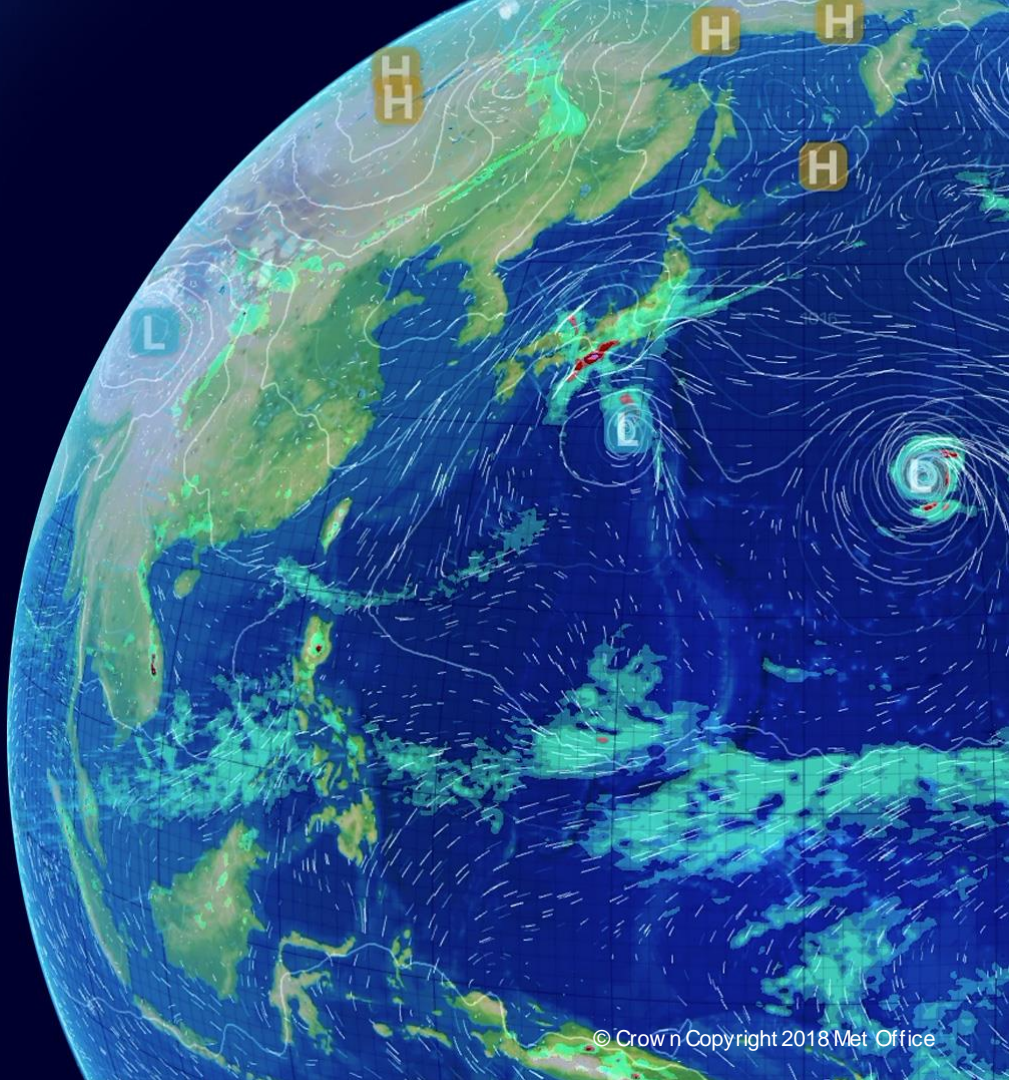


Rose & Cylc: User Update 2019

Sadie Bartholomew

Oliver Sanders

... & the Modelling Infrastructure
Support Systems Team (*Metomi/MISS*)



