Prototyping of the ESMF Using DOE's CCA

S. Zhou (a), A. DaSilva (b), B. Womack (a), G. Higgins (a)
(a) Northrop Grumman IT /TASC, 4801 Stonecroft Blvd., Chantilly, VA 20151
(b) Data Assimilation Office, NASA Goddard Space Flight Center, Greenbelt, MD 20771

Abstract-**Typical earth system models involve coupled model components in high performance computing (HPC) environments. There are two emerging component-based frameworks for developing HPC applications. The Earth System Modeling Framework (ESMF), funded by NASA Earth System Technology Office (ESTO) Computational Technologies (CT) Project, defines a domain specific component interface for earth system models. Common Component Architecture (CCA), funded by DOE, defines a general component model for HPC environments. The purpose of this prototyping work is to investigate how CCA can support the earth system models that are ESMF compliant. In particular we will present our findings on the process of integrating a simple coupled climate model into the CCA's CCaffeine framework.**

I. INTRODUCTION

A physical earth system model typically consists of several model components, which are coupled through exchanging data. For example, the well-publicized *El Nino-Southern Oscillation* (ENSO) phenomenon is the outcome of atmosphere-ocean interaction. Most models solve partial differential equations on a large gridpoint set for a long time period and consequently require significant amounts of high-performance computing resources. Since earth system model components, such as global atmospheric circulation models, also contain many physical processes such as radiation, cloud formation, precipitation, etc., considerable time and manpower are needed to develop a production-quality model. In addition, adding new physical processes to the models requires cross-organization collaboration. Hence, models are becoming increasingly complex and difficult to modify and maintain. The Community Climate System Model (CCSM) development clearly shows the trend towards greater complexity, coupling of models into larger systems and cross-organizational collaboration [1]. However, there are still many models developed and maintained by individual organizations [2]. A software framework to facilitate coupling models would be a great benefit to these and similar activities.

To achieve that goal, NASA's ESTO/CT project has funded the development of the Earth System Model Framework (ESMF). The ESMF project enables close collaboration from major U.S. earth system modeling organizations: the NSF/National Center for Atmospheric Research (NCAR), NASA/Goddard Space Flight Center, the Massachusetts Institute of Technology (MIT), the University of Michigan, DOE/Argonne National Laboratory (ANL), DOE/Los Alamos National Laboratory (LANL), the NOAA/Geophysical Fluid Dynamics Laboratory (GFDL), and the NOAA/National Centers for Environmental Prediction (NCEP). The ESMF software will consist of a superstructure for coupling and exchanging data between component models (e.g., atmosphere, ocean) and model subcomponents (e.g., physics, dynamics); and an infrastructure consisting of (1) data structures for representing grids and fields and (2) an optimized, portable set of low-level utilities. The data constructs and low-level utilities will be used by the coupling superstructure and may also be used separately to compose scientific applications. Conceptually, an application running under ESMF may be thought of as a sandwich, with the upper and lower layers provided by the ESMF and the middle layer provided by the application developer. The ESMF superstructure sits above the components of an application, controlling inter-component data transfer and sequencing. The ESMF infrastructure lies below the components, offering integrated tools for intra-component communication, error handling, time management, profiling, and other standard modeling functions. More information on ESMF can be found at [3].

To encourage the collaboration among the laboratories of the U.S. Dept. of Energy (DOE), DOE also funded the Center for Component Technology for Terascale Simulation Software (CCTTSS) as an Integrated Software Infrastructure Center (ISIC) under the Scientific Discovery through Advanced Computing (SciDAC) program. The members are from Argonne, Livermore, Los Alamos, Oak Ridge, Pacific Northwest, and Sandia National Laboratories, Indiana University and the University of Utah. CCTTSS is dedicated to the development of a component-based software development model suitable for the needs of high-performance scientific simulation, particularly the Common Component Architecture (CCA). The CCA Forum was organized to define a minimal set of standard interfaces that a high-performance component framework has to provide to components, and can expect from them, in order to allow disparate components to be integrated to build a running application. Such a standard will promote interoperability between components developed by different teams across different institutions. More information on CCA can be found at [4].

These two emerging component-based frameworks, ESMF and CCA, have a common goal: to promote interoperability between components developed by different organizations. However, CCA provides a more generic component interface that is not specific to any one type of application while ESMF is designed with specific component interface methods that are common to earth system model components. To investigate how an earth system model component which is ESMF-compliant can be supported in CCA, we have designed and developed an ESMF-CCA Prototype. A simple coupled climate model was developed and is used to explore the compatibility issue between ESMF and CCA. In this paper, we will describe our ESMF-CCA Prototype and present our findings.

