IS-ENES3 Deliverable D8.4

Cylc / Rose Development Summary

Reporting period: 01/01/2022 – 31/03/2023

Authors: David Matthews (Met Office)
Reviewer(s): Hilary Oliver (NIWA), Jean-Christophe Rioual (Met Office)

Release date: 24/06/2022

ABSTRACT

Cylc is a general-purpose workflow engine that orchestrates cycling workflows very efficiently. Rose is designed for use with Cylc and is primarily a toolkit for writing, editing and running application configurations.

Major upgrades to Cylc and Rose are about to be released. They ensure the long-term viability of the tools and, more importantly, implement new features which will considerably enhance the experience of the expanding user base.

Dissemination Level		
PU	Public	X
CO	Confidential, only for the partners of the IS-ENES3 project	

Revision table			
Version	**Date**	**Name**	**Comments**
Release for review	20/06/2022	David Matthews	
Release for 2nd review	21/06/2022	David Matthews	
Final version	24/06/2022	David Matthews	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084

Table of contents

Executive Summary

Cylc is a general-purpose workflow engine that orchestrates cycling workflows very efficiently. Rose is designed for use with Cylc and is primarily a toolkit for writing, editing and running application configurations.

Cylc 8 and Rose 2 are major new releases which have been under development since the start of IS-ENES3 and are expected to be finalised in July 2022. They mark the move to Python 3 which is essential to ensure the long-term viability of the tools. However, more importantly, the new features implemented in these releases (e.g., web UI, workflow installation, platforms, spawn on demand, optional outputs) make them an even more compelling choice and will considerably enhance the experience of the expanding user base.

1. Objectives

Cylc is a general-purpose workflow engine that orchestrates cycling workflows very efficiently **[1]**. It was designed for use in production weather, climate, and environmental forecasting systems, but it is not specialized to those domains and is agnostic to the applications it manages. Rose is designed for use with Cylc and is primarily a toolkit for writing, editing, and running application configurations.

As an example of application, all CMIP6 delivery activities at the Met Office have been underpinned by Cylc and Rose, from scheduling the models on supercomputer platforms to post-processing and dissemination to ESGF.

This report describes the work done since the start of IS-ENES3 to further develop Cylc and Rose to ensure they continue to meet user requirements; increasing their usability, responding to changing technologies and improving performance.

2. Methodology and Results

2.1 Requirements / Planning

The agreed development priorities were reported in Milestone 8.2 (https://is.enes.org/documents/milestones/m8-2-cylc-rose-development-priorities-agreed/view). The main priority was the release of Cylc 8 and Rose 2 to address a number of key issues:

1. Migrate from Python 2 to 3.
2. Replace the obsolete PyGTK based Cylc GUIs with a new web UI.
3. Migrate the Rose functionality for running Cylc workflows ("rose suite-run") into Cylc.

The intention was to have the new releases ready by early 2021. Whilst the first beta release was available in March 2021, the first formal release of Cylc 8 has taken much longer than originally hoped. At the time of writing we expect Cylc 8.0 to be available in July 2022. There are numerous reasons for the delay including resourcing difficulties, impact of the pandemic and under-estimating the complexity of some of the changes. However, the most significant reason is that the scope of the improvements delivered in Cylc 8 has expanded. In particular, a new scheduling algorithm has been implemented which should dramatically improve the performance and scalability of Cylc (motivated by the fact that the old algorithm was making it hard to display all the tasks in the web UI). Therefore, whilst the delays have been frustrating, it means that the improvements delivered with Cylc 8 are much greater than originally anticipated.

The original plans for Cylc 8 were formed at a workshop held in Melbourne, December 2018 (see https://cylc.github.io/cylc-admin/dec-workshop-report). The Cylc developers gathered for a further workshop in Wellington, February 2020 (see https://cylc.github.io/cylc-admin/feb2020-workshop-report) which allowed us to review progress and plan the remaining work. We were fortunate to be able to hold this workshop just before the global pandemic took hold.

