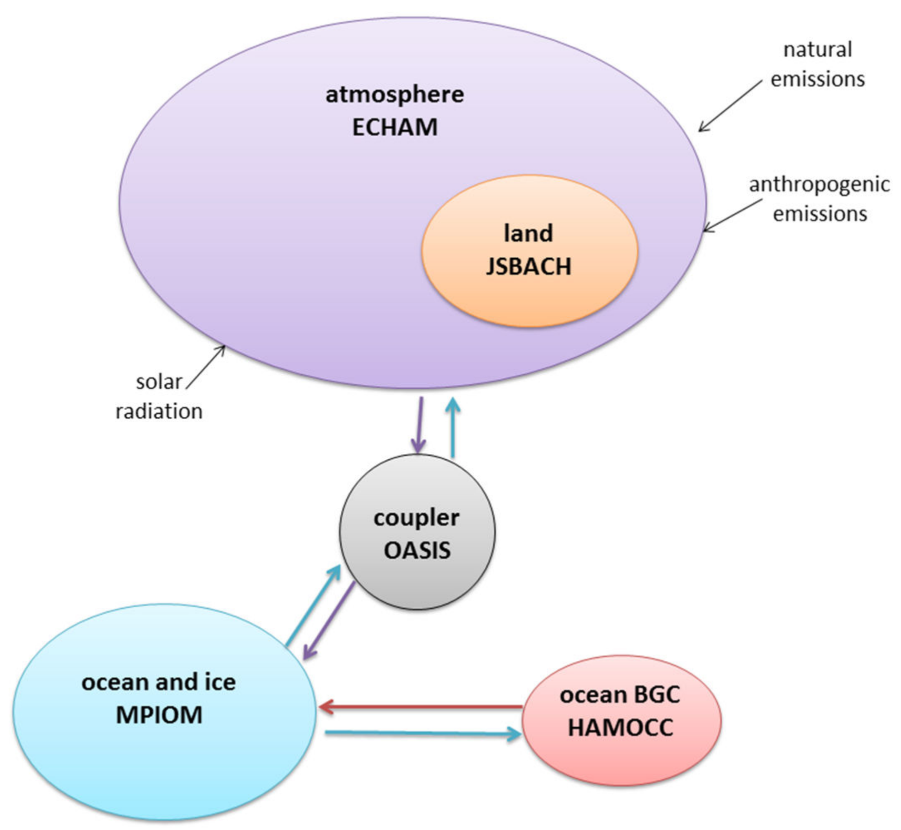
The Software Architecture of Global Climate Models

