OpenStep and Solaris

A White Paper

SunSoft
A Sun Microsystems, Inc. Business
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063
USA

Sunsoft
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

NeXT Computer, Inc.
900 Chesapeake Drive, Redwood City, California 94063 U.S.A.

Table of Contents

OpenStep and Solaris

Introduction

The SunSoft / NeXT collaboration combines the distributed computing strengths of Solaris, including the system-wide object services provided by Project DOE, with NeXT's OpenStep, the industry leader in object-oriented applications development and end-user computing. This white paper describes OpenStep's features and benefits and SunSoft's plans for incorporating OpenStep into Solaris.

The OpenStep Initiative

The primary goal of the OpenStep initiative is the creation of an open, high-volume portable standard for object-oriented computing. The benefit of this intended standard is to enable the rapid development of business applications, including distributed applications which fully exploit the power of client/server computing through the use of object technology.

Providing a rich object solution for client/server computing requires an open standard. This standard should include:

- A robust, scalable and reliable operating system foundation based on open systems standards;
- A common services architecture for distributed computing based on objects;
- Integrated and complete object-oriented development tools allowing the rapid creation of client/server applications;
- An object-oriented framework accessible to all applications and developers; and
- An end-user environment built to exploit the ease of use and integration provided by an object-oriented foundation.

The NeXT and SunSoft Partnership

NeXT and SunSoft have formed a partnership to make NeXT's OpenStep Application Programming Interface (API) the industry standard for object-oriented computing. By integrating NeXT's OpenStep API, SunSoft's Project DOE (Distributed Objects Environment) object system, and Solaris, SunSoft will provide customers with a complete solution that is compliant with the industry's first open systems standard for object-oriented computing.

The Goal: A Standard For Distributed Object Computing

The rapid development of distributed applications has been an elusive goal of corporate computing. Earlier attempts required the use of low-level networking protocols (such as Remote Procedure Call mechanisms), leaving the developer with the burden of focusing more on complex, low-level networking and systems management than on high-level business functionality. The network programming model also meant that implementation decisions had to be made at the time the application was being designed, rendering the application hard to modify and less able to benefit from the power of client/server computing.

By allowing programmers to work at a high level of abstraction and to reuse code with ease, object-oriented programming is being viewed as the software equivalent of the industrial revolution, replacing today's costly *hand-made* business applications with applications which can be *assembled* out of reusable components.

The NeXT and SunSoft partnership will accelerate the adoption of object technology in two ways. First, NeXT and SunSoft are extending the industry's current focus on *object-oriented languages* to a broader focus on developing a standard, *system-wide object architecture*. This architecture will provide an object framework which integrates languages into a complete client/server operating system optimized for object-oriented computing. Much richer than a narrow object-oriented language layer, this object architecture will harness the power of networked objects to build distributed, client/server applications easily.

Second, no standard for object-oriented computing exists today. NeXT and SunSoft are working together to make the OpenStep APIs the first high-volume, open and portable industry standard. This is the overall goal of the OpenStep initiative, to provide the industry's first open systems standard for object computing.

What Is OpenStep?

OpenStep is an Application Programming Interface (API), an interface to a set of functionality. One implementation of this functionality is the NEXTSTEP Application Environment. In terms of today's NEXTSTEP product, the OpenStep API corresponds to the operating system independent layers of NEXTSTEP. (See Figure 1.) The OpenStep interface is based on the NEXTSTEP 3.2 APIs. Thus, successful implementations of the OpenStep API will provide the functionality and user experience similar to the current version of NEXTSTEP.

