The Biomedical Problem Solving Environment, or BioPSE, lies at the computational core of our NCRR Center. In this environment we develop modules, design simulations, and conduct scientific studies; a carefully crafted design is required for such an environment to meet all of its users’ needs. This environment provides an infrastructure that liberates the scientist from the user-intensive, mundane tasks associated with most existing software tools. By commanding a modular, extensible, interactive set of software tools, scientists are free to apply their expertise to the science at hand.

**Driving Goals:** There are a number of challenges in building a powerful software architecture for biomedical applications.

- **Accessibility and Usability:** How to satisfy a wide range of users (from software developers to biomedical scientists) who have applications spanning a wide variety of biomedical domains (from bioelectric fields to mouse phenotyping).
- **Integration and Extensibility:** How to make the software extensible so users can extend it to fit their needs, and so it can interact with other software systems.
- **Performance and Control:** How to make the software easy to use and robust, while also making it efficient and high-performance.

### BioPSE Electric Field Simulation

The image below depicts a SCIRun (BioPSE) dataflow network used to solve a forward bioelectric field problem. SCIRun is a problem solving environment (PSE) that provides a framework which allows scientists to interactively create and connect a collection of modular components to solve a large variety of problems. Each component reads data from its input ports, processes the data, and then send the result on the next component through its output port. Components have a UI button that allows the user to bring up a graphical user interface to dynamically interact with that module.

### The SCIRun Software Framework

SCIRun provides a core software framework for a number of different research projects including the BioPSE System (NIH NCRR), the Uintah System (for the DOE ASCI Center for the Simulation of Accidental Fires and Explosions), the DOD Virtual Soldier project, and the Fusion package (funded by a DOE SciDAC). Each project benefits from the functionality provided through the other projects. This joint effort is what allows for the creation of such an extensive, extendable, and powerful toolkit.

### SCIRun PowerApps

SCIRun PowerApps were developed to remove some of the need to understand all of the complexity and intricacies of large SCIRun dataflow networks so the App user can concentrate on the problem at hand. As can be seen in the above figure, the PowerApp is layered on top of the dataflow network and individual component UIs. Only the most important user interface elements are exposed to the user. Each PowerApp is created to address a specific task, thus dramatically reducing the amount of software complexity that the user must understand.

### BioImage PowerApp

The BioImage PowerApp is used for processing and visualizing medical image volumes. Users can directly load DICOM, Analyze, VFF, NRRD, and PICT data formats, and can apply processing algorithms (cropping, resampling, histogram equalization) to specify and enhance regions of interest. Users can seamlessly move between the 2D and 3D views to precisely control how different features of their data are displayed, and to gain both quantitative and qualitative insights into their data.

### BioTensor PowerApp

The BioTensor PowerApp processes and visualizes diffusion weighted images (DWI). It can correct for common distortion artifacts, estimate tensors from the DWIs, and visualize the diffusion tensor field. An important application of diffusion tensor imaging is determining the connective structure of central nervous tissue. BioTensor allows the user to visualize individual tensor samples, determine the overall shape of anisotropic regions through isosurfaces of different anisotropy metrics, and trace the connectivity along white matter tracts by forming paths that follow the tensor principal eigenvector, or use the tensor-line algorithm.

This work was supported by the NIH/NCRR Grant P41RR12553 (PI: Chris Johnson), C-SAFE (B524196), and the National Fusion Collaboration (DEFCO201ER25457)