Background

A brief summary of Rose & Cylc & their development

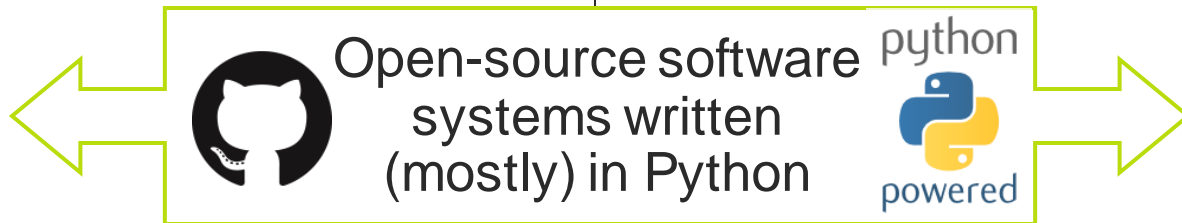
→ What are Rose & Cylc?



- a framework for management of (application) *configurations*
- for use with Cylc to manage suites
 - further suite configuration & control
- + extra helpful tools



- a general-purpose workflow engine (*meta-scheduler*)
 - runs *suites* of inter-dependent tasks
- designed for *cycling* systems, but supports non-cycling ones



→ Who are we?

Modelling Infrastructure Support Systems Team

Also known as: *MISS*, *MetOMI*, “the *FCM/Rose/Cylc team*”, ...

Joined Oct. 2018



Dave



Matt



Oliver



Sadie



Tim



Seconded

Lois

Wider Cylc Team

Lead developer:

