noWorkflow

Capturing, Analyzing and Managing Provenance from Python Scripts

https://github.com/gems-uff/noworkflow

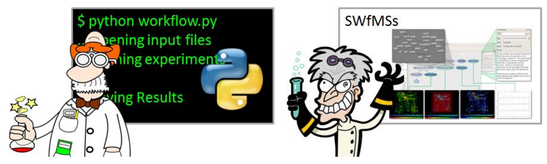
João Felipe Nicolaci Pimentel (UFF), Juliana Freire (NYU), Vanessa Braganholo (UFF), Leonardo Murta (UFF)

Instituto de Computação UFF

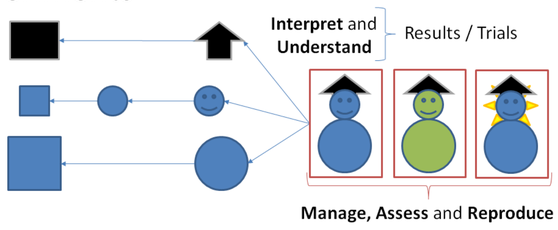
NYU POLYTECHNIC SCHOOL OF ENGINEERING

Motivation and Goal

- Python **scripts** vs **Scientific Workflow Management Systems** (SWfMSs).



- Scientists need **provenance** for computational experiments

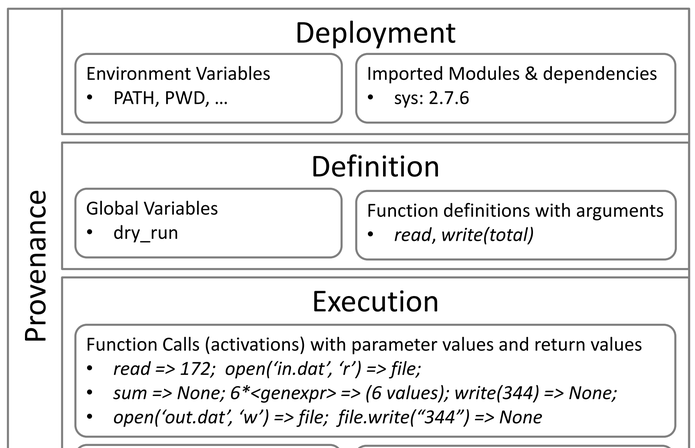


- State of the art
  - Workflow systems and annotated scripts require substantial user input
- noWorkflow
  - **Transparently** captures provenance of Python scripts -- no changes required!
  - Allows users to analyze provenance data

Usage

- Install: `$ pip install noworkflow[all]`
  - Installs noWorkflow, PyPosAST, Flask, IPython Notebook, PySWIP
- Run

`$ now run script.py 0`



`$ now run -e Tracer script.py 1`

- Default Provenance
- Variable assignments & dependencies

script.py

```
1  import sys
2  dry_run = False
3  def read():
4      global dry_run
5      with open('in.dat', 'r') as f:
6          result = sum(int(line.strip()) for line in f)
7      if dry_run:
8          result = 50
9      return result
10
11 def write(total):
12     with open('out.dat', 'w') as f:
13         f.write(str(total))
14
15 if __name__ == '__main__':
16     dry_run = bool(int(sys.argv[1]))
17     total = read()
18     total *= 2
19     write(total)
20
```

Analysis and Management

- List trials: `$ now list`

```
[now] trials available in the provenance store:
  Trial 1: script.py 0
           with code hash 51d3419aac4c17468d47ee862c7ee17ab5090c2d
           ran from 2015-05-03 16:07:16.587178 to 2015-05-03 16:07:16.627305
  Trial 2: script.py 1
           with code hash 51d3419aac4c17468d47ee862c7ee17ab5090c2d
           ran from 2015-05-03 16:07:19.270899 to 2015-05-03 16:07:19.336207
  Trial 3: script.py
           with code hash 51d3419aac4c17468d47ee862c7ee17ab5090c2d
           ran from 2015-05-03 16:13:20.320483 to None
  Trial 4: script.py <restore 2>
           with code hash 9c629bfd81ebfd8343ec51d31dd3234e1bb07c6b
           ran from 2015-05-03 16:16:31.471202 to None
  Trial 5: script.py 1
           with code hash 75d1106fac894036c40f5085411efd6caae6bacb
           ran from 2015-05-03 16:17:02.333177 to 2015-05-03 16:17:02.389163
```