2.2 Resourcing

The development resource provided as part of WP8 task 5 amounts to 24 person months. This contributes to the wider effort working on Cylc and Rose which currently consists mainly of staff at the Met Office and NIWA. This report describes the output of the entire Cylc / Rose development effort, not just that funded by IS-ENES3.

To give an idea of the number of contributors, GitHub reports that 19 different people have contributed to the cylc-flow repository since the beginning of 2019 (see https://github.com/cylc/cylc-flow/graphs/contributors?from=2019-01-01&to=2022-06-13&type=c). Note that Cylc is split across a number of repositories but cylc-flow (the scheduler and CLI) is the most active.

2.3 Key Developments

Python 3

Porting Cylc and Rose to Python 3 was relatively straightforward and was completed very early in the development. However, the GUIs used by both Cylc and Rose relied on an obsolete technology (PyGTK) not available with Python 3 so porting to Python 3 required removing the old GUIs. Implementing a replacement, web based, GUI, and the underlying architecture to support it, is the main reason the new releases have required so much work.

Web UI

The old Cylc PyGTK GUIs have been replaced with a new web UI. A new architecture was agreed at the Melbourne workshop to support the web UI (see https://cylc.github.io/cylc-doc/latest/html/reference/architecture/index.html). We take significant advantage of software provided by Project Jupyter (see https://jupyter.org/). In particular:

- The Cylc UI server (which serves the UI to the user's browser) is implemented as a Jupyter Server extension (see https://jupyter-server.readthedocs.io/).
- Cylc Hub (which provides multi-user access to the Cylc UI) is implemented using JupyterHub (see https://jupyterhub.readthedocs.io/).

The Cylc UI provides the ability to control and modify a user's workflow, so security is extremely important. These widely used technologies make it easier to secure the Cylc UI and easier for organisations to understand how this security is achieved.

The UI itself is written in Javascript using the Vue.js framework (see https://vuejs.org/). An example of the new UI is shown in figure 1.