II. DESCRIPTION OF ESMF-CCA PROTOTYPE

Coupling model components in an extensible and flexible way is a very challenging and domain-specific problem. CCA components interact by adhering to the uses-provides design pattern. This means that each component publishes the functionality that it allows other components to access. These published methods are known as *Provides-Ports*. Each component also publishes the functionality that it needs to have other components perform for it. These published methods are known as *Uses-Ports*. Conceptually a 'port' can be thought of as a contract between components of a system. It is equivalent to JAVA interfaces and pure abstract class definitions in C++. The CCA framework includes one additional type of port. The 'go' port is the starting point for executing systems of components. Driver components implement a '*go'* port which is used for scheduling and controlling the running sequence of components. A CCA compliant framework, like CCaffeine, is responsible for connecting and managing ports.

A typical earth system model such as climate models has three major functions that it must perform: initialization, run, and finalization. An initialization routine provides the functionality to prepare a model for simulation and includes initialization of parameters and boundary conditions. A run routine provides the functionality to allow the model simulation to manage its' time-evolution. A finalization routine provides the functionality to safely shut down operations, clean up memory and close any open files. ESMF's component model requires an ESMF-compliant model component to provide these functions and the ESMF superstructure is responsible for mapping these routines to the standard interfaces that an ESMF application component (driver) expects to call. In addition, ESMF provides a standardized, self describing, format for data exchange among model components through the ESMF_State data type.

A key difference between the CCA component environment and the ESMF component environment is ESMF's requirement for the user to supply additional standard methods (e.g. initialization, run and finalization) beyond those required for registering the component in the framework. ESMF provides a standard component adapter class, ESMF_CompComponent that maps the standard component interface to the functionality implemented by the model developer's code. CCA is not application specific and does not require any additional methods or standard interfaces.

Our ESMF-CCA Prototype will utilize CCA's Uses-Provides design pattern to create and couple CCA components that use the ESMF standard component interfaces described above. This will allow ESMF compliant components to run within a CCA component framework. CCA's Uses-Provides design pattern has been successfully used for coupling various components in a non-intrusive way.

III. DESCRIPTION OF A SIMPLE COUPLED CLIMATE MODEL

To explore the compatibility issues between ESMF and CCA and test the ESMF-CCA Prototype, we have designed and developed a 2D Simple Coupled Atmosphere-Ocean Model (SCAOM). SCAOM is computationally similar to, but physically simpler than, the Coupled Shallow Water Model [5]. The atmosphere model component uses a coarse rectangular grid while the ocean model component uses a fine grid (see Fig. 1). Each model component consists of a forward finite-difference advection scheme, and periodic boundary conditions along the x direction. Data is exchanged at the overlapped boundary along the y direction. In addition, each model component can choose an independent timestep. To couple an atmosphere with an ocean model component, we also develop two simple couplers: one from atmosphere to ocean and another from ocean to atmosphere. Basically, the coupler transforms data from one grid to another grid. The code is written in C++ since CCA's Ccaffeine framework supports C++ naturally. We will use ESMF utilities to deal with the issue of language nteroperability between C++ and Fortran in a follow-on project.