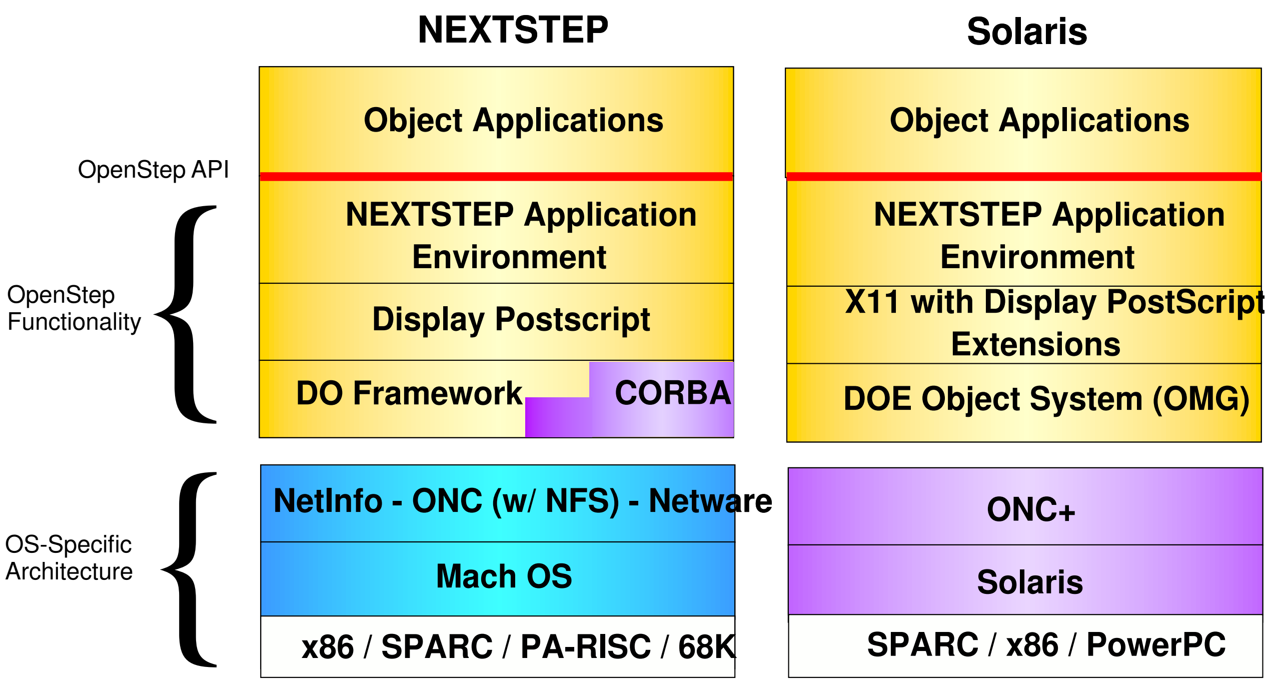

Figure 1 The OpenStep API as implemented in NEXTSTEP and in Solaris

SunSoft is adopting the OpenStep API as its sole object standard. By integrating Solaris and OpenStep, SunSoft is ensuring that the functionality found today in the NEXTSTEP Application Environment is part of Solaris.

The formal OpenStep interface specification will be published by June 30, 1994 and will be available to all interested parties, enabling others to develop and market alternate implementations of the OpenStep standard. The OpenStep interfaces will be clearly documented and evangelized to promote the consistency and broad availability of the OpenStep standard on multiple platforms.

As an open, public standard, the OpenStep API will be administered by an independent standards body. Any organization whose implementation of the OpenStep API passes the OpenStep certification test suite may freely license the OpenStep name and trademark.

Towards an Open Systems Market for OpenStep Objects

The integration of the OpenStep API into industry-standard operating systems enables vendors to bridge the gap between operating system and application functionality. By adhering to standards such as the Object Management Group's (OMG) CORBA (Common Object Request Broker Architecture) and extending the benefits of object orientation to distributed computing, the commercial customer will realize many benefits. Chief among them will be the growth of a broad market for objects, a market not dominated by a single supplier.

The ability to obtain a wide variety of commercially available software objects and incorporate them into a development environment dramatically improves the creation of robust enterprise-ready applications. This is one key promise of the OpenStep initiative. Achieving this goal will require not only the best technology in the industry, but also the volume adoption that can be created through open systems.

NeXT and SunSoft are committed to creating an open environment where market forces greater than any single company can accelerate the availability of technology and solutions. Support for creating OpenStep-compatible objects and assistance in obtaining them will be provided by NeXT and SunSoft.

OpenStep Features

Objects: The Basis of OpenStep

Objects are the foundation of the object-oriented OpenStep API. Objects are self-contained, reusable software modules containing data and associated behavior (methods). Objects help developers manage complexity because developers only need to understand what an object does (the messages it responds to)–not how the object "works" internally–to use an object developed by another party.

