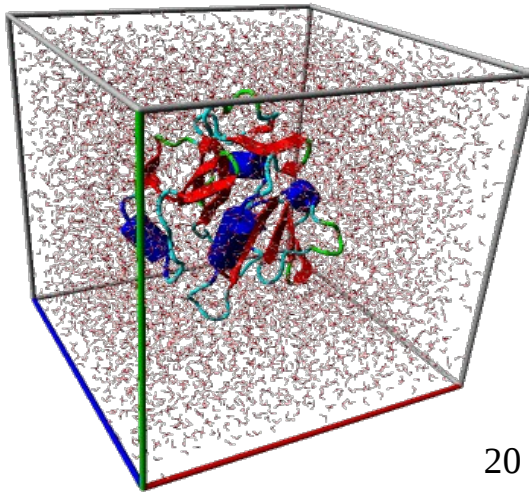
- WROs encapsulate scientific workflows and additional information regarding their context and resources.



Research Object Model

Running Example

- Molecular Dynamics Simulations
  - Many branches of material sciences, computational engineering, physics and chemistry.
  - Scripts (shell script), programs (NAMD, VMD, Fortran)
  - **Phases**: set up, simulation and analysis of trajectories.
  - **Inputs**: protein structure, simulation parameters and force field files.
  - **Output**: trajectories and analysis results.



eScience UNICAMP

Step 1

Generate Abstract Workflow

```
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
```

Script code.

Step 1

Generate Abstract Workflow

```
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
```

Script code.

Manually annotate

```
# @BEGIN split
# @IN initial_structure @URI file:structure.pdb
# @IN directory_path @AS directory
# @OUT protein_pdb  @URI file:{directory}/protein.pdb
# @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
# @OUT water_pdb  @URI file:{directory}/water.pdb
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
# @END split
```

Annotated script code.

Step 1

Generate Abstract Workflow

```
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
```

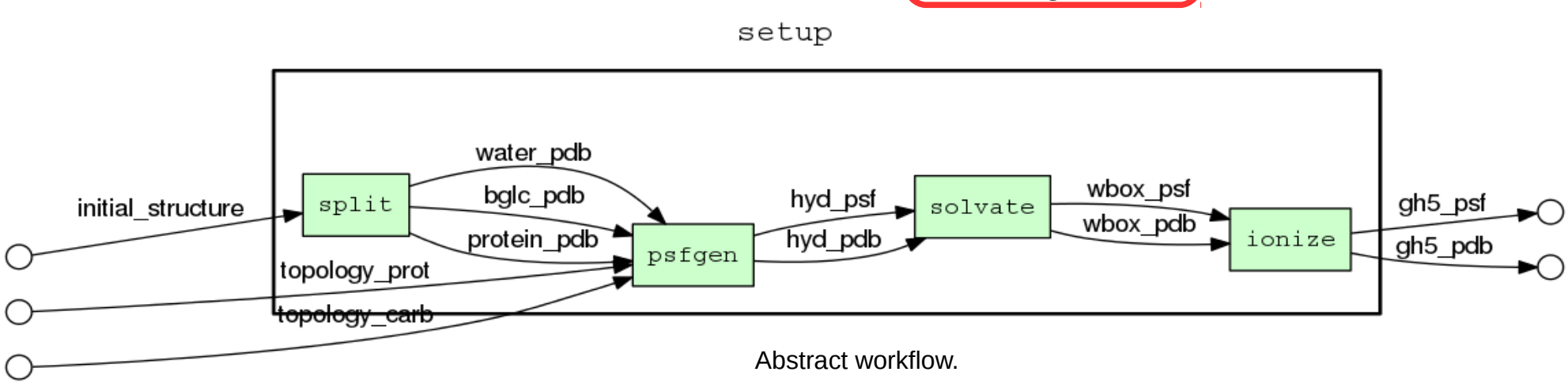
Script code.

Manually annotate

```
# @BEGIN split
# @IN initial_structure @URI file:structure.pdb
# @IN directory_path @AS directory
# @OUT protein_pdb  @URI file:{directory}/protein.pdb
# @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
# @OUT water_pdb  @URI file:{directory}/water.pdb
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
# @END split
```

Annotated script code.

Create workflow-like view



Abstract workflow.

Step 1

Generate Abstract Workflow

YesWorkflow
McPhillips et. al, 2015

- Code comments
- Tags:
  - @begin
  - @end
  - @desc
  - @in
  - @out
  - ...

```
# @BEGIN split
# @IN initial_structure @URI file:structure.pdb
# @IN directory_path @AS directory
# @OUT protein_pdb  @URI file:{directory}/protein.pdb
# @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
# @OUT water_pdb  @URI file:{directory}/water.pdb
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
# @END split
```

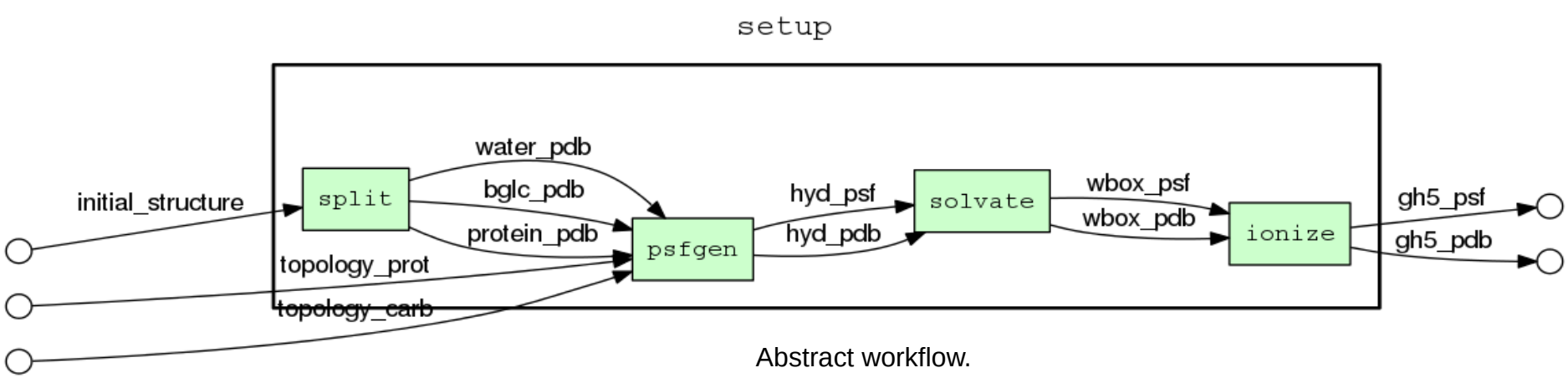
code blocks

Input/ouput

Annotated script code.

