

# Integrated Open-Source Geophysical Processing and Visualization

Glenn Chubak<sup>1</sup> and Igor Morozov<sup>1</sup>

<sup>1</sup>University of Saskatchewan, Saskatoon, Saskatchewan, Canada ([gdc178@mail.usask.ca](mailto:gdc178@mail.usask.ca))

## Abstract

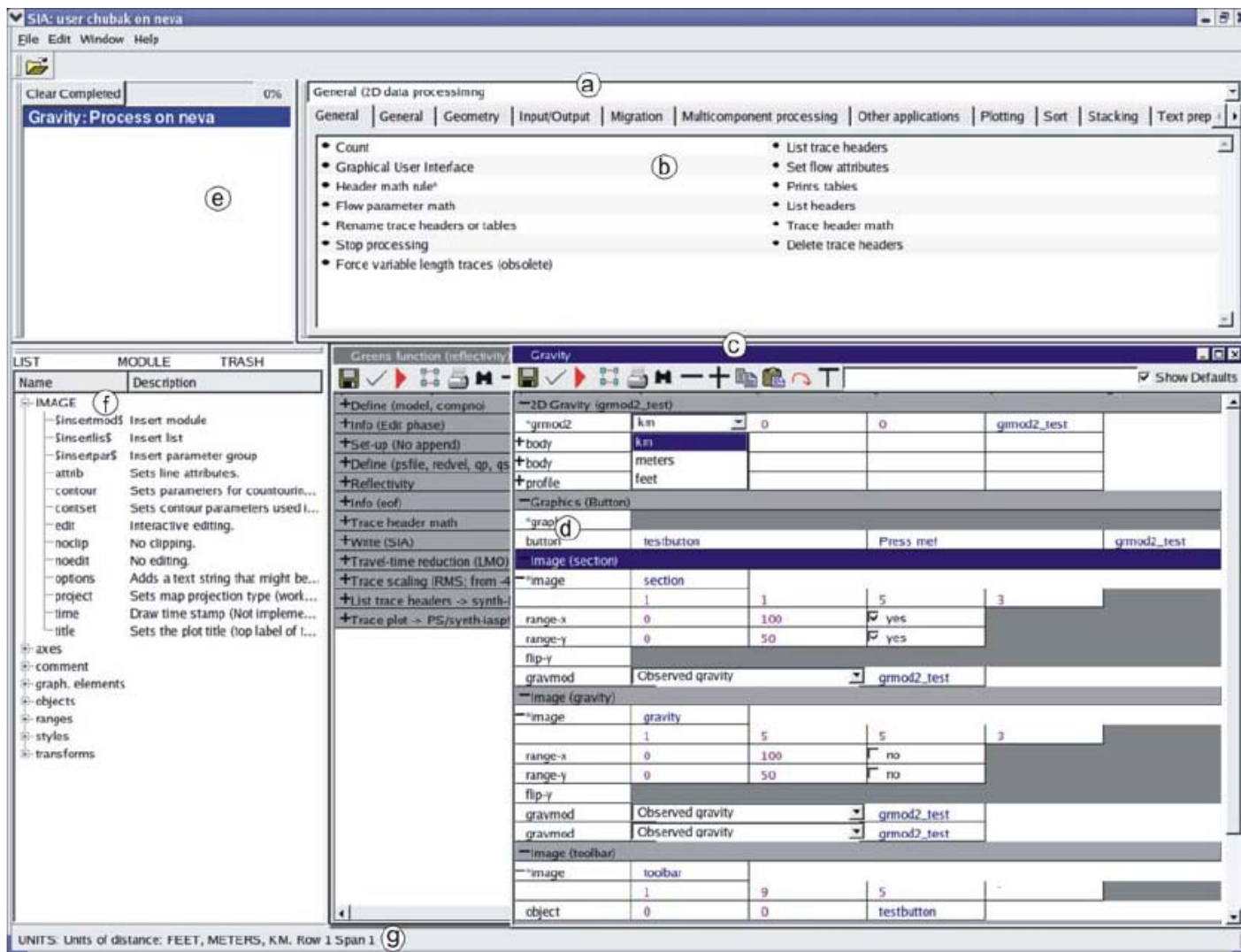
Open-source software is rapidly gaining acceptance for use in both academic and industry environments due to its ability to be adapted to specific tasks by the end user, often low cost, and wide spread adoption of the Linux operating system. However, geophysical data processing is still mostly done by commercial packages as the open-source alternatives, is limited in their scopes or lack the required integration and sophistication. Our SIA package (<http://seisweb.usask.ca/SIA>) attempts to respond to this challenge by creating a general open-source code framework to meet the needs of both academic and commercial researchers in several areas of geophysics. Here, we present the recent updates of the package while focusing on three key aspects of this framework: its data processing, software development, and collaboration models.