Object-orientation simplifies application development, makes team programming easier and maintenance simpler, and gives programmers a more comprehensive view of an application's architecture. Most importantly, object orientation encourages code reuse, greatly reducing the costs associated with the software maintenance lifecycle.

By integrating a powerful object framework into standard operating system environments, the commercial customer will realize unique benefits. These include greater ease of use for all applications, reusable third-party objects, improved developer productivity, and greater insurance that applications are portable, scalable and extensible. Business applications will evolve rapidly and gracefully reducing development costs.

The Goals of OpenStep

The development of OpenStep was guided by these principles:

- The industry needs an open object standard for enterprise computing, bringing to object-orientation the broad market forces of the open systems movement;
- OpenStep applications should be portable across the full range of OpenStep and NEXTSTEP platforms;
- Developers in the 1990's need an application development *architecture*, not a collection of poorly integrated tools that weren't designed to work together. The OpenStep API defines a complete architecture ensuring that OpenStep-compliant programming languages, windowing and graphics systems, user interface toolkits, class libraries, database tools, and project management tools will work together in a tightly-coupled, integrated way;
- A development platform must provide a high level of functionality that all developers can depend on–functionality common across applications. This frees developers from writing code common to most applications and requires them to

write only the code which is unique to their business application. The OpenStep API is implemented using rich object frameworks providing the core functionality required by most business applications.

- Interface creation should be the least labor-intensive part of development. Experimentation with the human factors required to make an application easy to use should be enabled by good GUI development tools. To make this possible the OpenStep API is supported by a companion product providing OpenStep application development tools whose Interface Builder makes interface creation a breeze.
- A development environment should employ the same windowing system and imaging model for screen and hardcopy output, and all applications should support key multifont text and graphics standards.
- A development environment should encourage programmers to provide high-quality and consistent graphical user interfaces that can be easily localized for different foreign languages. By its separation of user interface files from business logic, the OpenStep architecture fully supports easy application localization.
- Custom applications should integrate commercial productivity tools. OpenStep accomplishes this by using a single, uniform object messaging interface between objects within and among all applications, easily accessed by end-users through consistent cross-application services and information hot-linking.
- At a minimum, the object API should support multiple languages. The OpenStep API will support applications written in ANSI-C, Objective-C and C++.
- The environment should facilitate the development of distributed applications to promote client/server architectures and distributed computing. The burden of networked applications development should be handled transparently through use of distributed objects conforming to a common model.
- Finally, the system should define a model for creating objects that will enable independent authorship of a large number of reusable software objects by third party developers–in much the same way that today's NEXTSTEP developers are building reusable ObjectWare.

These are the goals of the OpenStep API, together with SunSoft's goals in providing NEXTSTEP Application Environment functionality for Solaris users, and the goals of companion products for Solaris users, including application development tools for OpenStep programmers.

OpenStep Functionality

The OpenStep API defines a single uniform, portable object-oriented environment. The cohesiveness of the system simplifies application development, makes cooperation among programmers and applications easier, renders maintenance simpler, and gives the developer a more comprehensive view of a program's architecture. A complete environment for object-oriented end-user computing, the OpenStep API is implemented using a number of object frameworks, including the NEXTSTEP Application Framework, frameworks providing access to relational databases and frameworks providing support for distributed computing using objects. The OpenStep API is supported by companion products, including application development tools for OpenStep. Each of these is described below.

NEXTSTEP Application Framework

The NEXTSTEP Application Framework is a set of approximately 150 core objects which have been developed and refined over eight years and which have been shipping for more than five years. These objects provide the framework required by any application, and a powerful set of interface and supporting objects that provide advanced functionality well beyond the windowing toolkits found in other environments. In addition to standard interface elements such as scrollers, cursors, buttons, sliders, windows, and panels, the Application Framework includes high-level objects for such functions as:

- Managing events
- Cross-application services
- Information hot-linking across applications
- Imaging
- Displaying TIFF and EPS images
- Manipulating, editing and spell-checking multifont text
- Selecting colors
- Opening and saving files
- Creating hierarchical information browsers
- Device-independent printing and faxing (Group III and IV formats)

Nearly every object in the Application Framework is extensible. Developers can create subclasses of OpenStep's object classes, and even those not designed to allow subclassing often provide hooks for dynamically adding new components at run-time.