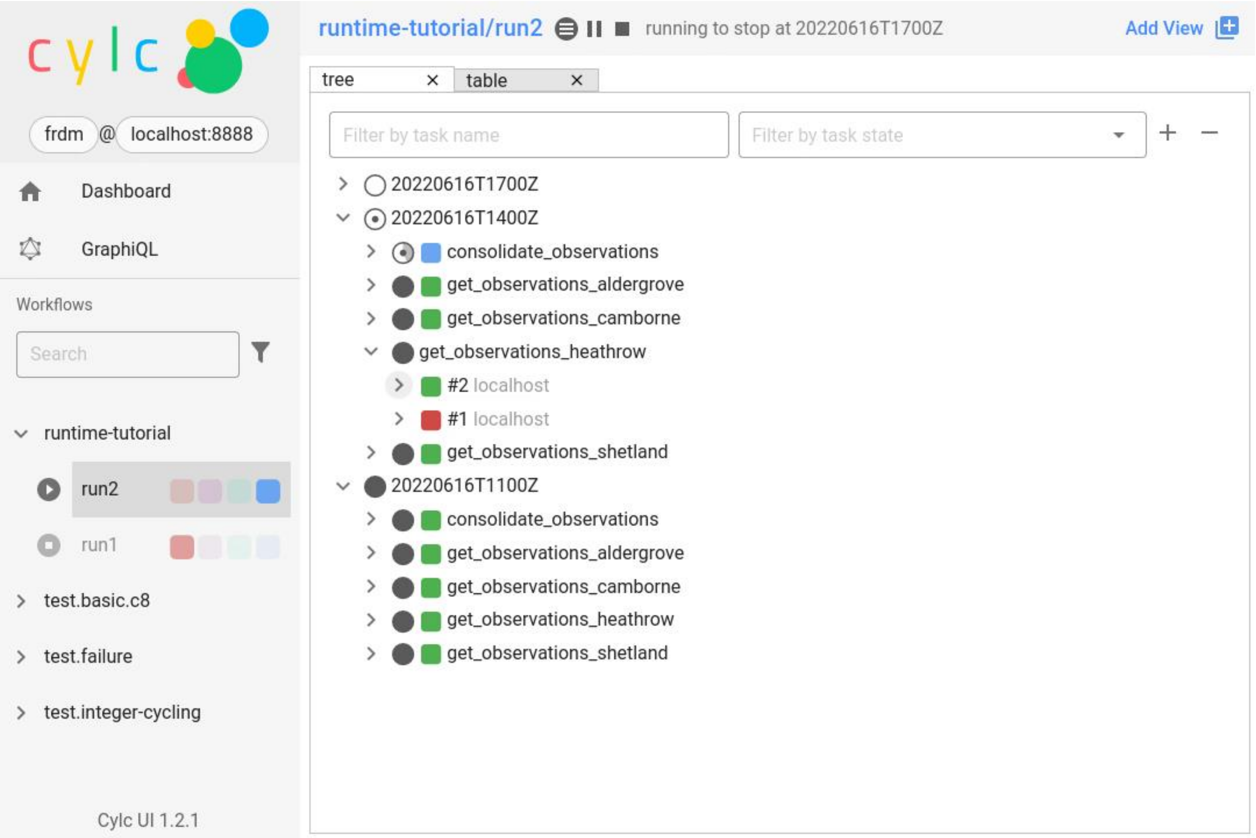
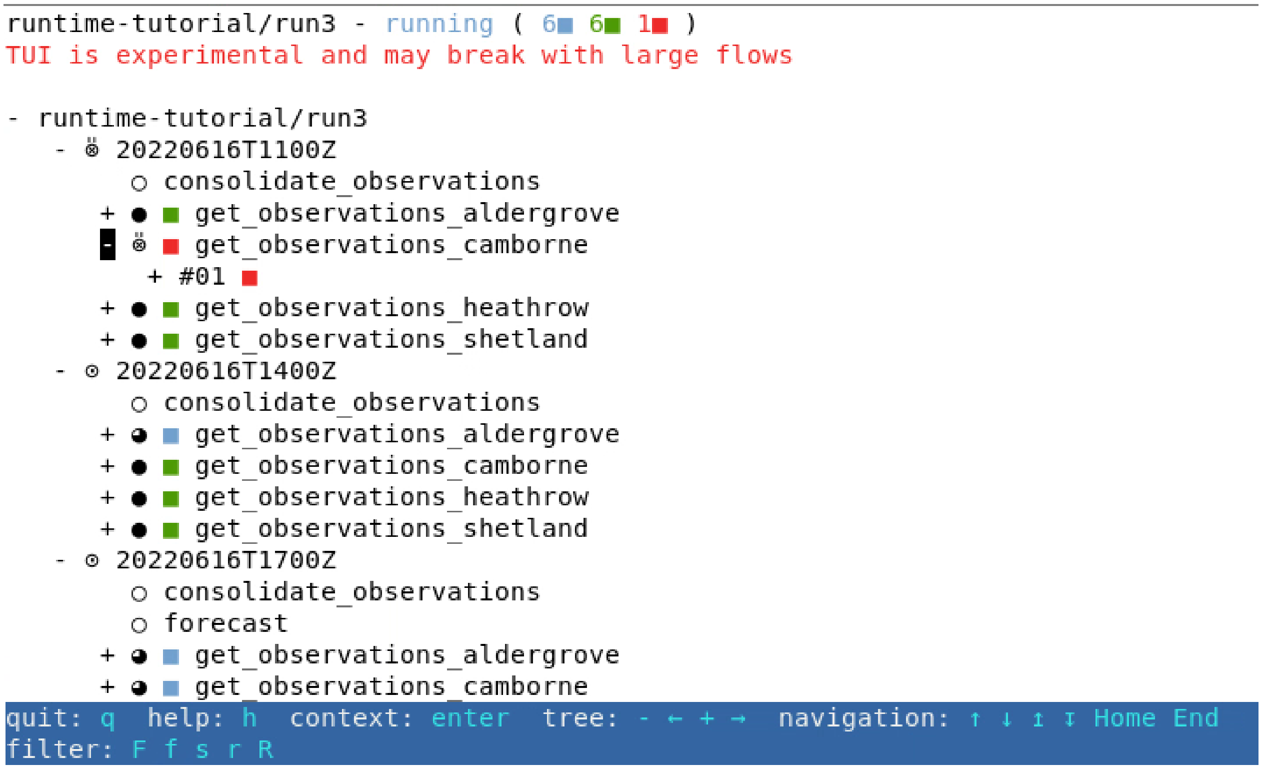


