Reviewer 2

Comments:

From the perspective of a model user, easy installation and clear instructions for the setup and execution are critical for effective reuse. Even when models are openly available, complex installation procedures or incomplete documentation can make them difficult to adopt in practice. Users benefit greatly from straightforward installation processes, well-documented dependencies, and clear guides for running and adapting the model. Including example workflows or usage demonstration can further help new users understand how to use the model. Emphasizing how these aspects are done by the two catalogues will strength the discussion of the model adoption and reuse (Reusability)

We agree with the reviewer that clear installation instructions and well-documented model code are essential for enabling the reuse of, and further development based on, existing models. Additional contributions from model developers or the user community, such as providing example use cases for executing simulations, for instance in the form of executable notebooks, can further lower the barrier to model adoption. At the same time, we recognize that not all scientific projects have the resources to support such efforts. Accordingly, while well-documented model code and the inclusion of executable notebooks is strongly encouraged for submission to the described model catalogs, it is not a formal requirement.

For clarity, we addressed the importance of this in the first paragraph of section 2.4.2: "Software typically relies on other software components, such as libraries, modules, or external packages, to function properly. These dependencies can be cumbersome to install or integrate. It is a good practice for model developers to provide detailed documentation regarding dependencies such as up-to-date Makefiles, environment.yml, requirements.txt, and README documents. This supports potential model users that have less experience with a model to execute a model simulation on their platform of preference. Following good coding practices, this information is included in the source code, consequently, CSDMS does not include these detailed dependency specifications as part of its metadata records."

And also, we propose to extend the following text of section 3.3 Continuing to make models more FAIR, to make this more a priority in future efforts; line 608:

"Next steps for *reusability* could include developing practical mechanisms that make it easier for model developers to meet principles related to documenting provenance and providing executable code notebooks. For example, this could involve offering standardized GitHub repository templates that guide users in capturing workflows, installation documentation, and software dependencies, as well as integrating tools like executable notebooks that bundle code, data access, and documentation in an

executable format. Such approaches would lower the barrier to adoption while promoting more consistent and transparent research practices."

In section 2.4, the paper could include more examples of how models have been reused across projects or disciplines to illustrate the practical benefits of FAIR principles. The example could include: workflow, required adaptations, and the time saved compared with a "pre-FAIR" approach.

This is an excellent suggestion that we propose to address by extending the section from line 317 onwards: "Using model components of LandLab, Campforts et al. (2022) developed the landscape evolution model Hylands to investigate the influence of landslides and their associated sediments on landscape evolution. This effort involved creating three new LandLab components, reusing two existing components (FlowAccumulator and ChannelProfiler), and leveraging five service components, such as functionality for reading NetCDF data files. While we cannot quantify the exact time savings, the reuse of existing LandLab components and services likely reduced development time substantially by eliminating the need to reimplement and validate core functionality. This enabled the authors to focus their effort on developing new process representations rather than rebuilding standard modeling infrastructure."

The paper has omitted in the "Interoperability & reusability" discussion the portable images using containerization technologies like Docker and Apptainer. These tools are essential for enhancing the FAIR aspects and enabling easier access and functionality for model users. These technologies facilitate the creation and sharing of reproducible research environments. For instance, Docker allows user to package applications with their dependencies into portable containers and makes them accessible across various systems. Similarly, Apptainer emphasizes security and use in high-performance computing context.

In section 3.3 we propose to add the following to include the mentioning of Docker and Apptainer, something CSDMS is already experimenting with: "CSDMS might explore the use of containerization technologies such as Apptainer and Docker, which encapsulate code, dependencies, and runtime environments into portable, versioned units (Vanegas Ferro et al., 2022). This approach supports reproducibility and interoperability across platforms while enhancing the findability, accessibility, and reusability of numerical models through easier distribution and reuse. In parallel, the USGS Model Catalog next steps could include working with narrower modeling communities to develop modular add-ons to the metadata fields to help accommodate specific needs of specific disciplines, a publicly accessible glossary to help with comprehension between different disciplinary scientists, and promotion of increased use of the existing "software: runtime_image" field to point to runtime images such as container images."

Recommendation:

The manuscript is well organized and easy to read, and offers a valuable, community-oriented account of how FAIR-4RS can be used for Earth-surface models. I recommend minor revision based on my previous points.