Hilary Oliver



David Sutherland



Bruno Kinoshita



Martin
Ryan

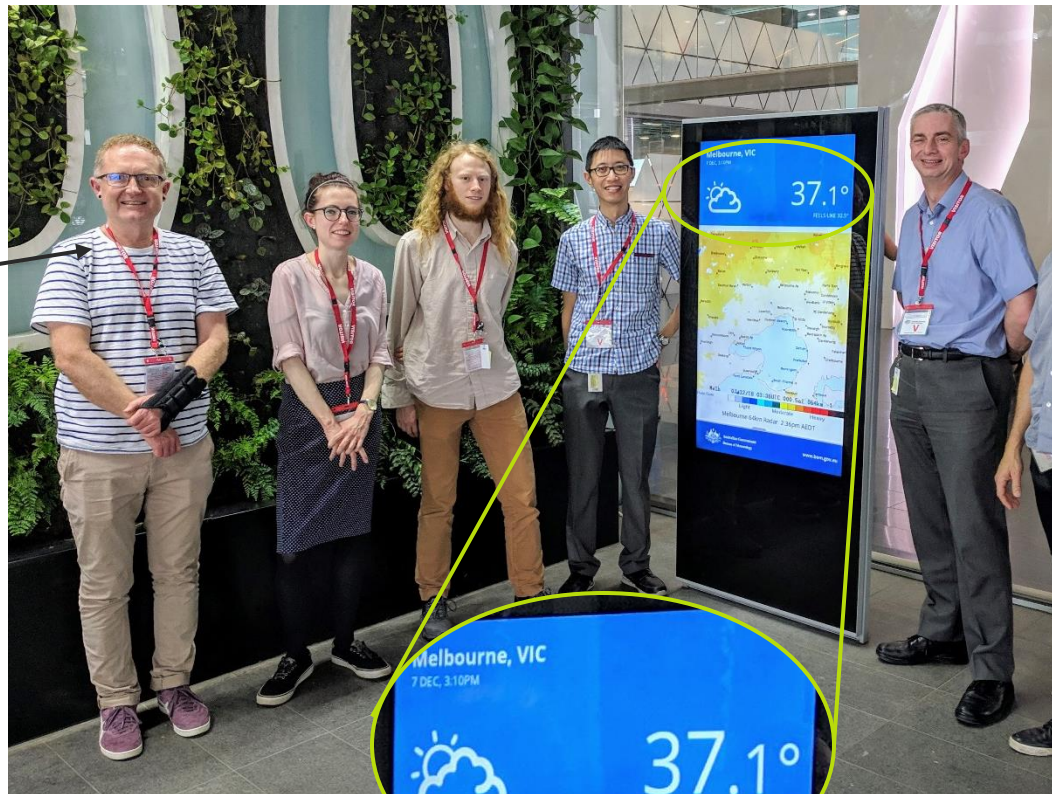


Altair

Sujata
Patnaik

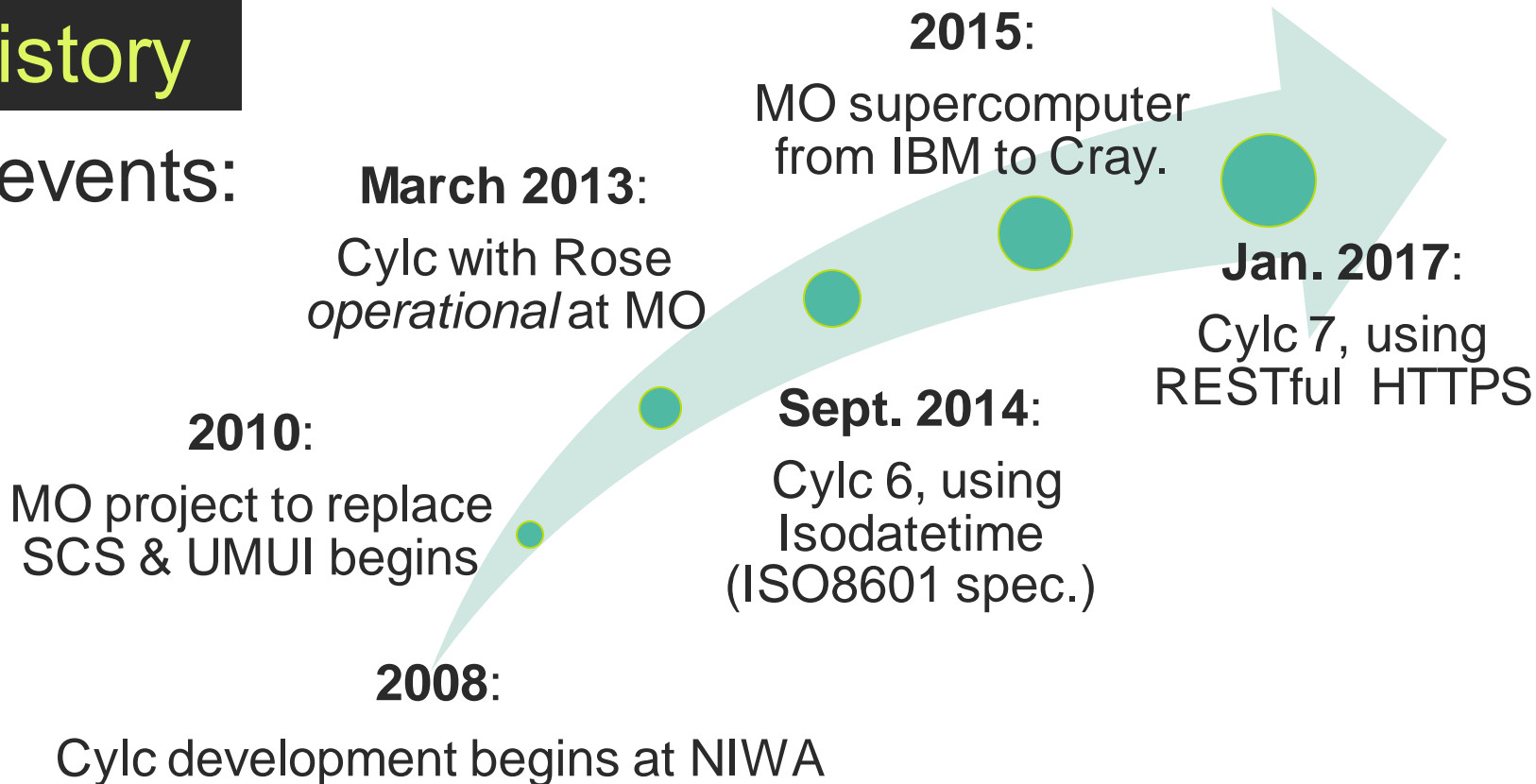


International development...