Create Workflow-like view

T. McPhillips et al. (2015), "Yesworkflow: A user-oriented, language-independent tool for recovering workflow information from scripts," International Journal of Digital Curation, vol. 10, no. 1, pp. 298–313, 2015.



Abstract workflow.

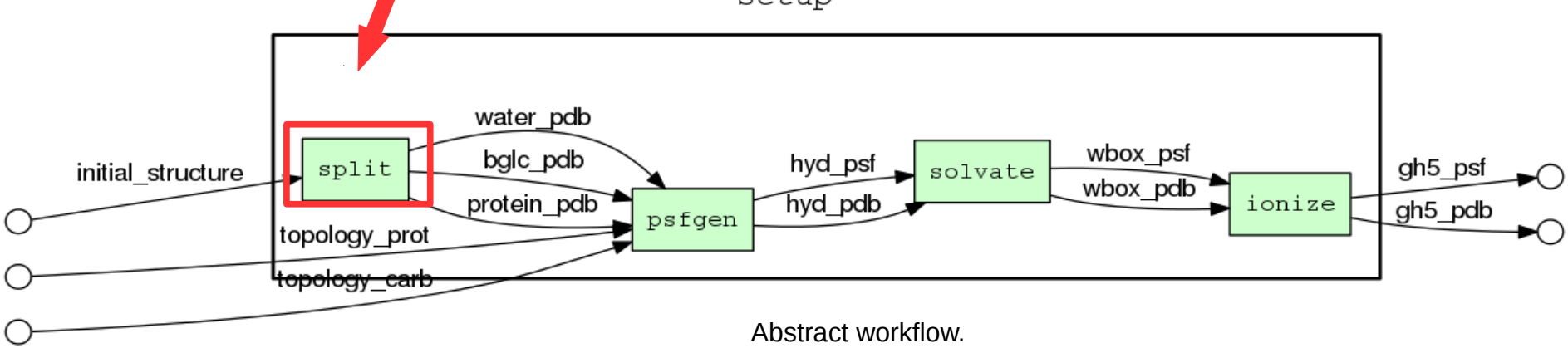
Step 1

Generate 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
```

Annotated script code.

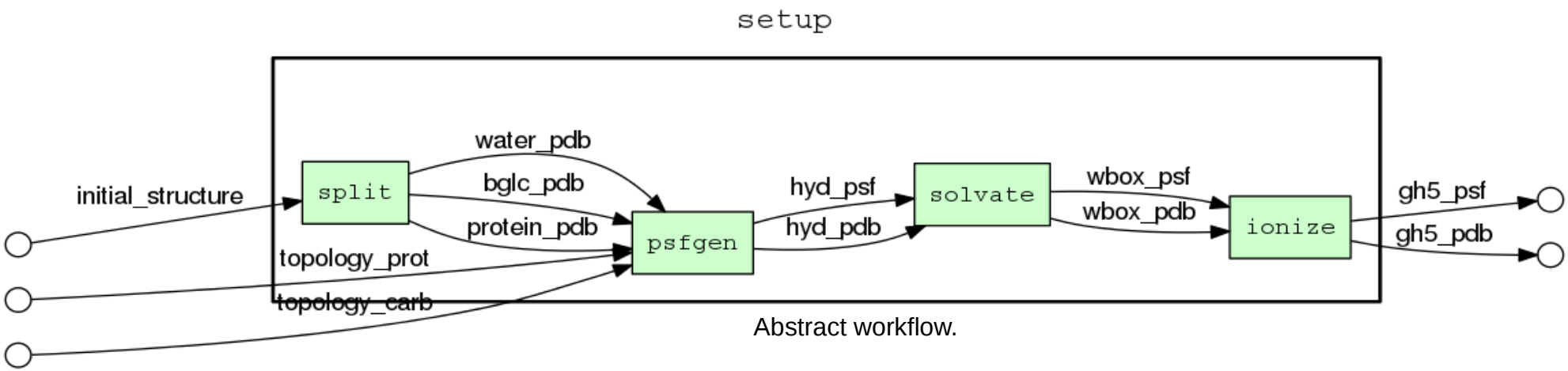
