National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

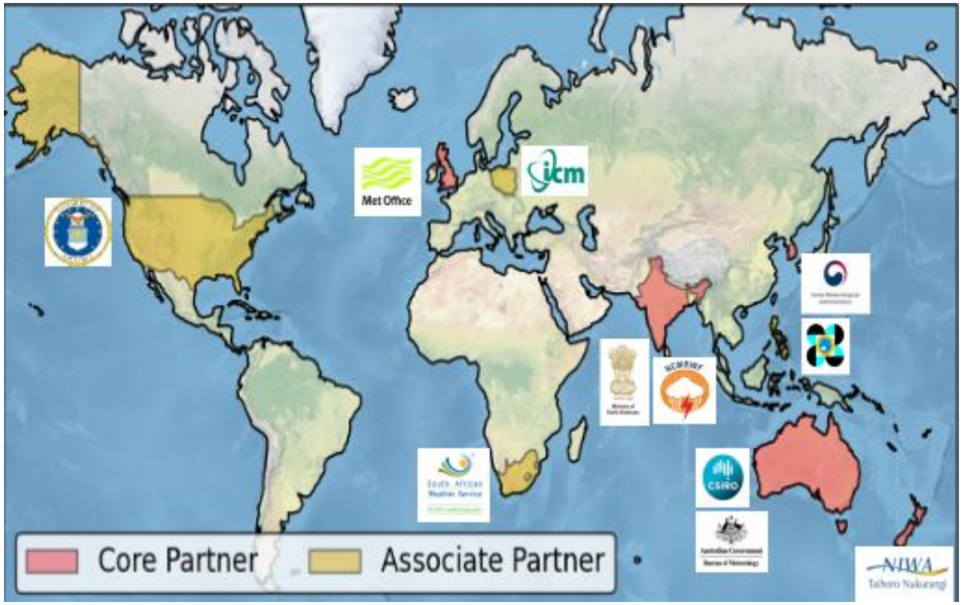
NCAS
Unified Model Introduction

Part 1: Overview of the UM system

University of Leeds, 7-9 February 2023

The Unified Model

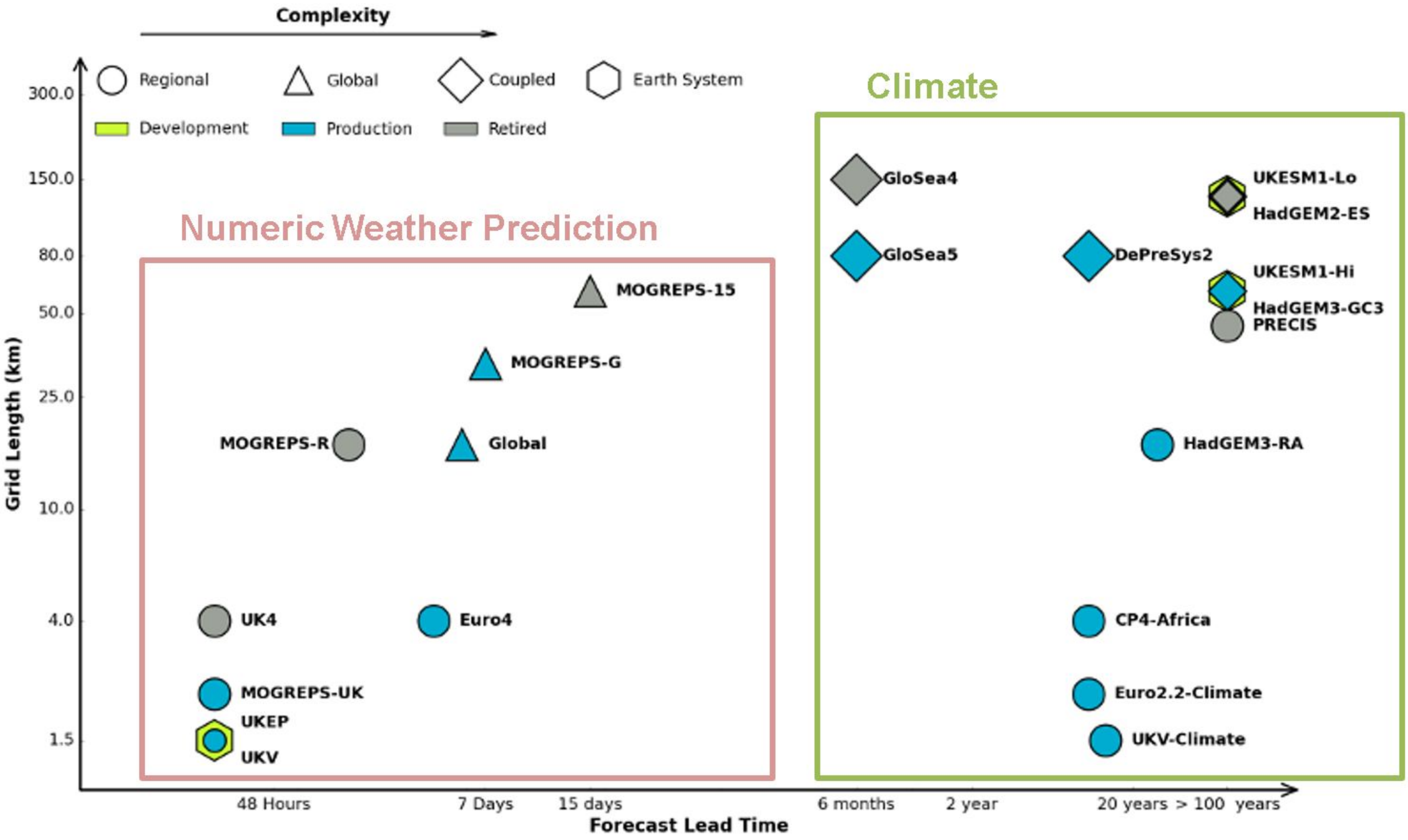
- The UM is a ***numerical modelling system,*** developed by the UK Met Office, and used for operational weather forecasting and climate prediction.
- It is used by the UK academic community for research. There are collaborations between the Met Office and the academic community for research and development (JWCRP and MOAP).
- It is used by forecast centres and climate agencies around the world.



Seamless modelling

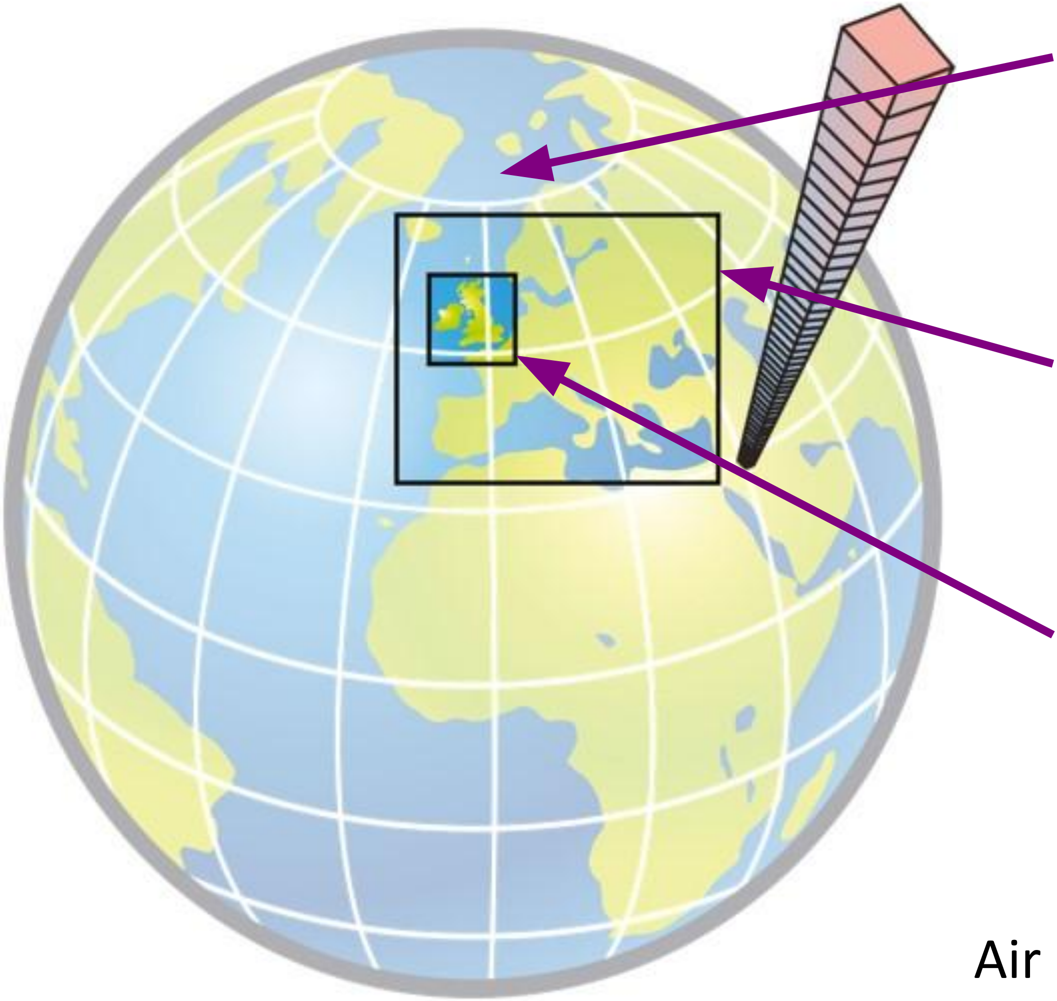
- The same core model is used across spatial and temporal scales:
  - Weather to climate timescales
  - Global and regional models
  - High to low resolution (horizontal and vertical)
  - Various model heights
- It can be used in atmosphere only mode, or coupled to:
  - NEMO ocean and CICE sea-ice via OASIS coupler
  - UM ocean for vn6.6.3 or earlier (e.g. HadCM3, HadGEM2)
  - UKCA chemistry and aerosols
  - JULES land-surface
- It can also be used in other modes, including:
  - Single Column Model (SCM)
  - Aquaplanet
  - Exoplanet

Unified Model family



Global & regional NWP system

Parallel suite 41



© Crown copyright Met Office

Global

- Deterministic: (10 km L70)
- Ensemble: (20 km L70)
- Data assimilation

Euro4

- 4 km 70 levels
- No data assimilation

UKV

- Deterministic (1.5 km L70)
- Ensemble (2.2 km L70)
- Data assimilation

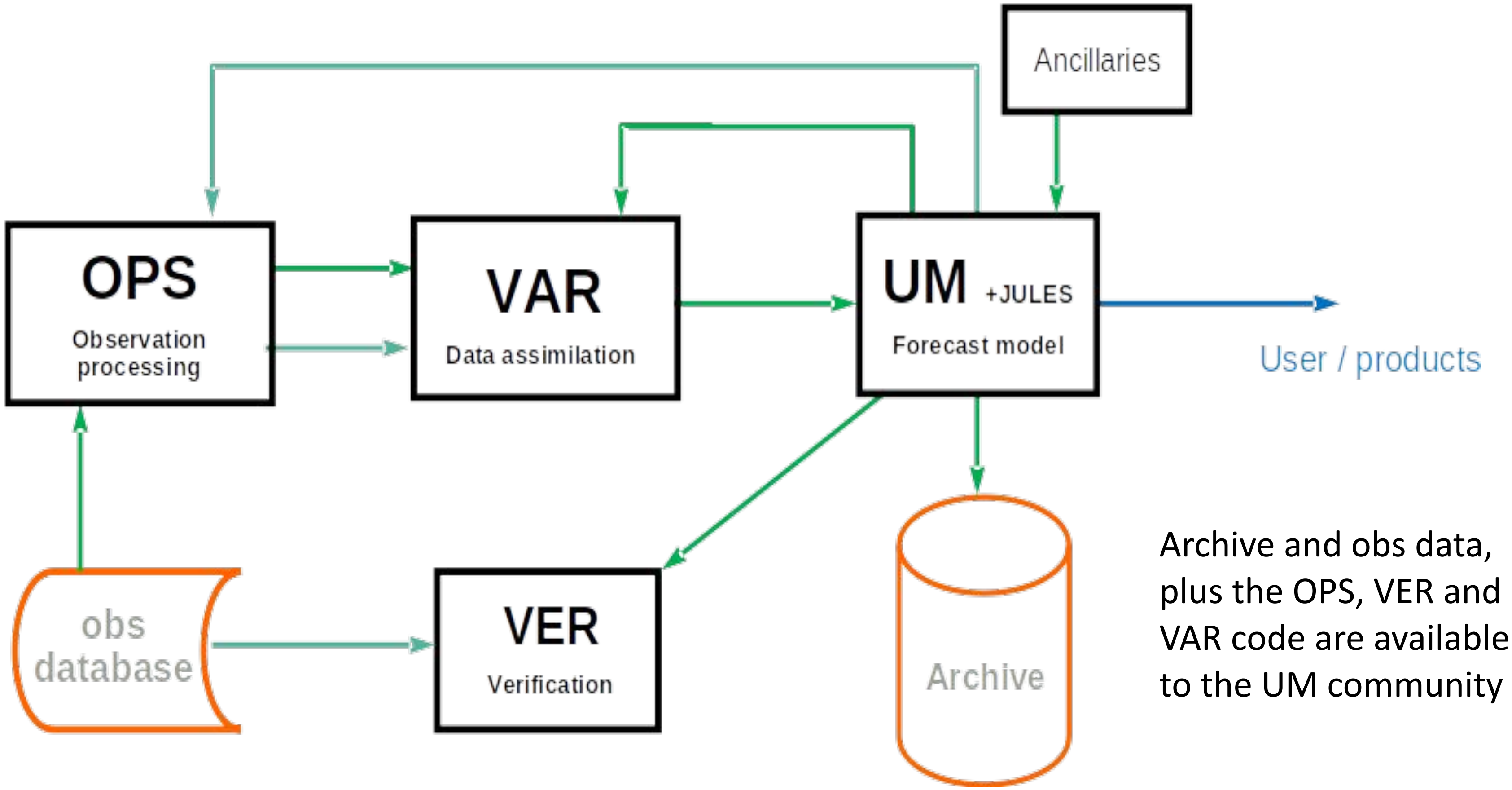
Air quality (AQUM)

- UK + NW Europe

NWP at the Met Office

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

The UM is just one part of a larger forecast system, e.g.



Archive and obs data, plus the OPS, VER and VAR code are available to the UM community

Nesting suite

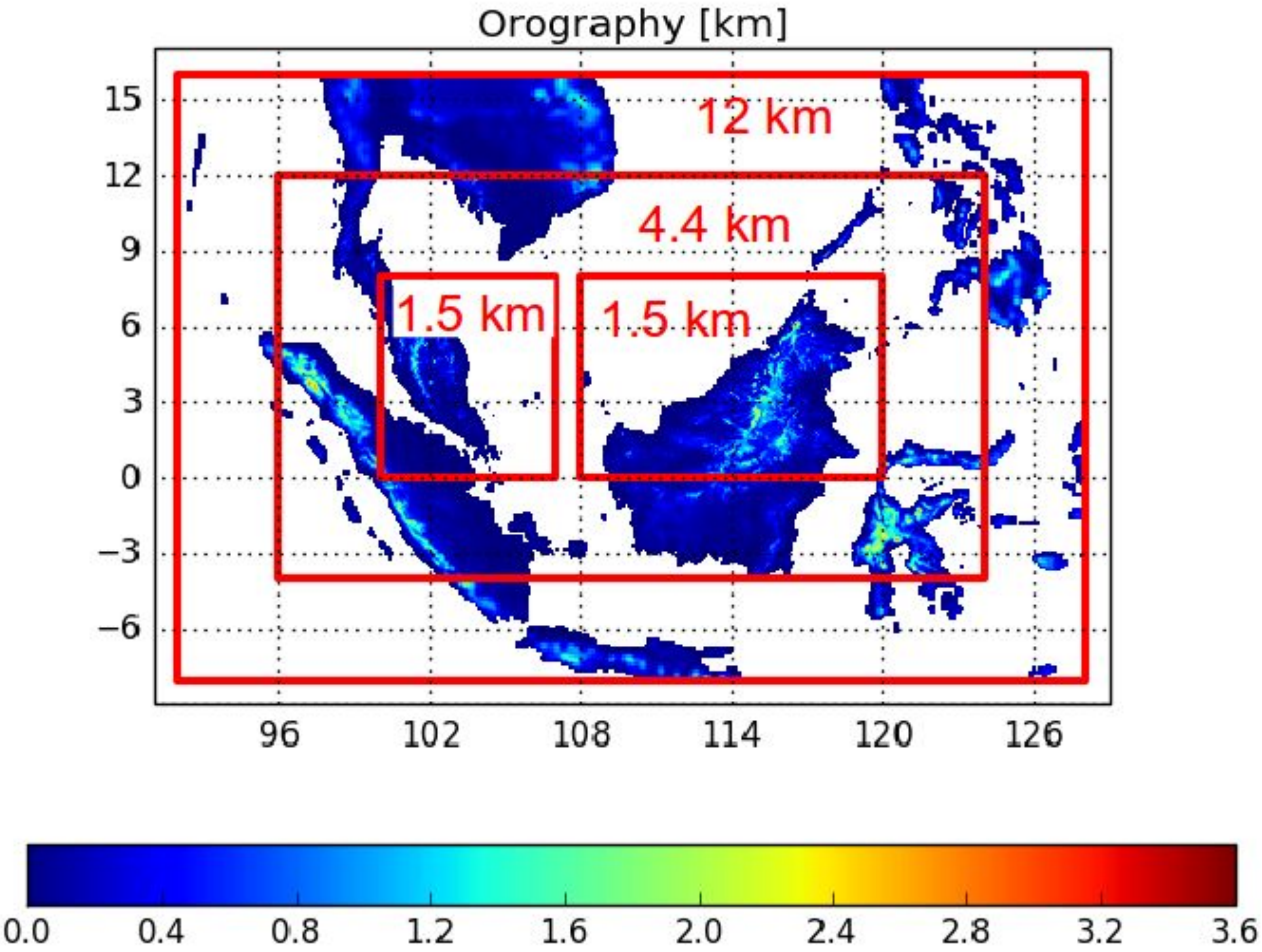
Example:
Newton funded
Malaysia project.

GL→ 12 → 4.4 →1.5

GL → 12 ——→ 1.5

GL ——→ 4.4 → 1.5

GL ——→ 1.5



Global resolutions

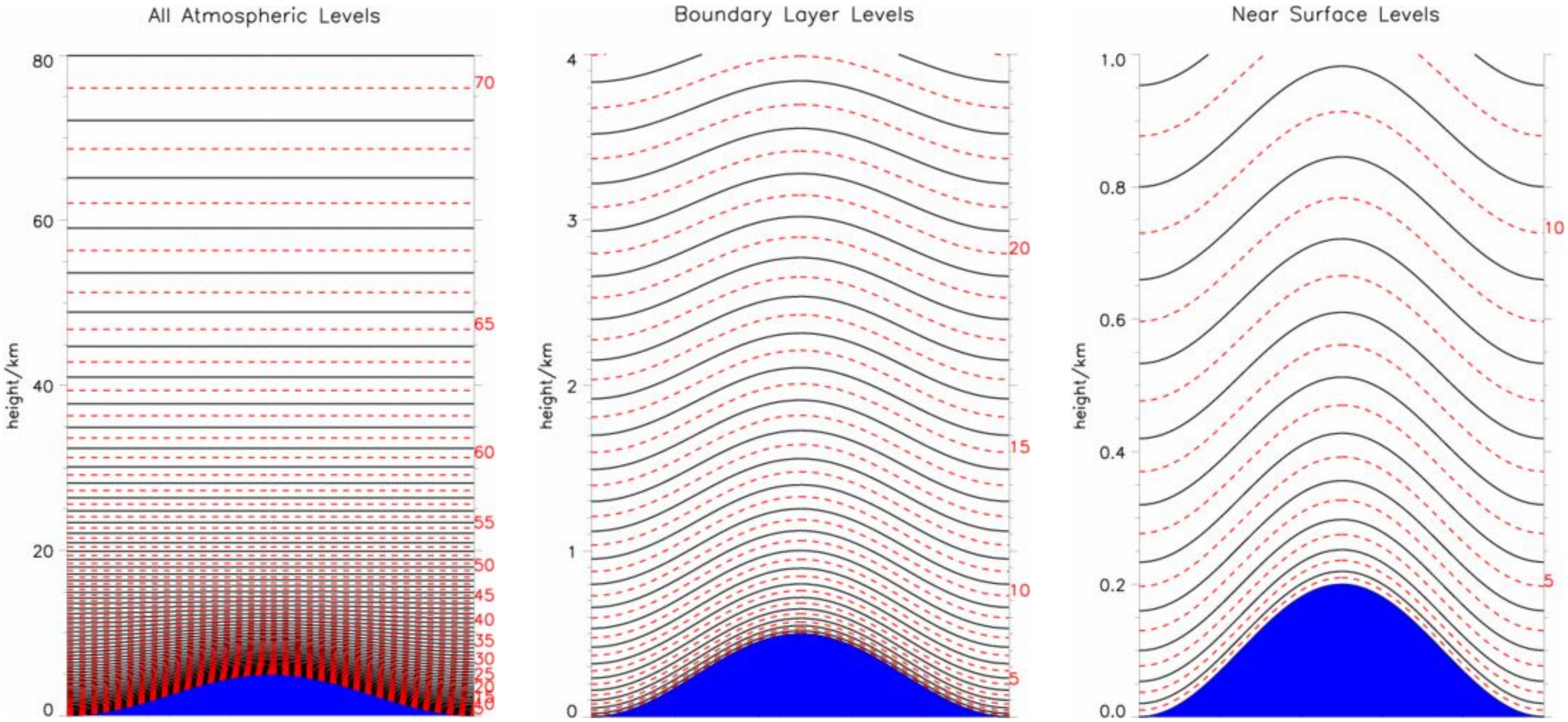
Standard fixed resolutions:

Resolution	Grid points	Grid cell size	Spacing at mid-latitudes
N48	96 x 73	3.75° x 2.50°	~270 km
N96	192 x 145	1.88° x 1.25°	~135 km
N144	288 x 217	1.25° x 0.83°	~90 km
N216	432 x 325	0.83° x 0.56°	~60 km
N320	640 x 481	0.56° x 0.38°	~40 km
N512	1024 x 769	0.35° x 0.23°	~25 km
N768	1536 x 1152	0.23° x 0.16°	~17 km
N1280	2560 x 1920	0.14° x 0.09°	~10 km

Vertical resolutions

Standard resolutions, defined by model levels and height.