→ History

Key events:



The past year: a summary

An overview, & review of large-scale changes, from Feb. '18 to March '19

The year in figures:



2018.02.0 → 2019.01.1

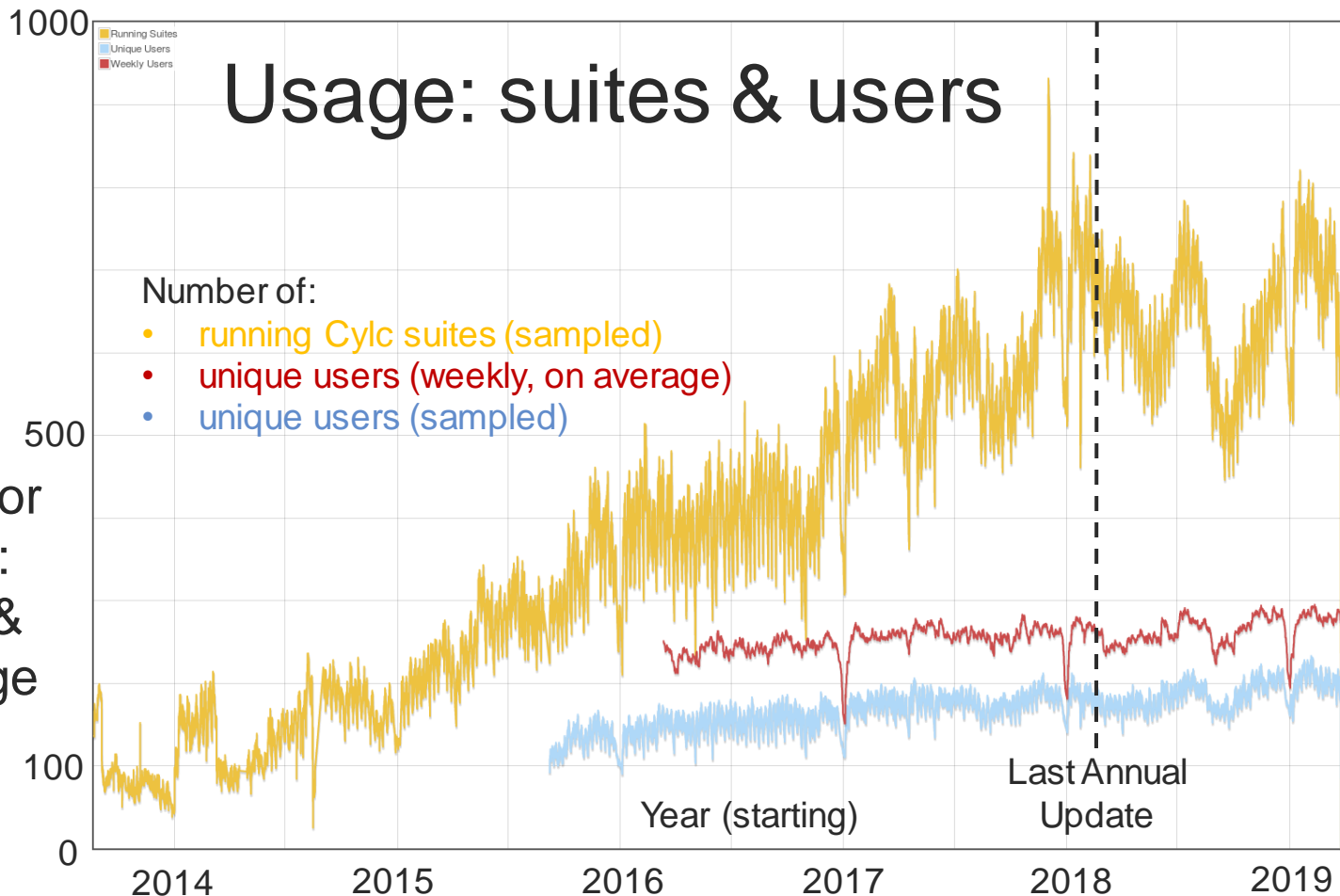
- **5** releases (+1 imminent)
- **89** merged pull requests
 - ~82 from the Met Office
- **491** commits
 - 471 from the Met Office
- **24** issues closed
- **10** contributors
 - 7 from the Met Office



