Geoscientific
Model Development
Discussions

Open Access

EGU

Interactive comment on "Rapid development of fast and flexible environmental models: The Mobius framework v1.0" by Magnus Dahler Norling et al.

Magnus Dahler Norling et al.

magnus.dahler.norling@gmail.com

Received and published: 19 November 2020

This paper presents a framework to support the development of environmental models, referred to as the Mobius framework (v1.0). The aim is to allow scientists with potentially limited programming skills to develop component models within the framework, which can then be linked together. This is an important and timely contribution as new approaches to modelling are urgently required particularly as we head towards the need for integration of models. The framework is developed for hydrology and water quality analyses but should be applicable to other settings. The framework is also available as an open source tool with a link

provided in the paper to the relevant github This is a well-written and accessible paper. My own major concern is how it is framed. It is written very much as a description of the approach rather than as a research paper. To be a fully-fledged research paper it would have: research questions and/ or a guiding hypothesis, consideration of the state of the art and gap identified, methodology and evaluation/reflections/discussion. These elements are largely missing. The paper would be much stronger being re-framed as more of a research contribution. I pick up on these points in my more specific comments below.

We thank the reviewer for taking their time to do the review! The main criticism by the reviewer is that the paper is not framed as a research paper. Please note that the paper was submitted as a 'Model description paper', which is a form outlined by GMD, and allows for descriptions of models and modelling frameworks. See https://www.geoscientific-model-development.net/about/manuscript_types.html Substantially evaluating if using a (or this) framework vs not using one is productive would make this into a different kind of paper, but that was not our intended scope.

We will address some of the specific comments:

The second section provides an overview of Mobius. I found it quite hard to get to the crux of the design, and it is quite short and lacks any real depth. I am a computer scientist by training and I wanted to see things like an overarching software architecture and also an explanation of key design decisions with rationale. This is missing from me. It would be very hard fro example for other researchers to consider the text here and get anywhere near reproducing 'the approach'.

We can go into some more detail in the description of the software implementation details. Note that the form outlined by GMD says that implementation details should be included if they will substantially affect the output results. For models this is straightforward, but for frameworks what this implies is a little more up to interpretation. Note

also that the documentation included in the repository (and the source code itself) is an important part of the supplementary material to the paper, which does go into more detail. See especially https://github.com/NIVANorge/Mobius/blob/master/Documentation/ model_builder_documentation.pdf for a detailed description of how one uses the framework to build models.

Section 3 is then a 'demonstration of Mobius' and this title says a lot about the way the paper is framed. To me, it should not be a demonstration of a given approach but rather should be an evaluation of how well the approach achieves its goals, with the evaluation being rigorous and scientific. Instead, it steps through the GUI (but curiously not in a visual way) and also the use of Python wrappers (the key to interoperability in their approach), but not in a way that allows the reader to see beyond the "what" to the "why" this is done (and other alternatives that could have been considered). This section also contains a case study – but again its stated purpose is to demonstrate not to evaluate. It is also quite a small example and it is not clear how this would scale up to something more substantial. The section concludes with some benchmark figures, which are interesting, but it is not clear why performance is measured and nothing else is evaluated, when performance is not mentioned as a goal. Section 4 contains a discussion but to me this is way to narrow and specific and lacks a true element of reflection, e.g. what has worked, what has not worked, what are the strengths of the approach and weaknesses, and so on.

There is a screen shot of the GUI, which shows a substantial part of its functionality. We can see if there is more we can do to make it more clear how it is used. (note that in the submission form of the manuscript all figures are at the end of the document. Presumably they will be inserted at appropriate places during editing). The GUI is not the main focus of the paper, but is described because of how it aids in making the framework easy to use both during development and for external users of the models.

The purpose of the demonstration section is to show that the framework allows you to

build, modify and evaluate models, with the goals that are stated earlier in the paper, among others:

1. it is very little work to modify models in order to evaluate different model structures.

2. It is easy to reuse existing modules (in this case a hydrology module)

3. Speed of model runs (performance) (this is definitely one of the important stated goals, see e.g. the paragraph around line 60)

4. Models can easily be run many times such as in auto-optimization and sensitivity analysis, which are important tools for evaluating model success. The goal of this is stated on line 60, and ties in to why it is important that the models run fast. We show that having auto-optimization and sensitiviy algoritms that can be used with any model built in the framework is useful because it allows you to very quickly apply them to any newly created model structures with little extra effort.

To see if the claims hold one can also inspect the source code used in the examples, which is given as a part of the supplementary material. Putting all the source code in the main paper would in our opinion be excessive.

We could make it more clear in the intro to section 3.2 that performance is a goal also for that demonstration since it allows for quick feedback on model success.

When it comes to scaling this up to larger models, examples of this are in the repository. We can try to give a better description of how this is done. It is for the most part about just having more equations, but the approach is the same as in a small model.

There is also so much more could be done in such a framework and these dimensions are not considered, e.g. running the model multiple times, perturbing parameters to carry out sensitivity analyses, running ensembles of models, looking at model coupling in a more sophisticated way, and so on. Finally, there is

a lack of consideration of related work and yet there are a significant number of other frameworks in existence with similar goals.

Running the model multiple times and sensitivity analysis is discussed around lines 60, 140, 175-195 (and demonstrated with source code in the Jupyter notebook in the link at line 199). Then one line 280-325 we do this for the specific study case. Note that MCMC is a form of sensitivity analysis that involves perturbation of parameters.

We also describe the API you can use to communicate with the model (exposed both in C++ and python and in principle any language with a C foreign function interface) (lines 136-140, and after 170). For instance we say that you can set new parameter values in the model, call the model run function, then read out the result values. It should be clear that this can be used for quickly scripting many kinds of sensitivity analysis setups (that are reusable across all Mobius models), or do other kinds of multiple model runs, set up ensemble runs, and to couple with external models.

We can make it more explicit in the final manuscript what is the possibilities and limitations of this API.

For instance you can also read and write input time series eg. in order to automate the setup of multiple climate scenarios (without being limited to using our specific file format).You can not have coupling with external models where there is a need for a per-timestep communication back and forth between the models (though this would not be that difficult to add if it is needed later) unless both models are created in Mobius. If you want to couple with an external model currently you need to take the entire model output of the other model as an input to the Mobius model (or the other way around). Apart from that, are there any other substantial forms of model coupling that should be addressed?

Other modelling frameworks are discussed in the paragraph starting at line 65. The point we make that we think separates our framework from others is that our models have a higher degree of customizability, while also not requiring the user to do much

explicit programming except for some very easy types of syntax. This may be clearer when we revise the description of the framework.