Geosci. Model Dev. Discuss., 7, C1066–C1072, 2014
www.geosci-model-dev-discuss.net/7/C1066/2014/

Geoscientific Model Development Discussions

Interactive comment on "C-Coupler1: a Chinese community coupler for Earth System Modelling" *by* L. Liu et al.

Anonymous Referee #2

Received and published: 8 July 2014

General comments

This paper discusses the C-Coupler1 coupling software including its goals, design, architecture, and an evaluation of the software. The paper describes a significant software development undertaking and the evaluation section indicates that C-Coupler1 can be used in realistic scientific settings. That being said, the scientific significance of the paper could be greatly improved by better placing the C-Coupler1 in context with the large number of existing coupling technologies. Specifically, it should be clear why existing technologies do not meet new coupling requirements and how C-Coupler1 pushes forward the state of the art. When compared with existing coupler designs, the unique aspects of C-Coupler1 should be emphasized, especially when novel choices have been made that offer new insights into how couplers should be designed in light

of new scientific requirements. I have rated the paper fair with respect to scientific significance although I expect that this can be improved in a careful revision of the paper.

Based on my understanding of existing coupling technologies, the feature that seem most unique to the C-Coupler1 is the use of a procedure registration system and configuration metadata to support a wide number of coupled model configurations while maintaining the same codebase for component models. In other words, the C-Coupler1 as described seems to be among the most dynamically configurable coupler and the implications and tradeoffs of this approach should be discussed in the paper. A related design aspect is the use of generation technology to automate some of the configuration. As discussed in the specific comments section, the use of the configuration files should be clarified and the new level of flexibility and reduction in maintenance that this approach affords.

The scientific quality of the paper is good and the methods in the evaluation section can adequately show how the C-Coupler1 meets the identified software requirements.

The scientific reproducibility I have rated fair. I think this aspect of the paper could be improved in a revision where the overall workflow of C-Coupler1 is described including an explanation of the interaction of the configuration files, API and, runtime system. For example, a scenario could be described in which a model developer takes an existing model code and prepares it for use in the C-Coupler platform, including all phases. What steps are required?

I have rated the paper fair for presentation quality, primarily due to a number of grammatical errors which can be easily corrected in a revision. As it stands, the paper communicates the main ideas well, but readability can be improved.

Specific comments

Section 1 identifies four key requirements for coupling future ESMs, including 3-D cou-

pling, high-level sharing, common model software platform, and better parallel performance. 3-D coupling and better parallel performance are adequately explained and are well known in the community. The meaning of “high-level sharing” is not immediately obvious suggesting the need for a more descriptive term. It has to do with the fact that the same component models (e.g., atmosphere code) are used in many different configurations, e.g., coupled to different components or run in standalone mode. I would argue that this is not a new requirement, although it is certainly an ongoing requirement. There is minimal discussion about how this requirement is currently handled. OASIS, for example, supports different model coupling configurations without necessarily requiring code changes to the component models. ESMF does as well, although code changes could be required in the driver. Additional discussion of what level of changes are required would also help to clarify the issue, e.g., some model codes could have compiler directives (#ifdef) for different configurations (requiring a recompile) while others could read a file at runtime and be configured dynamically.

The term “coupled model version” is used throughout the manuscript and could be confused with software release versions (e.g., CAM4 vs CAM5). The term “coupled model configuration” may be clearerâĂŤi.e., the same version of a model code is used in multiple configurations.

The definition of “common model software platform” is unclear in this section. The idea of running multiple coupled models in the same environment is understood conceptually as a way to promote interoperability, although it is not clear what it means that “various kinds of model simulations [are run] in a same manner.” I suggest adding more technical details.

Section 2 covers existing couplers adequately and, to my knowledge, most of the relevant coupling technologies are listed there. I suggest adding a section on the Bespoke Framework Generator (BFG) (which is already referenced in the paper) including details of the configuration metadata for that system (see the process Define, Configure, Compose, Deploy). One reason to do this is because C-Coupler1 requires a substan-

tial amount of configuration and a comparison of the way configuration is done in both systems would better situate C-Coupler1 in the context of existing work.

In the summary section 2.7 it should be pointed out that ESMF also supports 3-D grid interpolation (http://www.earthsystemmodeling.org/esmf_releases/last/regridding_status.html).