7.6.0 → 7.8.2

- **7** releases (+1 imminent)
- **268** merged pull requests
 - ~120 from the Met Office
- **852** commits
 - 402 from the Met Office
- **111** issues closed
- **19** contributors
 - 8 from the Met Office

- General upward trend continuing...
- New records for user numbers:
233 sampled &
295 on average weekly



Rose → Cylc

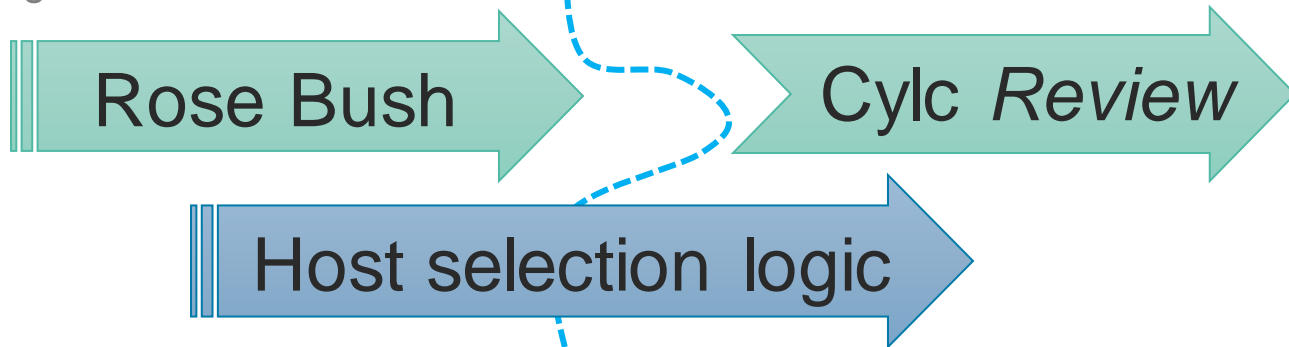
Functionality Migrations



- a framework for management of (application) *configurations*
- for use with Cylc to manage suites
 - further suite configuration & control



- a general-purpose workflow engine (*meta-scheduler*)
 - runs *suites* of inter-dependent tasks



Cylc Review

Cylc Review @ els032

uktrials.desktop mi-ay187 cycles list task jobs list broadcasts list cylc files rose files

log/rose-suite-run.conf
log/rose-suite-run.log
log/rose-suite-run.version
rose-suite.info

Rose Bush @ els032

uktrials.desktop mi-ay187 cycles list task jobs list broadcasts list cylc files rose files rose-suite.info

log/rose-suite-run.conf
log/rose-suite-run.log
log/rose-suite-run.version
rose-suite.info

Suite ■ is stopped, last activity 2 days ago

| task status | job status | cycle point | task name | job # | submit time | queue Δt | run Δt | job host | job name |
|-------------|------------|----------------|-----------------|--------|-------------|----------|--------|----------------------------|--------------------|
| ✓ succeeded | 🟢 | 20190325T0900Z | housekeep | 1 of 1 | 2 days ago | 0:01 | 0:01 | exvcylc12.metoffice.gov.uk | background |
| ✓ succeeded | 🟢 | 20190325T0900Z | ukv_archive | 1 of 1 | 2 days ago | 0:40 | 4:29 | xcslr0 | pbs[2080850.xcs00] |
| ✓ succeeded | 🟢 | 20190325T0900Z | ukv_um_recon_ls | 1 of 1 | 2 days ago | 1:26 | 3:18 | xcslr0 | pbs[2080372.xcs00] |

