Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., author comment AC3
https://doi.org/10.5194/gmd-2021-311-AC3, 2022

Reply on RC2

Reiner Jung et al.

Author comment on "CP-DSL: Supporting Configuration and Parameter Selection of Ocean Models with UVic (2.9) and MITgcm (67w)" by Reiner Jung et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-311-AC3, 2022

Thanks for your valuable feedback to our submitted paper!

We respond to the comments that require changes.

Your comment:

- "The paper spends a good amount of time discussing the requirements of an ocean model configuration system (important), but there is considerable less attention to explaining what the actual objectives of this project are and how the chosen approach helps to achieve these. As the authors quite rightly state the added complexity and risks (in terms of new dependencies) need to provide benefits, but there is only a limited discussion of what these are, and in particular how the specific choices in this project deliver these."

Reply:

- We agree that the goals and objectives should be more specific. We have revised Section 6 to reflect this and have explicitly linked the design decisions to the challenges outlined in the introduction. We have also expanded the paragraph in the introduction to state this more clearly.

Your comment:

- "This, and the fact that the actual implementation is only described in a rather abstract way, with only a few restricted excerpts and no concrete examples, make it hard to judge to what extent such benefits are delivered by this project."

Reply:

- We extended the examples in the paper significantly and instead of presenting portions of it in separate listings, we integrated them into a longer listing giving a better example on how the specification files look like. Furthermore, we extend the replication package with a complete step by step instruction on how to repeat example setups with UVic and MITgcm.

Your comment:

- "I would really like to see a more complete overview of the features that have been implemented, and a lot more concrete examples of actually what goes into the "configuration model" vs. the "declaration model" so we can get a better idea how universal the language is and able to specify things in a model independent way."

Reply:

- The examples added for the previous comment were designed to demonstrate the features in detail. See the new Listings 1 through 5 in Section 6, which are a full setup for UVic and an excerpt for MITgcm.

Your comment:

- "the most problematic section in my view is section 5 which provides a very abstract overview of the syntax of the proposed language but only through a few excerpts that do not give a very clear picture of the language as a whole."

Reply:

- Similar to a complete example, a complete syntax would also be very extensive within the paper. Therefore, the replication package was further expanded to emphasize this. We also provide the complete grammars in the replication package, including references to the actual grammars in the code base.

Your comment:

- "I also do not find the UML diagrams to be particularly enlightening (figures 3-6)."

Reply:

- The diagrams show the DSL implementation in a compact form that would otherwise be very extensive. They represent the abstract syntax of the DSLs. Nevertheless, we have revised Section 5 (now Section 6) to address your concern.

Your comment:

- "it's only in section 6 that we finally get told what CP-DSL is actually made of, but the description of the key components in section 6.1 is very terse. Please explain for instance what EMF is."

Reply:

- We have added short explanations for the used technology in that section (now Section 7.1). An excerpt regarding EMF from the additions to the text of this section: "XText is a DSL development framework and toolchain which provides its own IDE and allows to create a DSL, generator, editor and other facilities for a DSL mainly based on a grammar specification. XTend is the template and programming language used with XText and served as inspiration for our Template DSL. Finally, EMF is an implementation of the essential subset of the Meta-Object Facility (EMOF)~\citep{mof20042} that allows to specify metamodels. XText uses EMF to model the abstract syntax of grammars."

Your comment:

- "As a side note, it also seems that some of these components bring in a dependency on a specific version of Java which seems to be in contrast with one the requirements (line

251-253) and isn't particularly well supported on some of the HPC systems that ocean models run on."

Reply:

- We agree that this is a potential risk as we use a technology not widely used in the domain. However, we used this technology stack, as it allows for agile language development, i.e., we can create and modify the DSL in short time and gain feedback from users on a working prototype. This allows to advance faster and closer to the users' needs. Furthermore, on the HPC systems in our partner organizations, Java is available. It is also possible to execute our tools in a Docker container, e.g., utilizing Singularity. Especially, the Jupyter setup is designed for this purpose. On standard workstations, Java is not an issue. As mitigation of this risk, we provide the abstract syntax and metamodel to facilitate porting the parser and code generator to another technology stack, if necessary.

Your comment:

- "as mentioned before the authors do not really evaluate or discuss what has actually been achieved in this work. The evaluation by other users in section 7 is described in a rather superficial way. The first evaluation in section 7.1 seems to be about a quite different version of the language, so the only information we get, about a second evaluation is "The division in general parameters and modules was considered useful."