Example: L70 with top at 80 km



Global science configurations

- Common scientific configurations used across scales (weather, seasonal and climate):
  - GA = Global Atmosphere (UM)
  - GO = Global Ocean (NEMO)
  - GSI = Global Sea Ice (CICE)
  - GL = Global Land (JULES)
  - GC = Global Coupled (all of the above coupled together)

- These are ongoing developments with fixed releases.
- Documented on the Global Model Evaluation and Development (GMED) pages:

https://code.metoffice.gov.uk/trac/gmed/wiki

HadGEM3(-AO)

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

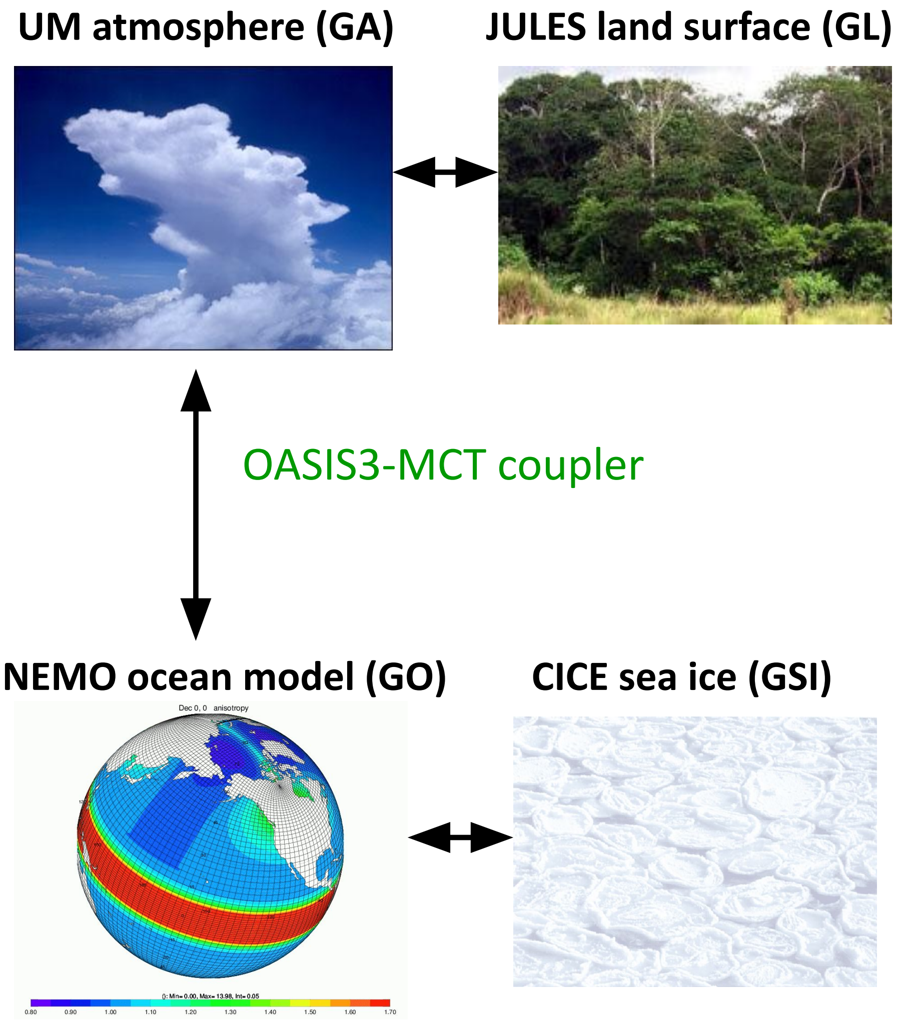
Under continuous development. There are versions with:

- different code releases
- different scientific configurations (latest GC5)

Multiple resolutions:

- N96-ORCA1/ORCA0.25
- N216-ORCA0.25
- N512-ORCA0.25/ORCA12

Can run atmosphere-only



UM atmosphere (GA)

JULES land surface (GL)

OASIS3-MCT coupler

NEMO ocean model (GO)

CICE sea ice (GSI)

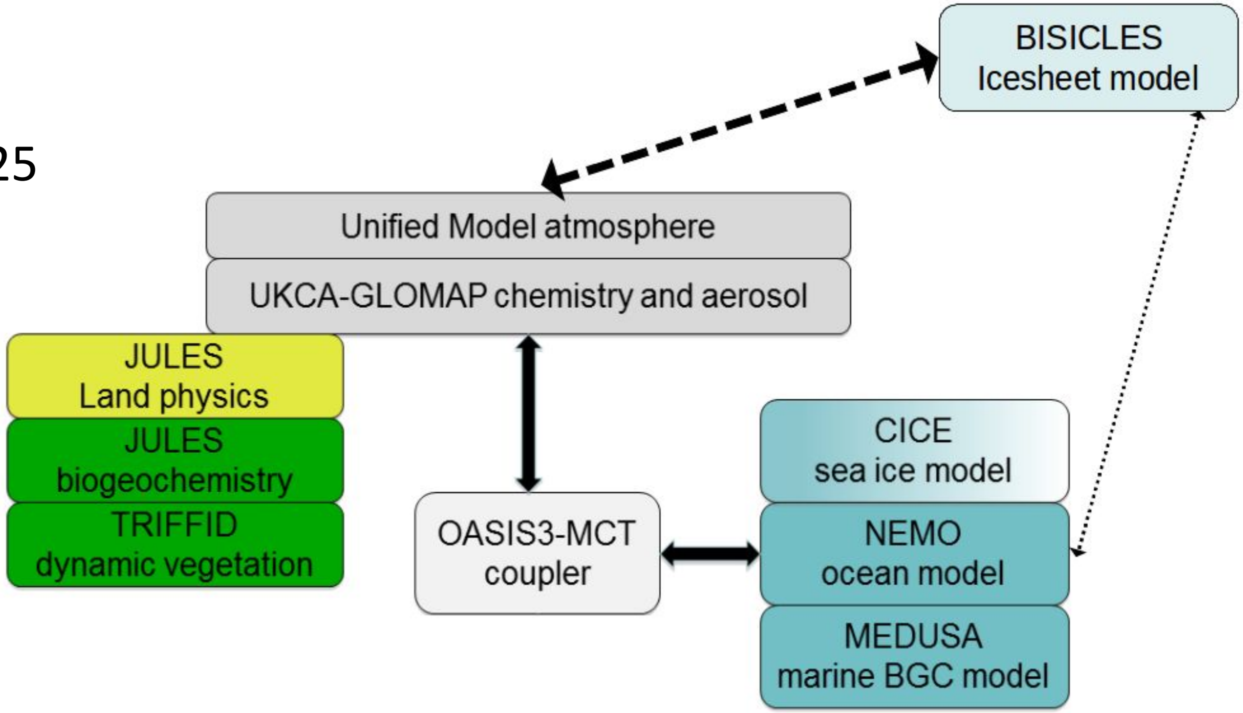
(Modified from a Met Office diagram.)

UKESM

- Earth system model based on HadGEM3 but with additional components & couplings.
- Joint Met Office and NERC development & will contribute to CMIP7.

Resolutions:

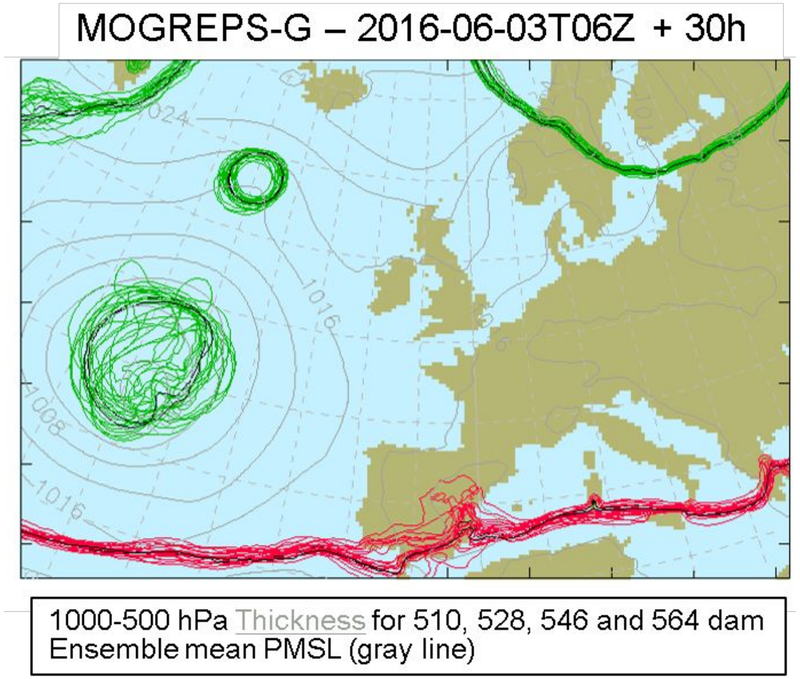
- Low: N96-ORCA1
- High: N216-ORCA025
- Hybrid: N216-ORCA025 with reduced res for UKCA & MEDUSA



UKESM

Other modelling systems

- MOGREPS: ensemble forecast
  - Based on Parallel Suites
- GloSea: seasonal prediction
  - Based on HadGEM3
- DePreSys: decadal prediction
  - Based on HadGEM3
- HadGEM2:
  - Previous Earth System Model
- HadCM3:
  - Previous climate model



MOGREPS-G – 2016-06-03T06Z + 30h

1000-500 hPa Thickness for 510, 528, 546 and 564 dam
Ensemble mean PMSL (gray line)

© Crown copyright Met Office

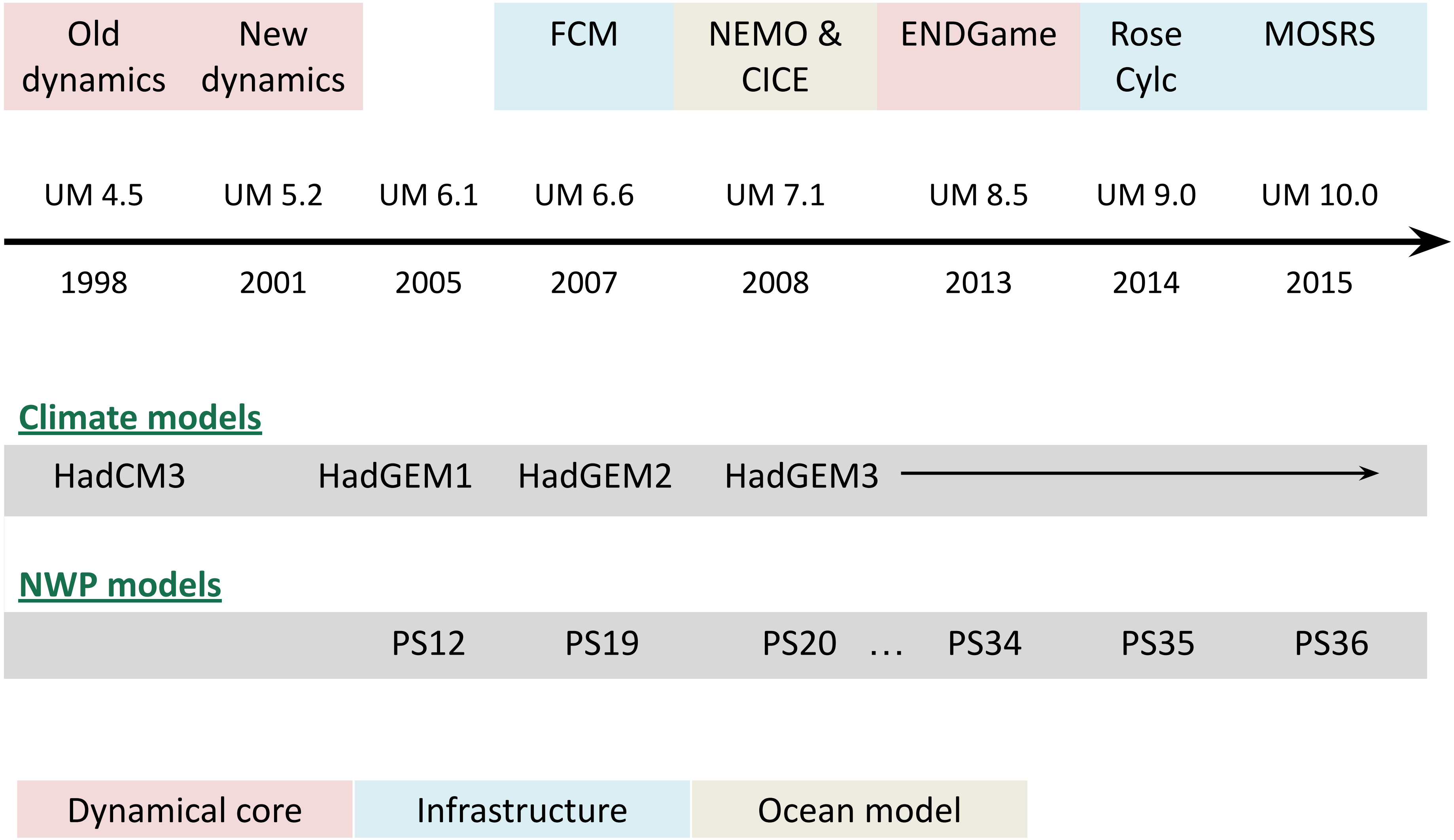
NWP v Climate models

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

	NWP	**Climate**
Run length	5 day operational forecast, 15 day ensemble forecast	Months (seasonal) Years, decades, centuries +
Global resolution	Global model: N1280L70 with 4 min ts Ensemble: N640L70 with 7.5 min ts	Low resolution: N96L85 with 20 min ts High resolution: N512L85 with 15 min ts
Calendar	Gregorian	Seasonal: Gregorian Other: 360 day
Dynamics	Non-bit reproducible	Bit-reproducible
Aerosols	Climatological concentrations specified.	Interactive schemes with emissions specified (not seasonal).

Major UM developments

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



Old dynamics

New dynamics

FCM

NEMO & CICE

ENDGame

Rose Cylc

MOSRS

UM 4.5 — 1998

UM 5.2 — 2001

UM 6.1 — 2005

UM 6.6 — 2007

UM 7.1 — 2008

UM 8.5 — 2013

UM 9.0 — 2014

UM 10.0 — 2015

Climate models

HadCM3 HadGEM1 HadGEM2 HadGEM3 →

NWP models

PS12 PS19 PS20 … PS34 PS35 PS36

Dynamical core

Infrastructure

Ocean model

MOSRS

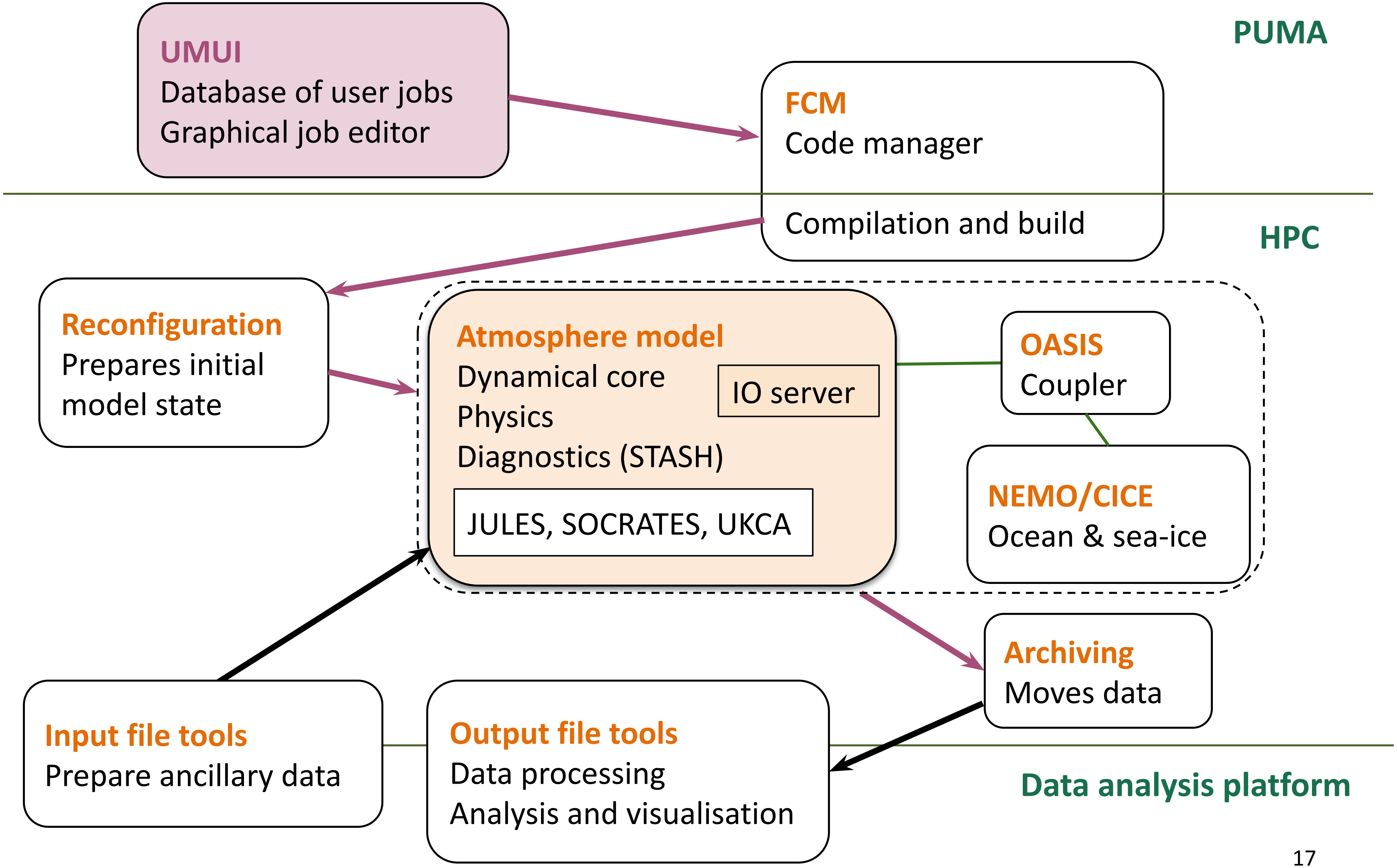
- **Met Office Science Repository Service**

  https://code.metoffice.gov.uk

- Hosts the UM code (from version 10.x)
- Plus other Met Office scientific software (e.g. JULES, SURF) and model developments (e.g. GMED, UKESM)
- One single repository for all UM users around the world:
  - Immediate releases
  - Shared code developments & documentation
  - Shared suite repository

Old UM software (pre vn9.0)

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



PUMA

UMUI
Database of user jobs
Graphical job editor

FCM
Code manager
Compilation and build

HPC

Reconfiguration
Prepares initial model state

Atmosphere model
Dynamical core
Physics
Diagnostics (STASH)
IO server
JULES, SOCRATES, UKCA

OASIS
Coupler

NEMO/CICE
Ocean & sea-ice

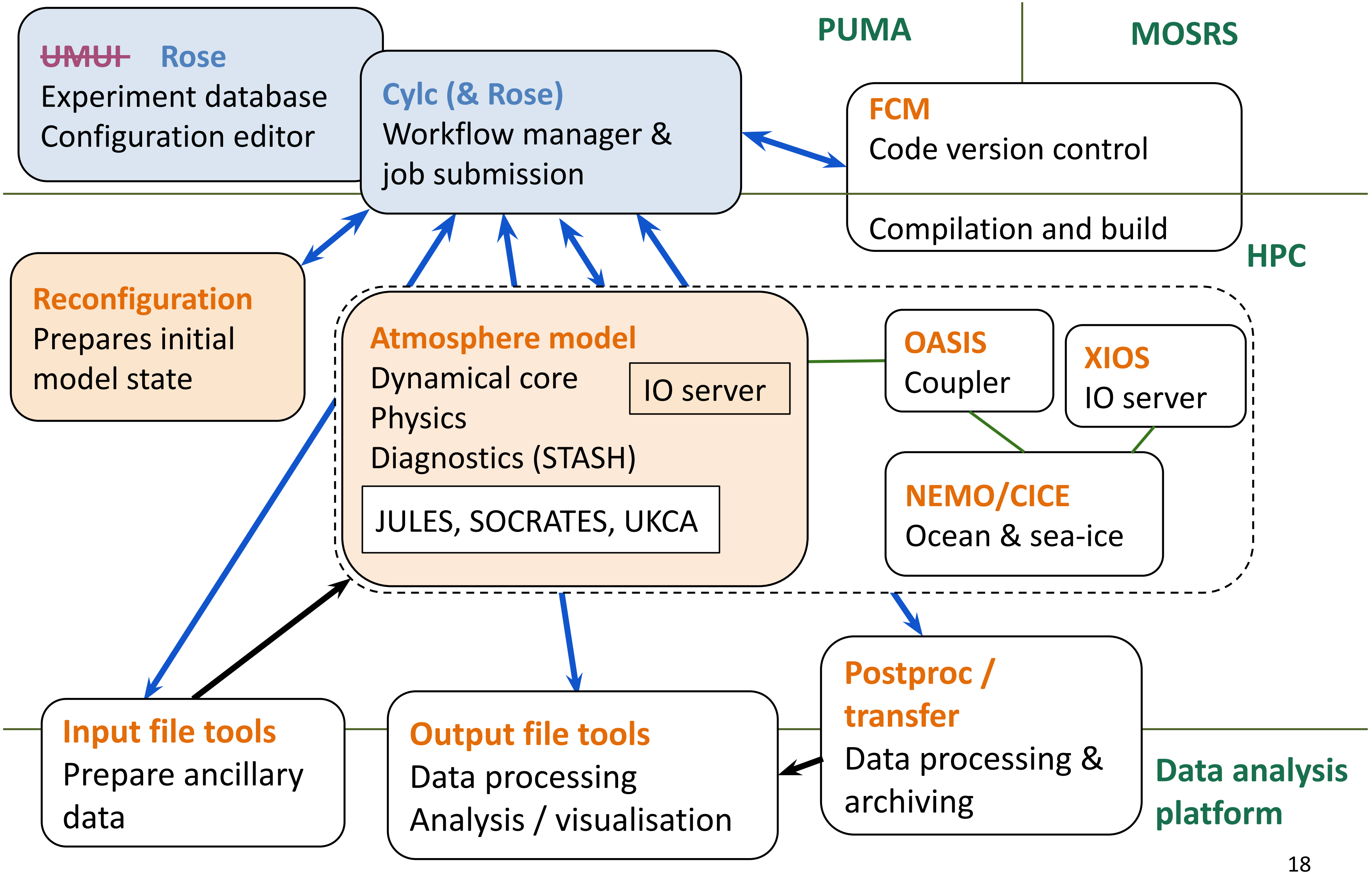
Archiving
Moves data

Input file tools
Prepare ancillary data

Output file tools
Data processing
Analysis and visualisation

Data analysis platform

UM software (from vn10.0)



~~UMUI~~ Rose
Experiment database
Configuration editor

Cylc (& Rose)
Workflow manager & job submission

PUMA

MOSRS

FCM
Code version control

Compilation and build

HPC

Reconfiguration
Prepares initial model state

Atmosphere model
Dynamical core
Physics
Diagnostics (STASH)

IO server

JULES, SOCRATES, UKCA

OASIS
Coupler

XIOS
IO server

NEMO/CICE
Ocean & sea-ice

Input file tools
Prepare ancillary data

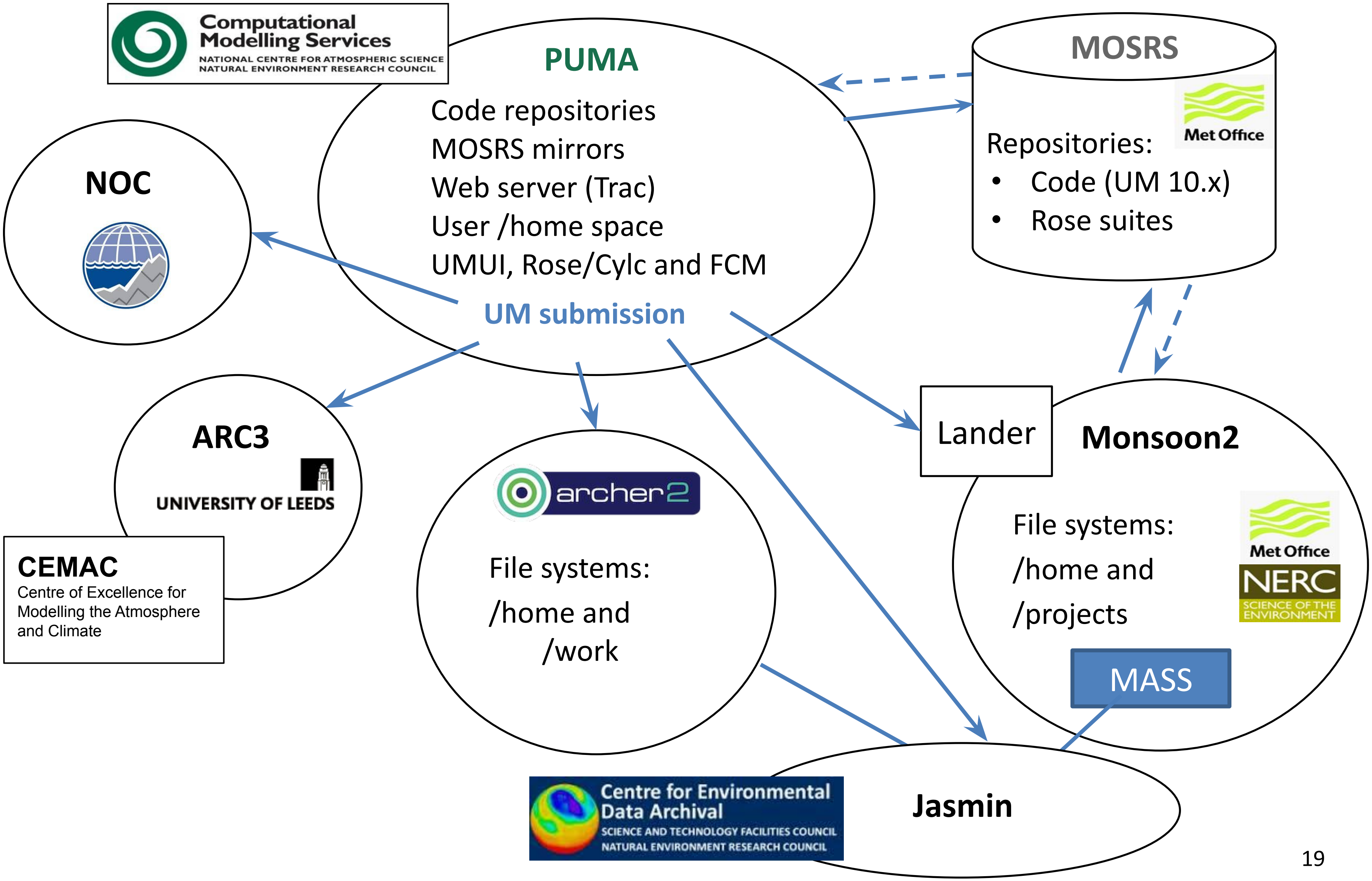
Output file tools
Data processing
Analysis / visualisation

Postproc / transfer
Data processing & archiving

Data analysis platform

UM submission workflow

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



Computational Modelling Services
NATIONAL CENTRE FOR ATMOSPHERIC SCIENCE
NATURAL ENVIRONMENT RESEARCH COUNCIL
PUMA
Code repositories
MOSRS mirrors
Web server (Trac)
User /home space
UMUI, Rose/Cylc and FCM
UM submission
MOSRS
Met Office
Repositories:
• Code (UM 10.x)
• Rose suites
NOC
ARC3
UNIVERSITY OF LEEDS
CEMAC
Centre of Excellence for Modelling the Atmosphere and Climate
archer2
File systems:
/home and
/work
Lander
Monsoon2
File systems:
/home and
/projects
Met Office
NERC
SCIENCE OF THE ENVIRONMENT
MASS
Centre for Environmental Data Archival
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL
Jasmin

NCAS
Unified Model Introduction

Part 2: Running the UM

University of Leeds, 7-9 February 2023

Rose, cylc & FCM

The UM is configured and controlled through **Rose**:

- Rose is a set of tools for running and managing scientific applications.
- Developed by the Met Office but not UM-specific (can be used for other codes, e.g. NEMO).

Rose is tightly integrated with:

- **cylc**: Workflow management & scheduling system
- **FCM**: Code management tool built around Subversion and a Make-like build system.

Creating a UM suite

- A Rose **suite** defines an experiment set up (e.g. UM run plus pre- and post-processing).
- To get started, copy an existing UM suite from UKMO, NCAS or a colleague that is close to what you wish to run.
  - Check that it runs before making changes.
- Contact CMS for:
  - Standard suites for ARCHER2 or Monsoon2.
  - Advice on porting suites to different platforms (as this may not be straightforward).

UM application settings

A UM suite is defined by:

- UM code version (e.g. vn13.0) plus changes to the main code base in the form of FCM branches
- Horizontal and vertical resolution and domain
- Scientific schemes and parameterizations
- Input files: start files, ancillary files and lateral boundary conditions
- STASH requests specifying the diagnostic fields to be output
- Control settings such as run length and cycling (resubmission) frequency.

