

Dear authors,

The revised submission of your manuscript was first reviewed by two reviewers, who had also evaluated the original manuscript. Because their estimates of the manuscript after revisions were quite different, I decided it was needed to have the work evaluated by additional reviewers. That is why it took up quite some extra time and we ended up with 4 reviews, of which 3 have provided additional comments to be addressed. The additional reviewers have evaluated the work considering all existing review comments. The overall result is that the manuscript needs further revisions before it can be published in GMD. The comments are given below. I suggest we take on the point raised by reviewer 3 about which GMD manuscript type to assign the manuscript to at a later stage.

Kind regards

Heiko Goelzer

Reviewer #1

Suggested major revisions.

I appreciate the improvements to the manuscript, but the authors leave out some details.

No information model in any form is presented. Just a graphical abstract of the system is presented in the manuscript without the description of the architecture with classic software engineer representations.

Moreover, the bibliography has been updated without the suggested papers that I mentioned in the 1st round of review, especially the one regarding adaptive GUI generation based on DICOM:

Gambino O, Rundo L, Cannella V, Vitabile S, Pirrone R. A framework for data-driven adaptive GUI generation based on DICOM. J Biomed Inform. 2018 Dec;88:37-52. doi: 10.1016/j.jbi.2018.10.009. Epub 2018 Nov 9. PMID: 30419365.

where the authors can learn how to depict a good software design.

Look forward to the revised version of the manuscript.

Reviewer #2

Believes that the authors have addressed the comments from the previous review and has recommended to accept the manuscript for publication.

No additional comments.

Reviewer #3

Suggested minor revisions.

In Inishell 2.0, the authors present a tool that converts an XML-based description of a numerical model configuration structure into a Qt based GUI for use on a desktop operating system. Specifically, this tool is designed to facilitate the creation of INI-based configuration files used by numerical models.

The two reviewers' comments appear to fall into three main concerns. I give my summary of the reviewers' comments in addition to my thoughts.

1. The original manuscript and indeed the manuscript paper read more like a technical report rather than a science paper.

I am of two minds on this. On one hand I am missing a strong scientific contribution by this tool – what new science can be done with it? There is no new testable hypothesis and no new method codified in a tool. Rather, it is a tool that facilitates configuring existing models by translating a series of setting options into a layout the authors deem suitable. However, I believe such a contribution falls within the GMD submission guidelines for “development and technical papers, describing developments [...] or technical aspects of running models such as the reproducibility of results.” There is a significant push in the earth science community for reproducible science, including the configuration of numerical models. On line 80 the authors note “This has the advantage that a copy of the said configuration file is then a reproducible description of the numerical simulation that has been performed [...] keeping a changelog of the configuration files [...]” which clearly links with the idea of reproducible science and codifying the steps taken during the workflow of setting up a model. However, this point is never fully described. How are the changelogs produced? Where are they stored? How does the user interact with them?

Throughout the revised manuscript there are anecdotal comments that lack sufficient context related to end-users interacting with the models and configurations. I am familiar with this groups SNOWPACK model and their ongoing work with practitioners in the avalanche hazard forecasting community. Thus, when they make statements, for example, that users struggling to use the command line terminal, it is unclear who had this issue – are these practitioners or scientific users?

I therefore think that the authors need a bit of polish to pull together the context they added in revision two to better link the concepts of reproducible science, end-users who are domain experts but not model experts, and what types of GUI features are required to support these use cases. Having a GUI that facilitates reproducible research and ensures domain experts have access to appropriate models seems to be a useful contribution. I believe that citations from the reproducible science literature to support the GUI decisions would address the reviewer's concerns [0,1]. I understand [1] to want a further description of what I believe the authors want to address: “how do we help domain experts who aren't modellers use our tools in a reproducible way.” At the moment the synthesis that the authors added in revision 2 to address this feels tacked-on.

[0] "A validation section is absent. I do not pretend a user experience study, but a section where the use of your system is easier than the "traditional" way."

[1] "Substantially, there isn't a good background literature where other methodologies are applied to solve similar problems."

