Environmental Modelling & Software 55 (2014) 25–31

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Short communication

A functional test platform for the Community Land Model

Dali Wang[a,*], Yang Xu[b], Peter Thornton[a], Anthony King[a], Chad Steed[a], Lianhong Gu[a], Joseph Schuchart[c]

[a] Climate Change Science Institute, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[b] Department of Geography, University of Tennessee, Knoxville, TN 37966, USA
[c] Joint Institute for Computational Sciences, University of Tennessee, Knoxville, TN 37966, USA

ARTICLE INFO

Article history:
Received 8 August 2013
Received in revised form
7 January 2014
Accepted 9 January 2014
Available online 1 February 2014

Keywords:
Community Earth System Model
Community Land Model
Functional test
Photosynthesis

ABSTRACT

The realistic representation of key biogeophysical and biogeochemical functions is the fundamental of process-based ecosystem models. A functional test platform is designed to create direct linkages between site measurements and the process-based ecosystem model within the Community Earth System Models (CESM). The platform consists of three major parts: 1) interactive user interfaces, 2) functional test models and 3) observational datasets. It provides much needed integration interfaces for both field experimentalists and ecosystem modelers to improve the model's representation of ecosystem processes within the CESM framework without large software overhead.

Published by Elsevier Ltd.

1. Introduction

Over the past several decades, researchers have made significant progress in developing high fidelity earth system models to advance our understanding on earth system, and to improve our capability of better projecting future scenarios (Washington and Parkinson, 2005). The Community Earth System Model (CESM, http://www2.cesm.ucar.edu) is one of the US leading earth system models. CESM is being actively developed to support Department of Energy's climate and environmental research. Within CESM, the Community Land Model (CLM) is the active component to simulate surface energy, water, carbon, and nitrogen fluxes and state variables for both vegetated and non-vegetated land surfaces (Bonan, 1998; Dickinson et al., 2006; Oleson et al., 2010). In order to minimize uncertainty, error and bias in the earth system simulations, it is vital to get the fundamental processes correct and to investigate new theories of ecosystem function and new process representations within the context of earth system behavior. However, the complexity of the current CESM framework (both conceptual design and software implementation) makes function-level testing and exploration very difficult, especially at scales and levels of organization below that of the landscape and whole-ecosystem where many relevant field measurements are made.

The realistic representation of key biogeophysical and biogeochemical functions is the fundamental of process-based ecosystem models. In this paper, we present our approach to create direct linkages between site measurements and the process-based terrestrial ecosystem model (CLM). A functional test platform is designed to eliminate the majority of software complexity to allow scientists to interactively select external forcing, manipulate Plant Functional Type (PFT)-specific ecophysiological parameters and compare the key ecosystem functional representations with measurements and observations. It also preserves the maximum portion of code segment related to those key ecosystem functions. We believe that our experience in the design of the functional testing platform for the CLM can be beneficial to many other research programs which adapt the integrated environmental modeling methodology (Estreguil et al., 2014; Laniak et al., 2013).

2. The software system of the Community Land Model

Within the CESM framework, the CLM is designed to understand how natural and human changes in ecosystems affect climate. The model represents several aspects of the land surface including surface heterogeneity and consists of submodels related to land biogeophysics, the hydrologic cycle, biogeochemistry, human dimensions, and ecosystem dynamics.

* Corresponding author. Tel.: +1 8652418679.
E-mail addresses: wangd@ornl.gov (D. Wang), yxu30@utk.edu (Y. Xu), thorntonpe@ornl.gov (P. Thornton), kingaw@ornl.gov (A. King), steedca@ornl.gov (C. Steed), Lianhong-gu@ornl.gov (L. Gu), joseph.schuchart@zih.tu-dresden.de (J. Schuchart).

1364-8152/$ – see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.envsoft.2014.01.015

The software system of the global offline CLM includes two groups: models and scripts. The model group includes physical earth system components, such as the CLM, data atmosphere, stub ocean, stub ice and stub glacier. It contains a driver to configure the parallel computing environment and a coupled simulation system (physical earth system components and flux mapping functions between those components). It also includes several shared software modules and utilities, such as a flux coupler, a parallel Input/Output (IO) library, performance profiling libraries. The schematic diagram of the offline CLM software structure is shown in Fig. 1, which demonstrates that the CLM has to be incorporated with atmosphere and coupler as well as parallel IO, etc.

