

Interactive comment on “The Oceanographic Multipurpose Software Environment” by Inti Pelupessy et al.

Anonymous Referee #1

Received and published: 14 October 2016

General comment

This paper presents the Oceanographic Multipurpose Software Environment (OMUSE), its design, model and support components, and illustrates its use by presenting few diverse applications. OMUSE is a python scripting framework that allows running numerical experiments involving different, possible coupled, ocean models and different tools for on-line analysis or post-processing of these ocean model results, on possibly heterogeneous computing resources. This paper is clear and well written and the discussion part includes a comparison with other existing frameworks and some reflections about the current limitations of the system. As such, I consider it deserves publication after considering the minor revisions proposed here.

Specific comments

[Printer-friendly version](#)

[Discussion paper](#)



- Title: Please provide more details about OMUSE in the title (not just its name) so to better reflect the content of the paper. Also, it is recommended to provide in the title a version number of the latest OMUSE version available (see also A. Kerweg's comments).
- L15: I am not sure the application examples described show the efficiency of OMUSE; I would insist more on the flexibility but less on the efficiency.
- L63: You write "This has the benefit of the parallelism and . . .". This seems incoherent to me with what you write in the discussion L716-L717: "limitation of the current design of OMUSE is the fact that the communication between solvers is handled by the master script. This imposes a bottleneck for the performance of the communication between e.g. two parallel codes." Also, L105: What is the benefit of providing built-in parallelism in the MPI-based remote function protocol if, as written in the discussion the handling of the communication through the master script imposes a bottleneck? Can you clarify?
- L64 and other places: You mention the "bookkeeping inherent to code coupling". Can you explain what you mean by bookkeeping? In particular, what do you mean precisely L441-442 by "extensive automation of bookkeeping operations"?
- L116: What do you mean by "the sockets channel is mainly useful for cases were a component process is to be run on one machine". Of course, a process is always run on a machine! Do you mean "the sockets channel is mainly useful for cases where a component process is to be run a different machine".
- L126: you write: "the communication requirements between processes must not be too demanding. Where this is not the case (e.g. when a strong algorithmic coupling is necessary) a different approach may be more appropriate." Why should the (MPI) communication be "not too demanding"? What do you mean by "a different approach"? Do you mean something else that OMUSE?
- L190-191: I am not sure I understand the difference between Cartesian and Regular.

[Printer-friendly version](#)[Discussion paper](#)

What do you mean by “constant”? constant in time (I suppose not)? Constant with respect to the dimension? Please clarify.

- L241: can you clarify what you mean by “This functionality is preferably not used within OMUSE.” Does OMUSE support code using their own I/O library or not? The word “preferably” is ambiguous.

- L385-388: What is the relation between these two sentences (starting with “In case of stationary . . .”) and the restrictions on the forcings discussed in this paragraph?

- L401-402: SCRIP library is included in OASIS but it is not in MCT, even if MCT can use grid remapping weights and addresses generated (separately) by SCRIP.

- L422-423: the source grid has to be structured not because of the way SCRIP computes the area integrals but for the calculation of the gradients needed for the 2nd order.

- L486: Section 4.1.2 is about non-overlapping domains. Why is it named “Domain decomposition”? Also, are you talking here really about “non-overlapping” domains or more precisely “partially-overlapping” domains, as in the 5.2 example?

- L507: What do you mean by “preceding examples”? Are these the 4.1.1 and 4.1.2 couplings? If so, maybe put the numbers 4.1.1 and 4.1.2 for clarity.

Minor comments:

- L14: Remove the , after “solver”

- L23: I would not write “current CMIP5” as CMIP5 is over now and CMIP6 is on-going.

- L123: maybe replace “the requested subroutine calls” by “the requested simulation code subroutine calls” (if I got this right)?

- L144, change “OMUSE implements a” for “OMUSE implements also a”

- L509, add a) after viewpoint.

Printer-friendly version

Discussion paper



Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-178, 2016.

GMDD

Interactive
comment

Printer-friendly version

Discussion paper

