



Call: H2020-SC5-2014-two-stage

Topic: SC5-01-2014

PRIMAVERA

Grant Agreement 641727



**PRocess-based climate sIMulation: AdVances in high resolution modelling and
European climate Risk Assessment**

Deliverable D1.4

Report on strategy for integrating metrics software

Deliverable Title	<i>Report on strategy for integrating metrics software</i>	
Brief Description	<i>Discussion of the strategic options for development of individual metrics and how these might be incorporated into existing packages</i>	
WP number	1	
Lead Beneficiary	<i>Malcolm Roberts, METOFFICE</i>	
Contributors	<i>Thomas Voigt, METOFFICE Javier Vegas-Regidor, Francisco Doblas-Reyes, BSC</i>	
Creation Date	16.01.2017	
Version Number	01	
Version Date	16.01.2017	
Deliverable Due Date	30.11.2016	
Actual Delivery Date	31.01.2017	
Nature of the Deliverable	<input checked="" type="checkbox"/>	<i>R - Report</i>
	<input type="checkbox"/>	<i>P - Prototype</i>
	<input type="checkbox"/>	<i>D - Demonstrator</i>
	<input type="checkbox"/>	<i>O - Other</i>
Dissemination Level/ Audience	<input checked="" type="checkbox"/>	<i>PU - Public</i>
	<input type="checkbox"/>	<i>PP - Restricted to other programme participants, including the Commission services</i>
	<input type="checkbox"/>	<i>RE - Restricted to a group specified by the consortium, including the Commission services</i>
	<input type="checkbox"/>	<i>CO - Confidential, only for members of the consortium, including the Commission services</i>

Version	Date	Modified by	Comments
1.0	16.01.2016	Malcolm Roberts	
1.1	23.01.2016	Malcolm Roberts	Based on comments from Javier 19.01.2016

Table of Contents

1. Executive Summary.....	4
2. Project Objectives.....	4
3. Detailed Report.....	5
3.1 <i>ESMValTool</i>	5
3.2 <i>AutoAssess</i>	9
3.3 <i>Key points</i>	16
3.4 <i>Future strategy</i>	16
4. Lessons Learnt.....	17
5. Links Built.....	17

List of Figures

Fig. 1: Screenshot of AutoAssess in the Rose workflow manager.....	14
Fig. 2: Dependency graph of the AutoAssess processing pipeline, produced with Rose...	14

1. Executive Summary

An aim of many climate research projects is to develop metrics to allow assessment of model fidelity and comparison of different models. Much resource is often spent in this way. However, once projects end, the metrics and code are often lost to the community or at best not maintained.

Here we attempt a strategy for enabling individual metrics to be incorporated into larger metrics packages, which themselves are maintained for the longer term. This requires that the metrics themselves are written in the best way (e.g. using open source software, well-documented) and also that the metrics packages are clear on how additional metrics might be incorporated into their workflow.

2. Project Objectives

With this deliverable, the project has contributed to the achievement of the following objectives (DOA, Part B Section 1.1) WP numbers are in brackets:

No.	Objective	Yes	No
A	To develop a new generation of global high-resolution climate models. (3, 4, 6)		
B	To develop new strategies and tools for evaluating global high-resolution climate models at a process level, and for quantifying the uncertainties in the predictions of regional climate. (1, 2, 5, 9, 10)	X	
C	To provide new high-resolution protocols and flagship simulations for the World Climate Research Programme (WCRP)'s Coupled Model Intercomparison Project (CMIP6) project, to inform the Intergovernmental Panel on Climate Change (IPCC) assessments and in support of emerging Climate Services. (4, 6, 9)		
D	To explore the scientific and technological frontiers of capability in global climate modelling to provide guidance for the development of future generations of prediction systems, global climate and Earth System models (informing post-CMIP6 and beyond). (3, 4)		
E	To advance understanding of past and future, natural and anthropogenic, drivers of variability and changes in European climate, including high impact events, by exploiting new capabilities in high-resolution global climate modelling. (1, 2, 5)	X	
F	To produce new, more robust and trustworthy projections of European climate for the next few decades based on improved global models and advances in process understanding. (2, 3, 5, 6, 10)		
G	To engage with targeted end-user groups in key European economic sectors to strengthen their competitiveness, growth, resilience and ability by exploiting new scientific progress. (10, 11)		

H	To establish cooperation between science and policy actions at European and international level, to support the development of effective climate change policies, optimize public decision making and increase capability to manage climate risks. (5, 8, 10)		
---	---	--	--

3. Detailed Report

One of the key objectives for PRIMAVERA (B above) is to develop new tools for process-based assessment of high-resolution climate model simulations. However, we also want to secure such tools for the future – in many previous projects, much effort and resources have developed tools, but post-project they are often lost to the community. This can happen because they are not maintained, use proprietary software, or are poorly documented.

