NeXT on Campus™

Winter 1991

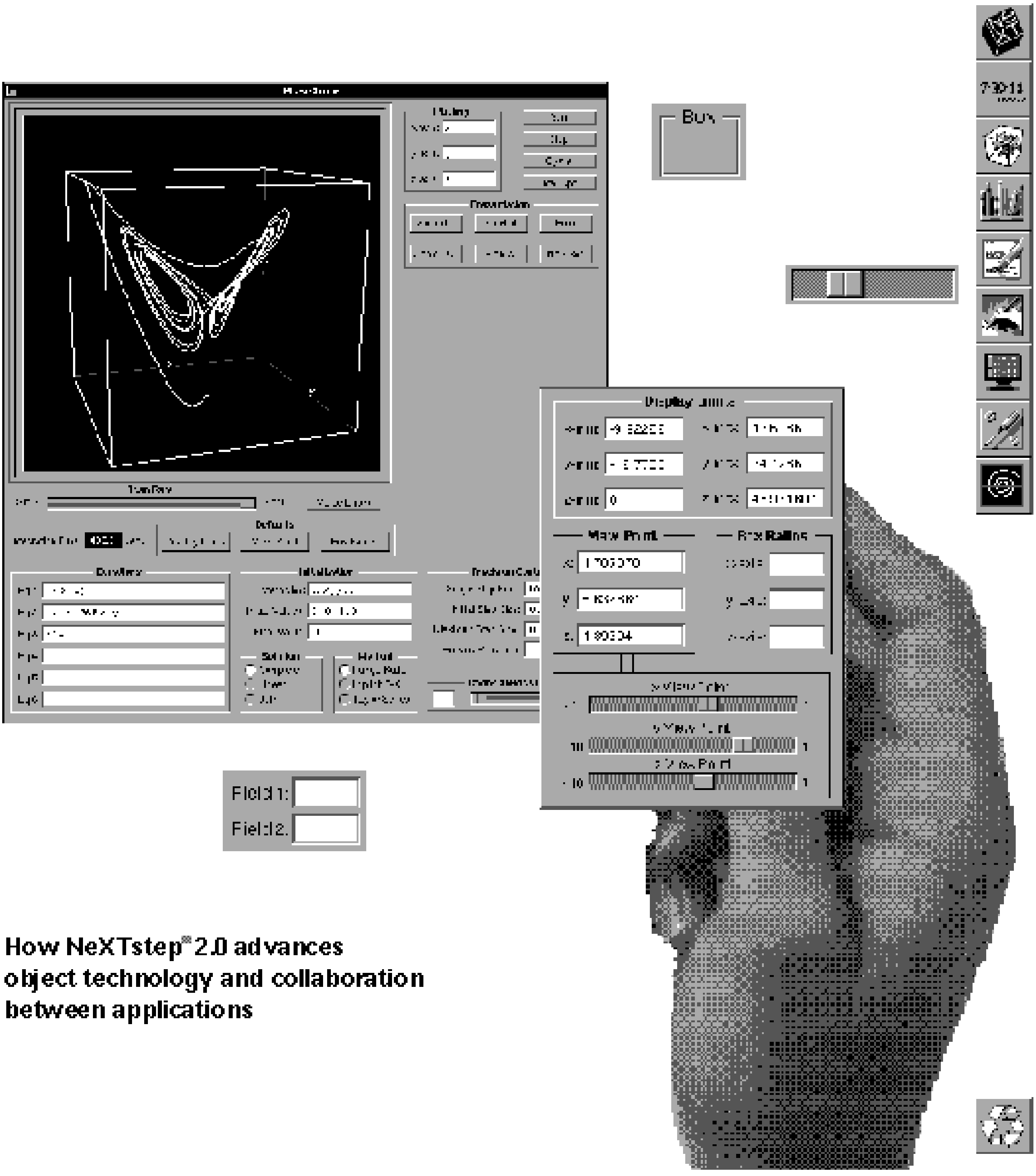
In this issue:

PhaseScope® energizes the study of mathematics

Increasing medical understanding through signal processing

NeXTanswers: hundreds of answers to technical questions

NeXTstep: the platform of choice for *Mathematica*® studies



How NeXTstep® 2.0 advances object technology and collaboration between applications

NeXT on Campus™ is published three times a year by NeXT Computer, Inc. for the higher education community. Each issue of this journal highlights advances in instruction and research in which NeXT™ technology has played a key role, and also presents NeXT-related news, feature stories,
products, and resources available to NeXT users. *NeXT on Campus* welcomes your comments, suggestions, and contributions. You may send electronic mail to next_on_campus@next.com or write to *NeXT on Campus*, NeXT Computer, Inc., 900 Chesapeake Drive, Redwood City, CA 94063. To receive a subscription to *NeXT on Campus,* send your request to the electronic mail address or U.S. mail address above.

COLOPHON
The winter 1991 issue of *NeXT on Campus* was produced using NeXT computers. Text was written with WriteNow® 2.0. Page composition was done with FrameMaker® 2.0. Proofs were printed using a NeXT 400 dpi Laser Printer. All text and image files were transferred directly from a NeXT optical disk to film using NeXT computers and an electronic output device.

ABOUT THE COVER
PhaseScope™ by Michael J. Mezzino, Jr., chairman of the mathematics department at the University of Houston - Clear Lake. PhaseScope is a comprehensive user interface and general graphical display program for qualitatively analyzing the solution and the stability characteristics of systems of first order ordinary differential equations. PhaseScope takes advantage of the NeXTstep® multitasking environment by using *Mathematica*® to solve differential equations.
See “PhaseScope energizes the study of mathematics,” pages 10-11.

CREDITS
Managing Editor: David Spitzler
Designer: Joan Howard
Technical Editor: Jeff Wishnie

Advisors: Jay Capela, Jean Craig-Teerlink, Conrad Geiger, Patty Kammerer, Mike Matlack, Mark Meinrath, Steve Sieczko, Barry Silverman, Julius Smith, Karen Steele, Ronald Weissman

Writers & Editors: Beth Conover, Kevin Cracraft, Richard Crandall, David Grady, Joe Hutsko, Amanda Isles, Anne Kaplan-Neher, Eric Larson, Susan Mandell, Susan Mobley, Rhonda Raider, Ronald Weissman, Kristofer Younger

Special thanks to the faculty, researchers, and students whose work appears in this issue of *NeXT on Campus.*

Winter 1991, Volume 2, Issue 2

NeXT on Campus

Table of Contents

Winter Feature

Academic Projects

Campus Profiles

NeXT Technology

Resources

Object orientation: from applications to environments

by Ronald F.E. Weissman, Director of Higher Education
Kristofer A. Younger, Senior Technical Consultant
NeXT Computer, Inc.

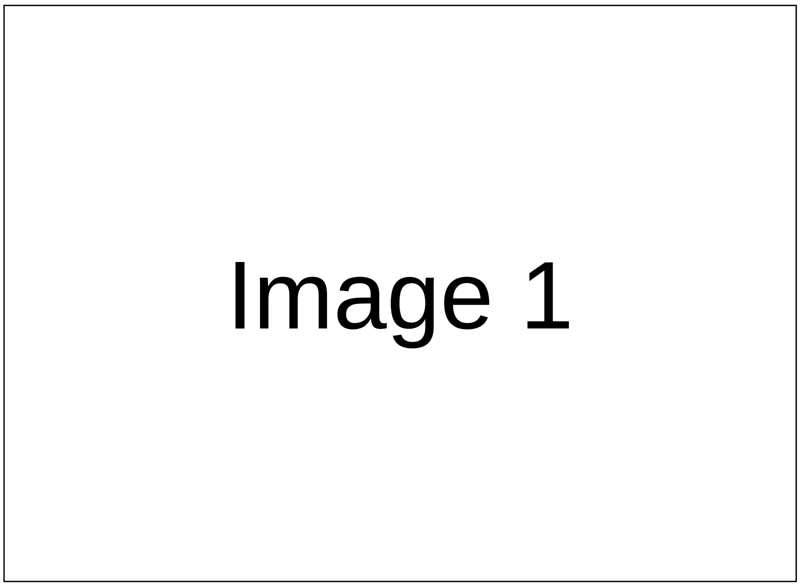
Despite sudden industry attention, the object-oriented software revolution is proceeding slowly. Though the underlying model dates back to research at Xerox Parc in the 1970s, it is only recently that object-orientation has become a mainstream paradigm. As we enter the 1990s we are seeing large-scale adoption of the 'object-oriented' model, even in the more retrograde sectors of the computer industry.

In the most conceptually advanced segments of the industry, whole software systems, such as the entire NeXT development environment, depend on object-orientation as a primary software engineering tool. The ability of object-oriented languages to allow modular components and create reusable code has made them the 1990s solution to productivity and software maintain-ability. The current focus of object-oriented technology has been on improving the productivity of programmers, a focus which is too limited. We believe the true benefits of object-orientation will accrue to end-users of software applications as much as to professional programmers.

What is object-orientation from a user's perspective?

At its most elemental level, object-oriented systems revolve around two key concepts: objects and messages. All software in these systems is constructed of objects that can send and receive messages. An object-oriented user-interface system, such as NeXTstep®, is designed around its objects—buttons, cursors, windows, scroll bars, and menus. Objects communicate with other objects by sending messages. This is the only way objects actually do anything. When a user drags a scroll bar, for example, lines of text change position on screen when the scroll bar object sends a message to the text view object to move the lines appropriately.

To use an object, one need only know the messages that the object understands (the object's protocol or interface), not how it works internally. To use an object-oriented spelling checker, for example, one need only know that the spelling checker is invoked by sending it a "checkSpelling:" message. One need not know how, exactly, the code works or what kinds of search, sort, and comparison algorithms give the spelling checker its power. Indeed, as long as the messages sent to the spelling checker remain the same, one could replace today's spelling checker with a more powerful version in a manner absolutely transparent to users.



The text object provided in NeXTstep 2.0 not only offers spell checking; it also provides font and ruler management. Any user who includes a text object in an application automatically benefits from these capabilities.

We are accustomed to view a well-designed software application as being built from a set of objects. A spreadsheet, for instance, is composed of cell objects and matrix objects. But what if we viewed commercial software applications themselves as objects? What if tomorrow's end-user worked in a world in which spreadsheets, databases, word processors, statistical tools, presentation packages, graphics packages, and communications tools were not standalone 'solutions' but software objects?

Then, tomorrow's software environment would be composed of applications that could send and receive messages from other applications. This—the real promise of object-orientation—would give serious computer users the ability to create rich, customized environments, and not simply empower a few thousand professional programmers.

What will object-oriented applications be able to do?

First, they will be able to send, receive, and share data with other applications. An object-oriented bibliography built in this way would, for example, be able to provide citation data to research databases as well as provide footnote data to word processed papers and articles. Second, object-oriented applications will be able to provide services to other applications. An object-oriented spelling checker might offer its spell-checking services to any application using text, whether it was a spreadsheet cell, a database record, or a word processed document.

Release 2.0 of the NeXT system software supports this notion of a 'service' that one object-oriented application can provide to another application. Through the Services menu, one application can message another and ask it to perform a service.

Envision, then, a new type of interactive computer environment: During the 1990s object technology will empower users through Object-Oriented Applications. Applications will become richer given their new-found ability to communicate with other applications.

Today, the most advanced environments are composed of interacting object-applications based on the best of today's personal productivity applications. This environment is enriched by networked data sharing, support for messages, and support for an 'interpersonal' style of computing—allowing groups of people to share data and tools collaboratively.

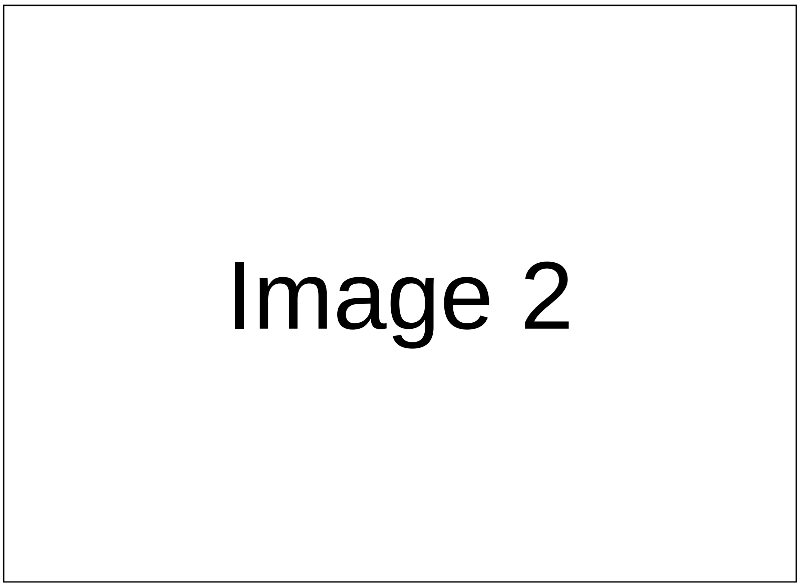
THE NeXT OBJECT ARCHITECTURE

How does NeXT's Release 2.0 Software support object-orientation today?

Much has been written about NeXT's object-oriented programming tools, its Application Kit,™ and Interface Builder.™ What few realize, however, is how thorough-going this object orientation is.

NeXT technology provides much more than an object-oriented language compiler. For the professional programmer, NeXT provides the common objects that most applications need. This allows NeXT to provide developers a full set of user interface objects including windows, scrollers, simple button-oriented cause-effect triggers, and a text object that would stand out as a great word processor on most other UNIX® systems. From the Mach operating system kernel, through its user-level services, the NeXT software environment is fully object-oriented, enabling programmers today and end-users tomorrow to reap the benefits of a true object architecture.

From the Mach operating system kernel, through its user-level services, the NeXT software environment is fully object-oriented, enabling programmers today and end-users tomorrow to reap the benefits of a true object architecture.



A custom application in progress, created with Interface Builder by implementing the custom objects TimeLine, BarGraph, and WorldMap. Objects were dragged off the palettes menu and placed in the Histogram window.

The importance of Mach

At its most fundamental level, the NeXT environment is based on the architecture of an object and message system. NeXT chose Carnegie Mellon University's Mach as its operating system precisely because Mach is built around an efficient and effective object messaging and interprocess communication architecture.

Mach's messaging facility is implemented by two kernel abstractions: ports and messages. A port is a protected communication channel to which messages may be sent and queued. The port is also the basic object reference mechanism in Mach; its use is similar to that of object references in an object-oriented system. That is, operations on objects are requested by sending messages to and from the ports that represent them.

