IS-ENES2 DELIVERABLE (D -N°: 4.1)

Configuration management initial workshop report

File name: IS-ENES2_D4.1.pdf

Authors (v1):	Mick Carter Mike Rezny	Reviewers (v1):	Dave Matthews Reinhard Budich Glenn Carver
Author (v2):	Mick Carter	Reviewer (v2):	Reinhard Budich

Reporting period: *01/04/2013 – 30/09/2014*

Release date for review: 15/06/2015 Final date of issue: 16/06/2015

Revision table			
Version	Date	Name	Comments
1	02/01/14	Mick Carter	First formal release
2	15/06/15	Mick Carter	Re-released after request at project mid-term review.

Abstract

IS-ENES2 work-package 4 (WP4), Task 2 focuses on Configuration Management Tools via two workshops and a supported community evaluation of the FCM tool in particular.

Based on the first workshop, the author recommends that the community should move to extending the scope of existing configuration management tools to include, as far as possible, (i) experiment design as well as model code; (ii) the complete workflow through to final results and (iii) tools and processes to support the management of the code base to improve code quality by tracking testing and review processes. As partner sites have working change control systems, it is not appropriate to recommend that people with existing solutions should move to FCM. Where, in the wider community, institutions do not have such tools, they are encouraged to evaluate FCM. In particular, the strength of the FCM build systems in dealing with Fortran code dependencies and delivering a parallelized make solution is recommended as a complement to existing, off the shelf change control solutions, such as Subversion and Git.

Project co-**funded by the European Commission's Seventh Framework Programme (FP7; 2007**-2013) under the grant agreement n°312979

Dissemination Level		
PU	Public	
PP	Restricted to other programme participants including the Commission Services	
RE	Restricted to a group specified by the partners of the IS-ENES2 project	
CO	Confidential, only for partners of the IS-ENES2 project	

Table of contents

Executive Summary

1. Background

The description of work for NA3 Task 2 on configuration management states:

The FCM (Flexible Configuration Management) system developed at the Met Office for both configuration management and building code is now also being used at CNRS-IPSL. In both organisations, its use has extended to new projects as a result of voluntary uptake.

The networking activity will share those experiences at two workshops organised by the Met Office and support a community evaluation at both CNRS-IPSL as experienced users, where the Met Office will first respond to requests, and at MPG as first time users, where the Met Office will support the migration of model code to FCM.

The evaluations will feed into the second workshop where the aim will be to encourage more uptake of FCM if it is considered suitable for recommendation within the ENES community.

The first of these workshops was held at the Met Office on Monday 23rd and Tuesday 24th of September to deliver these aims. The agenda and attendees of the workshop are provided in Appendix 2. There were 20 attendees from 8 institutions including representatives from outside the work-package and from outside the IS-ENES2 consortium. External contributors were:

- Glenn Carver and Paul Burton from ECMWF who were able to bring their experience using FCM to the workshop
- Balaji from the University of Princeton who was able to bring experience of Configuration management processes at GFDL.

Appendix 1 provides a summary of the meeting and a link to the workshop and the presentations can be found here:
https://verc.enes.org/ISENES2/events/is-enes2-configuration-management-workshop-09-2013

2. Summary of the Outcome

The workshop met its aims. The community were pleased to see good reports of the FCM build systems from both IPSL (using the deprecated "fcm build") and ECMWF (using the build system of "fcm make"). In particular, the independent take-up of the build system of "fcm make" by ECMWF for the academic versions of the ECMWF IFS and single column models provided others with confidence to trial the build system of "fcm make". The required evaluations were planned (details later) and a summary of the key points of the presentations is provided in the section below.

A common theme across a number of presentations is the interest in the community to migrate from Subversion (which the version control aspect of FCM is based upon) to Git. Although the FCM team has some experience with using Git for their software projects, moving a research community using FCM to a new version control system would require development of new working practices and detail planning. Given the other priorities of the FCM team (to meet the needs of Unified Model partners and to complete development of the Rose environment) this was not likely to be progressed in the short or medium term. There are not the resources in this work-package to further investigate the migration to Git within FCM but the group will continue to share experiences and best practice with Git in future workshops as a number of groups are actively moving to Git.