Frameworks for Accessing Relational Databases

Because of integrated support for relational databases, every OpenStep application can access and incorporate relational databases. Unlike traditional fourth-generation languages, using OpenStep there is no distinction between database applications and other OpenStep applications. All applications use the same development tools and access the same underlying facilities. Every OpenStep application can be a database application and every database application can exploit the other rich facilities of OpenStep such as cross-application services.

These frameworks radically shorten the time required to design and implement database applications that have graphical user interfaces (GUIs) and simplify management and control of database transactions within an application. These frameworks allow developers to design applications independently of the databases they will be used with. The resulting applications can be migrated to a different database without modification, that is, without recoding the user interface or applications logic. These frameworks enable a single application to integrate information from multiple database vendors' products.

Distributed Object Framework

The OpenStep API is designed to support distributed object applications. The OpenStep object model is the same within applications, between applications residing on the same computer, and between processes running across enterprise networks. OpenStep's reliance on a distributed object model simplifies the development of client/server applications by enabling OpenStep objects to reside anywhere on a network of clients or servers. NeXT and SunSoft both endorse the OMG CORBA standard for distributed object computing and will use CORBA to ensure interoperability across different OpenStep implementations. OpenStep's object distribution capabilities ensures that objects can interoperate across all OpenStep clients and servers.

Available today for distributed computing is NeXT's Distributed Object Framework, a component of the OpenStep API. Distributed applications based on this distributed object framework, including applications based on NeXT's PDO (Portable Distributed Objects) product for servers, will easily migrate to OpenStep implementations.

Application Development Tools for OpenStep

An integrated set of application development tools will be available as a companion product for building OpenStep-compliant applications. These tools will include language support for applications written in Objective-C, ANSI-C and C++. In addition, they will include visual programming tools such as Project Builder, for managing all of the files and resources in a given application; and Interface Builder, for managing all of the objects in an application.

Interface Builder is the primary tool for graphical interface development and management of an application's objects. Interface Builder improves programmer productivity with its complete environment for laying out, constructing, and testing user interfaces. It also provides powerful software engineering tools to incorporate, access, edit, and manage live objects, including off-the-shelf as well as customer developed objects. Interface Builder is:

- A complete object-oriented user interface development, layout, prototyping, and testing tool;
- An object editor that manages the interactions between the objects in a program. Using Interface Builder, programmers define the messages objects may send each other, whether they are user interface objects or any other kinds of objects;
- A tool that encourages modularity because of the ease with which a program can be split into separate modules;
- A class hierarchy manager for supplied classes and classes of objects added by developers, and a tool for creating subclasses of existing classes; and
- An object and palette manager, allowing palettes of custom objects to be displayed and manipulated in the same way as OpenStep supplied objects.

OpenStep Benefits

Leverage Reusable Third-Party Objects

OpenStep's standard object interfaces combined with multi-vendor support for the base object framework will ensure that a wide variety of industry-specific and general purpose objects are developed, of the sort described in NeXT's *ObjectWare Catalog*. Developers will be able to rely on in-house objects as well as commercial objects to provide much of the functionality required, concentrating their efforts on deploying applications constructed of reusable, tested objects, resulting in higher overall software quality and faster time to completion.

Greater Functionality For Every Application

OpenStep's object frameworks provide the common functionality needed by most business applications–as much as 85% of common business functionality according to a study by Booz • Allen & Hamilton of the related NEXTSTEP product. OpenStep will raise the lowest common denominator, providing all programs with a greater level of consistency in behavior and basic functionality such as the ability to message other applications or fax information from within any application. OpenStep's rich application framework allows developers to focus the majority of their development energy on the unique, high value aspects of the application, leveraging the core functionality already provided by OpenStep.