Message passing is the primary means of communication both among tasks and between tasks and the operating system kernel itself. System services for example, are invoked by one task sending a message to another task that provides the desired service.

How does NeXT support object-oriented applications?

NeXTstep 2.0 provides three fundamental facilities for creating and manipulating object-oriented applications: the Speaker/Listener protocol, Interface Builder palettes, and Workspace Manager™ Services.

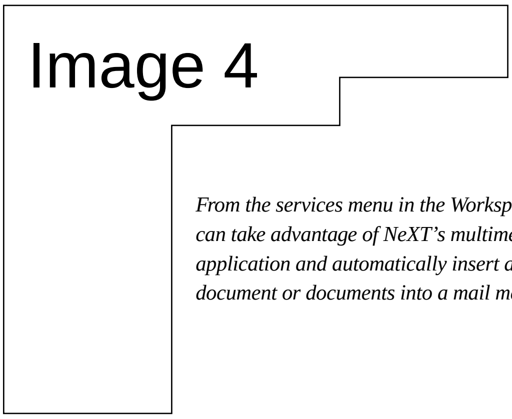
Speaker/Listener. On top of Mach's interprocess communications facility, NeXT has built an interapplication communications facility. Each application on NeXT can have two communications objects: a Speaker object, which is used to send messages to other applications and the Workspace Manager, and a Listener object, which receives messages from other applications. Both of these objects are built on top of Mach's ports. For example, if a user needs to define a word while using the WriteNow word processor, the Speaker object in WriteNow sends the Listener object in Digital Webster™ a "defineWord:" message. All message passing between applications is implemented through basic Mach messaging.

Interface Builder and support for object-oriented software. Interface Builder has increased its functionality in Release 2.0 by dynamically supporting loadable groups of reusable objects called palettes. A palette is a collection of objects ranging from such basic items as text fields, buttons, and sliders, to more complex objects which can represent databases or specialized charts that can be dragged off a palette window to drop into an application's windows. In Release 2.0, one can add new palettes (from commercial developers or one's own projects) of objects to Interface Builder's palette window and drag these new objects off into one's applications.

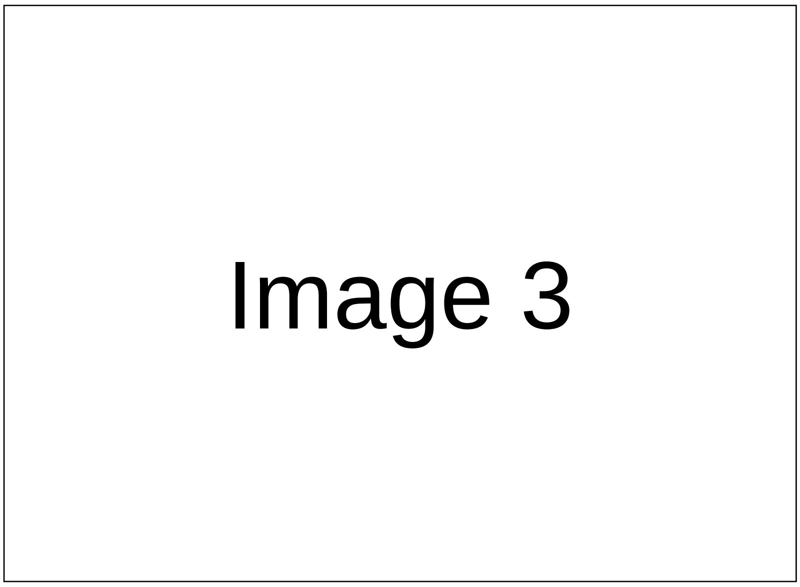
Objects from palettes may be examined with inspectors; these inspectors change the internal properties and attributes or default values from within Interface Builder. In Release 2.0, Interface Builder knows how to generate project files (such as ".palette" files) of different kinds. Once a palette has been created, it can be shared with colleagues.

Release 2.0 offers users the opportunity to generate different kinds of small, highly functional palettes: visualization objects, control objects, music objects, graphic objects, and analysis objects. An organization that has taken advantage of this is Objective Technologies, Inc.
Its product, called [OT Palettes: 2.0],™ is a collection of custom objects integrated into palettes providing added functionality to NeXT's Application Kit. Objective Technologies also provides custom Interface Builder inspectors for modifying the non-graphic features of all of the objects in [OT Palettes: 2.0]. Some of the palettes currently available are Chooser,™ GraphPalette,™ MathPalette,™ and SmartFields.™ (For a description of each palette, see pages 32-33.) Many of tomorrow's 'paletted' objects will be full featured miniapplications. By dragging objects off palettes and linking objects, end-users with some programming experience will be able to create their own custom applications. In tomorrow's object-oriented world, users will be able to link database access objects (this will be available in late 1991 with NeXT's database object kit) to visualization objects, to create custom analysis tools—all without significant programming.

Services. NeXTstep 2.0 introduced the concept of user-level services, which allows one application to take advantage of the tools provided by another application. Services are registered with the NeXT Workspace Manager and become available to any application through the now-standard Services menu.



From the services menu in the Workspace, users can take advantage of NeXT's multimedia mail application and automatically insert any document or documents into a mail message.



A researcher might build an interactive physics palette which would allow students to create their own simulations in Interface Builder. For example, the image shows an ideal gas molecule chamber, several buttons and sliders, and an object to graph the motion

Image 5

After receiving a fax in NeXTmail,™ the user views it with the FaxReader application included with NeXTstep 2.0. To convert the fax (or a portion of it) to editiable text, the user chooses OCR Servant from the Services menu, then chooses Selection.

Release 2.0 offers users the opportunity to generate kinds of small, highly functional palettes: visualization objects, control objects, music objects, graphic objects, and analysis objects. Many of tomorrow's 'paletted' objects will be full featured mini-applications.

For example, every NeXT computer includes, among others, these common services:

- Any application can message Digital Webster to define selected words.
- Any application can message Digital Librarian™ to search for a selected word or words in a user-specified body of reference or user-written text materials.
- Any application can avail itself of full communications services to mail, print, or fax any document or portion of any document.

In addition, a user can acquire commercial applications providing these kinds of services to other applications:

- HSD's OCR Servant accepts ".tiff" files (bitmapped scanned images) and converts them into editable ASCII text; and
- Presentation Builder sends a matrix of numerical data (tab-delimited at one line per row) to the Lotus® Improv™ charting application, PBuilder.

OUR CHALLENGE
From applications to environments

In today's traditional applications environment, commercial and academic software developers typically view their products as standalone tools. Underlying platform technology, such as that provided by NeXT, offers the possibility of moving from a world of discrete applications to a world of cooperating, integrated, and user-customized environments. In higher education, for example, one can envision scholarly environments composed of component software application objects such as:

- word-processor, text-database, hypertext linking, concordance, and dictionary objects that communicate to create a humanist's scholarly environment.
- cartographic, statistical analysis, spreadsheet, and database objects that communicate to create a social scientist's environment.

- spreadsheet, simulation, database, and linear modeling objects that communicate to create an economist's environment.
- CAD, spreadsheet, project-management, instrumentation and data acquisition tools, specialized analysis tools, visualization, and statistical objects that communicate to create an engineering environment.
- MIDI tools, waveform editors, sampling and synthesis tools, and score editing objects that communicate to create a musician's environment.

The underlying technologies to create environments exist today. The task of moving from applications to environments is one of vision, not one of technology.

The underlying technologies to create environments exist today. Through the Services registration process, applications can avail themselves of the functionality provided by other applications. And through dynamically loadable palettes of objects, tools can be combined into 'meta-applications' based on the cooperation of several smaller, functionally specialized tools.

The task of moving from applications to environments is one of vision, not one of technology. Commercial software vendors need to to understand the benefits of object technology as perceived by users; software developers need to embed an application's functionality in communicating objects, and make a rich set of their application's functionality available from within the Services facility.

By creating a new software paradigm and a world in which communicating objects come to replace today's world of standalone applications, we will, finally, have harnessed the vast potential of object-orientation to empower tomorrow's serious computer users.

Academic Projects

University of Florida

Increasing medical understanding through signal processing

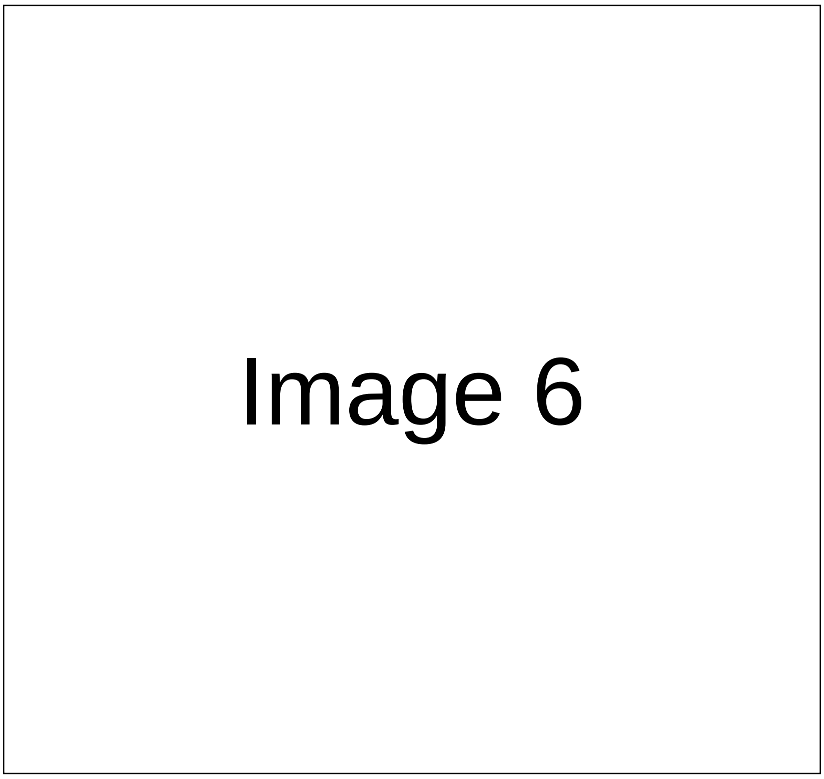
For more than 14 years, Jose Principe, an associate professor of electrical engineering at the University of Florida, has explored biomedi-cal engineering and signal processing. He is particularly interested in research involving the analysis of electroencephalograms (EEGs)—tracings that show changes in electri-cal potential caused by brain activity.

Working with 14 graduate students in a lab of networked NeXT computers, Principe developed several interrelated projects that explore the uses of digital signal processing to medicine. To work toward their goals, Principe and his students rely primarily on *Mathematica*® and the Motorola DSP56001 digital signal processor. Principe says, “The *Mathematica* environment is excellent for prototyping and it offers good plotting facilities. And, of course, I place a high priority on the DSP. Biomedical signals (signals of physiological origin) are concentrated at low frequency, and you can do a lot of processing with the DSP—even real-time processing. Along with the NeXT user interface, this combination is what I’ve always wanted in a computer.”

Principe adds, “Computers were developed to work with text—as substitutes for typewriters—but that’s not the correct paradigm for biomedical engineering, which lives with images and signals. To make the computer work optimally for us, we needed to develop tools to function like a text editor, but for graphics and signals instead of for letters and words.”

Principe used a NeXT computer to build that tool. The Signal Editor is a software package created for NeXT computers by Principe and graduate student Haan-go Choi, used to visualize all types of electrical signal—from EEGs to machine tool data to audio waves (music). Principe says, “With the Signal Editor, the computer becomes a scope, or, actually, more like a microscope, because the computer offers very accurate time and amplitude measurements of signals.”

Working with the Signal Editor, a user can import a previously collected signal file, zoom in on specific portions, and measure amplitude/duration and slope. According to Principe, “The Signal Editor has display capability, but there’s more. It’s structured so that we can create our own signal processing functions and then visualize graphically the results of the process-



Signal Editor is shown here displaying a one-second segment of EEG containing an epileptic spike. Neurologists look for spikes in the EEG to see if the subject is epileptic, to classify epilepsies, and to study the effectiveness of anti-convulsant drugs.

ing on the com-puter screen. For instance, you can apply various digital filters with different parameters and compare the results of the filtering. So it offers an efficient way of visually comparing signal processing algorithms."

A practical medical application using this general-purpose tool relies on the Signal Editor and *Mathematica* to pinpoint the focus of epilepsy in the brain. The idea is to place a two-dimensional grid of electrodes on the exposed cerebral cortex of epileptics before brain surgery. The Signal Editor computes a spatial map of the brain's electrical field. Working with *Mathematica,* it is possible to animate a graphic representation and look for special features, that might indicate the region of epileptic focus. When removing the affected brain tissue, this will provide neuorosurgeons with the ability to plan their incisions more precisely.

Principe says, "My idea is to have this device in the operating room and to work closely with the neurosurgeon, providing a visual indication of the most probable focus of epileptic activity. Of course, it cannot be done in real time yet, but that is a goal of the project. We're still developing the algorithm and validating the method, but with the power of the DSP, we will be able to do it in real time."

The Cortical Mouse: a new computer interface

Another project in Principe's lab uses samples of brain waves. Principe explains, "The Cortical Mouse project had two motives. One was to research new ways of communication between people and computers. But what really attracted me to this project was to help quadriplegics communicate with computers."

To explore these areas, Principe and graduate student Sina Eatemadi chose to look at event-related potentials (ERPs)—transient waveforms produced by brain structures in response to external stimulus. According to Principe, "It is known that ERPs contain information about the intention of the subject and in response to stimulus from the environment. The problem with ERPs is that they are transient signals of small amplitude, and in the large background noise of the ongoing brain activity they are hard to quantify."

Conventional methods of detection use mathematical averaging techniques, but these were inadequate for Principe's purposes because he wanted real-time response. In addition, simple quantification is not enough: he needed to extract reliable parameters from the ERPs. That, according to Principe, is where neural networks come in.

To detect ERPs and obtain information from them, Principe and Eatemadi use a neural network they developed in assembly language and implemented in the DSP.
The subject's brain waves are sampled using an Ariel Digital Microphone™
, and then sent to the DSP, where the neural network is trained to recognize patterns in the ERP responses for each individual subject. The DSP processes the electrical signals and forwards them to the NeXT computer as directional cursor commands.