Reply:

- The evaluation is primarily driven by our two case studies and uses (a) setups for existing experiments as examples, and (b) reviews by research software engineers. We have revised the evaluation section to better reflect this. We also added details of the used example and further comments on the DSL from domain experts to Section 8.

Your comment:

- ""Also the reworked YAML syntax was rated easy to understand." This is actually the very first place (right at the end of the manuscript) where it is actually mentioned that the CP-DL syntax is closely (?) related to YAML - the other places YAML is only mentioned to contrast with XML and JSON."

Reply:

- We added YAML as source for our concrete syntax design in the revision of Section 6. This was the result of our first evaluation step. While the abstract syntax remained the same, we adopted concrete syntax elements from YAML.

Your comment:

- "As discussed ocean modelling already brings together a variety of expertises (e.g. oceanography, numerical analysis and HPC) and the CS flavoured approach followed in this paper with DSLs, context-free grammars and metamodels adds a whole layer on top of that. This makes it important to have a clear view of the intended audience and adjust the language to it, briefly explaining key concepts. In particular if the intended audience is ocean model developers, who may be familiar with many advanced computational and numerical techniques, but not with tools and terminology common in DSL approaches, some more guidance would be helpful."

Reply:

- We are aware that the terminology is ambiguous. We therefore added a new section on terminology to introduce all necessary terms, see Section 2. Our DSL adds another tool to the toolchain of research software engineers or model developers. However, they do not need to understand the building blocks of the DSL. Instead, they can focus on the declaration of parameters and configuration options (Declaration view), the specification of configurations and parameter selection (Configuration view). These two views separate concerns of these aspects for model setup, including additional checks to ensure working setups. This reduces mistakes and error. As the DSL allows to support a wide range of different output files for each ocean model, it allows to reduce the complexity for a user to comprehend all the different syntaxes and conventions, as only the syntax of the Configuration DSL is used.

Your comment:

- "Here it is also important to be aware of how terminology various between different communities and be specific about what definition is being used. As an example, the authors already point out that the concept of language models adds a new meaning to word model in the ocean modelling context, but within the ocean modelling community the word already has a range of meanings: a conceptual model of the global ocean, a description of its physics, a translation of that into a mathematical model, which in turn are translated into numerical equations whose specific software implementation is also referred to as an ocean model, and finally a specific configuration of such a model for a specific scenario is again referred to as an ocean model. In this paper the authors choose the (appropriate) definition of a specific software implementation (line 16), but then in line 140 we have "The Model Developer is a software developer and responsible for transferring the ocean models into code" which contradicts that definition and makes it hard to understand what the difference between a "Scientific Modeler" and a "Model Developer" is."

Reply:

- As the terminology was ambiguous in this context, we have expanded Section 4 and changed "model developer" to the more fitting term "Research Software Engineer" as this provides a better description of this role.

Your comment:

- "As another example, the authors make a distinction between configuration and parameterisation. Here it is important to note that "parameterisation" already has a very specific meaning in the ocean (and atmosphere) modelling community, it refers to processes that are not modelled through PDEs on a numerical grid, but rather through empirical parameterisation of these processes (typically on the sub-grid scale). "parameter selection" might be a better description of what is meant in the paper."

Reply:

- We also find the definition to be more plausible than the one stated previously, thus we replaced it with the proposed one, i.e., "parameter selection". Thanks for this suggestion.

Your comment:

- "Configuration is defined in line 37 as "the selection of features and code to beused for the model, as well as, the build configuration." but then on line 268: "For each simulation experiment, we need to define a configuration and a parametrization. Independent of a specific experiment, we declare settings that are specific to an ocean

model. Thus, the parameters and configurable features are declared with CP-DSL in a Declaration Model specific to each supported ocean model, as depicted in Figure 2. The Configuration Model is independent of a specific an ocean model, it defines the settings of a concrete experiment, whereby the declarations in the Declaration Model for the specific ocean model are referenced. This way, we separate the ocean-model-independent and the ocean-model-dependent settings." which seems to bring an entire new definition of configuration which is used along with the old in the same paragraph."

