Fermi National Accelerator Laboratory

FERMILAB-Conf-94/093

The Distributed Development Environment for SDSS Software

E. Berman, V. Gurbani, B. Mackinnon, H. Newberg, T. Nicinski, D. Petravick, R. Pordes,
G. Sergey and C. Stoughton

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

R. Lupton
Department of Astrophysical Sciences, Princeton University
Princeton, New Jersey

April 1994

Presented at the *Conference on Computing in High Energy Physics 94*,
San Francisco, California, April 21-27, 1994

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CHO3000 with the United States Department of Energy

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The Distributed Development Environment for SDSS Software [*]

Eileen Berman, Vijay Gurbani, Bryan Mackinnon, Heidi Newberg, Tom Nicinski
Don Petravick, Ruth Pordes, Gary Sergey, and Chris Stoughton
Computing Division
Fermi National Accelerator Laboratory
Batavia IL, 60510

Robert Lupton
Department of Astrophysical Sciences
Princeton University
Princeton, NJ

Abstract

We present an integrated science software development environment, code maintenance and support system for the Sloan Digital Sky Survey (SDSS) now being actively used throughout the collaboration.

INTRODUCTION

The SDSS is a collaborative effort between Fermi National Accelerator Laboratory (Fermilab), the U. of Chicago, Princeton University, the Institute for Advanced Study (at Princeton), The John Hopkins University, U. of Washington, the U.S. Naval Observatory and the Japan Promotion Group. Its main results will be an imaging survey of $10^4 deg^2$ and a red shift spectroscopic survey of 10^6 galaxies and 10^5 quasars producing approximately 1.2×10^{13} bytes of data over the 5 year running period (1995 - 2000). This will produce a three dimensional map of the Universe.

Software Development Environment

The Fermilab Computing Division supports a set of standard tools and software for development of the SDSS data processing and analysis code. Survey standards are documented for : portable software environments, ANSI C, C++, FORTRAN, writing scripts, accessing survey databases, *make* methodologies, product structure and source code use. These standards were created to aid in the development and maintenance of a unified software system written by many people at different institutions. The goal of the standards is to meet the requirements of the survey and collaboration, be sufficiently benign as to be universally adopted and followed, be maintained, supported and extended with the available resources, and be expected to allow the needed evolution as the survey progresses.

In order to help accomplish the above goals, the following set of standard development tools were chosen.

RCVS

[*] Sponsored by DOE Contract number DE-AC02-76CH03000 and the Alfred P. Sloan Foundation.

Remote extension of Concurrent Version System (RCVS) was written by Terry Hung at SLAC[3] and is built on top of CVS (written by Per Cederqvist[4]). It provides for source code management across a wide-area network.

In general RCVS has supplied adequate functionality to support concurrent software development by multiple collaborators at physically distinct sites. We use RCVS for maintenance of all software used by SDSS (which includes approximately 1 million lines of code). Through RCVS any collaborator may obtain a copy of any SDSS software product. Access restrictions limit those who may include their changes back into the software. All modifications made to the software are logged within RCVS. RCVS supports a complete revision control system with script invocation on most of the RCVS commands, tags and branches.

Most of our users operate effectively using only a few commands, and consult with an expert when necessary. RCVS is a complex system and as such has a fairly steep learning curve when trying to understand it at a deeper level (i.e. tags).

Some of RCVS's drawbacks include possible use of much disk space since multiple copies of a product may exist simultaneously, network speed limitations, and dependencies on specific versions of other layered products (gnu diff, rcs). We have always maintained a good working relationship with the RCVS maintainer and any bugs we have encountered have been readily fixed.

UPS

Unix Product Support[5] (UPS) was written at Fermilab. It is a configuration management tool providing support for creating and maintaining multiple versions of software products on different Unix platforms.

UPS provides good support for handling the various products that are part of SDSS software. All software written for the survey is packaged in Fermi product form and is accessed via UPS. UPS includes a mechanism for registering 'dependent' products along with the main product so that specific versions of the dependencies are linked with a specific version of the main product. UPS is easily expandable and several suggestions made by SDSS developers have already been implemented.

UPS shell commands are not intuitive. UPS also depends on Fermi utilities (funame and dropit) that are not part of the UPS product.

UPR

Unix Product Retrieval[6] (UPR) was written at Fermilab. It provides a software distribution environment allowing remote users to copy versions of software to their nodes and performing installation of the software in UPS. If a product has dependent products, those versions are copied too.

UPR has been used successfully by remote collaborators to obtain copies of SDSS software. It is menu driven and requires no detailed knowledge to use. UPR insulates the remote user from having to know the details of UPS.

UPR is not proactive: the remote user must check for new versions of products. It does not inform the user how much space will be needed on the remote disk or check to see if the appropriate amount of space is available.