In order to avoid this happening to PRIMAVERA-developed metrics we propose a strategy that enables individual metrics to be incorporated into larger metrics packages, which themselves are maintained for the longer term. This way PRIMAVERA metrics will benefit the climate modelling community for a much longer time period, while the packages gain extra capability and utility.

This report is split into two sections detailing the status of the two packages considered for this integration, ESMValTool and AutoAssess. These packages were chosen to be further evaluated because they are mature tools at this stage but also have their future secure due to the strong support from each user community. It was also important for their selection that PRIMAVERA will be able to take part in their core developments, allowing us to solve any issues that may appear during the metrics integration.

In the final section, the issues and conclusions are brought together and the strategy explained.

3.1 ESMValTool

Introduction

The Earth System Model eValuation Tool (ESMValTool; Eyring et al. 2016; www.esmvaltool.org) is a community diagnostics and performance metrics tool for the evaluation of Earth System Models (ESMs) that allows for routine comparison of single or multiple models, either against predecessor versions or against observations. The ESMValTool is a community effort open to both users and developers encouraging open exchange of diagnostic source code and evaluation results from the CMIP ensemble. The core development team includes Veronika Eyring (Project PI), Bjoern Broetz, Axel Lauer and Mattia Righi from the Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) and Martin Evaldsson from the Swedish Meteorological and Hydrological Institute (SMHI).

Installation and deployment

ESMValTool does not currently have an automated installation, although its developers are planning to make it available through PyPi or Conda in the future. At the moment, the user has to fill a form at the tool web page, download a tar.gz file and uncompress it at the desired location. This also means that users are responsible for installing all the dependencies themselves.

The ESMValTool main code dependencies are few and well documented, although each diagnostic can also have its own. This means that the installer has to ensure that all of them are also satisfied. This policy provides a lot of flexibility to developers but, if developers choose different tools for the same functionalities, it can unnecessarily complicate the management of the dependencies. To avoid this situation that can be confusing for the users, diagnostic developers are encouraged to 1) stick to the ESMValTool libraries as much as possible and 2) rely only on external dependencies when the internal functions do not provide the required functionality. It seems reasonable to expect that the need for external dependencies will be reduced as the development continues and those libraries improve, simplifying the installation procedure. Also, a user installing it for their personal use can omit dependences for diagnostics that they do not plan to run.

Another important issue to address when deploying ESMValTool is that it can not run multiple instances simultaneously from the same installation. This prevents using the installation to analyse more than one independent experiment, or to design an efficient workflow to speed up the analysis. The developers are aware of this limitation and have already started the work needed to remove it. Unfortunately, this enhancement is not expected to be generally available any time soon. This issue might be an important problem when considering the use of ESMValTool on Jasmin or any other shared platform as each user will have to use a personal copy instead of a unique installation for everyone (the preferred solution). The installer can circumvent this issue by providing an easy way to create, launch and remove a clone of ESMValTool for each run, but this approach adds about 2000 files and 50 MB of used space for each execution, along with the computing time required to manage these clones.

Contributing to the development

The ESMValTool core development team encourages users to contribute as much as possible to the development of the tool and/or the diagnostics. Although the project SVN repository, bug tracker and wiki page are private, the core development team provides access to anybody willing to collaborate. Finally, and just to emphasise their attitude towards collaborative development, a excerpt from the ESMValTool User Guide:

“In addition to using the software, we would therefore like to encourage the community to join the Software Development Team and to contribute additional diagnostics and performance metrics or other software improvements. Contributing back the new diagnostics and performance metrics or other software improvements will help to enhance the capability of the Software, which is of vital importance for securing future funding. You should consider this an obligation if you have taken advantage of the Software, which represents a product of considerable effort by the development team.”