Principe explains, "The information encoded in ERPs is not very specific. With the present technology, we can't discriminate between actions that the subject wants to do and what he thinks. So, presently, the interface works as a dialog initiated by the computer. The quadriplegic sits in front of the computer, which has a menu-driven screen. To initiate a specified action, the subject must move the cursor to the appropriate command. So the first thing to do is control the motion of the cursor."

To do this, the computer presents the words up, down, left, and right in sequence on the screen. Every time a word is flashed, the subject is instructed to think "yes" or "no." Principe points out that the process of controlling on-screen cursor movements in this experiment is a highly structured and slow communications protocol. It is in no way simply a matter of "controlling the computer with brain waves."

"What we are currently doing is very structured and difficult to extend," he says. "But it has elements that may be highly effective for specific applications. After all, yes/no information can be extremely powerful in terms of communication. It all depends on the question, right?"

For more information, contact:
Professor Jose Principe
principe@brain.ee.ufl.edu

Academic Projects

University of Houston - Clear Lake

PhaseScope energizes the study of mathematics

Higher mathematics has traditionally been a pencil-and-paper discipline employing abstract symbols, sometimes illustrated with rough graphs, that depend on the mathematician's ability to envision correctly the relationships signified by those symbols. For years, students in upper-level mathematics courses learned techniques to solve difficult problems—such as those in the realm of differential equations—but were unable to visualize the concepts inherent in the solutions. A new computer application called PhaseScope™ may help change the way students conceptualize complex mathematical problems.

"I wanted a tool for analyzing the solutions and stability of systems of differential equations," says Michael Mezzino, chairman of the mathematics department at the University of Houston - Clear Lake. "Students can solve the equations, but lack the ability to see the dynamics of the system. PhaseScope illustrates the dynamical aspect of these problems."

Differential equations are fundamental building blocks of a class of mathematical models that attempt to describe real-world situations. Typically, they are a set of related equations, derived from natural laws and hypotheses, that express the rates of change of one or more variables—that is, these equations express change in the dynamics of a model over the course of time. A system of differential equations provides a mathematical framework for understanding dynamical systems, or situations of change, movement, or flux. Differential equations are used to plot the motion of celestial bodies in orbital mechanics, to chart the speed of enzyme reactions in chemistry, to measure the rate of radioactive decay, and to predict demographic changes in competing population groups.

PhaseScope uses the metaphor of an oscilloscope, which measures the fluctuations of an electrical current, to graph the fluctuations of these dynamical systems. With each sweep of PhaseScope's "oscilloscope," the change in a system over time is shown. The innovation of PhaseScope lies in the ease in which it allows students to make changes to the equations, experiment with different parameters, and view the results in simulated real time. For example, this exploration can lead to demonstrations of the new physics of chaos: a previously stable

system, fluctuating in a consistent manner, can, with a small change in initial values, become seemingly wildly erratic.

The operation of PhaseScope is simple. "Differential equations are entered into PhaseScope, which sends them off to *Mathematica*. PhaseScope then receives the solution from *Mathematica*, parses it, and plots it in either two or three dimensions. Students can then zoom in or out and change the perspective of the resulting graph," explains Mezzino. They can also plot multiple solutions in the same graph and change the display style of individual solutions.

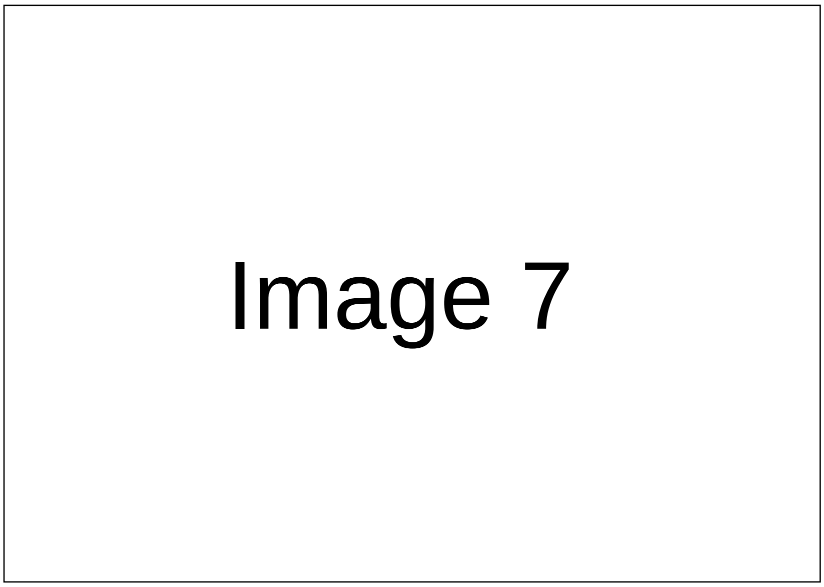
Mathematica and Interface Builder provide a solid foundation

PhaseScope rests securely on the foundations of *Mathematica* and Interface Builder. "*Mathematica* provided the power to solve the equations symbolically, just as students would—they could type in equations just as they would write them. And the extensibility of *Mathematica* allowed me to integrate it with PhaseScope, writing some of my own routines to handle certain kinds of displays and provide the user with additional numerical integration techniques which can easily be included into the application."

"The key to a product that students will use is in the interface," Mezzino continues. "I've had some experience writing software, and knew that the least appealing—and most time consuming—aspect of programming was creating the user interface. I remember when I saw a demonstration of Interface Builder with sliders and buttons that actually worked—I was amazed! Here was an application that could be used quickly to design a friendly user interface."

It took Mezzino three months of part-time work to develop the initial version of PhaseScope—"a testament to the rapid development ability of Interface Builder and the NeXTstep environment," says Mezzino. The key was learning Interface Builder, which took about three weeks. He examined the programming examples included with the NeXT bundled software and used some of them in his application. PhaseScope is now a polished software application, with an easy-to-use interface, panel and voice alerts, and the ability to save and print graphs.

Mezzino's students are already using the application, and their re-sponse to it has been positive. "The students who have used PhaseScope were electrified," he observes. "It's clear that we have been working on the right problem. My students' reactions inspire me to develop other projects like PhaseScope."



PhaseScope is a comprehensive user interface and general graphical display program for qualitatively analyzing the solution and the stability characteristics of systems of first order ordinary differential equations. The Lorenz Attractor equation, included in the Examples menu, is displayed here.

Not only have students been impressed; software professionals have also been recognizing the power and originality of PhaseScope. IMPACT Publishing, a software company based in Ithaca, New York, recently sponsored a contest for the most innovative software application developed for the NeXT computers. Mezzino submitted his application, and IMPACT awarded Phase-Scope first place. Mezzino received a prize of $1,000 and IMPACT is now pursuing distribution rights.

Building the mathematics lab for the future

The successes of PhaseScope in the NeXTstep development environment have stimulated Mezzino and his colleagues to conceive of other projects using NeXT hardware and software technology to supplement mathematics courses. He and his staff are submitting a proposal to the National Science Foundation to fund a mathematics lab of 15 NeXTstation™ Color computers and a file server, and a teaching assistant to support students in their mathematical studies.

"We're going to use the lab to fill in the rough spots in students' knowledge of calculus," Mezzino explains. The University of Houston at Clear Lake enrolls students from the junior year (in college) to masters candidates. Because they spend their first two years at other institutions, the mathematical experience and skills of entering juniors varies. A mathematics lab using *Mathematica,* electronic notebooks—structured documents that can contain *Mathematica* input and output, graphics, and text—(including those notebooks developed on other campuses), and custom software will allow students to correct their deficiencies in calculus on an individual basis.

The lab will also provide an opportunity for improved classroom instruction. Mezzino and his colleagues have already modified the syllabuses in four of their courses to include specific laboratory exercises. "The lab work will first encourage the students to explore the software, and then will set specific goals as problems for the week," he adds.

"We're especially excited about Interface Builder and *Mathematica*. These tools will significantly extend and supplement our students' work. The NeXT computers inject a sense of fun into the mathematics curriculum, and that's important," concludes Mezzino.

For more information, contact:
Professor Michael J. Mezzino, Jr.
mezzino@cl.uh.edu

Academic Project

Rose-Hulman Institute of Technology

Integrated curriculum encourages students to become problem solvers

At Rose-Hulman Institute of Technology in Terre Haute, Indiana, 39 freshmen are midway through the first year of an integrated science, mathematics, and engineering curriculum. Rose-Hulman designed the curriculum to meet two goals: to show the connections among basic concepts in technical disciplines, and to promote problem-solving skills, independent learning, and the inclination to explore.

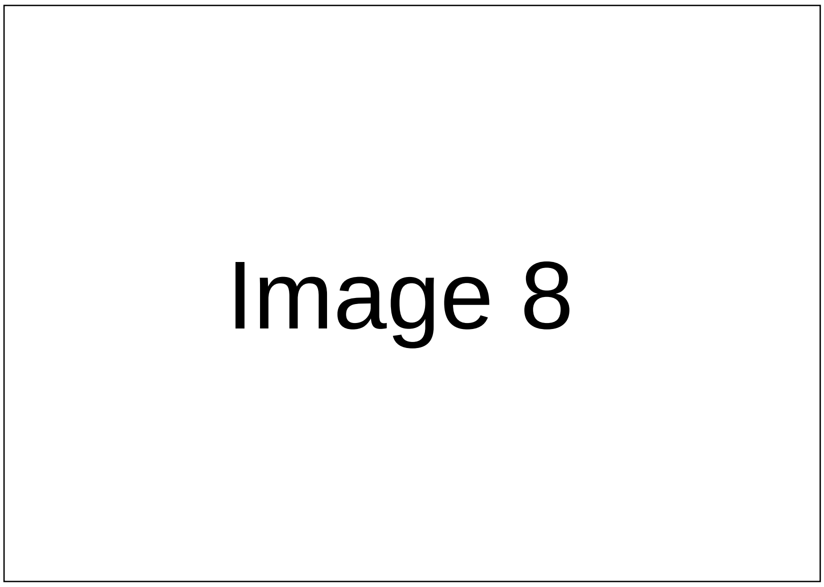
Computers provide a unifying link for the integrated curriculum

A team of five professors teach the curriculum in two classroom/computer labs equipped with 60 NeXT computers. Jeffrey Froyd, professor of electrical and computer engineering and one of the curriculum coordinators, explains that Rose-Hulman looked for a platform with a specific combination of built-in features, including networking, electronic mail, *Mathematica,* a graphical user interface, and virtual memory. Froyd says, “The user interface enables students to become familiar with the computer rapidly, and the operating system allows them to open as many applications as they need to work on a particular problem. For example, in this environment a student might simultaneously open the mail application to communicate with other members of a group, *Mathematica,* to look at data, and FrameMaker to prepare a project report.”

Custom applications help students visualize important concepts

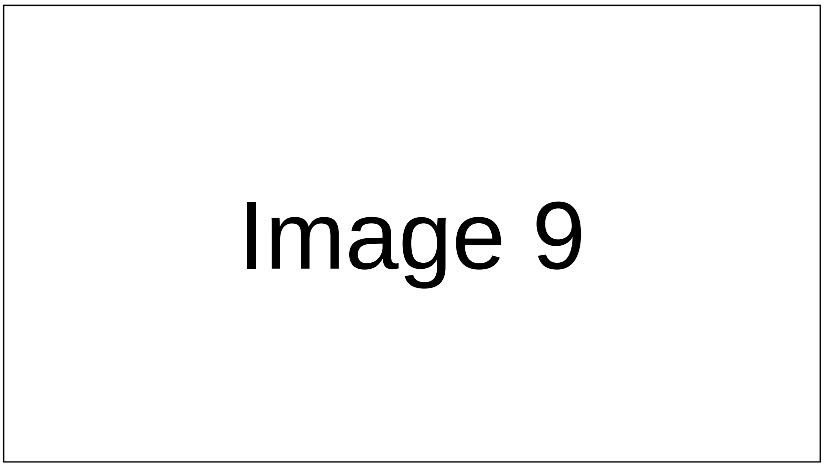
More than 30 custom applications have been developed to help students understand various topics and concepts covered in their coursework. For two months during the summer of 1990, eight Rose-Hulman students—two seniors, two juniors, two sophomores, and two freshmen—used Interface Builder to develop the applications, including the following:

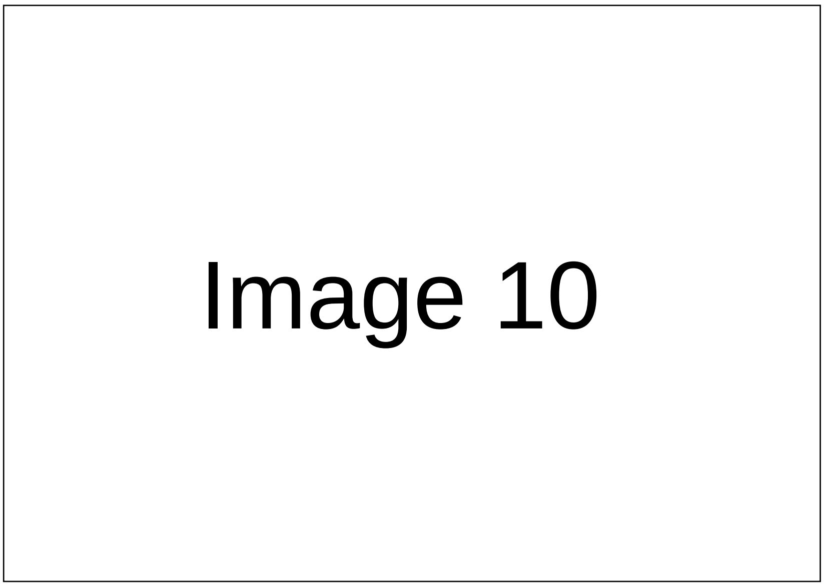
- Physics World allows students to put particles in gravitational fields, electrostatic fields, and magnetic fields, and watch their motion. Says Froyd, “The students can adjust the parameters of the particle in terms of mass and charge, they can adjust the field, they can assume the particle would be moving with some type of air resistance, and they can get a visual feel for the particle behavior.”



A File Viewer displaying some of the courseware applications developed at Rose-Hulman Institute of Technology.