The whole CLM modeling system consists of more than 1800 source files and over 350,000 lines of source code. Fig. 2 shows the CLM software call tree using Yifan-Hu algorithm for graph layout. Each circle represents an individual subroutine with the area of circle showing the time spent on the subroutine. The directed edges show the procedure of software subroutine calls. The width of each edge indicates the number of subroutine invocations. The upper part is generally the software overhead of CLM, such as the IO, parallel communication, time management, and non-land earth system models setups. The lower part represents the CLM submodels, including several biogeophysical and biogeochemical models. More information on CLM computational characteristics can be found in another paper (Domke and Wang, 2012). Landscape surface is the basic data structure for CLM model development and software design.

Fig. 3 shows the hierarchical data structure for CLM landscape surface. Within this structure, lower level data arrays (e.g. pft-level data arrays) are accessible via references from the higher level data layers. Inside CLM, the land surface is represented by five primary landcover types: glacier, lake, wetland, urban, and vegetated portion. The vegetated portion of a gridcell is further divided into patches of PFTs, each with its own leaf and stem area index and canopy height. This hierarchical data structure makes the CLM initialization very complicated, since the CLM contains over 400 variables across multiple data layers over more than half million landscape surface gridcells.

3. Functional test platform design

In the CLM, PFT is the basic modeling concept. In our project, each biogeophysical or biogeochemical function of PFT is treated as a functional element. The purpose of functional testing is to provide direct comparison between model function and site experimental measurements at each functional element level. As shown in Fig. 3, the CLM landscape surface is represented by a hierarchical data structure, which make the model initialization complicated. In our platform, a single gridcell model is implemented, which consists of all the possible PFTs in the current version of CLM, and keeps the input and output interface of the functional element (e.g., key subroutine or module for ecosystem functions) unchanged, except the method to access global data arrays. Specifically, in our platform, a simplified global data structure is implemented to allocate memory space to host only necessary data for the testing of each individual functional element on a single gridcell landscape.

Fig. 4 shows the key testing data structure, which make the model initialization much simpler. Every data arrays can be accessible directly without hierarchical references. It eliminates the hierarchical data structure from the original CLM model, and the dimension of PFT-level data array is fixed at the maximum number of PFTs (that is 25) within the current version of CLM in order to represent the heterogeneity of each vegetated landscape surface. The functional test platform consists of three major parts: 1) interactive user interfaces, 2) standalone test models for functional elements and 3) query database for observational datasets. Based on those designs, our platform provides intuitive ways to enable direct model verification and model-data validation at individual functional element level.

3.1. Interactive user interfaces

The main function for those interfaces is to provide intuitive ways to setup computational experiments for functional testing, observational database establishment, and model-data comparison. Considering visualization capabilities, cross-platform compatibility, and future web-based presentation, our interfaces are implemented in Java using Java-based open source visualization libraries.

3.2. Standalone test model for functional element

The standalone test model for each key functional element consists of five key parts: 1) a generic functional test driver, which is designed to configure the functional test computing environment, configure a single gridcell model, and initialize physiological parameters and external forcing; 2) a generic functional test data structure, replacing the hierarchical CLM data structure, for a single gridcell model; 3) a initialization and output function for the single gridcell model; 4) a target functional element (e.g., "stomata" function in the following case study); 5) as well as other shared CLM data definitions (e.g. physical and chemical constants) which are used to eliminate unnecessary software structure changes to the target functional element. Users can either keep the original functional element unchanged, or modify the mathematical formula inside the functional element based on their own research. After that users can generate a standalone executable for the target functional element.

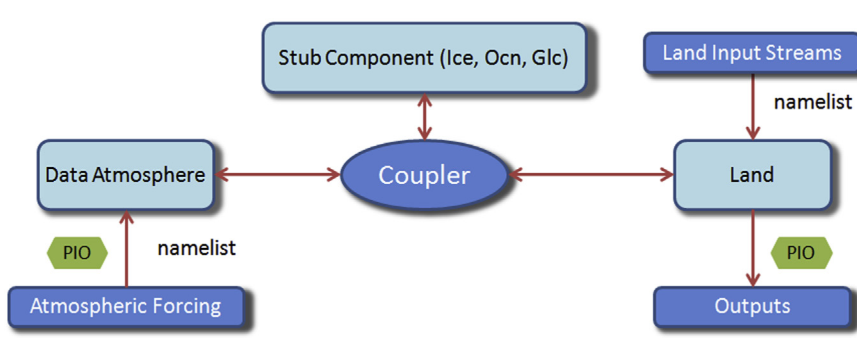

Fig. 1. Software configuration of a global offline CLM simulation. Several earth system model components are listed, including a land model (Land), a data atmospheric model (Data Atmosphere), stub sea ice model (Ice), ocean model (Ocn) and glacier model (Glc). PIO stands for Parallel IO.