- Show trial provenance: `$ now show 5 -a`

```
[now] trial information:
  Id: 5
  Inherited Id: None
  Script: script.py
  Code hash: 75d1106fac894036c40f5085411efd6caae6bacb
  Start: 2015-05-03 16:17:02.333177
  Finish: 2015-05-03 16:17:02.389163
[now] this trial has the following function activation graph:
  126: /home/joao/projects/nowissues/scipyla/script.py (2015-05-03
16:17:02.385048 - 2015-05-03 16:17:02.389106)
       Globals:
       Arguments:
       Return value: None
    17: read (2015-05-03 16:17:02.385256 - 2015-05-03 16:17:02.387567)
        Globals: dry_run = 0
        Arguments:
        Return value: 40
      5: open (2015-05-03 16:17:02.385448 - 2015-05-03 16:17:02.385975)
         ...
```

- Compare trials: `$ now diff 1 2`

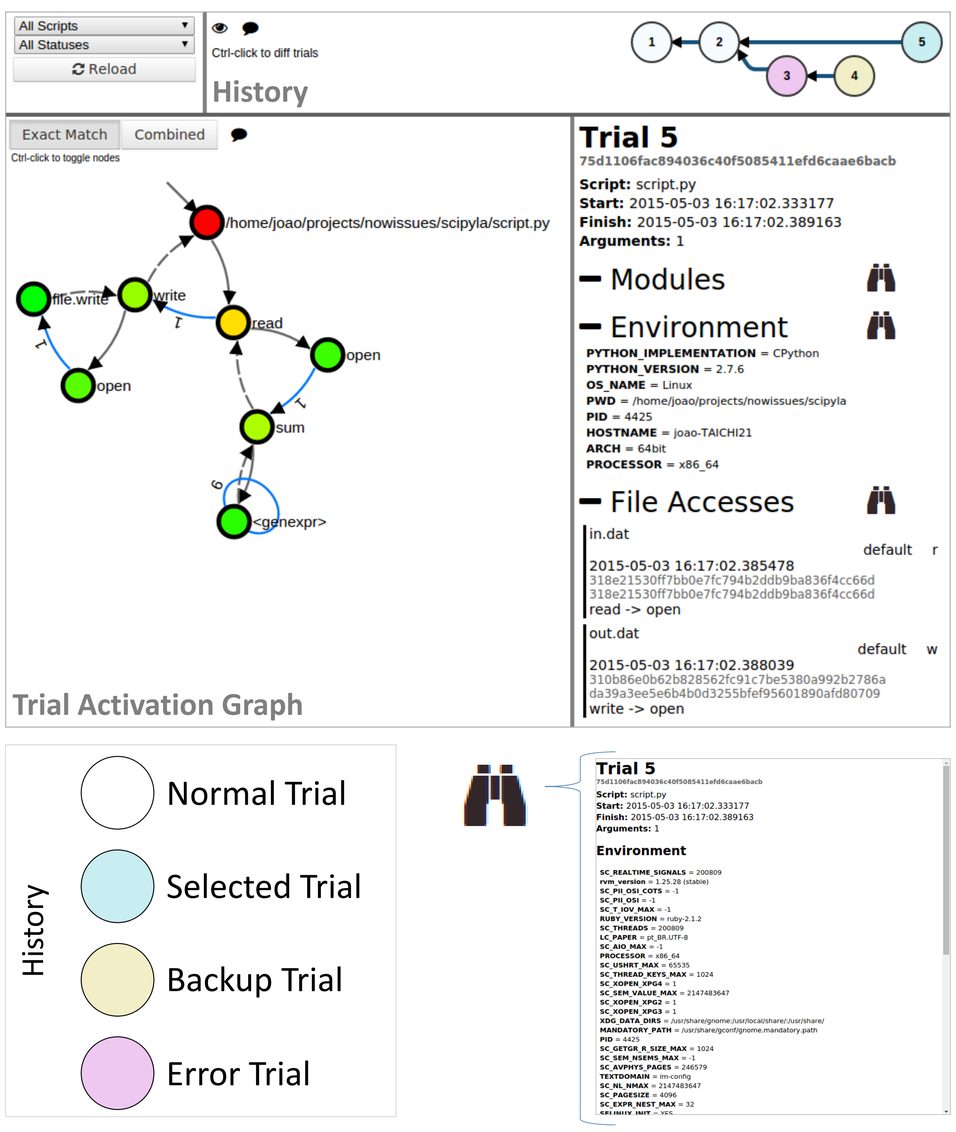
```
[now] trial diff:
  duration changed from 40127 to 65308
  start changed from 2015-05-03 16:07:16.587178 to 2015-05-03 16:07:19.270899
  finish changed from 2015-05-03 16:07:16.627305 to 2015-05-03 16:07:19.336207
  arguments changed from 0 to 1
  parent_id changed from None to 1
```

