Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-339-AC1, 2021

Geoscientific Model Development Discussions

Interactive comment on "Inishell 2.0: Semantically driven automatic GUI generation for scientific models" *by* Mathias Bavay et al.

Mathias Bavay et al.

bavay@slf.ch

Received and published: 5 February 2021

Answer to Anonymous Referee #1

>1. There are plenty of widget toolkits. Why do you use Qt? Is this framework suitable to other widget toolkits/programming-languages?

We have chosen Qt for its multi platform abilities that go beyond the widgets, including some low level functions that allow us not to have to care about low level platform specifics while offering a good integration within each supported platform. Moreover, it is object oriented which we found helpful for our tool and it is open source which means that are source code can be recompiled by all our users. Qt is also one of the oldest and best maintained toolkits to write generic (C++) GUIs and a de facto standard in

this domain.

>2. Is this framework general purpose?

Yes, Qt is general purpose (it offers both GUI widgets and non graphical functions to fully abstract the platform differences).

> 3. You use too much the word "model". I think that you use the word "model" also to point out the "template of the GUI layout". Moreover, the "numerical model" is an entity where the association between one widget and one parameter is accomplished (?). On the contrary, Must the association be "handmade"?

All occurrences of "model" in our manuscript refer to a numerical model, that is a computer program that simulates physical processes such as the Snowpack snow cover model, the WRF (Weather Research and Forecasting) model, the ECMWF models or any Numerical Weather Forecast model.

We have significantly rewritten the introduction and the "Principles" section because it appears that they severely lacked clarity and we have also introduced a discussion of our approach from the point of view of data models since it should be an interesting addition. A new figure has been created to show the general principles of numerical models from the user's point of view and the figure showing the principles of Inishell has been slightly expanded for clarity.

>4. The information model is not defined in your work. For e.g., you could use an ERD scheme to define the structure of the information in your work. The UML class diagram could show the code structure and the relationships among the classes. (See the first suggested reference at point 8. to see how to define an information model).

This is also a consequence of our lack of clarity: we do not define a data model in our work because Inishell consumes a data model provided as an XML file to populate a GUI where a user can enter configuration inputs (for example the spatial resolution or the timestep) for a numerical model (for example Snowpack) that will be stored as an INI file. This INI file is then read by a numerical model (for example Snowpack) to produce some outputs. So the numerical model assumes a given data model for its configuration data that is then formerly laid out in the XML file (to be written by its developers, not by Inishell's developers). Therefore, we have the following components: 1) a numerical model written by a third party (ex. Snowpack) that can read its configuration from an INI file; 2) an XML file that describes the data model of the configuration inputs of the numerical model (to be written by a third party, for example Snowpack's developers); 3) a GUI populated by Inishell based on the provided XML file; 4) configuration inputs provided by the numerical model user by the mean of the Inishell generated GUI and stored as an INI file by Inishell.

This has now been described in the rewritten sections.

>5. There are two aspects in a GUI: the layout and the callback functions. I cannot see the latter aspect. The entire system seems to be a "semi-automatic form filler" because there are no functionalities to be evoked.

There are not really any callback functions besides implementing what has been specified in the provided XML file: the GUI widgets receive the user inputs, validate them based on the rules defined in the XML file and write them down into an INI file. In this sense, Inishell is an automatic form generator with a few auxiliary features (such as generating a preview of the INI file or loading an existing INI file). Inishell does not "process" data or produce any outputs other than the conversion of the user provided inputs in the GUI into an INI file with its syntax.

>6. The system is not completely automatic: it seems an interface to generate another interface. Indeed, the GUI layout should be inferred from the parameters. Given a specific domain, there should be a module aimed to perform the creation of the layout on the basis of the parameters. This module should be able to create the XML related to the layout. It shouldn't be necessary an artificial intelligence approach: just a data management approach could be sufficient. For e.g., a table containing the associations: INI

ENTRY station1=xxx.smet -> ASSOCIATION TABLE station=filename -> AUTOMATICALLY GENERATED XML CODE: <parameter key="STATION#" type="filename"

In order to remain generic, there is no domain-specific knowledge in Inishell. Each numerical model might come with its own parameter names and meaning. For example, even within a specific numerical model the same parameter name might have different meanings depending on the context: a configuration entry such as STATION1 might receive a filename or a database key or even geographic coordinates within the same numerical model, depending on the context! This means that there is not enough information to generate a relevant GUI only with the parameter names. Moreover, the INI file is quite user friendly (because of it is only very loosely structured) but it can not provide any semantic information (such as data types, ranges, regular expressions for validation or dependencies). Since the INI file will be edited or even created from scratch by the users, it is also not possible to store there any help text or other information. Therefore, the XML file contains all the semantic information for a specific numerical model (and it is often the only place where such information is formally expressed besides partly in an implicit, diffuse way within the numerical model source code, as assumptions or checks about the configuration data).