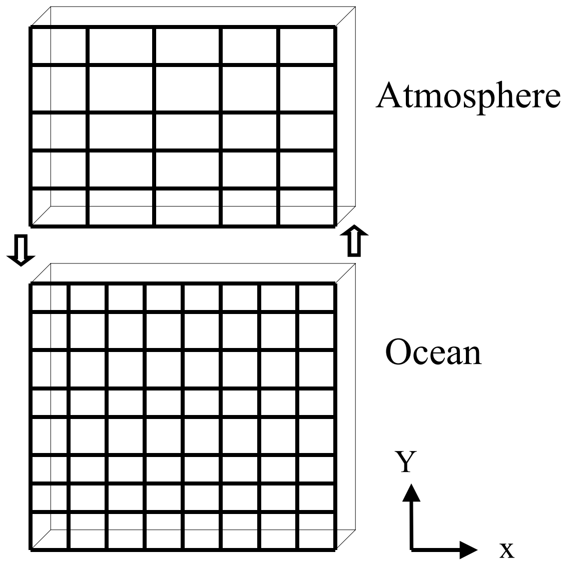


Fig. 1. Illustration of the geometry of a simple 2D coupled climate model

IV. RESULTS

A typical sequential coupling sequence between an atmosphere model and an ocean model is depicted in Fig. (2a), where t_{global} is the time advance in one coupling cycle and n_{cycle} is the number of coupled cycles. At the beginning of a simulation, each component is created. In the ESMF-CCA Prototype, we use CCA's setService utility to register the components of atmosphere, ocean, coupler from atmosphere to ocean (CplAtmXOcn), and coupler from ocean to atmosphere (CplOcnXAtm). In addition, data exchanged is implemented through import or export state which are of type ESMF_State (see Fig. (2b)). The atmosphere component first *provides* its data at its boundary, exportAtm, to the coupler, CplAtmXOcn. CpleAtmXOcn *uses* exportAtm, transforms it into importOcn with an interpolation routine, and *provides* importOcn to the ocean component. With importOcn, the ocean components run its finite-difference advection scheme for $n*t_{Ocn}$ timesteps while satisfying its periodic boundary condition in the x direction. After that, the ocean component *provides* its data at the boundary, exportOcn, to the coupler, CplOcnXAtm. The coupler, CpleOcnXAtm *uses* exportOcn, transforms it into importAtm, and *provides* importAtm to the atmosphere component. Similar to the ocean component, the atmosphere component *uses* importAtm, run its finite-difference advection scheme for $m*t_{Atm}$ timesteps while satisfying its periodic boundary condition in the x direction. Then, the atmosphere component *provides* its data at the boundary, exportAtm, to the coupler, CpleAtmXOcn. This completes a coupling cycle with the time advanced, t_{global} ($=n*t_{Ocn} + m*t_{Atm}$).

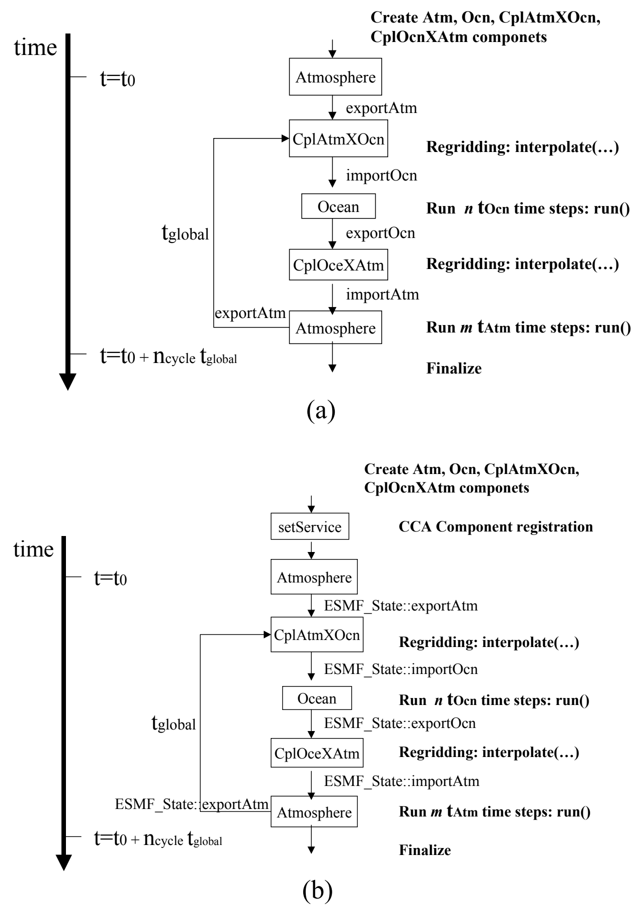


Fig. 2. (a) A typical sequential coupling between an atmosphere and an ocean model component. (b) The flow diagram (a) as implemented in the ESMF-CCA Prototype.

