Integrated Open Source Geophysical Data Processing

Glenn Chubak and Igor Morozov – University of Saskatchewan

Introduction

Open-source seismic processing provides a low-cost alternative to commercial software and, with an appropriately directed development, an ability to adapt to the changing research needs. The best-known examples of such kind are Stanford Exploration Project (SEP) software, SIOSEIS, and Seismic Un*x (SU), a free reflection processing system developed at the Colorado School of Mines (Stockwell, 1999). It has been broadly used in research and teaching seismology (e.g., Templeton & Gough, 1998) and also in smaller-scale seismic processing industry. However, all these packages are still strongly geared for CMP reflection processing, and their ability to handle more complex data is limited.

Here, we present our ongoing development of a system that also gives the flexibility, functionality, and value that are found neither in SU nor commercial packages. The system, called SIA, was initially developed at the University of Wyoming and continued at the University of Saskatchewan. It represents extensive efforts to integrate academic-style seismic data analysis with the polish and performance of a commercial seismic processor. In the following, we describe the development of the package since the previous publications (Morozov and Smithson, 1997; Morozov, 1998).

SIA Seismic and geophysical data processing system

Initially, SIA started as a replacement for Cogniseis DISCO processing system to support many DISCO processing modules written by the students of the Program for Crustal Studies at the University of Wyoming. Consequently, its key design requirements were typical of massive CMP processing: 1) high throughput achieved by means of processing tools (modules) operating in a common address space, with custom executables built for each job, 2) seismic processing sequences ("jobs") described using a specialized scripting language and executed in (normally) unattended processes, and 3) multi-user development and processing environment. In addition, several important extensions of the CMP data model were made: 1) multicomponent "trace gathers" as the key data structures allowing high flexibility of data access and 2) an original backpropagation call logic (Morozov and Smithson, 1997). The system supported (as it does now) job processing scripts similar to those of DISCO.

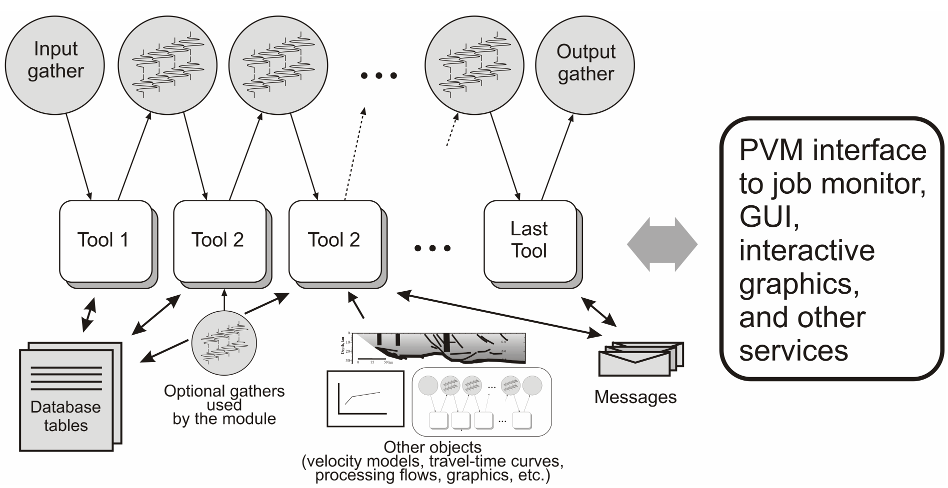

Figure 1 Structured trace ensemble gathers

Because the sequence of tool invocations in SIA is driven not by the input seismic data but rather by a logical inference mechanism (inspired by PROLOG programming language), no restrictions on the types of input data or character of processing were imposed. Data could be loaded, removed, or event directed backward in the processing sequence, or it could operate without input data at all (Morozov, 1998). Morozov and Smithson (1997) suggested that this system could be used in any seismic and even non-seismic applications. In the course of its application to several areas of geophysics (mainly WARR active-source and teleseismic seismology, and recently 3-D potential fields), additional features were implemented (Fig. 1):

