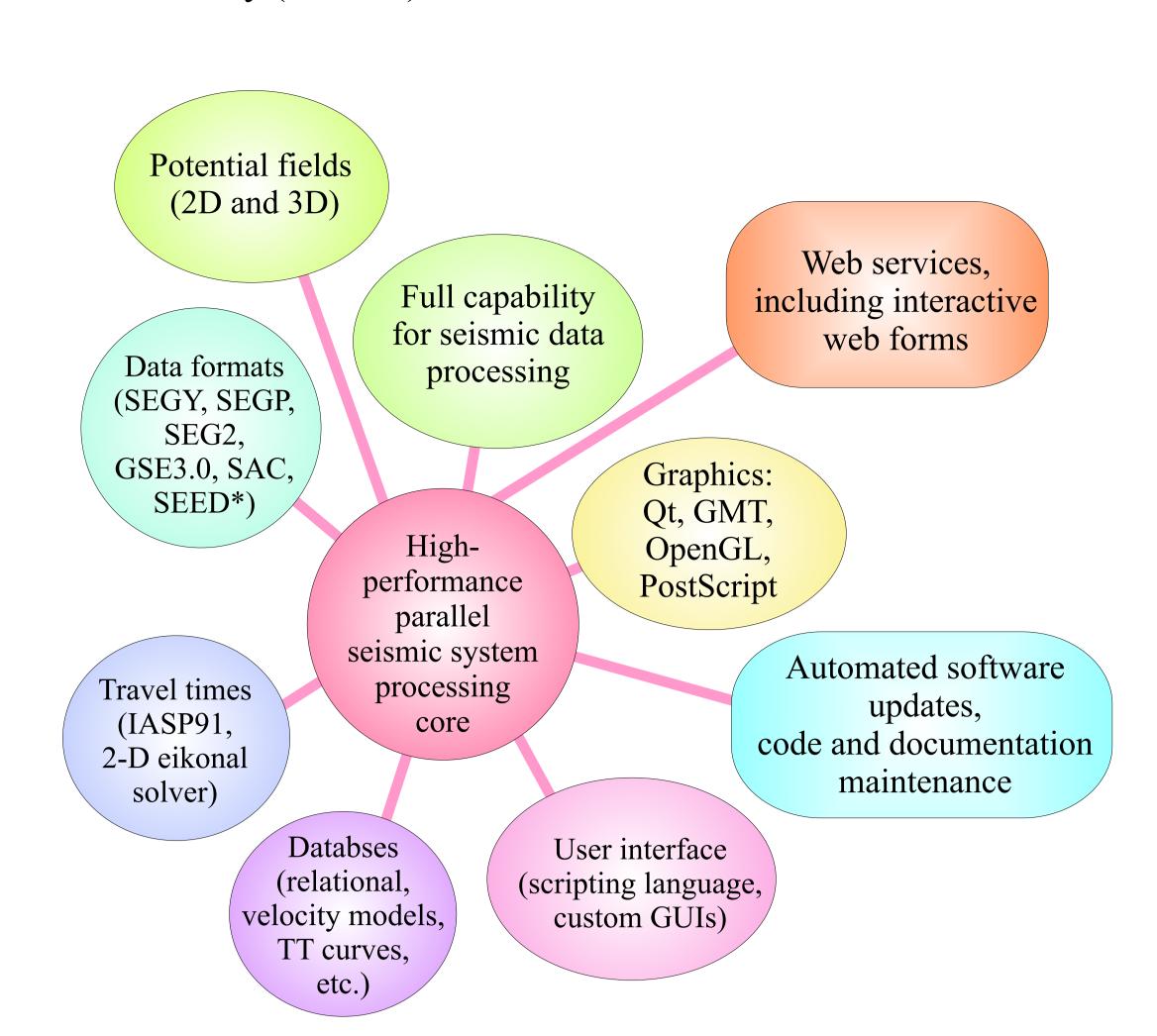
1. Scope

he system has been used in handling and processing of CDP (high-resoultion, GPR, and crustal), wide-angle reflection/refraction, earthquake, and non-seismic data. Currently, integration of 2-D and 3-D gravity modeling is underway (Panel 6).

