ELSEVIER

Computers & Geosciences 33 (2007) 835–837

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

Short note

# Automated maintenance of geophysical software from distributed web repositories ☆

Glenn Chubak*, Igor Morozov

*University of Saskatchewan, 114 Science Place, Saskatoon, SK, Canada S7N 5E2*

Received 12 January 2006; received in revised form 16 October 2006; accepted 25 October 2006

As in most computationally intensive disciplines, geophysical data analysis involves numerous algorithms. Large volumes of code have been created, including complex multi-function processing systems, which are particularly well developed in reflection seismology (Stockwell, 1999). In most cases, data management, processing, or modeling operations can be subdivided into smaller tasks (e.g., input/output, or some filtering), whose code could be standardized and reused. Ideally, good solutions to problems should be implemented once in a generic fashion so that others could benefit from them. Two critical issues arise in the development of such a general processing system: (a) a versatile code integration protocol and a common processing environment suitable for its use in different applications are required, and (b) with growing body of software, code maintenance tools are needed. Topic (a) above was recently discussed by Chubak and Morozov (2006); in this note, we describe the development of topic (b) in our geophysical data processing system.

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. Typical codes are developed by a single group, relatively compact, and can be directly exchanged by the researchers. However, in the more general, complex, and extensively developed packages used by numerous researchers (such as SU and GMT—Stockwell, 1999; Wessel and Smith, 1995), the need for consistent distribution support is already felt, leading to development of installation web sites and shell scripts.

Complex software packages quickly become difficult to maintain. For example, Seismic Un*x (SU; Stockwell, 1999) consists of several hundred programs that must be installed to use the package. The SIA system (Morozov and Smithson, 1997; Morozov, 1998) includes over 100 modules in the dynamically shared library, over 200 tools written in a variety of languages, and numerous documentation files. Each piece of software may have its own prerequisites (PVM, graphics, third-party software, etc.), compiler options and other configuration issues. Installation and maintenance of such packages represents a significant investment of time and effort from the user. The traditional approach of using a `configure` script and the `make` utility to assist the user in the installation could become cumbersome as it is not designed for the diversity of code found in processing packages, nor does it address the need to update only certain code without

☆Code available from server at http://seisweb.usask.ca/SIA/cs.php.

*Corresponding author. Fax: +1 306 966 8593.

*E-mail addresses:* glenn.chubak@usask.ca, gdc178@mail.usask.ca (G. Chubak).

0098-3004/$ - see front matter 
doi:10.1016/j.cageo.2006.10.007

affecting the entire system. Complex software systems thus require sets of specialized utilities which could automate maintenance and simplify installation, ideally by means of a web-based update service keeping the codes up to date as they are being developed at multiple sites.

Automated code updates are broadly used in modern software (such as Microsoft Windows or Adobe Acrobat). The open-source (particularly Linux) community is addressing the broader needs of updating and maintaining programs by using multiple software repositories. Programs such as `apt`, `yum`, `urpmi`, and `emerge` provide the ability to easily update and install software on several types of Linux systems. Using this model, we have implemented an automatic update and installation tools for the SIA system (Morozov and Smithson, 1997).

SIA represents a major effort for providing a common framework for data management and processing encountered in nearly any field of geophysics. The system is infinitely scalable, high data pass-through, capable of extensive seismic, travel-time, and potential-field processing. It includes a feature-rich Graphical User Interface (GUI, Chubak and Morozov, 2006), interfaces to popular academic applications (such as SU and GMT), capabilities for parallel computations, can operate as a web service (Morozov et al., 2006), and the development of a 2-/3-D OpenGL graphics layer is underway. In order to streamline code maintenance and to enable collaborative code development, a set of utilities was added to allow users and developers to effortlessly share their code with the community. These codes are being currently used to synchronise the software versions in our group and also for recent distributions.