Create Workflow-like view



Abstract workflow.

Step 2

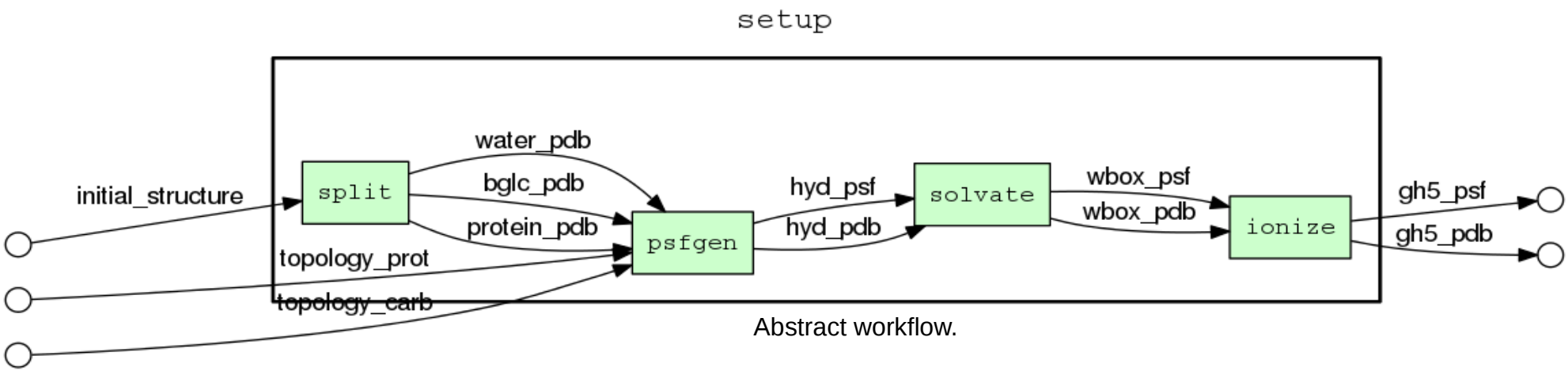
Create an executable workflow



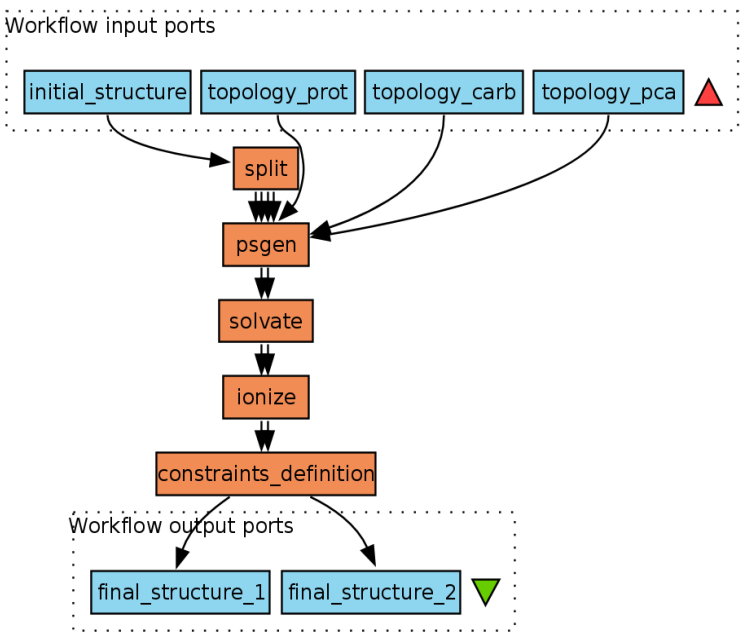
Abstract workflow.

Step 2

Create an executable workflow



Abstract workflow.



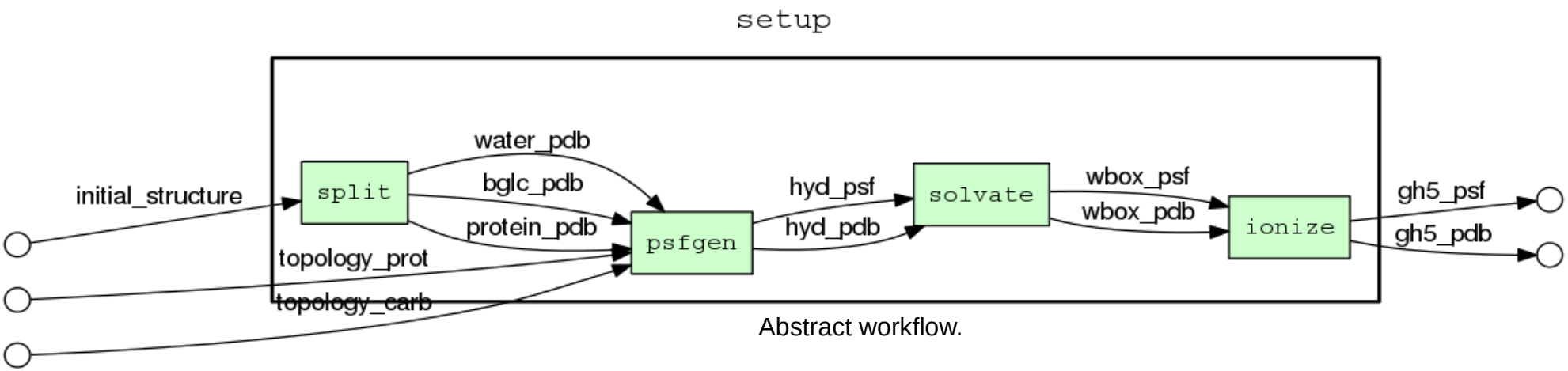
Executable workflow.

Create implementation of activities

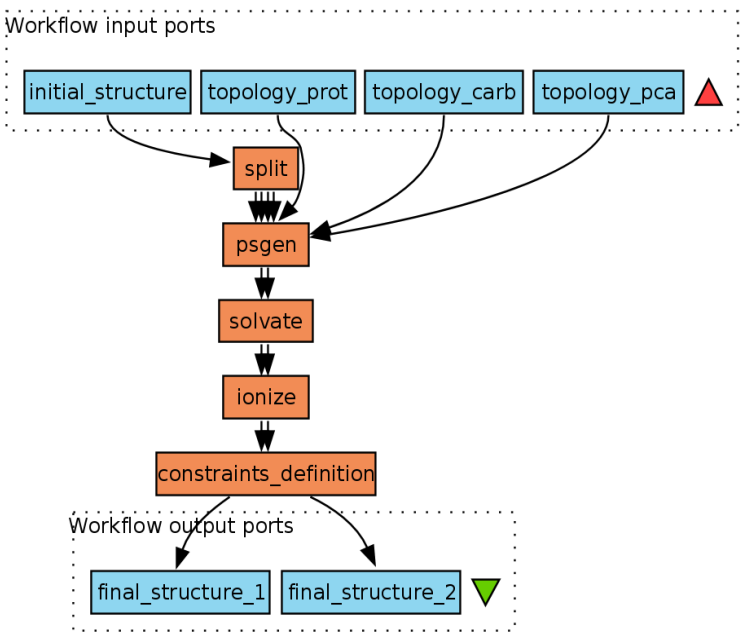
