We thank all reviewers a lot for the comments and suggestions.

Reply to R. Redler

1. After reading the manuscript it is still not clear to me whether the C-coupler is a piece of software, a library, which manages the tasks for coupling (interpreting the domain decomposition, performing the neighbourhood search, the calculation of interpolation weights, handling of the data exchange and coupling date/time events, …) . On the other hand you also talk about a framework that allows to compile software components, to set up a (coupled model) run-time configuration, and to manage the whole workflow for running an experiment. Is the C-coupler at the end a composite of many different things: libraries, programs, and scripts? To structure the manuscript into sections exclusively dedicated to the coupling as such and another section about the framework and possibly workflow could help the reader to get a better insight.

Response: We modify the manuscript (P5 L13 - L21) to clearly state that "The C-Coupler1 contains a library with functions for coupling a number of component models together and a uniform runtime environment (we call it the C-Coupler platform) with scripts and configuration files for creating, configuring, compiling and running model simulations…" We combined the coupling library and model platform together in the C-Coupler because we find that we must co-develop them in order for the general design introduced in Section 3 of the manuscript. We still keep the introduction to the C-Coupler platform (section 4.3) as a subsection because it is a part of the C-Coupler1. We only briefly introduce the C-Coupler platform in the manuscript because its users' guide is available for more details.

2. It is worth mentioning that the OASIS3 and OASIS3-MCT namcouple configuration file is an ASCII formatted file which requires its own parser routine to read and evaluate its contents. On the contrary OASIS4 does not require an OASIS3-like namcouple file. Here the configuration is described in a xml file. Furthermore, OASIS4 is not a 3D version of OASIS3 but a complete rewrite (except for some SCRIP routines which are used in both software packages. I am a bit confused by your description of OASIS3 versus OASIS3-MCT when it comes to the capability of performing a 3d interpolation. In which way does OASIS3-MCT inherit the 3d coupling function from OASIS4 (P3895, L10)? In your section about MCT you make a very different statement, which is that the user is responsible for providing appropriate weight matrices.

Response: Section 2.1 has been modified accordingly (P6 L10).

3. How does the codecouple of CPL6 and CPL7 compare with the OASIS3(-MCT) namcouple and the MCT codecouple?

Response: The CPL6, CPL7 and MCT use codecouple. Compare to the MCT codecouple, the codecouple in the CPL6 and CPL7 manages more utilities, such as flux computation and the processor layout in the CPL7. The codecouple approach is not flexible enough for constructing various coupling models with an identical code version of the coupler. For example, if we want to couple a land ice model into the CCSM3, the old version of CPL6 must be modified and a new code version of CPL6 is produced. Then we will encounter a difficulty that how to merge these two code versions into one. As the C-Coupler targets to be used for any kind of coupling models, it must achieve the identical code version in various model versions. Therefore, we prefer the approach with configuration files, similar to the namcouple in OASIS3(-MCT) and the XML configuration in OASIS4. The configuration files can be viewed as the input parameters of a model version.

4. One the other hand I asked myself to which extent such details are really relevant to understand your approach. Wouldn't it be sufficient to summarise the state of the art in coupling in one short paragraph only and refer to the publications by Valcke et al. (some of them are published in GMD and are thus easy accessible) for further details? The condensed summary should of course touch upon those aspects that help the reader to understand and classify your C-coupler development.

Response: We further improve the summary for existing couplers (Section 2.8, P11 L1).

5. P3898 L14: I think your statement is not correct (or I misunderstood your point). Even with OASIS* the characteristics of the coupling do not have to be specified in the source code. That is what the namcouple and the OASIS4 xml configuration files are for. Even though OASIS* does not provide explicit algorithms for flux-computation (apart from the conservative remapping used to interpolate fluxes) I would still claim that OASIS* is also managing the algorithms for the processing of coupling fields.

Response: The corresponding statement has been modified as "However, the configuration files in the OASIS coupler do not specify external algorithms for calculating coupling fields, such as flux calculation algorithms." (P11 L23 - L24).

6. P3904 L14: A matter of taste, but I found the analogy more confusing than helpful.

Response: This analogy has been removed.

7. P3908 L19: I really did not get what you mean by " … that require equation group solving to support, such as spline". This needs to be rephrased.

Response: The spline remapping algorithm has been further interpreted (P22 L6-L7).

8. P 3908 L25: What is unique in your approach concerning your time, data, and restart manager? Isn't this something which is provided by all coupling software used for Earth system modelling. Could you indicate where you go beyond existing solutions?

Response: The timer manager functions similar to that in other couplers. For the data manager, please refer to P24 L11-L20. For the restart manager, a remarkable feature is that it can automatically detect the fields that are necessary for restarting (P24 L24 - L25).

9. P3911 L18: What makes the internal parallel decomposition that much different from the target decomposition … I am not 100% sure what you mean by "reduction for sum between multiple processes").

Response: Please refer to P26 L15 – P27 L2.