Rose files still recognised

log/rose-suite-run.conf
log/rose-suite-run.log
log/rose-suite-run.version
rose-suite.info

- Selects the “best” host on which to launch *suites* by default for `cylc run` & `cylc restart`
- Host considered best is configured under `global.rc` `[suite severs]` using metrics such as free memory
- To request a specific host, use `--host=` option

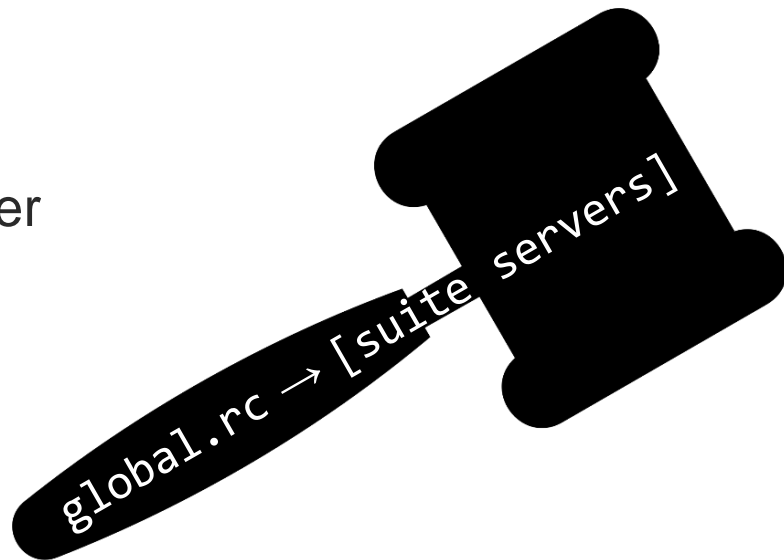
→ Cylc Docs Section 15.2.10.7



Migration Between Hosts

- Tell running suites to *automatically* migrate to another available host for server maintenance etc.
- Also set-up under global configuration

→ Cylc Docs Section 13.31



CONDEMNED

- New accessible format & styling
- *New material* including tutorials with practicals
- Contains a wealth of reference material




Table of Contents

1. Introduction: How Cylc Works
2. Cylc Screenshots
3. Installation
4. Cylc Terminology
5. Workflows For Cycling Systems
6. Global (Site, User) Configuration Files
7. Tutorial
8. Suite Name Registration
9. Suite Configuration
 - 9.1. Suite Configuration Directories
 - 9.2. Suite.rc File Overview
 - 9.2.1. Syntax
 - 9.2.2. Include-Files
 - 9.2.3. Syntax
 - 9.2.4. Highlighting For Suite Configuration
 - 9.2.4. Gross File Structure
 - 9.2.5. Validation
 - 9.3. Scheduling - Dependency Graphs
 - 9.3.1. Graph String Syntax
 - 9.3.2. Interpreting Graph Strings
 - 9.3.3. Graph Types

Here's a more complex graph using two parameters ([runtime] omitted):

```
[cylc]
[[parameters]]
  run = 1..2
  mes = cat, dog
[[scheduling]]
[[dependencies]]
  graph = """prep => init<run> => model<run,mes> =>
    post<run,mes> => wrap<run> => done"""
```

Fig. 32 shows the result as visualized by cylc graph.

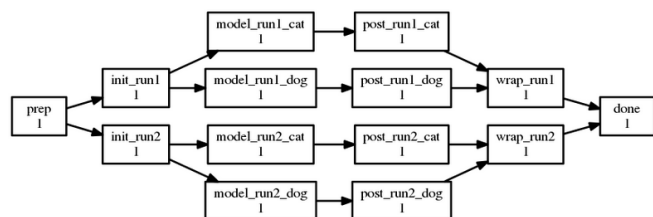


Fig. 32 Parameter expansion example.