Copy code blocks from the script.

Step 2

Create an executable workflow



Abstract workflow.



Executable workflow.

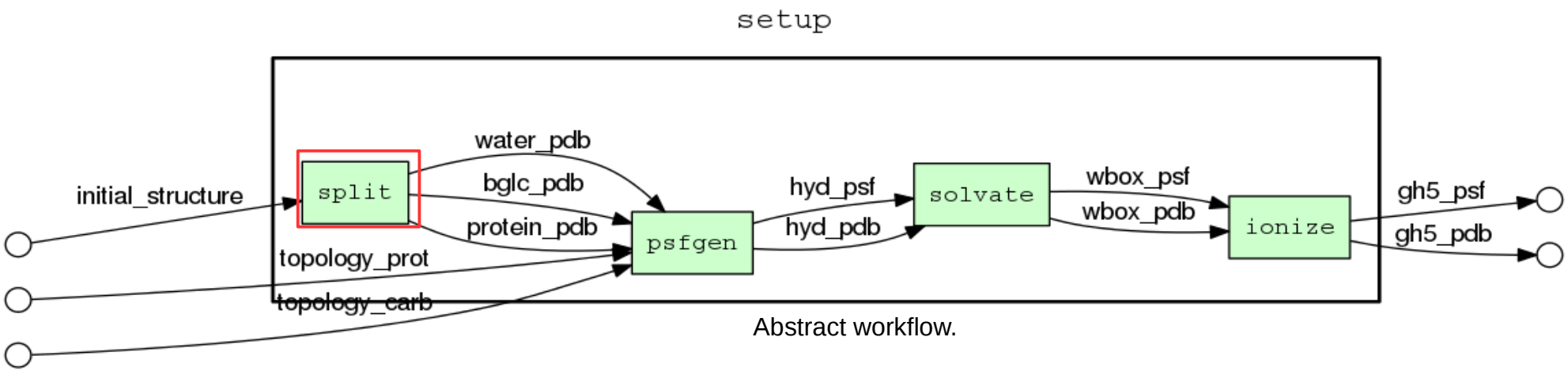
Create implementation of activities

Copy code blocks from the script.

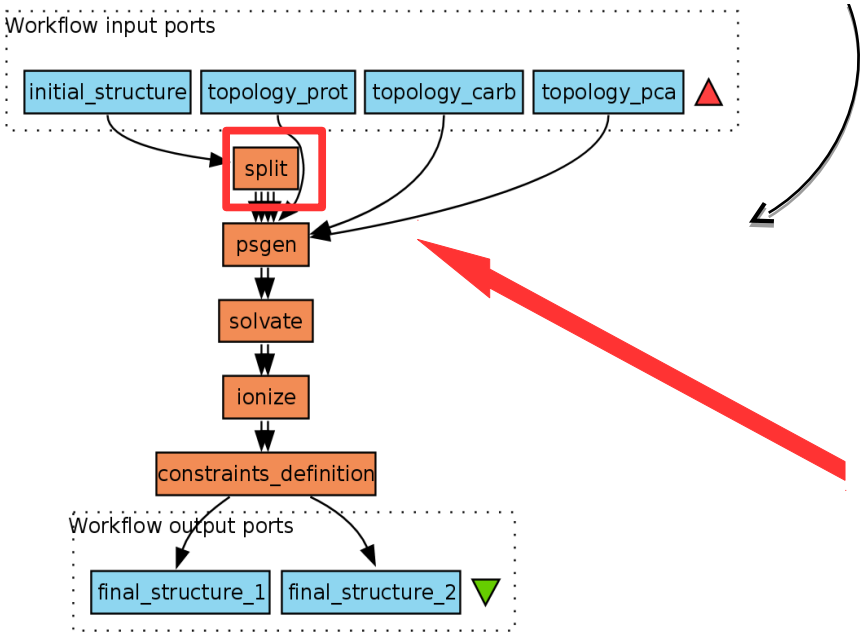
Taverna

Step 2

Create an executable workflow



Abstract workflow.



Executable workflow.

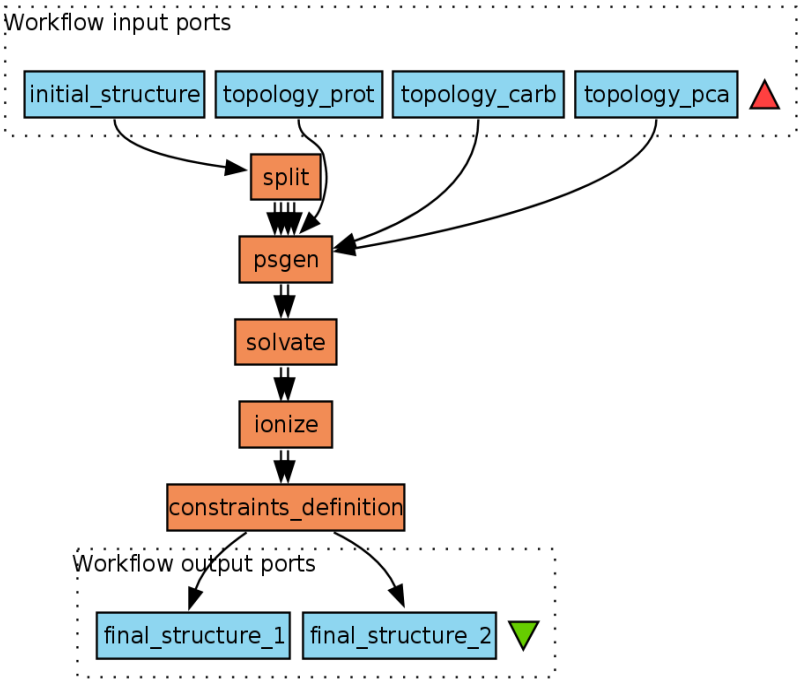
Create implementation of activities