To test the ESMF-CCA Prototype, the 2D simple coupled climate model is integrated into the ESMF-CCA Prototype. The first step to integrate the model is decomposing the original application into components and sorting out the relationships among model and coupler components. Componentization appears to work quite naturally with the climate model since the ocean model and atmosphere model each have their own physical processes. Once identified and componentized, integrating those components into the ESMF-CCA Prototype is straightforward. Careful design of each component should be able to modularize and encapsulate each model well. Well-define interfaces enable the model to communicate with other components. To illustrate a simulation running in the environment of the ESMF-CCA Prototype, we choose the following parameters as listed in Table 1:

TABLE 1 SIMULATION SPECIFICATION

	Atmosphere component	Ocean component	Climate component (Driver)
Initial condition	zero	Gaussian distribution	
Grid size	16 x 16	31 x 31	
Number of timestep in each coupling cycle	5	1	
Number of coupling cycles			10

In the ESMF-CCA Prototype, each component is developed, compiled as a shared library (.so), and stored in an individual directory. With a script, a user can list those components to be selected from. After running the script, those components are loaded in and displayed in the CCA GUI environment as seen in the left side of Fig. 3 (palette). The user can select from the loaded-in components and drag them to the right side (arena). Connections can then be made between various components in the arena by clicking on a components provides port and dragging over to another components complimentary uses port. The red wires show the Provides-Uses relationship of the components and does not necessarily indicate the actual sequence of data flow between components. The *Go* port of the driver component controls the sequence of execution and data flow in this simulation.

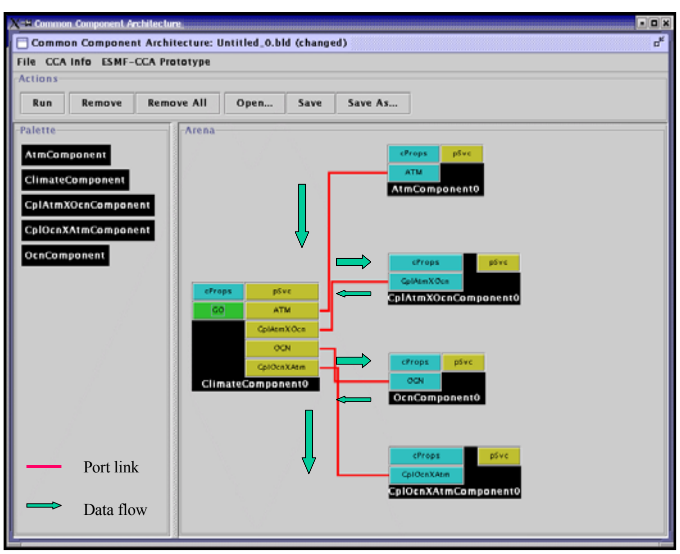
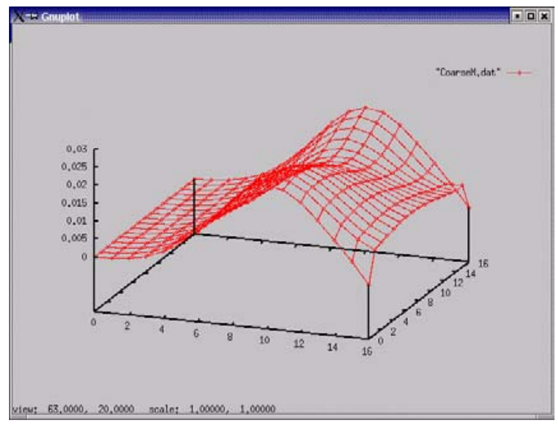
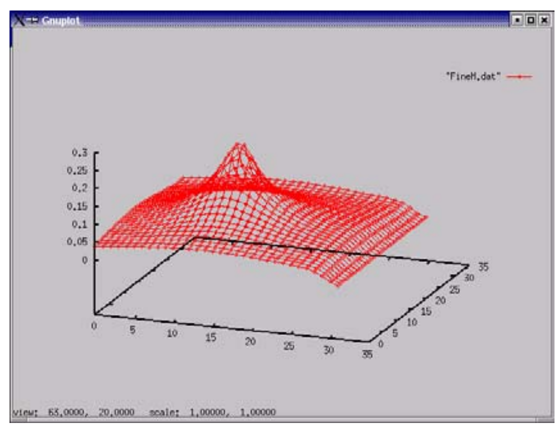


Fig. 3. Component relationship via CCA wiring diagram in a simulation of coupled atmosphere and ocean model components

During the simulation, the atmosphere and ocean component output data after each coupling. Currently, we process simulation data after the simulation finishes. In a follow-on project, we will develop a component, which can be used to visualize the simulation data in real time. Currently, there is a CCA visualization component for multi-processor environments, developed at the Oak Ridge National Laboratory. Fig. 4 clearly shows that a "wave" in the ocean model component has been advected into to the atmosphere model component, which had a zero field at the beginning, after 10 coupling cycles.



