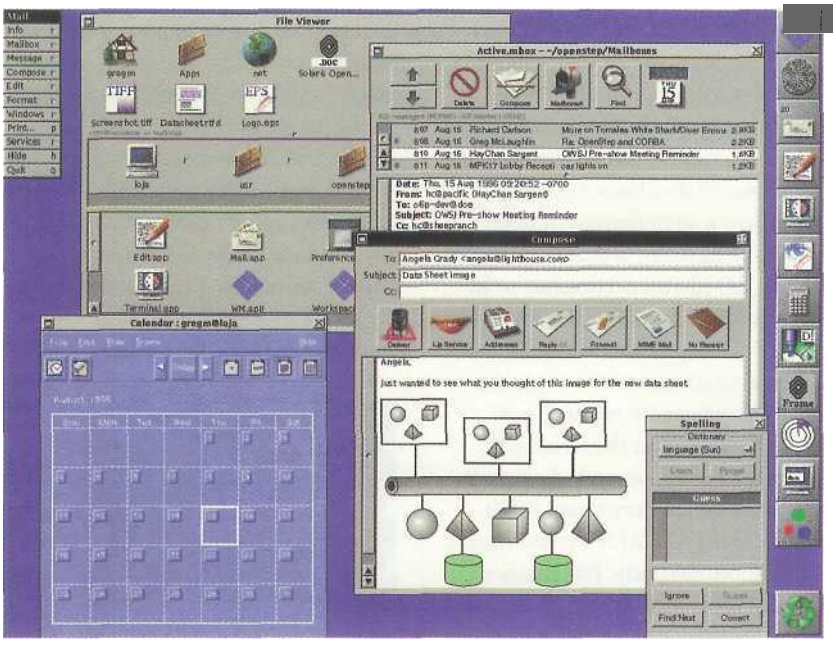
OpenStep for Solaris

by SIMSON L. GARFINKEL

More than three years after it was announced, Sun Microsystems Inc. has finally shipped OpenStep for Solaris. This new object-oriented system brings an easy-to-use user interface and an easy-to-program developer's environment to an operating system that has been sorely in need of both for years. OpenStep is a step in the right direction for Sun, and if the company supports this new environment, it could become a significant competitive advantage.

OpenStep is based on the technology that Sun acquired from NeXT Computer Inc. in 1993. The system is written in C, C++ and Objective-C (an object-oriented language that has an object model similar to Java's). It comes with a starter set of applications, including an editor, an email system, a PostScript previewer, an improved terminal application and a "preferences" application. And it's completely integrated with Solaris:



For NextStep lovers, OpenStep is like a dream come true: the full NextStep user interface, running on Sun hardware and able to run with today's Solaris applications.

OpenStep's cut-and-paste features interoperate with Solaris/X cut-and-paste, OpenStep applications can use Solaris printers, and Solaris icons show up on the OpenStep desktop.

Before continuing, I should say

that I'm not your typical unbiased reviewer. For four years, I used a computer running the NextStep operating system on a daily basis. During that time, I enjoyed the unparalleled ease of use that the operating system offers. I got used to administering my UNIX system completely from a graphical user interface and not feeling like I was being hamstrung by my tools. I got used to a multimedia email system that smoothly implemented drag-and-drop from every other graphical application. I came to rely on being able to paste PostScript graphics from my drawing program into my word processor. And I got accustomed to a terminal window that was fast and offered unlimited scrollback.

So I've seen the future. And for the past two years, I've been computationally homeless and stateless, having moved from the hospitality of NextStep to a strange and twisted combination of Solaris, Windows 95 and Mac OS. Nevertheless, I was forced to make the switch: NeXT's hardware was growing increasingly obsolete, while new programs that I needed to run were coming out for the other computers.

But now there's hope. For people like me who used NextStep, OpenStep is like a dream come true: the full NextStep user interface, running on Sun hardware and able to run with today's Solaris applications. For new users, this means that they can enjoy the wonders without having to give up Netscape Navigator, FrameMaker 5.0 and all of those other Solaris and Wabi apps.

But beware, OpenStep 1.1 comes with a few catches.

Evaluation

I loaded Solaris OpenStep 1.1 beta on a SPARCstation 10 with 32 MB of RAM. Although I disapprove of reviewing beta software, Version 1.0 was too difficult to install to be acceptable, and Version 1.1 final was not ready for review at press time. So I was forced to review a custom write-once CD-ROM that had been prepared for me by the OpenStep engineering team.

OpenStep 1.1 installs smoothly from CD-ROM by running a single installation script that's aptly named "install." This script figures out what packages you need to install and installs them automatically–no more manually adding Solaris packages. This is a step in the right direction; the rest of Sun should take note.

After rebooting, the CDE login panel gives you a choice of three windowing environments: OpenLook, OpenStep or CDE. Once you've tried them all, you're sure to be an OpenStep convert.

Log in and within a minute you'll see a window appear called "File Viewer." This is the OpenStep workspace manager. It's a window on your computer's file system, sort of like the Finder under the Macintosh operating system or the Explorer under Windows. (It's so much better than the OpenLook and CDE finders that

I didn't want to mention them in the same sentence.) But the OpenStep File Viewer is better than the competition.

For starters, there is a powerful browser view that lets you see both the path through the file system to the current directory as well as parts of all of the other directories that you followed on the way down. There's an icon view that's as easy to use as the Mac's.

In addition, there's a shelf at the top of the File Viewer where you can drag directories or files to hold them while you are working on them. (The shelf isn't a separate directory or anything like that; it's just a reference to the file system object that's remembered by the File Viewer program itself.) Click a directory or file that's in the shelf, and the File Viewer automatically jumps to the location of that object in the file system. It's sort of like setting presets on a car stereo.