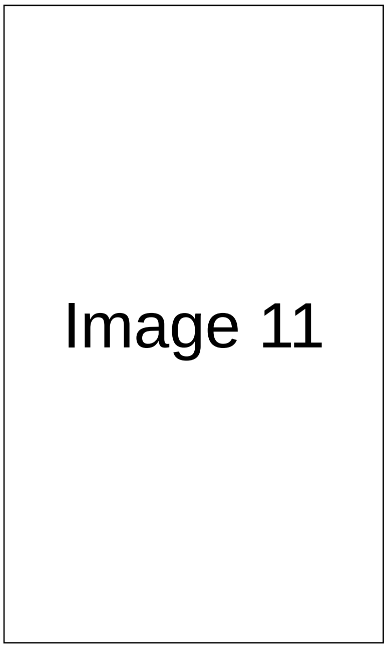
The Taylor Series application can display a function, a Taylor Series expansion for the function, and a graph comparing the original function with the Taylor Series expansion.





- Taylor Series provides an interface to *Mathematica* that makes it easier for students to find the Taylor series of a given function. Explains Froyd, “ Students can, for example, enter a function and say ‘Let’s look at the Taylor series of such an order about such a point. What does the Taylor series polynomial look like? What do the graphs (of the function itself and of the Taylor series approximation) look like? How far off are they at a certain point?’ They can play with that more readily than they could in *Mathematica*. And this kind of exploring can lead to the kind of insights we’re hoping they’ll have.”
- Function and Sons displays graphs of a function and its first and second derivatives, in random order, and asks students to identify which is which.
- Skiing forces students to do an inverse problem, says Froyd. “Instead of giving them a slope showing a skier going down it, and asking them to find the skier’s horizontal position, velocity, and acceleration, we give them graphs of these three things and say ‘Now construct the slope.’ So instead of doing analysis, they have to do synthesis. They have to do some qualitative reasoning, with some quantitative backup.”
- Slope plots a graph and allows the student to estimate the slope of a tangent line, which can then be plotted on a separate curve simply by clicking a point. “The computer shows what the derivative really looks like, giving the student visual feedback,” says Winkel.

- Area finds the area beneath a curve. Explains Winkel, “We’re trying to have the students understand that just as the notion of ‘derivative’ has many interpretations—velocity, rate, slope—t he integral has a number of interpretations, too, and one of them is a geometric one that they can see.”



Area Game finds the area beneath a curve.

- Gas Law shows the relationship of the variables in the ideal gas law. For example, the student can hold certain variables fixed and vary others—and generate real-time graphs that would be difficult to produce another way.

- Rate Law allows a student to set up a simulated chemical reaction that can proceed from reactants to products and in the reverse direction. The student can set the rate at which the process occurs. Edward Mottel, associate professor of chemistry, says this gives students a better feeling for something that often cannot be observed directly. “To be able to change variables and see what the results are will give students a much better understanding of cause and effect,” he says.

  “Immediate feedback—seeing what happens when you manipulate variables in an emulation of a physical process—is extremely helpful.”

- Field Simulator performs electric field simulations, drawing electric field lines and equipotential lines. Says Michael Moloney, professor of physics and applied optics, “The application gives students a visual sense of where the electric field is very large, and which way it’s pointing throughout the region they’re looking at—they can look at electric fields and potentials visually.”

The applications have helped explain difficult concepts and show how concepts from various disciplines within the curriculum are interrelated. “Anything we can give the students in the way of experience—visual feedback—we think is valuable,” says Winkel. “A comment we’ve heard frequently from students working with these applications is ‘I took this in high school, but never really understood it.’ ”

Creative computer use facilitates learning

The NeXT computers have also been used as the basis of several innovative assignments. In one of these projects, students were asked to listen to a recording of a golf ball bouncing (they weren’t told what it was) and calculate from what height the object was dropped. As Jerry Fine, associate professor of mechanical engineering, explains, “The golf-ball design project would have been difficult or impossible to do with any other computer. Students were able to run a spreadsheet, *Mathematica,* a digitized sound recording that could be played over and over, and a graphic representation of the waveforms associated with the sound—and all of these things integrated nicely into a desktop computer and made it easy to solve a difficult problem.”

Along with using NeXT computers for assignments and “exploration” in the computer lab, students often use the computers during class, at the professor’s request, to try new applications or plot graphs, for example. And at the end of each class, students are asked to send comments, anonymously, to a “Comments” file—providing what the faculty considers to be extremely valuable feedback. Students also use Wingz™ or *Mathematica* to analyze data they’ve collected in the school’s chemistry lab and new physics lab. Finally, they use the computers for their exams.

In addition, spending time with the computers actually increases student interaction. “They feel a sense of collegiality and closeness here,” says Winkel. “They spend a great deal of time in the computer labs, and they interact a lot: if you don’t know how to do something, you turn to the guy next to you, and if he doesn’t know, somebody else does. A lot of learning goes on.”

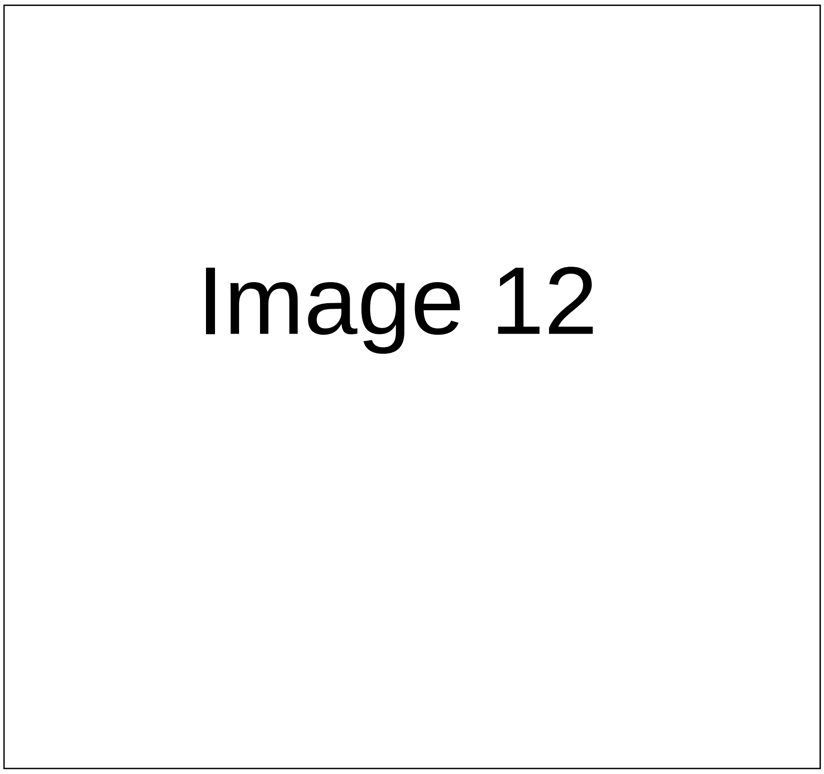
Good results already—and more on the way

"The students are becoming better self-learners," says Froyd, summarizing the results of the curriculum so far. "They're moving toward concentrating on concepts and many of the students are enjoying the approach. They're becoming more comfortable with the environment and the applications we ask them to use because they're working on a machine where you don't ask 'can it do it?,' but rather, 'how long will it take?'"

All of the faculty are looking forward to fine-tuning the curriculum. Making a point echoed in slightly different form by the other faculty members, Froyd says "One of the things that the integrated curriculum lets me do is work more closely with the students who will be studying electrical engineering—giving them a firm foundation on which to build. That way, they'll be that much more prepared for the sophomore year."

For more information, contact:
Professor Jeff Froyd
froyd@sem.rose-hulman.edu

Field Simulator performs electric simulations, drawing electric field lines and equipotential



Outside funding provides support for curriculum

In planning and establishing the integrated curriculum, Rose-Hulman received financial assistance from four organizations:

- The National Science Foundation provided funding for the NeXT computers in the classroom labs; for the new physics lab; and for support of faculty and students involved in course development, application development, and instruction.
- Lilly Endowment, Inc., provided seed money for faculty during the summer of 1988 and additional funding for the development of the final curriculum syllabus in 1990.
- The Westinghouse Foundation provided support for the students who developed applications during the summer of 1990.
- The General Electric Educational Foundation is providing funding for a thorough evaluation of the curriculum by consultants as well as by faculty from other institutions.

Jeff Froyd believes that Rose-Hulman has been successful in obtaining this funding because the philosophy embodied by the integrated curriculum meshes with a growing national awareness of the need for changes in engineering education to keep up with rapidly changing technology.

Academic Projects

Ohio Supercomputer Center

Bringing data to life in color

For the scientific researcher, mountains of raw data can often present as much of an obstacle to analysis as they do an opportunity to advance research. To harness the power of computers to reduce data to visual data form, a team of researchers at the Ohio Supercomputer Center (OSC) have created a tool that allows researchers in all data-intensive disciplines to visualize data using full-color graphics and animation. Called apE,™ which stands for animation production environment, this software is changing the way researchers see their work.

How apE works

The apE software is a graphic environment for creating scientific and artistic imagery. apE is a toolbox of predefined functions for creating, modifying, and manipulating scientific data. apE is based on a dataflow model—in a dataflow model, information flows like water through a system of pipes, and is changed along the way by filters and operators that are controlled by the re-searcher. In essence, the user builds a graphical flow chart of the application, and the apE system handles the physical details of invoking, connecting, and managing the flow of data through the system. Using this system, the raw data can be converted into a variety of sophisticated images, any of which might provide important insights into the meaning of the data.

Because the apE dataflow process can be controlled graphically, researchers can perform extremely complex three-dimensional visualizations, fully rendered and even animated, without writing a single line of computer code. Users have complete interactive control over the transformation of data.

A medical researcher, for example, can examine data collected through MRI or CAT scans in a variety of ways. Individual slices can be colored or contoured, or slices can be combined to form a three-dimensional model. Surfaces can be detected and colored within the volume, or rendered as a continuous translucent block. All these visualizations could be constructed and manipulated interactively, without ever compiling a program. To change the visualization, the scientist merely changes the flow chart. Final images can be recorded on video tape or saved for later printing on film.

Computational fluid dynamics provides another example of how apE can be used. To evaluate the flow of air along the edge of an aircraft wing, with apE a researcher can select several variables from the calculation (pressure or temperature, for instance) and apply color to the contours. In a three-dimensional simulation the researcher injects particles into the flow to observe their motion, much like releasing dye into a waterflow.

"When we were developing apE, we knew we couldn't count on scientists to use any particular type of computer, so we had to think globally from the start," says Scott Dyer, associate director of OSC and head of the apE project. For this reason, the project leaders designed apE to run in a UNIX environment, since the UNIX operating system was both powerful enough to support the software's capabilities and nearly universal throughout the academic community. Says Dyer, "We found that UNIX fully supported what we wanted to do, so we didn't aim for personal computers because they lacked the necessary operating system support for apE."

NeXT provides high-quality color

OSC houses Ohio's only public-access supercomputer facility, and has a network of approximately 50 UNIX workstations that includes NeXT, Sun,® Macintosh,® and Silicon Graphics® machines.

Says Dyer, "We in the graphics project have been strong NeXT supporters. In the past, you really needed two computers, one for UNIX tasks, one for word processing and spreadsheets. The new color machines will combine those tasks onto a single platform."

Developers at OSC believe that color should be used only when it enhances comprehension or communication. As a result, apE uses color only in actual images. Other areas, such as the application interface, do not use color.

On most of the UNIX workstations, it is common to use color-mapped framebuffers. The result is reduced color fidelity, due to having only 256 colors available, as well as distracting color shifts when switching from one application to another. With NeXTstep, apE draws the true color it wants; the Display PostScript® software then represents that color as accurately as possible on either the 16-bit or 32-bit NeXT displays. In essence, the NeXTstep environment eliminates the need for color mapping.

“We don’t have to deal with the color mapping problem with the NeXT color machines,” Dyer says. “Even on the 16-bit machine [NeXTstation Color], you can work as if you have full color available. It makes faster, better pictures possible.” The apE software runs on all NeXT computers equipped with Release 2.0 of the NeXT system software, but it is most effective on the NeXTdimension™ and the NeXTstation Color systems.

An alternative approach to software distribution

In addition to generating an innovative and practical software tool, the apE project also inspired OSC to enter the software distribution business. Members of the apE team knew they wanted to make their software widely available throughout the academic community, but were reluctant to follow the usual procedure of licensing the technology to a commercial company for marketing.

“We were concerned that the prices of software in the commercial world would make apE too expensive for our target users,” Dyer says. “We didn’t want our software to be ruined by either overcharging or lack of proper maintenance.”

The obvious but unorthodox solution seemed to be for OSC to take responsibility for marketing and distributing apE itself. To obtain the 2.0 version of apE, users have the options of magnetic tape media, and NeXT users can receive the software over a network or by sending an optical disk to OSC.

The software is large—consuming more than 100 megabytes of disk space—and includes both binary and source code. A fee of $75 covers everything, including complete documentation.

The apE team plans to continue expanding the capabilities of its software, such as allowing direct feedback between supercomputers and workstations, as well as expanding its distribution. Future plans will build on the project’s original goal: to allow as many scientists, engineers, and artists as possible to take advantage of apE to add new dimensions to their research efforts.

For more information, contact:
Scott Dyer
Associate Director,
Ohio Supercomputer Center
scott@osgp.osc.edu

Wrench is the apE data flow network construction tool. Iconic representations of common operations are joined to form a flow chart of the desired visualization.

Image 13

Campus Profiles

McGill University

Teaching, research, and new network technology

McGill University of Montreal has a full-time student enrollment of 18,000 and its faculty number approximately 3,000. Recognized for research, McGill has more than 4,000 graduate students. The university offers a three-year undergraduate program leading to a bachelor's degree, and graduate studies leading to master's and doctoral degrees.

Until last year, McGill offered programs leading to M.S. and Ph.D. degrees but undergraduate courses only in conjunction with engineering and mathematics major programs. In 1990 the Quebec government gave McGill permission to add a computer science major, and provided funds for the university to purchase computer equipment. After evaluating several computer systems, McGill selected NeXT computers and purchased 53 NeXTstation and 8 NeXTcube™ computers in addition to the four NeXTcubes the school already owned. The School of Computer Science plans to use the NeXT computers for teaching, computer architecture research, database research, software engineering, computational robotics, computational geometry, and artificial intelligence.