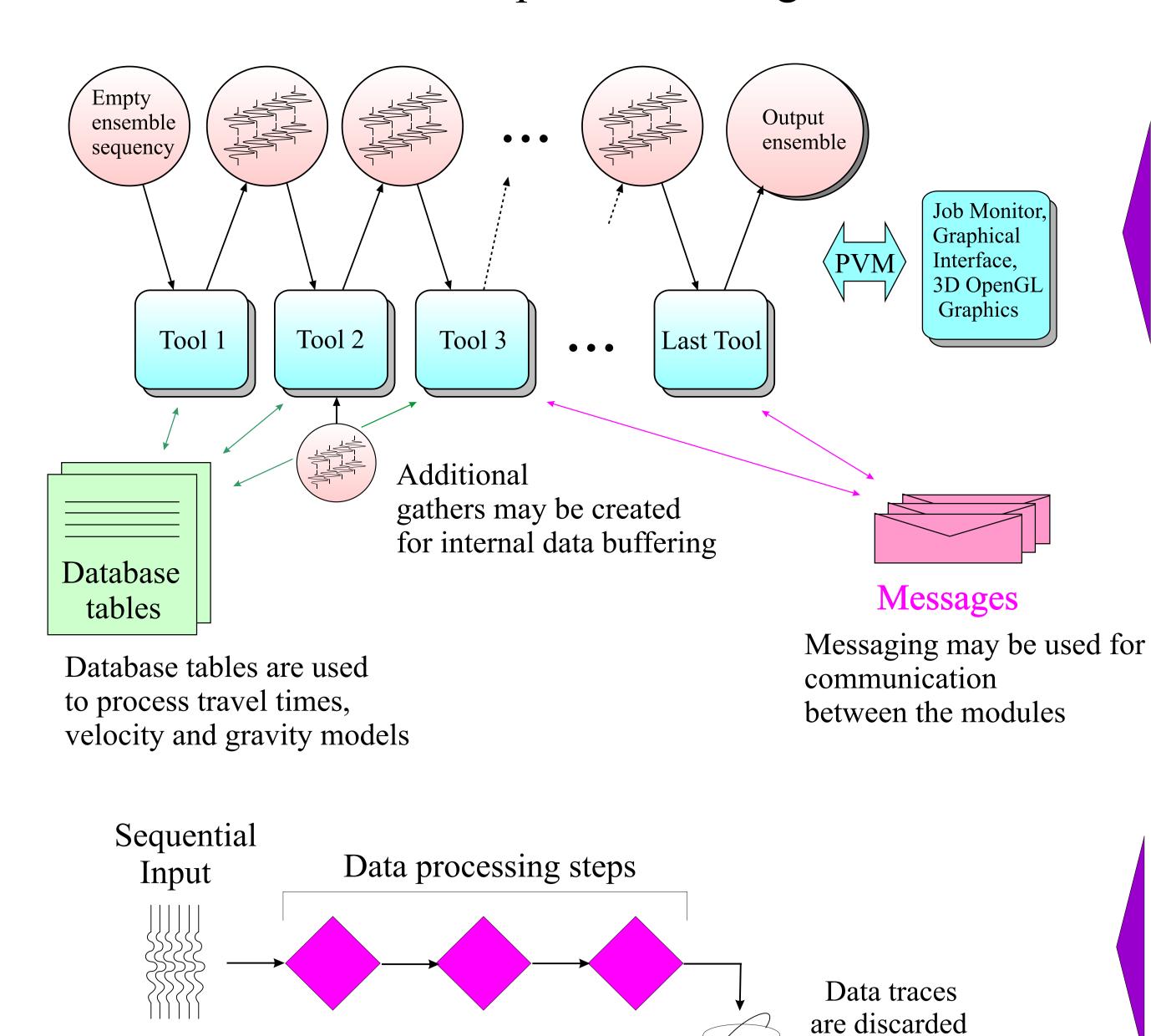


2. Processing Concept

after

the processing

Structured multicomponent "trace gathers"



Processing flow is arranged by the user and consists of modules sharing the program address space and working virtually concurrently. Most modules are written in C++ with large proportion of older tools implemented in C and Fortran. The sequence of tools and their parameters are described in a script (similar to that of Disco) or built using a Graphical User Interface.

The modules have access to structured trace ensemble gathers, databases, and are able to send data in any direction in the flow. Data buffering between the tools is implemented automatically, by the job monitor.

PVM (Parallel Virtual Machine) provides communication between the processing core and various components such as the graphical interface, Job monitor, and the 3D visualization program.

By contrast, in a typical CDP processing system (such as ProMAX, Disco-Focus, or SU), processing is driven by the sequence of data traces in the input file. The database structure is fixed and geared towards CDP processing.

