Automated Tools for Information Systems Design
H.-J. Schneider and A.I. Wasserman (eds.)
North-Holland Publishing Company

A Software/Documentation Development Environment Built From The UNIX Toolkit

M. H. Bianchi

R. J. Glushko

J. R. Mashey

Bell Laboratories
Whippany, New Jersey 07981

ABSTRACT

The popular UNIX* System[1] contains a large kit of tools useful in software development. Toolkits are easy to learn, use, maintain, and adapt to different needs and organizations. A good set of tools works well together, but is not an integrated environment, on purpose[2]. An integrated environment is often optimal for a specific situation, and often difficult to move or adapt to new situations.

We have seen UNIX users develop their own integrated environments. Most customize the default environment by adding commands of their own. The early additions simply capture sequences of commands that they find themselves repeating. Later, they simplify their work environment by structuring it with directories that contain similar objects, such as subroutines, main routines, documents, control files, commands, etc. and then writing commands work within that structure. Because each organization views its world differently, customizations differ widely according to local needs. However, basic concepts keep appearing, even though the environments are quite different. We've watched the evolution of environments, observed the concepts they implemented, and generalized those concepts into a portable environment that overtly supports customization.

SOLID, a System for On-Line Information Development, is a more "integrated" programming and documentation environment built from the UNIX system's tools[3]. It implements the concepts of Source, where all the source for programs and documents is kept, Product, where all the executable programs and formatted documents reside, and Generation Procedures, which translate all types of Source into all forms of Product. These concepts are implemented as UNIX command language procedures that combine existing tools. SOLID also provides Skeletons, the most common outline form of each type of Source. The developer writing a new module of code or documentation does not start with an empty file. Instead, they are given an outline which reduces drudgery and painlessly enforces some standards of style and content.

SOLID is notable for what it does and what it doesn't do. On the "does" side, it provides a single environment that is being used to develop all documentation for computer systems, all programs, and mixtures of both. It comes with a default environment that is

---

* UNIX is a trademark of Bell Telephone Laboratories.

occasionally used unmodified.

On the other side, it is not a monolith that restricts the user to those options that are built in. Since all SOLID source is under SOLID control, the user can change or extend any of SOLID's capabilities. Some portions, in particular Generation Procedures, are intended to be extended to support new Source and Product types. It provides no editor; people use any that they like. It doesn't enforce any particular programming or documentation methodology, although Skeletons have proved useful in encouraging the methodologies and styles a group prefers.

Although released only recently for internal use, SOLID is now used by a dozen organizations at Bell Laboratories. Some use it for their documentation only, others for their entire product. Because it is based on observing the ways people have customized UNIX to suit their needs, it seems to match the way many groups do business. Because it is designed to be modifiable and is supported by its own mechanisms, people easily extend and customize it to suit their own needs. Because it stands firmly on standard UNIX, as UNIX becomes available on new or different hardware, SOLID is also. Finally, it preserves the tool orientation of UNIX. New capabilities added to UNIX are immediately available to SOLID users.

[1] Ritchie, D. M. and Thompson, K., "The UNIX Time-Sharing System", The Bell System Technical Journal 57, No. 6, Part 2 (July-August 1978) 1905-1929.

[2] Kernighan, B. W., and Mashey, J. R., "The UNIX Programming Environment", Computer 14, 4 (April 1981) 12-24.

[3] Glushko, R. J., and Bianchi, M. H., "On-Line Documentation: Integrated Development, Distribution, and Use", 10th Mid-year Meeting of ASIS, Durango, Colorado, May 14-16, 1981.