Selected for operating system and interoperability

"When we evaluated computer systems, our requirements were a multitasking operating system that could run a large body of existing UNIX-based software, an object-oriented environment for fast prototyping and developing, high-speed networking capability, and the ability to co-exist with our existing workstations, which include Sun, MIPS,™ IB-M,® Hewlett-Packard,® and Solbourne," says Peter Deutsch, systems manager for the School of Computer Science. The school also had stringent requirements for the operating system. "We wanted a friendly interface because many students are taking our courses as electives and we see no reason to inflict raw UNIX upon them," says Deutsch. "At the same time, our computer science students need to be able to 'lift the lid' on any conventional graphical user interface and access the raw operating system." To evaluate operating systems, Deutsch tried compiling and running a number of third-party UNIX utilities not included in the NeXT system software. "We were able to compile every tool but one without error. Underneath the NeXTstep graphical user interface is a stable implementation of Mach, which does a great job of looking like Berkeley UNIX."

Computer science teaching laboratory

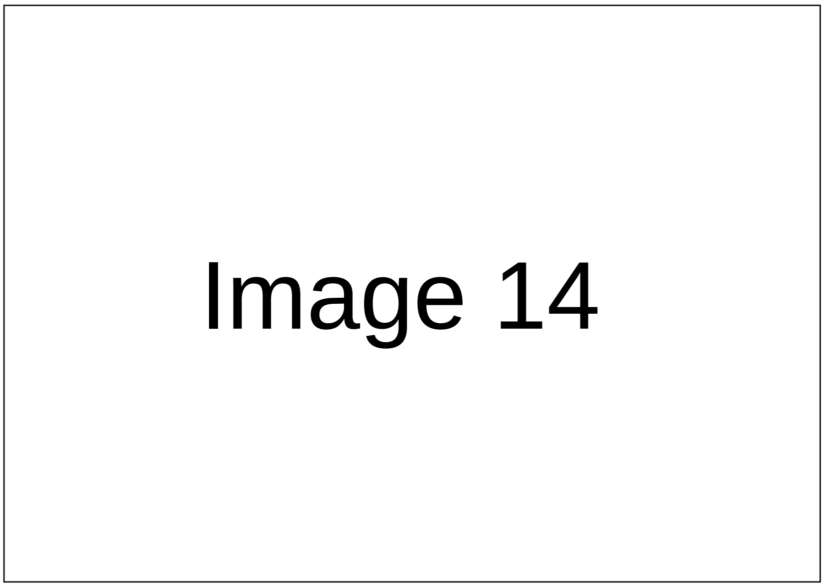
The NeXT teaching lab will have 60 NeXTstations: 50 for students and 10 for staff or for use as laser printer drivers and network information servers. The classes slated to use the lab include Files and Databases, Topics in Artificial Intelligence, Automated Reasoning, High-performance Computer Architecture, and Computer Networks. "Initially, we've selected courses that need to have access to loosely coupled parallel processing capabilities, which a network of NeXT computers provides," says Deutsch. "As we get a feel for the lab load, we may extend use of the lab to students in first-year courses, such as Comparative Languages, Compilers, and the undergraduate Computer Architecture course."

Research in parallel computation

The teaching lab will also support research in parallel computation applications. Renato de Mori, director of the School of Computer Science, plans to use the NeXT-station network to apply speech recognition algorithms to large voice samples, which he collects using Ariel dual-channel stereo microphones for NeXT computers. "Speech recognition is a compute-intensive job, that is well suited to parallel computation," says Deutsch.

McGill plans to use the NeXT computers for teaching, computer architecture research, database research, software engineering, computational robotics, computational geometry, and artificial intelligence.

To distribute computation tasks to individual NeXT com-



Students in the Person-Machine Communications course used Interface Builder to design various database interfaces. The window shows a potential interface for a music store inventory management system.

puters, de Mori's team will use Zilla, a demonstration application included with NeXTstep 2.0 that allows one master server machine to remotely execute processes on slave machines (see "Tales of Godzilla: Adventures in distributed computation, *NeXT on Campus,* June, 1990). As a particular computer completes a task, it sends the result to the server and requests more work. The research won't interfere with students' computer time because Zilla assigns a task to a particular NeXT computer only when it is not being used.

Students use NeXT tools to study person-machine communication

de Mori also uses NeXT computers to teach a course in Person-Machine Communication. "In the course, students develop interfaces using icons, menus, windows, speech, speech recognition, and speech playback," says de Mori. Last year, de Mori asked students to develop a standalone sound editor for the NeXT computer, based on the Lip Service™ panel in NeXTmail. Sound editors are used to edit speech waveforms and build messages from a basic set of spoken sentences without repeating voice acquisition. de Mori explains, "To make an announcement in an airport, for example, you need the phrase 'flight number,' followed by the digits, then the destination, then the phrase 'gate number' followed by the digits. Using a sound editor, you can create whatever sentences you need without doing voice acquisition for every possible message. There are a lot of little tricks—if you build up a message by isolated words, it doesn't sound natural."

When the School of Computer Science was evaluating NeXT computers, de Mori asked an on-staff programmer if developing a standalone sound editor for the NeXT computer was a reasonable eight-week project. The programmer studied the problem during the weekend, and came back Monday not only with a "yes," but with a complete solution. He developed a sound editor that could record, display, edit, and save a message—in 30 lines of code. "While we could have developed the sound editor using X Window™ System instead of NeXT-step, it would have required substantially more code," says de Mori. The speed of programming in the NeXT-step environment allowed de Mori to expand the assignment, asking students to customize the base sound editor for specific applications, such as making bank transactions by phone. "Our students really like the machine because it allows them to concentrate on the programming work instead of the tedium of manipulating bits," he says.

Using speech and icons in telerobotics

In another application of speech technology, de Mori is using a NeXT computer to study the use of speech and icons in telerobotics applications. "On the MegaPixel Display we simulate a robot that would assist physicians performing eye surgery,"says de Mori. "We also simulate an e ye and two microscopic needles. The physician, who presumably has his or her hands busy, gives voice commands to the robot, who manipulates the needles with more precision than a human could."

McGill’s biomedical engineering department intends to build a working prototype of the simulated robot for actual use in microsurgery.

Database application development

Faculty member Tim Merrett is using a NeXT computer to develop a new database programming language that integrates any database and an application. Called Aldat, for Algebraic Data Language, Merrett’s language is unique because it operates on any data in secondary storage (on disk), including maps and pictures as well as text. “The watchword of our concept is integration,” says Merrett. “By integrating a database into a programming language, we can provide the flexible operations for data in secondary storage that programs such as Pascal, Objective-C®, and ADA provide for data in main memory. This makes it easier for a programmer to write programs that involve database applications.”

Potential applications for Aldat include Hypertext™ programs that find a given word in all its contexts, indexing, and analyzing maps and pictures—for example, determining what state contains a given city. “Algorithms for this kind of data manipulation in main memory are well known,” says Merrett. “To have them run in a database programming language is new. Programmers who work on databases generally have to add special constructs to deal with pictures and maps. With Aldat, programmers don’t need to add anything special for maps, texts, or expert systems, because the language is based on general concepts. Aldat works at the same level with files as FORTRAN does with numbers. Where you can add numbers in FORTRAN, you can merge files—or relations—with Aldat.”

Besides integration, Aldat’s other primary advantage is the flexibility it gives the programmer. “Aldat is a flexible development language,” says Merrett. “It enables you to build a program incrementally, working from what you understand now, to what you need to build. That’s usually a no-no in software engineering. We’ve made concepts as few and broad as possible. Therefore, the language has relatively few syntactic constructs.”

The value of Aldat’s spare syntactic constructs was validated by Merrett’s experiment with artificial intelligence programming, for which the language is not specifically designed. “We rewrote an existing expert system shell using Aldat in just one man-month, compared to the several man-years it might have taken using another language.” Merrett attributes the faster development time to Aldat’s high-level language and its built-in editors. Because Aldat is such a high-level language, the shell occupies 170 lines of code. That’s an advantage in itself, because if the code is short, there’s less likelihood that bugs can creep in. Once you’ve done your editing, the data is absorbed into the relational system.” The built-in editors include a Prolog language editor—useful for artificial intelligence application development—a picture editor, and a text editor. The programmer can turn them on to do a specific function, then turn them off to return to the relational database.

In putting Aldat in perspective, Merrett states that the advance it represents is not in providing new programming capabilities, but rather in wrapping existing technology in a neater package. “We view Aldat as ordering the field of database application programming languages so it can become the next stepping stone.”

Merrett plans to develop a NeXTstep interface for Aldat. NeXT is the most favorable target machine because it

Shown here is a student-developed sound editor used in McGill’s Person-Machine Communi-cations course. The sound displayed is a voice reply for an Automatic Teller Machine.

Image 15

comes with useful software and data that we want to integrate into Aldat—for example, the Digital Librarian text database, its object-oriented interface, and PostScript® for drawing pictures and formatting documents," says Merrett. "These are all features we'll integrate into Aldat."

> *"We rebuilt the MIT Release 4 of X Window System, changing its 8-bit server for color screens into a 2-bit server for the monochrome NeXT," says Deutsch. "The initial port took just one month."*

The School of Computer Science is currently using Aldat in the undergraduate Databases course as an example of a relational database system. In graduate courses, students investigate new applications and capabilities for Aldat, helping them prepare to do their own research on language concepts.

Port of X11 Release 4

McGill also required X Window System support because the systems staff uses X Window System to manage the School of Computer Science network and to display the output from their minicomputer debugger. A staff member from the McGill Research Centre for Intelligent Machines was enlisted to port X11 Release 4. "We rebuilt the MIT Release 4 of X Window System, changing its 8-bit server for color screens into a 2-bit server for the monochrome NeXT," says Deutsch. "The initial port took just one month." The beta version runs in cooperation with NeXTstep version 1.0a. A release for NeXTstep version 2.0 is planned, and both versions will be submitted to MIT for inclusion in the standard X11 release.

Server for directory of Internet archive sites

The School of Computer Science has developed and manages a server that enables users to locate programs on the Internet, a collection of 300,000 computers at universities and research institutions around the world. Approximately 750 sites are archive servers, which make files available to other members of the Internet. These servers include software, mailing lists, bulletin boards archives, and digitized images of Hubble telescope photographs. "The archive site storage is in the gigabytes," says Deutsch. "The biggest problem users face is finding the program you want." To simplify the task of locating files, a programmer at the School of Computer Science wrote software that lists and updates all files at archive sites. It proved so popular that McGill now makes the service available to anyone on the Internet. McGill calls its server Archie, the "server server."

Archie allows users to search for specific program names or strings and find which server contains the program, or to list all the software on the server at a particular site. Currently, Archie runs on UNIX workstations. "We plan to port the server code so it will run on a NeXT computer at McGill and also to develop a graphical user interface for NeXT computer owners who want to use Archie," says Deutsch. To use Archie, users with UNIX machines can telnet to quiche.cs.mcgill.ca. [132.206.2.3] as user archie.

Versatile programming environment

Some researchers, programmers, and other faculty at McGill use NeXTstep, others use raw UNIX, and still oth“ers plan to use X Windows when the port is complete. The primary advantage of the NeXT development environment for our research is the object-oriented environment, which enables us to do fast prototyping and developing," explains Deutsch. "We consider it an enabling technology because it allows us to attack projects we couldn't have before."

For more information, contact:
Peter Deutsch
peterd@cs.mcgill.ca

NeXTstep as the platform of choice for *Mathematica* studies

by Richard E. Crandall
Director, Scientific Computation Group
NeXT Computer, Inc.

In the realms of education and research *Mathematica*'s promise is by now well appreciated. The exciting story that beckons to be told involves the emergence of NeXTstep as the outstanding platform for *Mathematica* studies. This research-er has looked into dozens of problems via *Mathematica* on NeXT computers. Without reservation he claims that if *Mathematica* were running on the alternative vendors' computers, he may have looked into only a few of these problems.

How does one justify such a pointed statement, that one "requires" NeXTstep to do the most useful *Mathematica* work? Such justification is the point of this article. The overall reason for the superiority of the platform is that in NeXTstep, *Mathematica* runs with unprecedent-ed efficiency in an unprecedented galaxy of disparate modes. Here is a list of NeXTstep advantages for the *Mathematica* user:

- Concurrent window interface: *Mathematica* runs elegantly along with everything else.
- High-resolution graphics: Certain studies require the MegaPixel resolution.
- Superior speed and memory: Many problems previously unassailable can now be resolved on NeXT computers.
- Media processing: Sounds and mages may be acquired via NeXT computers and further analyzed via *Mathematica.*
- Testing, prototyping, and porting: *Mathematica* can be the starting point for algorithms that are later ported for great speed in the C or Objective-C languages.
- nterprocess communication: Other programs may message *Mathematica* in order to obtain answers in an automated way.
- Parallelization: Multiple *Mathematica* sessions may be invoked on networks of NeXT machines.

The remainder of this treatment is dedicated to elucidation of these advantages.

Concurrent window interface

The *Mathematica* application not only has extensive and complete menus, multiple windows, and dialog panels, but all text and graphics follow the standard PostScript imaging model. This means that one may readily paste information be-tween *Mathematica* windows and other application windows. For example, there is a standard demonstration application called Circuit-Builder that generates explicit circuit equations into a window. These often formidable equations can be immediately pasted into a *Mathematica* window. When the equations are solved via *Mathematica,* one then confronts a host of fortuitous options shared by all PostScript–supporting applications. You may graph frequency response curves, or work out circuit parameters numerically and generate tables, or perform statistical analyses.

Above all, the NeXT user may keep *Mathematica* active—but "hidden"—in the application dock, ready for instant recall.

High-resolution graphics

The MegaPixel Display affords a distinct advantage in areas such as modern studies of chaotic phenomena. Whether the problem involves bifurcation, fractals, or phase transitions, one generally confronts considerable complexity for the study of which the high-resolution display is an invaluable tool. Color graphics are just as natural as gray-scale, thanks to the existence of the color paradigm as an integral part of the PostScript model.

Superior speed and memory

The claim of superior speed is by no means artificial. The author finds that *Mathematica* calculations run about an order of magnitude faster on the 68040-based NeXTstation than they will run on a Macintosh IIci. Perhaps equally important for the physical sciences, where huge data storage is often the rule, NeXT computers enjoy superior virtual memory performance. When you obtain *Mathematica* with your NeXT computer, you do not have to buy more memory to get the application to run. Furthermore, virtual memory handling is so seamless that, depending on your disk configuration, you can run sessions that require tens, or even hundreds of megabytes of storage. And finally, if you write some erudite *Mathematica* recursion for which storage runs on to infinity, you will not hurt your computer. NeXT computers are uniquely graceful in the way they handle the exhaustion of resources.