Like most GUIs, OpenStep uses icons to represent both application programs and the documents edited *By* those applications. The icons appear in the File Viewer: You can launch an application by double-clicking on either kind. You can also drag the program icons into the "dock" that runs down the right-hand side of the screen. Unlike other systems, a single instance of an OpenStep application might have many windows–one for each document that you're editing. For example, you can have five different terminal windows up but be running only a single instance of the terminal application. This is actually much more memory-efficient than having a separate program instance for each. The OpenStep application kit makes it easy to write multiwindowed application programs that run without a hitch.

OpenStep Mail

OpenStep comes with an email client that shows off many of the OpenStep features. Called Mail, the application is virtually indistinguishable from NeXTmail Version 3.3. Back when NeXT cut its deal with Sun, NeXTmail was widely acknowledged as the best email application ever written. Although email programs like Eudora have nearly caught up in the past few years, OpenStep Mail is still among the best mail programs you can find.

OpenStep Mail is a traditional email application with messages, mailboxes and a "compose" window for creating new messages. Each mailbox is displayed in its own window, which is divided between a list of messages on the top and the message being viewed on the bottom. Messages can be easily transferred between mailboxes a number of different ways. For kicks, you can create a photos database to display the image of the person who sent you the mail message. Surprisingly, such a database goes a long way toward preventing flame wars within an organization: It helps you to remember that there is a person at the other end of the network, not just an email address.

Click a big "compose" button and you can write a new message. OpenStep Mail is completely MIME-compliant. You can send rich mail with boldface, italics and enclosed attachments. Attachments that are TIFF, EPS or any other understandable image format are displayed in place. If you are sent a file with attachments, you can double-click on the attachment's icon to launch the appropriate application program.

What's missing from OpenStep Mail is automated filters for siphoning email from mailing lists out of your main inbox and putting them in their own folders. But even without filters, I easily handled more than 300 incoming email messages a day with NeXTmail back in 1992. The user interface speeds email reading and writing like nothing that's been developed since.

OpenStep ships with a few other applications that both demonstrate the underlying technology and help you to use it. Edit is a multifont editing tool for manipulating both English and code. And Preferences is a single application for controlling systemwide preferences such as fonts, language (English, French, etc.), placement of menus, command-key equivalents, security, and keyboard and mouse. In addition, Preferences has a plug-in architecture so that other programs (or other companies) can add their own.

Application Kit

The OpenStep AppKit is the underlying technology that gives OpenStep programs their distinctive look and feel. AppKit contains many standard widgets: check boxes, radio buttons, push buttons, scrollbars and so forth. But each object has been tuned to make it a joy to use.

Consider the scrollbars. They're similar to the scrollbars in Windows 95: the scrollbars' "thumb" resizes to indicate the amount of the window that you can view, and they scroll their contents when you move them. (Where do you think that Microsoft got its idea for Windows 95 scrollbars?) But trust me: These are the most user-friendly scrollbars in the industry today.

For starters, the scrollbars are responsive; if you want to scroll two thirds of the way down in the document, just click two thirds of the way down in the scrollbar, and you're instantly there. If you want to scroll, you press the up or down buttons. It's easy to reverse direction because the buttons are next to each other, rather than at opposite ends of the scrollbar.

This attention to detail permeates the OpenStep experience. Click on a button in a matrix, change your mind and drag to another button, and that second button goes down, without the need to release the mouse button and click it again. Widgets use color to create a gentle 3D look–again similar to Windows 95.

OpenStep also features a complete "defaults" system which eliminates the need for application programs to create their own dot-files in the user's UNIX home directory. Because it is so easy to use from a programmer's point of view, OpenStep programs tend to be more customizable than programs written for other GUI environments.

The Catches

So if OpenStep is so good, what's the catch? OpenStep–at least the OpenStep 1.1 beta that I reviewed–is a serious memory hog. I tested it on a system with 32 MB of RAM, and it was all but unusable because of heavy swapping. My friend on the OpenStep developer team says that he hasn't noticed this problem with his system. The reason: he's got 64 MB of RAM.

Today's OpenStep is in fact a memory hog. The memory footprint of a typical set of applications is a whopping 88 MB. That's due to a lot of faults, including fundamental problems with the Solaris shared library system, memory problems in the X Window server, and unhealthy interactions between X and Display PostScript.

The second catch is that most of the wonders of OpenStep only work with OpenStep application programs. You can fire up a copy of Netscape Navigator under OpenStep, and it's got those ugly Motif scrollbars and menus. Well, at least it works. The real solution is to use existing Solaris applications as bridges until suitable OpenStep applications are written (or translated from NextStep).

But the biggest problem with OpenStep is that Sun's level of commitment to this new environment remains unclear. Over the past two years, there have been persistent rumors that Sun was about to cancel the OpenStep project. Now OpenStep has finally shipped, but it's unclear how it relates to CDE and Java. Sun made OpenStep 1.0 freely available for download on its Web server last summer. But the real success for OpenStep will only come if Sun makes it a standard part of Solaris. That will only happen if Sun's customers or its internal engineers try out this new environment and tell Sun's upper management that OpenStep is a better choice for the company's future than the alternative, such as CDE.

But if Sun is turning its back on C and UNIX entirely and building a new Java-based operating system, all bets could be off. In that event, I hope it bases the Java GUI not on AWT, but on OpenStep. It's a better technology.

OpenStep 1.1 for Solaris

Company
SunSoft Inc.
2550 Garcia Ave.
Mountain View, CA 94043

Phone
(512)345-2412

Fax
(512) 218-3866

Best Feature
NextStep look and feel running under Solaris.

Worst Feature
Poor performance in current release.

Price
$295

Circle 172