Diagnostics

ESMValTool can run diagnostics coded in NCL, Python or R, but not all these languages have the same support level. The ESMValTool libraries are much more developed in NCL than in Python, while R libraries are well behind both of them. Nevertheless, users are not limited to those libraries and can use whatever third party libraries they need, an aspect that adds a lot of flexibility and potential to the tool.

ESMValTool comes with a set of ready-to-use diagnostics and a set of example namelists to use them. Those diagnostics are a good starting point but present some problems. First of all, small bugs are quite frequent, something usual in the first released version of any tool. Most of those bugs can be easily solved and will be greatly reduced in the future if ESMValTool users grow in number and the collaborative approach suggested by the developers is embraced.

A more worrying issue is that there are large differences in performance and resource consumption between diagnostics. For example, the `tsline` (time series plot) memory consumption is huge, peaking at more than four times the size of the data, while `anncycplot` (annual cycle plot) is consistently much less consuming even if it is a more complex diagnostic. This can be a serious problem when dealing with high resolution experiments, as its requirements are already much bigger than the requirements of the standard resolution ones.

Also, an important concern for some models is that some widely used functions can not deal with irregular grids (i.e. `area_operations`, used for subset, average and sum over lat-lon boxes), forcing the users to interpolate the data before running the diagnostic that require it. There are also some diagnostics that were created for outputs of the NEMO ocean model and will not work for other ocean models.

Finally, diagnostics and example namelists are often not as informative as desirable. For example, the `namelist_eyring13jgr.xml` description states “*This namelist produces figures 1, 2, 4, 6, 7, 10, and 11 of Eyring et al. (2013).*” The name of the diagnostics used by the namelist are in the form `eyring13jgr_figXX.ncl`, so there is not a clear idea of what are they actually offering.

Data management

To ensure that the data is in an acceptable format for the tool and to simplify the diagnostic development, ESMValTool first step consists in preprocessing the data to a common format that will be later used by the diagnostics. A plugin-like system is used for this, which offers each user to develop their personal methods if none of those provided satisfies the needs. Standardized outputs can be used out-of-the-box by using the provided CMIP5 CMOR adaptor.

The main caveat of the ESMValTool data management is the creation of an intermediate file for each variable-time interval combination used, even for CMIP5 CMOR compliant inputs. This not only increases the storage requirements, but it also increases substantially the computing time. For example, a `tsline` diagnostic of a variable from an ORCA1 NEMO output

and a 60-year period creates a 6.4 GB intermediate file. The execution time also increases by more than a factor of 4 when preprocessing is needed (if the intermediate file is present, ESMValTool will not preprocess). Wasting computing time and storage space by converting to a non standard format is not convenient, and gets much worse with high-resolution experiments, whose storage and computation requirements are much higher.

Most of these concerns will disappear if the tool is modified so that the CMIP-compliant CMOR format is used without preprocessing, only converting the data if it is provided in other formats to match the CMIP CMOR conventions. In this case, the preprocessing will not only be useful for ESMValTool but also for other tools that can make use of the standardized data.

Parallelism

ESMValTool does not provide parallelized diagnostics nor an internal workflow manager to run different parts in parallel. The only method to achieve parallelism without modifying the tool is to use a workflow manager to run multiple instances at the same time.

However, the issue with concurrent runs highlighted in the installation and deployment section makes the tool harder to combine with workflow managers. Nevertheless, the clone approach explained above can also be used for this purpose. In this approach the user will have to divide the desired namelists into pieces that can be run in parallel. As those namelists are xml files, creating a program that can automatically do this splitting and generate the workflow is feasible but requires work. It is important to be aware that each namelist will need its own clone, introducing an overhead that can be significant when namelists are split.

Future plans

The future plans for ESMValTool were defined at the second ESMValTool workshop (Munich, 14-16 November 2016). The most crucial decision made there was to rewrite ESMValTool main framework (usually called “backend”) to improve the performance and add new capabilities to the tool. The new backend will be coded in Python and will allow task parallelization, better performance (due to the migration of some critical routines from NCL to Python) and the capability to run multiple ESMValTool instances from one installation. The complete requirements for this rewriting are [open for discussion](#), and they will include most of PRIMAVERA requirements as they are critical for the ESMValTool future due to the need to manage CMIP6’s HighResMIP outputs. The new backend development will start at a coding workshop to be held at MetOffice from 6th to 10th February 2017 with the presence of some PRIMAVERA partners (BSC and the Met Office).