Reply to Referee1

1. There is little discussion of technical details such as how coupling fields are defined in the system, how they are reconciled between models, how models are advanced in time in an orderly fashion, or how sequencing/concurrency/lags are established between models. Some discussion needs to be added of these aspects.

Response: Please refer to P23 L1 – P24 L10.

2. Can the C-Coupler infrastructure handle unstructured grids?

Response: The C-Coupler can handle some unstructured grids. Please refer to P22 L7 – L10.

3. It's suggested in the description and implementation that CPL6 served as a starting point for much of the work, and that models that were running with CPL6 were modified to run with the C-Coupler. In many ways, the C-Coupler seems to have been built to plug in where CPL6 existed. What is missing is a discussion of what changes had to be made to the component models to allow them to couple with the C-Coupler. In addition, what is required of new components to allow them to couple with the C-Coupler?

Response: A subsection 4.4 "How to couple a component model" has been added accordingly (P31).

4. The implementation seems extremely flexible but also quite complex. Can the authors provide some addition insight regarding what's working well and what is more difficult than expected. How difficult is the C-Coupler to use and how steep is the learning curve? How robust is the system?

Response: A subsection 4.4 "How to couple a component model" has been added accordingly (P31).

5. How expensive is the remapping weights generation in the C-Coupler, is it parallel, and does the performance scale?

Response: Please refer to P21 L21.

6. It would be interesting to discuss how the 3-D interpolation is setup. I expect it's just a large linear weights matrix. Please clarify that point. How is the vertical spline interpolation handled? How will further non-linear or equation solving be handled? Will those methods scale well?

Response: Please refer to P22 L16 – L27.

7. Is it possible to provide some more content about scaling to high processor counts, even if just qualitatively. Have tests been carried out on large (ie. high resolution) grids? Please provide additional technical details in the performance results. In 5.5.1 and 5.5.2, please clarify what is being coupled/remapped. Is it a single field with 400,000 horizontal grid points? Is it a 3-D field? How are the different vertical scales reconciled if there is no regridding? Can performance at higher processor counts be provided. At 48 processors, the decomposition is still extremely coarse (about 10,000 points per processor). 400,000 horizontal grid cells is relatively large. Scaling performance out to 100 or 1000 gridcells per processor should be shown (ie. 100s of processors). It's typical for the scaling to level off or even turn over for communication dominated kernels, please show that.

Response: More processors have been used for evaluation and more test cases have been added. Please refer to Section 5.4 (P36 L15).

8. pg 3904: Line 4: not sure what "which will introduce a lot of works to do" means. Are you saying that it takes a large effort to create the runtime configuration files and you are proposing to automate that step?

Response: Yes. Please refer to P17 L23.

9. pg 3908: Line 20: Instead of "Now,", consider using "At the present time," Is the timer manager really a manager of performance timers as well as a manager of

model time (calendars)? Please clarify.

Response: Please refer to P23 L16.

10. pg 3911: The specific implementation of the parallel remapping algorithm is guaranted to reproduce bit identical answers, but comes at some performance cost. Leveraging the partial sums can signficantly improve performance in some cases.

Response: Please refer to P26 L26 – P27 L2.

11. pg 3917: Line 9: what is "intra"?

Response: Please refer to P35 L26.

12. pg 3919: Section 5.5.2, Please clarify the sizes of the grids that you are remapping between. Could you show scaling out to several hundred processes?

Response: Please refer to Section 5.4 (P36 L15).

Reply to Referee2

1. 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?

Response: A subsection 4.4 “How to couple a component model” has been added accordingly (P31).

2. 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.

Response: Many grammatical errors have been corrected according to the reviewer’s suggestion.

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

Response: We do not like the use of compiler directives such as #ifdef to specify the different configurations of the same component model, because the compiler directives always make the code of model hard to be read or be further developed. Please refer to P11 L16 - L20.

4. 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 clearerâ˘Aˇ Ti.e., the same version of a model code is used in multiple configurations.

Response: We have modified the manuscript according this suggestion.

5. 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.

Response: Please refer to P4 L20 – P5 L3.

6. 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 substantial 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.

Response: Section 2.7 (P10 L3) has been added to introduce the BFG.

7. 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.

Response: We modify the manuscript accordingly. Please refer to P8 L11 – L13, P11 L3, P11 L5 and P12 L3 – L13.

8. 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?

Response: We modify the manuscript accordingly. Please refer to P13 L2 – L3.

9. What steps are required to take an existing non-compliant model and make it run on the C-Coupler platform?

Response: A subsection 4.4 "How to couple a component model" has been added accordingly (P31).

10. 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?

Response: Please refer to P13 L14 – L27.

11. 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.

Response: Please refer to Section 3.3.1 (P14 L12 – P15 L29).

12. 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?

Response: Please refer to P15 L19 – P15 L29.

13. The authors do point out that writing configuration files manually can be labor intensive. Would assume that error-proneness also be an issue?

Response: Section 4.5 about the software reliability has been added accordingly. Please refer to P32.

14. 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.