Figure 1- Cylc Web UI

Terminal UI

As well as having a new web UI, Cylc 8 also comes with a new Terminal UI (see figure 2 for an example). This makes use of the Urwid library (see https://urwid.org/) to provide a UI which supports most of the features of the Web UI. This should prove very useful for users not able to use the web UI.

Note that this feature is currently experimental – further work is planned to add more features and ensure it works with larger workflows.



```
runtime-tutorial/run3 - running ( 6🟦 6🟩 1🟥 )
TUI is experimental and may break with large flows

- runtime-tutorial/run3
   - ⊗̈ 20220616T1100Z
        ○ consolidate_observations
      + ● 🟩 get_observations_aldergrove
      - ⊗̈ 🟥 get_observations_camborne
         + #01 🟥
      + ● 🟩 get_observations_heathrow
      + ● 🟩 get_observations_shetland
   - ⊙ 20220616T1400Z
        ○ consolidate_observations
      + ◕ 🟦 get_observations_aldergrove
      + ● 🟩 get_observations_camborne
      + ● 🟩 get_observations_heathrow
      + ● 🟩 get_observations_shetland
   - ⊙ 20220616T1700Z
        ○ consolidate_observations
        ○ forecast
      + ◕ 🟦 get_observations_aldergrove
      + ◕ 🟦 get_observations_camborne
quit: q  help: h  context: enter  tree: - ← + →  navigation: ↑ ↓ ↥ ↧ Home End
filter: F f s r R
```

Figure 2- Cylc Terminal UI

Platforms

Cylc 7 only had support for running jobs on individual remote hosts (although it was possible to select from a set of hosts via a Rose utility). Cylc 8 now supports platforms which consist of a group of hosts (typically login nodes for a compute cluster) and a shared “job runner” (batch system). This reduces the number of ssh connections made by Cylc and, more importantly, allows everything to continue working correctly should one of the login nodes become unavailable. It has also allowed us to add support for defining a set of default batch scheduler directives to be used on a platform.

Installing and running workflows

Many users of Cylc 7 relied on utilities provided by Rose to install and run their workflows. Considerable effort has gone into migrating this functionality into Cylc. This will ensure all Cylc users work in the same way which makes training and support much easier. It has also been possible to implement many enhancements as part of the migration now that the functionality is properly integrated into Cylc. The main features are as follows:

- A new `cylc install` command for installing workflows from a source directory into the run directory.
- Workflow source files (e.g., scripts, configuration files) are automatically installed on remote platforms at run time.
- Workflow directories can be configured to use different file systems via symbolic links (e.g., to allow workflow data files to use a different file system than the log files).
- A new default run directory hierarchy is used, with each installation using a separate run directory. This makes it easier to refer back to previous runs and also reduces the danger of accidently using or overwriting files from a previous run.
- A new `cylc clean` command to clean up old runs, on remote job platforms as well as locally
- The functionality previously provided by Rose via the `rose-suite.conf` file (file installation, template variables, etc) continues to be supported via a plugin.

