

## Center for Component Technology for Terascale Simulation Software

**PI:** Rob Armstrong (SNL), **Co-PIs:** David Bernholdt (ORNL), Lori Freitag (SNL), Dennis Gannon (Indiana University), James Kohl (ORNL), Gary Kumfert (LLNL), Lois Curfman McInnes (ANL), Jarek Nieplocha (PNL), Steven Parker (University of Utah), and Craig Rasmussen (LANL).

### Summary

*The Center for Component Technology for Terascale Simulation Software (CCTSS) is dedicated to accelerating science by bringing a “plug and play” style of programming to high-performance computing. Through our programming model called the Common Component Architecture (CCA), we can dramatically reduce the time and effort required to compose independently created software libraries into a new terascale applications. We have two major early-adopters of our technology in the application areas of combustion and quantum chemistry. A third application group, the Community Climate System Model, has CCA-enabled prototypes and are planning to adopt the CCA. By design, the customer sacrifices almost no performance when using CCA components. Moreover, components that are written in various languages, such as Fortran, C++, and Python, all work together transparently.*

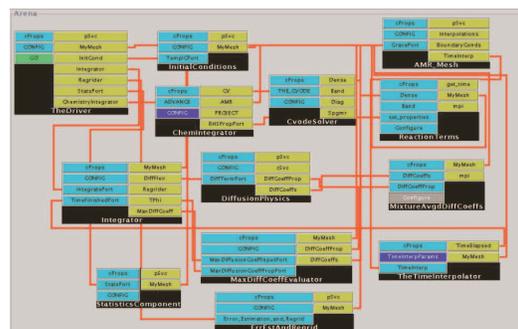
The CCTSS center was formed around the grass roots Common Component Architecture (CCA) Forum, already in existence for three years prior to the start of the SciDAC initiative. The purpose of the CCA and the tools provided by the CCTSS is to bring the software component paradigm to high-performance computing, thereby reducing barriers to scientific software reuse and enabling scientists to be more productive when constructing their codes. This mode of programming is common in industry, but was heretofore unknown in high-performance scientific computing. The CCA effort is differentiated from those in the commercial sector because we address the challenges associated with maintaining high performance, work with a broad spectrum of scientific programming languages and computer architectures, and aim to ensure that the DOE investment in legacy codes is preserved.

### CCA Usage in SciDAC Applications

The vision behind the CCA specification, and guaranteed by CCTSS tools, is to allow independently developed software modules to be composed together to form complete high-performance simulations. If a scientific programmer follows the “rules” that are laid out

in the CCA specification, or uses the tools supplied by the CCTSS, then the resulting component is guaranteed to work with any other that is also CCA compliant. This paradigm enables a fundamental shift in the software development process. Rather than a handful of ‘hero’ programmers creating a monolithic executable, many developers and domain experts can collaboratively contribute components, which can then be assembled and reused in production scientific simulations.

Nothing illustrates this vision better than the software “facility” being developed by the SciDAC application center, *A Computational*



Component-based combustion application (CFRFS).

*Facility for Reacting Flow Science (CFRFS)*, led by H. Najm. Just as physical laboratories supply scientific facilities to visiting researchers, so this facility of CCA-compliant software components provides the building blocks of combustion simulations. Investigators can compose applications from these tools and add more components to prove a scientific point or otherwise increase the value of the facility.

In addition to this work, CCTTSS researchers are collaborating closely with application scientists to create massively parallel simulation codes in the areas of quantum chemistry and climate modeling.

### **CCA Technology Development**

To enable CCA-compliant applications to function properly, the CCTTSS is addressing a number of challenging research questions. In particular, we are pursuing ongoing work in framework infrastructure development, language interoperability issues, scientific component design, and parallel data redistribution.

*Frameworks:* A CCA framework is the tool that creates components, composes them into applications, and destroys them when they are no longer needed. The CCTTSS maintains three frameworks: Ccaffeine, which is devoted solely to parallel components and applications; XCAT, which supports components running on distributed remote machines (via Grid standards); and SCIRun, which supports both modes of programming. Ongoing research addresses interoperability among these tools.

*Language Interoperability:* Because application scientists use a variety of programming languages, the CCA has been designed to be language neutral. Our Scientific Interface Definition Language (SIDL) and associated compiler, Babel, allow component interfaces to be written and used from any of C, C++, Python, and Fortran with minimal performance costs. Chasm is a set of automatic language transformation tools that ease the burden of developing components from existing code. Future work focuses on integrating Chasm and Babel to provide a single powerful tool for automatic language interoperability.

*Scientific Components:* As a part of its mission, the CCTTSS has developed production components that are used in scientific applications as well as prototype components that aid in teaching CCA concepts. These freely available components include various service capabilities, tools for mesh management, discretization, linear algebra, integration, optimization, parallel data description and “MxN” data redistribution, visualization, and performance evaluation. The CCTTSS is collaborating with domain experts in the TSTT and APDEC SciDAC centers to define common interfaces for mesh-based scientific data management. These data-centric components will define the parallel layout of data across processors and form the *lingua franca* for other components that manipulate the data. We are also working in conjunction with the TOPS SciDAC center to define component interfaces for linear, nonlinear, and optimization solvers.

### **CCA Outreach Activities**

Much of CCA’s success thus far is due to the outreach done by the CCTTSS in the SciDAC and larger computational science community. During the past year, we conducted six tutorials and have provided extensive documentation on our web site, <http://cca-forum.org>.

### **Benefits from SciDAC**

SciDAC funding has accelerated both CCA technology development and the insertion of this technology into massively parallel scientific applications. After one year of funding, CCA components and infrastructure are used in two major SciDAC applications, numerous CCA-compliant application components have been developed and deployed, and the underlying infrastructure is maturing and establishing itself in the SciDAC scientific community.

### **Further Information:**

<http://www.cca-forum.org>

### **Contact Information:**

Robert Armstrong (PI)  
Sandia National Laboratories  
925-294-2470  
rob@sandia.gov