Portability

Applications written to OpenStep will be portable across the entire range of platforms supporting the OpenStep API, resulting in an open competitive space for platform vendors. Platforms supporting OpenStep applications will include native NEXTSTEP for Intel, SPARC and PA-RISC; and OpenStep-compliant Solaris running on SPARC, Intel and PowerPC processors; and other OpenStep-compliant products from other vendors.

A Volume Standard

A key benefit of the NeXT and SunSoft collaboration will be to make OpenStep a standard, evangelizing the industry-proven OpenStep APIs to as large an installed base as possible through volume shipment, attractive licensing, and open publication of the specifications by independent standards bodies. This will create a larger market for third party software solutions of all kinds.

Extensibility and Graceful Evolution

Because the public interfaces of objects are separate from their implementation, existing applications can extend their functionality by replacing older objects with newer versions, greatly reducing the maintenance and testing requirements which prove to be so expensive in non-object-based systems. In this way, applications can be extended easily and evolve to meet new business needs.

Scalability and Transportability

Given the OpenStep distributed object framework, objects can reside anywhere on the network, and a network object model ensures that objects hide the details of client/server and network-based computing.

Underlying OpenStep will be SunSoft's OMG-standard DOE architecture which supports the full range of granularity, from fine to coarse grained objects. This scalability, combined with the distributed object capability to build networked applications easily and transparently supports the widest set of distributed architectures. Complex network programming is no longer required, due to the powerful location transparent mechanisms which manage global naming, location, and communication among objects.

Reduced Development Costs

Assembling new applications out of an inventory of in-house or third-party components greatly reduces the cost of new development. Moreover, once an object is developed and tested, it will not need to be tested again, since objects are fully independent modules which are not affected by changes to other objects. NEXTSTEP, the basis of OpenStep, is recognized for its ability to reduce development costs through rapid prototyping and object reuse, and enabling developers to work with real, live dynamic objects.

OpenStep and Solaris

SunSoft and NeXT are bringing the best of object-oriented and distributed computing technology to Solaris. SunSoft's technology leadership has led to the adoption of core DOE technology in the Object Management Group's standards including CORBA (Common Object Request Broker Architecture) and COSS (Common Object Services Specification), of which SunSoft was the primary author.

OpenStep complements these system-wide services. NeXT has led the industry in developing object-oriented environments and developer tools. Users of NEXTSTEP have achieved dramatic productivity gains in developing high quality applications, shortening from years to months the time required to create sophisticated and easily maintained business solutions. The incorporation of NeXT's object technology into Solaris will result in the best distributed object system in the industry.

OpenStep and CDE

When SunSoft's OpenStep integration is complete, Solaris will include two application environments, CDE and the NEXTSTEP Applications Environment implementation of the OpenStep API. The CDE environment will ensure compatibility across multiple UNIX platforms. SunSoft is committed to support CDE as one of its primary standards.

For users and developers desiring to gain the benefits of object-orientation such as rapid time to market for mission-critical software, Solaris will support the full NEXTSTEP user environment. To encourage standardization of object applications, SunSoft has chosen the NEXTSTEP Applications Environment, accessible via the OpenStep API as its object standard. (See *Figure 2*.)