Suite repository

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Rose suites are held in a repository on MOSRS:

- Suites are under version control.
- Suites have a unique id, e.g:

u-aa774

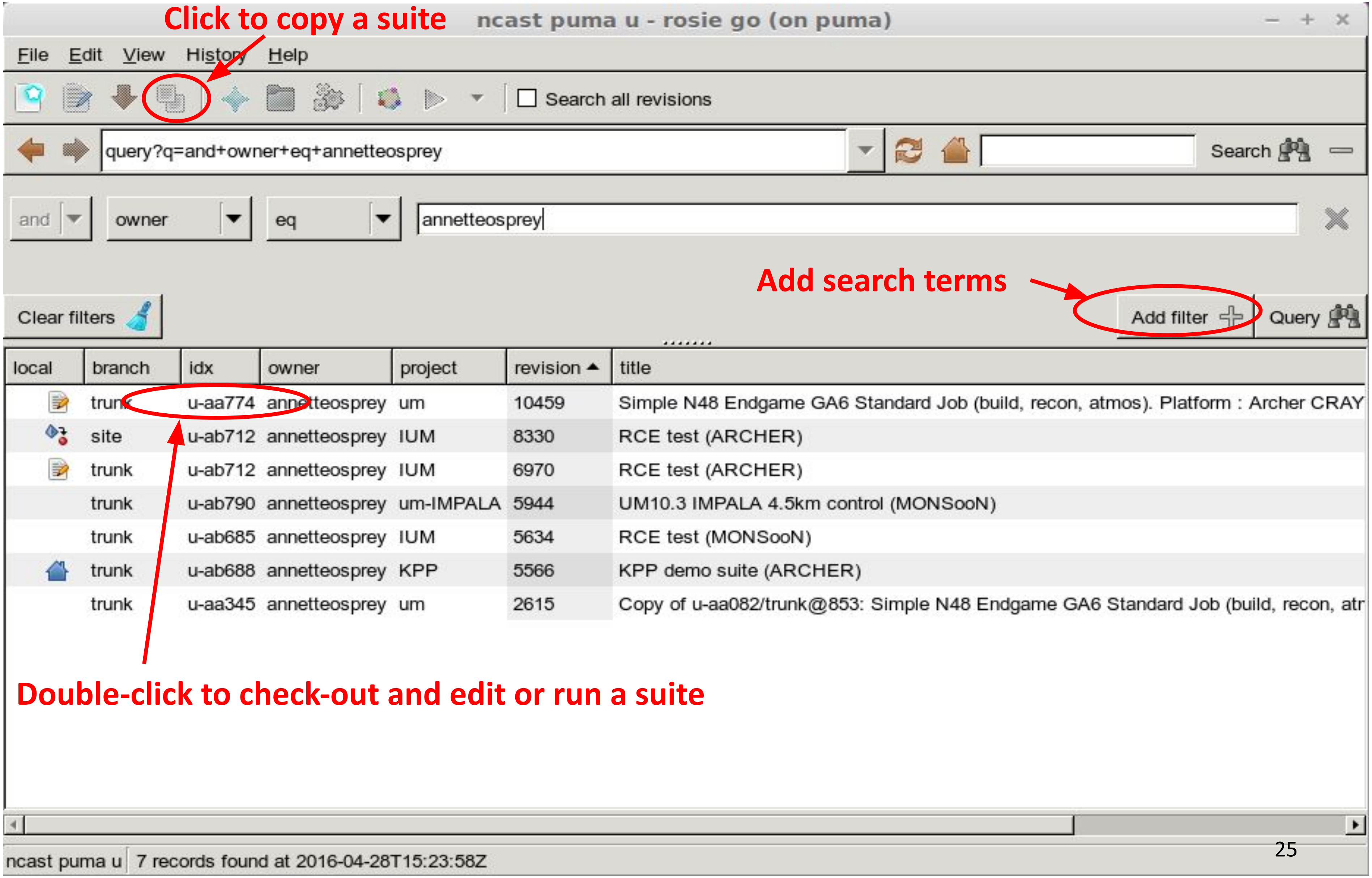
To launch the graphical suite repository viewer:

```
rosie go &
```

You can copy suites via the GUI or command-line tools:

```
rosie copy <suite-id>
```

Rosie go


Click to copy a suite
ncast puma u - rosie go (on puma)
File Edit View History Help
Search all revisions
Search
Clear filters
Add search terms
Add filter
Query
Double-click to check-out and edit or run a suite

query?q=and+owner+eq+annetteosprey

and | owner | eq | annetteosprey

local	branch	idx	owner	project	revision	title
	trunk	u-aa774	annetteosprey	um	10459	Simple N48 Endgame GA6 Standard Job (build, recon, atmos). Platform : Archer CRAY
	site	u-ab712	annetteosprey	IUM	8330	RCE test (ARCHER)
	trunk	u-ab712	annetteosprey	IUM	6970	RCE test (ARCHER)
	trunk	u-ab790	annetteosprey	um-IMPALA	5944	UM10.3 IMPALA 4.5km control (MONSooN)
	trunk	u-ab685	annetteosprey	IUM	5634	RCE test (MONSooN)
	trunk	u-ab688	annetteosprey	KPP	5566	KPP demo suite (ARCHER)
	trunk	u-aa345	annetteosprey	um	2615	Copy of u-aa082/trunk@853: Simple N48 Endgame GA6 Standard Job (build, recon, atr

ncast puma u | 7 records found at 2016-04-28T15:23:58Z

Editing suites

To launch the Rose suite editor GUI, double-click a suite in the Rosie GUI.

Suites consist of a set of text files, which can be edited directly

- Checked-out suites live in a directory:

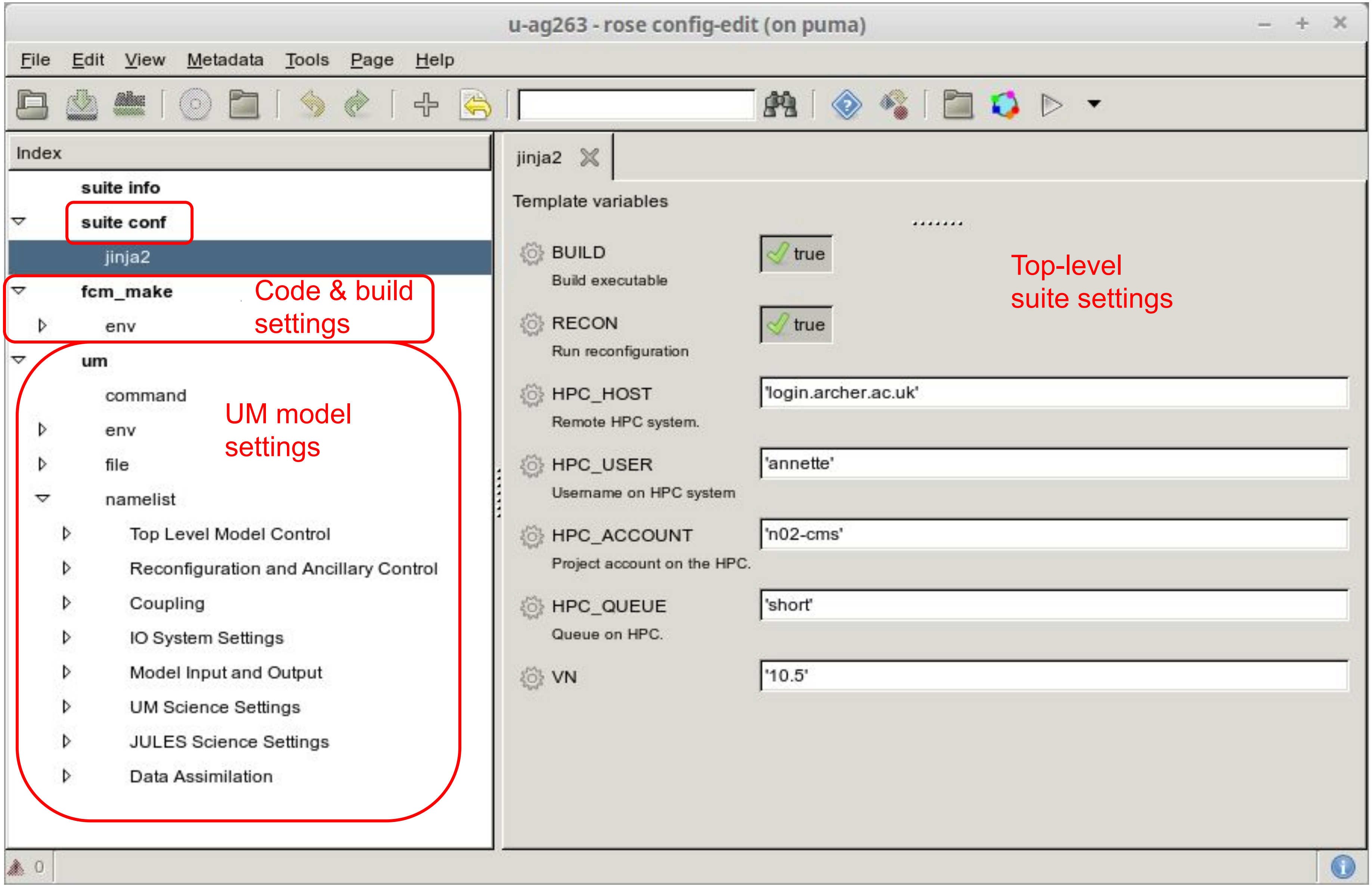
  `$HOME/roses/<suite-id>/`

To launch the Rose editor GUI from the command-line:

- `cd` to the suite file directory
- Run: `rose edit &`

Rose edit

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



Code & build settings

UM model settings

Top-level suite settings

BUILD ☑ true
Build executable

RECON ☑ true
Run reconfiguration

HPC_HOST 'login.archer.ac.uk'
Remote HPC system.

HPC_USER 'annette'
Username on HPC system

HPC_ACCOUNT 'n02-cms'
Project account on the HPC.

HPC_QUEUE 'short'
Queue on HPC.

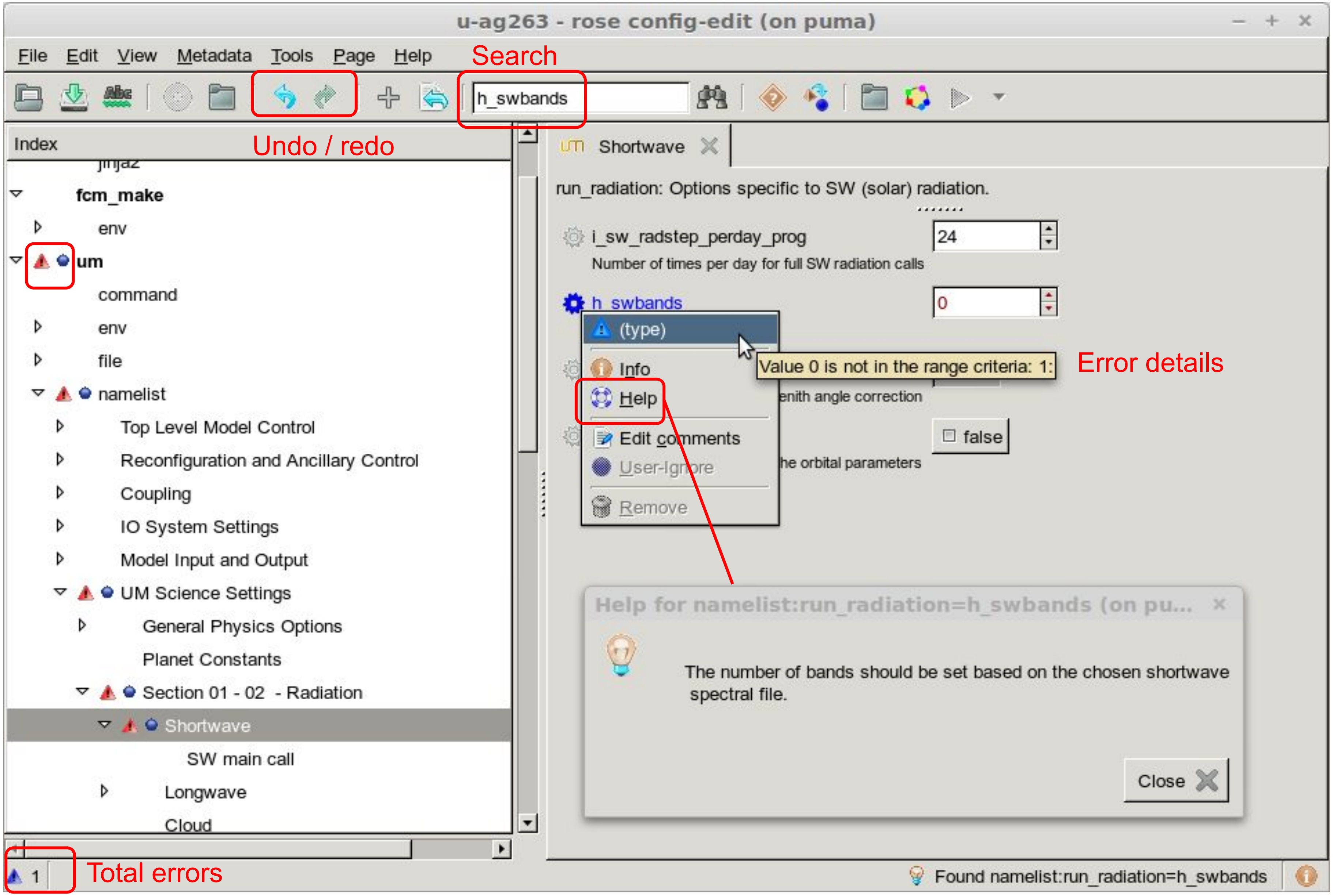
VN '10.5'

Using the GUI

Features:

- Search facility (or can `grep` suite files).
- Undo/redo button.
- Highlighting of unsaved changes.
- Basic description of each field plus additional information (only as useful as metadata provided!)
- Immediate type checking (where specified in metadata).
- Further checking with “Check fail-if/warn-if” macro.

Editing a UM suite



u-ag263 - rose config-edit (on puma)

Search: h_swbands

Shortwave

run_radiation: Options specific to SW (solar) radiation.

i_sw_radstep_perday_prog: 24
Number of times per day for full SW radiation calls

h_swbands: 0

Value 0 is not in the range criteria: 1:

false

Help for namelist:run_radiation=h_swbands (on pu...

The number of bands should be set based on the chosen shortwave spectral file.

Found namelist:run_radiation=h_swbands

Search

Undo / redo

Error details

Total errors

Running suites

- ARCHER2 suites are submitted from PUMA.
- Monsoon suites are submitted from the Monsoon login nodes.
- Run from editor GUI or command line:

  ```
  rose suite-run
  ```
- Launches a GUI which displays progress of suite.
- Running suite is controlled through a daemon running on PUMA (or equivalent).
- Users can:
  - stop, pause, and restart suites
  - edit running suites
  - re-run parts that have failed.

A basic UM suite

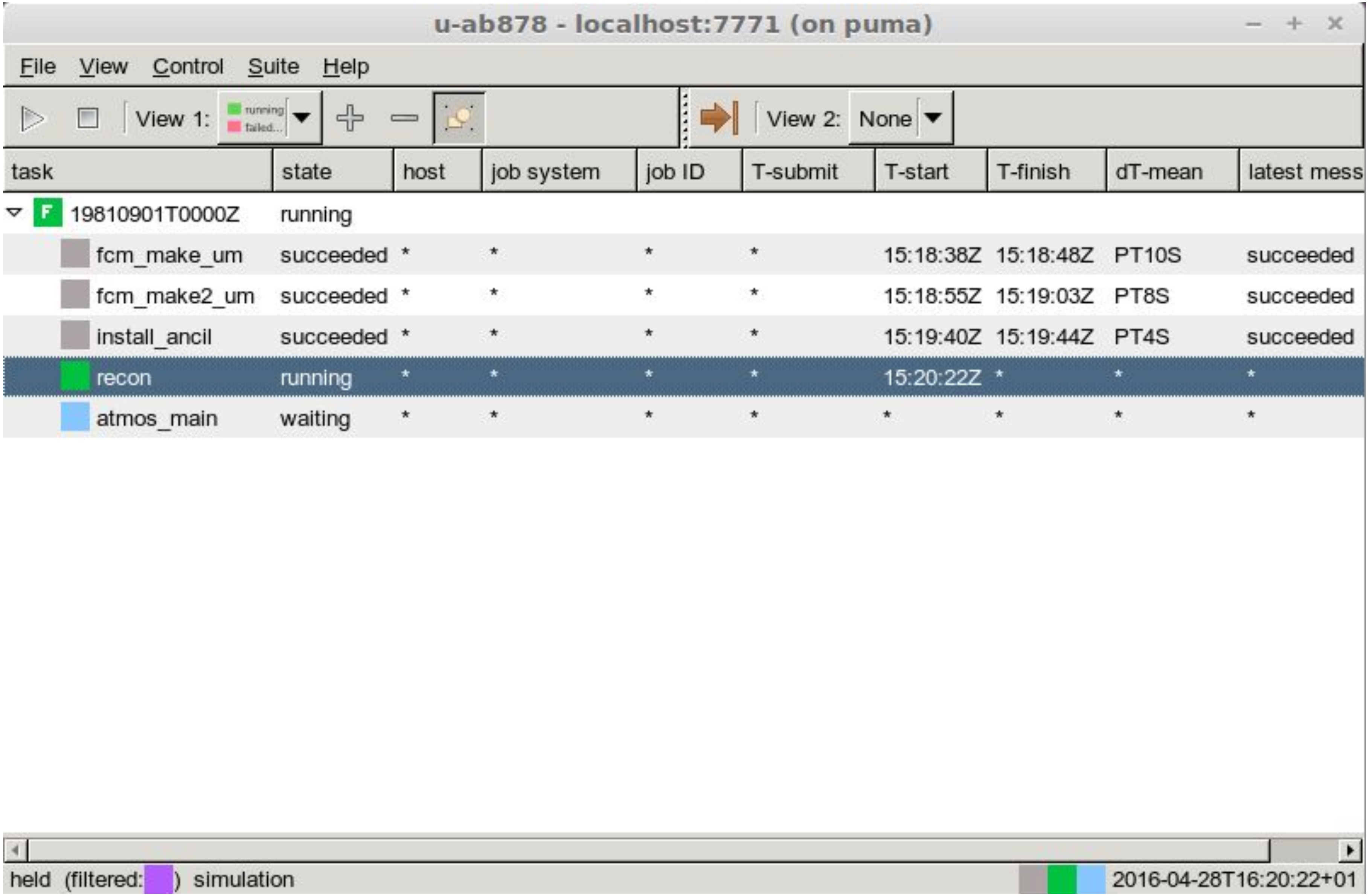
A basic UM suite consists of the following tasks:

`fcm_make`	Code extract and mirror
`fcm_make2`	Code pre-processing and build
`recon`	Reconfiguration: preparation of start data.
`atmos`	Model run

Single step on Monsoon2

- Tasks may have different names in different suites.
- More complex suites may have additional tasks, for e.g. post-processing, archiving logs, testing etc.

Rose suite-run



task	state	host	job system	job ID	T-submit	T-start	T-finish	dT-mean	latest mess
▽ F 19810901T0000Z	running								
▨ fcm_make_um	succeeded	*	*	*	*	15:18:38Z	15:18:48Z	PT10S	succeeded
▨ fcm_make2_um	succeeded	*	*	*	*	15:18:55Z	15:19:03Z	PT8S	succeeded
▨ install_ancil	succeeded	*	*	*	*	15:19:40Z	15:19:44Z	PT4S	succeeded
🟩 recon	running	*	*	*	*	15:20:22Z	*	*	*
🟦 atmos_main	waiting	*	*	*	*	*	*	*	*

held (filtered: 🟪) simulation ▨🟩🟦 2016-04-28T16:20:22+01

Viewing log files

Rose bush is a system for browsing suite log files:

PUMA: Currently not available

Monsoon: From xcslc0/xcslc1 launch browser by running:

`firefox` http://localhost/rose-bush/

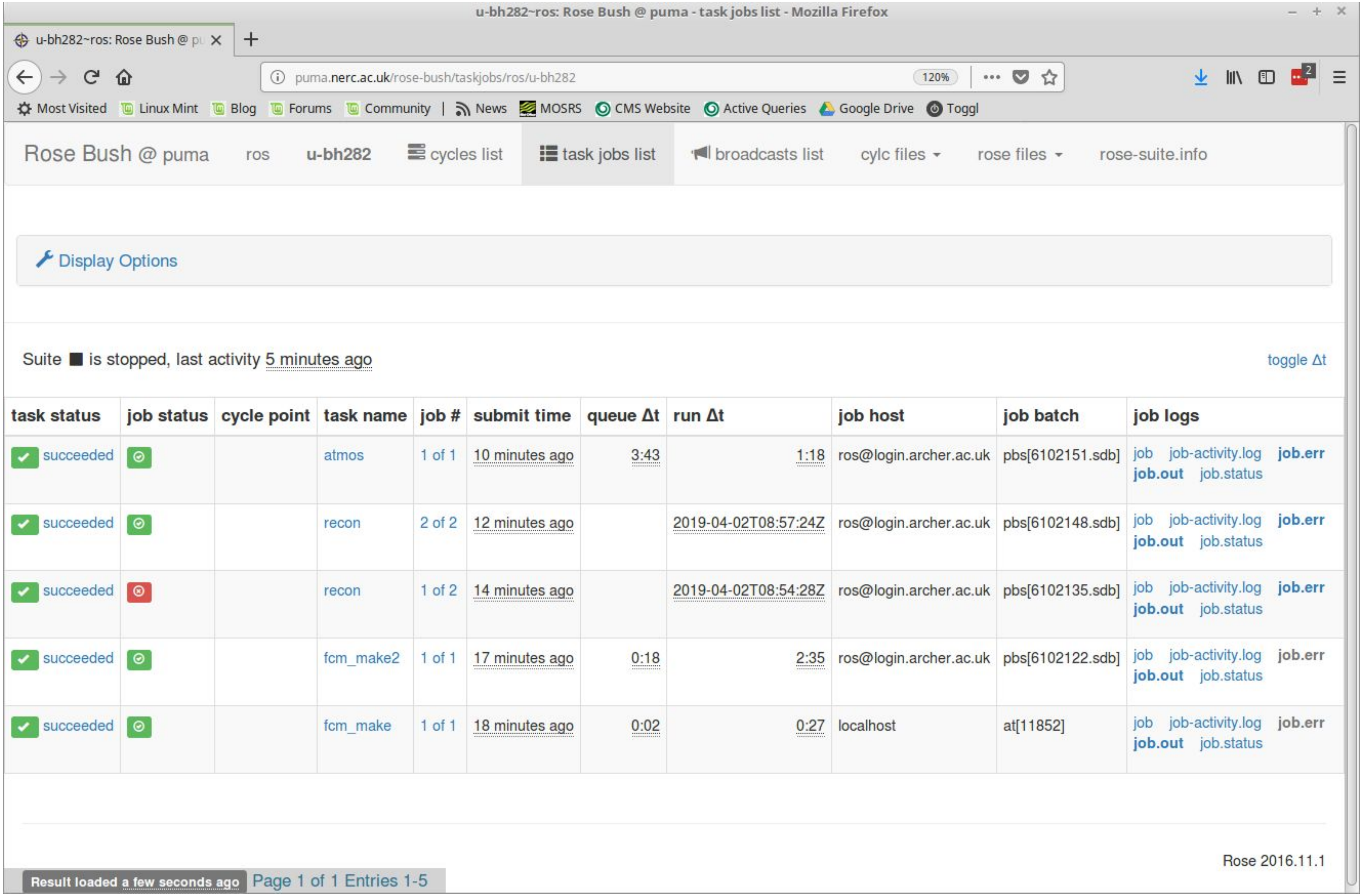
Note: Rose bush only shows files from last run.

Can view log files directly in:

```
~/cylc-run/<suite-id>/log
```

- Logs from older runs will be tarred up (unless deleted with housekeeping or `--new` run).
- Cylc-run directory can become large very quickly, so regularly delete.

Rose bush



Suite ■ is stopped, last activity 5 minutes ago

task status	job status	cycle point	task name	job #	submit time	queue Δt	run Δt	job host	job batch	job logs
✓ succeeded	⊙		atmos	1 of 1	10 minutes ago	3:43	1:18	ros@login.archer.ac.uk	pbs[6102151.sdb]	job job-activity.log job.err job.out job.status
✓ succeeded	⊙		recon	2 of 2	12 minutes ago		2019-04-02T08:57:24Z	ros@login.archer.ac.uk	pbs[6102148.sdb]	job job-activity.log job.err job.out job.status
✓ succeeded	⊗		recon	1 of 2	14 minutes ago		2019-04-02T08:54:28Z	ros@login.archer.ac.uk	pbs[6102135.sdb]	job job-activity.log job.err job.out job.status
✓ succeeded	⊙		fcm_make2	1 of 1	17 minutes ago	0:18	2:35	ros@login.archer.ac.uk	pbs[6102122.sdb]	job job-activity.log job.err job.out job.status
✓ succeeded	⊙		fcm_make	1 of 1	18 minutes ago	0:02	0:27	localhost	at[11852]	job job-activity.log job.err job.out job.status

Rose 2016.11.1

Result loaded a few seconds ago Page 1 of 1 Entries 1-5

Interpreting UM log files

Log messages from the model are sent to `job.out` and `job.err`.

- The output listing can be quite large and confusing.
- Check for keywords like `ERROR`, `ABORT`, `"file not found"`
- Timings are reported at the end of the listing file.
- You can control the volume of log messages by setting `PRINT_STATUS`.

