A New Seismic and Geophysical Data Processing System

Igor Morozov and Glenn Chubak, University of Saskatchewan, Saskatoon, Canada

Summary

We summarize the recent development of our broad-range seismic and geophysical data processing system. The system implements a novel data handling and analysis framework, currently includes ~200 tools and allows inclusion of unlimited number and types of algorithms. It provides standardized input/output, common parameterization of all tools, common graphical look and feel, automated maintenance, and distributed, cross-platform functionally. Its algorithmic design and throughput are similar to those of commercial seismic processing systems while its architecture supports significantly more general processing, such as handling multicomponent, WARRP, and earthquake seismic records, support for custom travel-time and other databases, creation of publication-quality PostScript graphics, and integrated support for non-seismic (e.g., 2-D or 3-D gravity) applications.

SIA

Open-source seismic processing provides flexibility, functionality and value often not found in commercial packages. A seismic processing system called SIA (http://w3.uwyo.edu/~seismic/sia), initially developed at the University of Wyoming (Morozov and Smithson 1997) and continued at the University of Saskatchewan, represents one of the most extensive efforts to integrate general seismic data analysis with polish and performance of a commercial seismic processor. At present, SIA offers capabilities for nearly complete reflection processing, combined with built-in support for multicomponent, variable-format seismic data, extensive database capabilities, I/O in several formats (e.g., exploration formats: SEG-2, SEGY, PASSCAL-SEGY, Seismic UNIX, and earthquake data formats GSE3.0 and SAC), original inversion codes (e.g., 2-D and 3-D reflection and receiver function migration, genetic algorithms and artificial neural networks), maintenance utilities, and interfaces to popular packages such as Datascope, GMT (Wessel and Smith, 1990), Seismic UNIX (SU), and also some key programs, such as *rayinvr* (2-D ray tracing by Zelt and Smith, 1992) and *reflectivity* (1-D waveform modeling, Fuchs and Műller, 1971).

The data and processing sequence organization in SIA is shown in Figure 1 (modified from Morozov and Smithson, 1997). Processing tools (modules) are provided with their "input" and "output" structured trace ensemble gathers and are free to perform any operations with them. The tools usually pass certain numbers of traces into their outputs while performing the required operations, but they also can create additional gathers, access (create, delete or modify) database tables, travel-time curves, velocity models or other objects, or communicate with other tools (Figure 1). As in commercial systems or SU, tools can initiate X-windows sessions and communicate with the user or access external devices. A module can also wait until, e.g., a certain number of traces is accumulated at its input.

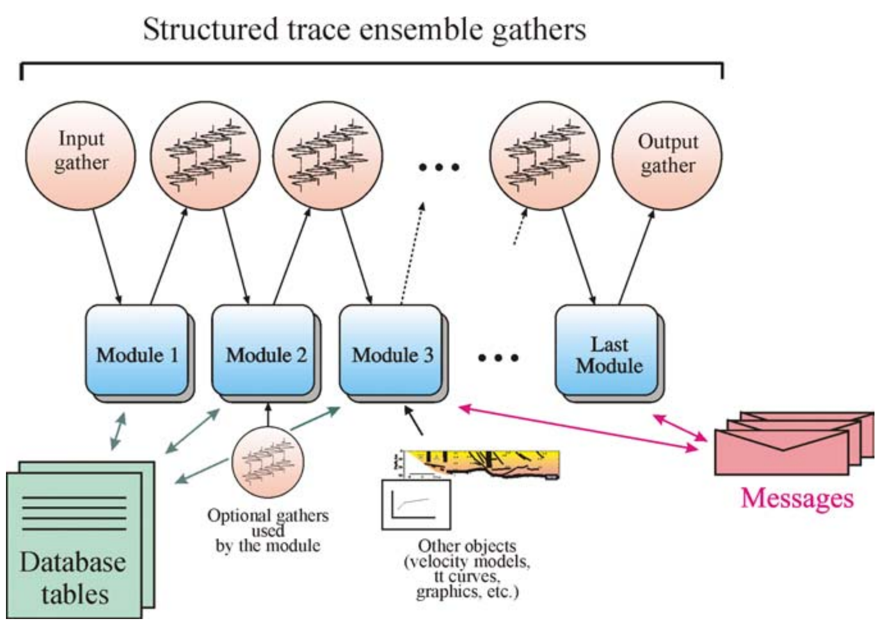

Figure 1. Data and processing flow implementation in SIA. Each processing tool (module) is presented with structured data input and output, as well as with access to a variety of custom databases and data objects. The input and output gathers are usually empty, but they can be used to exchange data with other similar flows (e.g., executed remotely).

