WIP: Live Restructuring of Data Architecture

Walton Macey[1]
wmacey@utk.edu

Dali Wang[2]
dwang@ornl.gov

Peter Thornton[2]
thorntonpe@ornl.gov

Audris Mockus[1]
audris@utk.edu

[1]University of Tennessee, Knoxville
[2]Oak Ridge National Labs Oak Ridge, Tennessee

ABSTRACT

In large-scale Earth System simulation codes, such as the Accelerated Climate Model for Energy (ACME), complex user derived data types (containing large number of variables) are designed to represent the interactions of atmosphere, ocean, land, ice, and biosphere to project global climate under a wide variety of conditions.

The following is our proposed approach to restructure the data architecture of a land component within the ACME project while the project is undergoing active development. The data architect for the land subsystem defines the new datatype requirements that would greatly simplify the implementation of terrestrial land submodels by converting more than 50 to just eight primary data-types. Since the code is developed with the community governance, we have to ensure that the restructuring does not interface the other development which, with dozens of changes occurring every day, make it impossible to work on a shared development branch. The active development also occurs on almost five hundred branches, making it extremely difficult to assess potential interactions.

To address these challenges we have designed and started an iterative procedure for implementing the data restructuring and estimating both the effort it takes to restructure and the effort would save once the restructuring is implemented.

Keywords

Software Productivity, ACME, ALM, Data Refactoring

1. INTRODUCTION

The Accelerated Climate Model for Energy (ACME) has been designed to accelerate the development and application of fully coupled, state-of-the-science Earth system modeling, simulation and prediction for scientific and energy applications. With an early emphasis on improving software design and practice, development has included maintaining build, test, and performance tools for the relevant computer platforms, and providing rapid development and debugging capabilities to the team. The ACME code repository both expedites the merging and testing of the fully coupled system and supports a distributed development environment where separate features are being co-developed at different sites [1].

Inside ACME, the scientists/modelers have developed a terrestrial land model,called the ACME Land Model (ALM). ALM is a process-based model with a collection of key biogeophysical and biogeochemical functions that represent the energy-water-biogeochemical interactions between the atmosphere and the terrestrial landscape. ALM contains hundreds of energy, water and ecosystem variables which may be used or modified by many terrestrial ecosystem functions/modules. Designing data types to contain variables in a more modular fashion can facilitate scientific model development and unit testing [6].

In our previous work [3] we have described quantitative ways to evaluate such data modularity and a search algorithm that finds the optimal assignment of variables to modules. Based on the prior analysis and ongoing and future requirements of ALM development, the fundamental data types of ALM needs to be "remodularized".

2. BACKGROUND

Modularization of software systems is a well known and widely used concept in the field of Software Engineering [2, 4]. However, the discussion on software modularity tends to revolve around modularity of the source code [4]. In this paper we are concerned with the modularity of variables and data[1] used in software where very complex data structures are shared among many specialists, each focusing only on a small subset

[1] We use variables and data interchangeably in this paper.

SE4Science'17, 2016, Buenos Aires, Argentina

ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

of that data structure. A guiding principle for the new data structures has been to represent the primary functionality of the model as isolated modules connected through clearly defined interfaces.

In order to achieve this goal, it is worth considering why ALM simulations involve such complex data structures. To run very large simulations the data representing landscape surface is subdivided into a grid. That way, in each simulation step the information is exchanged only among neighbors of each grid cell, reducing the communication overhead on the supercomputers in which the simulation is being run and improving execution performance. Variables representing the state of the grid cell are collected into a single structure simplifying message passing. However, the human aspect of the scientist modeling a specific process within a cell, has not been previously considered to be as important and the proposed restructuring is aiming to address this.

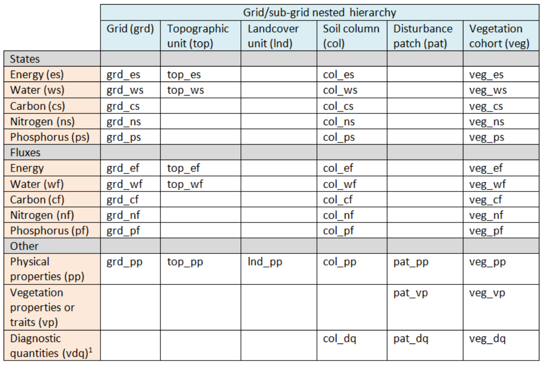
3. ROAD MAP

The critical part of the restructuring involves separating the task into small pieces that could be easily and rapidly implemented and tested. We are also starting from manually implementing changes and using the experience to design automation tools that would reduce the effort spent on repetitive tasks. In particular, our initial implementation would guide us which parts of the procedure could benefit from automation and whether the introduction of automation would reduce the overall effort.

