

# SCIRUN AND KEPLER DATAFLOW INTEGRATION

Steven G. Parker, Ayla Khan, Oscar Barney



## ABSTRACT

Building useful scientific workflows often requires disparate software to interoperate in the workflow pipeline. The C++-based SCIRun problem solving environment provides a multi-threaded GUI-based platform for building dataflow networks. SCIRun networks are built using integrated scientific computation algorithms, data processing, simulation and visualization tools. The Kepler project, based on Ptolemy II software, provides developers with tools for building complex, distributed and data-intensive scientific workflows using the Vergil interface. The Scientific Process Automation (SPA) project uses Kepler to build loosely coupled workflows using web services and workflow automation tools.

Our goal is to incorporate SCIRun in SPA workflows. To this end, we used Java Native Interface (JNI) functions to create an interface encapsulated in a Ptolemy-based actor to instantiate SCIRun in the same address space as the Java-based Kepler process. As a prototype of this integration, a triangle mesh dataset was generated in Vergil and carried to SCIRun through this interface, where a SCIRun network converted the data into a SCIRun mesh and used SCIRun models to visualize the results. Using similar methods, SCIRun can be inserted into SPA workflows as an automated analysis tool controlled by a Kepler-based actor.

- A Kepler actor declares native methods implemented in the SCIRun Ptolemy package using JNI and calls native functions to access mesh data from Java.
- Multi-dimensional arrays are converted from Java object to nrrd (nearly raw raster data) format.

Nrrd is a library and file format that supports image processing involving N-dimensional raster data and supports a large range of data types (<http://teem.sourceforge.net/nrrd/index.html>).

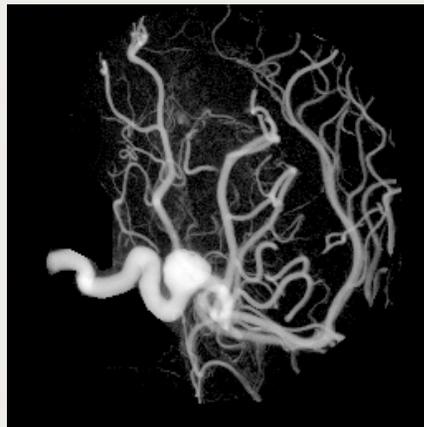


Figure 2. Example of data stored in a nrrd (<http://teem.sourceforge.net/nrrd/files/an-var.jpg>).

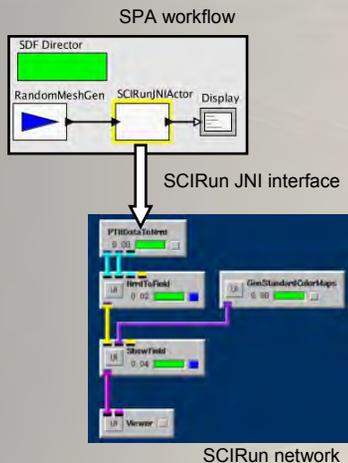


Figure 1. Randomly generated triangle mesh converted to a SCIRun mesh.

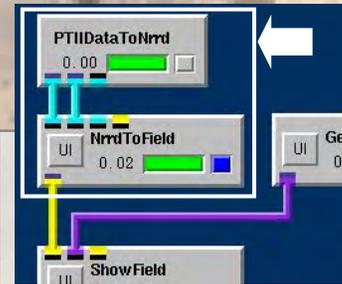


Figure 3. SCIRun modules that convert multi-dimensional arrays from Java object to nrrd format, and nrrd to SCIRun type.

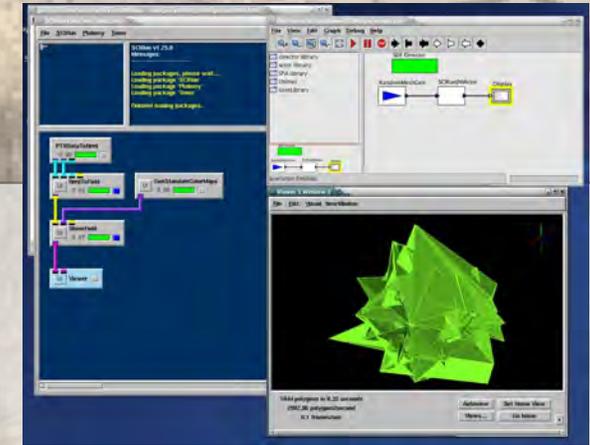
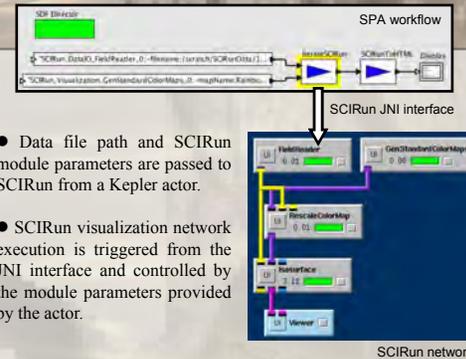


Figure 4. SCIRun Kepler interface.



- Data file path and SCIRun module parameters are passed to SCIRun from a Kepler actor.
- SCIRun visualization network execution is triggered from the JNI interface and controlled by the module parameters provided by the actor.

Figure 5. Automated SCIRun Network Execution



Figure 6. Automated SCIRun network execution.

## CURRENT CAVEATS

- The implementation currently requires a pre-existing SCIRun network file containing the SCIRun modules (name and path of saved network is set by SCIRunJNIActor)
- The SCIRun-Kepler interface is somewhat unstable due to thread model incompatibilities
- Data from other actors must be encapsulated in a SCIRun data object
- The Teem package converters may limit access to the full range of SCIRun data types

## NEXT STEPS

- Target specific workflows
- Expand support for SCIRun data types
- Improve SCIRun synchronization with Java
- Add the ability to send data from SCIRun to Java actors
- Increase the efficiency of reading large data sets from Java
- Allow the processing of multiple data sets in one iteration or over multiple iterations