3. Graphical User Interface

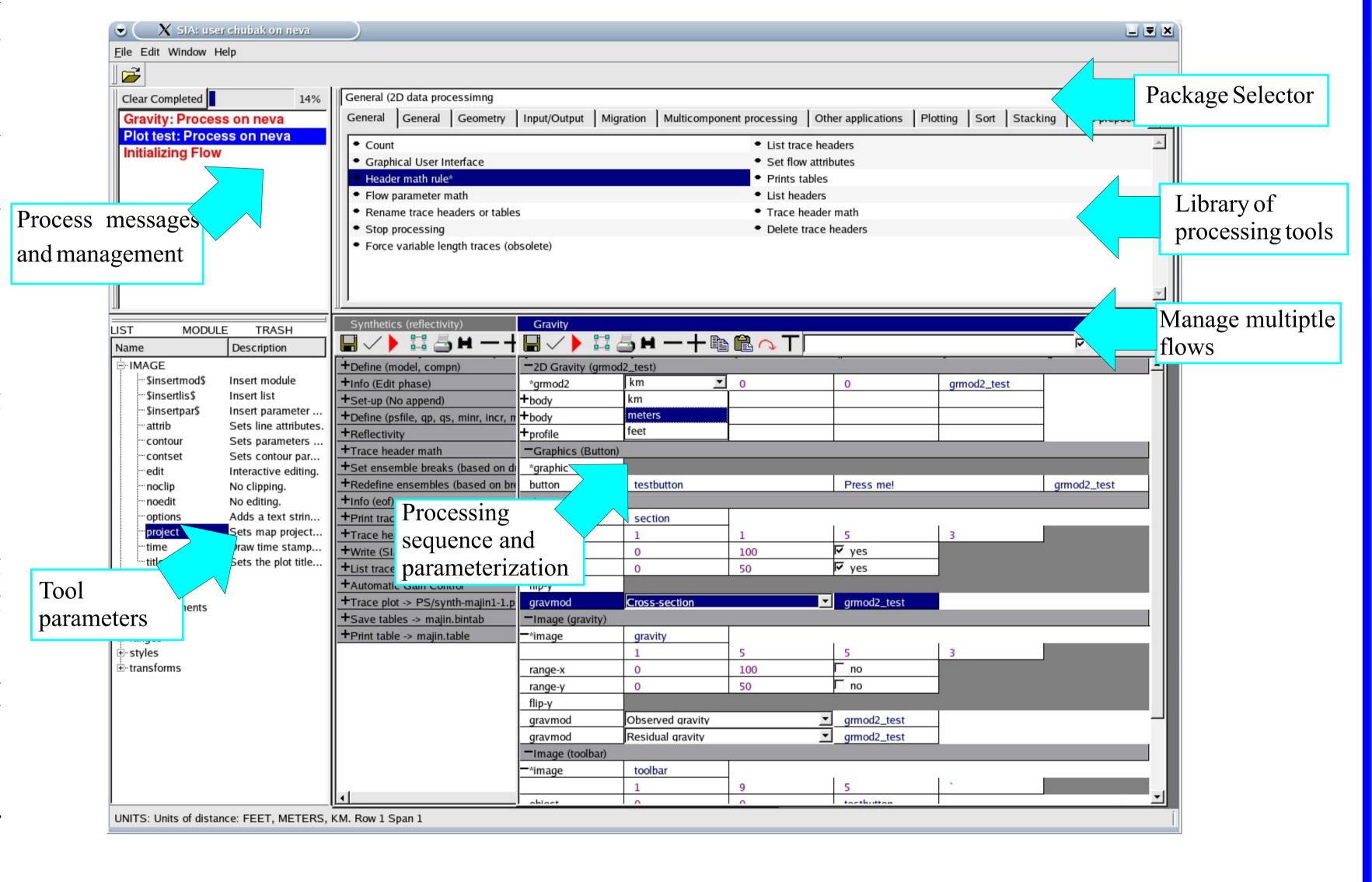
Apple's OSX should work with minimal effort. and features while reducing screen clutter. technical users access to the SIA processing capabilities, without the need to learn scripting or programming languages. The interface generally resembles that of ProMAX. SIA GUI, however, is not a script builder 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 Process messages tool(s) can be interacted with in the same and management manner as those selected from within the interface. This added functionality allows the user to easily move between traditional scriptbased operation and the simpler GUI by using familiar drag and drop actions.