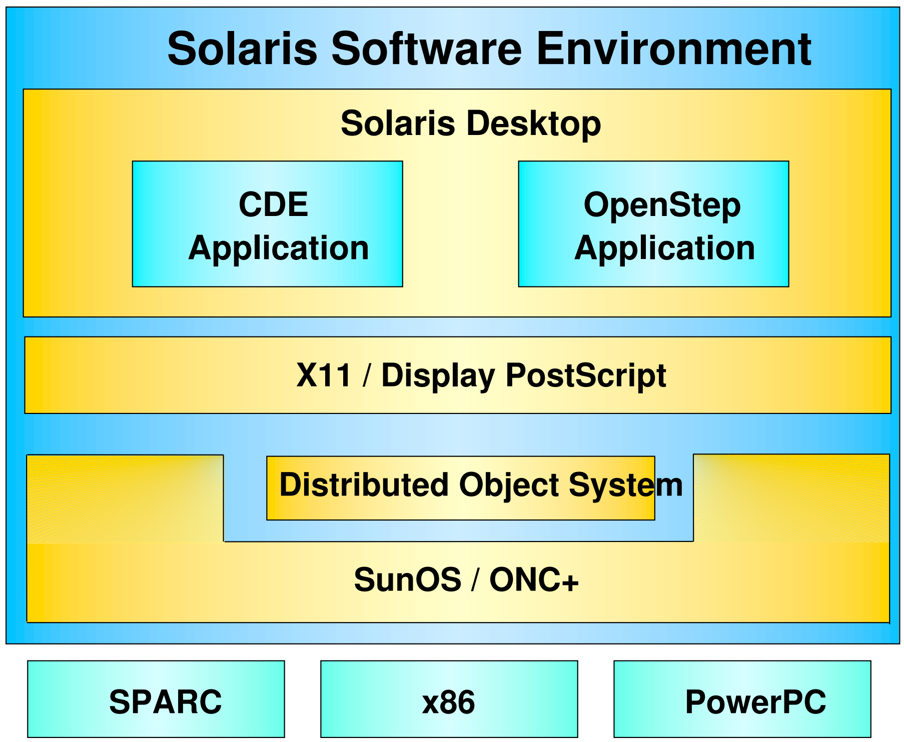

Figure 2 The Architecture of Solaris / DOE Integrated with OpenStep

SunSoft will ensure that CDE and OpenStep applications integrate and interoperate on a common desktop.

Object-oriented applications will be developed by writing to the OpenStep API. This API will provide consistent and high-level access to the features of the NEXTSTEP Applications Environment, the NEXTSTEP user interface and object environment, including NeXT’s core object frameworks.

OpenStep and Project DOE

To further enable client/server applications for enterprise computing, the OpenStep API will be supported by a powerful distribution infrastructure, DOE, that allows objects to collaborate across the network. The DOE Object System includes a full range of distributed computing services based on the OMG CORBA specification for heterogeneous computing. The distributed object system will provide:

- Transparent distribution, via OMG CORBA technology, allowing all objects to message all other objects, regardless of location. This flexible capability supports client/server and peer architectures, enabling software right-sizing;
- Multilingual interoperability, via OMG standard interface definition language (IDL) mappings, allowing a client object written in one language to access a server object written in another;
- Scalability, provided via a distributed object architecture, allowing systems to be constructed of any size, utilizing any number of machines, all connected with via standard mechanisms supported by over 350 companies through the OMG standardization process; and
- Standardization, via adherence to industry standards, as well as existing agreements with other major computer vendors such as IBM and HP. Standardization provides the ability for system level objects to be ported across multiple platforms, as well as establishing a common object model for third party developers.

SunSoft will support the OpenStep API as an integral part of Project DOE's distributed object environment, leveraging Solaris multithreading and multiprocessing capabilities.

Using OpenStep Within Solaris

Because the OpenStep interface specification is based on NEXTSTEP 3.2, the OpenStep API is designed to ensure maximum portability, compatibility and consistency with native NEXTSTEP. And OpenStep will serve as the API for future releases of native NEXTSTEP itself.

Users will find using an application written to the OpenStep API identical to the experience of using a native NEXTSTEP application. In a similar fashion, programmers will use the OpenStep developer tools based on the existing NEXTSTEP Developer family of tools to build OpenStep applications. Developers will notice only minor changes from today's NEXTSTEP 3.2 API.

The experience of using or developing for OpenStep-compliant environments will be very similar to the experience of using or developing native NEXTSTEP applications. Thus, the best way for developers to begin developing OpenStep-compliant applications is to begin using the OpenStep Developer Starter Kit.

Summary

Through the OpenStep initiative, NeXT and SunSoft have defined an open systems standard for objects. OpenStep integrated into Solaris provides an enterprise-ready framework for distributed object computing. The solution offers software that has both a low learning curve and high software development productivity, combining the best of distributed and object-oriented computing.

OpenStep's integration into Solaris addresses many of the key problems in developing and using application software today–reducing the cost of development and ownership, improving software quality, ensuring that all functionality is easy-to-use, improving interoperability, and providing a truly scalable, network-transparent architecture for enterprise computing.

SunSoft, Inc.
2550 Garcia Avenue
Mountain View, CA 94043

NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063
USA

Printed in USA