

An Overview of the Earth System Modeling Framework

V. Balaji

SGI/GFDL Princeton University

Computing in the Atmospheric Sciences

Annecy, 9 September 2003

Talk overview

- Why frameworks have become hot: science
- Programming models for underlying hardware
- Architectural hierarchy: memory models, grids, components
- The ESMF project

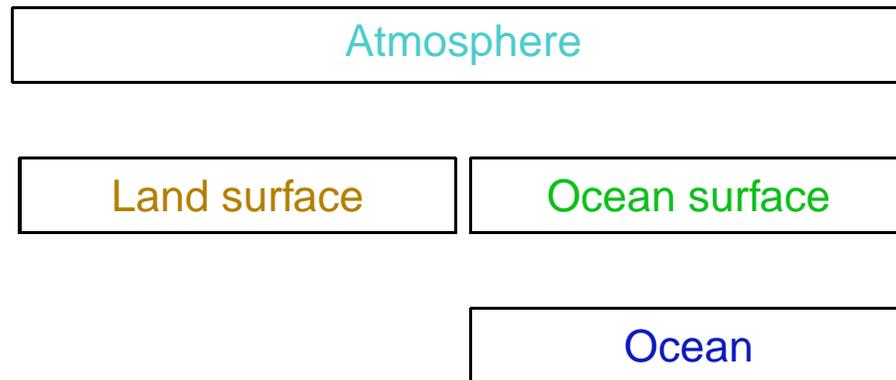
Components of the climate system

Atmosphere atmospheric fluid dynamics and thermodynamics, moist processes, radiative transfer, transport and chemistry of trace constituents.

Ocean World ocean circulation, ocean biogeochemistry.

Land surface Surface processes, ecosystems, hydrology.

Ocean surface Sea ice, wave processes.



Complexity of climate simulations

Models have grown increasingly complex with time.

70s	80s	early 90s	late 90s	today	early 00s	late 00s
Atm	Atm	Atm	Atm	Atm	Atm	Atm
	Land	Land	Land	Land	Land	Land
		Ocn, Sealce				
			Aerosols	Aerosols	Aerosols	Aerosols
					C Cycle	C Cycle
		Aerosols			Ecosystems	Ecosystems
		Land C			Chemistry	Chemistry
	Ocn, Sealce	Ocn Carbon			Ocn Eddies	Ocn Eddies
	Clouds	Chemistry	C Cycle	Ocn Eddies		Clouds

Components are developed “offline” (bottom left) and then are integrated into comprehensive coupled models.

Timescales and space scales

Clouds minutes to hours; 10 m – 100 km.

Ocean eddies hours to days; 10 km.

Weather patterns days; continent scales.

Ocean currents days; ocean basin scales.

Short-term climate seasonal-interannual; El Niño; monsoons.

Climate change Decadal-centennial; ocean overturning and deep water formation.

Paleoclimate Millennia to deep time; ice ages; orbital changes.

Discretization in space and time

Each process has its own intrinsic time and space scales.

Older models did not allow subcomponents to be on independent grids and timesteps.

- Old way: sharing of data through arrays in common blocks.
- New way: independent model grids connected by a coupler.

Each physical *process* component becomes an independent *code* component that can be separately instantiated, initialized, stepped forward, and terminated.

Gridded and ungridded components

Components may be associated with a grid (gridded component), associated with more than one component (coupler), or none at all (local computations, e.g chemistry).

Each component has its own data dependencies, decomposition strategies and coupling requirements. It must be possible to take full advantage of these features in a parallel code.

