

Components, the Common Component Architecture, and the Climate/Ocean/Weather Community

J. Walter Larson, Boyana Norris, and Everest T. Ong

Mathematics and Computer Science Division, Argonne National Laboratory

9700 S. Cass Avenue, Argonne, IL 60439

{larson,norris,eong}@mcs.anl.gov

David E. Bernholdt, John B. Drake, Wael R. El Wasif,, and Michael W. Ham

Computer Science and Mathematics Division, Oak Ridge National Laboratory

P.O. Box 2008, Oak Ridge, TN 37831-6367

{bernholdtde,drakejb,hammw,elwasifwr}@ornl.gov

Craig E. Rasmussen

Advanced Computing Laboratory, Los Alamos National Laboratory

MS B287, Los Alamos, NM 87545

rasmussn@lanl.gov

Gary Kurfert

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

Box 808, L-365, Livermore, CA 94551

kurfert1@llnl.gov

Daniel S. Katz

Jet Propulsion Laboratory, California Institute of Technology

4800 Oak Grove Drive, Pasadena, CA 91109

Daniel.S.Katz@jpl.nasa.gov

Shujia Zhou

Northrop Grumman IT/TASC

4801 Stonecroft Blvd., Chantilly, VA 20151

szhou@pop900.gsfc.nasa.gov

Cecelia Deluca

Scientific Computing Division, National Center for Atmospheric Research

P.O. Box 3000, Boulder, CO 80307

cdeluca@ucar.edu

Climate/ocean/weather applications abound with challenges in high-performance computing, software development, and data management. Forecasters require highly reliable software. Researchers require analysis tools that offer ease-of-use and flexibility. Modelers desire access to powerful numerical schemes and acceleration of the model development/validation cycle. The existence of millions of lines of legacy Fortran code, and the advent of modern programming languages such as C++ introduce a language barrier to be surmounted. We believe these requirements can be met through component-based software engineering (CBSE).

CBSE is an emerging approach to help manage software complexity and increase the productivity of software developers and users. In CBSE, units of software are encapsulated as "components," interacting with other components only through well-defined interfaces, and allowing the component's internal implementation to remain opaque. Components can be composed to form applications, using either a GUI or a

scripting language such as Python. Often, well-designed components will be reusable across a number of different applications. CBSE builds on object-oriented programming concepts, and extends and refines the common practice of using widely shared software libraries. CBSE has many similarities with "domain-specific" computational frameworks, but is a more general and flexible approach; unlike domain-specific infrastructure, CBSE by its nature yields more readily reusable software.

The availability of many components of various functionalities dramatically simplifies the construction of applications. A diversity of components sharing the same functionality means adaptation to different computational environments can be as easy as replacing one component implementation with another that is more appropriate. In addition to productivity increases, CBSE frees researchers to shift focus from low-level details of their applications to more relevant issues, and simplifies group software development because components are natural units of work.

The Common Component Architecture (CCA, <http://www.cca-forum.org>) project strives to bring the benefits of CBSE to high-performance scientific computing. It is specifically designed to impose little performance overhead, support both tightly-coupled parallel and distributed computing, and lower barriers to incorporating legacy code into the CCA environment. The CCA uses the Babel <http://www.llnl.gov/CASC/components/> language interoperability tool to support components written in various languages (currently Fortran, C, C++, Java, and Python).

DOE supports CCA under the Scientific Discovery through Advanced Computing (SciDAC, <http://www.scidac.org>) program. Outreach to the climate community is multi-faceted, including collaborations with the DOE Climate Change Prediction Program (CCPP, <http://www.scidac.org/climate.html>), the NASA Earth System Modeling Framework (ESMF, <http://www.esmf.ucar.edu>) project, and other projects.

Collaboration with CCPP centers on the NCAR Community Climate System Model (CCSM, <http://www.cesm.ucar.edu>). CCSM is a coupled climate model, comprising several mutually interacting component models (atmosphere, ocean, sea ice, land-surface, river routing, flux coupler). Main focus areas are introducing components to the CCSM's atmosphere and flux coupler, and the development of a river routing model based on CCA components.

CCSM's atmosphere is the Common Atmosphere Model (CAM, <http://www.cesm.ucar.edu/models/atm-cam/>). CAM has been refactored to disentangle its dynamical core from its subgridscale physics parameterizations. This physics/dynamics split allowed a proliferation of dynamical cores, and CAM currently has three: a pseudo-spectral method; the Rasch-Williamson semi-Lagrangian method; and the finite-volume Lin-Rood scheme. Standardization of the interfaces to these dynamical cores and the overall subgridscale physics package is underway. Once complete, these software modules can be packaged as CCA components. This will allow one to use CCA to compose and invoke a stand-alone version of CAM in which the user may plug in the dynamical core of choice. Furthermore, the developer of a new

dynamical core need only code it to the interface standard and it will be ready for immediate validation.

The current CCSM flux coupler was implemented using the Model Coupling Toolkit (MCT, <http://www.mcs.anl.gov/mct>). MCT is a software package for implementing coupling between messaging-passing parallel applications. MCT is implemented in Fortran90, and its programming model is based on F90 module use to declare MCT-type variables, and invocation of MCT routines to create couplings. We are using Babel to create interlanguage interfaces to MCT, which will allow other languages to use MCT. We are repackaging major MCT services as CCA components, most notably the parallel sparse matrix-vector multiplication scheme used to implement intergrid interpolation. Componentization of MCT will export some MCT services to CCA, and will allow MCT-based applications to replace some MCT services with other equivalent components. We are also using MCT to construct a parallel, directed-graph-based river transport model, which will subsequently be built using MCT-inspired CCA components. Finally, MCT's services offer CCA a more detailed picture of model coupling requirements.

NASA supports ESMF to specify and develop a domain-specific framework for earth system model interoperability. The ESMF comprises an infrastructure and a superstructure. The infrastructure includes low-level utilities and a data model used to define component interfaces. The superstructure comprises ESMF components and the means to manage them. ESMF provides an environment for instantiation and composition of components and subsequent execution. We are working with ESMF to ensure interoperability between the ESMF and CCA frameworks; i.e., to guarantee that ESMF components may be used within CCA, and that ESMF will be able to utilize the wide variety of CCA-compliant components that exist or are under development. An ESMF-CCA prototype has been developed that adopts the features of component registration and GUI for component connection from CCA. Each component is designed to meet the requirements for an ESMF component, and data exchange between components is through a standardized, self-describing datatype similar to the ESMF_State.

Observational data streams and reanalysis data are vital to climate/ocean/weather applications, and we are developing component interfaces to meet this need. We have prototyped a netCDF (<http://www.unidata.ucar.edu/packages/netcdf/>) component, and will be extending this work to include the parallel I/O-enabled netCDF. We also wish to develop CCA components capable of handling other major geophysical data formats such as GRIB and HDF-EOS (<http://hdfeos.gsfc.nasa.gov>). Such data components would allow users to construct with great ease powerful data-intensive analysis and modeling applications.

We have described CBSE, the CCA, and its outreach to the climate/ocean/weather communities. This is work in progress and we are seeking collaborators to expand its scope. We believe CCA can offer great benefits to the climate/ocean/weather community.