Here is an example of the joys of speed-with-memory. The author was recently presented with a tough problem, communicated by a NeXT Campus Consultant, David Bressler. Evidently, various "alternative" computer configurations had failed to solve a horrendous nonlinear system in ten unknowns, said system arising from a chromatographic model. This problem was solvable symbolically on a NeXTstation running *Mathematica*. During the successful computations, virtual memory ranged up to 20 megabytes. Most alternative computers would not have provided the required, accurate disk thrashing or the computation speed required for this problem.

Media processing

A striking feature of NeXT computers is that they may acquire sounds and images, in some situations on a real-time basis. Furthermore, one may experiment with media processing via *Mathematica.* Here is a good example: Say you have a sound file and you want to create a spectrogram or a sonogram. Read the sound file into *Mathematica,* perform the requisite Fast Fourier Transforms (FFTs) using built-in functions `Fourier[]` and `InverseFourier[]`, then graph the results in meaningful format. For image processing, you can read in an image file, then easily apply two-dimensional FFTs, or antialiasing, or de-blurring routines via *Mathematica.*

Testing, prototyping, and porting

Say you have an algebraic problem whose solution you know will necessitate an eventual expense of billions of floating point operations. You may experience the fear that if a mistake is present in your code (which could be C or Objective-C) then at least the first pass at calculation will amount to “wasted billions.” Often you can test the problem in *Mathematica* first. If everything checks out, you can transfer the calculation to the C or Objective-C environments available on all NeXT computers.

The author once planned a computation involving gigantic polynomials. These were to be finite series approximations to the function:

$$f(t) = \sin(t)\ \sin(\omega t)\ \sin(\omega^2 t)\ \sin(\omega^3 t)\ \sin(\omega^4 t)\ \sin(\omega^5 t)\ \sin(\omega^6 t)\ \sin(\omega^7 t)$$

where ω is an eighth root of unity, $\omega = e^{i\pi/4}$. But *Mathematica* investigation showed that the first few terms of f(t) are:

$$-t^8 + \frac{t^{16}}{4725} - \frac{1838\ t^{24}}{162820783125} + \frac{29039641\ t^{32}}{3028793579456347828125} + \ldots$$

This suggests that the function f(t) might be in reality a series in powers of t^8. Indeed this is rigorously so, and the testing exercise enabled the author to save a factor of 8 when the problem was ported to C routines.

Interprocess communication

The NeXTstep Application Kit allows interprocess communication via a Speaker/Listener paradigm. This means you can write an application that asks deep questions of *Mathematica*. Furthermore, you can choose to interrogate the application *Mathematica* itself, or its executable kernel. In a previous column of *NeXT on Campus,* the author de-scribed the application *Gourmet*—an experimental supercalculator (see “Adventures in supercalculator design,” Fall 1990). *Gourmet* messages the raw *Mathematica* kernel, sending problems and displaying the answers in standard ScrollViews. Similarly, *RealTimeAlgebra*—a demonstration application—does messaging between itself and the *Mathematica* application.

A striking interprocess example, which signals perhaps a new era in mathematical text processing, is an unreleased application called ExpressionBuilder, written by Josh Doenias of NeXT’s Scientific Computation Group. ExpressionBuilder lets the user to build PostScript expressions, as found in professional monographs, and to send these expressions to *Mathematica* in the appropriate format.

Parallelization

The author previously described a supercomputing network comprised of “many NeXT computers (see Tales of godzilla: Adventures in distributed computation,” *NeXT on Campus*, Summer 1990). There is now a demonstration application called Zilla, which provides a window interface for such supercomputing, with available machines appearing as icons within an application window. One may launch many *Mathematica* kernels with Zilla, each kernel being fed a unique set of seed parameters. In this way, one may multiply the power of *Mathematica* one-hundred-fold.

Afterword

The author hopes these descriptions have gone some distance in support of the idea that NeXTstep running on NeXT computers represents the platform of choice for *Mathematica* users. For further details on scientific applications in general, one might consult the author’s texts: *Mathematica for the Sciences,* released in late 1990, and *Scientific Applications for NeXT Computers,* due in early 1991, both published by Addison-Wesley.

NeXTanswers

On-line answers for commonly asked questions about NeXT computers

Have you ever wished for a personal technical support engineer? NeXT-answers provides the next best thing. It is a folder of commonly asked questions and the corresponding answers about NeXT technology. Using Digital Librarian users can easily search NeXTanswers.

NeXTanswers is a powerful, time-saving tool that is useful to everyone from a novice NeXT user to an administrator supporting hundreds of NeXT computers. Topics in-clude user applications, system administration, and programming. There is also information about NeXT user groups, archive servers, and how to contact NeXT to become a registered developer. In addition to NeXTanswers, users can obtain NeXT Technical Support Notes—longer documents that cover topics such as Serial Connectivity, UUCP, and performance tuning.

How to get NeXTanswers

NeXTanswers is updated and distributed monthly. It and NeXT Technical Support Notes are available from an authorized support center or by anonymous ftp from a public archive site, for example, Purdue University . (See "NeXT Archives," page 36.) You should feel free to distribute NeXTanswers.

NeXT also encourages users to contribute to NeXTanswers. If you want to contribute, search NeXTanswers with the key words "submitting NeX-Tanswers," and you'll find detailed instructions on how to submit your contribution. NeXT appreciates all submissions, but does not guarantee that all will be included.

Using NeXTanswers

Suppose you want to find out what kinds of SCSI hard disk drives work with software Release 2.0. Use Digital Librarian to search NeXT-answers for the key word "SCSI," and you will find the answer. All NeXT-answers follow a basic question and answer format.

Q: *What SCSI hard disk drives are compatible for use with NeXT computers?*

A: NeXT offers a complete line of hard disks for use with NeXT computer systems. They range in size from 105 MB to 1.2 GB. These drives have been carefully qualified by NeXT and include NeXT's one year warranty. Contact your authorized NeXT reseller for more information on these drives.

The NeXT system software also supports many third-party SCSI hard disk drives not sold by NeXT. For the NeXT system software to support the drive, your system must be running software Release 2.0 and be connected to the drive with the correct cable. Also, the drive must support the following SCSI commands and revisions:

- "Inquiry" and "Read Capacity" commands
- SCSI Rev. 17b or greater
- 10 (ten) byte read and write command blocks

Check with your SCSI disk drive manufacturers for details. NeXT does not provide information about third-party drives.

See also: QA579 regarding SCSI cables.

Finding More Information

The SCSI hard disk NeXTanswer references another NeXTanswer. Search for "QA579" to find more information on SCSI hard drives.

Q: *I have a 68040 upgrade kit for my computer. Do I need to order a cable adapter for my SCSI drive?*

A: The new interface cable is only needed for peripheral SCSI devices. This cable has a new 50-pin SCSI-2 type connector to the NeXT machine, and a SCSI-1 connector to the SCSI device. The part number is N7007. As of October 5, 1990, the price is $50.

The following NeXTanswers include some of the questions NeXT Technical Support groups are most frequently asked.

Deleting text in WriteNow

Q: *I deleted some text in a WriteNow document. It's still in the .wn file. Why? What's going on?*

Q: *How do I ensure that text I delete in a WriteNow document is actually removed from the .wn file?*

A: WriteNow preserves the previous version of a document in the .wn file, along with changes from that version to the current version. To ensure that deleted text is truly deleted, hold down the Alternate key while saving the file. In WriteNow's Window submenu, the " Revert to Saved" and "Revert to Backup" commands are both dimmed, indicating the backup version is no longer available. When you close the file, a panel appears briefly indicating "Reclaiming space in documents."

If you edited the file, another approach is to close the window before saving. A panel comes up in which you should click "Save/Compact." This also deletes the backup file.

Cleaning Optical Disks

Q: *Is there an approved way to clean dust from an optical disk?*

Q: *I have an optical disk that continually spins up and spins down. What can I do?*

A: Dust can cause problems when it contaminates an optical disk. For the long term, NeXT is investigating how to reduce the amount of dust that finds its way into an
optical disk.

For the present, cleaning a dusty optical disk isn't difficult. Use a soft lens-cleaning tissue (available at any photo supply store). Hold the gate open on the disk case and look at the shiny side of the optical disk. With the tissue, GENTLY wipe the disk from the center towards the edge. Rotate the disk and repeat the process until you clean the entire disk.

Appending text to the end of a text object

Q: *How can you add text to the end of a text object?*

A: The trick is to make a zero length selection at the end of the text, and replace the text there. Your code might end up looking something like this:

```
docView = [myScroller docView];
length = [docView textLength];
[docView setSel:length :length];
[docView replaceSel:newString];
[docView display];
```

Restoring the default NetInfo database

Q: *How do I go all the way back to a default (vanilla) NetInfo™ setup?*

Q: *How do I undo making a machine a Configuration Server?*

A: You need to do two things: get a default NetInfo database, and get an original host configuration database. Type the following in a Shell window:

```
cd /etc/netinfo
foreach db (*.nidb)
    mv ${db} ${db}-old
end
cp -rp /usr/template/client/etc/netinfo/local.nidb .
cd ..
mv hostconfg hostconfg.old
cp -p /usr/template/client/etc/hostconfg .
halt
```

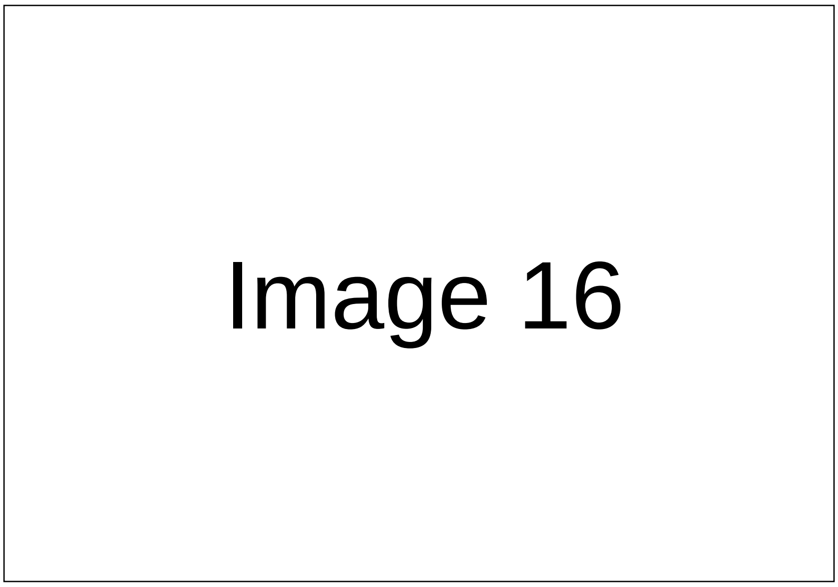
Next, disconnect the computer from the network and reboot.

Note that this is not sufficient for undoing a clone server!

Release 1.0 and 2.0 NetInfo Compatibility

Q: *Is 1.0 NetInfo compatible with 2.0?*

A: 1.0 and 2.0 NetInfo are functionally completely compatible. You may mix 1.0 and 2.0 NetInfo servers as you wish. However, there is a difference between 1.0 and 2.0 NetInfo databases. 1.0 NetInfo data files are upwardly compatible to 2.0, but not the other way around. In other words, you can move a NetInfo database from a 1.0 to a 2.0 system, but not in the other direction.



In this typical NeXTanswers session, Digital Librarian is used to search the NeXTanswers database by for the keyword "WriteNow." The Digital Librarian retrieves all appropriate NeXTanswers. The NeXTanswers window displays the answers selected from the Digital Librarian.

Academic Projects

The purpose of this section is to facilitate communication among NeXT users in higher education. Many of the projects are under development. If a project's status is not listed, please contact the project manager for more information.

This is a partial list of projects. Projects listed in previous issues of *NeXT on Campus* may not appear here due to space constraints.

Artificial Intelligence

PROTEGE—An automated system for the generation of interfaces to perform knowledge acquisition for expert systems. The interfaces help experts enter their expertise directly into an expert system's database. *Status: project is ongoing.*
Mark Musen
Assistant Professor of Medicine and Computer Science
Stanford University School of Medicine
musen@sumex-aim.stanford.edu

Business

Decision Sciences 101—A class to teach business students how to use computers in a business environment. Students learn to use WriteNow, NeXTmail, and other productivity tools.
Danail Duricy
Miami University of Ohio
Associate Professor of Decision Sciences
(513) 529-4838
duricy@next6.acs.muohio.edu

Design

DBA—A project to research issues of user interface to a relational database management system. All aspects of database use will be examined, including query, design, and administration. The goal is to produce a graphical user interface that addresses all aspects of user interaction in a system and network independent fashion.
Saeed Rahimi, Ph.D.
Professor of Computer Science
(612) 647-5153
s9rahimi@stthomas.edu
Max Tardiveau
m9tardiv@nextserver.cs.stthomas.edu
St. Thomas University

Engineering

Fast Algorithms for Signal Processing and Numerical Analysis—Work in progress includes low bitrate sound and picture compression by adapted transform coding. Experiments have yielded compression rates of up to twice the current standard. Algorithms have been de-signed to exploit the compressed data's reduced complexity, in particular Karhunen-Loeve factor analysis for pattern recognition.
Mladen Victor Wickerhauser
Assistant Professor of Mathematics
Yale University
(203) 432-7312
victor@lom1.math.yale.edu

Geography

Teaching Historical Geography—A series of modules to teach students a combination of history and geography. An introductory module teaches latitude and longitude and the difference between globes
and projections. Planned modules will focus periods ranging from Ancient Greece to geographic conflicts during the Cold War and the end of the bipolar world.
Sunil Punnoose
Director of Academic Computing
St. Mary's College of Maryland
(301) 862-0414
sunil@oyster.smcm.edu

Geology

Earthquake Hazard Estimation—*Mathematica* is used to model the dynamic response of the ground to large earthquakes. Seismic wave propagation through realistic models of the earth are shown to develop chaotic behavior. Research may help explain why earthquake damage patterns seem, in general, randomly distributed.
Jose A. Rial
Associate Professor of Geophysics
University of North Carolina at Chapel Hill
(919) 966-4553
jar@antipode.geosci.unc.edu

Language

Flexible Environment for the Recognition of Impaired Speech (FERIS)—An impaired-speech recognizer design tool that provides a dynamic environment for speech researchers to study relationships among the recognition rate, the speech parameters, and the degree of dysarthria. FERIS will also help researchers better understand the unintelligible nature of dysarthric speech. *Status: under development.*
Bon K. Sy
Assistant Professor of Computer Science
Queens College of the City University of New York
(718) 997-3500
bon@qcvax.bitnet