Types of grids

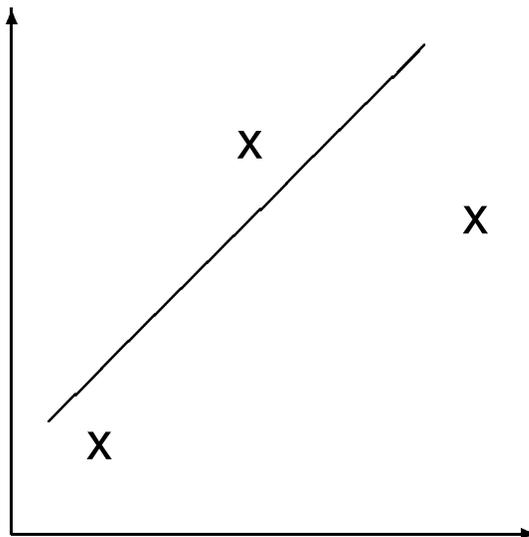
- Spectral grids.
- Lat-lon grids.
- Logically rectangular grids in generalized curvilinear coordinates:
 - Displaced pole
 - Tripolar grids
 - Cubed sphere
- Reduced grids.
- Icosahedral and geodesic grids.
- Unstructured grids; catchment grids.

Data assimilation

Data can appear at unpredictable locations in time and space (radiosondes, buoys, satellites) and have no clear radius of influence (“location streams”).

Models will “slosh” if incompatible with data.

Data assimilation involves bringing models and data into acquiescence. Assimilation algorithms can be treated as gridded components.



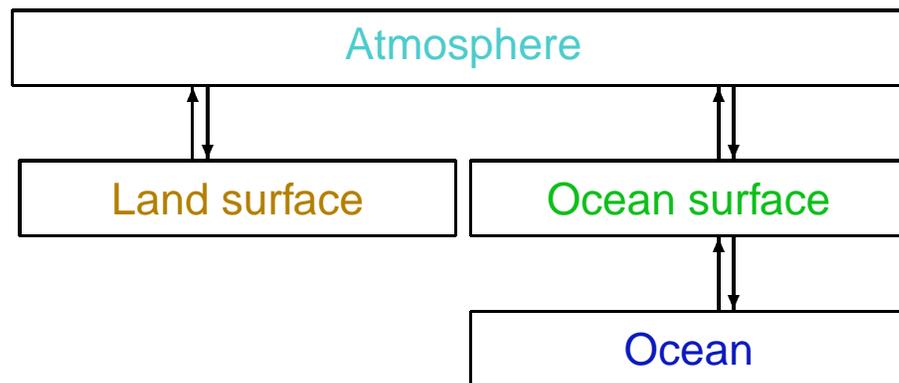
Model sizes and times

Climate Typically measured in simYears/wallDay. Resolution is adjusted till an experiment runs within a time useful for research. Current high-end research: $\sim 10^2$ variables integrated for $\sim 10^6$ timesteps on $\sim 10^6$ gridpoints. (Earth Simulator considerably raises this bar, but there is currently no relevant experience to compare against).

Weather Operational forecasting is measured in simHours/wallHour. Resolutions about 4×4 higher; $\sim 10^3$ timesteps.

Data exchange

Data exchanges between components take the form of fluxes of heat, momentum and tracers.



Conservation is very important for climate models; less so for weather.

Memory models

Shared memory signal parallel and critical regions, private and shared variables. Canonical architecture: UMA, limited scalability.

Distributed memory domain decomposition, local caches of remote data (“halos”), copy data to/from remote memory (“message passing”). Canonical architecture: NUMA, scalable at cost of code complexity.

Distributed shared memory or ccNUMA message-passing, shared memory or remote memory access (RMA) semantics. Processor-to-memory distance varies across address space, must be taken into account in coding for performance. Canonical architecture: cluster of SMPs. Scalable at large cost in code complexity.