9.5.3.1. Zero-Padded Integer Values

Integer parameter values are given a default template for generating task suffixes that are zero-padded according to the longest size of their values. For example, the default template for $p = 9..10$

Rose Documentation
2019.01.0

Search docs

USER GUIDE

- Installation
- Getting Started
- Cylc Tutorial

Rose Tutorial

- Rose Configurations
- Rose Applications
 - Application Configuration
 - Running Rose Applications
- Rose Metadata
- Rose Suite Configurations
- Rosie
- Summary
- Further Topics

Cheat Sheet

Glossary

ROSE API REFERENCE

- Rose Configuration
- Command Reference
- Rose Environment Variables
- Rose Bash Library
- Rose Built-In Applications

Running Rose Applications

An application can be run using the `rose app-run` command:

```
$ rose app-run -q # -q for quiet output
Hello Earth!
```

The Rose application will by default run in the current directory so it is a good idea to run it outside of the `application directory` to keep run files separate, using the `-C` option to provide the path to the application:

```
$ rose app-run -q -C path/to/application
Hello Earth!
```

Practical

In this practical we will convert the `forecast` task from the `weather-forecasting` suite into a Rose application.

Create a directory on your filesystem called `rose-tutorial`:

```
mkdir ~/rose-tutorial
cd ~/rose-tutorial
```

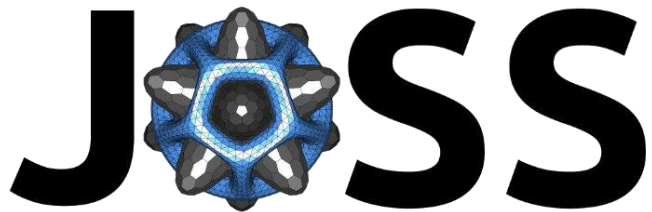
1. Create a Rose application

Create a new directory called `application-tutorial`, this is to be our `application directory`:

```
mkdir application-tutorial
cd application-tutorial
```

2. Move the required resources into the `application-tutorial` application

Research publications



[Journal of Open Source Software]

Cylc: A Workflow Engine for Cycling Systems

Oliver et al. (2018)

Published

*Workflow Automation for Cycling
Systems: The Cylc Workflow Engine*

Oliver et al. (2019)

Accepted

computing
in **SCIENCE & ENGINEERING**

The past year: specific updates

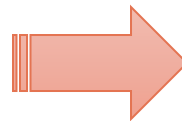
Assorted recent changes we'd like to highlight

→ Goodbyes

rose suite-hook

- Command long deprecated, will be made *obsolete soon*
- For common event hooks, use Cylc's built-in event handlers instead

~~rose suite-hook <EVENT> <SUITE> <MESSAGE>~~



<suite-event-handler> %(<EVENT>)s %(<SUITE>)s %(<MESSAGE>)s

× Rose Docs: To be removed

→ Cylc Docs Section 15.1.3.13

- Specify ICPs relative to the current date-time, `now`, using `next()` & `previous()`

- For example, to set ICP to be...

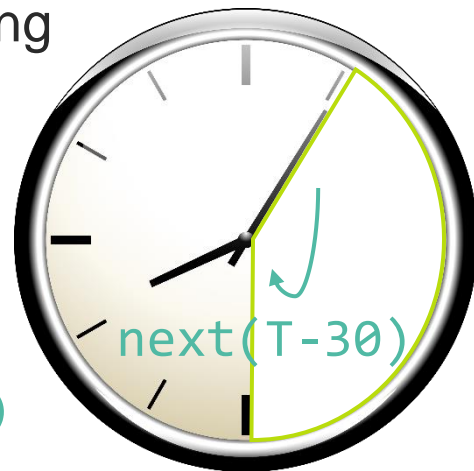
- the *next* time it is half past some hour:

```
[scheduling]
```

```
initial cycle point = next(T-30)
```