Submission failures and suite timeout messages are reported in:
`job-activity.log`, `suite/out`, `suite/err`

If you are making model changes, always check things look OK before proceeding, i.e. executable exists, start file exists, output data looks sensible.

Output data

All Rose/cylc suite files are held in the directory:

```
~/cylc-run/<suite-id>/
```

UM output such as restart and diagnostic files usually goes to:

```
share/data/History_Data/
```

Or it may be written to the task directory for that cycle, e.g.:

```
work/
    19810901T0000Z/atmos/
    19811001T0000Z/atmos/
```

Build files go to:

```
share/fcm_make/
```

Terminology recap

Suite:	Experimental set up
Rose:	Application management system
Rosie:	Suite repository manager
Rose edit:	Application editor
Rose bush:	Output log viewer
Cylc:	Scheduling & workflow system
FCM:	Code management & build system

NCAS
Unified Model Introduction

Part 3: FCM and the UM

University of Leeds, 7-9 February 2023

What is FCM?

- Flexible Configuration Management system
  - Written by the Met Office.
  - A set of tools for managing and building source code.
  - Uses subversion for code management
    - Defines a common process and naming convention
    - Adds a layer on top of subversion

Overview of FCM

- FCM system consists of 3 components:
    - Integrated Configuration Management, wiki and issue tracking system
    - Extract system
    - Build system

Trac

- Integrated wiki and issue tracker
  - can be used for project documentation
  - keep track of bugs, development, etc
- Browser for subversion repository
- Timeline view of issues, wiki pages and subversion repository

  https://code.metoffice.gov.uk

Extract System

Purpose:

- Extracts source code from the repository ready for the build system

Features:

- Combines code from a number of branches
  - Only if the modified files do not overlap
- Mirrors code to a remote system (e.g. ARCHER2)
- Generates a configuration file for the build system

Build System

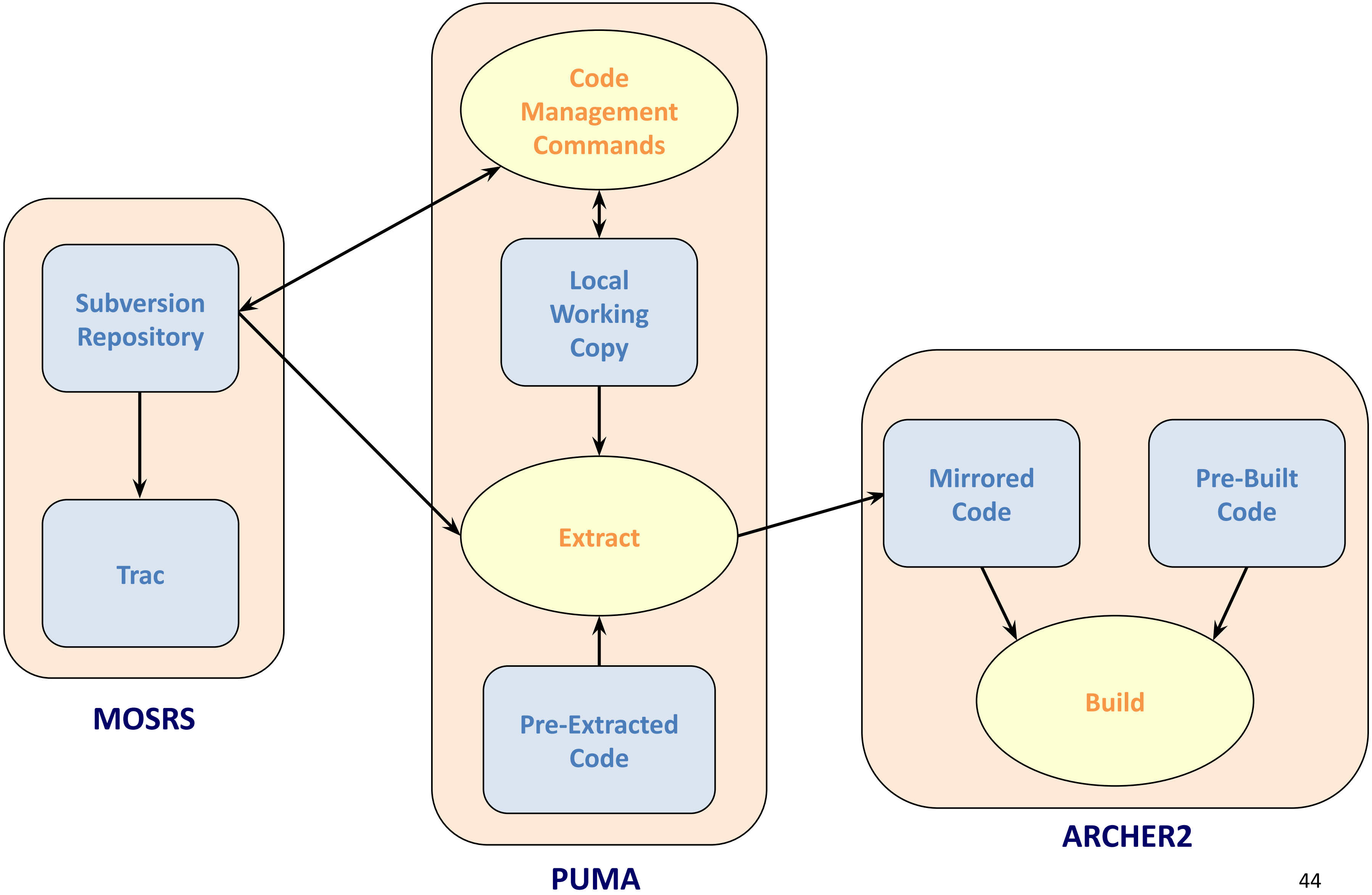
Purpose:

- Builds the code assembled by the extract system and creates the model executable.

Features:

- Parallel build
- Incremental build
- Build dependency analysis

FCM System Components



Features (1)

- Full history of source file
  - Accessible on the web via Trac
- Edit source code directly on PUMA
- Version control of directory tree
  - Copy, rename, add, delete files and directories
- Atomic commits
  - All or nothing is committed to the repository

Features (2)

- Access to all functions is via the *fcm* command
  - FCM command syntax is simple and easy to use
  - Simple GUI wrapper to the subversion commands
  - Commands used only on PUMA
  - Build system is installed on ARCHER2, Monsoon2, etc

- Comprehensive User Guide

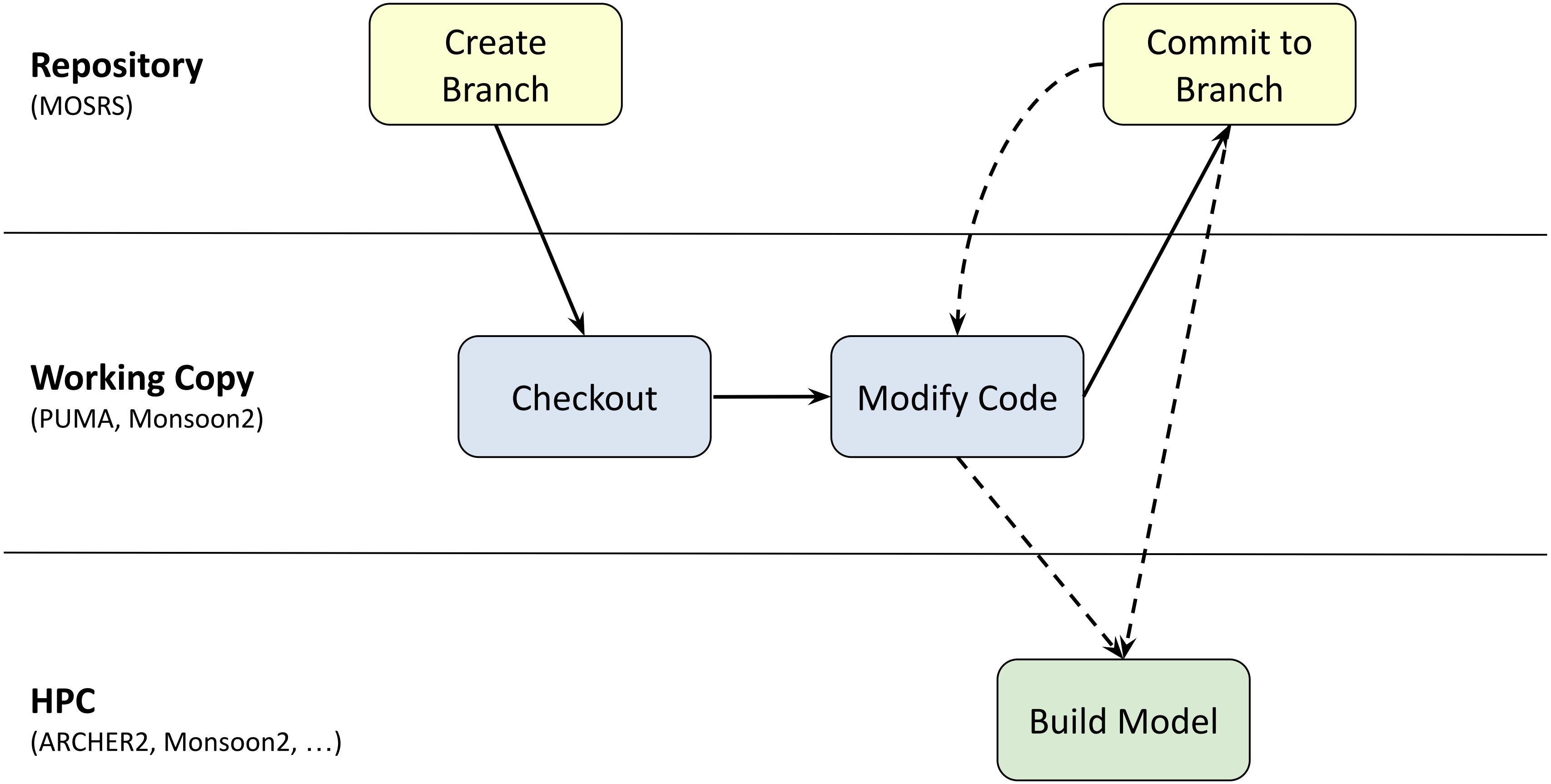
  http://metomi.github.io/fcm/doc/user_guide/

Subversion – some terminology

- The UM code is held in a ***subversion repository***
- The ***trunk*** is the consolidated master version of the code
- A ***release*** is a specified revision of the trunk (identified by a revision keyword)
- ***Branches*** are the method of making and tracking changes – held in the repository
- A ***working copy*** is a local copy of a branch (possibly with changes to it)

Typical Workflow

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



MOSRS

Met Office Science Repository Service

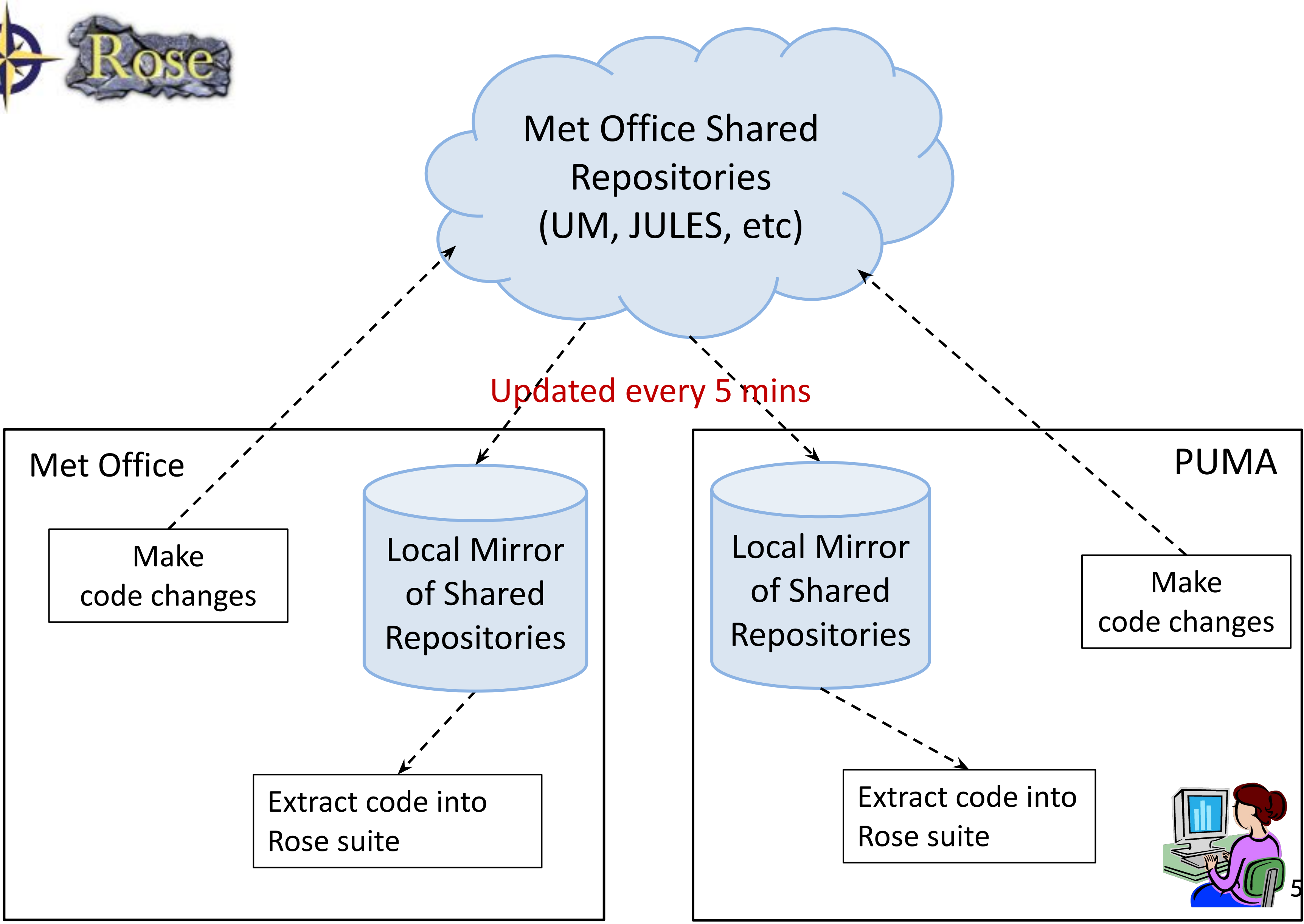
- Common subversion code repositories for all users:
  - Hosts UM code from vn10.0
  - Other model codes; JULES, MONC, etc
- Document collaborative projects (e.g. UKESM, GMED):

  https://code.metoffice.gov.uk/trac/home/wiki/ProjectList

Working with MOSRS:

- Users access Trac system (wiki and tickets) via web.
- Code changes are made by remotely checking out and committing to MOSRS.
- Sites have local read-only mirrors, used by suites when extracting code (faster than accessing MOSRS).

MOSRS



Met Office Shared Repositories (UM, JULES, etc)

Updated every 5 mins

Met Office

Make code changes

Local Mirror of Shared Repositories

Extract code into Rose suite

PUMA

Local Mirror of Shared Repositories

Make code changes

Extract code into Rose suite

UM Trunk Developments

Specific procedure for submitting code changes to the UM trunk, which includes:

- Liaising with code owner
- Documenting changes & testing in a code ticket on MOSRS
- Running rose-stem developer tests (on Monsoon or colleague may run inside Met Office).

Further details:

https://code.metoffice.gov.uk/trac/um/wiki/working_practices

Submission schedule and release deadlines:

https://code.metoffice.gov.uk/trac/um#ReleaseSchedule

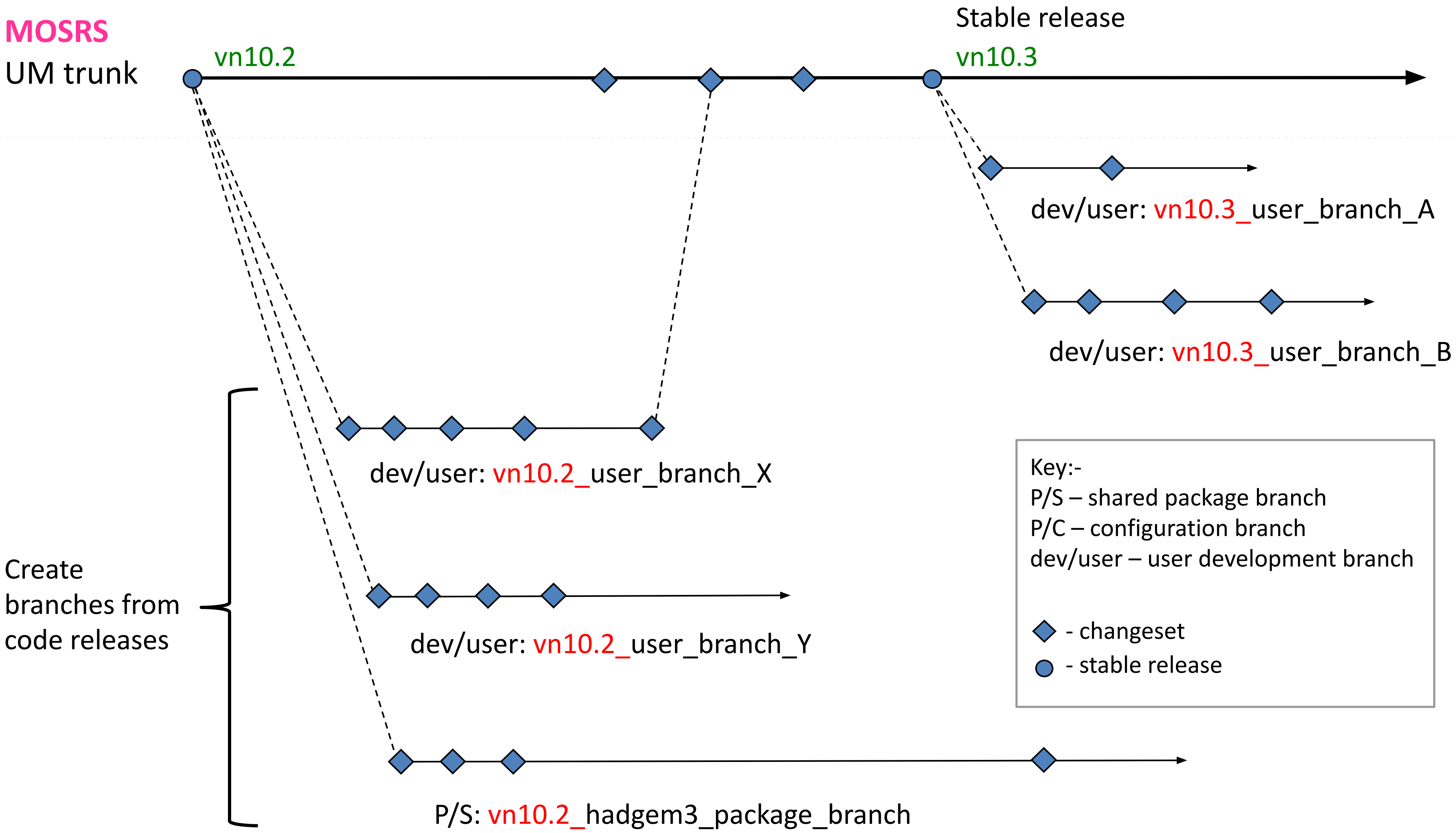
FCM and Revision Keywords

- FCM URLs are cumbersome - keywords provide a shortcut
- Specified with the `fcm:` prefix
- Run `fcm kp` for a full list

Keyword	Value
um.x	https://code.metoffice.gov.uk/svn/um/main
um.x_tr (or um.x-tr)	https://code.metoffice.gov.uk/svn/um/main/trunk
um.x_br (or um.x-br)	https://code.metoffice.gov.uk/svn/um/main/branches
um.xm (local mirror)	svn://puma/um.xm_svn/main

In a similar way *revision* keywords are used, so that you don’t have to remember the specific revision number that relates to a UM version. E.g. The revision keyword vn10.4 denotes the revision of the repository that is UM version 10.4, in this instance r18260.

UM Repository Diagram



MOSRS
UM trunk

vn10.2

Stable release
vn10.3

dev/user: vn10.3_user_branch_A

dev/user: vn10.3_user_branch_B

dev/user: vn10.2_user_branch_X

dev/user: vn10.2_user_branch_Y

P/S: vn10.2_hadgem3_package_branch

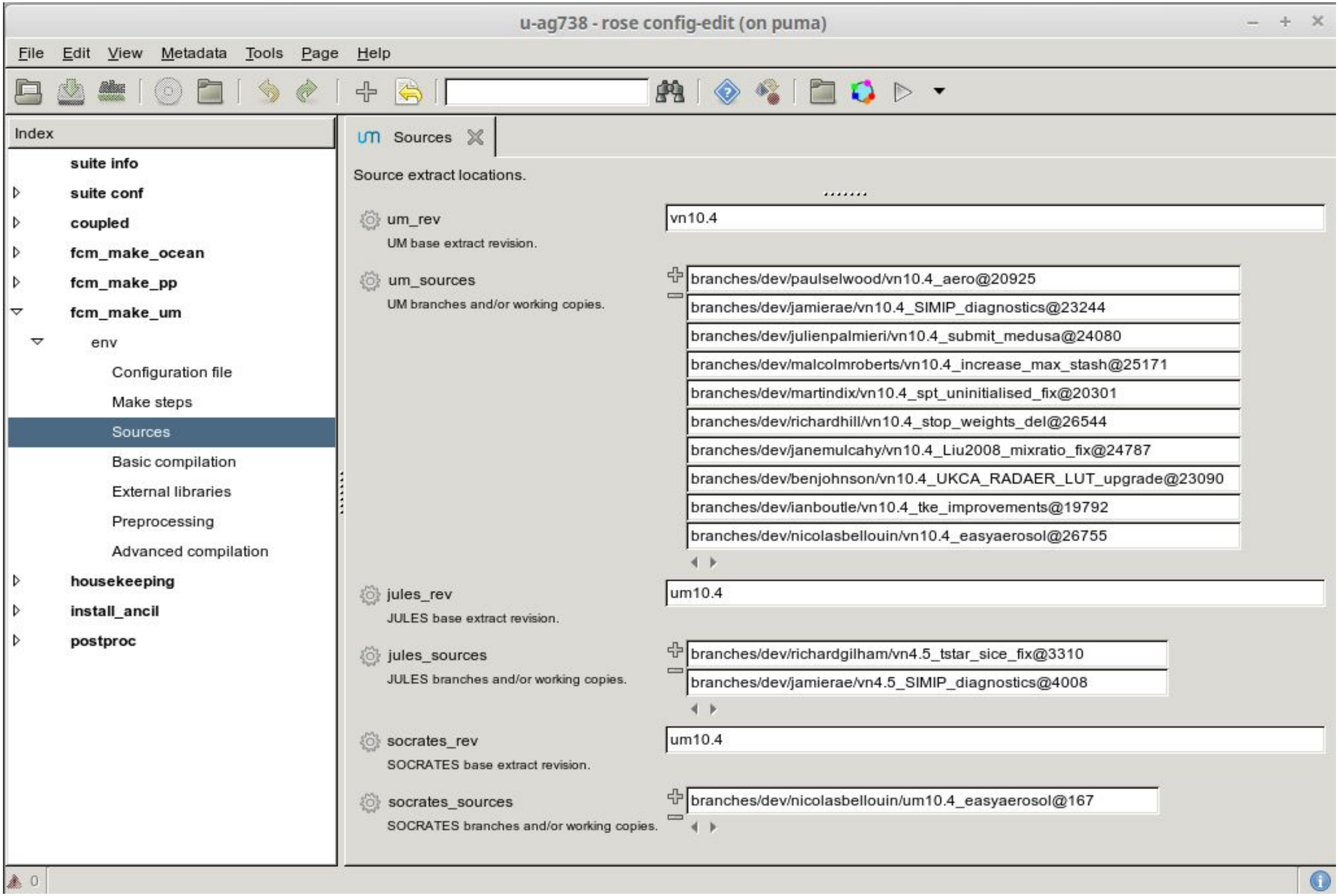
Create branches from code releases

Key:-
P/S – shared package branch
P/C – configuration branch
dev/user – user development branch

◆ - changeset
● - stable release

Specifying Branches in a Suite

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



u-ag738 - rose config-edit (on puma)

Index
- suite info
- suite conf
- coupled
- fcm_make_ocean
- fcm_make_pp
- fcm_make_um
  - env
    - Configuration file
    - Make steps
    - Sources
    - Basic compilation
    - External libraries
    - Preprocessing
    - Advanced compilation
- housekeeping
- install_ancil
- postproc

um Sources

Source extract locations.

um_rev
UM base extract revision.
vn10.4

um_sources
UM branches and/or working copies.
branches/dev/paulselwood/vn10.4_aero@20925
branches/dev/jamierae/vn10.4_SIMIP_diagnostics@23244
branches/dev/julienpalmieri/vn10.4_submit_medusa@24080
branches/dev/malcolmroberts/vn10.4_increase_max_stash@25171
branches/dev/martindix/vn10.4_spt_uninitialised_fix@20301
branches/dev/richardhill/vn10.4_stop_weights_del@26544
branches/dev/janemulcahy/vn10.4_Liu2008_mixratio_fix@24787
branches/dev/benjohnson/vn10.4_UKCA_RADAER_LUT_upgrade@23090
branches/dev/ianboutle/vn10.4_tke_improvements@19792
branches/dev/nicolasbellouin/vn10.4_easyaerosol@26755

jules_rev
JULES base extract revision.
um10.4

jules_sources
JULES branches and/or working copies.
branches/dev/richardgilham/vn4.5_tstar_sice_fix@3310
branches/dev/jamierae/vn4.5_SIMIP_diagnostics@4008

socrates_rev
SOCRATES base extract revision.
um10.4

socrates_sources
SOCRATES branches and/or working copies.
branches/dev/nicolasbellouin/um10.4_easyaerosol@167

FCM Dos and Don’ts

DO:

- Regularly commit your working copy changes to your branch at appropriate intervals. This means you can recover to a previous state easily.