Uniform interface to memory models: DEs and Layouts

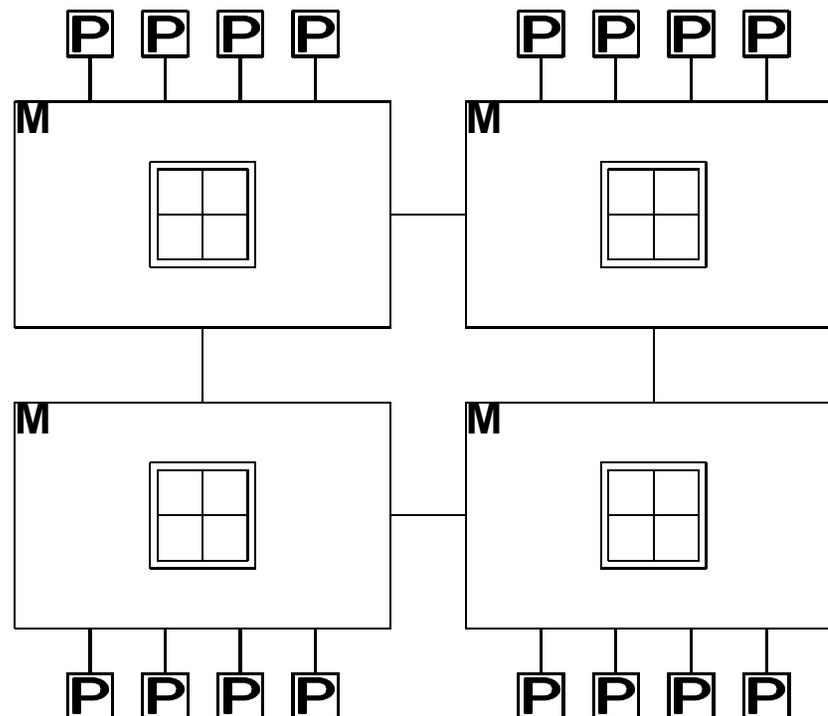
Decomposition Element A DE is the logical atom of a distributed object, which may or may not map onto a single PE.

Layout A layout describes how DEs are organized along some logical dimensions. We may choose to map these dimensions on to shared and distributed memory.

Machine model A machine model describes the hardware and applicable memory semantics: a description of the architecture in terms of **aNodes** and **mNodes**.

Uniform interface to the grid index space: the `distGrid`

A `distGrid` describes how to distribute the index space of a component across a DE layout, and most importantly, the *data dependencies* between DEs.

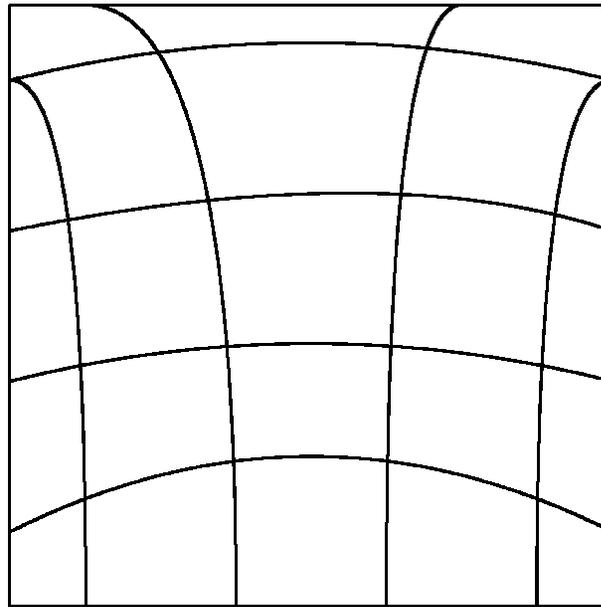


Halo updates and redistributions are typical operations performed using `distGrids`.