## Processing System

SIA currently is a nearly complete seismic processing system, with many tools reaching to the broader geophysical applications (Chubak and Morozov, 2006). The current system scope includes reflection, wide-angle, and to certain degrees earthquake seismology, 2- and 3D potential field processing and inversion, and graphics. Nearly 200 dynamically linked plug-in modules are closely integrated with a content-agnostic processing monitor and often between each other. Almost any type of data can be handled by the system making it possible to merge multiple data types. The following features further enhance the flexibility of the system:

- 1) “Trace records” can be of variable data formats, sampling intervals, record lengths and time starts. They can also contain linear arrays (seismic records) or 2- and 3D arrays representing multi-component seismic records, or 2D grids used in potential field processing.
- 2) Other types of data are broadly used and often introduced by new tools: velocity models, travel-time curves, database tables, Artificial Neural Networks, and various graphical objects.
- 3) “Tool” parameterization is unusually flexible, and allows using trace headers, database fields, symbolic text substitutions, and UNIX command-line parameters interchangeably with constants. Many tools support complex structured parameterizations allowing, for example, to design custom graphical user interfaces (GUI) or build PostScript plots. Tools can be represented by binary codes or macro-commands combining other tools, with coherent parameterization and optimized for a particular task.
- 3) All flows can operate from the user’s GUI or in parameterized command-line (batch) mode allowing execution of complex self-documented processing sequences.

As with other seismic data processing packages, flows consist of a sequences of tools, which are applied to the data. Constructing processing flows is greatly simplified by a modern GUI, which also provides the utilities users expect from commercial software, such as project management, process monitoring and control, search, and extensive context-sensitive help (Figure 1). The GUI is based on the Qt libraries from



**Figure 1. Main SIA Graphical Interface window including: a) selectable tool packages, b) tool library for the selected package; c) multiple-job flow editor; d) parameterization of the selected tool; e) job monitor. Note that the jobs can be executed concurrently on remote and multiple systems. Status line (f) gives brief information about any item at which the cursor is pointing. For a compact display, tool parameterizations can be hidden leaving only one-line summaries that may change to display job progress (window c). Job flow editor windows (c) also include tool bars providing flow-related functionality.**

Trolltech, so that SIA can be ported to a variety of operating systems, such as Linux, Solaris, or even OS X with only minimal effort. In a grid or cluster environment, its configuration is also done from within the GUI by specifying the nodes on which a particular flow and its components (subflows, I/O, display tools) is to be run. This allows multiple processing jobs to be either ran in parallel on a Beowulf cluster or distributed over a peer network.

Visualization and interaction with the data is a key step in processing and interpretation. To this end, the currently ongoing efforts have focused on integration of seismic display and interpretation tools through introduction of generalized 2D/3D visualization model based on OpenGL. The visualization system is entirely controlled by the processing flows and is able to render a variety of basic data types including:

- 1) Seismic traces with adjustable settings and arbitrarily positioned in 3D.
- 2) Lines and surfaces with variable styles, colors, markers, etc.
- 3) Bitmap-style graphics rendered on any plane oriented in 3D.
- 4) Customizable line styles, color, color palettes, labels, push buttons, etc.

Complex objects (such as velocity and gravity models) are composed of the objects above by the corresponding tools. In addition, user-specified coordinate transformations are available, so that images can be rendered on arbitrary surfaces. This allows, for example, drawing 3D fence diagrams or displays on the topographic relief or on the surface of ellipsoidal Earth. When fully implemented, this custom object display protocol will allow building complex interactive tools including, for example, multiple picking, ray tracing, and gravity modeling sessions on a single screen.

The visualization system is highly customizable by the flow without the need for any programming. Any display component (trace and model displays, buttons, labels, and graphics) can be added to the visualization window by inserting the appropriate tool in the flow and configuring its parameters. Interaction with the user can cause the host processing flow to modify its operation or even execute other processing flows. In this way, it is possible for a user to create entirely new ways of interacting with both the processing system and the data.

An object tree is built by the visualization system allowing the user to interact with each component that is rendered and change parameters, such as show/hide, change color, or edit the corresponding object. The mouse controls view direction and zoom as it is done in other interpretation programs. OpenGL ensures that the system will work on a wide variety of systems and takes advantage of the hardware acceleration on graphics cards. It also allows native 3D rendering on stereoscopic displays, such as Geowall (<http://geowall.geo.lsa.umich.edu/>).

