WHAT IS OBJECT-ORIENTED PROGRAMMING?
A brief Q & A on the next revolution
in computing technology
October, 1991

All of a sudden, everyone's talking about "object-oriented" programming and computer environments. What is object-oriented programming, and why the big deal?

Computer hardware has made great leaps in performance — almost doubling every two years — but improvements in software have developed at a comparative snail's pace. New hardware features remain untapped because software cannot be developed fast enough to take advantage of them, a phenomenon called the "software gap."

Object-oriented programming is a method of programming that greatly eases the process of developing and creating software. In fact, it is rapidly closing the software gap.

Instead of creating a program with lines and lines of procedural code, object-oriented programming produces chunks of discrete code — "objects" that perform specific functions. Programmers manipulate groups of objects, which send messages between each other, to create an application. Object-oriented programming improves development of software by creating reusable code with greater flexibility and reliability and by allowing programmers to add more complexity to applications.

Object-oriented programming is the first revolution in personal computing technology since the development of the graphical user interface. The ability to rapidly develop new programs may actually bring a productivity improvement to the workplace — a benefit that has eluded many businesses that invested heavily in personal computing technology.

Is object-oriented technology just more of those pie-in-the-sky, future technologies that will take years to become real?

In some cases, for the vendors just now beginning to develop object-oriented systems, it's true that their object-oriented products are years away. But NeXT Computer's complete line of professional workstations, which have been object-oriented from the start, offer all the benefits of this "new" approach, today.

Both commercial software developers and in-house corporate programmers are already using NeXT's object-oriented development environment, NeXTstep, to create exciting applications. Some of these applications are marketed to the general NeXT population, while others — the custom applications — are used by companies to gain a competitive

edge in industries ranging from financial trading and publishing to law, medicine and government.

For corporate programmers, the advances offered through object-oriented technology are particularly important. Many businesses have software development projects that have never been started because the cost of development, in time and dollars, was too high. But with object-oriented environments such as NeXTstep, development time can be reduced from 18 months to three months. This is a tremendous productivity gain for business.

What Is NeXTstep?

NeXTstep is the first mainstream object-oriented application software development and user-interface environment. NeXTstep has two faces: For developers, it simplifies the creation of complex, multitasking applications that have graphical user interfaces. To users, it is a graphical user interface that makes the complex, powerful, multitasking UNIX operating system accessible.

NeXTstep provides software developers with a set of interactive objects — e.g. buttons, scroll bars, windows — which they can arrange to create any kind of application, complete with a graphical user interface.

This object-oriented process removes a major obstacle in the development of an application. Freed from the burden of time-consuming and difficult user-interface programming, developers using NeXTstep can concentrate on the features and capabilities specific to their applications.

Why is programming in an object-oriented environment so much faster?

Object code is reusable, which is the key breakthrough in object-oriented programming. As a result, common programming tasks can be written as objects and used over and over again. In NeXTstep, those common tasks are included in a library of more than 100 objects for handling voice data, text data, window drawing and color selection, to name a few. By providing these functions as pre-written objects, programmers can spend their time adding new functionality to a program, rather than writing code to create a scroll bar, a menu or a printer driver.

What are the overall benefits of an object-oriented architecture and development environment?

Object-oriented programming, unlike other “buzz word” technologies such as artificial intelligence, offers immediate benefits to both end users and programmers. These benefits include:

Fast Application Development

Programmers using object-oriented development tools create software five to ten times faster than they do with traditional procedural-oriented programming languages.

Improved Reliability

By reusing objects, rather than recreating lines and lines of code for a particular feature or function, new programs are more likely to work right. When an object is perfected, it can be used over and over again in many applications.

Repairs to "broken" software are also made much more quickly because it is easy to locate the broken functionality in an application (since the functionality of the application is divided up into neat boundaries called objects). If one object in a program is having problems, it can be fixed without disrupting other objects in the program.

Increased Flexibility

The ability to easily modify objects creates greater programming flexibility. Changes can be made while a program is being developed without creating huge time delays. In addition, objects can be easily connected, one to another, creating greater interaction and communication between functions within programs, as well as between programs.

Ability to Manage Greater Complexity in Software

Objects are self-contained units that have clearly defined functions and responsibilities within an application. Because of this, an application is clearly delineated into separate objects, and it is possible for the developer to keep track of what's happening inside his application much more efficiently. Since the management of complexity really boils down to keeping track of what is going on everywhere, object-oriented design naturally reduces complexity of even very large systems.

What specific advantages does NeXT's object-oriented architecture offer to programmers?