1. Variable sampling intervals, record lengths, and trace time starts.
2. "Traces" can be linear arrays (seismic records) or 2- and 3-D arrays (multicomponent seismic records, or grids used in potential field processing).
3. Integrated general-purpose database system including lists and multidimensional arrays of entries structured as trace headers. These databases are used to represent survey geometry settings, travel-time tables, stacking and migration velocity models, gravity maps, and other ancillary data objects.
4. Custom data objects (such as velocity models, Artificial Neural Networks, or graphics elements) created and shared by groups of tools.
5. Graphics subsystem for rendering complex images in PostScript and building interactive Graphical User Interfaces (the latter, however, is currently being revised to incorporate modern Qt and Open GL functionality);
6. Dynamic linking at run time from shared libraries, with no limit either on the number of the tools nor on the types of operations they perform. This feature accelerates start-up and simplifies development; it could also be useful for porting older code that may require reconfiguring and recompiling before each use.
7. Parallelization of processing, with parts of the processing flow executed in concurrent processes on different compute nodes and exchanging data.
8. Maintenance utilities including automatically generated HTML documentation and tools to generate processing examples.

These features appear quite general, and not related specifically to CMP or WARR processing. Our experience suggests that the chosen code framework is able to accommodate virtually any type of (at least) geophysical data processing. The advantages from porting a code into this framework invariably were in achieving its uniform parameterization, access to numerous other tools (e.g., input/output or plotting), code maintenance, and improved documentation.

At present, SIA offers capabilities for nearly complete reflection seismic processing, combined with built-in support for multicomponent, variable-format data, extensive database capabilities, I/O in several formats (e.g., SEG-2, SEGY, PASSCAL-SEGY, GSE3.0, SAC), original inversion codes (e.g., 2-D and 3-D reflection and receiver function migration, genetic algorithms and artificial neural networks), and maintenance utilities. Interfaces to popular packages such as Datascope, Generic Mapping Tools (Wessel and Smith, 1995), *rayinvr* (Zelt & Smith, 1992), *reflectivity* (Fuchs & Müller, 1971), and Seismic Un*x.

Graphical User Interface

Recently, a modern graphical user interface (GUI) was designed for the system (Fig. 2). It was based on the freely available, cross-platform Qt libraries from Trolltech, the same libraries on which the popular KDE Linux interface is based. Using Qt allowed us to incorporate many of the most up-to date GUI design approaches, such as the multiple-document interface, window docking, themes, and platform-independent configuration. The interface presents the processing flow structure to the user and offers interactive means for communication with it (Fig. 2). Importantly, the SIA user interface (Fig. **Error! Reference source not found.**) is not merely a script builder but rather an integrated part of the run-time environment which allows interaction with the tools while building and executing the flows.

The **tool library** (Fig. 2a,b) offers access to over 220 processing tools, about 30 of which are to various degrees experimental. The tools are arranged into packages (e.g., CMP, travel-time, earthquake, potential field data processing, graphics, or development) which may be tailored by the administrators to meet the needs of a variety of users. Within each package, groups of tools (such as input/output, plotting, etc.) are displayed on tab panes (Fig. 2).

The **job editor** (Fig. 2c) is the central component of the user interface. A multiple-document interface allows several flows to be opened simultaneously, and so the user can edit and execute multiple jobs. Docking windows and tool bars allow a user to customize the layout of the program to make effective use of multi-display systems. Tools and configurations may be copied between jobs, saving the user time and reducing entry errors. Clipboard functions, tool tips, and context-sensitive help are provided to further improve the user experience. Several types of parameters are currently defined, and are rendered differently; for example, Boolean values are represented by check boxes, and selectable values – by drop-down lists. Color highlighting distinguishes between the floating point, integer, and character values. Additionally, errors issued from a running job cause the offending parameters to be highlighted and the corresponding error message displayed in the tool tip so that the user can quickly correct a problem.

