# Applications Using CCA Components

These wiring diagrams for three scientific applications using the CCAFFEINE framework demonstrate the reusability and composability of CCA-compliant components.

Two applications solve a time-dependent PDE: one uses adaptive structured meshes and the other uses unstructured meshes; the third application solves an unconstrained minimization model using structured meshes. All three applications re-use common components for parallel linear solvers, data description, data redistribution, and visualization.

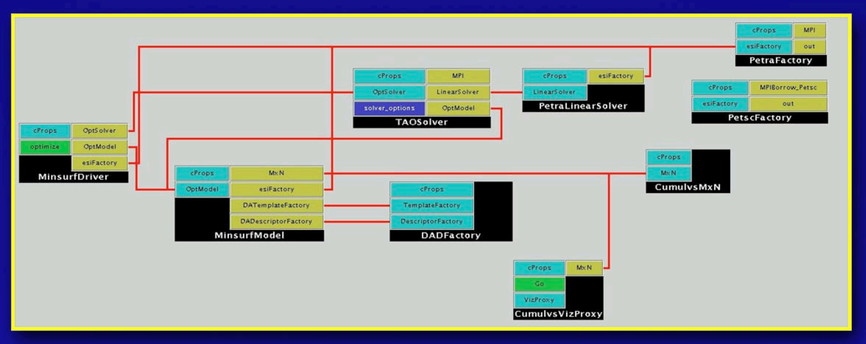
These components were created by seven researchers at Argonne, Oak Ridge, and Sandia National Laboratories, leveraging and extending parallel software tools developed at different institutions, including CUMULVS, GrACE, LSODE, MPICH, PETSc, PVM, SUMAA3d, TAO, and Trilinos.

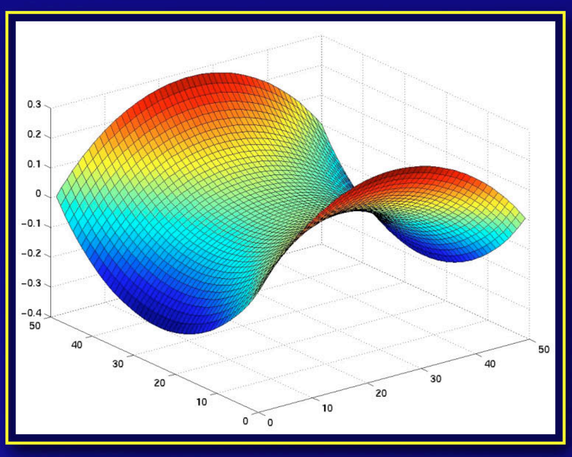
### *Component Pallet*

The component pallet is a "bag of tricks" from which applications are composed. Each component is created from this pallet of component classes. Often multiple instances of the same component class are used in a given application.


### *Solution of an unconstrained minimization problem (determining minimal surface area given boundary constraints) using the TAOSolver optimization component*

***TAOSolver*** uses linear solver components that incorporate abstract interfaces under development by the Equation Solver Interface (ESI) working group; underlying implementations are provided via new ESI interfaces to parallel linear solvers within the PETSc and Trilinos libraries. These linear solver components are employed in the other two applications as well.



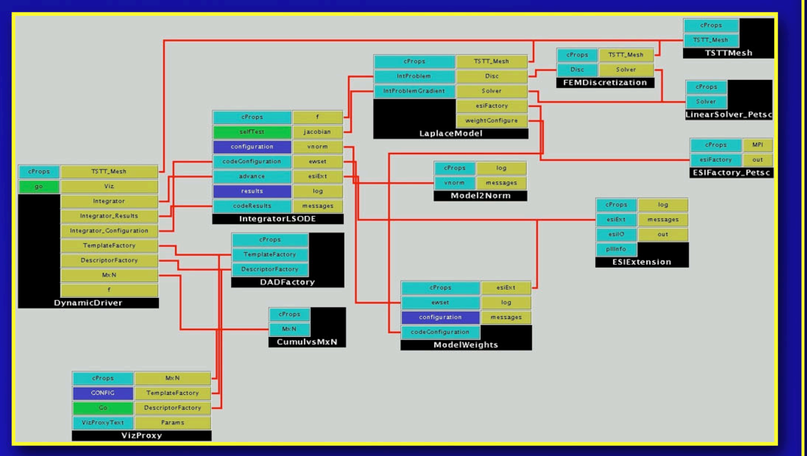

### *Wiring Diagrams*

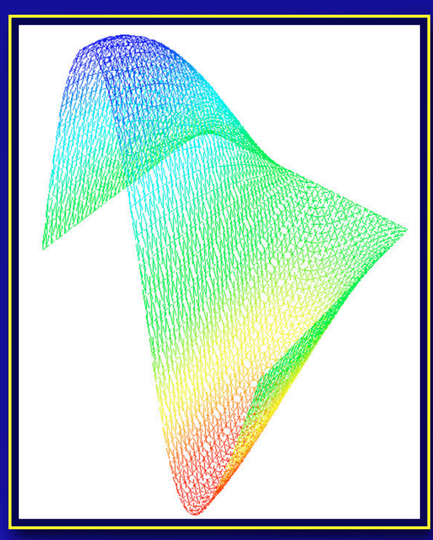
Within the wiring diagrams, each large black box represents a component, while the small blue and gold boxes indicate component ports. Blue boxes represent "provides" ports, or interfaces that a component provides to others; gold boxes represent "uses" ports, or interfaces that a component wants to use from another component.

### *Solution of a time-dependent PDE using a finite element discretization on an unstructured mesh*

***IntegratorLSODE*** provides a second-order implicit time integrator, and ***FEMDiscretization*** provides a discretization.

This application (and the other two applications as well) use the ***DADFactory*** component to describe the parallel data layout so that the ***CumulvsMxN*** data redistribution component can then collate the data from a multi-processor run to a single processor for runtime visualization.



### *Solution of a two-dimensional heat equation on a square domain using an adaptive structured mesh*

***IntegratorLSODE*** provides a second-order implicit time integrator, and ***Model*** provides a discretization. The remaining components are essentially utilities that construct the global ODE system or adaptors that convert the patch-based data structures of the mesh to the globally distributed array structure used for runtime visualization.