The new SIA code maintenance package includes four key utilities:

(1) Program `sia-config` provides code customization for the current system. It allows specification of the compilers and their switches for the various phases of building the codes.
(2) Program `sia-update` is the general code maintenance utility performing packing and unpacking of the specified source code components, their building and installation. When installed on a web server, the program also executes most of the code maintenance server requests (`http://seisweb.usask.ca/SIA/cs.php`). On a client, when called with the appropriate switches, the code also tests the resulting binary codes, adds users, generates lists of code repositories and performs file cleanup.
(3) Program `sia-install` is the command-line code installer. It obtains the specified components of the package through a web service and installs it by using `sia-update`. For example, command `sia-install http://seisweb.usask.ca –distribution CG .all` installs the "Computers and Geosciences" (CG) subset from our web server.
(4) Utility `pdf` registers new plug-in tools with the system (Morozov and Smithson, 1997). This registration includes creating UNIX `make` files, parameter descriptors, GUI menus, and documentation web pages (http://seisweb.usask.ca/SIA//index/).

A characteristic feature of our model is that the SIA code is configured identically on all the code repository servers and processing clients. Code servers are therefore also capable of performing full data processing, including remote processing as a web service (`http://seisweb.usask.ca/SIA/ps.php`; Morozov et al., 2006). Conversely, if a standard web server is available on a system used, for example, for specialized data processing and development of the corresponding tools, it can automatically share these tools with others. As a code is updated or added, all clients which connect to this repository will immediately (as soon as the version number is advanced) have access to it. Such symmetrical design makes installation and maintenance of multiple copies of the package easy and reliable.

In the SIA GUI, users responsible for "administrative" tasks are able to add the URL of any code repositories (Fig. 1) they are interested in. Upon launch of the GUI, the update client builds a list of


Fig. 1. Configuration of SIA software repositories. Note that only root entry points are shown as web addresses. For example, actual code server for selected line is `http://chubak.ca/SIA/cs.php`. Buttons below allow user to edit list.


Fig. 2. Choosing software components to update. Note that components whose names begin with a period are system libraries or configuration directories, and the rest are plug-in processing tools (Morozov and Smithson, 1997). For each component, its current and updated version numbers, and source of update are displayed. User can select some or all components which will be downloaded, compiled, and installed.

locally installed SIA packages including their version information. It then obtains a similar list from each of the servers in the repository list (Fig. 1). The versions are compared and the user is notified if new packages are available or if updates to already installed packages have been made. If the user chooses to update or install a software component (Fig. 2), the source code is downloaded from the appropriate repository and compiled by the `sia-update` utility on the local system. The downloaded code can consist of multiple files in various languages (C++, Fortran, or Java). This utility takes care of all aspects of the installation including generating the `make` files, documentation, and ensuring that the resulting code is optimized for the local architecture.

In conclusion, the ongoing development of the SIA code framework shows that the entire scope of critical issues facing geophysical data management and processing can be solved in a consistent manner. The codes are highly integrated, streamlined for data- or computationally intensive seismic and non-seismic processing and modeling, make broad provisions for parallelization and remote (web service) operation, and incorporate some of the key community software. With the newly developed web distribution service, the codes can also be developed by multiple authors and seamlessly maintained up to date.

## References

Chubak, G., Morozov, I.B., 2006. Integrated software framework for processing of geophysical data. Computers & Geosciences 32 (6), 767–775.

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

Morozov, I.B., Smithson, S.B., 1997. A new system for multicomponent seismic processing. Computers & Geosciences 23 (6), 689–696.

Morozov, I.B., Reilkoff, B., Chubak, G., 2006. A generalized web service model for geophysical data processing and modeling. Computers & Geosciences 32 (9), 1403–1410.

Stockwell Jr., J.W., 1999. The CWP/SU: Seismic Un*x Package. Computers & Geosciences 25 (4), 415–419.

Wessel, P., Smith, W.H.F., 1995. New version of the Generic Mapping Tools released. EOS, Transactions American Geophysical Union 76, 329.