Scopira: An Open Source C++ Framework for Biomedical Data Analysis Applications – A Research Project Report

Aleksander Demko¹,², Rodrigo A. Vivanco¹, Nick J. Pizzi¹,²

¹Institute for Biodiagnostics, National Research Council of Canada, Winnipeg, Canada. ²Department of Computer Science, University of Manitoba, Winnipeg, Canada

Scopira

Scopira is an open source, object-oriented and generic C++ framework for scientific computing applications with emphasis on biomedical data analysis.

- Spectra and images (magnetic resonance, infrared, gene microarrays, Raman, mass spec).
- Visualization (2D and 3D via OpenGL).
- Computationally efficient.
- Parallel algorithm development (MPI and a dedicated agent facility).

Applications

Scopira used to build a suite of applications and plug-ins, each with varying scopes of complexity:

- Algorithms prototyping, possibly with visualization (BigVol).
- Parallelized algorithms, via MPI, for the analysis and classification of biomedical data (SFS).
- Plug-ins for pattern recognition algorithms, visualization, and data projection (RDP).
- Full, stand-alone applications (ScopiraPA, Evident®, Opus)

Core Subsystem

Scopira provides a dynamic system (via run-time registration) for applications and plug-ins to declare modules and views.

- Models are objects that are monitored by zero or more views. Views themselves can be graphical or non-graphical.
- A project is a model that organizes a collection of models in a tree-like fashion. Users may use projects to save their data sets as related workspaces.

DirectIO: Memory Mapping

The basekit subsystem contains various numerical routines and the core numerical data structure, narray.

- Generic, via C++ templates structure. Can be used with ANY data types (ints, floats, complex, etc) and ANY dimension (vector, matrix, cube, etc).
- Complementary nview virtual sub-view, any size <= narray dimensions.
- As good-as-C performance via templates and inline methods.
- Range-checked access via assert(). (debug mode only).
- STL-style iterators and thereby usable with STL algorithms.
- DirectIO back-end, to directly access file as if they were in memory (via the operating system's memory function).

Basekit Subsystem

The subsystem provides a host of generic facilities and utilities for all types of Scopira based applications.

- Reference counting with "smart pointers".
- Threads and concurrent programming.
- Random number generators ("real" and pseudo) and distributions.
- Input/output & object serialization.

I/O & Serialization: flows

Scopira contains a layered I/O system with flows. Flows provide a three layer interface architecture for the I/O of bytes, simple data types and objects (serialization).

- Binary sinks (and terminators for binary streams).
- Binary transformers (convert basic data types to bytes).
- Data type encoding layer (converts basic data types to bytes).

Object serialization layer (converts objects to data types).

Parallel Computing

This subsystem provides a light (inner) helper layer for Scopira apps that use the MPI (Message Passing Interface) API.

- Aware of narray and uses type informations (via C++ traits classes) to deduce many typological parameters needed for many MPI functions.
- Dynamically adjust the amount of information needed from the programmer.

- Tasks (agent-managed) that send messages to other tasks & migrate between agents over the network.

Remote Agent Infrastructure

Remote Agent Instance

A user application and various agent instances managing a collection of tasks