At the same time, it was also decided that the process to merge AutoAssess and ESMValTool shall continue, focusing first in what is called the “scientific merge”: allowing both tools to run diagnostics from each other with ease. A complete merge is envisioned for the future.

From a PRIMAVERA perspective, it is also relevant that ESMValTool is going to be deployed at the ESGF nodes with the intention to allow CMIP6 participants to run diagnostics directly there. There are also plans to integrate ESMValTool with web interfaces like [Birdhouse](http://bird-house.github.io/) (http://bird-house.github.io/) or [Freva](https://freva.met.fu-berlin.de/) (https://freva.met.fu-berlin.de/), allowing

it to be more user-friendly and, in the case of Freva, improving its provenance and results sharing capabilities.

Support and development of the ESMValTool is funded by the CRESCENDO and CMIP6-DICAD projects; and also by Copernicus calls such as MAGIC, so there is no doubts about its future maintenance.

3.2 AutoAssess

3.2.1 Existing AutoAssess

AutoAssess was started around 2008 as a collection of assessment routines to facilitate the process of routine model output comparisons during global model development in the Met Office. It provided a single command to run this set of standard assessments in a standardised way on a desktop machine. It was mainly written in IDL.

When the Met Office decided to move from IDL to Python as its main data analysis language translation of existing IDL routines started. For the following areas a Python version exists:

- Conservation
- African Easterly Waves
- Global Troposphere
- Hydrocycle
- Land surface
- Madden-Julian Oscillation
- Monsoon
- Stratosphere
- Mid-latitudes (15LOC Fortran)

Currently in use are only three Python assessment areas:

Python

- African Easterly Waves
- Land surface
- Stratosphere

IDL:

- Aerosols (no Python version)
- Conservation
- ENSO
- Hydrocycle
- Mid-latitudes
- Madden-Julian Oscillation
- Monsoon
- Radiation

Each assessment area can have two types of functions:

a.) Assessment of a single model run

b.) Assessment/Comparison of two model runs. (These functions sometime require the single run functions to be run before.)

Data Retrieval

AutoAssess retrieves the model output in the PP format from MASS (Met Office data archive) using the MASS retrieval client Moose. Therefore each area provides query files for all required data, one for each of the output streams of the Unified Model (example below). The query files contain the output variable code (STASH), but can also contain other PP header data (similar to CF meta data but different syntax; mostly encoded as numbers). For example one could specify to only retrieve data on a certain pressure level.

```
conservation.seasonal.s.txt:
```

```
begin
```

```
    stash=(1201,1207,1208,2201,3217,3332,5215,5216,30419)
```

```
    lbtim=(121,122)
```

```
    mon=(3,6,9,12)
```

```
end
```

```
moo select /foo/bar/conservation.seasonal.s.txt moose:/crum/u-ab261/ap.s.pp /foo/bar/data
```

The filename follows AutoAssess convention [areaname].[meaning period].[ap stream letter].txt. The meaning of the meta data keywords is explained in the UM documentation¹.

Central Data Storage

Model Output Caching

The retrieved model output for a suite ID is stored at a central location (/project/hadgem3/data). Model output data will be retrieved only once and then reused by later runs using the same data. This caching on disk is to avoid slow retrieval from the MASS archive, something often possible from tape storage.

The Met Office Unified Model produces its output in the proprietary PP-format (Post Processing). These files contain binary sequential data, which makes bulk reading and writing fast. However, data extraction becomes very slow as the entire file has to be read to discover its content. A workaround is the use of stashsplit: PP-files are split into a directory tree containing single fields.

However, this caching approach is only feasible with current low model resolutions. The MASS archive has now a large disk cache (several 100 TB), and parallel retrieval from disk is reasonably fast.

A stashsplit directory tree contains around one million files. Even a simple ls command may take several minutes to display a file list.

Climatologies and Observations

Climatologies and Observations used in the AutoAssess assessment areas are stored centrally in /project/cma/clim.

Ancillary data

Ancillary data (such as orography, land-sea-masks) used in the AutoAssess assessment areas are stored centrally in /project/cma/ancil.