DON’T:

- Copy another user’s working copy, either in whole or part. A working copy contains hidden subversion files which, if edited or moved, will cause problems.
- Copy sub-directories around within working copies, this can also lead to problems, for the same reason as above.

Basic CM Sub-Commands

- The essential sub-commands are:
  - checkout
  - commit
  - diff
  - status
  - branch
  - merge

- Help is available for all sub-commands:

```
fcm help <sub-command>
```

Further Information

- FCM User Guide
  http://metomi.github.io/fcm/doc/user_guide/
- Command-line help:
  ```
  fcm help <subcommand>
  ```
- Hands-on FCM Tutorial in exercises.

NCAS
Unified Model Introduction

Part 4: Managing suites

University of Leeds, 7-9 February 2023

Contents

- **Suites and version control**
- Running and controlling suites

Suites and version control

- Suites are held in an FCM (Subversion) repository on MOSRS.
- This is the `roses-u` repository:
  - https://code.metoffice.gov.uk/trac/roses-u
- There is also a web interface for browsing suites:
  - https://code.metoffice.gov.uk/rosie/u

- You can use the same FCM commands as for managing code, plus some extra rosie commands.

Rosie checkout & copy

- You need to check out a suite to edit and/or run it.
  - Double-click on the suite in rosie go
  - On the command line: `rosie checkout <suite-id>`
- All checked out suites go to your `~/roses` directory on PUMA.

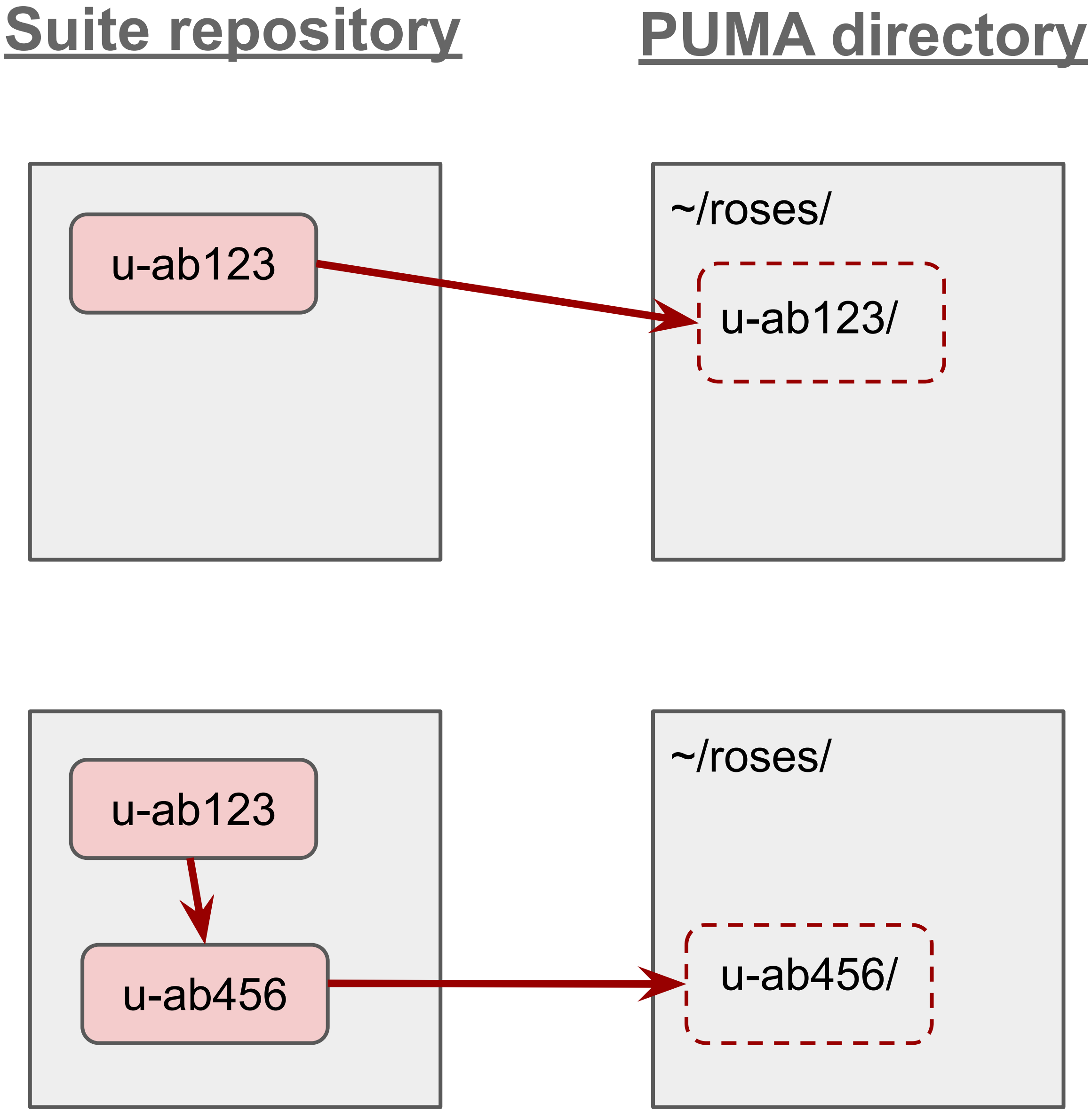
- You can check out and run another user’s suite but you won’t normally be able to save any changes you make.
- To copy into your own suite (& check out):
  - `rosie copy <suite-id>`

rosie checkout

- Pulls files into local directory.
- Does not alter repository

rosie copy

- Creates a new identical item in the repository.
- *And* pulls files to local directory.



Suite repository

PUMA directory

FCM commits

- Any changes you make in the Rose editor are saved in your checked out working copy only.
- To save changes to the repository:
  - `cd ~/roses/suite-id`
  - `fcm commit`
- Useful commands:
  - `fcm status`
  - `fcm diff`
- There is a Trac system (with wiki & ticket system) for the suite repository so you can document changes.

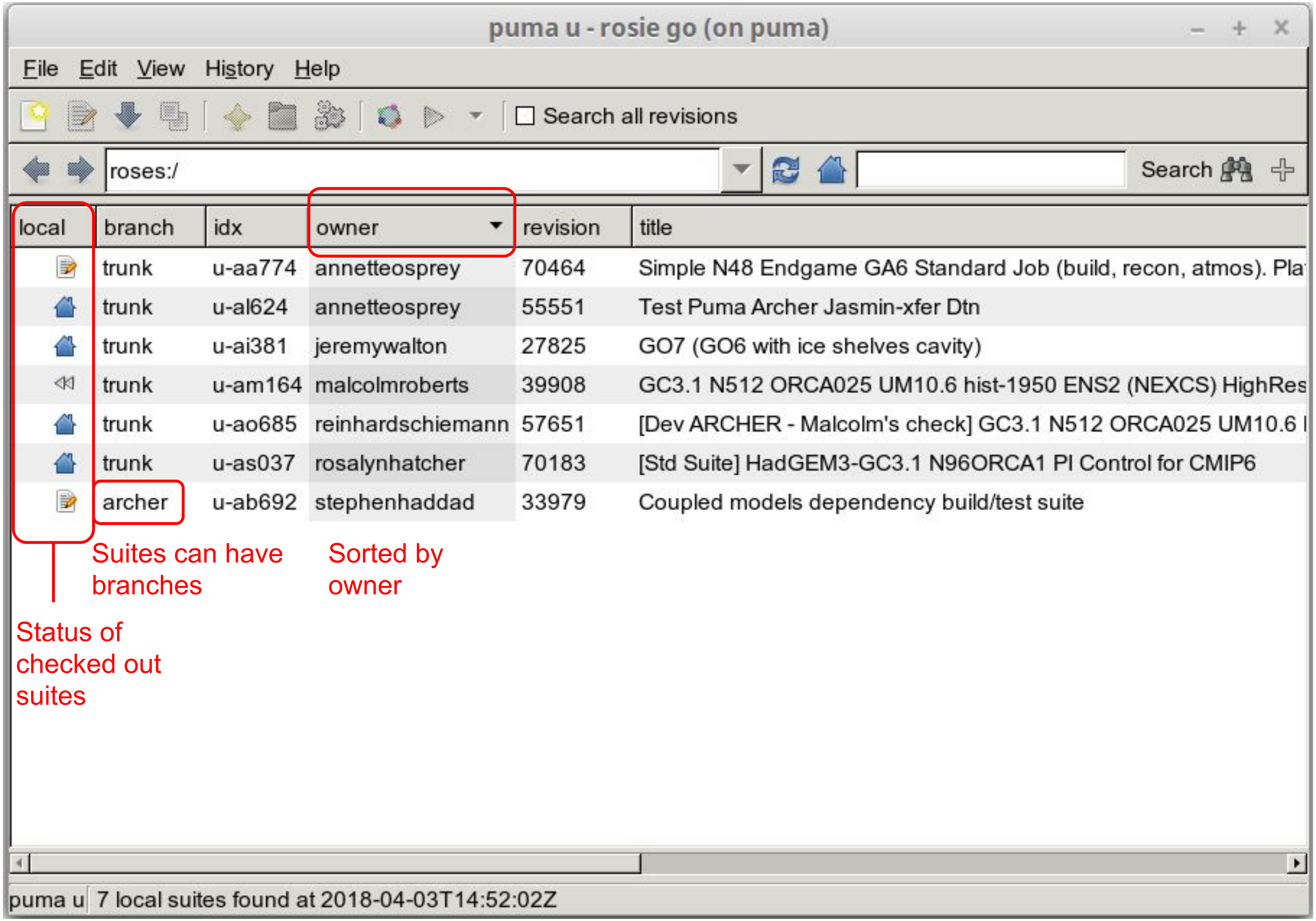
Rosie suite browser

`rosie go`: Graphical suite browser

- By default lists your checked out suites only.
- Provides suite status information such as repository modifications, local changes etc.
- Search facility to browse all suites in the repository.
- Can copy, checkout, edit and run suites from here.

`rosie ls` : Lists checked out suites on command-line

```
= u-aa774/trunk@70604 annetteosprey um          Simple N48 Endgame GA6 Standa
= u-ao685/trunk@57651 annetteosprey HighResMIP  GC3.1 N512 ORCA025 UM10.6 his
```



local	branch	idx	owner ▾	revision	title
✎	trunk	u-aa774	annetteosprey	70464	Simple N48 Endgame GA6 Standard Job (build, recon, atmos). Pla
⌂	trunk	u-al624	annetteosprey	55551	Test Puma Archer Jasmin-xfer Dtn
⌂	trunk	u-ai381	jeremywalton	27825	GO7 (GO6 with ice shelves cavity)
⏪	trunk	u-am164	malcolmroberts	39908	GC3.1 N512 ORCA025 UM10.6 hist-1950 ENS2 (NEXCS) HighRes
⌂	trunk	u-ao685	reinhardschiemann	57651	[Dev ARCHER - Malcolm's check] GC3.1 N512 ORCA025 UM10.6
⌂	trunk	u-as037	rosalynhatcher	70183	[Std Suite] HadGEM3-GC3.1 N96ORCA1 PI Control for CMIP6
✎	archer	u-ab692	stephenhaddad	33979	Coupled models dependency build/test suite

puma u | 7 local suites found at 2018-04-03T14:52:02Z

Suites can have branches

Sorted by owner

Status of checked out suites

Running suites

To run a suite:

- Click the play button from the rosie or rose edit GUIs.
- From the command-line:
  - Navigate to the suite directory, e.g: `cd ~/roses/u-aa774`
  - Then: `rose suite-run`

There are many options to this command - some useful ones are listed in the following slides.

Rose/cylc provide powerful options for pausing, stopping & restarting suites.

Rose suite-run

Useful options to `rose suite-run`:

`--new` `-N`	Delete any files from previous runs.
`--restart`	Restart suite from where it stopped.
`--reload`	Update suite definition for an already running suite
`-v -v --debug`	Debug/get more info.
`--no-gcontrol`	Do not launch cylc GUI (progress monitor).
`--no-log-archive`	Don’t tar up (archive) log files from previous runs

`--name=NAME`
`-n=NAME`
	Give the suite a different name to the base directory.

Rose suite-run

Options that allow you to see what the suite does without actually running - useful for complex suites:

`-- --hold`	Hold (pause) suite on startup.
`-- --mode=simulation` (cylc options)	Perform a dummy run (executes all tasks as 'sleep 10').
`--install-only` `-i`	Construct app files & copy over to remote hosts but don’t run
`--local-install-only` `-l`	Construct app files but don’t copy over to remote host.

When a suite is launched

When you launch a suite:

- The run scripts and application files (e.g. namelists) are copied to the `cylc-run` directories on the required machines.
- A suite manager process is launched on the suite host (PUMA or the Monsoon login nodes).
- Cylc keeps track of the progress of running tasks.
- Cylc submits tasks according to dependencies defined in the suite (e.g. build, then reconfiguration, then atmosphere model).
- On PUMA cylc monitors remote (e.g. ARCHER2) tasks by polling every 5 minutes.

Cylc GUI

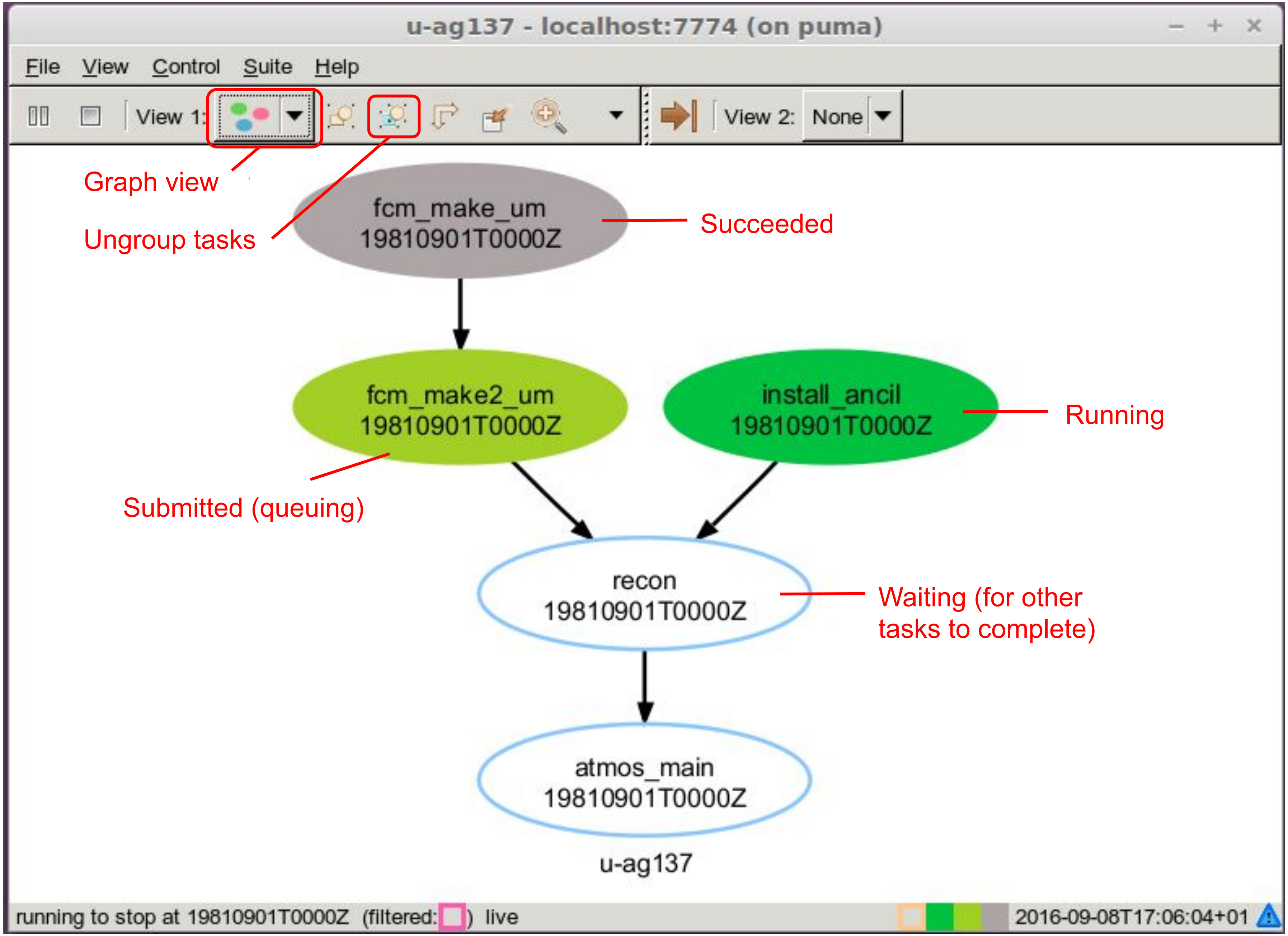
The Cylc GUI (gcylc) is launched automatically when you submit a suite.

- You can safely shutdown the GUI and log out of PUMA without affecting the suite.
- To relaunch the GUI run:

  `rose sgc` or `rose suite-gcontrol`

The Cylc GUI allows you to:

- View the task dependencies in your suite
- Monitor suite progress and view log files
- Kill, re-run and change the status of tasks
- Pause, release, and shutdown the suite



Graph view

Ungroup tasks

Succeeded

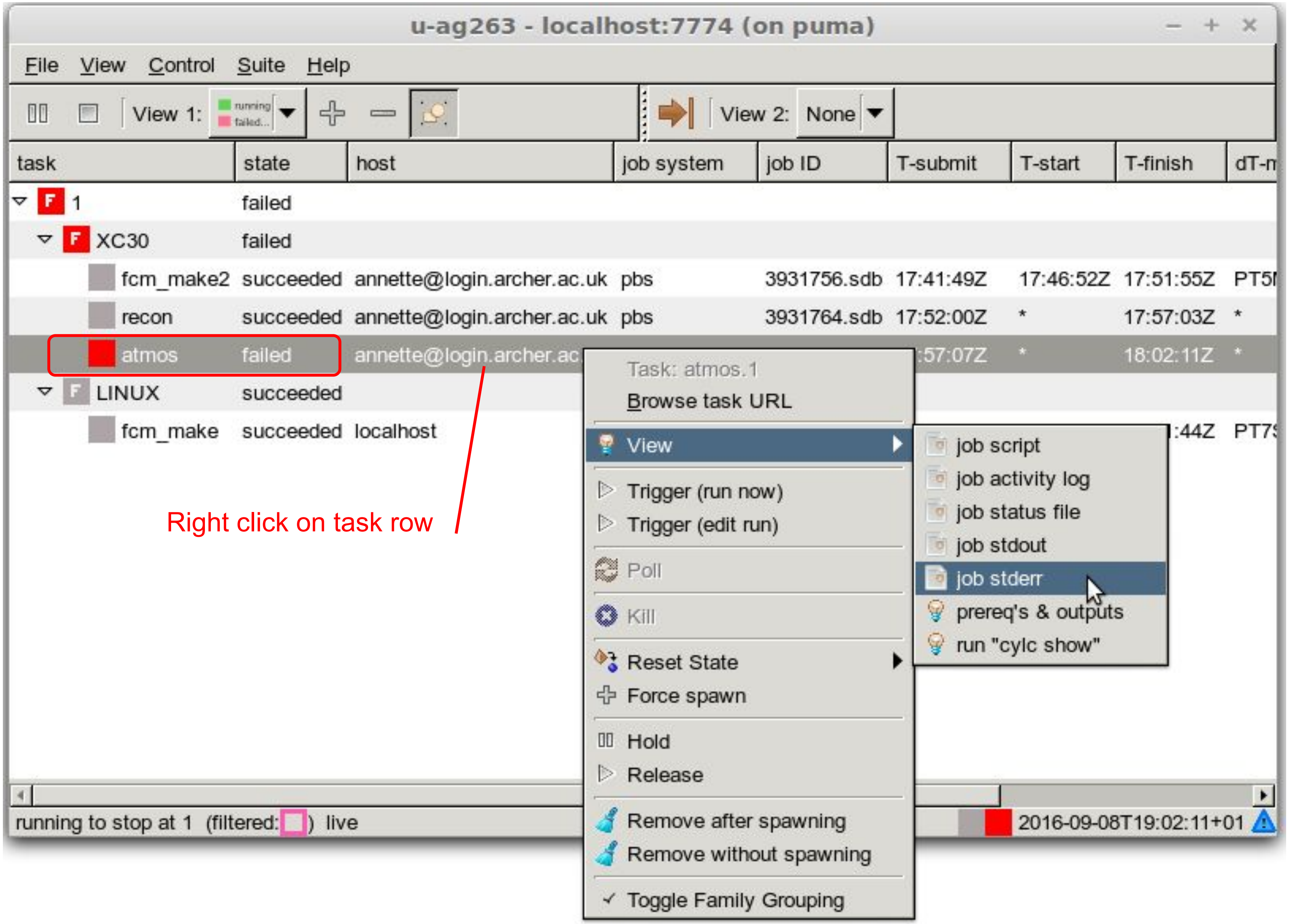
Running

Submitted (queuing)

Waiting (for other tasks to complete)

running to stop at 19810901T0000Z (filtered:) live

2016-09-08T17:06:04+01



task	state	host	job system	job ID	T-submit	T-start	T-finish	dT-m
F 1	failed							
F XC30	failed							
fcm_make2	succeeded	annette@login.archer.ac.uk	pbs	3931756.sdb	17:41:49Z	17:46:52Z	17:51:55Z	PT5I
recon	succeeded	annette@login.archer.ac.uk	pbs	3931764.sdb	17:52:00Z	*	17:57:03Z	*
atmos	failed	annette@login.archer.ac			:57:07Z	*	18:02:11Z	*
F LINUX	succeeded							
fcm_make	succeeded	localhost					:44Z	PT7

running to stop at 1 (filtered:) live

2016-09-08T19:02:11+01

Right click on task row

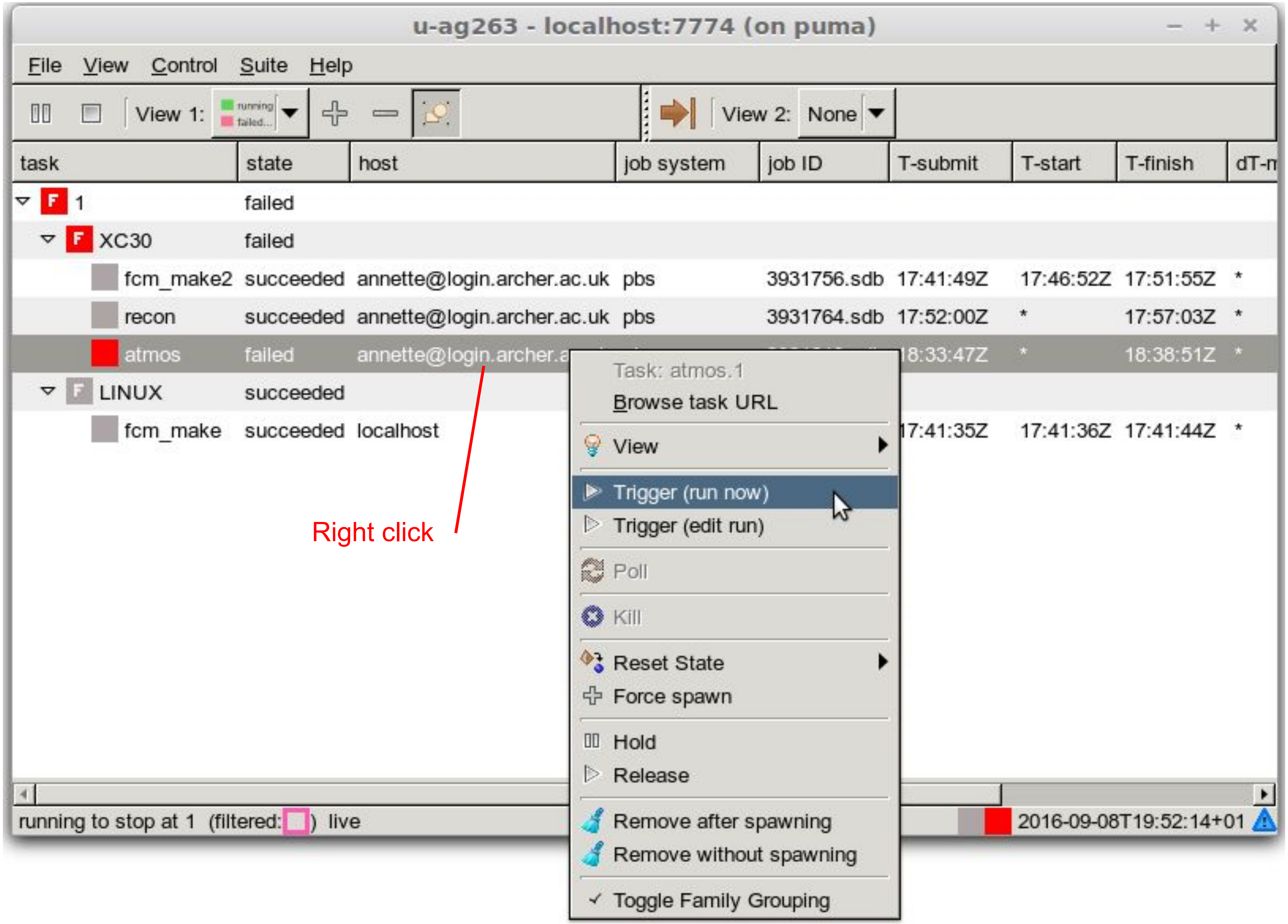
Reloading a suite

- You can make changes to a suite whilst it is running by reloading the suite definition.

```
rose suite-run --reload
```

- Warning: if you try to reload through the Cylc GUI - it will only pick up changes to the suite.rc file & not anything else.

