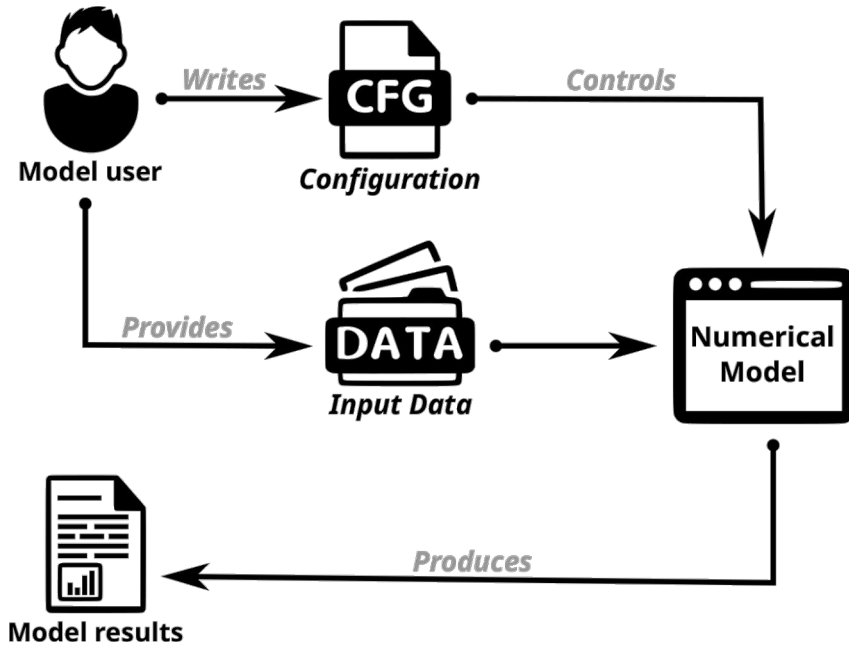




Inishell 2.0: Semantically driven automatic GUI generation for scientific models

M. Bavay, M. Reisecker, T. Egger, D. Korhammer

Context: numerical models



- *computational models designed to simulate and predict the behaviour of real-world or physical systems*
- Widespread, useful but often challenging to configure
- People don't read documentation [1]

→ One solution: Graphical User Interface (GUI)

Context: Graphical User Interfaces

- *Rapid Application Development Environment* vs standard toolkit
- Time consuming (50% of development time [1])
- Numerical models might have many configuration parameters (~350 in Snowpack [2] and its MeteoIO preprocessor [3])
- Models coupling: high modularity → requires high modularity in the GUI
- Danger: the GUI slows down the model development

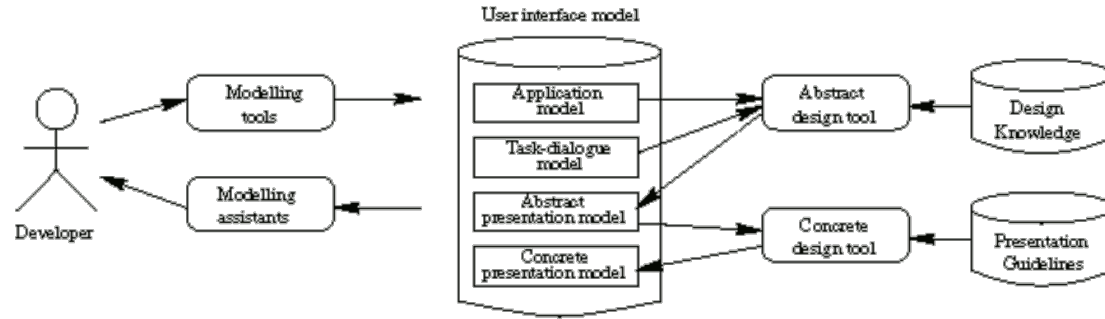
[1] Myers, B. A. and Rosson, M. B.: *Survey on user interface programming*, in: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 195–202, ACM, 1992.

[2] Lehning, M., Bartelt, P., Brown, B., Fierz, C., and Satyawali, P.: *A physical SNOWPACK model for the Swiss avalanche warning: Part II. Snow microstructure*, *Cold regions science and technology*, 35, 147–167, 2002.

[3] Bavay, M. and Egger, T.: *MeteoIO 2.4. 2: a preprocessing library for meteorological data*, *Geoscientific Model Development*, 7, 3135–3151, 2014.

