Tutorial
Workflow Systems: OCCAM
Big picture, workflows, examples, demo

Bruce Childers
Associate Dean for Strategic Initiatives
Professor of Computer Science
School of Computing & Information
childers@pitt.edu
http://www.cs.pitt.edu/~childers
If a tree falls in a forest and no one is around to hear it, does it make a sound?
If you create a model and use it in an experiment, and nobody can access it (& repeat!), does it exist?
70% failed replication trial (50% of their own!); 45% due to data/code, 80% contributed.

Weak repeatability for 32.3% of 601 papers in computer systems research.
Path to the Emerald City

I. Community and culture
   - Galvanize, incentivize, & educate
   - Review, mandates, **funding**, expectations
   - Governance, policy & procedures
   - Work with existing practices to push boundaries

II. Technology
   - Ease effort to **accelerate research**
   - **Leverage** for higher quality, new discovery
   - **Path of least resistance** is “right one”
   - Work with existing methods to push boundaries
Technology: Workflow Systems

A place to do experiments
- Computational experiments (model + data + compute)
- Setup your own, or use one that is available

Accelerate your research
- Focus on research rather than the infrastructure
- Leverage shared data, models, & other resources
- Capture & document experiment (FAIR)
Workflow System
Workflow System

(1) Capture

Front-end User Interaction
- Create & edit workflow
- View & manipulate results
- User workspace
Workflow System

(1) Capture

(2) Engine

Middle-End Management
- User interface
- Generate & dispatch
- Repository (database)
Workflow System

(1) Capture
(2) Engine
(3) Workers

Back-End Compute
- Model/simulation workers
- Compute servers
- Local, private HPC, cloud
Workflow

Sequence of computational steps

- Directed acyclic graph (DAG)
- Nodes: Process (executes), data (input/output)
- Edges: data flow of one step to the next
- Operators: of the language (e.g., transformation)

By itself, workflow is a specification

- Realized: Program, script, visual language
- Represents experiment structure, data, steps
Workflow Pattern: Process

- Data input (May be typed)
- Model
- Process executable
- Output
Workflow Pattern: Pipeline

- Data
- Init
- Model
- Visualize
- 2nd stage
- 2nd input parameters
- Intermediary data
Workflow Pattern: Pipeline

DRYAD data → harmonize → model → reduce → visualize

- Format transform for model
- Summarize filter / ROI volume
Workflow Pattern: Fork & Join

Fork
One source to multi sink

Join
multi source to one sink

Format
same format to combine

data

input harmonize

input harmonize

model A

output harmonize

output harmonize

ensemble

same format to combine
Capture: How to Specify

- **Text**

```
cwlVersion: cwl:draft-3
class: CommandLineTool
baseCommand: echo
inputs:
  - id: message
type: string
inputBinding:
  position: 1
outputs: []
```

**message:** Hello World Modelers!

**Credit:** CWL 3.0 hello world example
Capture: How to Specify

- Text
- Draw (visual)
Capture: How to Specify

- Text
- Draw (visual)
- Generate

Example generators:
- Montage: Image stitching for astronomy
- CyberShake: Earthquake modeling for So. California
- LIGO: Analyze gravitational waveforms
Capture: How to Specify

- Text
- Draw (visual)
- Generate
- Program

```r
input_file <- "data/data.csv"
output_file <- "data/results.csv"

# read input
input_data <- read.csv(input_file)
# get number of samples in data
sample_number <- nrow(input_data)
# generate results
results <- some_other_function(input_file, sample_number)
# write results
write.table(results, results_file)
```

Credit: SW Carpentry Programming with R
© 2018 BRUCE CHILDERS
**Workflow Execution**

Need to run (i.e., “conduct experiment”)
- Workflow (few steps to 100Ks of steps)
- Actual input and configuration values
- Resources (e.g., machines)

*Workflow engine* handles the “magic”
- Compute scheduling: dispatch processes to resources
- Data orchestration: input, intermediary, result
- Management: Monitoring, logging, error recovery
- Metadata for experiment, workflow preserved
Workflow Execution (Simple)

Simple scheduling on non-distributed resources
Order imposed by workflow
Inputs and resource available to execute step
Workflow Execution (Complex)
Workflow Execution (Complex)

Data sources and sinks
Grouping computational steps & allocate resources
Failure and recovery
Some Factors to Consider

- Science domain
- Interactivity (vs. generated)
- Scalability
- Resources (e.g., laptop vs. distributed cloud)
- Collaboration (e.g., workspaces, identifiers)
- Metadata & preservation
<table>
<thead>
<tr>
<th>System</th>
<th>Application</th>
<th>Notable features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galaxy</td>
<td>Genomics</td>
<td>Commons, HPC, repeatability</td>
</tr>
<tr>
<td>Pegasus</td>
<td>General science</td>
<td>Data-centric, scalability, community</td>
</tr>
<tr>
<td>Kepler</td>
<td>Astronomy, others</td>
<td>Derived embedded systems modeling</td>
</tr>
<tr>
<td>Traverña</td>
<td>Biomedicine, others</td>
<td>Workbench, provenance</td>
</tr>
<tr>
<td>VisTrails</td>
<td>General science</td>
<td>Provenance, visualization, exploration</td>
</tr>
<tr>
<td>CK</td>
<td>Machine learning</td>
<td>Packaging, comparison, auto-tuning</td>
</tr>
<tr>
<td>ReproZip</td>
<td>Data science</td>
<td>Record &amp; replay for repeatability</td>
</tr>
<tr>
<td>Pachyderm</td>
<td>General science</td>
<td>Repeatability, provenance, commercial</td>
</tr>
<tr>
<td>CodeOcean</td>
<td>General science</td>
<td>Publication, editing &amp; run, commercial</td>
</tr>
</tbody>
</table>
OCCAM: *Experiment System*

- Workflows
- Research life cycle
- Contributory & collaborate
- Preservation, provenance
- FAIR metadata

1. Innovate
2. Evaluate
3. Review
4. Publish
5. Derive

Tools:
- Java
- R
- Python
- GitHub
- SSt
Demo! #1

- Start of research lifecycle
  - Creating, running & sharing an experiment
  - Influenza model with FRED
  - Visualize results
Select Object

Type

simulator

Name

FRED

Workflow Templates

Recently Used Objects

None

Library

Instantiated FRED (1 of 4)
Instantiated FREDs (4 of 4)
Configuring Model Generator

Household Confinement

Probability of household confinement: 0.3
### Vaccines become available

**Candidates Vaccinated**

<table>
<thead>
<tr>
<th>Start of event</th>
<th>End of event</th>
<th>Maximum number of events</th>
<th>Percentage of candidates vaccinated</th>
<th>Center latitude of circle of exposure</th>
<th>Center longitude of circle of exposure</th>
<th>Radius of exposure (km)</th>
<th>Fips code of affected region (zero - disabled)</th>
<th>Minimum age</th>
<th>Maximum age</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>999</td>
</tr>
</tbody>
</table>
Run!

Dispatch To

142.4.212.185

Run
Summary

Workflow systems!
- Accelerate research progress
- Leverage models, data, experiments
- Collaboration for modeling

Part of solution to “reproducibility crisis”

Collaboration, sharing, metadata as well
Community and culture are equally important