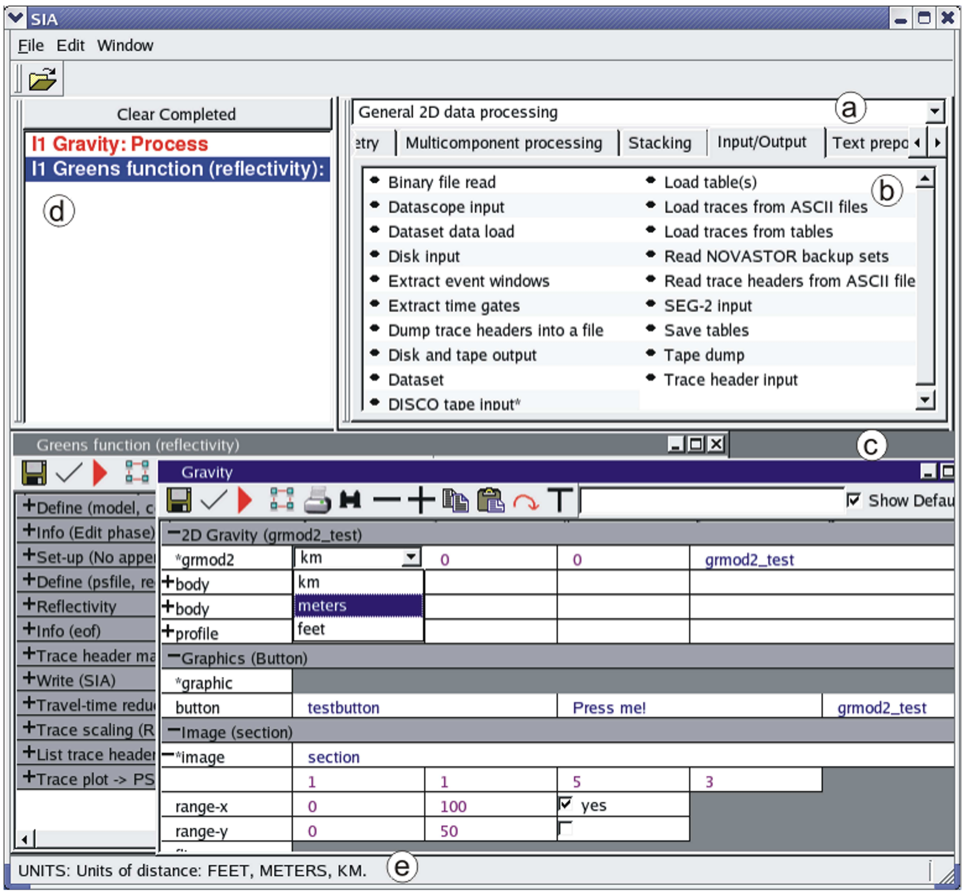

Figure 2 – Graphical User Interface

Job monitor. Processing jobs are submitted through a remote process communication interface utilizing the Parallel Virtual Machine (PVM) (which is similar to ProMAX). The system 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 executed either locally or on a single or multiple remote systems in a heterogeneous computing environment. Management of both local and remote processes is handled through the interface which reports errors and allows user interaction with the processes.

The GUI includes provisions for cluster scheduling, allowing processes to be executed on entire Beowulf clusters, subsets of their nodes, or local area networks connected to the PVM interface. A number of pre-defined and custom-named compute node configurations (as specified by the administrator) is available to each user for process submission. In addition, by including a tool called “cluster” in a processing flow, the user can also assign his/her own, named groups of nodes to the different sub-tasks of the flow. The statuses of nodes, including processor load can also be displayed within the cluster management portion of the software.

Discussion and conclusions

The scope of development and integration approach described above has grown beyond the limits of CMP seismic processing with which the package was originally associated. Although the GUI presented here is already quite rich in features, the content of the processing is still entirely controlled by the selection and parameterization of the tools (Fig.1). The tools, in their turn, are also not limited in the types of their operation (cf. Morozov and Smithson, 1997), and apparently nearly any computer application could be included into it, with the benefits of uniform parameterization, GUI, interaction with other tools, and unified software maintenance and documentation. Therefore, we view this system as a generalized software development and integration framework for geophysics.

Due to its modular design and scripts for the compilation and maintenance of code, development for SIA is unusually easy. Only a single C++ class and a description/documentation file needs to be prepared for a tool to become fully functional in the system. To implement a tool, no knowledge about monitor operation or presence of other tools is required. In our experience, a reasonably complex tool can usually be developed in 1-2 days.

Although the development of SIA is still far from being complete (for example, the interactive X-windows graphics is still in the works), it already represents a fully functional system exceeding its commercial analogs in many aspects important for University researchers. With ongoing further development, it will provide an excellent research tool and software development framework for many areas of fundamental and applied geophysics.

References

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

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

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

Stockwell, Jr. J. W., 1999, The CWP/SU: Seismic Un*x Package: Computers & Geosciences, May 1999.

Templeton, M. E. &. Gough, C.A., 1999, Web Seismic Un*x: Making seismic reflection processing more accessible: Computers & Geosciences, 25 (4), 285-288.

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. &. Smith, R.B., 1992, Seismic travel-time inversion for 2-D crustal velocity structure: Geoph. Journal International, 108, 16-34, 1992.