Overlay physical information on the `distGrid`:

`physGrid`

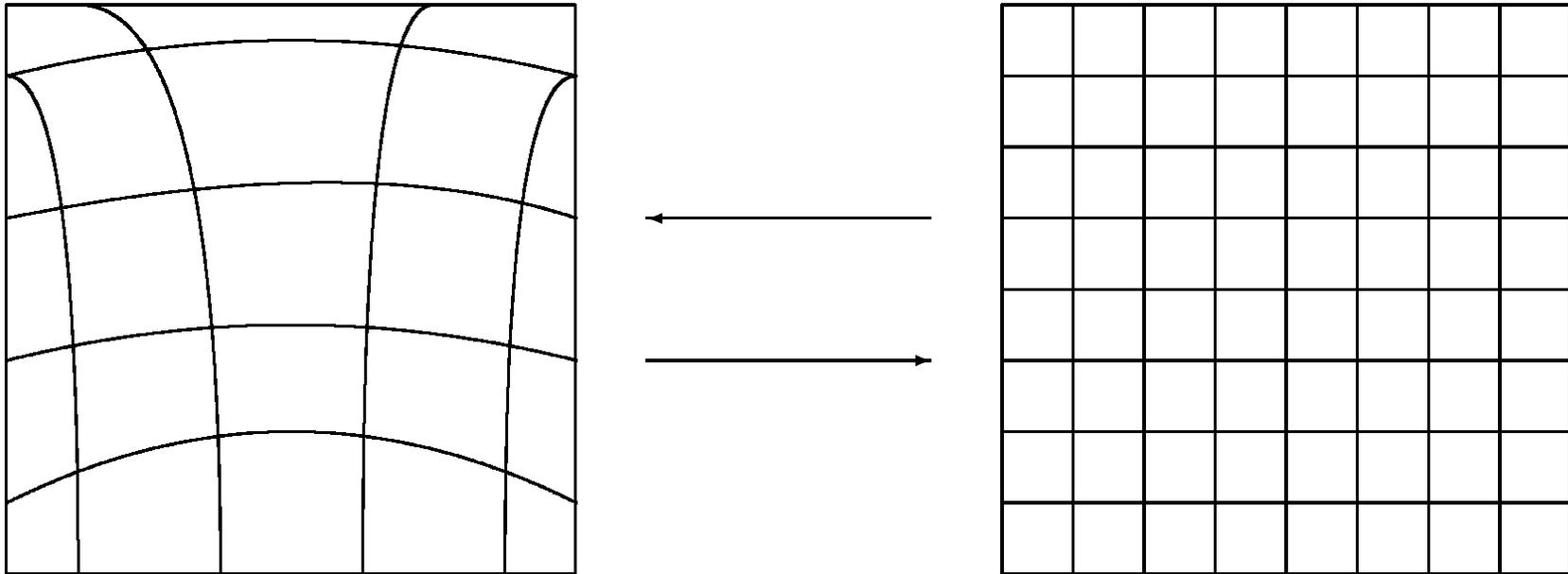
All the model numerics require knowledge of the physical locations, orientations, and interrelations of array indices.



Differential operators (gradient, divergence) are typical operations requiring **physGrids**.

Interpolate data between grids: reGrid

When components on independent **physGrids** must exchange data, they require a **reGrid**.



On parallel systems, we require all calls to be local; the **reGrid** object contains the optimized **routes** for communication.