(a)



(b)

Fig. 4. Simulation results after 10 coupling cycle. The x and y axes are grid location, and z axis is one physical variable such as wind speed. (a) Atmosphere component (b) Ocean component

We have also developed another ocean model component with the same interface but with a different initial condition. With the ESMF-CCA Prototype, a user can dynamically select any one of several ocean model components to participate in the simulation (see Fig. 5). That is the exact feature required by climate modeling since climate prediction is based on an ensemble of simulations with different initial conditions and parameters. The ESMF-CCA Prototype can also support this kind of operation: once a simulation is completed, the components in Arena can be removed, rearranged, and rewired to perform another simulation.

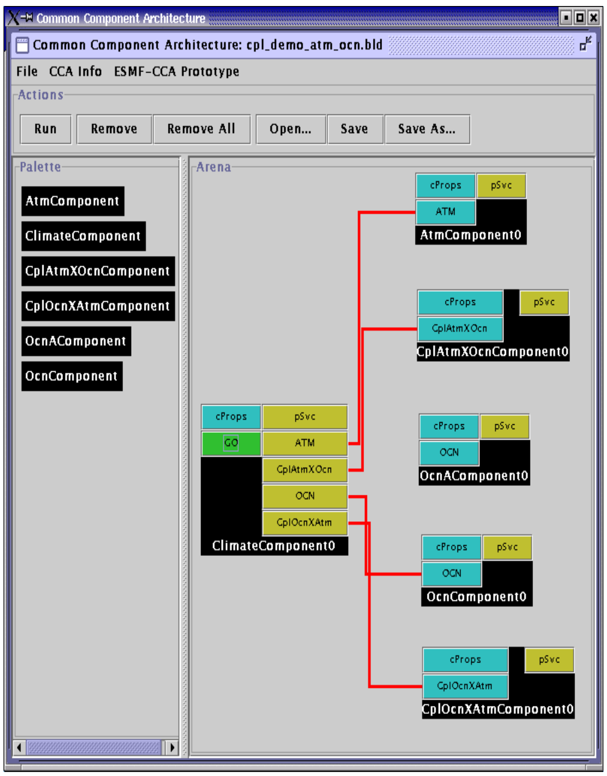


Fig. 5. Ocean model components are available to be selected during the running time of a simulation.

V. DISCUSSIONS

From the result of integrating a simple coupled climate model into the ESMF-CCA Prototype mentioned above, it appears that coupled climate models can be decomposed into modular components and can be naturally supported in component-based frameworks such as ESMF and CCA. ESMF and CCA complement each other nicely: ESMF standardizes the interfaces to earth system model components and provides utilities and standard data formats for building and coupling these components together. CCA provides a flexible method for controlling and managing components and their coupling. The ESMF's component design is important for creating an environment where similar components can be exchanged easily. This will allow comparison of components developed by various organizations without major changes to the climate model. However, the ESMF implementation does not provide the complete flexibility of easily connecting any components Provides-port to any other components matching Uses-port. CCA components can also be extremely small. All ESMF components must implement a non-trivial set of standard methods to be ESMF compliant. Once ESMF compliant earth system model components are available from the ESMF application development teams, we will integrate one or more of them into the CCA framework. In addition, we will replace the current simple coupled atmosphere-ocean model with the Cane-Zebiak Model, which is famous for its prediction of *El Nino* phenomenon[6]. The existing Cane-Zebiak model is written in Fortran and will be componentized. The ESMF utilities will be used to deal with the issue of binding Fortran and C++.

VI. CONCLUSION

We have successfully developed a prototype with the features of ESMF and CCA, and demonstrated that climate models can be naturally supported in a component-based software framework through integrating a simple coupled climate model into the ESMF-CCA Prototype.

ACKNOWLEDGEMENT

This is project is supported by NASA Earth System Technology Office Computational Technology (CT) project. We would like to thank W. Elwasif and D. Frantz for installing CCA's Ccaffeine framework and R. Armstrong, B. Allan, J. Ray for technical support in using Ccaffeine framework.

REFERENCES

[1] http://www.ccsm.ucar.edu/
[2] http://nccsinfo.gsfc.nasa.gov/ESS/esmf_tasc/index.html
[3] http://www.esmf.ucar.edu/
[4] http://www.cca-forum.org/
[5] C. Cruz, M. Kelly, M. Clausen, S. Zhou, and G. Higgins, "Coupled Ocean-Atmosphere Shallow Water Model,''TASC Internal Report (2001).
[6] S. Zebiak and M. Cane, Mon. Weath. Rev. 115 2262-2278 (1987)