3. Author’s Analysis

3.1 Best Practice Guidelines

The configuration management workshop provided evidence from the view of a number of different actors or roles from which best proposed practices guidelines have been derived. These guidelines will be further debated and refined at the final workshop IS-ENES2 workshop on configuration management to provide community recommendations. The guidelines are broken down into the needs of these various actors.

3.2 The model or component developer

It is standard working practice for software developers to use a change control tool, such as Subversion or Git, to help manage their development. Such tools provides a convenient way to be efficient, giving access to any committed version of the code integrated with record keeping. The tools are able to support a wide range of working practices, but it is important that a team of developers working on the same code base use a single repository with a common working practice and naming conventions. Working practices need to be clearly defined and suitable education needs to be given. A common approach within an organization will help people who need to work across multiple projects and when they move between groups. Scientists are also developers within the climate modeling community but often come from communities that do not use such tools and so the proposed guidance and training is particularly important within this community.

The developer also needs to apply good change-control discipline as widely as possible to all the code they develop, including personal code for things like data processing.

In summary, the recommendations are to ensure as much code as possible is developed within a configuration management tool using well documented, common working practices. There should be training and support to achieve this recognizing that not all developers have a background in software engineering.

3.3 The system owner

The system owner needs to ensure that code quality is maintained and that control is provided on the content of new releases of a software set. In this community, software is developed by a very wide range of people who sit within a broad management structure. For example, scientists not under the control of the system owner will be developing the code within a model. Tools such as Trac, which are integrated with configuration management tools, can support this process. Key elements of the process that the system owner needs to control include assignment of tasks, design and code review, testing, integration of a change into the wider system and control over what changes are allowed into a given release. All this can be controlled and monitored through a suitably configured ticketing system such as Trac.

3.4 The experiment designer and manager

Doing experiments using climate models is increasingly complex. Experiments need to couple multiple models, often developed by different institutions. Input data needs to be controlled. Experiments are increasingly made up of an ensemble of integrations. Data needs to be post-processed in a systematic way. It is important that suites can be shared and adapted to the needs of a particular experiment whilst maintaining their provenance and that it is possible to know exactly what was run from end-to-end.

A number of institutions have developed frameworks on which to build such suites and these frameworks should be properly version controlled to ensure that the following is clear (i) the provenance of a suite (was it based on a previous experiment) and (ii) how to reproduce the results.

3.5 The consumer of the output from experiments

The user of the output of an experiment needs to know details of the experiment design and configuration in varying levels of detail and they need this information in human readable form. This is the topic of the meta-data activity within this work-package (see below). This requirement contrasts with the core need of the experiment designer who needs to configuration manage the input data and scripts as used by code, rather than humans.

The generation of meta-data for the end user of the data from an experiment is a very time-consuming task. To reduce this overhead, it is important to automate the gathering of meta-data from the framework used to control the experiment design and to present the experiment designer with an interface that is as simple as possible to both design and run the experiment in the repeatable way and to provide the necessary meta-data to the end-user as there is a significant overlap between the two activities.

In summary, the recommended way forward to deliver such a solution from the current developments in the field is to develop and configuration managed framework to support suite design and to ensure that meta-data can be automatically extracted from that framework considering the needs of the end-user of the data as well as the needs of the experiment manager. Although this has not yet been demonstrated within the community, this framework should be integrated with mechanisms that allow the experiment designer and manager to record, in human-readable terms, as much as the meta-data that will be required by the end user within the experiment management framework as possible.

4. Author's Recommendations to the Community

This section of the document provides community recommendations as a result of the first workshop and the analysis of the Author. Such recommendations were not the target of the first workshop, and, such, not indicated to the participants as aim of the workshop. Hence, the recommendations shown here are the sole conclusions of the author and the reviewer of version 2 of the document and not those of the participants.

The author recommends that the community should move to extending the scope of existing configuration management tools to include, as far as possible, (i) experiment design as well as model code; (ii) the complete workflow through to final results and (iii) tools and processes to support the management of the code base to improve code quality by tracking testing and review processes. FCM provides a simplified interface to Subversion but, as partner sites have working change control systems, it is not appropriate to recommend that people with existing solutions should move to FCM, given the cost of such a move and the modest benefit it would bring. FCM also provides a powerful build system which has particular strengths in dealing with Fortran code dependencies and delivering a parallelized make solution. Where, in the wider community, institutions do not have such tools, they are encouraged to evaluate FCM based on the positive experience of many projects at the Met Office and, independently, at ECMWF.