- You can then re-trigger failed tasks.
- Note: the reload won't affect any tasks that have already been submitted. You will have to kill & re-trigger to run with the new changes.



task	state	host	job system	job ID	T-submit	T-start	T-finish	dT-m
1	failed							
XC30	failed							
fcm_make2	succeeded	annette@login.archer.ac.uk	pbs	3931756.sdb	17:41:49Z	17:46:52Z	17:51:55Z	*
recon	succeeded	annette@login.archer.ac.uk	pbs	3931764.sdb	17:52:00Z	*	17:57:03Z	*
atmos	failed	annette@login.archer.a			18:33:47Z	*	18:38:51Z	*
LINUX	succeeded							
fcm_make	succeeded	localhost			17:41:35Z	17:41:36Z	17:41:44Z	*

running to stop at 1 (filtered:) live

2016-09-08T19:52:14+01

Right click

Suite lifecycle

- Normally a suite will shutdown if it completes successfully.
- Usually if the suite fails, it will remain active until it times out (the timeout period is defined in the suite.rc file).
- To see what suites you have running:
  - `rose suite-scan` (command line listing)
  - `cylc gscan` (graphical view)
- Whilst a suite is active you can view it through the cylc GUI:
  - `rose sgc`
- Once it has stopped, you will have to look at rose-bush or the cylc-run directory to see whether it succeeded.
- Stopped suites can be restarted from where they left off.

Restarting a suite

Restarting a suite loads the state of a previous run, and allows you to continue on from that point, in order to:

- Fix a failed suite (without starting again from the beginning).
- Continue the run for longer.

Warning: note the difference between:

`rose suite-run --restart` : re-install & restart suite

- Use to update suite with any changes e.g. run length.
- Usually what you want to do.

`rose suite-restart` : restart without re-installing suite

- Will not pick up any suite changes.

When a suite restarts

- Rose/cylc checks the task states are all correct:
  - Because the suite can be shutdown or crash whilst the tasks are still running.
- If the previous run <u>succeeded</u>:
  - It submits from where it left off (continuing a long run).
- If the suite had <u>failed</u> tasks:
  - It leaves it as it is, allowing you to make any necessary fixes.
  - Then manually re-trigger failed tasks and the suite will continue on as normal.

Stopping a suite

To stop a running suite:

- Click the stop button in the GUI
- `rose suite-shutdown` or `rose suite-stop`

Different modes of shutdown:

(default)	stop the suite after current active tasks have finished
`-- --now`	stop immediately, orphaning current active tasks
`-- --kill`	stop after killing current active tasks

From the GUI select “Control -> Stop suite” to access options.

Further information

- For more information on the command line tools (for `fcm`, `rose` and `cylc`) use the command 'help', e.g.:

  `rose help suite-run` or
  `rose suite-run -h`

- There is a lot more that can be done with cylc, e.g. setting up complex workflows; adding new tasks at run-time; real-time triggering; configuring how the suite is visualised etc. See the documentation for details:

  https://cylc.github.io/cylc-doc/7.8.8/html/index.html
  http://metomi.github.io/rose/doc/html/index.html

- In the next presentation we go over Rose and cylc suite files.

NCAS
Unified Model Introduction

Part 5: Exploring suites

University of Leeds, 7-9 February 2023

Understanding Rose suites

A Rose suite consists of:

A task can be an application (app) such as the UM which is defined in the suite.

- Suite definition:
  - Describes tasks to be run and in what order
  - Describes where and how to run tasks
- App definitions:
  - Application settings such as input files, namelists etc
- Metadata:
  - Defines how settings are displayed in the GUI.
  - Can also provide help, logic, macros.
  - The UM picks up central metadata.

UM suite directory

Basic UM suite directory structure:

```
roses/u-cc519/
    rose-suite.conf
    rose-suite.info
    suite.rc
    app/
        fcm_make/
            rose-app.conf
            file/
                fcm-make.cfg
        um/
            rose-app.conf
    meta/
        rose-meta.conf
```

Suite definition & settings

Build settings:

- code branches, platform configuration

UM application settings:

- namelists, environment variables, file locations

Suite metadata

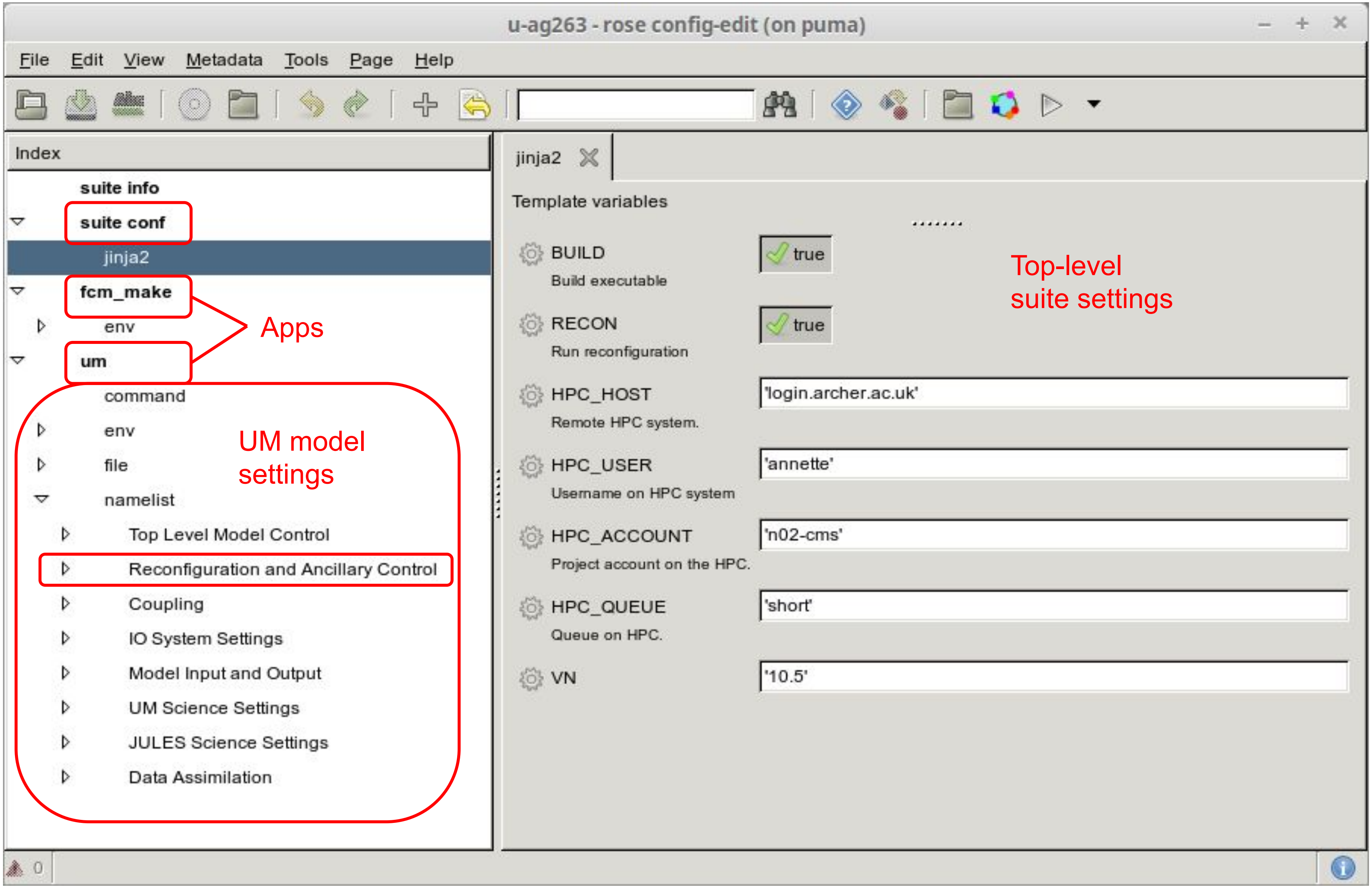
Basic UM suite

The basic UM suite consists of the following tasks:

`fcm_make`	Code extract and mirror	fcm_make app
`fcm_make2`	Code pre-processing and build	
`recon`	Reconfiguration	um app
`atmos`	Model run (with cycling)	

Task settings can be found under the app headings (fcm_make and um)

Basic UM suite



- suite info
- suite conf
  - jinja2
- fcm_make
  - env
- um
  - command
  - env
  - file
  - namelist
    - Top Level Model Control
    - Reconfiguration and Ancillary Control
    - Coupling
    - IO System Settings
    - Model Input and Output
    - UM Science Settings
    - JULES Science Settings
    - Data Assimilation

Template variables

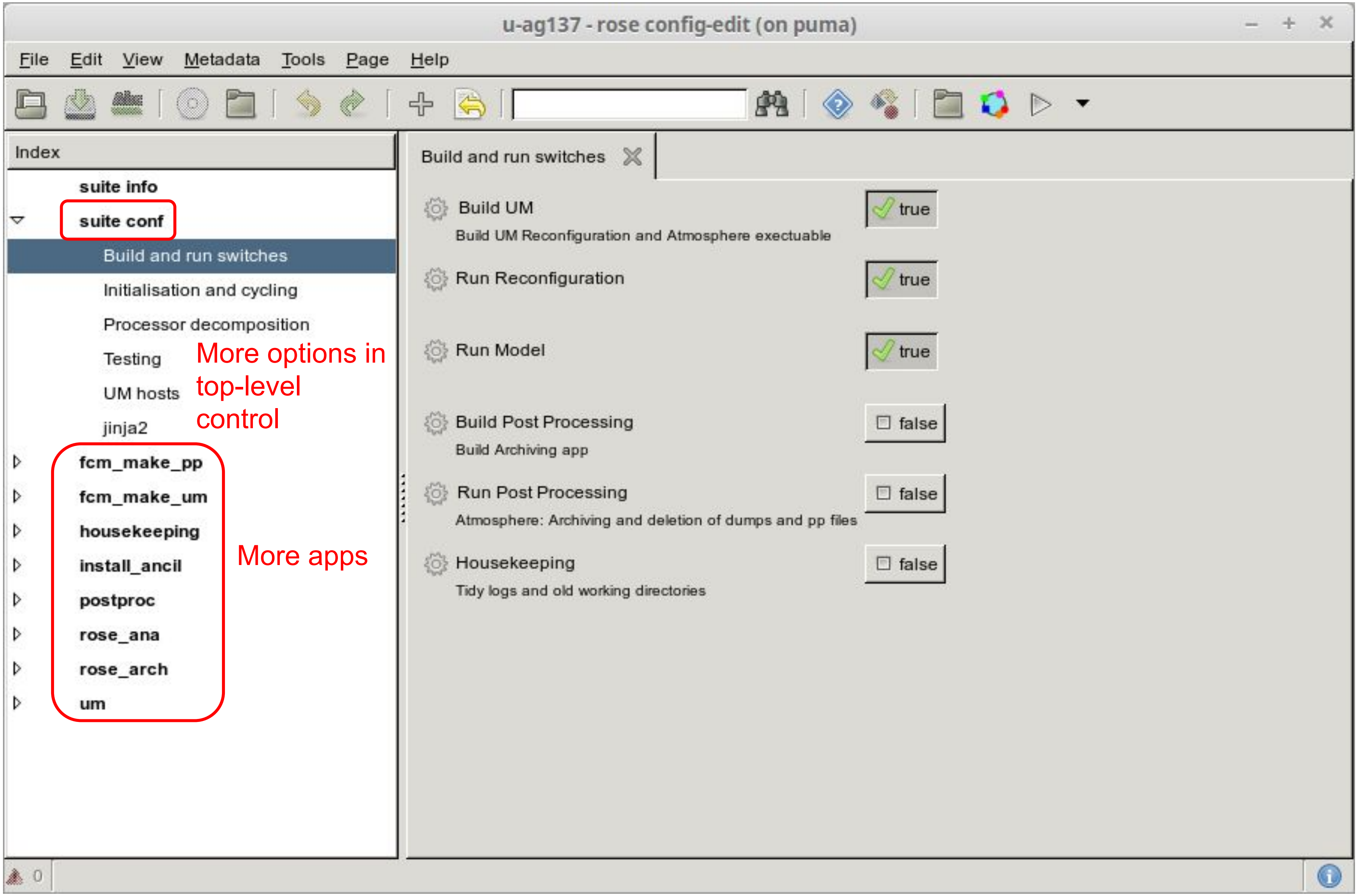
BUILD Build executable	✓ true
RECON Run reconfiguration	✓ true
HPC_HOST Remote HPC system.	'login.archer.ac.uk'
HPC_USER Username on HPC system	'annette'
HPC_ACCOUNT Project account on the HPC.	'n02-cms'
HPC_QUEUE Queue on HPC.	'short'
VN	'10.5'

Apps

UM model settings

Top-level suite settings

GA7 suite



Build UM ☑ true
Build UM Reconfiguration and Atmosphere exectuable

Run Reconfiguration ☑ true

Run Model ☑ true

Build Post Processing ☐ false
Build Archiving app

Run Post Processing ☐ false
Atmosphere: Archiving and deletion of dumps and pp files

Housekeeping ☐ false
Tidy logs and old working directories

More options in top-level control

More apps

GA7 apps

UM apps:

- install_ancil: Sets up ancillary files
- postproc: Data archiving and post-processing

Built-in Rose apps:

- housekeeping: Tidies up log files.
- rose_ana: Testing (e.g. to compare bit-reproducibility).
- rose_arch: Archiving of suite files.

Suite files

Top-level suite files:

`suite.rc`	Workflow & task definitions (cylc)
`rose-suite.info`	Rosie suite information (Rose)
`rose-suite.conf`	Suite settings (Rose)
`meta/rose-meta.conf`	Suite metadata (Rose)

Application files (under `app/.../`):

`rose-app.conf`	Application settings (Rose)
`file/fcm-make.cfg`	Build settings (FCM)

Additional suite files

Suite level:

`site/<machine>.rc`	Machine specific settings to be included in `suite.rc` file (Cylc).

Application level:

`opt/rose-app-<name>.conf`	Optional settings to be applied to application (Rose).

`meta/lib/`
`     etc/`
	Custom macros.

`bin/` `file/`	Other (non Rose/cylc format) files required by suite.

File formats

`rose-*.conf` files follow the Rose modified INI format:

- http://metomi.github.io/rose/doc/html/api/configuration/index.html

`suite.rc` and site include files follow the cylc extended-INI format with jinja2 templating:

- https://cylc.github.io/cylc-doc/7.8.8/html/appendices/site-user-config-ref.html#

If doing anything other than minor suite edits, it is worth looking at the documentation and working through the Rose tutorials.

Editing Rose suites

- The rose suite files are plain text so can be edited directly.
- The GUI just displays settings defined in these files according to the metadata.
  - So it is usually easier to use the GUI because of the extra information from the metadata.
- However: to edit the workflow (list of tasks & dependencies), and some of the runtime settings (depending on the suite), you will need to edit the `suite.rc` directly. The `suite.rc` file cannot be edited through the GUI.
- Warning: Some settings can look different in different suites.

Runtime settings

Most of the UM settings i.e. namelist entries are largely the same between suites as they use standard metadata.

However, certain runtime settings can appear in different places depending on the suite. These include:

- Model run length, cycling, start date
- Start file
- Job time-limits and number of processors
- Machine username and project code

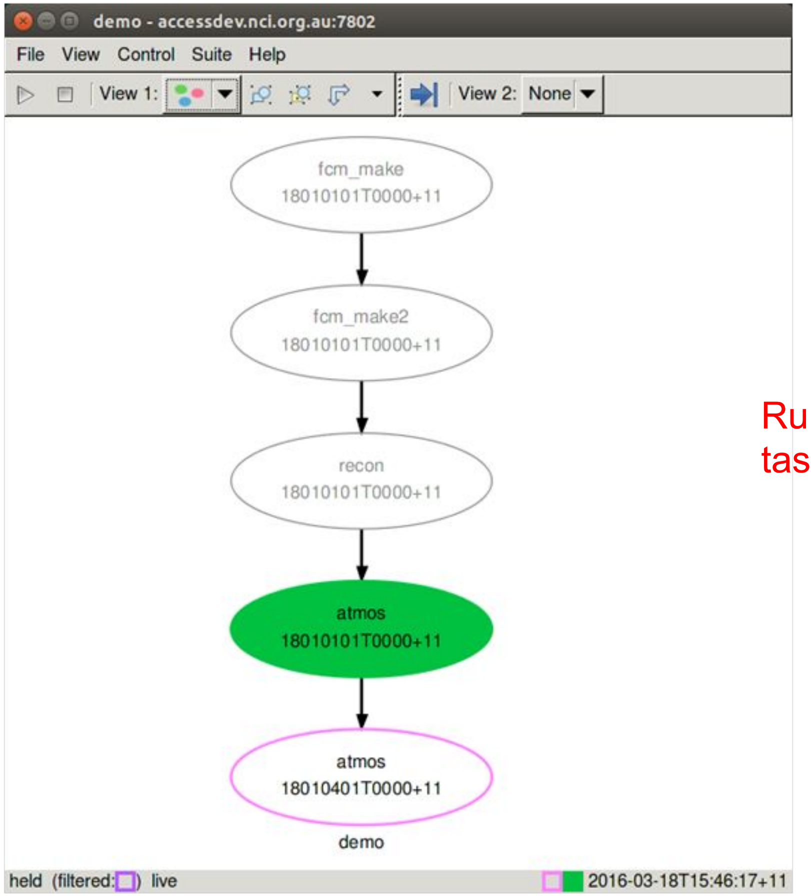
There is no standard format for what appears at the top-level suite-conf in the GUI - this is suite dependent. You may need to edit the `suite.rc` file directly.

suite.rc

Cylc suite configuration file (extended INI format), containing:

- Suite “workflow” in terms of:
  - Tasks (e.g. compilation, reconfiguration, and atmosphere run).
  - Dependencies between tasks (the order in which they should run) and cycling.
- Task definitions:
  - Each task runs a command or an app.
  - Need to specify where and how tasks are run (e.g. ARCHER2/Monsoon, time limit, project code etc).
- Suites can use inheritance and templating (though Jinja) to generate highly complex workflows.

Simple workflow



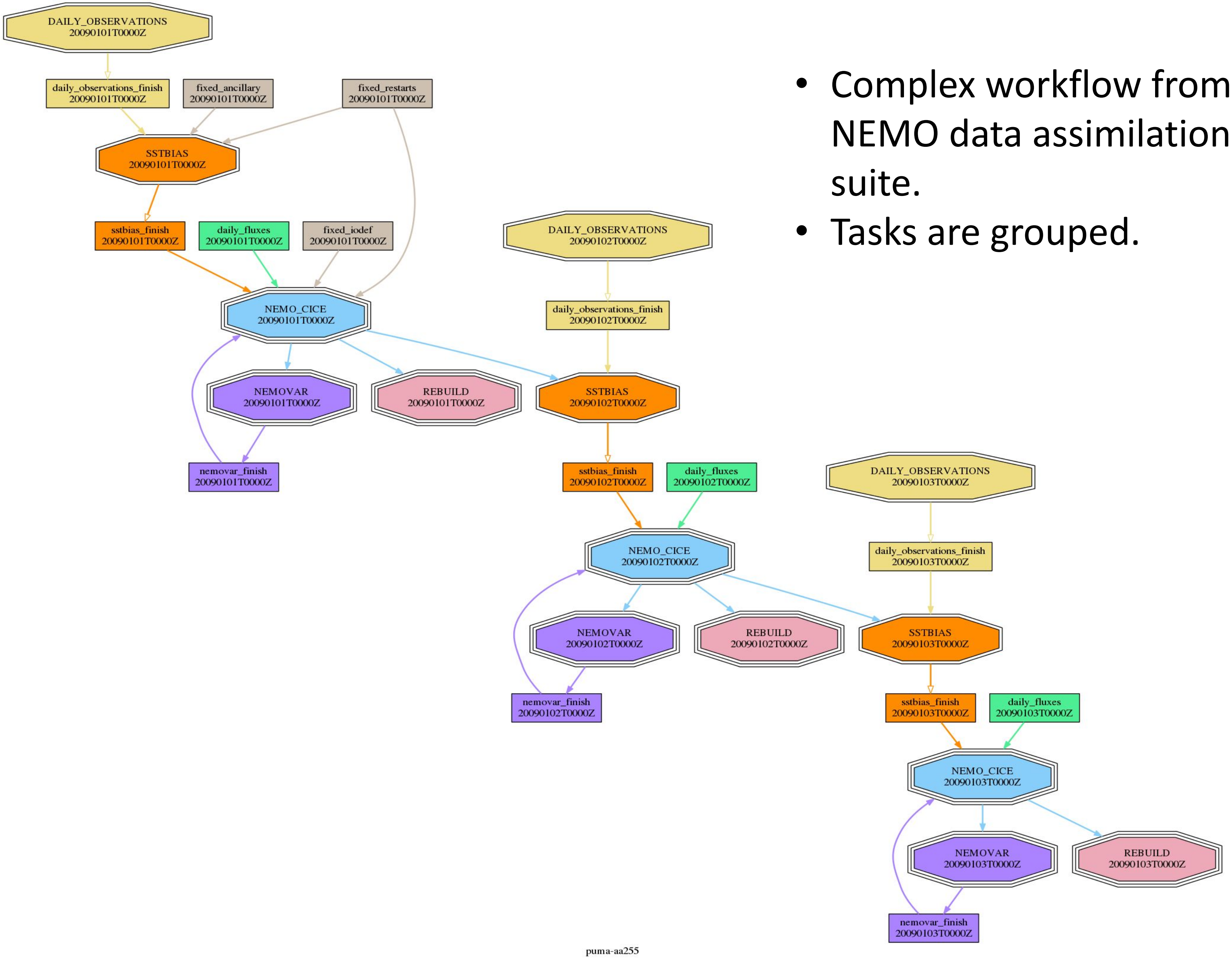
```
# suite.rc
[ scheduling ]
    initial cycle point = 18010101T0000
    final cycle point   = 18020101T0000

    [[ dependencies ]]
        [[[ R1 ]]]
            graph = “fcm_make => \
                     fcm_make2 => \
                     recon => atmos”
        [[[ P3M ]]]
            graph = “atmos[-P3M] => atmos”
```

Describes when tasks are run.

Run once - start up tasks

Cycling for long model runs.

(Courtesy of Scott Wales)


DAILY_OBSERVATIONS 20090101T0000Z
daily_observations_finish 20090101T0000Z
fixed_ancillary 20090101T0000Z
fixed_restarts 20090101T0000Z
SSTBIAS 20090101T0000Z
sstbias_finish 20090101T0000Z
daily_fluxes 20090101T0000Z
fixed_iodef 20090101T0000Z
DAILY_OBSERVATIONS 20090102T0000Z
NEMO_CICE 20090101T0000Z
daily_observations_finish 20090102T0000Z
NEMOVAR 20090101T0000Z
REBUILD 20090101T0000Z
SSTBIAS 20090102T0000Z
nemovar_finish 20090101T0000Z
sstbias_finish 20090102T0000Z
daily_fluxes 20090102T0000Z
DAILY_OBSERVATIONS 20090103T0000Z
NEMO_CICE 20090102T0000Z
daily_observations_finish 20090103T0000Z
NEMOVAR 20090102T0000Z
REBUILD 20090102T0000Z
SSTBIAS 20090103T0000Z
nemovar_finish 20090102T0000Z
sstbias_finish 20090103T0000Z
daily_fluxes 20090103T0000Z
NEMO_CICE 20090103T0000Z
NEMOVAR 20090103T0000Z
REBUILD 20090103T0000Z
nemovar_finish 20090103T0000Z
puma-aa255

- Complex workflow from NEMO data assimilation suite.
- Tasks are grouped.

Exploring complex suites

For complex suites it can be difficult to work out what the suite is going to do, because of:

- Complex dependencies and cycling.
- Task inheritance.
- Jinja code

To explore without running:

- Install files: `rose suite-run -i`
- View run graph: `cylc graph <suite-id>`

  Click button in GUI to ungroup tasks.

- Go to cylc-run directory and look at `suite.rc.processed` file which has evaluated jinja2 code.

Task definitions

```
[runtime]
...
    # Atmosphere Model Run
    [[atmos]]
       inherit = ARCHER2
       [[[directives]]]
          --nodes = 1
          --ntasks = 128
          --tasks-per-node = 128
          --cpus-per-task = 1
{% if HPC_QUEUE is defined %}
          --qos = {{ HPC_QUEUE }}
 {% if HPC_QUEUE == ‘short’ %}
          --reservation = shortqos
 {% endif %}
{% endif %}
      [[[environment]]]
          UM_INSTALL_DIR  = /work/y07/shared/umshared
          ROSE_TASK_APP    = um
          ASTART=../recon/atmos.astart
          FLUME_IOS_NPROC = 0
          OMP_NUM_THREADS = 1
          ROSE_LAUNCHER_PREOPTS="--cpu-bind=cores"
```

Definition for each task

Inherit settings

Slurm header

Jinja code

Variable set in rose-suite.conf

App to run

srun options

rose-suite.conf

Generally used to set top-level variables used in `suite.rc` file.

Apply these variables to suite.rc file

`rose-suite.conf`

```
[jinja2:suite.rc]
BUILD=true
HPC_ACCOUNT='n02-cms'
HPC_HOST='login.archer2.ac.uk'
HPC_QUEUE='short'
HPC_USER='annette'
RECON=true
VN='10.5'
```

`suite.rc`

```
[[ARCHER2]]
...

    [[[directives]]]
        --account={{HPC_ACCOUNT}}
    [[[job submission]]]
        method = slurm
    [[[remote]]]
        host = {{HPC_HOST}}
{% if HPC_USER is defined %}
        owner = {{HPC_USER}}
{% endif %}
```

Editing the suite.rc file

- Rose settings (variables that appear in `rose-suite.conf`) can be edited in the GUI.
- Cylc task settings (e.g. number of processors, OpenMP threads, time limit etc) may be variables or hard-wired in the `suite.rc` (or site include) file.
- The `suite.rc` file is also the place to edit:
  - Email notifications
  - Suite timeout limits
- UM `suite.rc` files can look different.
  - Settings may have different names & locations

Date and time format

Rose and cylc use the ISO 8601 date format to specify the start date, run length and cycling periods (CRuns).