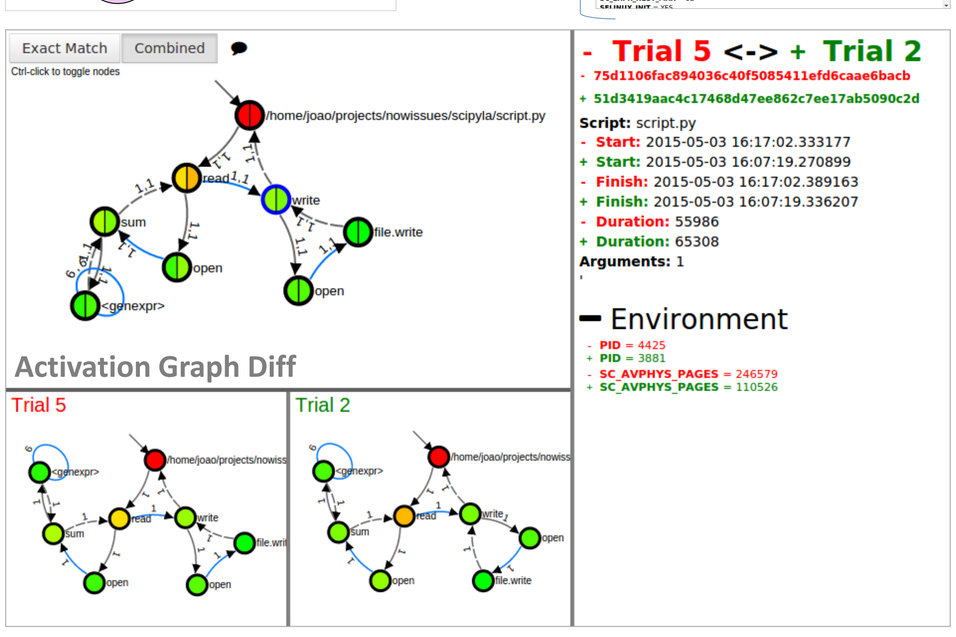
- Export
  - Trial to Prolog: `$ now export -r 1 > trial1.pl`
  - Trial to IPython Notebook: `$ now export -i 1`
  - History to Notebook: `$ now export -i history`
  - Diff to Notebook: `$ now export -i diff:1:2`

- Restore old script and files: `$ now restore -li 2`

```
[now] Backup Trial 4 created
[now] File script.py from trial 2 restored
[now] File in.dat from trial 2 restored
[now] File out.dat from trial 2 restored
```

- Web visualization tool: `$ now vis`





- Notebook: `$ ipython notebook`

```
In [1]: %load_ext noworkflow
        %now_set_default graph_width=430 graph_height=150
        nip = %now_ip

In [2]: dry = 0
        trial = %now_run --name ipython_script script.py $dry
        trial
Out[2]: Trial 6. Ctrl-click to toggle nodes

In [3]: trial.modules()
Out[3]: ([],
         [OrderedDict([('id', 1), ('name', u'sys'), ('version
        ', u'2.7.6'), ('path', None), ('code_hash', None)])])

In [4]: size = 5

In [5]: %%now_run --name ipython_script --out=out_var $size
        import sys
        l = range(int(sys.argv[1]))
        c = sum(l)
        print(c)
Out[5]: Trial 7. Ctrl-click to toggle nodes

In [6]: out_var
Out[6]: '10\n'

In [7]: nip.History()
Out[7]:

In [8]: %%now_prolog --result result {trial.id}
        duration({trial.id}, read, X)

In [9]: for match in result:
            print(match['X'])
        0.00296902656555

In [10]: %%now_sql
         SELECT DISTINCT script FROM trial
Out[10]:
| script |
| --- |
| script.py |
| ipython_script |

In [11]: diff = nip.Diff(1, 6)

In [12]: diff
Out[12]: Diff 1/6. Ctrl-click to toggle nodes

In [13]: (diff.trial1.script, diff.trial2.script)
Out[13]: (u'script.py', u'ipython_script')
```