2. Data model and GUI creation

There is a substantial body of research in the Human-Computer Interaction (HCI) literature on how GUIs are designed, A/B tests to determine optimal layouts, etc. The authors note that "[t]he focus is also not on the visual appearance of the GUI but on the data that has to be provided by the end user." Certainly, there must be some method to how the UI is laid out. Is there a novel way in doing so to optimize interaction with numerical models? This links with [0] but also [1] where presenting a UI in a manner to help domain experts interact with models requires some thought in how the UI is designed.

I believe it is to this point that the reviewer #1, in 339-RC2.pdf who recommended doi: 10.1016/j.jbi.2018.10.009, is reacting. It seems the authors attempted to address this in Table 1, however it is not clear to me how this table is supposed to be read – top to bottom? Left to right? I believe that the authors should tighten section 3.2 to make it more clear how the various levels in the hierarchical design works and exactly how the XML options are mapped to these internal data models for generating the UI. I would like a more detailed description of how basic range checking and regex checking are done – is it as the user types or when the INI file is created? Lastly, and this does also relate to point 3 (generalization) below, is that describing the internal data model and class relationship as suggested by the reviewer would help demonstrate applicability to a wider gamut of toolsets. Most models do not use INI configuration files. Thus, how easily can a user swap the INI writer for a, for example, a JSON writer? These internal data model descriptions and class layout diagrams would be helpful to understand this.

I downloaded and compiled IniShell. In general, it looks nice. However, there are non-standard UI decisions that, as a first-time user, were sufficiently quirky and non-standard that I was confused. I had to check the mouse-hover tooltip to understand what it did. For example, the save icon is not the standard floppy-disk icon. The open icon looks like the standard 'new file' icon instead of the more standard "open folder". The preview icon is a printer (I assumed this printed?). In the left-hand side panel, the example INI files are listed twice (seemingly). These types of GUI design decisions and departures from well-known paradigms have a 'scientist UI' feel to it versus "lovingly hand-crafted" by a UI expert feel. It is my opinion "focus is also not on the visual appearance of the GUI" requires more justification.

3. Generalization of the tool

Reviewer 1 and 2 both question the general applicability of the tool including the inclusion of geospatial data. The authors stress in the manuscript that although designed for their models, it is quite generalizable. Given IniShell is a "semi-automatic form filler" there is seemingly limited tight binding to the underlying numerical models. However, the authors

do note that the INI standard “[...] therefore has been defined more strictly for this work as well as extended [...].” It is unclear to me how easily it would be to implement this in another model or if INI conversion tools can handle this. I would like to see an explicit table of what was extended to clearly explain what would need to be added to a downstream consuming model to take advantage.

Regarding inclusion of geospatial data, I found the authors response satisfactory. Setting configuration keys that point to the input data is likely sufficient as detailed data validation is likely best in the pre-processing step or during model initialization.

Other notes:

I believe the authors have well addressed RC1 and RC2 questions regarding the use of Qt. In my view Qt is absolutely a reasonable choice for IniShell. Qt has been in active development for many years, has a mature community, open source, &c. Any of the big cross-platform widget toolkits would be appropriate for this work and I think Qt is perfectly valid. I agree with the authors that a web-based tool provides limited value for a scientific computing tool.

I am curious if the authors have thought about an approach to automatically generate the XML file from the model source code (also noted by the reviewer). The XML file still represents a non-zero maintenance burden and could end up out of sync of the source code despite all efforts. Code rot happens to the best of us.

Summary

Based on my review of the manuscript as well as the two other reviewers’ comments, I believe that this manuscript would make a contribution to GMD under “development and technical papers, describing developments such as new parameterizations or technical aspects of running models such as the reproducibility of results.” However, I think there is room for improvement in tightening the manuscript. Specifically, to: 1) improve the literature summary/gap and discussion to position this manuscript around reproducible research and expert-user/non-modeller usage to clearly meet GMD contribution guidelines; 2) more clearly articulate how the internal IniShell data model translates the input XML file into a GUI and could allow for generalization to different configuration formats; and 3) to clearly articulate the INI spec extension to ensure generalizability.

