What is CIME?

CIME (Common Infrastructure for Modeling the Earth):

CIME contains the support scripts, data models, essential utility libraries, a “main” and other tools that are needed to build a single-executable coupled Earth System Model.

Blue text is used to indicate some of the proposed future evolution of CIME.

What is CIME comprised of?:

CIME is comprised of:

1. A default coupled model architecture:
    a. A programmer interface and libraries (including MCT) to implement a hub-and-spoke inter-component coupling architecture.
    b. An implementation of a “hub” that needs 7 components (atm, ocn, lnd, sea-ice, land-ice, river, wave). a.k.a. “the driver”.
    c. The ability to allow prognostic and data components to be mixed in any combination as long as each component implements the coupling programmer interface.
2. Non-prognostic Data and Stub components
    . “Data-only” versions of 6 of the 7 components that can replace prognostic components at build-time.
    a. “Stub” versions of all 7 components for building a complete system.
3. A system of scripts (perl, csh and python) to support case configuration, executable compilation, workflow, system testing and unit testing infrastructure:
    . Scripts to enable simple generation of model executables and associated input files for different scientific cases, component resolutions and combinations of full, data and stub components with a handful of commands.
    a. Testing utilities to run defined system tests and report results for different configurations of the coupled system.
4. Additional stand-alone utilities and libraries useful in scientific applications in general and climate models in particular.
    . Parallel I/O library
    a. Timing library
    b. Stand-alone programs
        i. Parallel regridding weight generation program
        ii. Scripts to automate off-line load-balancing.
        iii. Scripts to conduct ensemble-based statistical consistency tests.
        iv. Netcdf file comparison program (for bit-for-bit).

CIME is available in a stand-alone package that can be compiled and tested without full prognostic components (by using the included data models)

CIME does not contain any prognostic components. In addition, CIME does not contain any intra-component coupling capability (such as CAM physics-dynamics coupling)

Where does ESMF/NUOPC come in?

ESMF/NUOPC only provide an alternative to item 1 in the above list. The current hub-and spoke inter-component coupling architecture (driver and mediator/coupler) is implemented with the MCT data structures and communication methods. ESMF/NUOPC could provide an alternative implementation of this driver and mediator infrastructure. ESMF/NUOPC does not provide a unified command-line build, test, configure, and workflow system for a fully coupled model, does not provide its own I/O library or data/stub versions of models and does not provide versions of the utility programs in 4c (except for the parallel regridding program which is implemented with ESMF).

1. Details of the CIME Coupling Architecture

`cime/driver_cpl,  cime/externals/mct`

CIME allows inter-component coupling capability through a hub-and-spoke architecture for models run as a single executable. In this architecture, the components are the spokes and the central hub is a coupler (or “mediator” in ESMF/NUOPC parlance). Implicit in this architecture is a top-level driver that controls the temporal evolution of the model system.. The coupler/mediator performs mapping between components, merging, flux calculations, and diagnostics. Currently, CIME provides the ability to couple seven component models (atmosphere, land, river, land-ice, sea-ice, ocean, surface wave).

In general, there are three versions of each component model; a prognostic model external to CIME; an internal “data model” that reads in observational or model generated data; or an internal “stub model” that satisfies interface requirements. The CIME coupling architecture is independent of the option chosen for each component, which allows for the flexible control of feedbacks between prognostic components. Furthermore, CIME can be run stand-alone by specifying all components as either data or stub models

Future Work:

- Support for configuring system with different couplers
- An alternate hierarchical coupling layer and driver using NUOPC/EMSF
- An alternative coupling layer and driver supporting exascale
- Support for runtime mapping weight calculation
- Support for second order conservative remapping
- Support for pause/resume/rewind capability and the addition of a new component that will perform ensemble kalman filter file-based data assimilation. Rewind capability will also be leveraged to incorporate new fault tolerant capabilities

2. Details on the CIME Data and Stub Components:

`cime/components`

As mentioned above, the ability to selectively turn two-way feedbacks on and off between prognostic components is achieved via the replacement of one or more components with either data or stub components. Although CIME is NOT accompanied by any prognostic components, it does contain the full suite of data and stub model components that can be flexibly utilized by the CIMEs coupler to define and control feedbacks.

Future work:

- Faster and more scalable data models
- More extensive share namelist generation across data models
- Addition of 3d spatial and time-interpolation capabilities
- Addition of a data land-ice component

3. Details on the CIME Scripting and Testing Infrastructure

`cime/scripts, cime/utils, cime/cime_config`

CIME contains a scripting infrastructure that enables users to easily create, build and run numerous out-of–the-box experiments encompassing a wide range of model resolutions, component configurations and model feedbacks. In addition, CIME provides users with a standardized mechanism for custom configuration of individual experiments. CIME provides a standard interface to execute jobs across a wide range of compute systems. Finally, CIMME provides support for long running simulations and short and long term archiving of model data.

CIME also has system and unit testing frameworks that allow users to run whole test suites (including configuring, building, running and comparing) with one command. For any complex software system, and especially for a coupled climate modeling system, system and unit tests to verify infrastructure functionality on a variety of machines and compilers are essential. These tests satisfy a number of high-level software requirements, such as the ability to give bit-for-bit answers after restarting from a checkpoint, providing answers independent of processor and/or OpenMP thread count, and comparing against baselines to ensure that code refactorings do not change answers unless they are expected to do so.

Future Work:

- Faster configure, test and build systems in a more modern language
- Extension of code and machine coverage in tests
- Separation of configure and build systems
- Faster and larger number of system tests
- Fewer invocations of build when testing
- Support in build for more compiled languages.
- Removal of hard-coding of fields exchanged by components and incorporation of new xml-based namelist generation mechanism to generate the required fields exchanged from the compset specification

- New post-processing diagnostic capability that will accompany an experimental case

4. Details on additional CIME utilities.

```
cime/externals/pio1, cime/externals/pio2, cime/externals/gptl, cime/share,
cime/tools
```

CIME is currently accompanied by a growing collection of general-purpose scientific application infrastructure and utilities that are developed by both CIME developers and others. A few key examples are:

PIO: Since limiting external storage access to a single master processor creates a serial bottleneck, an efficient parallel I/O subsystem is a critical component of parallel earth system models. CIME contains a parallel I/O library called PIO, which is the main I/O library in both CESM and ACME components (including MPAS).

ECT: A new ensemble-based consistency test (CESM Ensemble Consistency Test, ECT) provides a critical new capability to formalize and simplify the software verification process of climate models. ECT is an objective tool for the evaluation of climate consistency in porting and model development work where bit-for-bit answers are not expected. It can be a replacement for the historical approach that is both subjective and computationally expensive.

Load Balancing: Determining the optimal number of CPU cores to allocate to earth system model components (i.e. achieving an even load-balance) can have a dramatic impact on the computational cost of a model simulation. A static load-balancing tool provides an effective means of boosting scalability and performance of the code and is incorporated into CIME.

Regridding: The current CIME coupling infrastructure provides parallel runtime regridding capabilities, but utilizes mapping files that are generated offline. Leveraging ESMF offline parallel regridding utilities, CIME provides a simple tool chain for the user to rapidly generate all necessary mapping files needed for a new model configuration.

Future work:

- Support only PIO2 across the system (currently both PIO1 and PIO2 are supported),
- Performance improvements to pio2
- Update and test load balancing tool

Contact: Rob Jacob (jacob@anl.gov)