he interface organizes project data while hiding the system structure from the user (Upper Right). 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., Reflection, travel-time, earthquake data processing, gravity, or graphics) which may be tailored to meet the

IA GUI (Fig. on the right) is based on the needs of a variety of users. Context-sensitive freely available, cross-platform QT help can be accessed through the interface, in libraries from Trolltech. Currently, only addition to the HTML documentation Linux-based systems are supported but other automatically generated by the system. Tool UNIX type systems such as Solaris, BSD, or tips and status lines are used to identify options

> included in the help for common applications parameters. Multiple monitors can be used modifying its parameters. Both tools and construction.

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 interface was designed to provide non-rocessing flows are constructed by the user to document the flow construction. dragging tools from the packages and Context-dependent color highlighting and dropping them at the appropriate place in graphical representations are used wherever flow. Sample tool configurations are possible to improve readability of the and can be dropped into the flow to speed flow effectively by allowing parameters to be construction. The tool can then be configured by displayed in a separate window from the flow



5. Development Framework

was to provide an extensible framework capable of supporting virtually class exercises and also to further their research. In our experience, a any type of geophysical data processing, modeling, or interpretation. reasonably complex tool can be completed in only a few days. Templates However, due to the character of the previous applications, most of SIA have been created to aid in the development process and a complete set of tool development so far was related to seismology.

processing while taking advantage of other tools and extensive C++ class in cluded in the auto-generated HTML tree libraries, including Qt graphics, and 2) custom interactive graphical (http://seisweb.usask.ca/SIA/index). applications can be created by simply designing processing flows and without any computer programming.

rom its inception, SIA was not intended as a complete product to \ \ \ \ ew modules can be coded using a mixture of C, C++, FORTRAN, serve a specific narrow task, such as, for example, reflection seismic and even Pascal or Java. At the University of Saskatchewan, processing (Morozov and Smithson, 1997). Instead, the design goal 👃 🔳 graduate students routinely write new processing modules for compilation and linking tools are provided. New modules are integrated he system allows its users to rapidly add new functionality with a with the system by the maintenance utilities so that they become available minimal effort. Two principal features simplify development from the graphical interface and provide fully functional context-sensitive within SIA: 1) new modules can be added to perform custom data help to the user. The documentation provided by the programmer is also

UNIVERSITY OF SASKATCHEWAN

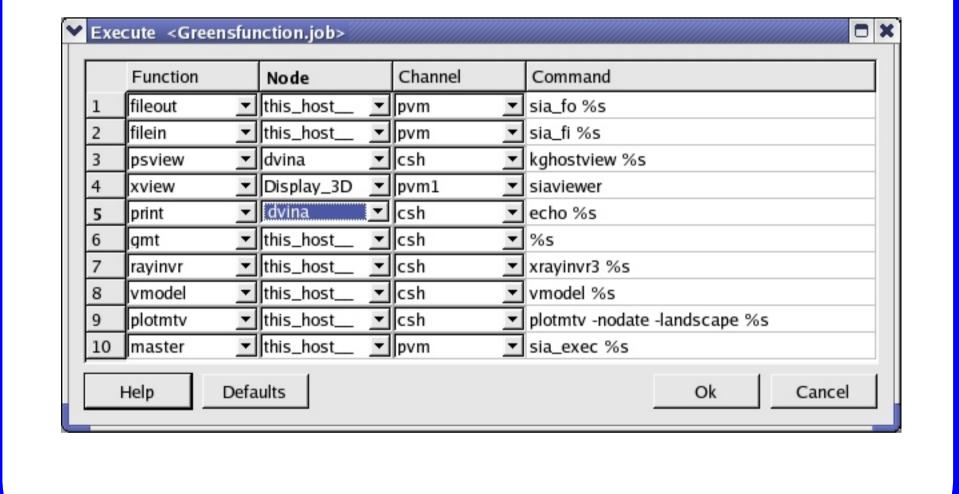
Introduction

or a number of years, we have been developing a new open-source package (called SIA) which serves as a tool for the processing and analysis of several types of geophysical data and also as highly integrated framework for developing geophysical applications software. Currently, the package consists of a processing core extending the capabilities of a seismic processing system, graphical interface, and customizable 3D OpenGL visualization tool similar to those used in seismic interpretation systems.