Fields and bundles

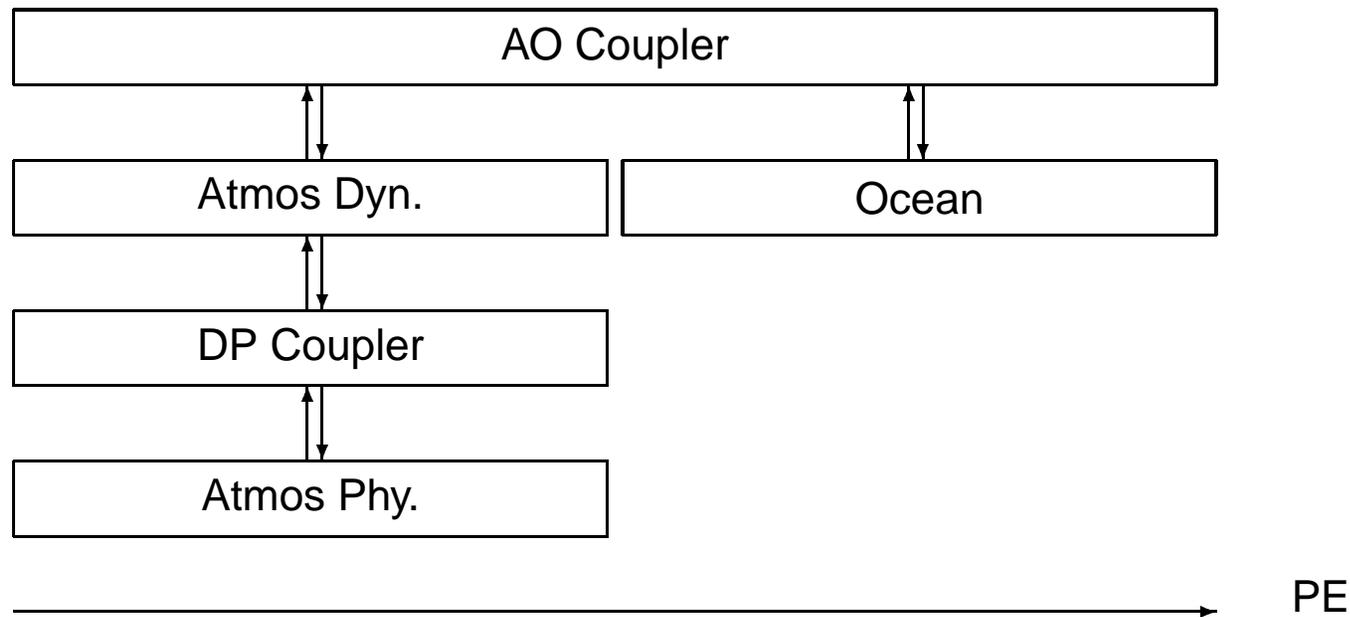
Alongside the grid information, each model variable is present as a distributed array. It is useful, indeed essential, to associate **metadata** that describe the physical content (name, units, grid locations, range of valid values) of the array. All this information together comprises a model **Field**.

Fields that share a grid may be treated collectively for certain grid operations. A collection of fields sharing a grid is called a **bundle**.

Model composition

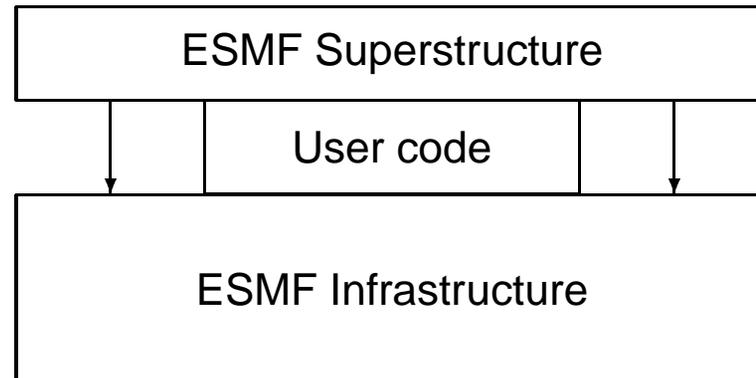
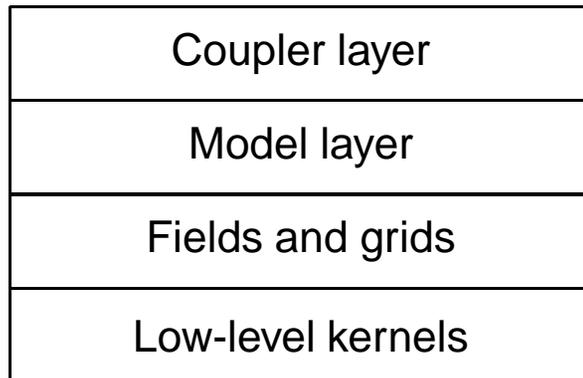
We develop a hierarchy of components and assign them to PEs as appropriate.