Response: We introduce more about the configuration system in Section 4.2 (P27 L10 – P30 L4), and give a summary through comparing to the BFG and OASIS coupler in Section 4.2.4 (P29 L11 – P30 L4).

15. 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.

Response: Please refer to P17 L11 – L18.

16. 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.

Response: We carefully think about this suggestion but do not modify the manuscript. This is because the coupling generator is not implemented in the C-Coupler1 and we do not want to highlight it.

17. 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?

Response: The manuscript has been modified accordingly. Please refer to P19 L19 – L25.

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

Response: logically rectangular and some unstructured grids can be supported. Please refer to P20 L16 – P20 L17.

19. 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.

Response: Please refer to P21 L15 – L16.

20. 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.

Response: Please refer to P30 L25 – P31 L2.

21. 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 discussedâ˘A ˇTe.g., the need to define the configuration files, etc.

Response: Please refer to P33 L3 – L5.

22. 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.

Response: Please refer to P34 L5 – L18.

23. 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.

Response: Please refer to Section 5.4 (P36 L15 – P38 L18).

Reply to Referee3

1. There needs to be a better distinction made between the library, the build/configure system and the existence of a separate coupler executable (needed for the FGOALS-gc example but not clearly identified). The word "platform" is overused. For example in CESM, "cpl7" refers to both the library of routines each component must add to talk to the coupler AND the main driver of the single executable system. But it doesn't refer to the build/configure system.

Response: Please refer to P5 L11 – L21.

2. The most unique feature of C-Coupler1 is the configuration files, their conventions and how they are interpreted/used by the system. More space should be spent on describing that. I'd like to see the lines used to define the "kernel_stage" in Figure 6. Also please list the grid and decomposition names recognized by the system. That could be a first in the coupler field.

Response: More detailed implementation about the configuration system has been added into the manuscript. Please refer to Section 4.2 (P27 L10 – P30 L4).

3. The main distinction between CSM and ESM is not the need for the 3D coupling but the inclusion of bio-geochemistry, dynamic vegetation and other scientific features. A CSM needs 3D coupling, for example, between the atmospheric physics and dynamics.

Response: The manuscript has been modified accordingly. Please refer to P3 L22 – L26.

4. The authors should better define what they mean by "namcouple" and "codecouple" before using the terms.

Response: Please refer to P7 L11 – L12.

5. MCT is not intended to be a coupler. Its a library from which one can build a coupled model with or without a coupler. Since MCT is not a coupler, it doesn't care about "codecouple" or "namcouple". See OASIS3-MCT for example of using MCT in a namcouple coupler.

Response: the statement "Unlike the OASIS coupler, the MCT does not provide the "namcouple" configuration file but uses "codecouple" configuration, which means that the characteristics of coupling exchanges are specified by users in the source code of a coupled model." in the original manuscript has been removed. Please refer to Section 2.2 (P7 L14 – L25).

6. Does CoR support unstructured grids? This should be more clearly stated.

Response: The C-Coupler can handle some unstructured grids. Please refer to P22 L7 – L10.

7. The text implies C-Coupler is handling all of the restart data for all the components of a coupled model. Fields needed for a restart aren't necessarily needed for coupling so why is C-Coupler1 handling that? What happens if a component already has its own restart system? Maybe this was mis-stated.

Response: Please refer to P24 L26 – L28.

8. p 3910, line 20: please clarify what is meant by different data types. float or double?

Response: Yes. Please refer to P25 L16 – L17.

9. C-Coupler has borrowed ideas and concepts from many existing coupler systems which is fine but those debts should be referenced alongside the description of the corresponding C-Coupler feature. For example, the steps to run a model in 4.3 are almost exactly the same as the CESM system including the invocation of a "create_newcase" script. That is probably not a coincidence. The parallelization of the remapping algorithm (p 3911) is the same as MCT. A registry of functions in a coupled system was first advanced by ESMF. The references could be made as the features are described in section 4. Section 2 can be shortened to compensate.

Response: Many ideas and concepts from existing couplers have been considered for the design and implementation of the C-Coupler1. Please refer to P8 L4 – L7, P16 L8 – L9, P18 L5 – L6, P21 L2 – L3, P21 L15 – L16, P26 L4, P26 L15 – L17, P29 L13 – L17, and P30 L25 – L31 L2.

10. Can C-Coupler support 3 or more components coupled simultaneously? The FGOALS example suggests that it can but Figure 5 is so stripped down its not clear C-Coupler is doing more that “atmosphere-surface” coupling. This should be clarified.

Response: The C-Coupler can support 3 or more components coupled simultaneously. Please refer to P33 L22 – L25.

11. Section 5.6: what does it mean to add a single model to the C-Coupler platform? I’m not sure what feature is being described here.

Response: Please refer to Section 5.5 (P38 L19 – P39 L9).

12. The claim that “more and more models are using C-Coupler for development” needs to be backed up. List some institutions and models. Use the “personal communication” reference if necessary.

Response: Please refer to P41 L14 – L19.

13. Section 5.2 doesn’t have enough information to be its own subsection.

Response: This subsection has been merged to Section 5.1.