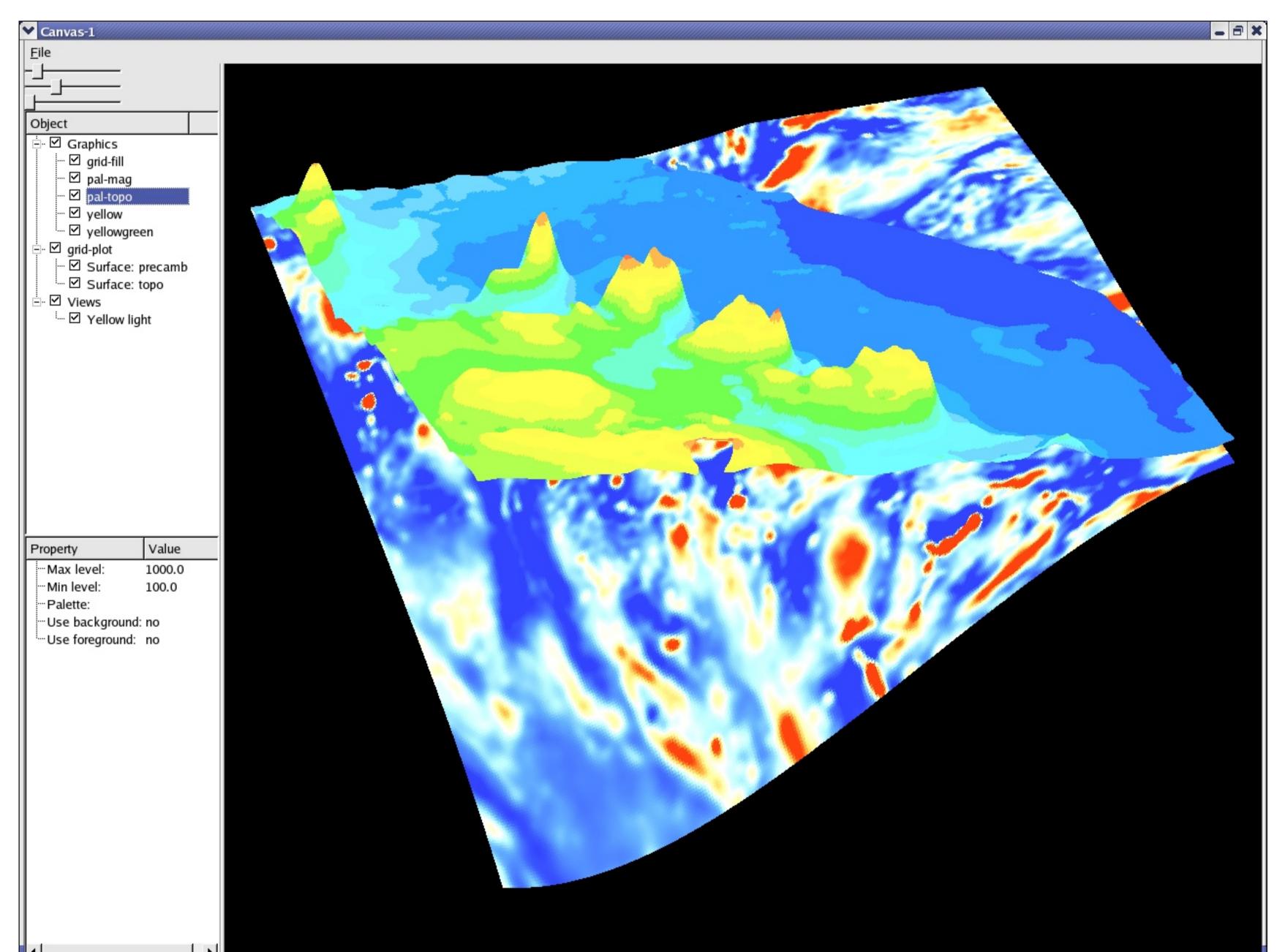
4. Grid and Cluster Functionality

rocessing jobs 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. Subflows 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.