Section 2.7 also describes “model software platforms” of CCSM/CESM and FMS. It is not clear whether these are runtime environments or configuration/build managers or both. The section implies that these systems are currently inadequate requiring dramatic code modifications to add new models. However, there are no technical details given. Was a formal analysis performed? What does it mean to incorporate a new model into one of these platforms, i.e., that it can be compiled and executed or that it can interoperate with other components already in the system? Since C-Coupler1 is claiming an advance in this area, it is important to understand in detail what is required to add models to these existing systems in order to see if C-Coupler1 has improved upon them.

Section 3.1 introduces the term “experiment model” as “a model version which can run on the C-Coupler platform.” This implies some compliance rules must be satisfied, but the compliance rules are not listed. Are certain configuration files required? Certain API calls? What steps are required to take an existing non-compliant model and make it run on the C-Coupler platform?

Section 3.2 states that in some cases a separate coupler is used (generated) and in some cases direct coupling is used for performance reasons. What are the tradeoffs of the two approaches? Is the decision made automatically by some heuristics or configured by the user?

In section 3.3.1, the difference between “runtime procedures” and “runtime algorithms” is not clear. I get the impression that the former is more abstract and the latter is an actual implementation and that the extra level of indirection allows for changing the

implementation without having to modify configuration files. If this is true, then it should be stated explicitly, and if not, the difference should be clarified.

To my knowledge, the ability to register external procedures and algorithms is a novel contribution of C-Coupler1, departing from most coupling technologies. However, the motivation and expected benefits for this design decision are not adequately described in the text and the tradeoffs are not discussed. For example, is there a performance implication since an extra layer of indirection has been added? Does the requirement for extra configuration reduce the readability of the code? The authors do point out that writing configuration files manually can be labor intensive. Would assume that error-proneness also be an issue?

In section 3.3.1 and later in section 4.2, several kinds of configuration files are mentioned, although it took several readings to understand the difference. One type, the "configuration files of a component model" describe the details of a software module. Instead of calling this a "configuration," the metadata structures of the Bespoke Framework Generator (BFG) consider this a model "Definition" since it is defining or describing a set of possible input fields, output field, algorithms, and interfaces, etc. but it not actually specifying a particular configuration.

Likewise, the "configuration files of an experiment model" is similar to the BFG "Composition" phase in which previously defined models are composed into a coupled system by hooking up input and output fields. Because C-Coupler1 requires a lot of configuration files, clearly laying out the different kinds of configuration metadata and where they apply in the overall workflow would greatly clarify the system description.

Section 3.3.2 describes the different managers in the runtime system. Since in most other coupling technologies these managers are combined, it would be helpful to describe the motivation for the more modular approach and the degree to which they are dependent on each other.

In section 3.3.3, the coupling generator is introduced. This comes fairly late in the

discussion. The idea of generation and its role could be introduced in the paragraphs at the end of section 1.

The last sentence of section 4.1.1 says that algorithms registered through the API "do not have parameters and return value[s]." In this case, how does the algorithm find its input data (and what is the nature of that data; is it model fields?) and where does the output go?

In the grid manager part of Section 4.1.2, what grids are currently supported? Logically rectangular?

In the remapping manager part of Section 4.1.2, it should be noted that ESMF supports both online and offline weight generation and also supports the SCRIP format. The C-Coupler platform described in Section 4.3 and depicted in Figure 3, especially the create case, configure, compile, run case sequence is very similar to the CESM build system. If CESM was the basis for the design, it should be mentioned in the text and any differences and/or improvements should be noted.

The number of different coupled model configurations listed in Section 5 is impressive and lends credibility to C-Coupler1.

Section 5.1 compares a C-Coupler version of CPL6 with the original system, pointing out that the main driver is only a few lines of code compared with the original 1000 lines. However, the limitations of the approach are not discussedâĂŤe.g., the need to define the configuration files, etc.

It is helpful to see the configuration files listed in Figures 5 and 6, although the purpose of each file is unclear, especially the second and third column of Figure 6. The 3-D data transfer and interpolation performance experiments were executed on a modest number of processors, at least compared with other recent literature involving thousands or tens of thousands of cores. Are there plans to scale up the number of cores? Replacing Tables 2 and 3 with plots would help the reader visualize the trend lines.

Technical corrections

An annotated PDF is attached with a number of editorial suggestions. Please view with Adobe Reader (not in browser) to see comments.

Please also note the supplement to this comment:
http://www.geosci-model-dev-discuss.net/7/C1066/2014/gmdd-7-C1066-2014-supplement.pdf

---

Interactive comment on Geosci. Model Dev. Discuss., 7, 3889, 2014.