Unlike other "academic" seismic processing packages (such as the Seismic UNIX or SIOSEIS), SIA is not built around a particular data file format but emphasizes the processing logic instead (Figure 1). This allowed introduction of abstract data loading methods, such as reading from "datasets" comprising entire directory structures of varying file formats, files listed in database tables, or hierarchical lists of other datasets. Seismic traces can circulate freely (backward as well as forward) through the processing sequence (Figure 1) facilitating, in particular, formulation of flows with random, adaptive navigation within 3-D datasets (Morozov, 1998).

SIA is the only system (of which we are aware) allowing creation of processing sub-flows embedded in the main processing stream (e.g., to implement plotting or arbitrary processing within the Common Image Gathers during migration, as done by Morozov and Dueker, 2003). Sub-flows can be executed within the same address space or concurrently on the same or different processing nodes and exchanging the data through the network.

An important feature of the system is its integration with other open-source packages. Thus, the SIA graphics package generates plots either by using the X-windows (currently being upgraded into QT) or by using GMT programs (Wessel and Smith, 1990) which has become the de-facto standard for PostScript graphics in the academic community. Also, a special tool enables plugging Seismic UNIX (SU) processing flows into any point in SIA processes (Figure 1). In such a way, the functionality of SIA could be greatly expanded by, e.g., the rich library of modelling and migration codes available in SU.

Like in other similar projects (ITA, SEP, Seismic UNIX, SIOSEIS), the above flexibility of process parameterization until recently came at the expense of an intuitive and consistent graphical user interface. Processing jobs had to be described using specialized scripts (somewhat resembling DISCO jobs), which resulted in a significant learning curve. However, recently we enhanced the system with a modern graphical user interface (GUI) that streamlined the organisation of processing and also improved the management of parallel processing.

Graphical User Interface

SIA GUI (Fig. 1) is based on the freely available, cross-platform QT libraries from Trolltech. Currently only Linux-based systems are supported but other UNIX type systems such as Solaris, BSD, or Apple's OSX should work with minimal effort. The interface was designed to provide non-technical users access to the SIA processing capabilities, without the need to learn scripting or programming languages. The interface generally resembles that of ProMAX, the industry-leading reflection seismic processing system by Landmark Graphics. SIA GUI, however, is not a script builder but rather an integrated software package which allows interaction with the tools while building the flow. Along with its GUI-based functionality, the system also retains its full batch operation mode allowing execution of complex processing meta-tasks via UNIX shells. Additionally, scripts may be imported into the interface by simply dropping text into the flow where it is interpreted, inserted into the flow, and displayed graphically. At this point the new tool(s) can be interacted with in the same manner as those selected from within the interface. This added functionality allows the user to easily move between traditional script-based operation and the simpler GUI by using familiar drag and drop actions.

The interface organizes project data while hiding the system structure from the user (Figure 2). Multiple processing flows may be opened simultaneously allowing the user to edit and execute multiple jobs. Tools and configurations may be copied between jobs, saving the user time and reducing entry errors. Nearly 200 tools are arranged into packages (e.g., CMP, travel-time, earthquake data processing, gravity, or graphics) which may be tailored to meet the needs of a variety of users. Context-sensitive help can be accessed through the interface, in addition to the HTML documentation automatically generated by the system. Tool tips and status lines are used to identify options and features while reducing screen clutter.