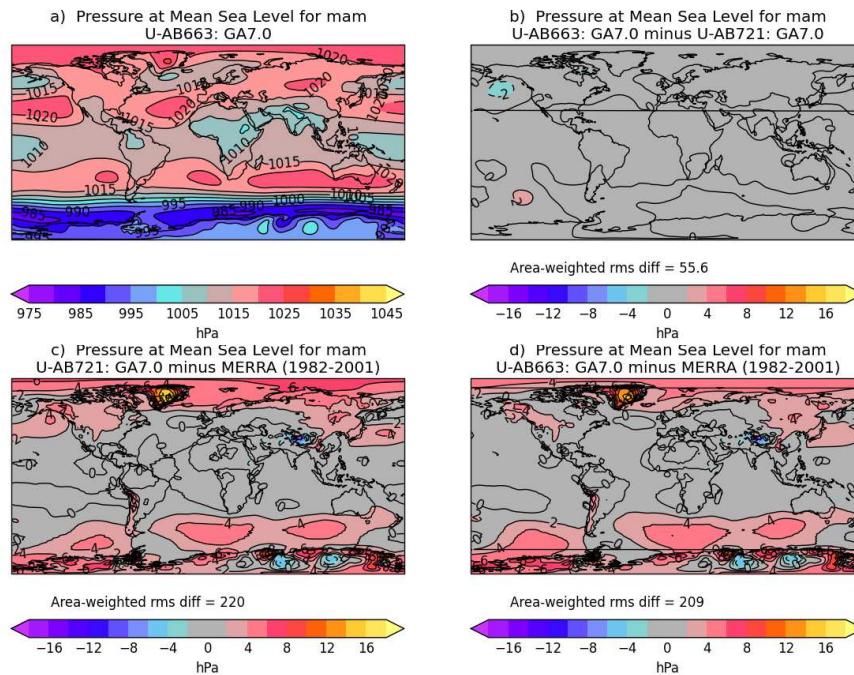
¹ https://code.metoffice.gov.uk/doc/um/latest/papers/umdp_F03.pdf

Metrics and Plots

The output of the AutoAssess assessment areas is stored in /project/gmed/auto_assess2. The symbolic link ~hadvg/public_html/auto_assess so that the results can be viewed in a browser in the intranet.

Validation Note

Validation note is a standalone tool that has been incorporated into AutoAssess. It produces many 2-by-2 plots (“validation notes”) such as the one below, compare two model runs with each other and with observations/reanalyses.



Supermeans

The Validation Note tool produces periodically averaged fields, such as the winter average temperature over 20 years. Periodically averaged seasonal and monthly averaged fields are called “Supermeans”. Also annually averaged fields averaged over multiple years are called “Supermeans”.

Supermeans are a common way to inspect model output. They are not produced by the model itself. Validation Note produces them using the PP-file specific tool ppmnvar and adds them to the model output in the MASS archive.

Several assessment areas use this data and it is therefore required to run Validation Note (as part of AutoAssess) before the assessment areas.

3.2.2 New AutoAssess

Aim

The aim is to adapt AutoAssess to work outside the Met Office on JASMIN with netCDF (CMOR) high resolution data as part of the PRIMAVERA project.

Issues

The previous version of AutoAssess and the assessment areas are bespoke to Met Office infrastructure, model output, and the PP file format.

AutoAssess

- Can only get PP data from MASS
- Retrieval is slow (serial), uses local disk cache to store all retrieved data
- Splits PP files into 1 million files per run to speed up access
- No tests
- Almost no documentation
- Not a package
 - There is no installation routine, and it assumes all dependencies to be installed. The versions of dependencies are also not checked.

Assessment Areas

- Assume PP data (when using AutoAssess API)
- Some assume 360 day years
- Central observations/climatologies repository
- Require “supermeans” produced by *Validation Note*
- Almost no documentation
- No tests

Strategy for new version of AutoAssess

- Create a package that can be automatically installed with user rights, and isolated from the system’s Python.
- Processing Pipeline
 - Clear separation of tasks
 - One executable for each task
 - Workflow controlled using workflow manager
 - Configuration using workflow manager GUI

This approach facilitates testing, as each individual component can be tested in isolation. The separation of tasks will also allow the use of a compute cluster, and its efficient usage. A compute cluster requires the specification of the compute resource limits (time, number of CPUs, and memory). Surpassing any of these thresholds will stop the program. With separate task the resource limits can be set closer to the actual usage, rather than the maximal usage of the program.