```
# @BEGIN split
# @IN initial_structure @URI file:structure.pdb
# @IN directory_path @AS directory
# @OUT protein_pdb  @URI file:{directory}/protein.pdb
# @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
# @OUT water_pdb  @URI file:{directory}/water.pdb
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
# @END split
```

Script code.

Step 3

Refine executable workflow



Executable workflow.

Modify resources:

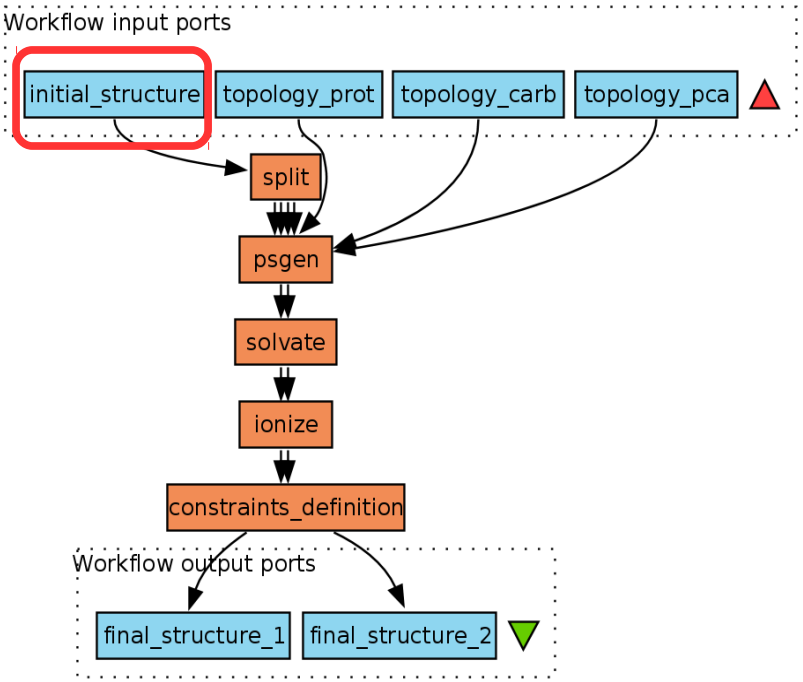
- Algorithms
- Data Sets
- Parallelization
- Web Services
- ...

New workflow version.