Processing flows are constructed by simply dragging tools from the packages and dropping them at the appropriate place in the flow. Sample tool configurations are included in the help for common applications and can be dropped into the flow to speed flow construction. The tool can then be configured by modifying its parameters. Both tools and parameters can be rearranged by the drag and drop process, making it easy to correct mistakes. Defaults and drop-down selections are provided and are customizable by the user. Tools and various options can be commented by the user to document the flow construction. Context-dependent color highlighting and graphical representations are used wherever possible to improve readability of the parameters. Multiple monitors can be used effectively by allowing parameters to be displayed in a separate window from the flow construction.

Grid and Cluster Functionality

Processing jobs (Figures 1 and 2) are submitted through the Parallel Virtual Machine (PVM), which allows multiple processes to be initiated and controlled from a single interface. Jobs may be submitted either for parameter checks or for full processing, and are executed either locally or on a single or multiple remote systems in a heterogeneous computing environment. Sub-flows performing parts of data processing can be deployed on other compute nodes thereby distributing the processing load. Management of both local and remote processes is handled through the interface which reports errors and allows user interaction with the processes.

The GUI (Figure 2) also includes provisions for cluster scheduling, allowing processes to be executed on entire Beowulf clusters or on subsets of their nodes. The status of nodes including availability and processor load is displayed within the cluster management portion of the software. To allow users to share processing resources, jobs are automatically allocated to the least busy nodes.