- Reply: Based on your review, we agree that the explanations seem to contradict themself, accordingly we rephrased these paragraphs. Also, we added a terminology section to the paper for further clarification. Also see our next reply.
  We added the following text: "For each simulation experiment, we need to define a configuration and a parameter selection. However, portions of experiment settings might be identical across several configurations, e.g., in parameter optimization or to test scientific model stability regarding certain parameter changes. These definitions are stored in a Configuration Model. Independent of a specific experiment, we need to declare which settings are genuine to a specific scientific model. This makes the CP-DSL agnostic to specific scientific models, as the declarations are stored in a Declaration Model. The Declaration Model declares the parameters and configurable features available for a specific scientific model, as depicted in Figure 3, and serve a similar purpose as declaring data types in a programming language."

Your comment:

- "As a final example, the word "deploy" and "deployment" is used in a number of places: "Model developers also deploy the software" (line 141), "the deployment is merely configuration and parametrization" (line 163) - and I don't really understand what is meant there - I'm more familiar with its usage as in line 207-214."

Reply:

- For a more clear explanation of these terms, we have included them in the new terminology Section 2.

Your comment:

- "As one of the key decisions in the design of DSLs is based around finding the right level of abstraction, it would be worth to extend this discussion to scientific models in general and explain why an ocean-modelling specific language is required, or whether it could be built on top of a more generic approach for the configuration of scientific models in general."

Reply:

- Since the starting point of our project are ocean models, we observed similarities with other models in Earth System Climate Models during the development of CP-DSL. Therefore, we see the basic consideration of supporting more scientific models, but find that the requirements of specific domains are too different. In addition, such a generalization would require many more examples at this time.

Your comment:

- "Already mentioned are Psyclone, Dusk/Dawn and Sprat which target other layers of the software stack, in particular PDE discretization in combination with automated code generation. In this context it might be worth mentioning the popular FEnics [1] and Firedrake [2], and DUNE [3] projects which make extensive use of such approaches (for context I'm one of the authors of Thetis [4] a coastal ocean model based on

Firedrake). You do mention ICON in the context of diagnostic configuration, but I believe there is more DSL-based ICON development described in [6]. Finally the Atmospheric Modelling Language (ATMOL) [7, 8] developed with the Royal Netherlands Meteorological Institute may be worth a mention."

Reply:

- So far, we considered recently used projects, but agree that ATMOL should be mentioned as one of the first DSLs for weather and climate modeling. Thus, we have expanded Section 3.1. Similarly, since the FEnics, Firedrake, and Dune frameworks also use DSL for high-level specification of model equations in scientific modeling, we have included them accordingly. As for the ICON DSL, we intend to emphasize the use of CDI-pio in the ICON Earth System Model. To clarify this, we replaced the reference with a more specific one. Nevertheless, we thank you for the valuable reference of ICON DSL. It is now cited in Section 3.1.

Your comment:

- "section 3.5: I don't really understand the difference being made between versions and variants, and how it is relevant to the rest of the paper"

Reply:

- For further clarification of these terms, we have included them in the terminology Section 2. We have also rewritten the paragraph to clarify the difference and the importance: "As a development tool, the DSL must fit into an existing toolchain of the domain. A major concern is documentation and, in this context, versions and variants of the model. The DSL design should take this into account since it has to comply with established version control systems."

Your comment:

- "why mention openmp but not MPI (the Message Passing Interface not Max Planck!)"

Reply:

- We agree that MPI should be mentioned here to be complete and have added it to Sections 3.1 and 4.6.

Your comment:

- "Code management and sharing via tar-balls, ssh and email is mostly deprecated and replaced by Git, often with the Git Large File Storage (Git LFS) extension." I think that's a little too strong; the use of ssh and shared file systems is still very common practice."

Reply:

- This is true, and although ssh and shared file systems are used, according to the experts interviewed, the advantages of Git outweigh these and are replaced accordingly where possible. Still, we rephrased the text to Section~5 to emphasize this fact: "Code management and sharing via tar-balls, ssh and email are used, but become less favorable among scientists who we interviewed."

Your comment:

- "what is meant by two layers?"

Reply:

- We agree that this needs more clarification and have extended the paragraph and added an explanatory figure: "The projects and developers use multiple repositories loosely following a repository hierarchy with the community repository at the top, followed by institute-wide repositories, research-group repositories and two layers of individual repositories labeled \emph{installation} and \emph{user} in Figure 2."

Thanks also for all the comments on typos, that we fixed in the paper.