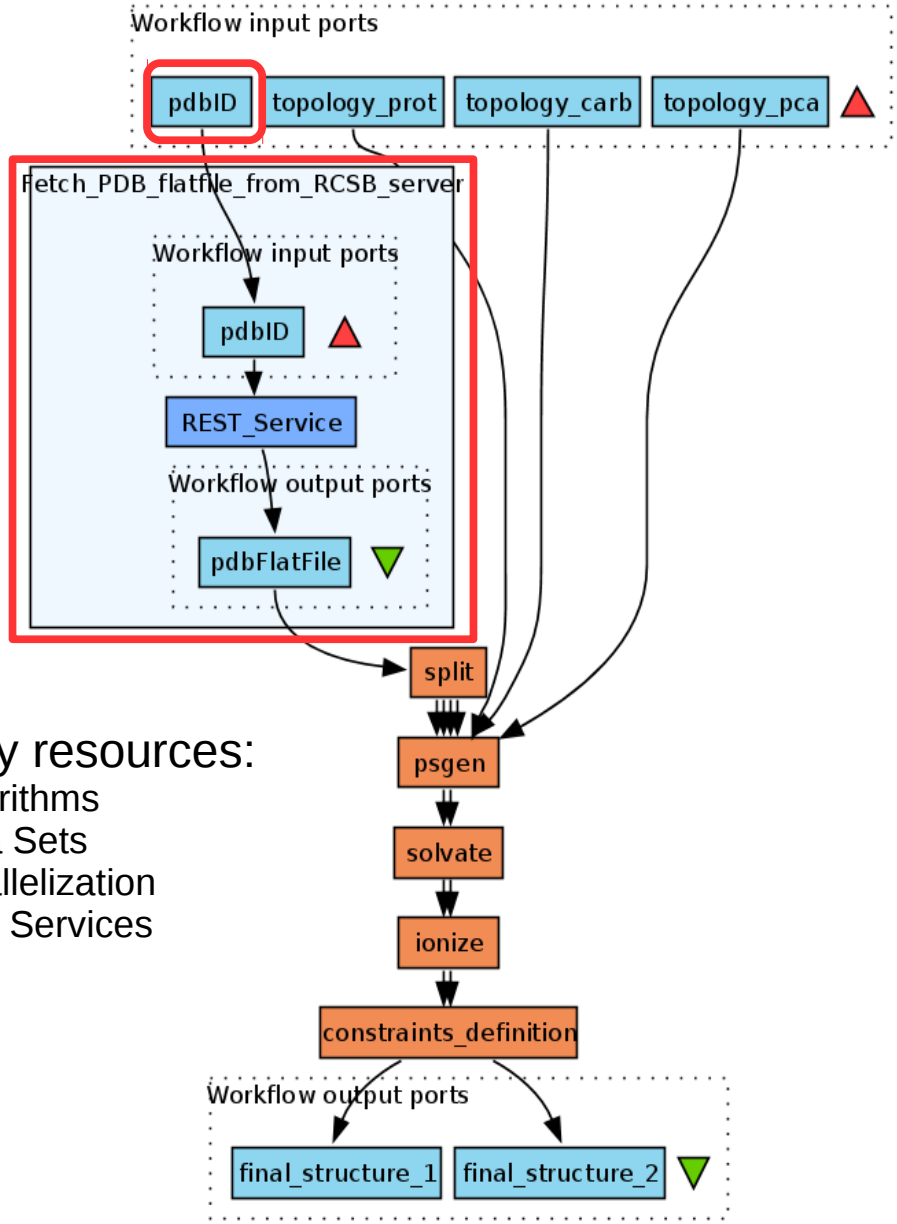
Step 3

Refine executable workflow

Create new version



Executable workflow.



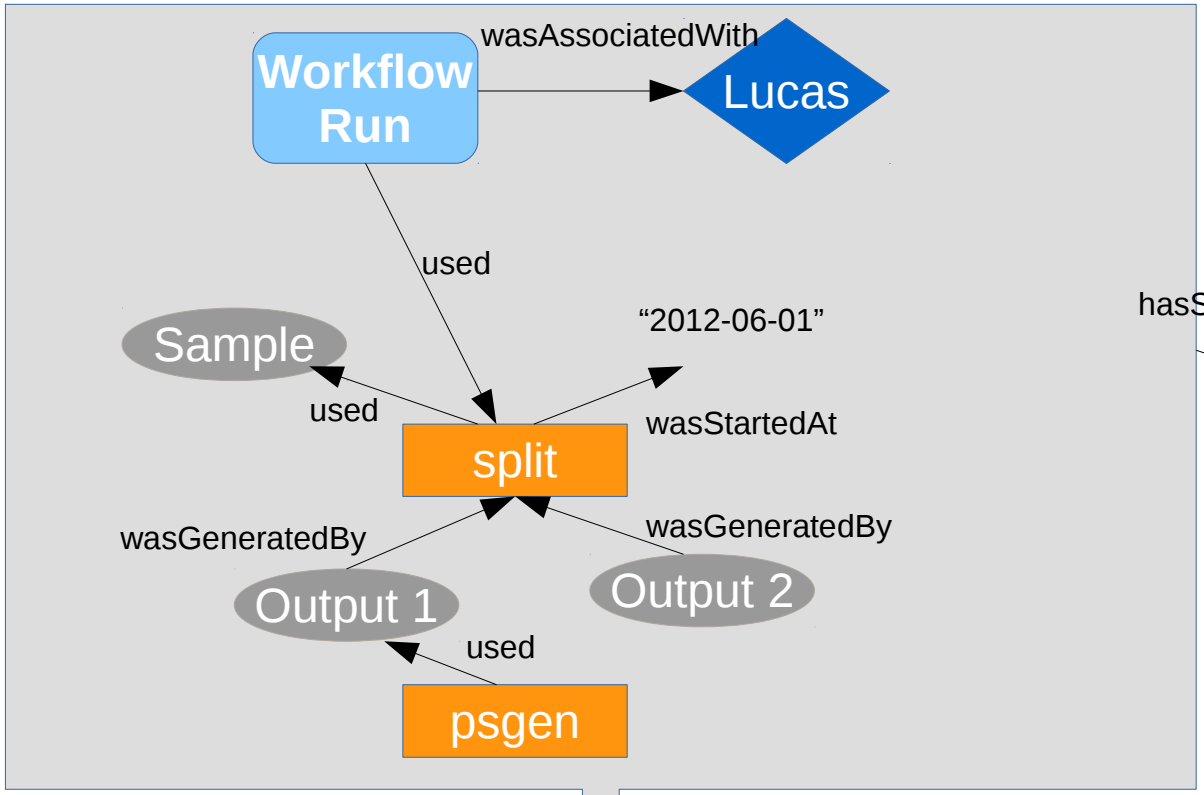
New workflow version.

Modify resources:

- Algorithms
- Data Sets
- Parallelization
- Web Services
- ...

Steps 2 3

Record provenance data: execution traces.

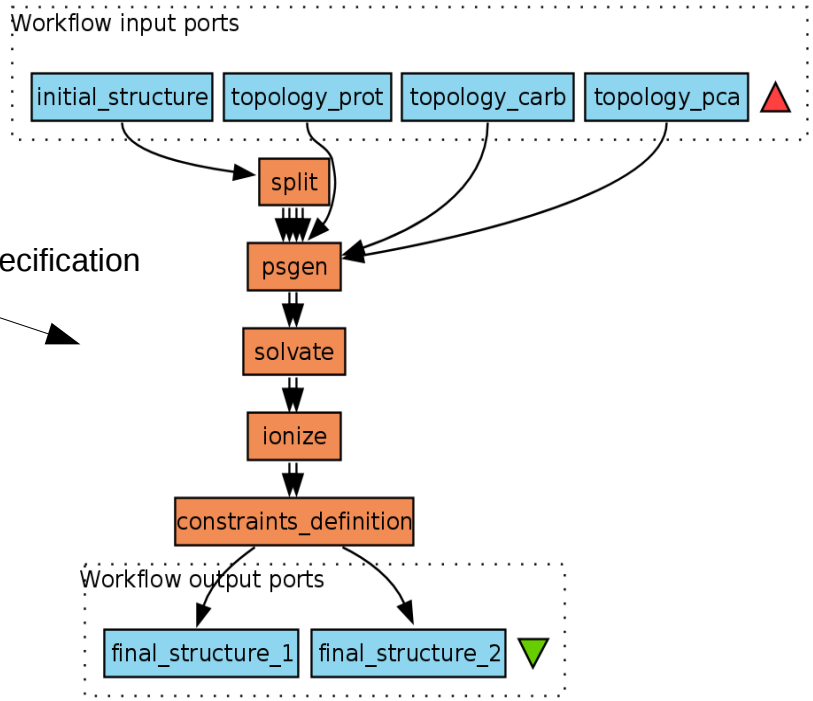


W3C PROV

wasEnactedBy

Taverna

hasSpecification



Executable workflow.

Steps 2 3

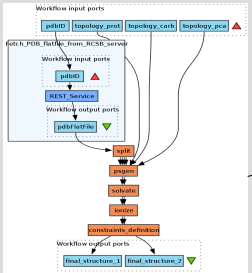
Record provenance data: conversion process.