This approach will also allow for easier adaptation or exchange of components. This is important as the use for PRIMAVERA, which uses the CMOR (Climate Model Output Rewriter) file structure, will require a different data retrieval method than the Met Office internal use.

As each task is done by a separate executable, the AutoAssess pipeline can also be run outside a workflow manager, for example when a new assessment area is being developed.

The Met Office model development research and the PRIMAVERA project use different file formats, and meta data standards, which describe the actual data. To circumvent this

problem the original data is read with the Met Office Python library Iris², the Iris data structure (Iris cube) is then serialised, and written to disk. The assessment areas read these serialised Iris cubes instead of a particular file type.

This reorganisation of AutoAssess should also be used to add tests. The rewritten pipeline components are tested with unit tests. Regression tests around the assessment areas, using 2-3 years of low resolution data will be used for verification purposes. This will help to adapt the assessment areas to new requirements without changing the original results.

AutoAssess provides two function calls to the existing assessment areas to access the Stashsplit data (load_run_ss) and the Supermeans data (load_run_super). In both function calls the data has to be specified using PP meta-data. There will be two steps to change to CF meta-data. First, the data selection process will be changed to work with the serialised Iris cubes instead of the stashsplit directories, without changing the call syntax. In a second step the call syntax will be changed to only use CF-conform meta data.

New assessment areas should not use these function calls, instead they should read the serialised cubes directly.

*Progress so far*³

- AutoAssess is now a package that can be installed into a virtual environment, or a conda environment.
- Rewrite AutoAssess as a Rose suite⁴ with separate components
 - o Data retrieval from MASS archive
 - o Data indexing, and *Iris cube* creation as file type agnostic data format
 - o Assessment area wrapper for existing assessment areas
 - Replacement of load_run_ss AutoAssess API component to select data from Iris cubes, rather than the stashsplit directory tree
 - This process was started with the assessment areas Monsoon and Stratosphere, so far these assessment areas are the only ones working in the new version of AutoAssess
 - o Create Normalised Assessment Criteria Plot (NAC plot) for each area
 - o Create HTML page for assessment areas
 - o Create overview page for AutoAssess run
- Regression Testing functionality
 - o Run added assessment areas with 2 or 3 years of data and automatically, and compare results to previously recorded reference results. (Currently only Monsoon and Stratosphere area)

