

Interactive comment on “Spud 1.0: generalising and automating the user interfaces of scientific computer models” by D. A. Ham et al.

Anonymous Referee #2

Received and published: 15 September 2008

General Comments:

As the prominence of numerical models is ever-increasing as a tool in modern scientific exploits, so too grows the challenges associated with their correct application and use. In addition to formulating a correct scientific methodology, scientists are faced with technical difficulties associated with input parameter choices, which must be made using often poorly documented model input files. Ham et al. describe in this article a new system, Spud, to simplify this procedure and increase its transparency, with the benefit of reducing user error. While the automatic generation of a user interface is indeed a very useful tool, I question the overall usefulness of Spud given the time required by the developer to setup the system. However, as the concept and implementation of Spud appear to be very thoroughly explored and implemented, I feel we should al-

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

[Discussion Paper](#)



low the community to decide its usefulness. In general, I find the article provides a good description of the system and its functionality, but lacks clarity at times and would benefit from further examples. I would strongly recommend this article for publication subject to a few minor corrections.

Specific Comments:

Section 2.1

1. In the description of current problems associated with input files (e.g. Section 2.1), no mention is made of the use of namelist files. As these files are commonly used as input for numerical models (in particular in oceanography, the author's area of expertise) and provide a similar set of advantages to the xml based method described by Spud, their omission is rather striking. In particular, namelist files allow the grouping of input parameters, the use of comments, and variable names, and allow an automatic input of a section of a namelist with a single command from the model. As such, adding options (and comments) to an input file and the model code is quite easy, and I find it difficult to see the advantage of using a formal grammar.

2. As an oceanographer, I am not accustomed to the jargon used here and I would imagine this to be true for scientists in most areas of study. A few specific examples to describe what are meant by terms such as "formal grammar" and "formal language" would greatly aid the readability.

3. Often numerical models are executed via job scripts that modify model input files (e.g. in the case of a sequential set of runs, the initial and final timesteps need to be updated). How difficult would it be to interface the input files generated by Spud with such a system?

Section 2.3

4. lines 11-22: I would argue that model performance is the most limiting factor, rather than the reasons stated here.

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

[Discussion Paper](#)



Section 4.3

5. Again, some examples would really help to clarify how Spud works. If there is insufficient space in the text and figures, perhaps some examples could be provided in the supplementary materials.

Section 6

6. It is not clear to me what exactly is meant by "in-memory"; and how a model accesses the parameters specified in the input file. Is it as simple as simply having a variable, for example, `r_pie`, which can then be given a value in diamond, which is then be accessed in the model using the variable "r_pie" (where `r_pie` is defined by libspud)? It doesn't appear to work this way...

7. Page 138, line 26: It would help if the statement beginning "Interfaces to the options tree are provided..." could be elaborated.

8. If a number of interface routines are required to obtain the input parameters, how does this simplify the work of the model developer? Please clarify, because as it is currently done, a developer is only required to define a variable and read it in, rather than setup a series of additional routines on top of an entire XML frontend.

Section 9

I agree that input files create a "significant barrier" to users of numerical models and can lead to errors in model setup. However, I would not consider input files to present a "serious development cost". Moreover, as stated in the comments above, I don't see how Spud reduces the development cost, but rather seems to increase it further.

Interactive comment on Geosci. Model Dev. Discuss., 1, 125, 2008.

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

[Discussion Paper](#)