Context: Declarative User Interface Models

- Alternative strategies:
Declarative User Interface
Models [1], Model-Based UI
Development [2]
- Highly theoretical, steep
learning curve [3]



(Da Silva, 2000)

[1] Da Silva, P. P.: *User interface declarative models and development environments: A survey*, in: *International Workshop on Design, Specification, and Verification of Interactive Systems*, pp. 207–226, Springer, 2000.

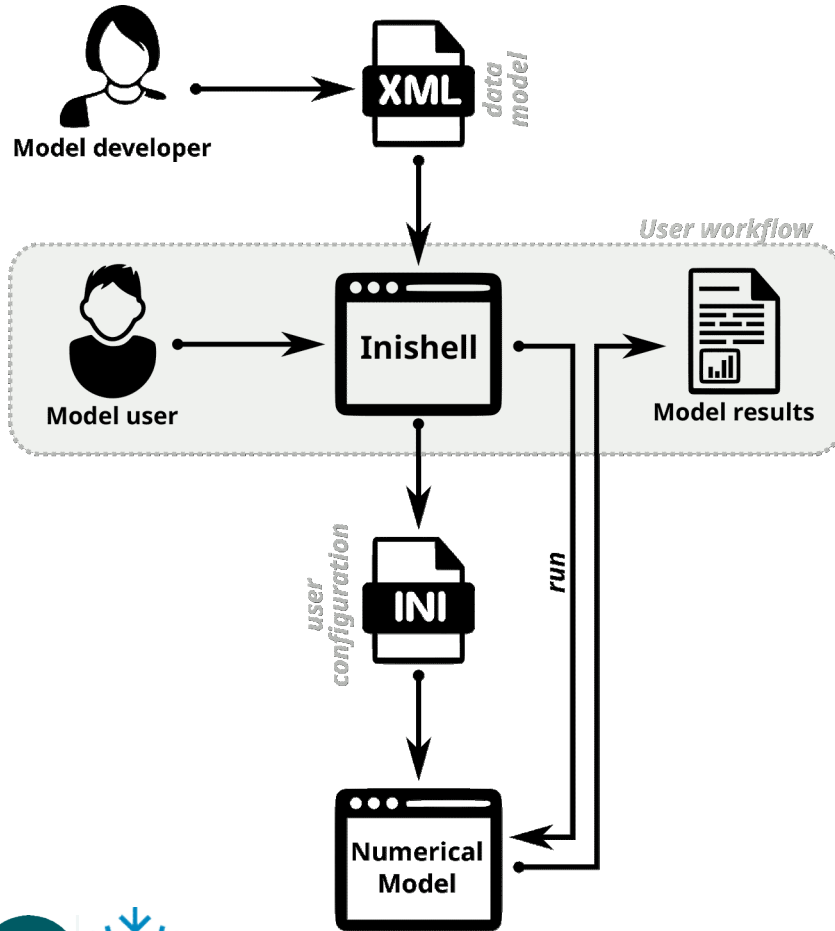
[2] Paterno, F.: *Model-based design and evaluation of interactive applications*, Springer Science & Business Media, 1999.

[3] Bogdan, C.: *Declarative interaction towards evolutionary user interface prototyping*, in: *IFIP Conference on Human-Computer Interaction*, pp. 83–90, Springer, 2017.