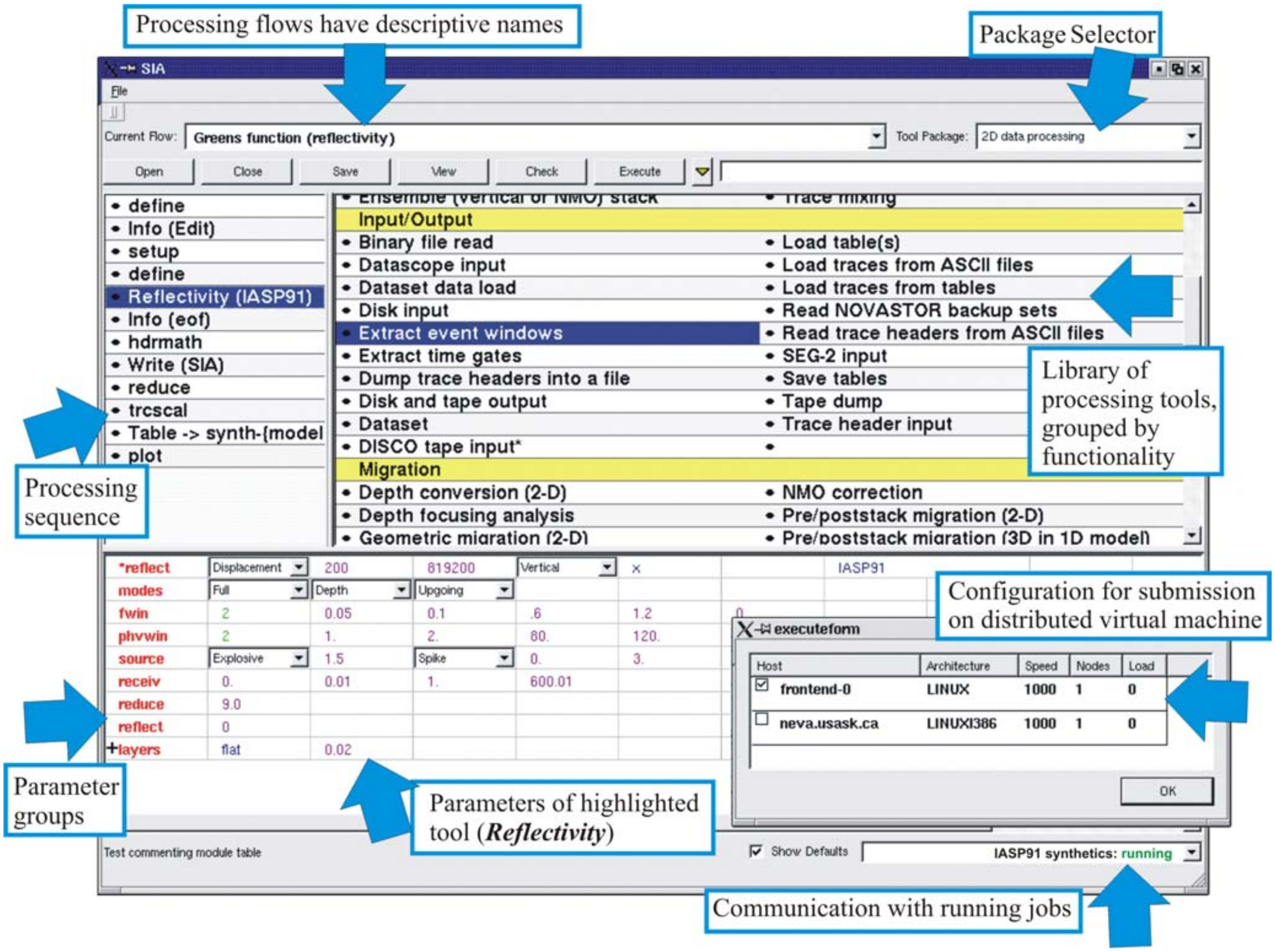

Figure 2. SIA Graphical User Interface. The GUI displays on-line documentation, status lines, tool tips, and allows keyword search for the desired tools and building multiple processing flows in the same session. Bottom pane provides an editor for the parameters of the highlighted tool. Processing tools are arranged into customisable "packages" (e.g., for CDP, RF, travel-time, or gravity inversion). Context highlighting (bottom window) is used to accentuate parameter types. Extensive drag-and-drop functionality and context-sensitive tool tips are implemented. When editing is complete, the job is submitted to a specified set of local or remote compute servers and is monitored on-line.

Development

Due to special efforts for package maintenance, code development for SIA is relatively easy. In our experience, a reasonably complex tool can be developed in 1-2 days. The addition of new tools does not require any modification to the monitoring program and can be done by the users. The tools are dynamically linked at run time from shared libraries, and thus there is no limit neither on the number of the tools nor on the types of operations they perform.

SIA includes utilities for compilation and maintenance of the codes, resolving library dependencies, and for documentation support. Libraries of C and Fortran subroutines and C++ classes are provided to facilitate development. The configuration of the system allows maintaining multiple versions for different computer architectures from a single set of source codes. Finally, provisions are made (work in progress) for automated maintenance of a knowledge base of processing examples and tips that could form the bases for an expert system that could assist the users in resolving processing difficulties and organize cataloguing of processing expertise.

Conclusion

With added GUI and PVM capabilities and improved ease of operation similar to that of ProMAX, the utility of SIA for the controlled-source community could broaden significantly. This package has been used successfully in several PASSCAL experiments, from initial pre-processing1 to final migration and preparation of publication-quality illustrations. With its shared UNIX libraries and PVM installed on multiple machines, grid computing and seamless data exchange could become a reality without any effort from the users. In passive-source work, which is quickly adopting reflection processing techniques, this industry-style processing could provide sophisticated, inexpensive, and high performance approaches to data retrieval and access to hundreds of signal enhancement tools developed in reflection seismics.

References

Fuchs, K., and G. Müller, 1971. Computation of synthetic seismograms with the reflectivity method and comparison with observations, J. R. Astronom. Soc., 23, 417-433.

Morozov, I. B. 1998. 3D seismic processing monitor, Computers & Geosciences, 24 (3), 285-288.

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

Morozov, I. B., and Dueker, K. G. 2003. Depth-domain processing of teleseismic receiver functions and generalized three-dimensional imaging, Bull. Seism. Soc. Am., 93, 1984-1993.

Wessel P., Smith, W. H. F., 1995. New version of the Generic Mapping Tools released. EOS Trans. Am. Geophys. U., 76, p. 329.

Zelt C.A. and R.B. Smith, 1992. Seismic travel-time inversion for 2-D crustal velocity structure, Geoph. Journal International, 108, 16-34.