Spawn on Demand

A new scheduling algorithm has been implemented which we refer to as spawn on demand (see https://cylc.github.io/cylc-admin/proposal-spawn-on-d.html). It should dramatically improve the performance and scalability of Cylc since the pool of tasks being managed by Cylc is typically much smaller. There are also many other advantages – see the proposal for details.

Required and Optional Outputs

The new scheduling algorithm led on to us adding support for required and optional outputs (see https://cylc.github.io/cylc-admin/proposal-new-output-syntax.html). This is a new graph syntax which allows Cylc to know whether a task output is required or optional. For example, in many cases tasks are required to succeed for the workflow to complete successfully. However, in other cases it may be fine for a task to fail (e.g., the graph includes a branch to automatically handle the task failure). The new syntax allows Cylc to know whether a particular task output is required and immediately flag a problem if a task finishes without completing all its required outputs (with Cylc 7 you could flag task failures but had no way to know whether the failures would prevent workflow completion).

Documentation

The Cylc User Documentation (see https://cylc.github.io/documentation/) has undergone a major overhaul in preparation for the release of Cylc 8. It includes a Migration Guide designed for users of Cylc 7 (and Rose 2019) who are upgrading to Cylc 8.

Installation

Installation of Cylc and Rose is now much easier due to the adoption of modern packaging systems (Conda and Pip), a reduction in the reliance on the system environment and the removal of the dependency on PyGTK. We make extensive use of GitHub Actions to automate as many aspects of the release process as possible.

Other Developments

This report only highlights the most significant developments. Many other enhancements have been made to Cylc over the period, some of which are listed in the cylc-flow release notes (see https://github.com/cylc/cylc-flow/blob/master/CHANGES.md).

2.4 Releases

Releases of the evolving Cylc 8 code have been made on a regular basis in order to allow testing and feedback by early adopters. The first alpha release was made in September 2019, the first beta in March 2021 and the first release candidate in February 2022. We expect Cylc 8.0 to be formally released in July 2022.

Whilst the focus has been on Cylc 8, we have continued to support Cylc 7 and have made regular minor releases throughout the period.

2.5 Future Work

Whilst the work of preparing the first release of Cylc 8 is almost complete, there remains further work to do, for example:

- The “graph view” (a widely used feature of the Cylc 7 GUI) has yet to be implemented.
- Support for viewing job log files (including historical tasks) needs to be added.
    - We have updated “cylc review” in Cylc 7 to support viewing Cylc 8 workflows as a short-term workaround for sites still able to install Cylc 7.
- Replacements for the Rose GUIs (“rose edit” and “rosie go”) need to be developed.
- The new web-based UI (and the new architecture) is much easier to develop that the old PyGTK based GUIs. We have many plans for future enhancements as time permits.
- Other enhancements identified in the agreed development priorities (Milestone 8.2), for example improved housekeeping and support for running multiple tasks in a single job.

3. Conclusions and Recommendations

The release of Cylc 8 and Rose 2 mark a major step forward. They mark the move to Python 3 which is essential to ensure the long-term viability of the tools. However, more importantly, the new features implemented in these releases (e.g. workflow installation, platforms, spawn on demand, optional outputs) make them an even more compelling choice and will considerably enhance the experience of the expanding user base.

We recognise that reliance on Python 2 had become a significant barrier to adoption of Cylc. Now that barrier has been removed it is an excellent time for new users to take a look at what Cylc can offer. We encourage use of the Cylc Forum (https://cylc.discourse.group/) for asking questions or seeking advice.

Whilst these new releases are a major step forward there remains much to be done. The resources funded by IS-ENES3 will continue to contribute towards this important work for the remainder of the project. The work funded by IS-ENES3 will make its way into operational numerical weather prediction systems at the Met Office, NIWA and at other sites, promoting innovation.

References

[1] Workflow Automation for Cycling Systems: The Cylc Workflow Engine**,** H. Oliver et al**,** Computing in Science & Engineering Vol 21, Issue 4, July/Aug 2019. DOI: 10.1109/MCSE.2019.2906593