Geosci. Model Dev. Discuss., 8, C61–C64, 2015
www.geosci-model-dev-discuss.net/8/C61/2015/

Geoscientific Model Development Discussions
Open Access

Interactive comment on "The software architecture of climate models: a graphical comparison of CMIP5 and EMICAR5 configurations" *by* K. Alexander and S. M. Easterbrook

Anonymous Referee #2

Received and published: 21 February 2015

Summary:

The authors analyze the source code of several climate models and determine their structure, i.e. how components are separated and interact from a software point of view. They visualize these structures in simple diagrams. They argue that these diagrams offer insights into the similarities and differences between models.

Recommendation:

This is an interesting and novel idea to look at code structures, a well written manuscript, and I recommend publication with minor changes in the discussion and implication sections. I'm still unsure whether we can learn anything about such analy-

sis from a climate point of view (see details below), and how much this analysis affects how people think or work on models, and the authors have not demonstrated that. The discussion should be sharpened here. But the question of how such models are built, documented, tuned, how information is shared between modeling groups, and therefore how ensembles of models should be interpreted, is a very important one. Few people have looked at those broader questions, and it is great to see contributions from the software engineering community in this field. I am sure this manuscript will stimulate discussions, and in the long run help people to understand both how models are built, and how to interpret the zoo of models out there.

Specific comments:

1) Abstract, page 352, line 11: "These diagrams offer insights into the similarities and differences between models, and have the potential to be useful tools for communication between scientists, scientific institutions, and the public.": I would argue the diagrams offer insight in the software structure, but not in the behaviour of the model. Two models can have similar structures but behave very differently, in fact even the same code with perturbed parameters can. In the end all of those models describe the same physical system, and whether a sea ice model is a subcomponent of the ocean or a separate component called from a coupler should not make any difference to the simulation of the climate, if things are done properly. The results from a model should be determined by the set of equations used (plus some parameterizations and numerical assumptions of course), and should be largely independent from the software framework, or the language used. The implication of this view is that such visual representations may be a useful communication tool for those who build and use climate models, but not necessarily for the public, and apart from the size of the components we cannot infer much (if anything) about how the model will behave, or how similar it will be to reality, or to other models. I do think the information provided here is valuable, but it is limited to software, and largely irrelevant for climate. It would be appropriate to state that clearly I think, unless the authors convince me that there is more value

that I have missed so far. In this case it would be good to see some specific examples and applications where such representations have been used, or have influenced something, or where people could infer climatic behaviour from the structure of the software.

2) Page 353, line 1: "We argue that these differences in module size offer a reasonable proxy for scientific complexity of each component": It would be useful to define complexity early here (it comes in a parenthesis statement on page 361 but not clearly). The statement suggests that by complexity the authors mean "amount of stuff", i.e., number of processes, or degree of detail considered. One could also interpret complexity by how rich the results are in terms of behavior. The equations of motion or a Lorentz system could be described as complex in terms of their behaviour, even though they don't need many lines of code. Increasing the resolution of a model could massively increase the complexity in terms of the simulated patterns or structures of clouds, fronts, storms, while keeping the exact same code.

3) Page 361, line 25: the land component is not just about carbon cycle, but also about land surface processes, vegetation, and hydrology.

4) Page 364, line 16: I'm not sure the interpretation of this factor of 20 is appropriate and useful. One might compare lines of code between different models of the same class (e.g. AOGCMs) but in a hierarchy of models the lines of code will obviously differ. An energy balance model can be coded with just handful of lines, and is also a climate model in some sense, but of course it has a very different purpose.

5) Page 364, line 17ff: I like the comparison here but would add a few things. The amount of expertise may determine the amount of development of individual code, but does not imply similarity or differences to other codes a priori. In practice it may be the case, but it's a hypothesis (which is hard to test, and I'm not asking for a test). Second, it would be good to emphasize that these two approaches take very different but complementary approaches. One looks at the structure of the code, how the model

is built, and the other looks only at the model output, without knowing anything about the code. Finally, there is an update on the model clustering for CMIP5 by Knutti et al., GRL 2013, doi:10.1002/grl.50256.

Interactive comment on Geosci. Model Dev. Discuss., 8, 351, 2015.