>7. 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.

More details have been added in the conclusion about our experience with and without Inishell from the point of view of our users and from the point of view of support requests.

>8. The paper is lack of recent bibliography. The following two papers are interesting but you should include other papers recently published. Orazio Gambino, Leonardo Rundo, Vincenzo Cannella, Salvatore Vitabile, Roberto Pirrone, A framework for data-driven adaptive GUI generation based on DI- C2 COM,Journal of Biomedical Informatics, Volume 88,2018,Pages 37-52,ISSN 1532-

0464, https://doi.org/10.1016/j.jbi.2018.10.009. Chunyang Chen, Ting Su, Guozhu Meng, Zhenchang Xing, and Yang Liu. 2018. From UI design image to GUI skeleton: a neural machine translator to bootstrap mobile GUI implementation. In Proceedings of the 40th International Conference on Software Engineering (ICSE '18). Association for Computing Machinery, New York, NY, USA, 665–676. DOI:https://doi.org/10.1145/3180155.3180240

Although there are some data driven parts in Inishell, it remains quite basic: some keys are shown or hidden depending on other keys (just acting as booleans). The second paper is quite interesting but Inishall actually does somehow the opposite: without any hints regarding how the GUI should look like, Inishell populates its GUI as some sort of a regular grid of widgets.

>From the introduction: The paper concerns a framework aimed at the automatic generation of GUI by means of an XML description and an INI file containing structured information about the parameters having a relationship with each widget.

This is what we must better describe: The XML file contains the structured information that allows populating a GUI to gather user inputs and write an unstructured INI file containing these user inputs. Inishell is therefore "only" a data entry software for configuration files with no domain specific knowledge of its own, instead relying on an XML file to provide it. The INI file is an output of Inishell (although it can re-read an existing one in order to edit it) and an input for the numerical model (such as Snowpack).

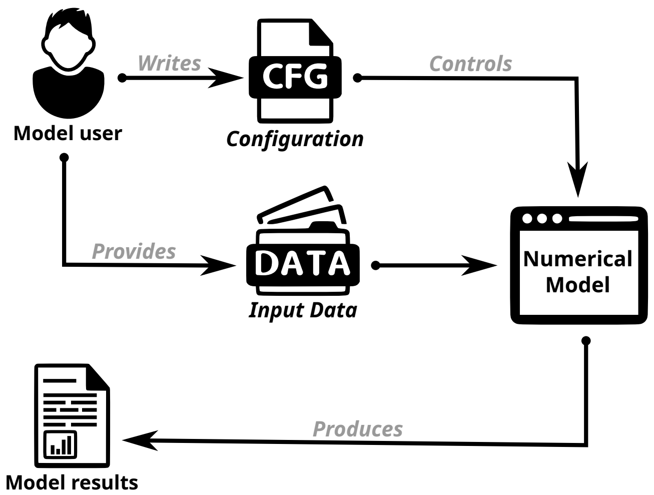

Fig. 1. Principles of numerical models from the user's point of view

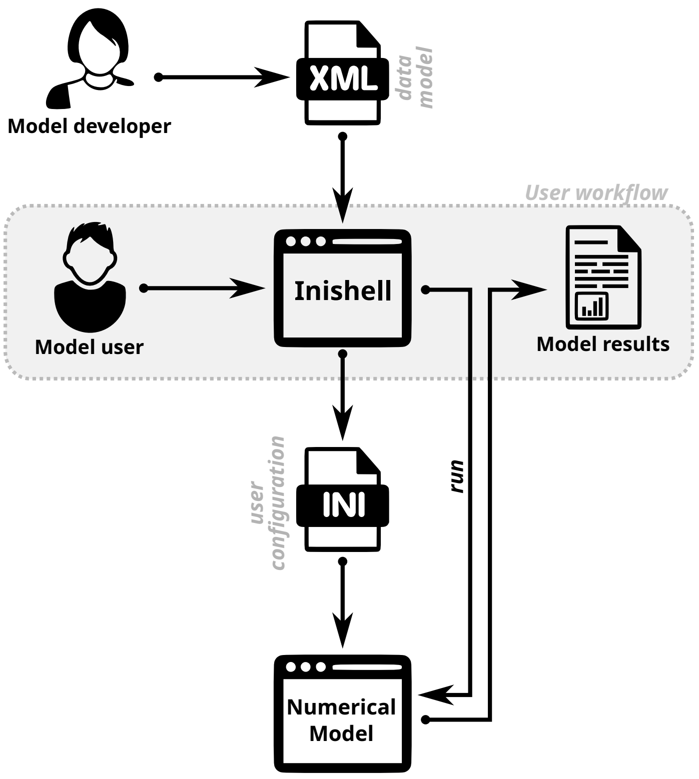

Fig. 2. Principle of operation of Inishell from the user's and model developer's point of view