```
# @BEGIN split
# @IN initial_structure @URI file:structure.pdb
# @IN directory_path @AS directory
# @OUT protein_pdb  @URI file:{directory}/protein.pdb
# @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
# @OUT water_pdb  @URI file:{directory}/water.pdb
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
# @END split
```

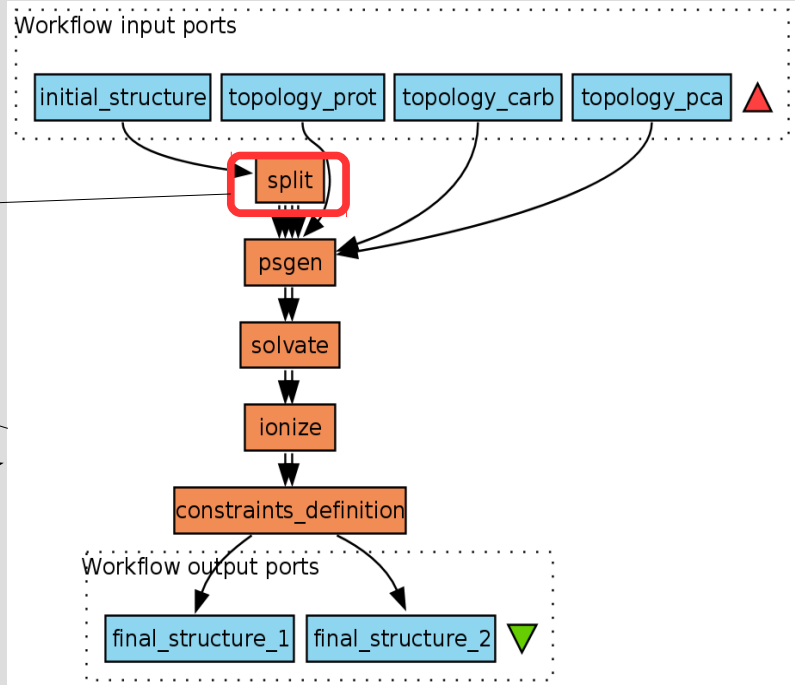
Script code.

wasDerivedFrom

wasDerivedFrom



wasDerivedFrom



Executable workflow.

New workflow version.

W3C PROV

wasAssociatedWith

Curator

Step 4

Annotate and check quality

- Annotations describing the workflow.
- Use provenance data
  - To check the quality of the conversion process.
- Run checks to verify the soundness of the workflow.

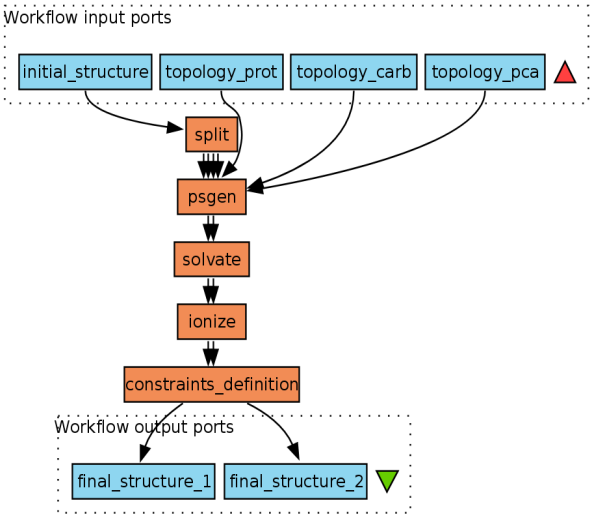
Step 4

Annotate and check quality