The key components of the system (the GUI, processing flows, visualization and display tools) operate asynchronously and communicate via a Parallel Virtual Machine (PVM) interface. Because of this, the many components of SIA can be distributed, allowing, for example, to distribute the processing load or for the visualization program to operate on a dedicated computer system.

Interestingly, SIA is apparently the first seismic processing system to operate as a web service (Morozov et al., 2006). A standard distribution installed on a system accessible via HTTP (e.g., <http://seisweb.usask.ca/SIA/ps.php>) receives processing jobs, executes them and returns the results, currently in the form of web pages or files ready for download. The content of this processing is entirely controlled by the client. The

client is even able to upload web forms on the server and associate them with processing jobs, thereby creating custom web data or processing services. This approach was recently utilized to generate a library of SIA processing examples, some of which are also executable on-line (<http://seisweb.usask.ca/temp/examples>).

### **Development Framework**

From its inception, SIA was not intended as a complete product to serve a specific narrow task, such as, for example, reflection seismic processing (Morozov and Smithson, 1997). Instead, the design goal was to provide an extensible framework capable of supporting virtually any type of geophysical data processing, modeling, or interpretation. However, due to the character of the previous applications, most of SIA tool development so far was related to seismology.

The system allows its users to rapidly add new functionality with a minimal effort. Two principal features simplify development within SIA: 1) new modules can be added to perform custom data processing while taking advantage of other tools and extensive C++ class libraries, including Qt graphics, and 2) custom interactive graphical applications can be created by simply designing processing flows and without any computer programming.

New modules can be coded using a mixture of C, C++, FORTRAN, and even Pascal or Java. At the University of Saskatchewan, graduate students routinely write new processing modules for class exercises and to further their research. In our experience, a reasonably complex tool can be completed in only a few days. Templates have been created to aid in the development process and a complete set of compilation and linking tools are provided. New modules are integrated with the system by the maintenance utilities so that they become available from the graphical interface and provide fully functional context-sensitive help to the user. The documentation provided by the programmer is also included in the auto-generated HTML tree (<http://seisweb.usask.ca/SIA/index>).

### **Collaboration and Code Distribution Model**

Within the academic community, the development of computer code is still generally performed in an ad hoc manner, without investing significant efforts in software distribution and maintenance. SIA breaks this tradition by offering an automated code distribution system (<http://seisweb.usask.ca/SIA/cs.php>) modeled after open-source projects such as apt-get and yum. This model allows each installation to configure a list of repositories, which will be checked for updates to currently installed or new tools. If updates are available, the user is notified through the GUI and is provided with their descriptions. When an update is selected to be installed, the source code is downloaded from the server and compiled on the local system. The entire process is automated and controlled from within the GUI, or it can be performed from a command line. By downloading source codes rather than binaries, the system is able to share tools across many supported architectures. Further, the code is compiled optimized for the hardware it is running on (i.e. AMD, Intel, or PowerPC) ensuring the best possible performance. The ability to install and update code is restricted to “administrative” users, which may be useful where there is a single installation for a large number of users.

Typically, researchers develop interesting geophysical processing codes, but often it takes significant effort for others to install the necessary environment and to learn to use it. Well-supported open-source code standardization could allow codes to be reused by the community and shared in a consistent, reliable, and architecture-independent manner. SIA accomplishes this by allowing any installation to also function as a code server if a standard web server is available on the system. In such a way, tools developed locally become immediately available for installation on all subscribing systems. Finally, the author of a new tool can easily arrange for automatic “bug reports” related to that tool to be received by the code web service above.

### **Conclusion**

SIA is currently a full-featured seismic processing system, which could be of interest to researchers in both academia and industry. Its strengths are in its unique processing concept, broad scope, modern interface, robust core, very general visualization system, and parallelization capabilities. Since new ideas in seismic processing constantly require new software, SIA is optimized to serve as a concurrent development framework allowing new processing tools to be rapidly developed while leveraging the existing code and graphical utilities to dramatically reduce the time and effort required. The display system seamlessly handles both 2D and 3D data while offering some unique features and allowing extensive customization by the user without the need for programming. A code update and distribution system provides easy and automated access to software updates and allows developers to share their work without the need for installation or maintenance utilities.

### **References Cited**

Chubak, G., and I. Morozov, 2006, Integrated software framework for processing of geophysical data: *Computers & Geosciences*, v. 32/6, p. 767-775.

Morozov, I.B., and S.B. Smithson, 1997, A new system for multicomponent seismic processing: *Computers & Geosciences*, v. 23, p. 689-696.

Morozov, I., B. Reilkoff, and G. Chubak, 2006, A generalized web service model for geophysical data processing and modeling: *Computers and Geosciences*, v. 32/9, p. 1403-1410.