Kaitlin Alexander, Steve Easterbrook

Software Engineering Lab, Department of Computer Science, University of Toronto

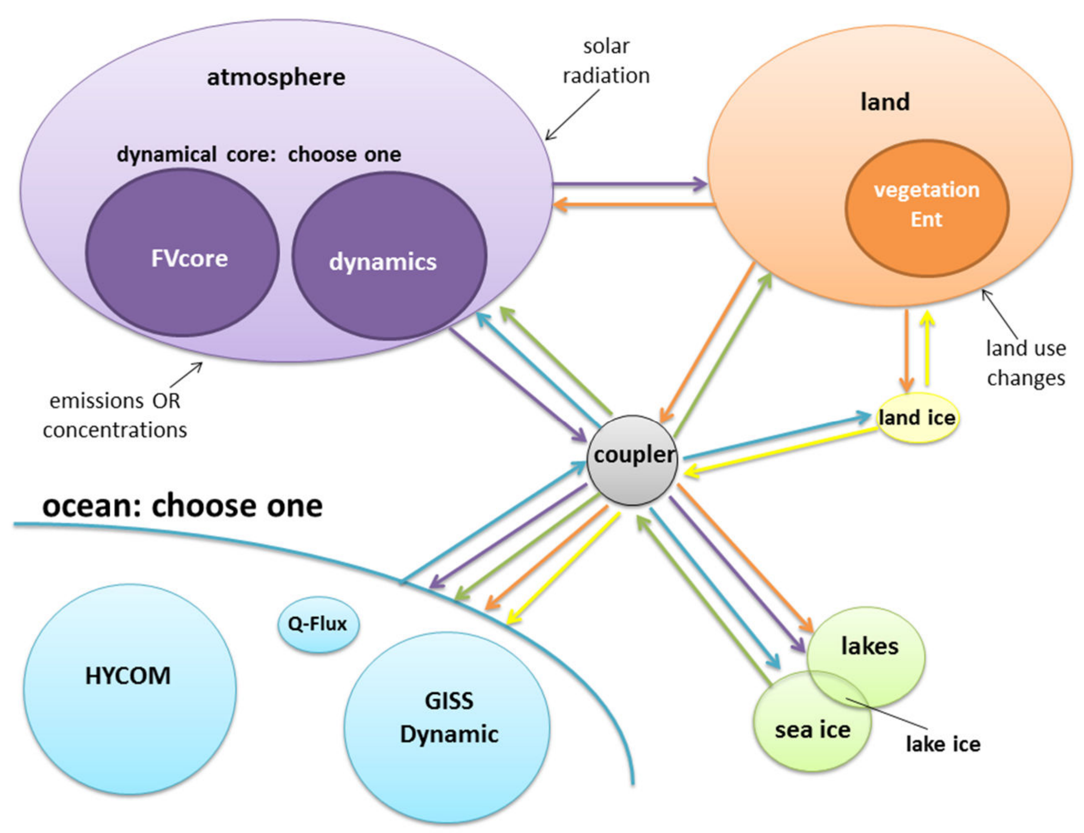
COSMOS 1.2.1

Max-Planck-Institut für Meteorologie, Germany