Appendix 1. Summary of the meeting, next steps and action plan

1. Summary

1.2 Opening Remarks

Mick Carter gave the opening remarks reminding the attendees of the aims of the workshop and the task within the ISENES-2 networking activity, NA3, task 2.

1.3 Overview and History

Dave Matthews then gave a presentation outlining FCM – its components, history and use. The components of FCM are:

1. A thin wrapper on top of the code management tool (subversion);
2. A bespoke make system for Fortran codes.

FCM provides a thin wrapper for SVN (Subversion) that simplifies the branching and merging process. Further, SVN can be used in many different ways and it is considered helpful to build a more tailored system on top that encourages a particular pattern of use and implements local policy. A useful feature of FCM is its ability to support the automated, run-time merging of a number of branches. This feature fits well with working practices in the climate community who need to mix-and-match science under development to create new configurations.

A bespoke build system has been developed that automates the dependency analysis required for Fortran 90 (and later) codes. The GNU make tool does not support this and again this meets the specific needs of the climate community who need to support a wide variety of configurations (with or without Ocean and with or without Earth System Components).

Dave Matthews presented graphs showing the increase in take-up and use of FCM since its introduction in 2005 at the Met Office. There are now more than 14 systems and 200 users of FCM at the Met Office. Today, FCM is also used by the large number of Unified Model users around the world (Australia, Korea, India, South Africa, Brazil etc). It is also being used with other codes at IPSL and ECMWF.

During the presentation, attendees shared their experiences using FCM in the Australian UM communities and at NCAS Reading. The Configuration Management (CM) approach used at GFDL was also compared. Useful experiences were shared around branch merging, the scope and requirements of the FCM software and how FCM is able to work with specific compilers.

1.4 The FCM Extract and Build System

Matt Shin presented on the FCM make system. He explained that a review had been done of the available tools and none of the alternatives available at the time had met the aims of the project to:

- Build Fortran source portable to multiple platforms.
- Reliably rebuild the same configuration.

- Support quick *develop-build-test* cycles
- Deliver high performance with parallelization

Some alternatives that were brought to the attention of the group were mkmf[1], scons[2] (looks good but slow) and cmake[3] (a cross-platform open source build system).

Discussion topics from this presentation included:

- Why did the Met Office build FCM instead of using automake?
  - Because automake caters for different use cases and has no dependency analysis suitable for Fortran because of the complexity of Fortran modules
  - Also, FCM better handles the consistency of compiler options.
- Exchanges on the differences between GIT and Subversion for commits and code management. The general advice is that a move to GIT is a move that should be taken with great care because of the changes to a distributed repository model.
- Is it possible for FCM to support the m4 code generator? Further analysis would be needed to reach any conclusion on this.

1.5 FCM and the IFS Model

Two presentations were given on this topic, one by Dave Matthews the other by Glenn Carver.

Dave Matthews described his experience adapting OpenIFS for use with FCM. The sorts of changes required to the OpenIFS code, such as standardizing the use of interface files and removal of repeated source files, were agreed to be code improvements. FCM needed to be extended to be able to cope with the cyclic code dependencies seen in OpenIFS. The performance of FCM was an improvement on the performance of the IFS build system because the IFS build system only exploits parallelism at the level of building libraries.

FCM had the advantage of being able to perform fast incremental and inherited builds and parallel performance showed scalability of the compile up to 8 processes with two different compilers.

Glenn Carver gave some background on ECMWF, the relevant codes and configuration management practices used at ECMWF. The OpenIFS initiative was set to provide the IFS model to the academic community and needed an open source CM and build system that was easy to use.

In evaluating FCM, ECMWF notes the following benefits:

  - FCM has an integrated & reliable dependency analysis
  - FCM is fast & parallel
  - FCM removes inter-project dependency issues

[1] http://www.gfdl.noaa.gov/~vb/mkmf.html

[2] http://www.scons.org/

[3] http://www.cmake.org/

- FCM makes it easy to change compile options per file

And the following issues:

- C++ support is not yet available (*now done: https://github.com/metomi/fcm/issues/47*);
- The log files would benefit from being more visible (*now done: https://github.com/metomi/fcm/issues/46*);
- The boundaries between verbose options are not ideal as –v does not show warning messages and –vv provides too much output;
- Temporary nature of final libXX.a for linking causes problems for tools like CrayPAT which inspect the link step (*now done: https://github.com/metomi/fcm/issues/14*).