```
# @BEGIN split
# @IN initial_structure @URI file:structure.pdb
# @IN directory_path @AS directory
# @OUT protein_pdb  @URI file:{directory}/protein.pdb
# @OUT bglc_pdb  @URI file:{directory}/bglc.pdb
# @OUT water_pdb  @URI file:{directory}/water.pdb
structure = $directory_path"/structure.pdb"
protein = $directory_path"/protein.pdb"
water = $directory_path"/water.pdb"
bglc = $directory_path"/bglc.pdb"
egrep -v '(TIP3|BGLC)' $structure > $protein
grep TIP3 $structure > $water
grep BGLC $structure > $bglc
# @END split
```

Script code.

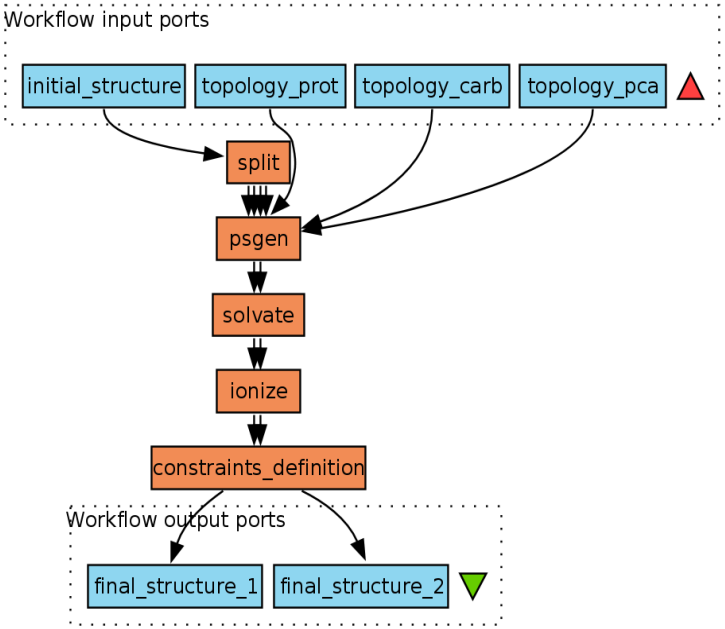
≈



Executable workflow.

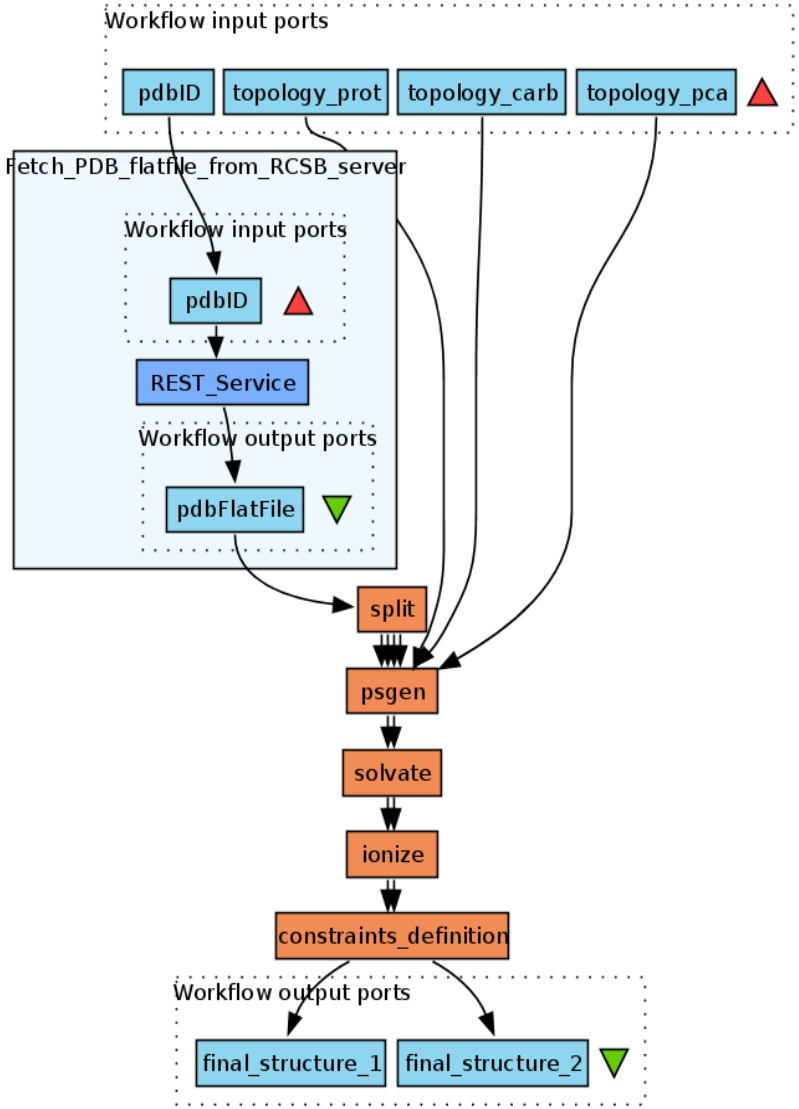
Step 4

Annotate and check quality



Initial Executable workflow.

≈



Workflow version.

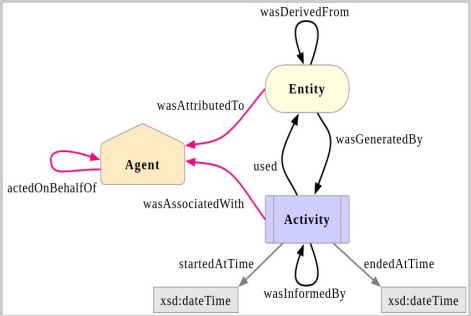
Step 4

Annotate and check quality

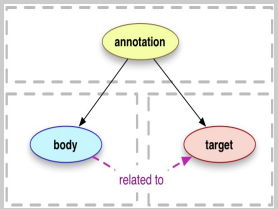
- Common mistakes during the conversion:
  - not clearly identified the main logical processing units in the script;
  - a mistake when migrating script code into the corresponding activity;
  - not provided the correct input files and parameters;
  - the coding of the workflow itself contained errors.

Step 5

Bundle Resources into a Research Object



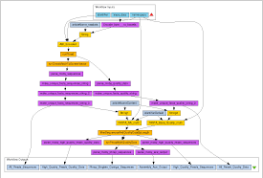
Provenance Data



Annotations

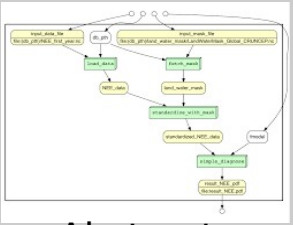
Attributions

Script



Concrete workflow(s)

Paper



Abstract workflow

Contributions

- A methodology that guides curators in a principled manner to transform scripts into reproducible and reusable WRO;
- This addresses an important issue in the area of script provenance;

Conclusions

- We addressed issues wrt understanding, reuse and reproducibility of script-based experiments.
- The methodology created was:
  - elaborated based on requirements;
  - showcased via a real world use case from the field of Molecular Dynamics;
- We exploited tools and standards from the scientific community:
  - Scientific Workflows, YesWorkflow, Research Objects, the W3C PROV recommendations and the Web Annotation Data Model.
- The bundle is available at http://w3id.org/w2share/s2rwro/

Next Steps

- Evaluation using other case studies;
- Evaluation of the cost of the effectiveness of our methodology;
- Extension of YesWorkflow to support the semantic annotation of blocks;
- Implementation of tools.

Acknowledgments

- FAPESP (grant # 2014/23861-4)
- CCES/CEPID (grant # 2013/08293-7)
  - Center for Computational Engineering & Sciences
- LIS (Laboratory of Information Systems)
- Prof. Munir Skaf and his group from Institute of Chemistry - Unicamp.

Converting Scripts into Reproducible Workflow Research Objects

Lucas A. M. C. Carvalho, Khalid Belhajjame, Claudia Bauzer Medeiros

lucas.carvalho@ic.unicamp.br

12th eScience 2016
IEEE INTERNATIONAL CONFERENCE

Baltimore, Maryland, USA

October 23-26, 2016