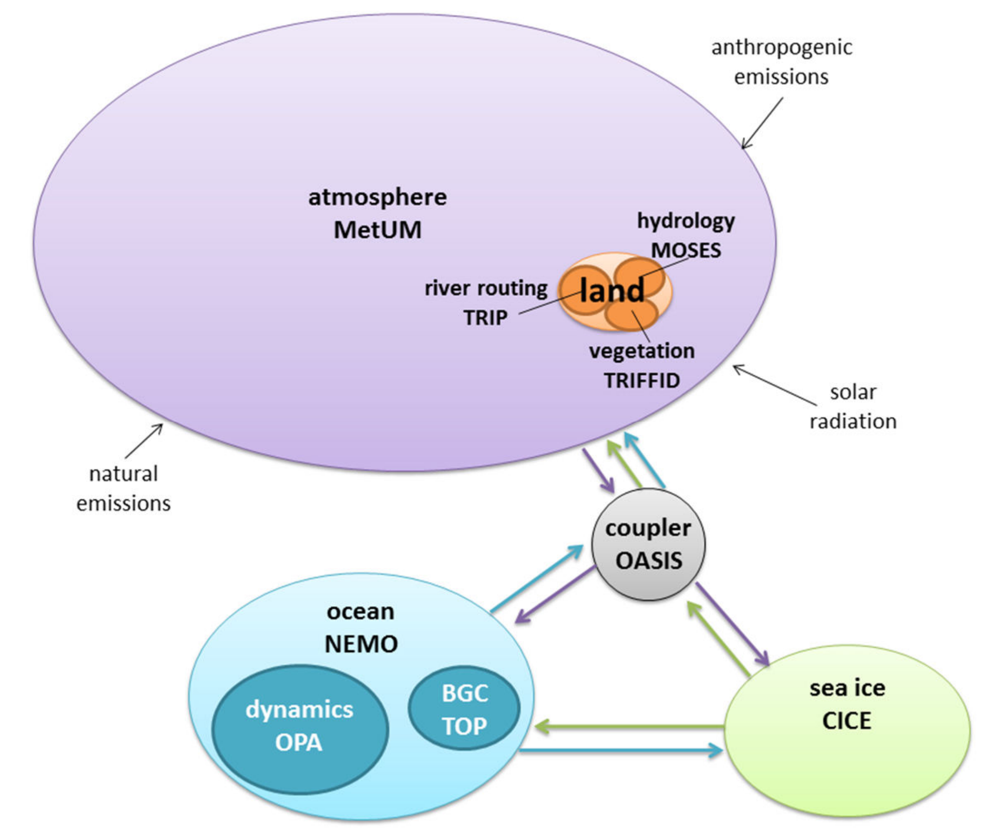
Model E June 17, 2011 revision

NASA Goddard Institute for Space Studies, USA



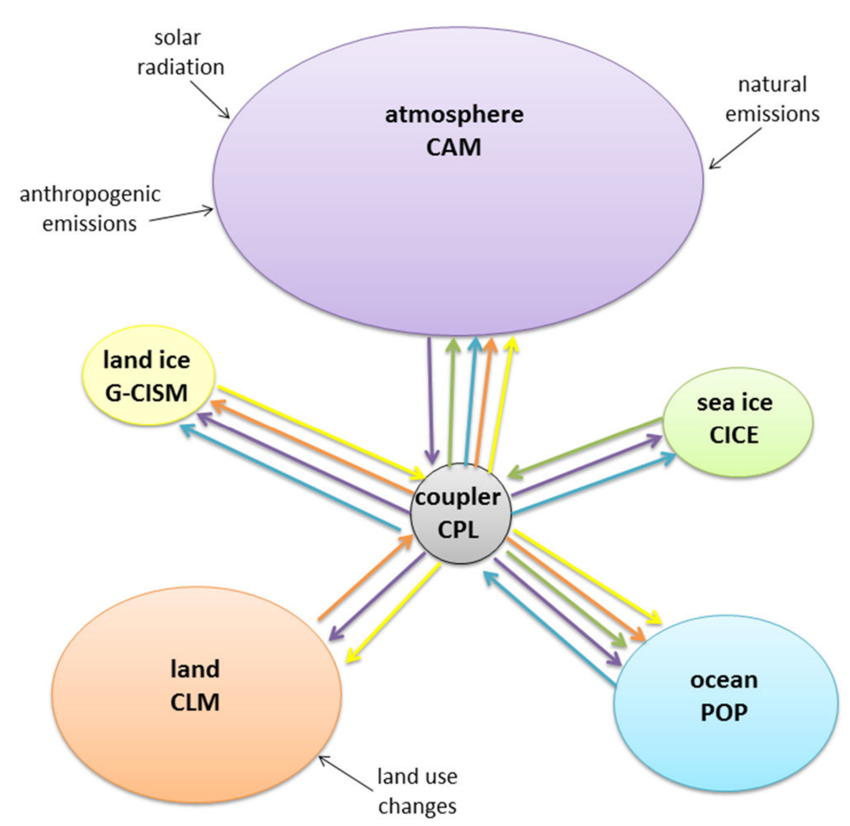
HadGEM3 August 3, 2009 revision