As a result of the positive evaluation, ECMWF plan to continue with FCM and extend its use to the full IFS model. ECMWF are interested in GIT integration.

The following points came up in the discussions:

- The importance of specifying compile times in HPC procurements;
- It would be advantageous if the order of compilations could be adjusted to allow the longest to start first (*see https://github.com/metomi/fcm/issues/49*);
- There was a potential for ECMWF to contribute to FCM development.

1.6 The IPSL experiment on the use of FCM build

Arnaud Caubel presented a paper on the experience with FCM build at IPSL. They were looking for a solution that would be able to support:

- Different components, developed by different groups
- Different configurations (one or more components, coupled or forced)
- Different computing centres (different hardware, software, . . .)
- Users with different backgrounds, different computer literacy level

And hence a flexible, portable, robust and easy to use tool. They chose the FCM build component to fit their needs as well as using SVN (directly) and Trac (which is well integrated with SVN). IPSL have found the FCM build system to be robust, easy to install and use as well as efficient.

1.7 The FCM Code Management Tool

In this presentation, Dave Mathews provided details on the working practices of using FCM for code management. Trac integrates well with subversion and provides a useful function for managing the code change process. Trac is loosely coupled to FCM and such issue trackers are recommended as they provide a powerful tool for managing, viewing and tracking individual changes as well as whole code releases. It is recommended that each logical change is developed on its own branch. Code should be tested and reviewed and only submitted onto the trunk in its final state. It is recommended that software is released from the trunk.

FCM commands simplify the command line for the working practices that are encouraged by the FCM team. It also allows revisions to be given more useful keyword names. A standard way of branch-naming encourages consistent use and easier user navigation of the repository. FCM also integrates a better merge facility than that provided by SVN but further improvements (e.g. support for reverse merges) would be welcomed. Xxdiff provides the graphical tool of choice to support merging.

FCM working practices encourage regular commits to the development branch so that a developer can go back to earlier versions. In the short term, there are plans to move to subversion 1.8 to gain benefits in the management of working copies. In the longer term, moving to GIT will be considered especially as this better supports distributed working and merging. However, development of a completely new set of working practices would be required and this will be a significant change for users.

The presentation ended with discussions and views on the working practices that FCM encourages and the method of using the trunk and branches.

1.8 Expectations and requirements in MPI for CM systems

Luis Kornblueh gave a presentation that looked at the requirements of configuration management as seen from the needs of the end-user of the data from climate models. The generation of results requires a complex workflow with:

- A complex, non-standardized tool chain
- various processing steps by various actors
- involvement of multiple infrastructures

This complex workflow needs to be managed in the same rigorous way as scientists manage laboratory experiments. The ultimate aim of CM when considering the output of the process (data) is:

- that the consumer of the data can understand how the data was produced in as much detail as they need to
- that others can reproduce the data

Luis also emphasized the need to be able to manage libraries well in a complex environment, which is another aspect of change control. As Earth System Model (ESM) applications become more complex with components developed by different groups, the complexity of library management grows significantly with different sub-components possibly needing different library versions. Also, the role of databases in the wider context of configurations management should not be forgotten.

It was noted that many of the ideas overlapped with work being done at the Met Office in the Rose project. Rose will deliver a system that puts a much larger part of the experiment design under change control, but does not deal with full provenance of run-time issues such as the machine, the compiler etc.

1.9 Using Git for community development

Git has gained significant ground as the CM tool of choice since the decision to use SVN for FCM. Matt Shin presented on his experience in using Git related to the development of Cylc & Rose. The general consensus that Git alone was not compelling but that Git used with Github did offer some advantages but also came with some dangers.

The main advantages of Git (+ Github) are:

- Merge and rename just works;
- Faster local performance hence better for distributed development;
- More readily available tools written by people who really understand GNU/Linux and Unix;
- Automatic offsite backup via Github: each clone is a full backup of the project;
- Many powerful features, e.g. remote, reset, etc, (but may allow users to mess up);

Others shared the concerns about the power offered by features such as reset.

Matt provided a comparison of features between Subversion + Trac compared with Git + Github and found a good match.

The conclusion of the presentation was that Git worked very well in the context of a small team of distributed developers. However, advantages over SVN were not compelling enough for this to be a priority for the FCM team especially given the size of the task and the user impact during transition.