² <http://scitools.org.uk/iris/docs/latest/index.html>

³ Source code: <https://code.metoffice.gov.uk/trac/cma/browser/autoassess/trunk>

⁴ <https://github.com/metomi/rose>

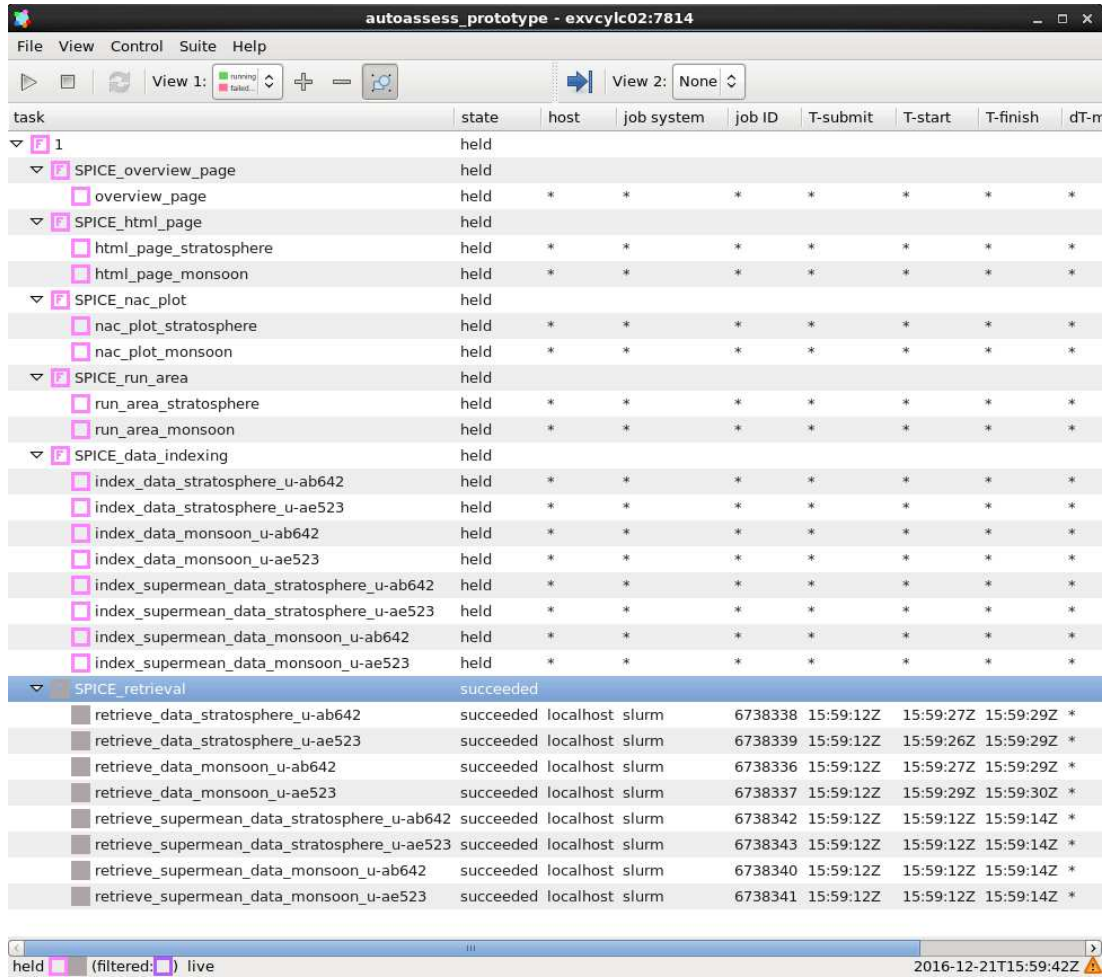


Figure 1: Screenshot of AutoAssess in the Rose workflow manager.



Figure 2: Dependency graph of the AutoAssess processing pipeline, produced with Rose.

Open tasks

- Replacement of load_run_super AutoAssess API component to provide “Supermeans”. This entails the retrieval of model output data and calculation of Supermeans using the Iris library. This will be done by adding separate retrieval and meaning tasks.
- Change the AutoAssess API, and all instances in the assessment areas where non-CF-compliant meta data, such as STASH codes are used to CF-compliant meta data
- Add the remaining areas to AutoAssess
- Add regression testing functionality for remaining assessment areas
- The remaining assessment areas are: Conservation, African Easterly Waves, Global Troposphere, Hydrocycle, Land surface, Madden-Julian Oscillation, Mid-latitudes
- Create functionality to retrieval data from CMOR file structure on JASMIN, and add it to Rose suite
- Tests with high-resolution data.

Advice for new areas

The work on the new version of AutoAssess has highlighted a few issues that should be avoided in the development of new assessment areas.

Clear documented intent of the assessment

- What is the purpose of the assessment? How is it done, i.e. which fields, what kind of analysis (no implementation details)?
- The information should be enough recreate the assessment capability without having access to the code.
- This documentation should be part of the code.

Code Review and code handover

Assessment areas are often funded by specific projects, and written by domain experts. Assessment areas should be written such that non-expert in this particular climate domain can easily read the code and understand the intentions of the original author, and modify or optimise the code if necessary.

This means meaningful name for variables, functions, etc. , concise comments, modular code reflecting the task hierarchy of the assessment code (avoid large classes/functions).

The code should also be written with tests. Testing is crucial to create and maintain reliable software.

Therefore, newly developed assessment code should be regularly reviewed from early on.

Use CF metadata or (descriptive placeholders) to access data

- CF standard meta data should be used to select and modify data
- Non-CF compliant assessment should be kept separate, for example model specific output variables.

Calendar

Assessment areas should be independent from any calendar used, and no particular calendar should be assumed other than the Gregorian calendar. Iris library abstractions can be used to refer to climatologic years, seasons and months. The same applies to unit conversions that include time periods, for example mm/month -> mm/day.

Diagnostic assessment, i.e. assessment that does not compare model output to observations, should be independent of the assessment period.