Met Office, UK



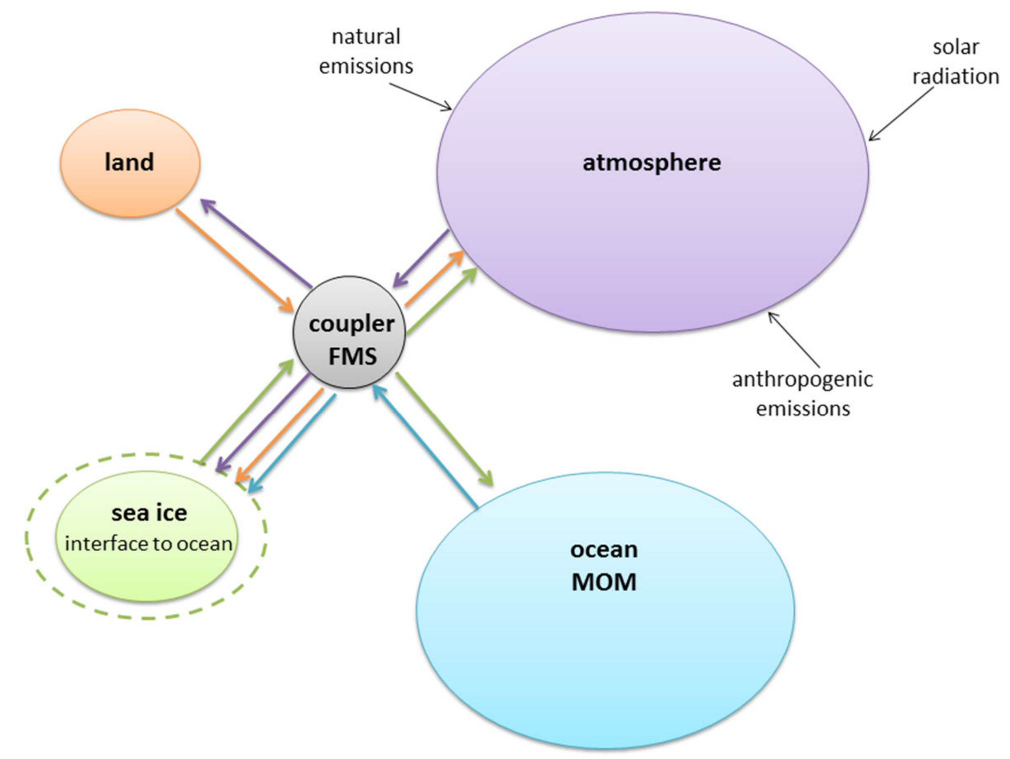
CESM 1.0.3

National Center for Atmospheric Research, USA



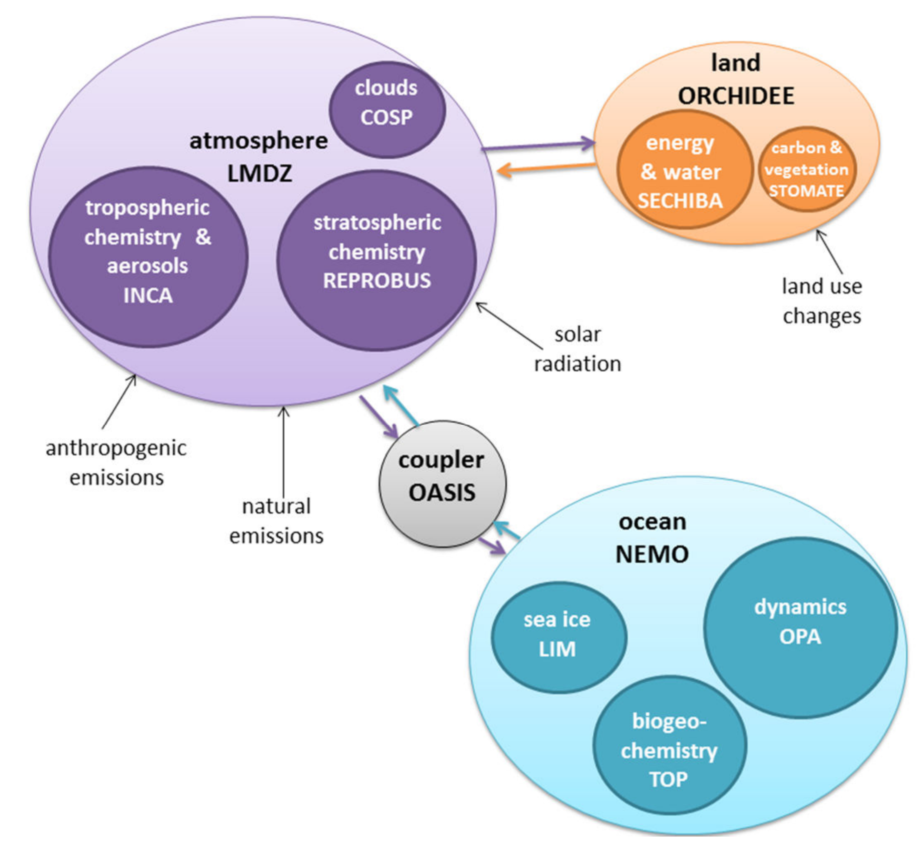
GFDL Climate Model 2.1 (coupled to MOM 4.1)