Some concern was also expressed about the reliability of public sites such as github. Options to mitigate this risk include taking local backups but ECMWF have chosen to implement a solution based on Atlassian Stash to provide the equivalent functionality.

1.10 The Unified Model change control process

Stuart Whitehouse gave a presentation on the governance process used to develop the Unified Model. The process was seen by the group as being very robust:

- All changes are registered and monitored via the Trac system;
- Changes are documented as are test results;
- All work is done on branches and fully documented, reviewed and tested before they are committed to the trunk;
- Reviews include both a science review and a technical review;
- Three level of documentation are expected: in-line code documentation, user documentation and a description of the change. Trac is able to link to all these documents and test reports.
- Test are supported by a test harness that is able to cover a number of model configurations;
- There are daily tests run of any code placed on the trunk to provide integration testing;

The presentation also went through the details of the release cycle for versions of the UM. These UM releases are used at the Met Office and provided as common baseline releases for UM Partners and Collaborators. There are releases 3 or 4 times a year and releases are limited in size to make the integration and testing more manageable.

The Unified Model, because it is used for both Research and real-time operational activities, has more rigorous testing than many research-only codes. A main conclusion of this presentation was the need to build testing and review into a managed process to maintain software quality.

1.11 CM perspective from Princeton University

V. Balaji gave a presentation describing the Earth System Model at GFDL (FMS). This is a very successful and well-engineered code. It has a runtime environment called FRE that includes the change control process. Teams at GFDL/Princeton University have experience of both CVS (older than SVN) and Git (newer than SVN) in the context of climate model development and management. The FRE system has to work in a complex distributed runtime environment and manages the climate process from start to finish including archiving and post processing. Experiment descriptions are in XML. Supported by Curator and the Model Database Interface (MDBI), it was possible to configuration manage whole experiment definitions.

Balaji had the most relevant experience of Git for the climate modelling community and warned that Git implied a new approach to CM. Gits strength was in the early development stage of codes. Balaji emphasised the importance of Github to support Git or the alternative (Gitlab) which can be deployed behind a firewall.

Another conclusion from the presentation was that further discussions on FRE would be useful in the forthcoming IS-ENES2 workshop on workflow solutions.

1.12The Wider Aspects of Configuration Management

Mick Carter led a discussion on the wider aspects of Configuration Management by looking at the various roles that are needed within the climate community. Configuration management in the climate community has a number of challenges that are atypical for standard software development exercised. For example:

- The user of the system is often a developer, changes need to be developed and applied in real-time by people developing the schemes in climate models;
- Users building new configurations need to build up a series of independent changes that cannot be fully tested in isolation and may not be committed to the trunk until they have been scientifically validated;
- It is not only the code that needs to be configuration managed but also the input parameters and files that are used in experiments as well as a definition of the experiments themselves;
- All this needs to be linked in some way to the output data.

Discussions around the need to cater for a distributed set of users (currently using different code repositories) came to the conclusion that improvements were key in this area and would greatly benefit from common code repositories. Also, stable interfaces between software components were seen as an important factor in maintaining increasingly complex Earth System Models. The importance of the data from the runs was also seen as key.

1.13 Rosie: Suite storage & discovery

Dave Matthews and Matt Shin presented on Rosie and Rose. These systems have similar aims to FRE, Curator and MDBI used at GFDL. Both systems maintain complete job descriptions. Rosie uses configuration management to manage suite definitions and a database to support finding these definitions. This presentation again emphasized the increasing overlap between configuration management and the workflow tools that are required to define and manage complex simulations. There was a decision to revisit this topic at the Workflow workshop to be held in Hamburg in 2014.

2. Next Steps and Action Plan

Balaji proposed to go back to GFDL to present on FCM with a view to discuss a possible evaluation with colleagues.

To support MPI's evaluation of FCM within NA3, an existing model (e.g. NEMO) would be provided within FCM with a target date of the end of 2013. The next step will be to evaluate ECHAM with the main focus being on FCM make. The main point of contact for the evaluation at MPI would be Reinhard Budich [Note: work has already started at the time of writing]. The main part of the activity should be completed in January/February with Mick Carter to arrange a teleconference in January.

IPSL are planning to evaluate FCM further by moving from FCM V1 to FCM V2 and to move from FCM-build to the more powerful FCM-make and to report back on the move. The aim is to have moved to FCM2 by June 1014 and will then arrange to discussion the move to FCM-make subsequent to that.