3.3 Key points

ESMValTool is software with a lot of potential but also with some problems at this stage of its development. The inability to run multiple instances simultaneously makes the deployment on facilities with multiple users like CEDA-JASMIN difficult, and it is even more difficult to use it with a workflow manager to achieve parallelism. The lack of performance and/or huge resource consumption of some of the diagnostics can also be a problem and becomes even more relevant when dealing with high-resolution experiments. The use of custom intermediate files for the data management is also a relevant issue due to the extra storage and computing resources that requires, another problem that also becomes worse with high-resolution experiments.

Most of the previous concerns should be addressed when the new backend is implemented. PRIMAVERA has the opportunity to join its development from the beginning to ensure that is in line with its needs and, at the same time, to benefit from the strong support coming from other projects.

Finally, one of the main objectives to achieve by the PRIMAVERA metrics sharing strategy is to reach a community as wide as possible. The future deployment of ESMValTool at the ESGF nodes for CMIP6 is a valuable opportunity to reach the global climate community. This will allow us to maximize the impact and visibility of PRIMAVERA activities and will also benefit each individual partner.

AutoAssess also has great potential for giving insight into model simulations, and complements ESMValTool in focussing more on comparing (typically two) models against each other and against observational/reanalysis datasets. So far AutoAssess has been a tool used only by Met Office scientists and collaborators, and as such requires more development work to be useable in a more generalised way for different models and data formats.

As noted previously, the process to merge AutoAssess and ESMValTool will continue, focusing first in what is called the “scientific merge”: allowing both tools to run diagnostics from each other with ease. A complete merge is envisioned for the future.

3.4 Future strategy

Based on the above discussion, the initial strategy will be to integrate the metrics within ESMValTool as resources allow. This is mostly due to its proven capability to work with

standardized data and in different environments and its future installation into the ESGF nodes for CMIP6.

The future plan for a 'scientific merge' will ensure that both communities will get to benefit from PRIMAVERA metrics. In addition, the different perspective offered by AutoAssess will continue to be examined, and once this has the capability to work with more standardized data, further development will be discussed.

The recommendations contained here for best development advise for the individual metrics will hopefully be absorbed by the project developers, so that incorporation into the packages can be as straightforward as possible.

Delay in delivery

This deliverable was due in M12 of the project (Nov 2016). It was delayed to enable the main developers to attend the ESMValTool workshop (Nov 2016), and allow time for further discussion of the implications at the second PRIMAVERA General Assembly in November/December 2016.

4. Lessons Learnt

A key lesson learnt from previous projects is that it is too easy for code to be lost once the project comes to an end. The strategy presented here is our outline of how we aim to prevent this from happening in this project. This is a 2-way process – encouraging the developers of metric code to produce it in a form that may be amenable for incorporation into packages, and developing the packages to make it as easy as possible to incorporate new metrics.

The previous development of both ESMValTool and AutoAssess tended to focus on typical lower resolution model simulations. With the advance in horizontal resolution, more careful consideration of workflow, memory management and data flow is needed for efficient and effective use of such packages. This is one of the foci of PRIMAVERA, and hence it is important for the package owners to take an active part in the project in order for these needs to be taken into account.

5. Links Built

The information here should help guide technical development of metrics within WP1, and also advise on the development of closely associated assessment code in WP2. As indicated, it will also position PRIMAVERA at the heart of CMIP6 by contributing to the provision of the ESMValTool code to be implemented on CMIP6 ESGF nodes. The collaboration with the ESMValTool community will also establish links with other projects like CRESCENDO, CMIP6-DICAD and C3S-MAGIC. It will likely be a key element of the planned meeting between H2020 and FP7 projects in early summer 2017 (including PRIMAVERA, CRESCENDO, Blue-Action, APPLICATE).

References

Eyring, V., Righi, M., Lauer, A., Ewaldsson, M., Wenzel, S., Jones, C., Anav, A., Andrews, O., Cionni, I., Davin, E. L., Deser, C., Ehbrecht, C., Friedlingstein, P., Gleckler, P., Gottschaldt, K.-D., Hagemann, S., Juckes, M., Kindermann, S., Krasting, J., Kunert, D., Levine, R., Loew, A., Mäkelä, J., Martin, G., Mason, E., Phillips, A. S., Read, S., Rio, C., Roehrig, R., Senftleben, D., Sterl, A., van Uft, L. H., Walton, J., Wang, S., and Williams, K. D.: ESMValTool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of Earth system models in CMIP, *Geosci. Model Dev.*, 9, 1747-1802, doi:10.5194/gmd-9-1747-2016, 2016.