Geophysical Fluid Dynamics Laboratory, USA



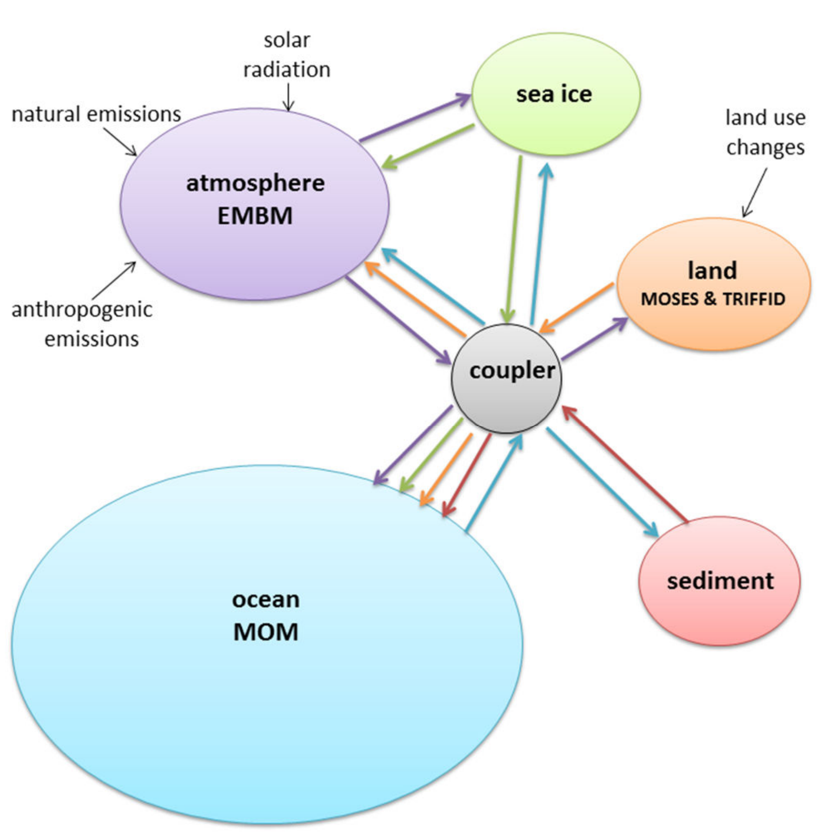
IPSL Climate Model 5A

Institut Pierre Simon Laplace, France



UVic Earth System Climate Model 2.9

University of Victoria, Canada



Key to Diagrams

Each component of the climate system has been assigned a colour:

atmosphere ocean land sea ice land ice sediment

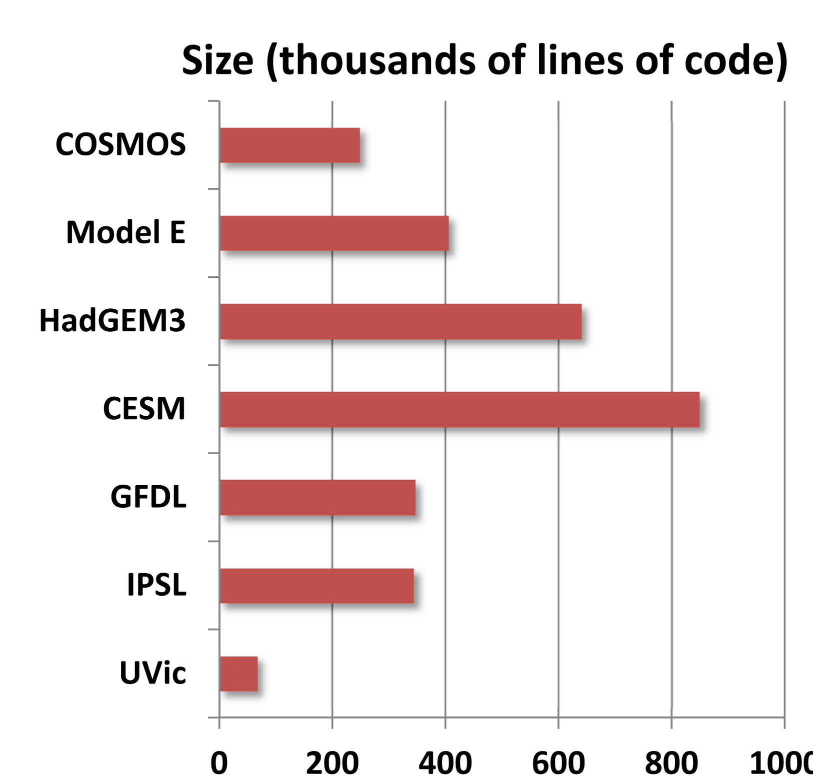
Model code for a component is represented with a bubble. Fluxes are represented with arrows, in a colour showing where they originated.