- the last (most *previous*) time it was 4 pm:

```
initial cycle point = previous(T16:00)
```

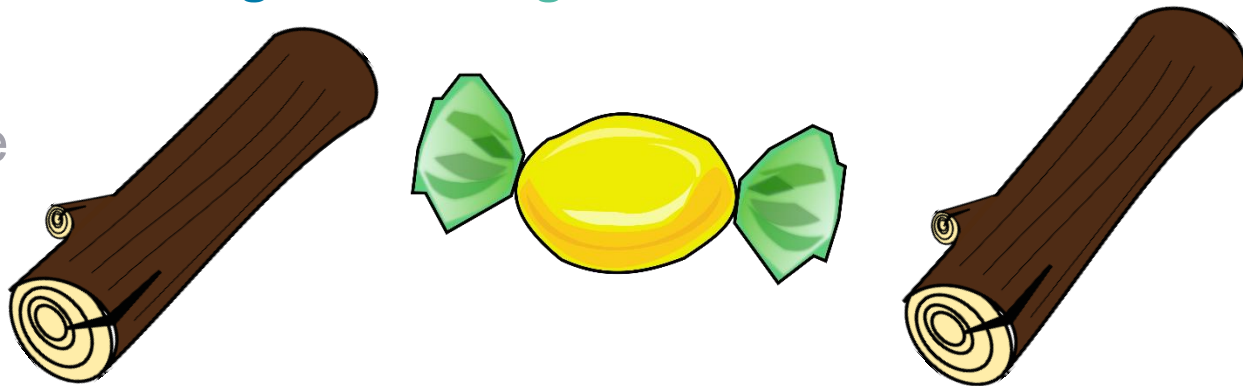


→ Enhancements

Improved suite logging

- suites no longer generate a `~/cylc-run/SUITE/log/suite/err` file
- all suite logs go to `.../log/suite/log`

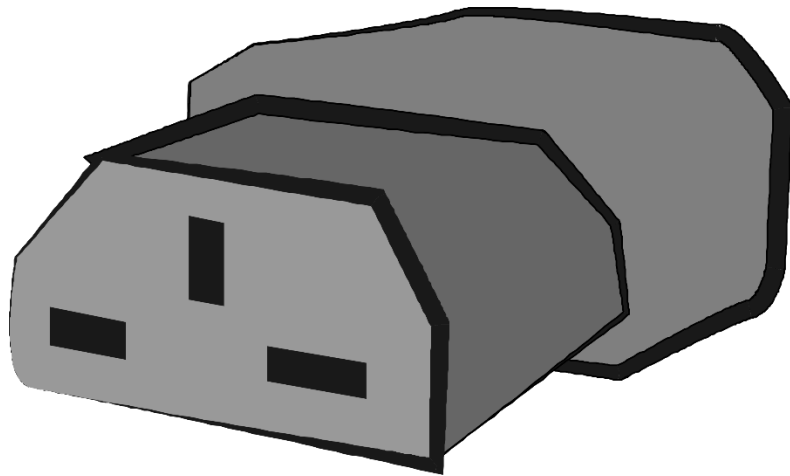
It's all there
in the:



- suite logs also contain more information on restart & roll

Trigger functions: *xtriggers*

- Trigger tasks off of anything by calling arbitrary custom functions
- Just normal Python functions
- Function return values should be...
 - if the trigger condition is satisfied:
`return (True, results)`
 - if it is not:
`return (False, {})`



→ Cylc Docs Section 12.3

Enhanced Jinja2 support



- pure Python modules can now be imported, like template modules
- [Jinja2Tests](#) for custom [is](#) tests & [Jinja2Globals](#) for custom global variables, complementing existing custom filters [Jinja2Filters](#)

→ [Cylc Docs Section 9.6](#)

+ [Official Jinja2 Docs](#)