Our requirements

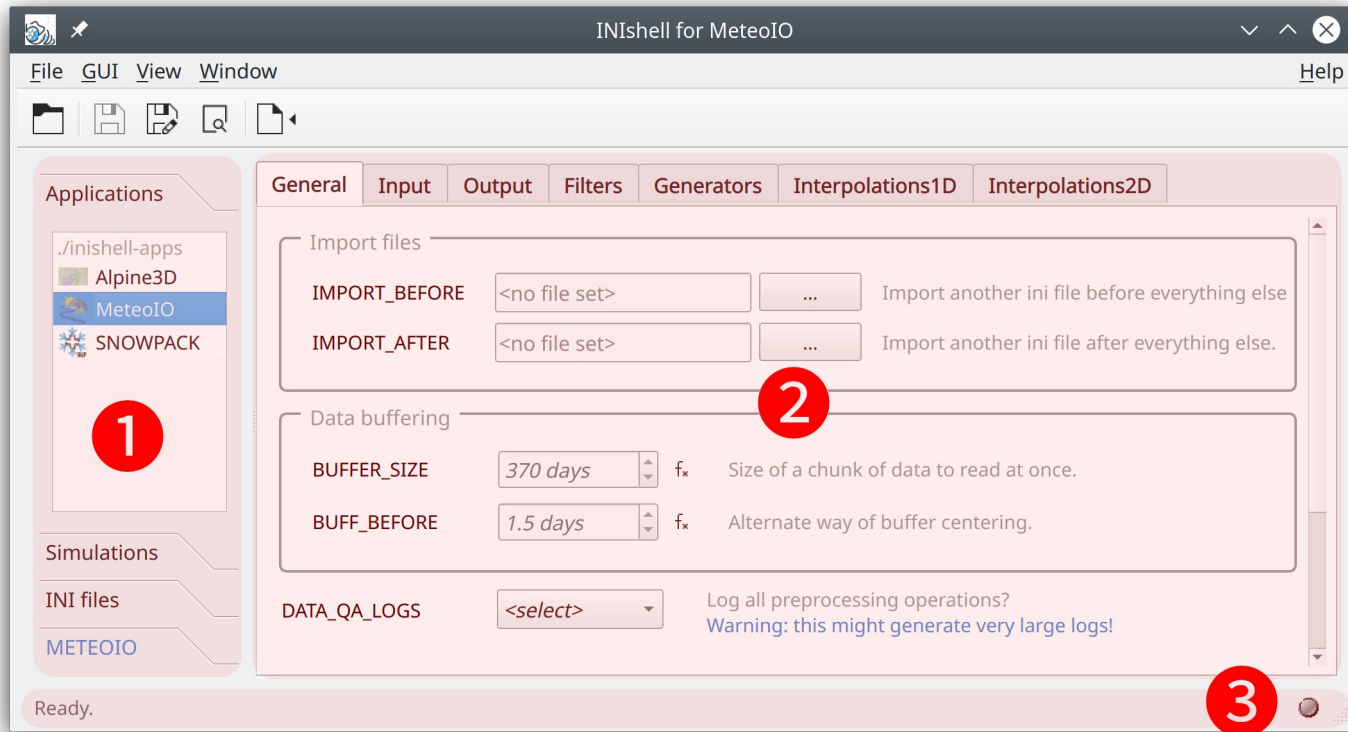
- GUI for our end users: graphical way to generate the configuration file (structured as configuration keys & sections)
- **Quickly & easily** add new configuration key
- Provide short help text for each configuration key
- Validate user inputs
- Write out a standardized configuration file
- Support manual editing of configuration file
- Support running command line models from the GUI

Our Approach



- Slimmed down derivative of Declarative User Interface Model
- Semantic description of the configuration keys in XML
- On the fly generation of the GUI
- Once the user is done, writes an INI file for the numerical model
- Optionally runs the model from within the GUI

Inishell Overview



- (1) Choose which numerical model to run
- (2) Configure the model
- (1) Optionally run the model (lowest drawer)
- (3) Warnings, infos

Example XML, GUI & INI

a) XML declaration

```
<parameter key="TIME_ZONE" section="Input" type="number"
           format="decimal" optional="false">
  <help>The time zone of your data</help>
</parameter>
```

b) Inishell GUI

TIME_ZONE f. The time zone of your data

c) INI entry

```
[Input]
TIME_ZONE = 1
```

- **data types:** strings, dates and times, paths to files and paths, decimal numbers, integral numbers, booleans

- **Validation:** data types, optional range checking, regular expressions, mandatory / optional, simple dependencies

Example XML, GUI & INI

a) XML declaration

```
<parameter key="STATION#" type="filename"
           mode="input" replicate="true">
  <help>File name for station number #</help>
</parameter>
```

b) Inishell GUI

STATION# + - File name for station number #

No 1:	WFJ.smet	...
No 2:	DAV.smet	...

c) INI entry

```
[Input]
STATION1 = WFJ.smet
STATION2 = DAV.smet
```

- Simple template system for multiple identical parameters
- Grouping elements: sections (as tabs), frames (as frames)
- Also workflows to run a model from the GUI & capture its output

Does it really works?

- First version of Inishell 10 years ago (in Java)
- Adding new configuration keys done in minutes → not holding back model development
- Very helpful for the end users, much better uptake of new model features
- **But** running the model from within the GUI was originally missing and is a key requirement
- Embedded text editor for the ini file also appreciated (quick cut and paste between simulations, reloading into the GUI)
- Rewrite in modern C++ of the old code base, faster reaction times of the GUI

Thank you!

More in paper (in review):

Bavay, M., Reisecker, M., Egger, T., and Korhammer, D.: *Inishell 2.0: Semantically driven automatic GUI generation for scientific models*, **Geosci. Model Dev. Discuss.** [preprint], <https://doi.org/10.5194/gmd-2020-339>, in review, 2020.