Format:

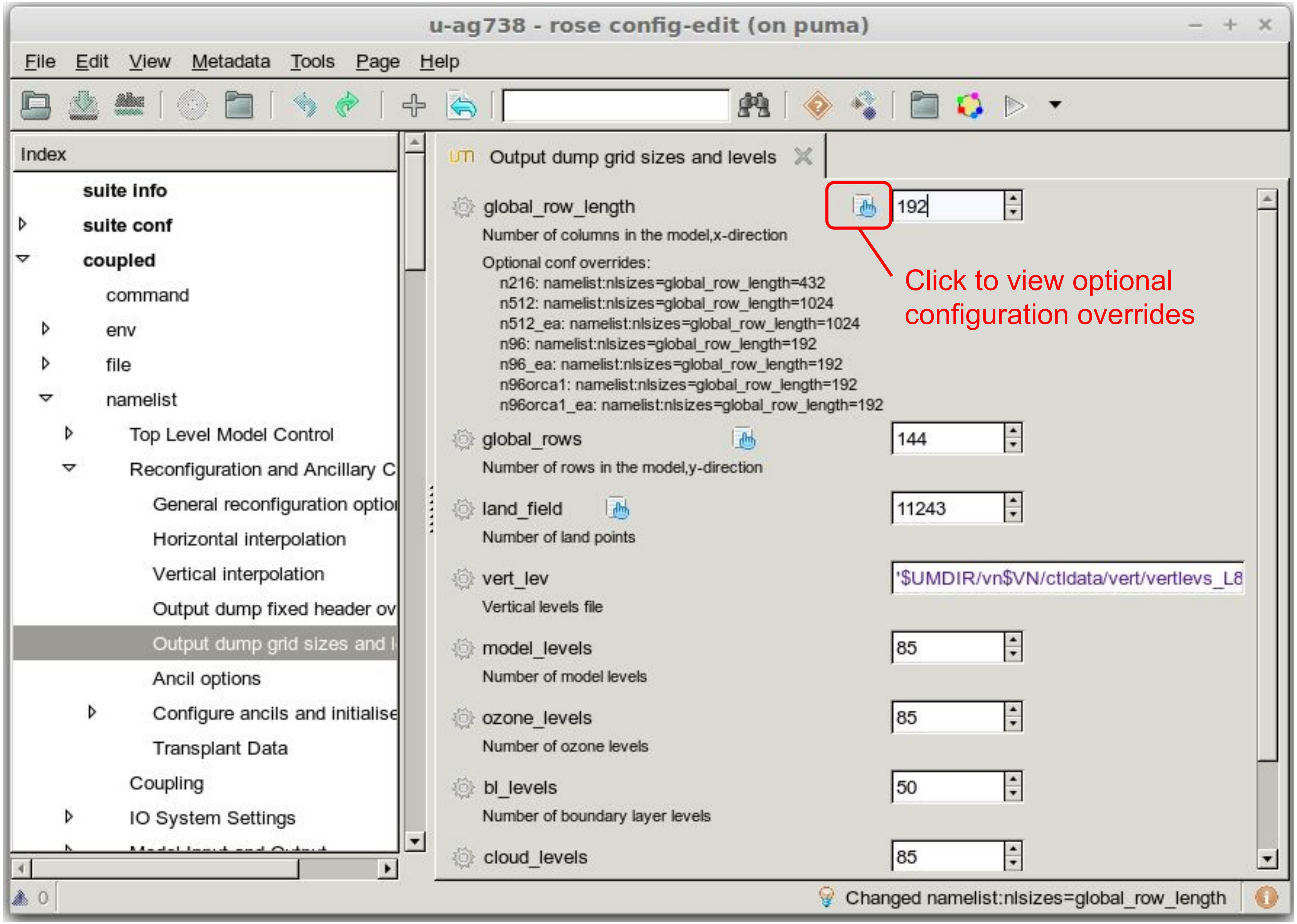
- Specific date & time: `CCYYMMDDThhmmZ`
  - E.g. `20160915T1430Z`
- Time period: `PnYnMnDTnHnMnS`
  - E.g. `P10Y6M`, `P3M`, `P1D`, `PT6H`

In some suites it may not be necessary to write ISO dates directly - they may be translated.

Common issues

- Be careful about editing suite files with the rose editor GUI open:
  - If you run the suite through the GUI or save the suite, any changes made outside the rose editor will be lost.
- Be aware of optional configurations:
  - You can’t make changes to these through the editor, but you can view them by clicking the icon.
- The `cylc_runs/` directory can fill up fast on PUMA:
  - These can usually be deleted as the logs are also on the HPC.

Optional configurations



u-ag738 - rose config-edit (on puma)
File Edit View Metadata Tools Page Help
Index
suite info
suite conf
coupled
command
env
file
namelist
Top Level Model Control
Reconfiguration and Ancillary C
General reconfiguration optio
Horizontal interpolation
Vertical interpolation
Output dump fixed header ov
Output dump grid sizes and l
Ancil options
Configure ancils and initialise
Transplant Data
Coupling
IO System Settings
Output dump grid sizes and levels

global_row_length 192
Number of columns in the model,x-direction
Optional conf overrides:
n216: namelist:nlsizes=global_row_length=432
n512: namelist:nlsizes=global_row_length=1024
n512_ea: namelist:nlsizes=global_row_length=1024
n96: namelist:nlsizes=global_row_length=192
n96_ea: namelist:nlsizes=global_row_length=192
n96orca1: namelist:nlsizes=global_row_length=192
n96orca1_ea: namelist:nlsizes=global_row_length=192

global_rows 144
Number of rows in the model,y-direction

land_field 11243
Number of land points

vert_lev '$UMDIR/vn$VN/ctldata/vert/vertlevs_L8
Vertical levels file

model_levels 85
Number of model levels

ozone_levels 85
Number of ozone levels

bl_levels 50
Number of boundary layer levels

cloud_levels 85

0 Changed namelist:nlsizes=global_row_length 1

Click to view optional configuration overrides

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

NCAS
Unified Model Introduction

Part 6: UM Data Files and IO

University of Leeds, 7-9 February 2023

Contents

- **Overview**
- UM File formats
- Model I/O
  - Model input
  - Model output

Overview

- Climate and weather models are nothing without input and output data.
- There are numerous data formats available.
- All consist of of files which contain the meteorological data and associated metadata.
- Metadata is "data [information] that provides information about other data".
- Data is useless without associated metadata.
- Generally the information contained in files of different formats will be the same.
- Different UM subsystems and analysis tools require different data formats.

Contents

- Overview
- **UM File formats**
- Model I/O
  - Model input
  - Model output

UM data formats

- This talk mainly concentrates on the UM atmospheric model.
- Historically the UM uses two propiritory data formats from the Met Office, fieldfiles and PP.
- UM Fieldsfiles are mainly used at runtime for direct model IO.
- PP files are derived from fieldsfiles and are used for data analysis.
- However, NetCDF is the main global standard for weather and climate data and its usage is increasing in the UM.
- There are numerous data format conversion tools available for the UM.

UM and PP formats

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

The two main types of UM file formats:

UM fieldsfile format (also called UM format, dump format, ancil format)

- Original format read and written by the model.
- Direct access files consisting of a primary header and a series of secondary headers that point to the data.
- 64-bit big-endian.

PP format

- Produced from UM fieldsfiles for data analysis (not used by the model).
- Files from the CEDA and UKMO archive may be in this format.
- Sequential files (header, data, header, data).
- 32-bit big-endian.

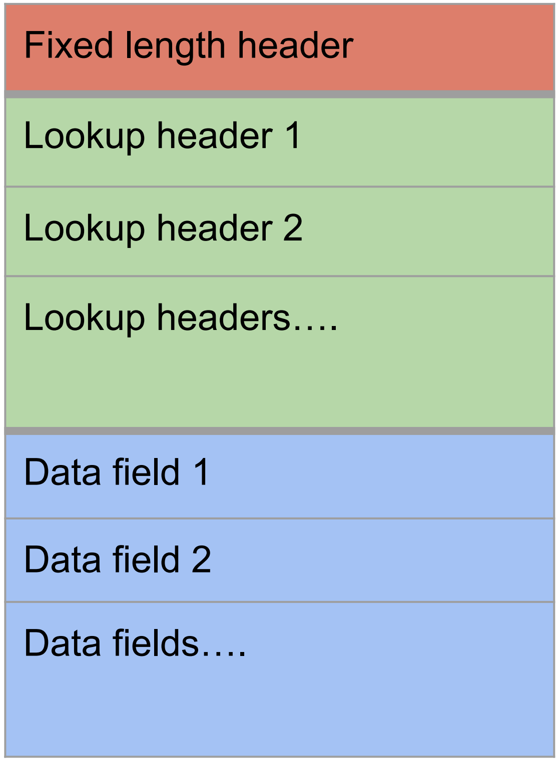
There can be confusion between these file types.

Described in UMDP F03:

https://code.metoffice.gov.uk/doc/um/latest/papers/umdp_F03.pdf

Fieldsfile and PP formats

UM Fieldsfile



Fixed length header

Lookup header 1

Lookup header 2

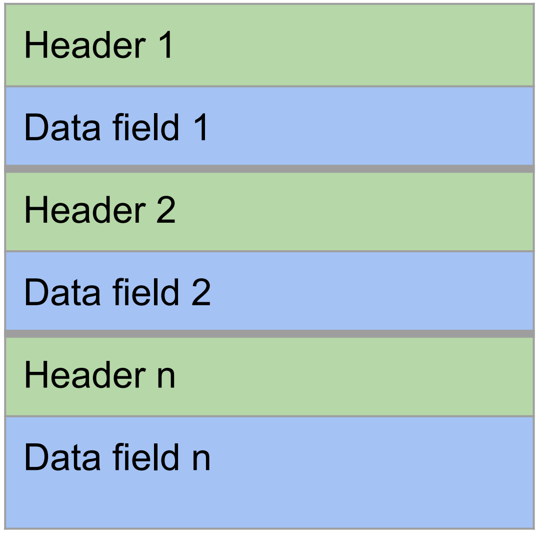
Lookup headers….

Data field 1

Data field 2

Data fields….

PP file



Header 1

Data field 1

Header 2

Data field 2

Header n

Data field n

Metadata and data are ordered differently.
Each data field is a 2D array.

(CF-)NetCDF

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Network Common Data Form:

- Portable and self-describing data format.
- Various supported and external libraries and tools.
- Widely used and a standard format for many scientific communities.

Climate and Forecast conventions:

- Set of standards for describing data (metadata).
- Definitive explanations of data variables (standard names).
- Definitions for temporal and spatial properties of the data.
- http://cfconventions.org/

NetCDF and the UM

- Common uses of NetCDF :
  - Input data sets may be in NetCDF format (e.g. CMIP6 scenarios).
  - Analysis tools/scripts may expect NetCDF.
  - NEMO and CICE use NetCDF files.
  - UKCA uses NetCDF for emissions and nudging.

- There are tools for converting between (CF-)NetCDF and UM/PP formats.

- The UM can now write CF-NetCDF directly (from vn10.9)

Contents

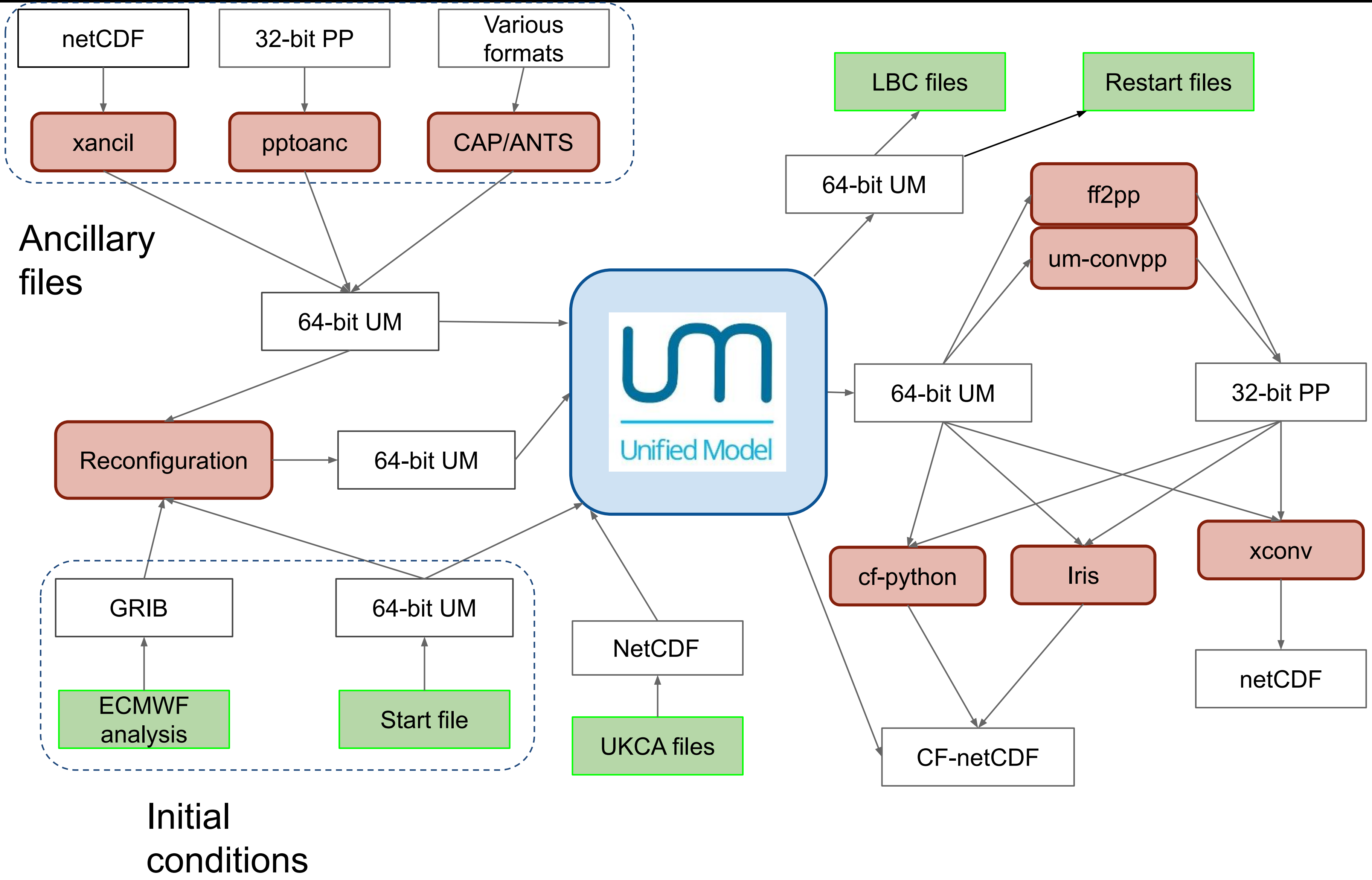
- Overview
- UM File formats
- **Model I/O**
  - Model input
  - Model output

UM files

Files in	Files out
Start files (also called initial files, dumps or restart files)	Restart files (or dumps)
Lateral boundary conditions (LBCs)	LBCs
Ancillary files - initial - forcing	Diagnostic files
Spectral files for radiation (ASCII)	Climate means
UKCA files (NetCDF)	

- Most model files (except for spectral and UKCA) are in UM fieldsfile format, with the header metadata distinguishing between dumps, ancillaries, LBCs and diagnostic files.

UM file infrastructure



Contents

- Overview
- UM File formats
- Model I/O
  - **Model input**
  - Model output

UM start dumps

- Start dumps provide the initial state for model variables.
- For climate models restart dumps from other UM runs normally used, which have to be processed by the reconfiguration system (more of which later).
- Start dumps for standard resolutions are available in `$UMDIR/`
- For NWP experiments or case studies, data for a <u>specific date</u> are available:
  - UKMO archives the last ~18 months of analyses.
  - For other dates, use ECMWF data and convert to a UM dump.
  - Contact the CMS helpdesk to request data and for advice on starting from ECMWF data.

UM ancillary files

- Ancillary files provide additional data to the model:
  - Initial data to be reconfigured into the start dump.
  - Fields to be updated regularly throughout the run.
- Normally contains predefined non-prognostic data such as land/sea mask, orography, emissions, SSTs and sea ice in atmosphere only models, land surface type...
- A number of tools are available to create ancillary files, detailed in another talk.
- Standard ancillary files can be found under

```
$UMDIR/ancil/atmos/
```

The Reconfiguration

- A compatible start dump may not be available for a particular model configuration.
- The reconfiguration is a standalone program within a suite which modifies (“reconfigures”) a UM start file.
- It ensures the start file has the correct prognostic fields (ie those variables which describe the model’s current state) required for the model to run.
- It can be used to:
  - Initialise fields with data from ancillary files
  - Initialise fields to zero, constants, missing data
  - Add new fields which may be absent in the initial dump.
  - Interpolate data to a new resolution or subdomain.

The Reconfiguration and ancillaries

- The reconfiguration is nearly always run at the start of any UM suite as part of the normal workflow process.
- Time invariant ancillaries are inserted into the initial dump file.
- Time varying ancillaries are updated while the model runs, and any pre-existing fields in the dump are ignored.
- It is good practice to initialise time-varying ancillaries to “missing data” in the reconfiguration section of the GUI, so that any inconsistencies will cause the model to fail.

Contents

- Overview
- UM File formats
- Model I/O
  - Model input
  - **Model output**

UM restart dumps

- Restart dump files contain model prognostic variables, and are written at regular intervals.
  - To restart the model for when things go wrong so that the run can be continued rather than starting again.
  - To continue long runs.
  - Provide standard initial conditions for other experiments.
  - Enable re-running of the model at later dates.

- Be aware that changing the dump frequency changes the results, so keep this fixed if reproducibility is important.

- The user can choose the dump archiving frequency.

Diagnostics and STASH

The UM has its own inbuilt diagnostic system, STASH:

Spatial and Temporal Averaging and Storage Handling

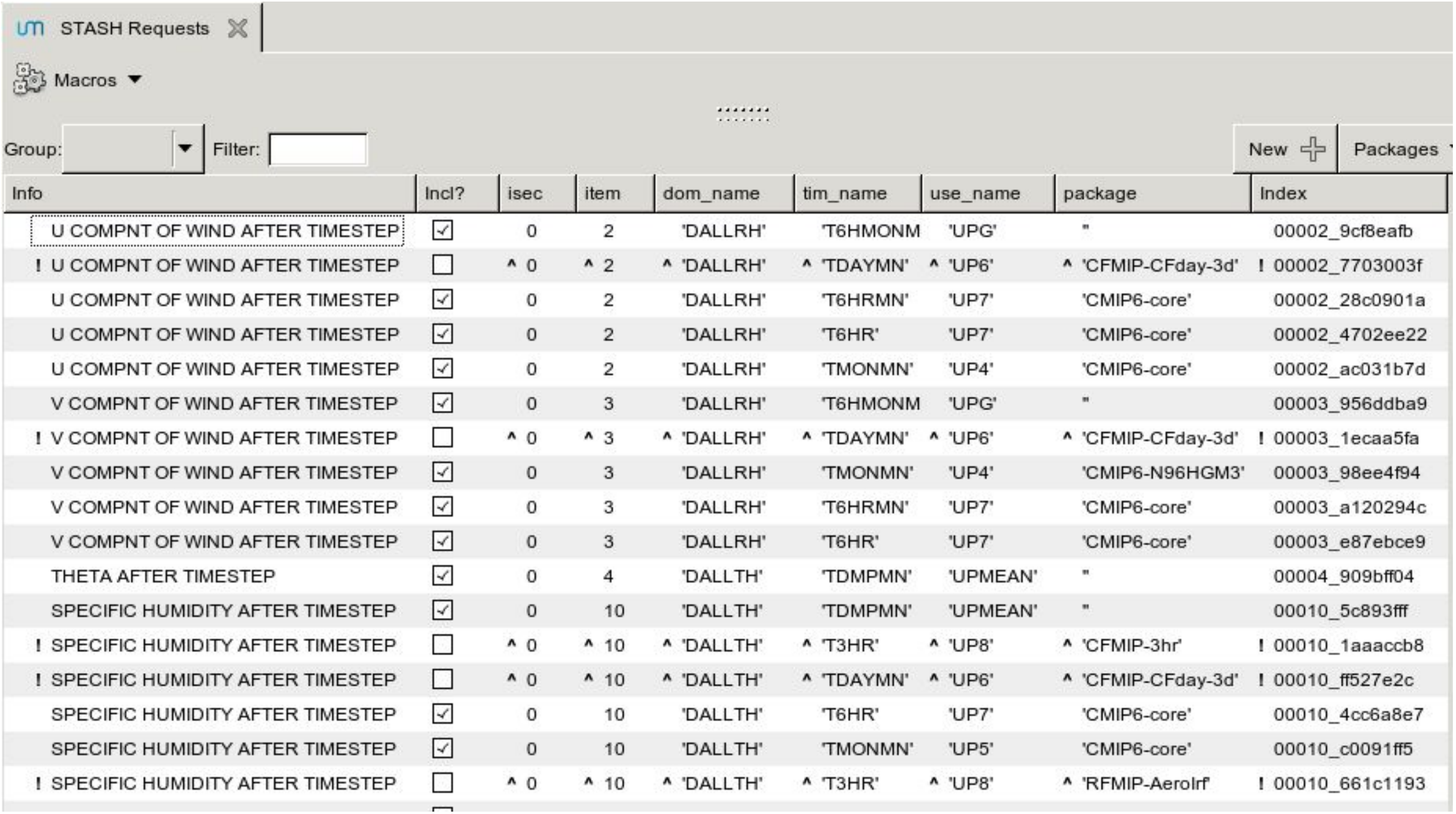
- All of the available runtime diagnostics has an unique identifier, a “STASH code”.
- STASH codes made up for 5 digits, the first two are the section number, the final three the item number.
- The section is a submodel or functional part of the model, such as LW radiation (02) or convection (05).
- STASH codes which start “00” are model prognostics.
- Examples:
  - 00024 Surface temperature
  - 05216 Total precip rate

STASH specification

For each output diagnostic, three profiles are used to specify how it is written, which roughly correspond to when, what and where to:

- **Time profile:**
  - When diagnostic is to be output: start & end time; frequency.
  - Time processing required: instantaneous; accumulation; mean (with sampling period); time series .
- **Domain profile:**
  - Vertical level type: model levels; interpolated pressure levels; soil levels...
  - Horizontal domain: limited area; land points; sea points...
  - Spacial meaning: [zonal, vertical, meridional, horizontal] with weights.
- **Usage profile:**
  - Output unit (file)

STASH requests table



Info	Incl?	isec	item	dom_name	tim_name	use_name	package	Index
U COMPNT OF WIND AFTER TIMESTEP	☑	0	2	'DALLRH'	'T6HMONM	'UPG'	''	00002_9cf8eafb
! U COMPNT OF WIND AFTER TIMESTEP	□	^ 0	^ 2	^ 'DALLRH'	^ 'TDAYMN'	^ 'UP6'	^ 'CFMIP-CFday-3d'	! 00002_7703003f
U COMPNT OF WIND AFTER TIMESTEP	☑	0	2	'DALLRH'	'T6HRMN'	'UP7'	'CMIP6-core'	00002_28c0901a
U COMPNT OF WIND AFTER TIMESTEP	☑	0	2	'DALLRH'	'T6HR'	'UP7'	'CMIP6-core'	00002_4702ee22
U COMPNT OF WIND AFTER TIMESTEP	☑	0	2	'DALLRH'	'TMONMN'	'UP4'	'CMIP6-core'	00002_ac031b7d
V COMPNT OF WIND AFTER TIMESTEP	☑	0	3	'DALLRH'	'T6HMONM	'UPG'	''	00003_956ddba9
! V COMPNT OF WIND AFTER TIMESTEP	□	^ 0	^ 3	^ 'DALLRH'	^ 'TDAYMN'	^ 'UP6'	^ 'CFMIP-CFday-3d'	! 00003_1ecaa5fa
V COMPNT OF WIND AFTER TIMESTEP	☑	0	3	'DALLRH'	'TMONMN'	'UP4'	'CMIP6-N96HGM3'	00003_98ee4f94
V COMPNT OF WIND AFTER TIMESTEP	☑	0	3	'DALLRH'	'T6HRMN'	'UP7'	'CMIP6-core'	00003_a120294c
V COMPNT OF WIND AFTER TIMESTEP	☑	0	3	'DALLRH'	'T6HR'	'UP7'	'CMIP6-core'	00003_e87ebce9
THETA AFTER TIMESTEP	☑	0	4	'DALLTH'	'TDMPMN'	'UPMEAN'	''	00004_909bff04
SPECIFIC HUMIDITY AFTER TIMESTEP	☑	0	10	'DALLTH'	'TDMPMN'	'UPMEAN'	''	00010_5c893fff
! SPECIFIC HUMIDITY AFTER TIMESTEP	□	^ 0	^ 10	^ 'DALLTH'	^ 'T3HR'	^ 'UP8'	^ 'CFMIP-3hr'	! 00010_1aaaccb8
! SPECIFIC HUMIDITY AFTER TIMESTEP	□	^ 0	^ 10	^ 'DALLTH'	^ 'TDAYMN'	^ 'UP6'	^ 'CFMIP-CFday-3d'	! 00010_ff527e2c
SPECIFIC HUMIDITY AFTER TIMESTEP	☑	0	10	'DALLTH'	'T6HR'	'UP7'	'CMIP6-core'	00010_4cc6a8e7
SPECIFIC HUMIDITY AFTER TIMESTEP	☑	0	10	'DALLTH'	'TMONMN'	'UP5'	'CMIP6-core'	00010_c0091ff5
! SPECIFIC HUMIDITY AFTER TIMESTEP	□	^ 0	^ 10	^ 'DALLTH'	^ 'T3HR'	^ 'UP8'	^ 'RFMIP-Aerolrf'	! 00010_661c1193

Exploring STASH

- STASH can look daunting:
  - Each STASH request & profile is listed by a hash-index (because of the way Rose processes namelists).
  - Select the top-level menu item (e.g. “STASH Requests” or “Domain Profiles”) for a summary table.

