

Review #1

This manuscript introduces a newly developed modelling framework copan:LPJmL, which can simulate a wide range of socio-ecological dynamics. This work makes an important contribution to addressing the conceptual and technical challenges in modelling intertwined human and nature processes. The overall structure of the manuscript is clearly presented. The three examples introduced well highlight this framework's flexibility and versatility.

Thank you very much for the extensive positive and constructive feedback. Below we provide a point-by-point response with suggested changes to improve the manuscript accordingly.

However, the manuscript could be improved by distributing the content more properly between the manuscript and the technical manual to enhance the ease of understanding for readers. In general, I would suggest leaving the conceptual content, modelling decisions/justifications, and model logic in the manuscript, while the technical specifications in the GitHub/Zenodo repo, so that readers can easily navigate from high-level conceptual description to implementation details. More suggestions follow here.

We agree on a better differentiation between conceptualization and API documentation. The API documentation clearly belongs to the software documentation/technical documentation in the software repositories (and hosted on the website copanlpjml.pik-potsdam.de) itself. In the revision, we will carefully evaluate all described entities, remove overly technical ones and retain only those essential for explaining the model structure in the paper (e.g. World, Cell, Input, Output).

1. copan:LPJmL is defined as a framework in the manuscript. It might be ambiguous to readers as a framework can mean different things, such as a conceptual framework, a model, a sub-model, or a wrapper for the existing LPJmL to interface with copan:CORE. It would be best to clarify what copan:LPJmL exactly is early in the text. This could also help readers know clearly what kind of contribution the manuscript is intended to stress.

Thank you, the term "framework" indeed requires clarification. We will explicitly define copan:LPJmL early in the Introduction, highlighting that it extends the copan:CORE modelling framework by integrating the biophysical Earth system model (LPJmL) via a coupling interface as the ENV taxon. This clarification will specify its role as neither a stand-alone model nor a conceptual framework, but as a software and modelling framework enabling tightly coupled human-Earth system simulations with process-based and spatially-explicit detail.

2. Usually, readers are motivated to invest efforts in looking into technical details (e.g., code) only if the concepts are attractive and clearly presented. A class diagram appears unable to serve as an efficient means of communication, as it normally provides neither conceptual simplicity for broad readers nor accurate technical details for developers. It would be best to replace the class diagram with a conceptual framework that illustrates the connections and information flow between model components.

The class diagram will be moved to the appendix and replaced in the main text by a conceptual diagram illustrating model components, data flow, and feedbacks between LPJmL, copan:CORE entities, and above all the user-defined model

3. Accordingly, much of the technical description, especially the parts that mix narratives with variable names defined in the code, could be organised into a README.md/txt file in the code repo, leaving only the description of the conceptual idea and model logic (using text, pseudocode, or equations) in the paper. I saw the current code repo only has a minimal technical description in contrast with that in the manuscript. A proper redistribution of the content might be necessary.

We agree and will redistribute content accordingly. Code-specific descriptions, API-level details, and variable-level explanations will be moved to the software documentation (Read the Docs and GitHub repositories). The manuscript will be revised to focus on conceptual design, modelling logic, and coupling principles. The online documentation will be expanded to ensure that all technical details previously described in the manuscript remain accessible.

4. The code shown in the figures seems to provide limited information about the model mechanism. If the code only serves as examples of the use of copan:LPJmL, the README file in the repo should be the best place.

The code snippets shown in the manuscript are intended solely as illustrative examples demonstrating the flexible interface and versatility that enables different modelling approaches. We see that this perspective is limited to Python developers and modellers and therefore will be removed or replaced by a conceptual figure. Detailed model mechanisms belong to the respective model implementations (e.g. InSEEDS) and are documented in their dedicated repositories, which we will cite explicitly.

5. The manuscript could be improved by avoiding relying on Python syntax in the narrative. Although Python and object-oriented programming are widely used in research, using syntax-dependent expressions like "world.input[1]" might not be an ideal way to express the gist of copan:LPJmL.

We agree that Python-specific syntax in the narrative may obscure the conceptual structure of the coupling framework. We will therefore revise the manuscript to remove implementation-dependent expressions (e.g. `world.input[...]`) from the main text. The underlying functionality will then be described using language-agnostic pseudocode and a schematic flow diagram that emphasize the abstract world-state concept and the annual exchange of inputs and outputs between ENV and MET/SOC.

6. It is fantastic to learn that `copan:LPJmL` can accommodate a range of modelling approaches. However, the three examples do not contain enough information about how this framework can achieve this. Developers or model users might not be clear about what efforts they should make to switch modelling approaches within `copan:LPJmL`. I would suggest describing the examples using an identical structure: for each example, including background information, concepts, model processes, model settings, and outcomes, while highlighting the role of `copan:LPJmL` and what model users should do. In addition, please cite the source code of each example in the text.

Thank you for this helpful suggestion. We will restructure all example sections following a consistent template including: (i) background and motivation, (ii) Setup and & configuration, and (iii) Illustrative outcomes. We will also cite the corresponding source code repositories in the text.

7. Line 287, the citation "Schwarz et al." is not complete.

This is a paper under review that is closely linked to this one as it builds on `copan:LPJmL` and describes the InSEEDS model in detail. We will correct the citation, i.e. its current preprint version.

8. Line 370, ")" is missed in "(Chapter 3.1". In addition, I am not sure whether "section" or "chapter" is more accurate in the context. Please check the convention regarding the word choice of GMD.

Chapter will be replaced by section and the closing parenthesis will be added.

9. Please cite the sources of the figures and code if they are already publicly available elsewhere.

We will add explicit citations for all figures and code where it is reasonable apart from the code and data availability statement.