Programmers make great development gains programming in NeXTstep. One commercial developer created a complex presentation package in only nine months. In addition, NeXTstep's object-oriented system software provides developers with "common code" objects that they don't have to recreate, such as objects for printing, faxing, opening/saving files, sound and image processing, storing data, inter-application communication, and choosing fonts and choosing colors, leaving them more time to concentrate on creating innovative and distinctive features.

Corporate programmers are also able to create NeXTstep applications five to ten times faster than is possible with other development environments. And, because NeXTstep provides developers with a ready-to-use library of interface objects, such as buttons, windows and menus, in-house programmers can create applications that have a look and feel consistent with commercial NeXT step applications, and that are as easy to use.

How does NeXT's object-oriented architecture benefit end users?

NeXT's object-oriented system software provides a number of benefits to end users. Custom applications written in NeXTstep look and feel like commercial NeXTstep applications. Because objects communicate easily with one another, NeXTstep provides users with features such as the "services menu," an access point to services including optical character recognition or a digital dictionary from within a word processor, for example.

Also, once the user has mastered one application, other applications are easy to learn. And because all applications are built from the same basic set of objects, the user is guaranteed a consistent, high-quality user interface.

What is an object?

Software objects are encapsulated chunks of code that contain both a program and data representing a specific "real world" object. For example, a page in a NeXTstep window includes text and graphics — i.e., the data — as well as the code that controls the behavior of the scroll bar or close box — i.e., the program. Programmers write groups of objects, which send messages between each other, to create an application.

What is object-oriented system software, and how does it differ from object-oriented programming?

Object-oriented system software brings all the benefits of object-oriented programming to system software. NeXT is the only computer available today that is built with object-oriented system software.

Object-oriented system software also allows NeXT to continually enhance NeXTstep objects in a way that is transparent to the developer. For example, in NeXTstep 2.0, NeXT added a fax button to the print panel, making it as easy to fax a document through a faxmodem as it is to print it. Once a fax object was added to the various objects that make up the print function in NeXTstep, any application that could print also gained the ability to fax without any effort at all on the application developer's part. Thus, users gained this benefit throughout the system simply by upgrading to 2.0 and without buying and installing a new program.

Object-oriented programming is used to create object-oriented system software. There are a number of object-oriented programming languages, including Objective C, C++, Object Pascal and SmallTalk, that can be used to create object-oriented applications. The NeXT platform offers the combination of NeXTstep and Objective C — and NeXTstep also supports C++ — which provides developers with a pre-written library of objects for common programming tasks in a graphical user interface.

By writing NeXTstep with Objective C, NeXT has created system software based on objects. NeXTstep has all the flexibility inherent in any other application written in an object-oriented programming language.

What other platforms offer object-oriented system software, besides NeXT?

SmallTalk is the only other object-oriented system software environment available today, but it does not have anywhere near the number, depth and breadth of objects contained in the NeXTstep environment, nor does it contain an extensible object editor like Interface Builder.

There are other vendors who claim to have object-oriented system software, but none of them have even written their system software in an object-oriented language, let alone built truly object-oriented system software.

What about the plans that Apple and IBM recently announced to create an object-oriented operating environment?

Apple and IBM announced that they would be developing a new operating environment based on object-oriented technology, but it is not expected to be available until the mid-1990s. Based on five years of experience developing NeXTstep, NeXT believes it will take them that long, and possibly even longer.

Why does NeXT claim that it will take so long for other companies to develop object-oriented systems to compete with NeXTstep?

It takes a long time to develop software and operating systems. NeXTstep took five years to create. Because NeXT's competitors, like IBM and Apple, have only just started to develop their object-oriented operating systems, they have another four or five years to go before they will be able to release a full-fledged object-oriented operating system.

Can you give me some examples of NeXT customers who are using object-oriented programming to their advantage?

Many NeXT customers are realizing real productivity gains through NeXTstep, primarily through rapid custom application development. Below are some quotes from customers who are using NeXTstep to develop mission-critical custom applications:

“The strongest selling point for us is the NeXTstep development environment. It allows us to develop programs in roughly one-third the time (this is a blended figure over many applications). On a NeXT you are encouraged, coddled and brought quickly up what would otherwise be a difficult learning curve in making good use of object-oriented programming techniques. This is possible because object orientation isn't just a veneer on the programming environment. The heart and soul of the NeXT is object-oriented. This allowed our developers to become object-oriented programming experts with relatively little pain. The benefits are monumental. It is inconceivable to me that we would ever go back to the days of functional programming.”

Hadar Pedhazur,
Vice President, Equities Technology manager
Equity Derivative Products

“NeXTstep is a phenomenally productive environment for proprietary applications development. We were able to develop a prototype trader’s workstation in less than three months. In other environments it would've taken two, three or even four times as long.”

Robert C. Wilen
Vice President, NeXT Office Automation Project
O’Connor Partnerships