Couplers are grey. Components can pass fluxes either directly to each other or through the coupler.

The area of a bubble represents the size of its code base, relative to other components in the same model.

A smaller bubble within a larger one represents a small, highly encapsulated model of a system (eg, clouds) that is used by the component.

Radiative forcings are passed to components with plain arrows.



Generated using David A. Wheeler's "SLOCCount".

What is a Climate Model?

We don't have access to multiple Earths for the purpose of experimentation. Instead, scientists have developed global climate models (GCMs): large pieces of software that simulate the climate system, and how it might react to agents of climate change. In this study, we compared and contrasted the software architecture of seven GCMs from Canada, the United States, and Europe.

Common Features

Infrastructure code (shell scripts and Perl) configures, builds, and runs the model. **Scientific code** (FORTRAN and some C) consists of calculations to simulate the climate system.

Cells (3D, ~100 km wide) are created by laying a grid over the Earth's surface and atmosphere. **Time steps** indicate how often calculations are performed (typically minutes to hours).

Dynamics calculations resolve fluid dynamics from first principles. **Physics** calculations are parameterizations: approximations for complex or small-scale processes.

Component-Based Software Engineering (CBSE)

A climate model is really a *collection* of models (components) for the atmosphere, land, etc. They are highly encapsulated, for stand-alone use as well as a mix-and-match approach that facilitates code sharing. CBSE pools resources, creating high-quality components that are used by many GCMs.

Components are modified when they are passed between institutions, to suit new GCMs. These modifications are encouraged by code sharing practices. Virtually anyone can get access to GCM source code, but only the core development team can modify the master copy.

A drawback of CBSE is the fact that the real world is not encapsulated. Relationships between sea ice and the ocean are particularly difficult to represent. Here are some examples of the different approaches taken:

- **CESM**: Sea ice is separate to the ocean, with a transient boundary.
- **IPSL:** Sea ice is a sub-component of the ocean.
- **GFDL:** Sea ice is an interface to the ocean. All fluxes to and from the ocean pass through "sea ice", even if no ice is actually present.

The Coupling Process

Couplers manage component interaction and fluxes (mass/energy transfers). Components often use different grids or resolutions, so direct interaction is difficult.

The **main loop** is present in the top level-program for the coupler. Each iteration, it calls components one by one, instructing them to complete their calculations for one time step and update fluxes.

- **CESM:** All interactions are managed by the coupler.
- **IPSL:** Only the atmosphere and the ocean are connected to the coupler. The land component is directly called by the atmosphere.
- **HadGEM3**: All components are connected to the coupler, but the ocean and sea ice have similar grids, so they interact directly.

Complexity and Focus

A line count of source code is a good proxy for complexity. Between the models, complexity varies widely. Within models, the bulk of a GCM's complexity is often concentrated in one component, due to the origin of the model and the institution's goals.

- **HadGEM3:** atmosphere-centric. It began as an atmospheric model, and is also used for weather forecasting, which requires high atmospheric complexity.
- **UVic**: ocean-centric. It began as an ocean model, but kept the combination of a complex ocean and a simple atmosphere due to its speed and suitability to very long simulations.
- **CESM:** atmosphere-centric, but land is catching up. It is embracing the "Earth System Model" frontier of terrestrial complexity, particularly feedbacks in the carbon cycle.

Conclusions

While some features of software architecture are common among every GCM we studied, other features show a wide range of different design choices. Coupler structure, distribution of complexity, and levels of component encapsulation all vary widely.

These architectural differences may provide new insights into variability and spread between model results. By examining software variations, as well as scientific variations, we can better understand discrepancies in GCM output.