Grammatica—A prototype application for interactively exploring the relationships of spatial grammars to their languages. Grammatica will work like a conventional drawing application with the added ability to graphically define rules of composition. *Status: being specified.*
Christopher Carlson
Doctoral Candidate in Architecture
Carnegie Mellon University
(412) 268-6272
cc3z+@andrew.cmu.edu

The Reading Assistant—A computer-assisted system for foreign language learning. Users can browse through foreign language texts. By pointing to words on the screen, students can access a context-sensitive on-line dictionary, translation aids, and hear the proper pronunciation of words and sentences. *Status: in testing.*
Dario Giuse
Senior Scientist, School of Computer Science
Carnegie Mellon University
(412) 268-7671
dzg@cs.cmu.edu

Mathematics

Learning about Differential Equations—A three-component course in differential equations. The first component consists of "interactive lectures" using *Mathematica* notebooks to explore topics in differential equations. The second component is homework projects. Students use *Mathematica* to derive, solve, and graph differential equations. The third component is a term paper/research project. Teams of students are assigned a differential equation to investigate and must discuss the physical derivation of the equation, the linearization of the equation and special small amplitude solutions, and the solution of the equation. The term papers require the use of WriteNow, CharFind, Draw, *Mathematica,* and FORTRAN.
Steven R. Dunbar
Associate Professor of Mathematics
University of Nebraska, Lincoln
(402) 472-7236
srd@mathcml.unl.edu

Medical

The Genome Machine—A genetic information access and analysis tool. Location of genes on chromosomes, DNA and protein sequences, physical and genetic chromosome maps, and data on genetic diseases will be accessible through a graphical user interface. The Genome Machine will be used to organize and analyze data from the Human Genome Project, a 15-year effort to map the entire human genome.
Dave Adler
Research Associate in Pathology
University of Washington
(206) 543-0716
dadler@milton.u.washington.edu

Health Care Professionals Workstation—Software to enable rapid, high-quality health care decisions. The software will provide smooth integration among existing hospital and clinical systems, access to the current literature of medicine, and the ability to network with health care professionals locally and statewide. *Status: being coded; early prototype exists.*
Kent A. Spackman, M.D., Ph.D.
Associate Director for Research and Development
Biomedical Information Communication Center at Oregon Health Sciences University
spackman@ohsu.edu

MidasPlus: Molecular Interactive Design and Simulation System—A molecular modeling and drug design support system that allows real-time display of color, line, and surface displays of several interacting molecules, while qualitatively monitoring the stereo chemistry. The mouse controls three-dimensional rotations and translations to provide concise control of viewing angles and positions. MidasPlus can display full wireframe structures with depth cuing, dot cloud solvent accessible and van der Waals surfaces, and full shaded color surfaces with multiple light sources and shading. MidasPlus allows users to view the complete model or any subsegment in isolation. *Status: completed.*
Thomas Ferrin
School of Pharmacy
University of California,
San Francisco
(415) 476-1100
tef@cgl.ucsf.edu

Multimedia

Auditory Navigation of Hypertext Documents—Researchers are using quadraphonic sound cues to construct sonic maps of hypertext documents. Sound cues allow viewers to concentrate on information retrieval rather than navigation.
Diane McKerlie
diane@snowhite.cis.uoguelph.ca
Max Stevens-Guille
max@snowhite.cis.uoguelph
Graduate Students in Human Computer
Computing & Information Science Department
University of Guelph

Gita and Siddhartha—Gita is a guided tour designed to assist in understanding the *Bhagavad Gita.* Students click on buttons in the margin of the text for information on the content of specific verses. Siddhartha is a tour of the life and times of Siddhartha Gautama. The application presents a map of Northeast India. To follow the path of Siddhartha's life, students click on points on the map to bring up images and text describing these locations.
David Carpenter
Associate Professor of Philosophy
St. Joseph's University
(215) 447-4447
dcarpent@sjuphil.sju.edu

MediaView™—An application that combines text, images, live video, sound, links to other applications, and various forms of paste-on notes to create an interactive learning tool. *Status: in testing.*
Richard Phillips
Staff Member
Los Alamos National Laboratory
(505) 665-1343
rlp@lanl.gov

Music

Acoustic Tutorials: Waveform Analysis and Simple Harmonic Motion—A project to develop a comprehensive series of interactive modules to help students understand the basic concepts and phenomena of acoustics. Tutorials focus on time-varying waveforms and their analysis, simple harmonic oscillators, and acoustic wave

Academic Projects

(continued)

phenomena.
Carol Dwyer
Instructional Designer
Pennsylvania State University
(814) 863-1654
eca@psuvm.bitnet

Bessie–An interactive tutorial for teaching frequency modulation (FM) synthesis for music. The tutorial includes an explanation of FM, a historical overview of FM, and a glossary of terms. Users can experiment with the various parameters involved in FM synthesis and hear results in real time.
Anothony G. Holland
Associate Professor of Music
Skidmore College
tholland@pars.skidmore.edu

Music Composition Lab—Users compose musical pieces using Music Kit, Cmix, and CSound. Projects include interactive sound editing tools, computer-music instrument design, and acoustic simulations. Other projects include computer improvisation and digital signal processing research.
Paul Lansky, Ph.D.
Professor of Music
Princeton University
(609) 258-4241
paul@princeton.edu

Neural Network Music Composition—Algorithmic music composition using a recurrent neural network called CONCERT. CONCERT is trained on a set of melodies written in a certain style. It then composes new melodies in the same style, as well as in real time. Currently, CONCERT is being trained using J.S. Bach minuets and marches. Output can be converted to Scorefile format.
Darren R. Hardy
Research Assistant in Computer Science
University of Colorado, Boulder
(303) 492-6196

Patchmix—A graphical user interface to the Cmix music programming language.

Stochastic Granular Synthesis—An application that allows users to generate Cmix files and access the DSP chip for real-time granular synthesis.
Mara Helmuth
Graduate Student in Music Composition
Columbia University
mara@woof.columbia.edu

Understanding the Historical Temperaments—An application for the performance of musical scores in various timbres and temperaments. *Status: being coded.*
Mary Simoni
Supervisor, Public Computing Operations
University of Michigan
(313) 764-1152
mary_simoni@um.cc.umich.edu.

Cognitive Science

Associative Processing—Efficient neuro-computation algorithm simulations for pattern recognition and adaptive control applications.
Dr. Ronald D. Fellman
Assistant Professor of Electrical and Computer Engineering
University of California at San Diego
(619) 534-4913
rfellman@ucsd.edu

Connectionist Model of Short-Term Sequential Memory—A project to develop a memory model that shows some of the same kinds of memory errors humans exhibit. The network does not use weight changes to learn the order of the input sequence. Consequently, the network learns from a single presentation of a sequence, rather than through multiple presentations.
Colin Allen
Assistant Professor of Philosophy
Texas A&M University
colin@snaefell.tamu.edu

Neural Network Modeling—Researchers are using neural network simulations to explore how neuron-like elements (units) work together to transform time-varying input signals into output activities. A NeXTstep application aids the research by allowing users to check how well a network has learned a task, view past activities of simulated neurons, and impose specific input activities on any unit.
Dr. Eberhard E. Fetz
(206) 543-1440
Larry Shupe
lshupe@milton.u.washington.edu
Regional Primate Center
University of Washington

Operations Research

Intelligent Guidance for Headway and Lane Control—An expert system for vehicle guidance and a highway traffic simulator. *Status: being extended to handle uncertainty.*
Axel Niehaus
Graduate student
(609) 258-5340
aniehaus@pucc.princeton.edu
Professor Robert F. Stengel
(609) 258-5103
stengel@pucc.princeton.edu
Department of Mechanical and Aerospace Engineering
Princeton University

Utilities

The Audio Disk Controller—A special SCSI disk controller that records and plays either AES/EBU, S/PDIF or analog directly to a SCSI disk. The system supports from 4 to 196 audio channels and uses the standard NeXT file system. *Status: completed.*
Doug Karl
The Ohio State University
(614) 292-4843
doug_karl@osu.edu

B-CUBE—An ongoing software engineering student project to create a personal database. The database will be used as an on-line ad-dress and tele-

phone book. The data-base will also store short notes and dates. Individuals' records will be groupable into organization records.
Stan Osborne
Lecturer in Computer Science
San Francisco State University
(415) 338-1008
stan@cs.sfsu.edu

MailManager and EasyMail—MailManager is an advanced electronic mail system that handles local (file) and remote (IMAP2) mailboxes, multiple simultaneous mailboxes, complex message searching, ad-vanced message status management, and file attachments using RFC-1154 encoding. EasyMail is a less-powerful version of MailManager for novice users. *Status: completed.*
Mark Crispin
University of Washington
(206) 543-5762
mrc@cac.washington.edu

PopMail—A port of the University of California at Berkeley Post Office Protocol version 3 server.
Bob Debula
Ohio State University
(614) 292-4843
bobd+@osu.edu

Research Paper Management System—An article/research paper management system that stores information on various articles along with keywords. The article base is searched and abstracts from the requested articles are displayed.
Kris Kamisetty
Research Assistant in Industrial Engineering
Arizona State University
(602) 965-7055
kamisett@enuxha.eas.asu.edu

Sampling Analog to Digital Converter Interface—An analog to digital Interface Board with 12 Bit A/D conversion resolution, 0-50kHz (through ~10us) sample rate range, bipolar input voltages over range of +/-3V peak, bipolar offset and full scale adjustment on board, LED display of converted signal, 2us track and hold amplifier, and an internal clock. Output code is 2's complement binary with LSB = FS/4096 or 6V/4096 =1.46 mv.
Robert Bowman
Professor of Electrical Engineering
University of Rochester
bowman@ee.rochester.edu

Visualization, Graphics & Animation

Atmospheric Modeling—Global atmospheric modeling on remote supercomputers using NeXT computers for code development, image manipulation, and running small models.
David Randall
Associate Professor of Atmospheric Science
Colorado State University
(303) 491-8474
randall@redfish.atmos.colostate.edu

3D_Interactive_Graphics—A 3D graphics tutorial that displays a simple wireframe, solid, or Bezier surface object according to the object transformation, viewing parameters and projection types. *Status: completed.*
Yvette Perry
Graduate student
yvette@beach.csulb.edu

Objective_PHIGS—An application that can execute a C program with PHIGS function calls and display the graphical output in a view window. All PHIGS functions are implemented using Objective-C. *Status: being coded.*
Henry Chiu
Graduate student
chiu@beach.csulb.edu

The Shader—A dynamic shadow projection tutorial for design disciplines. Users can manipulate a single light source in 3D space and view the resulting shadow.
Stein Tumert
Undergraduate student
stein@beach.csulb.edu
Advisor to the three previous projects:
Dr. Michael K. Mahoney
Professor of Computer Engineering and Computer Science
California State University,
Long Beach
mahoney@beach.csulb.edu

TopDrawer—A general mathematical drawing tool used to investigate knotted surfaces in four-dimensional space. It can be used to draw and study other areas of geometric topology such as knots, links and knotted graphs. *Status: being coded.*
Dennis Roseman
Associate Professor of Mathematics
University of Iowa
(319) 335-0779
droseman@umaxc.weeg.uiowa.edu

Visual Programming—The use of meaningful graphic representation to aid parallel programming.
R.H. Perrott
Professor of Computer Science
Queen's University (Belfast)
44-232-245133, ext. 3246
r.perrott@uk.ac.qub.v1

If you would like your project(s) included in future sections of Academic Projects, please submit the following information:
Project title and discipline
Project manager's full name and title
Institution/Organization
Department
Address, phone, and e-mail address
Number of systems in department
Brief description of project
Project status

Please send to:
next_on_campus@next.com or
NeXT on Campus/Academic Projects
NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063

The world's first electronic spread-

In focus: Lotus Improv

sheet, Visicalc, was invented because Dan Bricklin wanted to make doing his business-school homework easier. Since most of the tables in his assignments contained information that could be calculated automatically from information in the tables, he figured it would save time—and tedium—if he could instruct the spreadsheet to do the calculating.

Visicalc's market successor, Lotus 1-2-3®—and the graphical interface generation that came after it, including Microsoft® Excel and countless other brands that followed—generated what has become a multibillion-dollar-a-year market in the personal computer industry. On college campuses, the electronic spreadsheet empowered faculty and students all over the curriculum—wherever calculated tables of data proved useful in making decisions and performing "what if" forecasting.

Rows and columns. . .

"Spreadsheet"—the software category Lotus' new Improv product belongs to—describes a way of working with data and numbers that significantly predates the desktop computing phenomenon. At its base level, the category describes all tabular presentations of information; specifically, it suggests the types of tables used in business, for example, rows of budgets, "spread" across columns of time.

Though spreadsheets relieved some of the effort of performing multiple calculations, programming a spreadsheet was not effortless. Building a spreadsheet required knowledge of the application's arcane command language. The concept of programming a spreadsheet was simple enough: cells are named by the row and column they occupy, and can be added, subtracted, multiplied, divided, and so forth. For example, E7 = E5+ (E8...E12)/D13 seems logical when it's initially entered. But when you look at it a few weeks later, or if what you're looking at was programmed by another person, making sense of the relationships of cells to one another—and their func-tions—can be extremely difficult.

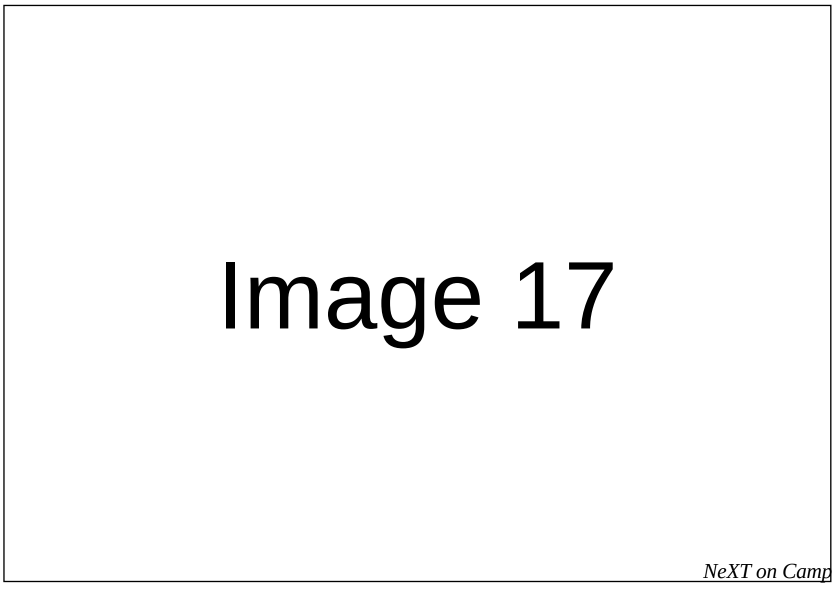
. . . categories and items

Lotus Improv is not a typical spreadsheet. With its English-like command language, you work with names that describe the rows and columns and the categories they belong to. Look at the example shown here. Products (lower left) is the category that contains the Product Summary row, plus the groups Frames and Accessories, and the rows belonging to them. Sales (upper right) is a category containing Units Sold and Revenue. Sales is in turn nested inside a category called Quarters that contains items Q1 through Q4, and all these categories are nested—in this view—in the category called Stores (upper left). Clicking the darkened arrow beside Stores changes the view of Winchester to Arlington, and its sales activity.