I would suggest acceptance with moderate revisions.

Reviewer #4

Suggested additional revisions that mainly address issues in structure and clarity from a design perspective.

This is an interesting paper that addresses a problem that deserves attention. The paper describes a solution to the problem, but fails to clearly identify the problem and the scope of the paper. Much relevant information comes implicitly throughout the paper, but the

paper as such would benefit from a clearer structure focusing on the problem/use case, methodology/approach/architecture, implementation, discussion and conclusion/summary. As it reads now it is difficult to follow the authors through this process. The impact of the paper would be greater if this is clarified.

In the methodology/approach/architecture, usage of interaction and sequence diagrams would be beneficial along with a clear description of the data model supporting this interaction. The illustrations that are provided are nice, but needs more support.

Specific comments

Section 1: I would like to see the introduction split to a more general background on generic GUI and a separate section on problem description which also explains the nature of numerical models and how they are configured and operated in more detail. As it is now, the sub section on requirements appears to lack justification. Additional references on the problem description would be useful.

Section 1.2 – last paragraph: this is a very bold statement which I can't see is justified in the current text. Either more references or a better problem description and analysis would be necessary.

Section 2: Should be renamed to methodology or similar preceded by a proper problem description and justification of selected approach.

Section 2: In order to justify inishell I miss a more thorough problem description and analysis of potential solutions. There are good reasons to choose inishell, but it doesn't mean that there aren't alternatives that appears more modern. Referring to INI as the informal standard is not justified, references are needed or a more thorough analysis of approaches.

Section 2.1 – paragraph 7: Referring to the XML file as the data model for the configuration files won't do. A more generic approach would be necessary using e.g. UML or similar, representing the various sections identified in the GUI presented later.

Section 2.2 – first paragraph: What is meant by "...numerical model profile and specific configuration profile...". I am not sure a "specific configuration file" is the same as "a model with loaded settings ready to run". A better explanation is required.

Section 2.2. figure 3: The GUI representation is jumped without a proper model describing the purpose of the different sections in the GUI. A logical workflow, potentially represented through an interaction diagram or similar would be useful to understand the context.

Section 2.2 – first paragraph: To my knowledge MeteorIO isn't a numerical model but rather a pre-processor of input data. Clarification would be welcome.

Section 3: As far as I can see both section 2 and 3 are implementation. A better separation of content would be welcome. Again referring to INI as the informal standard for configuration files has to be justified. In general I would claim that YAML is at least as widespread as INI, but there can be differences between communities that for legacy reasons stick with INI. References would help in this part.

Section 3.2: I am sure I would agree on this being the architecture. As mentioned earlier, to simplify reading a general introduction followed by a proper problem description and analysis, then a section on methodology or approach that contains the architecture, leading up to the implementation and a discussion of this would simplify the reading of the paper. As sections 2 and 3 are somewhat interleaved and not clearly defined scope wise a restructuring would be welcome.

Section 3.3: Not sure I understand what is meant by semantic names, seems repetitive.

Section 3.7: The work flow description would benefit from interaction or sequence diagrams. It is not clear if GUI and model has to run in the same environment or if remote control is possible, This would be interesting to know. Also is this more than a configuration system? Is it also a execution environment?

Section 3.8: I can't see that a numerical model (nor multiple) is loaded into INIShell. Rephrasing would be welcome to clarify what a numerical model is.

Section 4 – second paragraph: For the time being running numerical models is commonly done in HPC than cloud environments, but could depend on the community and type of model. Referring to the cloud without addressing HPC would be oversimplifying the complexity of numerical models.

Also referring to my comment on section 3.7, a clarification of execution environment would be beneficial up front, not indirectly addressed in discussion.

Section 4 – third paragraph: file formats for what?

Section 5: I can't see that the statement that a GUI gives better control over the configuration than a text based solution. Justification would be required, also in the discussion. And the justification should also refer back to the problem description that is the foundation for the paper.