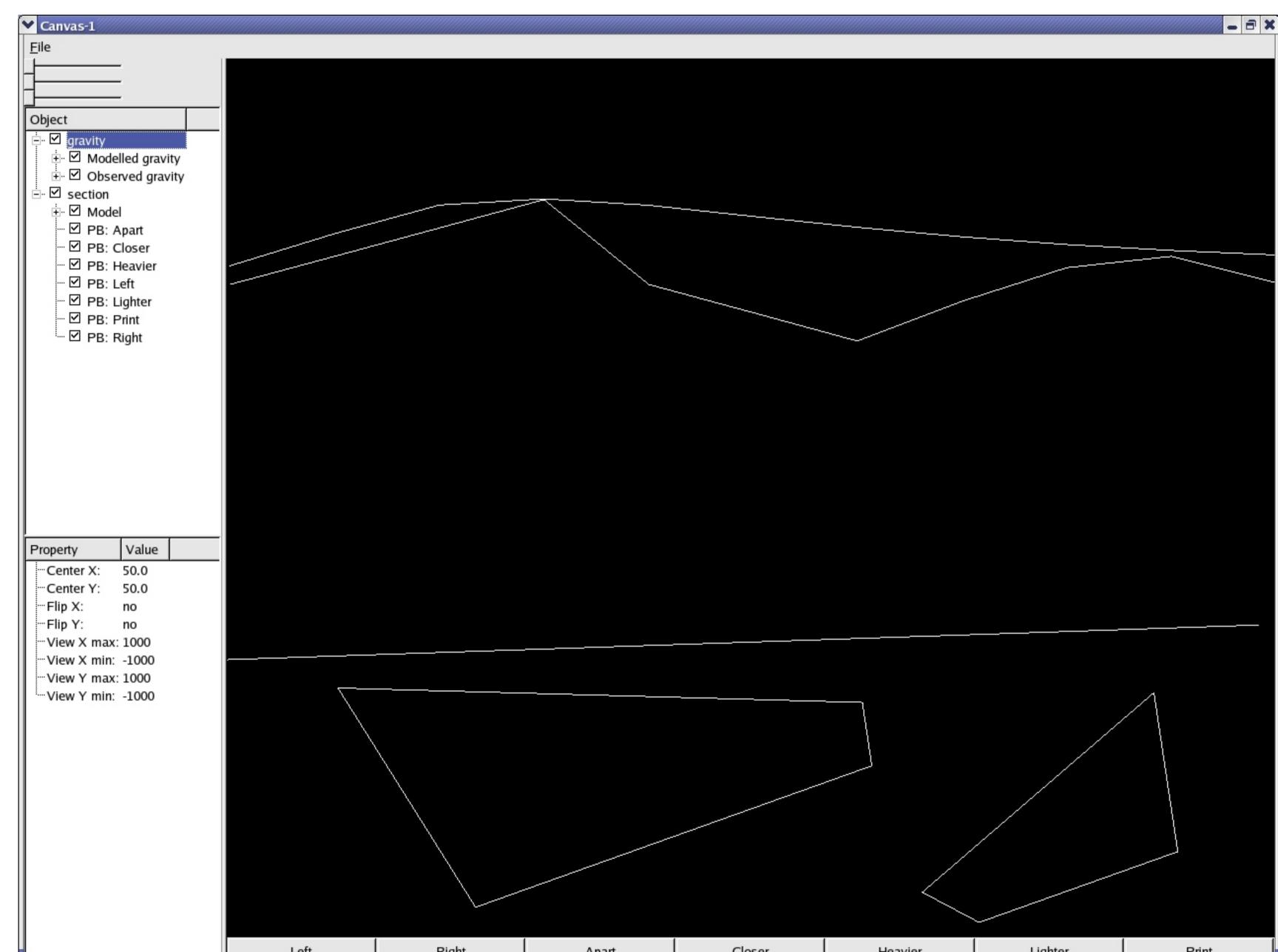
he GUI also includes provisions for cluster scheduling, allowing processes to be executed on entire Beowulf clusters or on subsets of their nodes. The status of nodes including availability and processor load is displayed within the cluster management portion of the software. To allow users to share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



6. 3D/2D OpenGL Visualization



3D rendering of topography and Precambrian basement from Saskatchewan and



Simple gravity modeling tool created entirely by a processing flow

isualization 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.

omplex objects (such as velocity and gravity models) are composed of the objects 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.

he visualization system (Figure 2) is 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. View direction and zoom are controlled by the mouse as it is done in other interpretation programs. OpenGL ensures that the system will work on a wide variety of systems and above by the corresponding tools. In addition, user-specified coordinate takes advantage of the hardware acceleration on graphics cards. It also allows native 3D transformations are available, so that images can be rendered on arbitrary rendering on stereoscopic displays, such as Geowall (http://geowall.geo.lsa.umich.edu/).