OASIS in FCM was considered another suitable test-bed but was considered lower priority.

ECMWF will continue to develop their use of FCM through the OpenIFS project and other models.

Appendix 2. Workshop Agenda and Attendee list

1. The Work-Shop Scope and Aims

IS-ENES2 Configuration Management Workshop 2013 (NA3)
Met Office, Exeter, UK
Monday 23 and Tuesday 24 September 2013

Scope and Aims

The overall aim of Task 2 of the "Environments" work package is to support a community evaluation (by at least CNRS-IPSL and MPG) of the Open Source FCM system. Also, the task will allow the community to share experiences.

The scope of the FCM system is:

- Change management Trac is used to support the processes and management of code releases.
- Version control through Subversion
- Extract (combining code from multiple repositories and working copies + merging of code from different development branches)
- Build (mainly aimed at building modern Fortran software applications)

A number of sites are planning to review the particular change management and version control tools that they use (e.g. Trac and Subversion) in the next few years and would be interested in learning about any experiences with other tools such as GIT as an alternative.

Hence, the proposed aims of the workshop are to share experiences across change management, configuration management, extract and build and to agree the scope and method of evaluation. (The Met Office has funded resources to help put codes into FCM.)

2. The list of invitees:

V. Balaji (Princeton Uni)
Reinhard Budich (MPI-M)
Paul Burton (ECMWF)
Mick Carter (Met Office)
Glenn Carver (ECMWF)
Arnaud Caubel (IPSL)
Martin Evaldsson (SMHI)
Kerstin Fieg (DKRZ)

Rachel Furner (Met Office)
Rosalyn Hatcher (Reading Uni)

Deike Kleberg (MPI-M)
Luis Kornblueh (MPI-M)
Sonia Labetoulle (IPSL)
Dave Matthews (Met Office)
Domingo Manubens (IC3)
Mike Rezny (Met Office)
Jan Sellmann (MPI-M)
Matt Shin (Met Office)
Stuart Whitehouse (Met Office)
Karl-Hermann Wieners (MPI-M)

3. Programme

a) MONDAY 23 SEPTEMBER 09:00 to 17:15 Room C2-1

Presentations include discussion time

08:30 – 09:00 Registration (Met Office Reception, main entrance)

09:00 – 09:15 Welcome and aims of meeting – Mick Carter, Met Office

09:15 – 09:45 An overview of FCM. History. The components, and how they are used. Facts and figures – Dave Matthews, Met Office

09:45 – 10:30 The FCM make system - code extraction and build – Matt Shin, Met Office

10:30 – 11:00 Tea + coffee

11:00 – 11:30 Experience using FCM make with the IFS model – Dave Matthews, Met Office

11:30 – 12: 00 Configuration management at ECMWF: current issues and plans – Glenn Carver , ECMWF

12:00 – 12: 45 The IPSL experiment on the use of FCM build and objectives in IPSLCMx configurations – Arnauld Caubel , IPSL

12:45 – 13:45 Lunch

13:45 – 14:30 The FCM code management tools - a simplified interface to Subversion – Dave Matthews, Met Office

14:30 – 15:15 Expectations and requirements in MPI for CM systems – Luis Kornblueh, MPI-M

15:15 – 15:45 Tea + coffee

15:45 – 16:30 Using GIT for community development – Matt Shin, Met Office

16:30 – 17:15 The Unified Model change control process – Stuart Whitehouse, Met Office

19:00 – 21:00 Workshop dinner at The Waterfront restaurant, Exeter Quay.

b) TUESDAY 24 SEPTEMBER 09:00 to 13:00 Room C2-1

Presentations include discussion time

09:00 – 09:45 Discussion of the wider aspects of CM and VC (with a short presentation from Mick Carter to introduce the discussion)

09:45 – 10:30 Rosie: Suite storage & discovery – Dave Matthews/Matt Shin, Met Office

10:30 – 11:00 Tea + coffee

11:00 – 11:45 CM perspectives – V. Balaji, Princeton University

11:45 – 12:45 Options for FCM evaluation – group discussion led by Mick Carter, Met Office

12:45 – 13:00 Wrap up and next steps – Mick Carter, Met Office

Optional tour of Met Office IT facilities 14:00 – 14:30