GCC

GCC is the Gnu C compiler written by Richard Stallman for the Free Software Foundation[7].

GCC has proved to be a useful tool for detecting software bugs. The compiler will give warnings on suspect C lines ignored by C compilers. Examination of these warnings has solved many bugs **before** the code was included in a software release.

In order to make GCC completely ANSI compliant, we have created companion software that is used when building with GCC. The product GCCFIX contains all ANSI supported functions and header files used by the survey software that are missing from the distributed GCC environment.

WWW

World Wide Web (WWW) is a wide-area hypermedia information retrieval system using a hypertext markup language (HTML) developed at Cern[2].

WWW has proved to be an invaluable tool for exchanging information between members of the collaboration. All of the SDSS software documentation is available via WWW. In addition we have made available software development help, user feedback (in the form of bug reports and user wish lists), and software version information. Using WWW, the entire collaboration has access to the most up-to-date documentation concerning SDSS software development.

There are many features of WWW (and HTML) that have aided in the establishment of this documentation environment. The small command set of HTML makes it easy to learn especially by example. The ability to immediately display the HTML files aids in quick debugging.

Writing documentation for display through WWW raises different issues and problems than when writing flat documentation. Since HTML documentation is not meant to be read cover to cover, each 'piece' needs to be written as a coherent whole that includes links to other documentation. We had to take care not to make our documentation too circular or the user would get lost on the web. The small number of HTML commands restricts how documents can be displayed, so one should learn HTML's restrictions before picturing what a document should look like.

TCL and TK

Tool Command Language (Tcl) is an extensible interpretive command language developed at the University of California, Berkeley by John Ousterhaut et. al. [8]. In addition we have integrated in Neosoft's Extended Tcl package[9].

In general we have had very good experience with building our software environment around Tcl. Using Tcl as the standard for our data processing software provides modularity and easy integration of separately written modules into a whole. The data analysis system consists of several hundred SDSS Tcl extensions written by many different people. This set of primitives provides a flexible framework on which the user can build specific analysis software, rapidly prototype, or work interactively.

The command interpreter is easy to learn and use. Procedures can be built up that are complex yet readable. It is good for making quick tests as there is no compilation step. Many of the SDSS collaborators develop using the Tcl extensions and then rewrite in C functions that run too slowly in Tcl.

The main problem we have had is that Tcl does not provide good support for mathematical operations. It does not do floating point arithmetic with complete accuracy. Therefore most mathematical computations we use need a Tcl extension linked to a C routine.

Tk (ToolKit) is an X Windows interface package integrated with Tcl and developed at Berkeley. Tk gives any user with Tcl competence the capability to build their own GUI.

Tk was used to help create the GUI for displaying system status information for the SDSS prototype, the Drift Scan Camera[10]. It is easier to use and learn than Motif. At the time, there was no interactive interface builder and precise layout was difficult.

Conclusion

We have had positive experiences with all of the tools described. Each is in daily use by many of the collaborators. The whole environment enables the scientists to integrate their code directly into the system. The standards enable testing scripts to be run on code without 'personal review'. Having the concept of a well defined and maintained environment early on has helped the software development process. We estimate that a total of approximately 1 year FTE (full time equivalent) was spent integrating all of the tools into a single environment and developing standards while the cost of developing a similar set of tools would take approximately 15 years FTE.

REFERENCES

1. " A Digital Sky Survey of the Northern Galactic Cap," Proposal, December 20, 1990.
2. T. J. Berners-Lee, R Cailliau, J. Groff, and B. Pollermann, "World-Wide Web: The Information Universe," Electronic Networking: Research, Applications and Policy, Vol. 2 No. 1, pp. 52-58 Spring 1992, Meckler Publishing, Westport, CT, USA.
3. Terry Hung, "RCVS: Remote extension of concurrent Versions System," unpublished document, Stanford Linear Accelerator Center
4. Per Cederqvist, "Version Management with CVS," unpublished manual, Signum Support AB, March 1993.
5. Fermilab Computing Division, "UPS User's Guide," unpublished manual, PN-426, Fermilab, March 1993.
6. V. Gurbani, "Product Distribution Via UPR," unpublished manual, Fermilab, December 1993.
7. R. Stallman, "Using and Porting GNU CC," unpublished manual, Free Software Foundation, May 1993.
8. J. Ousterhaut, et. al., "Tcl: an Embeddable Command Language," Proceedings of the Winter 1990 USENIX Conference.
9. K. Lehenbauer and M. Diekhans, "Extended Tcl Command Set," unpublished manual page, Neosoft, Inc., January 1992.
10. B. Mackinnon et. al., "Data Acquisition Systems for the Sloan Digital Sky Survey," IEEE Transactions on Nuclear Science, Vol. 41, No. 1, pp. 105-110, Feb. 1994.