- Groups of STASH items may be grouped into **packages**, to be easily switched on and off.

- You can copy the diagnostics settings from one UM job to another using the **stash_copy** macros:
  - stash_copy.STASHExport.transform
  - stash_copy.STASHImport.transform

Adding STASH

To add a new STASH request:

- Select “New” from the summary page, then edit entry in table.
- Just because a diagnostic is available doesn’t mean it works.

To add a new profile:

- Copy (“Clone”) an existing profile, give it a new name, and make at least one change to it.

Important:

- Run the STASH macro (TidyStashTransfrom) to generate indices for any new requests or profiles.
- Run the checker (Validate) macros to verify STASH items are available and are set up correctly.

Output streams

- Diagnostics set up via STASH are sent to an output stream that corresponds to a post-processing file
  - This may be a single file throughout the run or
  - The file may be “reinitialised” periodically.

- Maximum number of “reserved headers” is set for each file. This can be changed in ROSE. It corresponds to number of 2D fields.
  - Set limit appropriately.
  - Reinitialise files to stop the limit being exceeded.

- Files may be “packed” to save space - different packing profiles available.

Climate meaning

- System for generating long term means in a manageable way.
- Meaning periods are related to the dump frequency.
  - Typically, 10 day dumps and meaning periods of 3, 3, 4 and 10 → monthly, seasonal, annual and decadal means for 360 day models. 1 day dumps and real months, etc for Gregorian.
- To send diagnostics to climate mean system, tag with appropriate usage profile (conventionally called UPMEAN).
- Output files:
  - Usually of the form $RUNIDa.mm…, $RUNIDa.ms…
  - Temporary partial sum files written to preserve precision across restarts (of the form $RUNIDa_s2[ab]).
- Can be inefficient due increased IO required, especially for larger models, and is gradually being replaced by a CYLC controlled post-processing system.

Ocean and sea-ice

NEMO files:

- All in NetCDF.
- Diagnostic files can be written by IO server XIOS.
- May produce global output files or multiple files each covering a subdomain.
- Construct global file after run using **rebuild_nemo**, e.g.:

```
rebuild_nemo   xdodto_CU150_19780901_19780930_grid_T   16
```

CICE files:

- Restart files may be in binary or NetCDF format
  - There are tools to edit the start date and convert to NetCDF (contact NCAS-CMS for guidance)
- All other inputs and outputs in NetCDF (global files)

Appendix

Useful information for advanced users.

STASH macros

stash_testmask.STASHTstmskValidate	Check STASH output is valid given science configurations
stash_indices.TidyStashValidate	Checks if STASH and item namelists have the correct index.
stash_indices.TidyStashTransform	Correct any namelist indexes. Run this when adding new STASH requests or profiles.
stash_indices.TidyStashTransformPruneDuplicated	As above, plus removes duplicate entries.
stash_requests.StashProfilesValidate	Checks for problems with referencing between stash requests and profiles.
stash_requests.StashProfilesRemoveUnused	Removes any profile namelists which aren't being referenced by any stash requests.

STASHmaster file

The STASHmaster file describes the characteristics of all of the model prognostic and diagnostic fields including its grid, when the field is available etc.

- Full details in UMDP C4: https://code.metoffice.gov.uk/doc/um/latest/umdp.html#C04

The STASHmaster file lives in the UM code trunk. The file needs to be updated whenever a new ancillary, prognostic or diagnostic is added via new code in the UM.

- Either edit in a branch or set your suite to use a local STASHmaster file.
- See the UM Rose training on MOSRS for instructions: https://code.metoffice.gov.uk/doc/um/latest/um-training/stashmaster.html

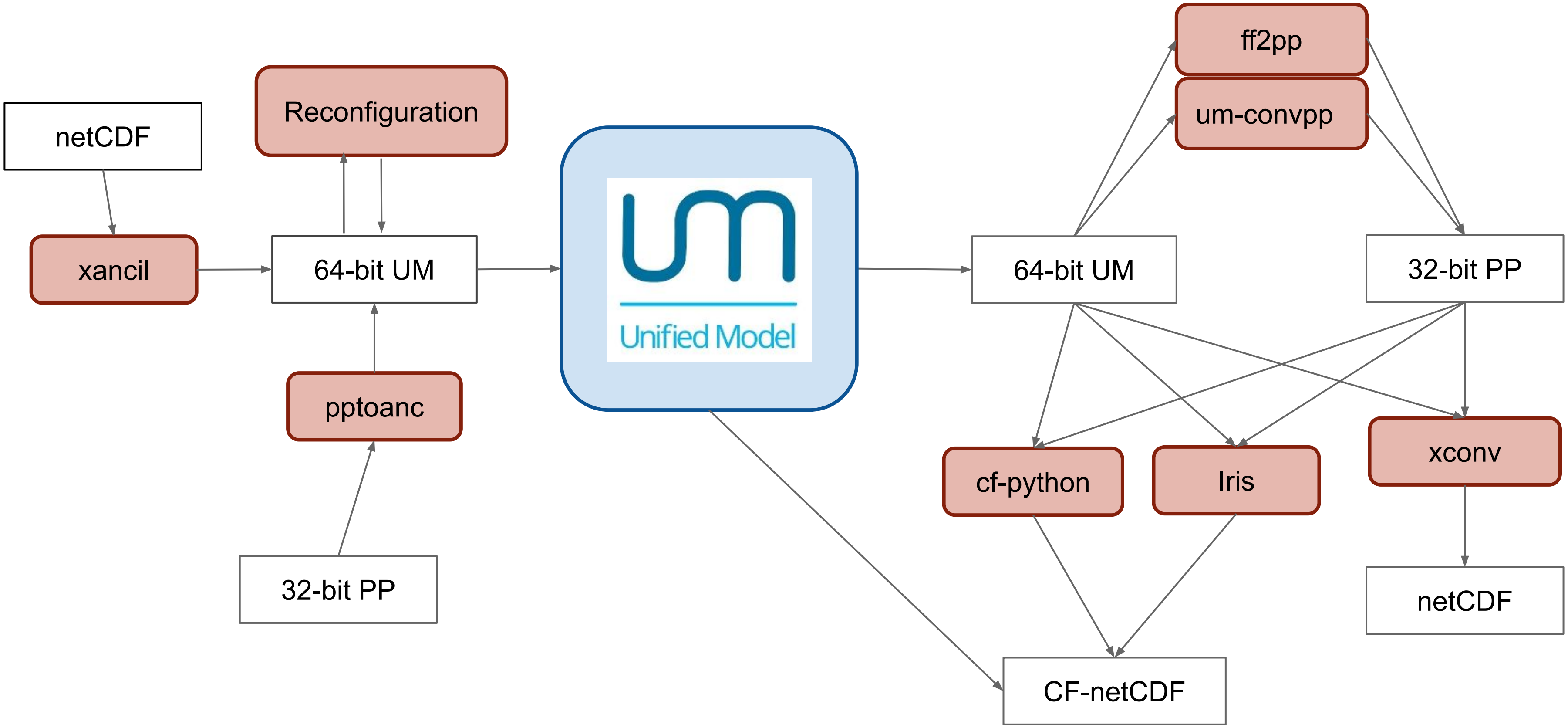
NCAS
Unified Model Introduction

Part 7: UM tools and utilities

University of Leeds, 7-9 February 2023

UM file conversion

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL



netCDF

xancil

Reconfiguration

64-bit UM

pptoanc

32-bit PP

ff2pp

um-convpp

64-bit UM

32-bit PP

cf-python

Iris

xconv

netCDF

CF-netCDF

Conversion tools

Tool	**Input**	**Output**
xancil	netCDF	UM (ancillary)
um-convpp	UM	PP (big-endian)
ff2pp	UM (fieldsfile)	PP (native-endian)
xconv	UM, PP, netCDF, GRIB, Grads	netCDF, Grads
cf-python	UM, PP, netCDF	CF-netCDF
Iris	UM, PP, netCDF, GRIB	CF-netCDF

UM file tools

Tool	**Input**	**Purpose**
um-cumf	UM	Compares two files. Can be used to check for NaNs in data.
um-pumf	UM	Prints header information.
uminfo	UM	Prints header information.
ppinfo	PP	Prints header information in a readable format.
createBC	UM	Creates LBCs for limited area model.

Mule

- Python API for reading and writing UM format files.
- Set of utilities for manipulating UM files (replaces old UM utilities):

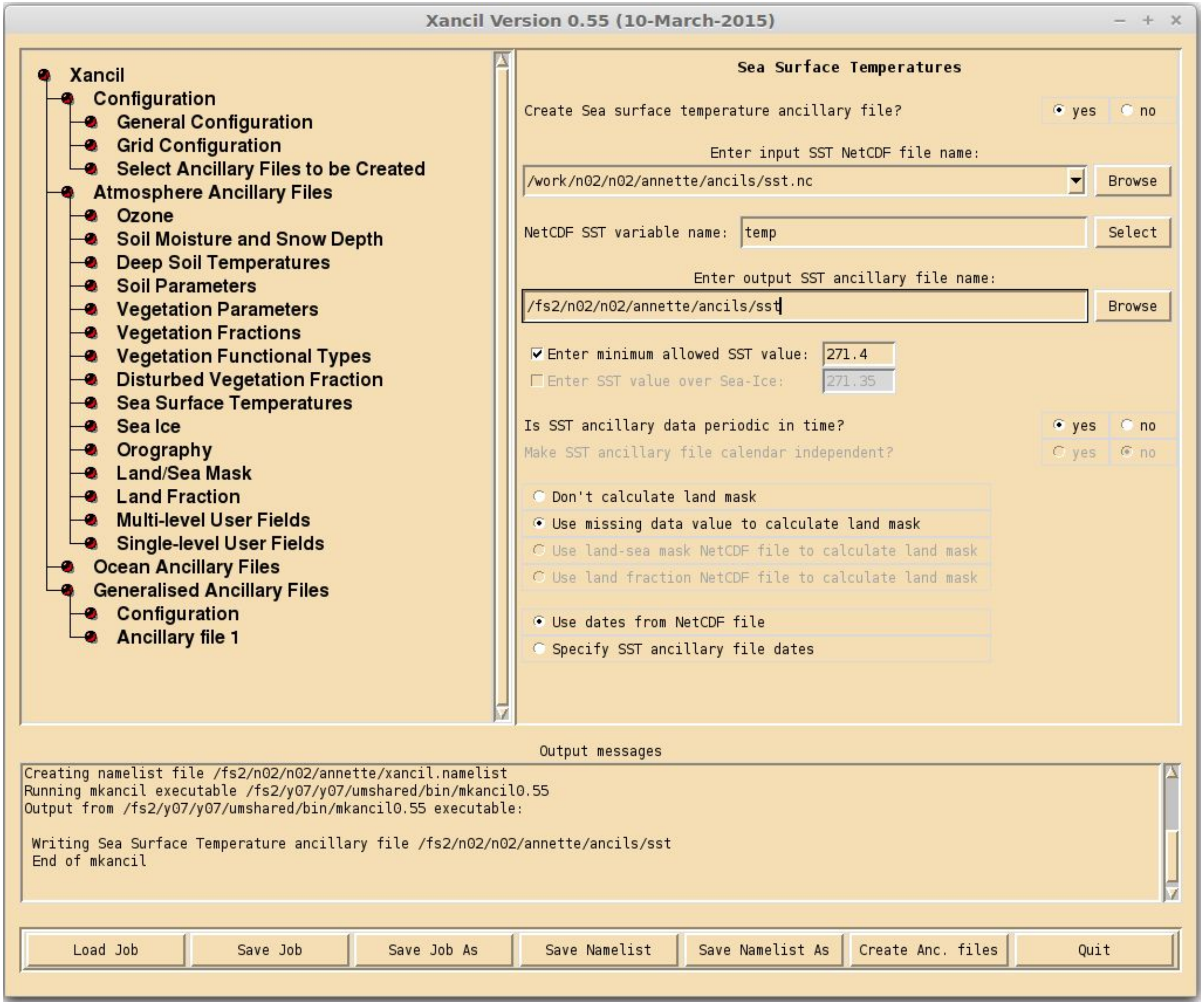
mule-cumf	Compare files
mule-cutout	Extract subregion from fixed resolution file
mule-fixframe	Convert MakeBC frame to CreateBC frame
mule-pumf	Pretty-print file headers
mule-select	Copy selected fields
mule-summary	List field lookup headers
mule-trim	Extract fixed region from a variable grid
mule-unpack	Unpack WGDOS packed files

https://code.metoffice.gov.uk/doc/um/index.html

Xancil

- Xancil provides a GUI for generating ancillary files from your own data.
  - Input data should be in NetCDF format.
  - Supports the creation of standard atmosphere ancillary files (and old UM ocean ancillary files).
  - Allows the creation of custom ancillary files.

- Xancil can take command-line options.
  - Complex workflows can be automated using tcl scripts.



Sea Surface Temperatures

Create Sea surface temperature ancillary file? ● yes ○ no

Enter input SST NetCDF file name: /work/n02/n02/annette/ancils/sst.nc

NetCDF SST variable name: temp

Enter output SST ancillary file name: /fs2/n02/n02/annette/ancils/sst

☑ Enter minimum allowed SST value: 271.4

□ Enter SST value over Sea-Ice: 271.35

Is SST ancillary data periodic in time? ● yes ○ no

Make SST ancillary file calendar independent? ○ yes ● no

○ Don't calculate land mask

● Use missing data value to calculate land mask

○ Use land-sea mask NetCDF file to calculate land mask

○ Use land fraction NetCDF file to calculate land mask

● Use dates from NetCDF file

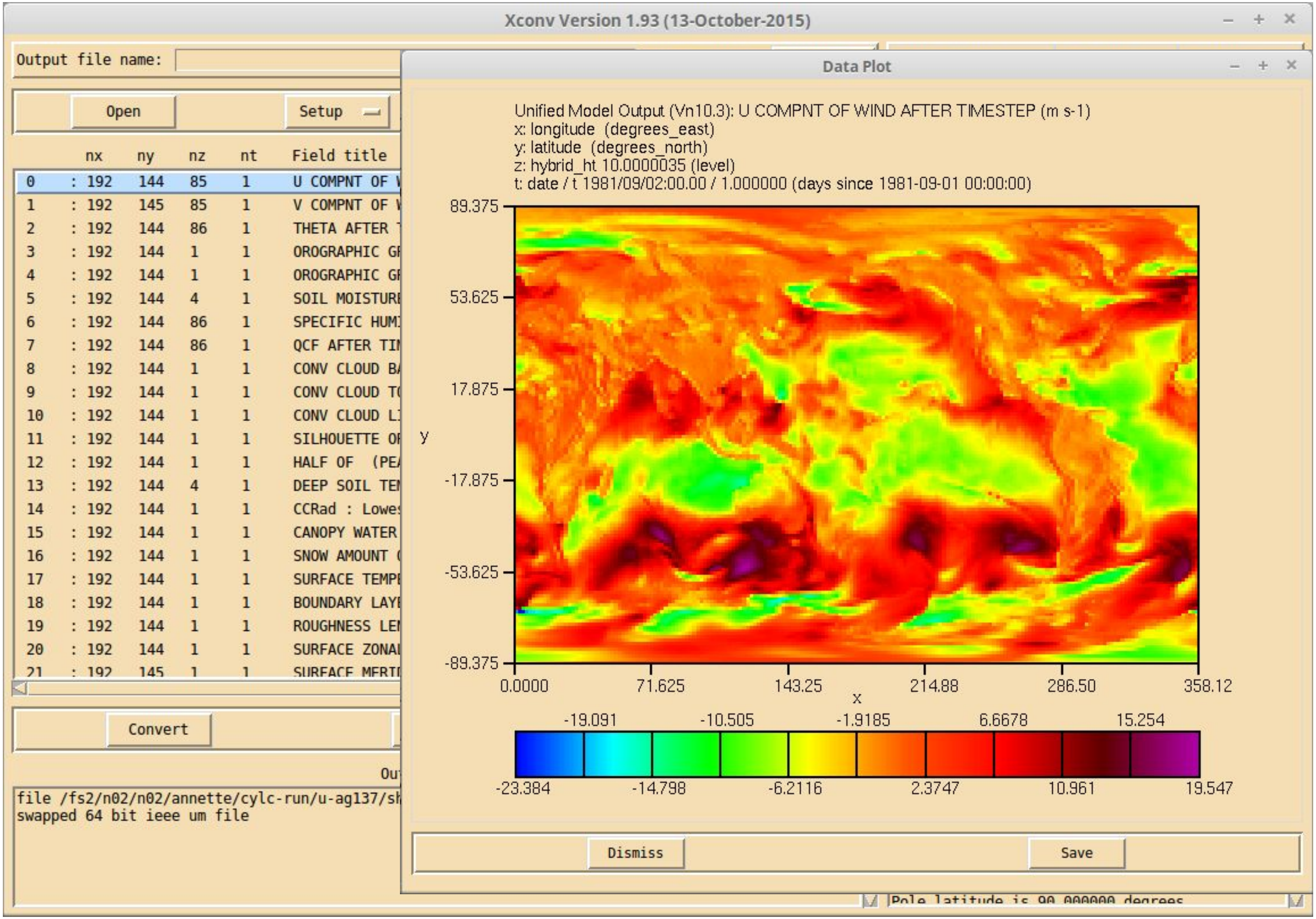
○ Specify SST ancillary file dates

Output messages

```
Creating namelist file /fs2/n02/n02/annette/xancil.namelist
Running mkancil executable /fs2/y07/y07/umshared/bin/mkancil0.55
Output from /fs2/y07/y07/umshared/bin/mkancil0.55 executable:

 Writing Sea Surface Temperature ancillary file /fs2/n02/n02/annette/ancils/sst
 End of mkancil
```

Xconv

- Xconv is useful for quickly viewing the contents of a data file, creating quick plots of fields, and viewing numerical data entries directly.
- Reads UM, PP, netCDF, GRIB and Grads.
- Converts data to netCDF.
- Data manipulations available:
  - Spectral to gridpoint
  - Interpolation (bilinear or area-weighted)
  - Conversion to and from rotated grids



```
      nx    ny    nz    nt    Field title
0   : 192   144   85    1     U COMPNT OF W
1   : 192   145   85    1     V COMPNT OF W
2   : 192   144   86    1     THETA AFTER T
3   : 192   144   1     1     OROGRAPHIC GR
4   : 192   144   1     1     OROGRAPHIC GR
5   : 192   144   4     1     SOIL MOISTURE
6   : 192   144   86    1     SPECIFIC HUMI
7   : 192   144   86    1     QCF AFTER TIM
8   : 192   144   1     1     CONV CLOUD BA
9   : 192   144   1     1     CONV CLOUD TO
10  : 192   144   1     1     CONV CLOUD LI
11  : 192   144   1     1     SILHOUETTE OR
12  : 192   144   1     1     HALF OF  (PEA
13  : 192   144   4     1     DEEP SOIL TEM
14  : 192   144   1     1     CCRad : Lowes
15  : 192   144   1     1     CANOPY WATER
16  : 192   144   1     1     SNOW AMOUNT (
17  : 192   144   1     1     SURFACE TEMPE
18  : 192   144   1     1     BOUNDARY LAYE
19  : 192   144   1     1     ROUGHNESS LEN
20  : 192   144   1     1     SURFACE ZONAL
21  : 192   145   1     1     SURFACE MERID
```

```
file /fs2/n02/n02/annette/cylc-run/u-ag137/sh
swapped 64 bit ieee um file
```

```
Pole latitude is 90.000000 degrees
```

Unified Model Output (Vn10.3): U COMPNT OF WIND AFTER TIMESTEP (m s-1)
x: longitude (degrees_east)
y: latitude (degrees_north)
z: hybrid_ht 10.0000035 (level)
t: date / t 1981/09/02:00.00 / 1.000000 (days since 1981-09-01 00:00:00)

Central Ancillary Program

- For creating standard ancillary files can use CAP
  - Orography
  - Land-sea mask
  - Soil moisture, snow
  - Vegetation
  - Aerosol
  - Sea surface temperature, sea-ice
  - Ozone
- Typically used for running the LAM:
  - Run as part of the nesting suite.
- Contact the CMS helpdesk for more information:
  https://cms-helpdesk.ncas.ac.uk/

ANTS

ANcillary Tools and Suites

- Developed by the Met Office
- Based on Python and Iris
- Gradually replacing functionality of CAP

https://code.metoffice.gov.uk/trac/ancil

Contains:

- Set of applications to:
  - Convert (Iris) data to ancillary files
  - Derive standard ancillary fields e.g. land-sea mask, vegetation etc
- Python tool kit for developing your own applications

Iris

Iris is a Python library for analysing and visualising meteorological data sets.

With Iris you can:

- Use a single API to work on your data, irrespective of its original format.
- Read and write (CF-)netCDF, GRIB 1&2, PP files and UM fields files.
- Easily produce graphs and maps via integration with *matplotlib* and *cartopy.*

Community tool developed at the Met Office:

https://scitools-iris.readthedocs.io/en/latest/

Availability of tools

- The UM file utilities (including mule) are available on ARCHER2 and Monsoon under:
  - `$UMDIR/bin`
  - `$UMDIR/vnX.Y/cce/utilities`

- CMS-developed tools (xconv, xancil, cf-python, cf-plot) can be downloaded to your own platform.

- Further information can be found through the CMS site:
  https://cms.ncas.ac.uk/tools-and-utilities/

jaspy

- Large collection of tools commonly used for atmospheric and Earth observation science, including:
  - netCDF and nco tools
  - cf-python, cf-plot and Iris
  - R ...

- Installed on the JASMIN systems at CEDA
- Reproducible environments

https://help.jasmin.ac.uk/article/4729-jaspy-envs-py3-rhel6-rhel7
https://github.com/cedadev/jaspy-manager

NCAS
Unified Model Introduction

Part 8: Finale

University of Leeds, 7-9 February 2023

Before contacting the helpdesk

1. Always search the helpdesk before raising a query. You may find that your query has already been answered.
2. On ARCHER2 please make your /home and /work directories readable by the CMS team so that we can help with any queries.

```
chmod -R g+rX /home/n02/n02/<username>
chmod -R g+rX /work/n02/n02/<username>
```

When contacting the CMS Helpdesk please make sure you supply the following information to help us answer your query more quickly.

- Cut and paste any error messages
- The id of the UM suite (e.g. u-aa774)
- The path to the log file containing the error.

UK UM support

- Through NCAS-CMS (National Centre for Atmospheric Science – Computational Modelling Services):

  http://cms.ncas.ac.uk

- Modelling helpdesk:

  https://cms-helpdesk.ncas.ac.uk/

- Email contact:

  cms-support@ncas.ac.uk

- Use the Helpdesk where possible (rather than emailing the team directly).

Other sources of support

ARCHER helpdesk: support@archer2.ac.uk

Monsoon helpdesk: monsoon@metoffice.gov.uk

Issue	**ARCHER2**	**Monsoon**
New account	CMS	Monsoon
Over quota / budget run out	CMS	Monsoon
UM specific issues	CMS	CMS
Machine problems	ARCHER2	Monsoon

Other sources of information

- UM documentation is available on MOSRS:
  https://code.metoffice.gov.uk/doc/um/latest/umdp.html

- Monsoon user guide.
  https://code.metoffice.gov.uk/doc/monsoon2/index.html

- Collaboration *twiki*
  https://gws-access.jasmin.ac.uk/public/mohc_shared/monsoon2/html

The CMS team

- Modelling support team:
  - UM: David Case, Jeff Cole, Annette Osprey, Simon Wilson, Rosalyn Hatcher, Grenville Lister
  - UKCA: Luke Abraham
  - Land surface: Patrick McGuire
- PUMA system:
  - Andy Heaps
- Strategic projects:
  - Valeriu Predoi, Marc Stringer (UKESM, ESMVAlTool)
  - David Hassell, Andy Heaps, Sadie Bartholomew (cf-python/plot)
  - Simon Wilson (NGMS)
  - David Livings (Hydro-JULES)
  - Jeff Cole (EXCALIBUR)

What CMS provides (i)

Machines and facilities:

- Computing time on ARCHER2
- Support for the UM on national HPC facility (ARCHER2) and Met Office/NERC collaboration HPC service (Monsoon2).
- A common platform (PUMA) for UM submission.
- Support for the UM on local computing facilities (e.g. Mobilis, HPC Wales).

Core activities:

- Training days and user meetings
- Documentation and information services
- Porting certain “standard” UM suites.

What CMS provides (ii)

Strategic projects:

- Development and support for visualisation and analysis tools (xconv, xancil, cf-python, cfplot).
- Modelling support and development through collaborative projects:
  - For example: UKHiGEM, CASCADE, CLIMIP, High resolution modelling, UKESM, SWAMMA, ParaCon.
- Future compute and storage technologies:
  - Software defined storage
  - Virtualisation and containerisation

Things to do next

- Register for an ARCHER2 or Monsoon2 account
- Finish working through the tutorial exercises:
- Find a suite that is similar to the one you want to run. Look at its options and settings in the rose editor.
- Read the UM user guide:

  https://code.metoffice.gov.uk/doc/um/latest/umdp.html#000
- Register for a JASMIN account:

  http://help.ceda.ac.uk/article/189-get-started-with-jasmin