A **coupler** mediates between pairs of components needing to exchange data, and matches fields in their **import** and **export States**.



The coupler runs on the union of PEs of its components.

Architecture of an Earth System Modeling Framework: the sandwich

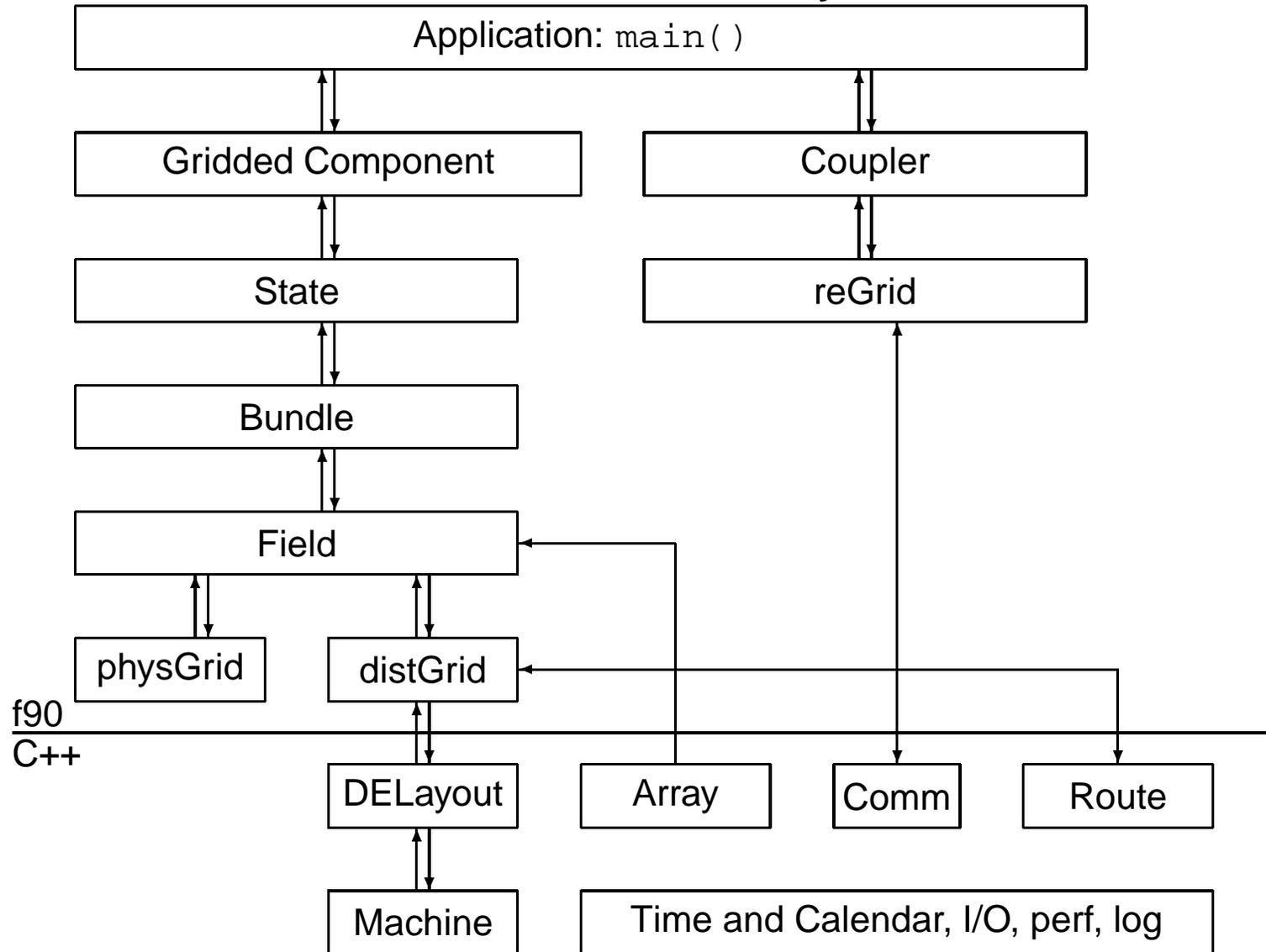


ESMF

The Earth System Modeling Framework is an end-to-end solution for the problem outlined here: supporting distributed development of models with many interacting components, with independent space and time discretization, running on complex modern scalable architectures. A capsule history of ESMF:

- The need to unify and extend current frameworks achieves wide currency (c. 1998).
- NASA offers to underwrite the development of an open community framework (1999).
- A broad cross-section of the community meets and agrees to develop a concerted response to NASA uniting coupled models, weather, and data assimilation in a common framework (August 1999). Participants include NASA/GMAO, NOAA/GFDL, NOAA/NCEP, NCAR, DOE, and universities with major models.
- Funding began February 2002: \$10 M over 3 years.
- First Community Meeting, Washington, May 2002: requirements review.
- Second Community Meeting, Princeton, May 2003: design review.
- Third Community Meeting, May 2004: prototype release.

ESMF Class Hierarchy



ESMF features

- ESMF is usable by models written in f90/C++.
- ESMF is usable by models requiring differentiability.
- ESMF is usable by models using shared or distributed memory parallelism semantics.
- ESMF supports serial and concurrent coupling.
- ESMF supports multiple I/O formats (including GRIB/BUFR, netCDF, HDF, native binary).
- ESMF has uniform syntax across platforms.
- ESMF runs on many platforms spanning desktops (laptops, even!) to supercomputers.

Language interoperability

Models increasingly are built to use high-level abstract language features (f90, C++) to facilitate development process across large teams. ESMF is written to be usable by models written in both languages.

Incompatibilities:

- Languages use entirely different memory semantics. ESMF distinguishes between “deep” and “shallow” objects: depending on whether memory is managed by callee or caller.
- High-level data structures do not cross language boundaries. ESMF implements cross-language data structures by implementing an intermediate layer in a low-level language (f77, C) where intrinsic datatypes can be cross-matched. This includes a unique implementation allowing f90 array pointers to be manipulated in either language.

The ESMF **Implementation Report** presents details of its cross-language technology.

Link to data frameworks

Community data frameworks are under development. For model output data to be scientifically useful, the researcher must have some knowledge of how the data was produced. Model data requires a *model's eye view* description of the data, another layer of metadata, which includes:

- Description of model components: e.g FMS BGRID atmosphere, land and sea ice coupled to MITgcm ocean.
- Description of grid configurations and resolutions.
- Choice of physics packages and input parameters.
- Model state and its fields.

ESMF and PRISM are emerging standards that allow the development of the model metadata layer, based on the state data structures and its base classes. Modeling framework data structures map directly on to community hierarchical metadata.

Summary

- The Earth System Modeling Framework supports distributed development of models with many interacting components, with independent space and time discretization, running on complex modern scalable architectures.
- Prototype release scheduled for May 2004.
- Future maintenance guarantee from NCAR: seeking funding for further development.
- Future directions: link to community data portals, enable new scientific applications, runtime environment.

Selected web references

<http://www.esmf.ucar.edu> General website for ESMF: documentation, code, examples, contacts.

<http://prism.enes.org> General website for PRISM: PRogram for Integrated Earth System Modeling.

<http://www.gfdl.gov/~fms> The GFDL Flexible Modeling System. Also links production models and climate model simulation data.

<http://nsipp.gsfc.nasa.gov> Focus on short-term climate variability.

<http://mitgcm.org> The MIT GCM. Considerable emphasis on data assimilation.

<http://www.wrf-model.org> The NCAR/NOAA Weather Research and Forecasting model.

<http://www.cesm.ucar.edu> The NCAR Community Climate System Model.

<http://www.ipcc.ch> Intergovernmental Panel on Climate Change. Comprehensive scientific synthesis of current thinking on climate.