The transition involves converting the current data structure of ALM.v0 with necessary modularity improvements, and develop a hybrid data structure for future ALM development (such as ALM.v1). In addition, we would like to evaluate the impact of ALM data structure on model development and software productivity.

The analysis of the current data structures revealed several types of variables that may require different types of treatment. At the higher level, we can classify the variables according to scientific domains they pertain, according to the part of the grid cell they represent, and according to their primary purpose being simulation or validation. In contrast to the previous implementations, the proposed architecture has separate types for variables at each level of the sub-grid hierarchy [5].

Our proposed data types would intrinsically define the relevant sub-grid level. In order to satisfy individual subroutines or model capabilities, these data types can be constructed as collections of pointers to the foundational types [5]. As new variables are added during development, the data structures should ensure these variables are introduced in a logical part of the subgrid hierarchy and should support energy balance constraints that are used to validate the simulation. For example, "a new soil column representation or a new vegetation cohort representation can be introduced and evaluated easily against existing module as long as the between-level interface variable naming and meaning is maintained" [5].



	Grid/sub-grid nested hierarchy					
	Grid (grd)	Topographic unit (top)	Landcover unit (lnd)	Soil column (col)	Disturbance patch (pat)	Vegetation cohort (veg)
States						
Energy (es)	grd_es	top_es		col_es		veg_es
Water (ws)	grd_ws	top_ws		col_ws		veg_ws
Carbon (cs)	grd_cs			col_cs		veg_cs
Nitrogen (ns)	grd_ns			col_ns		veg_ns
Phosphorus (ps)	grd_ps			col_ps		veg_ps
Fluxes						
Energy	grd_ef	top_ef		col_ef		veg_ef
Water (wf)	grd_wf	top_wf		col_wf		veg_wf
Carbon (cf)	grd_cf			col_cf		veg_cf
Nitrogen (nf)	grd_nf			col_nf		veg_nf
Phosphorus (pf)	grd_pf			col_pf		veg_pf
Other						
Physical properties (pp)	grd_pp	top_pp	lnd_pp	col_pp	pat_pp	veg_pp
Vegetation properties or traits (vp)					pat_vp	veg_vp
Diagnostic quantities (vdq)[1]				col_dq	pat_dq	veg_dq

Figure 1: Foundational data types for ALM.v1

Because ACME already has extensive automated test suites for the ongoing development of the model, we will utilize the existing testing procedure to verify that the overall model will be unchanged by our implementation of these new data structures. ACME uses an automated, regression test system for more efficient and robust development that allows the model to maintain a reliable and releasable state. We will be running the acme-developer test suite with each iteration, to instill a basic confidence that the set of changes will not break or change the model according to initial baselines.

The specific work breakdown of the data restructuring has the following steps:

- Manually convert one land datatype
- Systematic testing for code integrity
    - Single case testing (compile/execution)
    - Regression testing (acme developer test suite)
    - Bit-for-Bit output comparison
- Learn from analysis of manual implementation
    - How much effort and time did the process take
    - What parts can be automated or executed more efficiently
    - How best to improve productivity

The overall objective is to have new data structure for ALM.v1 within ACME.v1 (scheduled release date is July 2017) by getting the code into main trunk and summarize the lessons learned for the future restructuring.

4. SUMMARY

With our proposed restructuring of the data architecture, we expect sub-model implementation and modification to be much more straightforward for both scientists and engineers. To reduce risks to a project of this size undergoing active development, we have designed an iterative approach. For scaling up, we have incorporate steps to automate repetitive manual work and we rely on the existing testing infrastructure suites to verify the success of the restructuring.

5. REFERENCES

[1] D. Bader, W. Collins, R. Jacob, P. Jones, P. Rasch, M. Taylor, P. Thornton, and D. Williams. Acme project strategy and initial implementation plan. Technical report, Department of Energy, 2014.

[2] O.-J. Dahl, E. W. Dijkstra, and C. A. R. Hoare. *Structured programming*. Academic Press Ltd., 1972.

[3] Y. Ma, T. Dey, and A. Mockus. Modularizing global variable in climate simulation software: position paper. In *Proceedings of the International Workshop on Software Engineering for Science*, pages 8–11. ACM, 2016.

[4] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.

[5] P. Thornton. Data structures for acme land requirements. Technical report, 2016.

[6] D. Wang, Y. Xu, P. E. Thornton, A. W. King, C. A. Steed, L. Gu, and J. Schuchart. A functional test platform for the community land model. *Environmental Modelling and Software*, 55:25–31, 2014.