Formulas

The eight numbered formulas in the lower window pane describe the formula operations the way you think of them. Formulas are created by pointing-and-clicking or typing or a combination of both. Generally, formulas are cell-independent. They relate to your categories and items (for example, Product Summary = Frames.Total

In this sample Lotus Improv session, the window displays sales figures for a bicycle store and the formulas to calculate the sales figures.

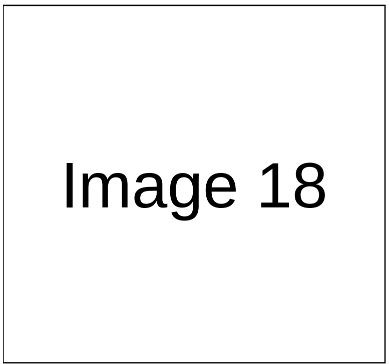


+ Accessories.Total) rather than the way typical spreadsheets are designed—by literal cell location. The formulas compute the calculated cells in the spreadsheet. If you add another store, for example, every calculated cell associated with that store is calculated. If you add new items to either Frame or Accessories, you simply add the formulas that name the unit prices for those items and multiply those prices by the appli-cable units sold entries.

In addition to clarity, this scheme of name-driven formulas provides exceptional flexibility once a spreadsheet is built. For example, suppose you want to compare sales for the two stores as rows right next to each other, sorted by item. With a conventional spreadsheet program you would have to start over and create a new spreadsheet. With Improv, you grab the Stores category name and drag it down to the inside edge of the Products category. The new view of the data appears instantly, and though the rows and columns are now rearranged, the formula logic remains the same—independent of the spreadsheet's actual cell locations.

Views

Improv spreadsheets can handle up to four categories on each of the three axes (horizontal, vertical, and paged). Any category name can be moved from axis to axis. Shifting categories, however, is only one of the ways to alter the data view. Groups (like Frames and Accessories in the example) can be collapsed to show only their totals. This lets you position items next to each other, enabling you to analyze and contrast patterns and differences. In a discipline outside business, for example, this capability allows you to create a study that replaces our categories by using Ethnic Groups instead of Stores, Occupations (grouped perhaps by College and High School to indicate educational requirements) instead of Products, Number Em-ployed and % of Eligible Population as items in Measures (replacing Sales), and Cities replacing Quarters.



Improv's charting tool, Presentation Builder.

Graphics

Improv comes with a second application called Presentation Builder, which directly messages spreadsheets via NeXTstep's Speaker/Listener facility. (For more information on Speaker/Listener in NeXTstep, see "Object orientation: from applications to environments," pages 2-7.) Presentation Builder provides a number of graph types, each offering labels, legends, scales, and other customization features. You can super-impose graphs over scanned images, and add both graphical and voice annotation comments anywhere.

Complex model management

Improv lets you save several views of a spreadsheet within an overall project model. The Model Browser lets you manage, name, and annotate each view, as well as graphs you create with Presentation Builder. In addition to multiple views, models can also contain multiple spreadsheets. Moreover, formulas communicate between spreadsheets within a model. For example, you can simplify the formula structure of the example by creating a second spread-sheet called Pricing, which is a table containing products and their prices. You replace formulas four through eight with the formula "Revenue = Units sold * Pricing::Net price," and as you change the items contained in the Pricing spreadsheet, the primary spreadsheet instantly reflects them.

The NeXT generation spreadsheet

Before Improv, we worked with typical spreadsheets by creating "collisions" at row-and-column intersections where we desired calculated results. Though these early-generation applications reduced the tedium involved in producing the spreadsheets, considerable attention was paid to the static geography of individual cell locations and the formulas imbedded in them.

Improv, on the other hand, lets you build spreadsheets by a process similar to the way you really ask yourself "what if"—by organizing categories of data and dynamically comparing and contrasting them against one another.

Third-Party Products

This section includes a sampling of software and hardware available for NeXT computers. For more information about these products, please contact vendors directly. For information on additional software and peripherals for NeXT computers, call 1-800-848-NeXT.

Software

Analytical Tools

Improv—The spreadsheet redefined with 1-2-3 compatibility and fine graphics. Included with all NeXT computers through March 1991. Lotus Development Corp.
1-800-577-8500

Publishing

The Font Company Type Library—More than 1,500 different typefaces, all in PostScript Type 1 format.
The Font Company. (602) 998-9711

FrameMaker 2.0—A technical publishing software package that includes word processing, graphics, page-layout, equation-editing, and book-building tools. Frame Technology Corp. (408) 433-3311

Illustrator 3.0—A graphic design and illustration program for generating high-quality artwork. Adobe Systems, Inc. (415) 961-4400

***Mathematica* 2.0**—A comprehensive software application for mathematical computation. Wolfram Research, Inc. (217) 398-0700

Redmark—A software application that allows multiple reviewers to proof and critique all types of documents. Options include standard proofreading symbols, drawing tools, text, voice recordings, and imported graphics. Epitome, Inc.
(615) 675-0910

TopDraw™ 2.0—An object-oriented drawing package with sophisticated illustration and layout capabilities. Media Logic, Inc. (213) 453-7744

WordPerfect®—A full-featured word processor. WordPerfect Corp. (801) 225-5000

Presentation & Training

Diagram!™—A general-purpose diagramming and charting tool. Lighthouse Design, Ltd. 1-800-366-2279

MediaStation™ 2.0—A tool for multimedia archiving, computer-based training, interactive documentation, and desktop presentations.
Imagine, Inc. (313) 487-7117

Information Management

Boss Document Management System—Work-flow and multimedia document management system for security, storage and retrieval, revision and configuration control, and notification. Boss Logic, Inc.
(515) 472-7740

Contact!™ 1.0—Business contact information database links to spreadsheets, letters, orders, invoices, and voice annotations. Boss Logic, Inc.
(515) 472-7740

OCR Servant—Full-featured optical character recognition software using automatic (omnifont) character recognition technology that ties in seamlessly with the "Services" menu feature of the NeXT's Release 2.0 operating system. HSD Microcomputer. (415) 964-1400

OnDuty™—An integrated set of utilities for office and personal tasks. Features include a phone directory and dialer; time log and contact information log; a notepad for storing, indexing, and retrieving notes; a calendar and appointment book; and a calculator. Digital Instrumentation Technology, Inc. (505) 662-1459

OMEN III—An integrated investment information systems for stockbrokersand other investment professionals. Microstat Development Corp. (604) 228-1612

PaperSight™—A personal digital file cabinet and image management system. Supports optional Kurzweil optical character recognition. Visual Understanding Systems, Inc.
(412) 687-3800

Professional PaperSight—A complete document image management solution that provides organizational features and facilitates communication among individuals. Visual Understanding Systems, Inc. (412) 687-3800

Virginia Tech InfoStation 1.0—A hypermedia information access system for library automation, including on-line catalog search and retrieval of audio, textual, and graphic data.
VTLS, Inc. (703) 231-3605

Who's Calling™ 2.0—A fully automated telephone tracking system with multiuser access and voice response.
Adamation, Inc.
(415) 452-5252

Medical

BioTRACE 8™—An eight-channel amplification acquisition, monitoring, and measurement system for biomedical research. It provides all the features necessary for digitizing, recording, real-time monitoring, data manipulation, and hard-copy output of biologically generated signals in medical research laboratories. Bio-Medical Design Group, Inc. (612) 645-9062

System Software

SoftPC 2.0™—A complete software emulation of an IBM PC, AT-class compatibility and performance. Insignia Solutions Inc. (408) 522-7600

Connectivity

3270Vision™—An integrated NeXTstep application that connects NeXT computers to IBM mainframes over SNA and TCP/IP
networks. Conextions, Inc.
(508) 475-5411

3270Vision Coax—A package consisting of 3270Vision and Coax Gateway software and an internal Coax Adapter. It connects a NeXT computer to an IBM 3174 or 3274 controller and emulates an SNA Distributed Function Terminal. Conextions, Inc. (508) 475-5411

Communicae™—DEC VT220™ and Tektronix 4010/4014™ terminal emulation and standard file transfer protocols. Active Ingredients, Inc. (617) 576-2000

FloppyWorks—File transfer software that reads and writes Macintosh 1.44 MB formatted files as well as MS-DOS® and OS/2® files in 720k, 1.44 MB, and 2.88 MB formats. FloppyWorks can be used with NeXT's internal floppy disk drive, DIT's CubeFloppy™ 2.9, and PLI's SuperFloppy 2.8. Digital Instrumentation Technology, Inc. (505) 662-1459

Microphone 3.0—Telecommunications package designed specifically for microcomputers with a graphical interface. Software Ventures Corp. (415) 644-3232

Worldtalk/400—The Worldtalk/400™ product family provides a ready-to-deploy solution for organizations that need an enterprise-wide messaging system. It interconnects many existing messaging applications, including NeXTmail.
Touch Communications, Inc.
(408) 374-2500

Languages & Tools

ACUCOBOL-85™—A COBOL compiler that produces optimized, machine-independent object code. Acucobol, Inc. (619) 271-7097

[OT Palettes: 2.0]™—A collection of custom object palettes that extend the development power of Interface Builder. Available palettes: **SmartFields,** a forms package in a field, provides an advanced set of editing, formatting, data entry, and data validation characteristics for entering numbers, dates and small amounts of text. **Chooser** provides a scrolled list of cells that may be incorporated into custom NeXTstep applications. **MathPalette** objects facilitate front-end development by providing communication between applications and the *Mathematica* kernel. MathPalette allows you to display results as single numbers, mathematical expressions, or PostScript graphics. **GraphPalette** allows fully interactive graphs within applications. The palette supplies a set of control objects which graph any type of multiple-range, two-dimensional data. Objective Technologies, Inc. (212) 227-6767

UNIX MUMPS 3.1—Massachusetts General Hospital Utility Multi-Programming System (MUMPS) is a database and programming language suited for interactive manipulation of storage of large volumes of medical and business data. Toltec Human Services. (405) 840-4254

Sound & Music

SoundWorks 2.0—A complete software package for viewing, editing, and manipulating digital audio or other sampled data. Metaresearch, Inc. (503) 238-5728

Hardware

Miscellaneous Hardware

Ariel QuintProcessor™—An add-in board with five 27 MHz M56001 DSP chips and on-board RAM.
Ariel Corp. (201) 249-2900

CubeFloppy 2.9—A floppy drive that reads and writes 3.5-inch disks (720K, 1.44MB, and 2.88MB). Each drive is shipped with a SCSI cable, FloppyWorks software, full documentation, and a one-year warranty. Digital Instrumentation Technology, Inc. (505) 662-1459

DaynaFILE™—An external disk drive that reads and writes to MS-DOS and UNIX floppy disks. Dayna Communications, Inc.
(801) 531-0600

Dazzl Model 16/12 Analog to Digital Converter—An add-in board with 16 single-ended or 8 differential channels, a maximum sampling rate of 200 kHz, and a centronics-compatible parallel port. Dazzl, Inc. (309) 674-9317

Digital Eye Color—An 8-bit color video digitizer, which accepts signals in NTSC and S-VHS formats. Metaresearch, Inc. (503) 238-5728

DoveFax—A 9600 baud fax modem and a 2400 baud data modem.
Dove Computer Corporation.
1-800-622-7627

Interfax Modem—A 9600 bps Group 3 fax modem with a 2400 bps MNP 5, Hayes-compatible data modem. Abaton. (415) 683-2870

IX-30F Image Scanner™—8-bit flatbed scanner with 300 dpi resolution. Canon U.S.A., Inc. 1-800-848-4123

PLI SuperFloppy 2.8™—An external SCSI 3.5 in. floppy disk drive capable of reading and writing to 720 KB and 1.44 MB MS-DOS or UNIX formatted disks, as well as the new 2.88 MB disk standard as implemented by NeXT. Peripheral Land, Inc. 1-800-288-8754

SCSI488/N—An IEEE 488 interface plus a software driver that enables a NeXT computer to control up to 14 IEEE instruments. IOtech, Inc. (216) 439-4091

NeXT User Groups

To find out more about a particular NeXT user group, contact the group directly or call 1-800-848-NeXT. If your user group is not listed here or if you would like to start your own group, please send electronic mail to user_groups@next.com to inquire for information.

Australia

Queensland

OzNeXT
Australian NeXT User Group
Bond University
(075) 952 291
paul@terrapin.bus.bu.oz.au

Canada

British Columbia

Vancouver NeXT User Group*
Simon Fraser University
(604) 291-4702
lionel_tolan@cc.sfu.ca

Ontario

TANG
Toronto Area NeXT Group
(416) 365-2133
David_LaVallee@next.com

Ottawa NeXT User Group
(613) 594-3352
danielc%nx1%dciem@
uunet.UU.NET

Quebec

Montreal NeXT Section of Club
Macintosh*
(514) 939-0382
paulhus@calvin.cs.mcgill.ca

France

FaNG
French area NeXT Group
(33) 67 64 07 97
73020.2442@compuserve.com

Japan

NeXus
NeXT User Society
NeXus-office@etl.go.jp
Fax: +81-03-351-0880

Mexico

Quetzalcoatl
Mexico NeXT Group
ITESM
(83) 58-20-00 ext. 4071
efrain@mtecv2.mty.itesm.mx

Netherlands

Netherlands NeXT User Group
lex@ica.philips.nl

Sweden

SnAG–Svenska NeXT-Anvandar-
Gruppen
+46 8 752 15 45
snaginfo@sics.se

United States

Alaska

Arctic Circle NeXT User Group
University of Alaska
(907) 479-2247
fsapm@alaska.bitnet

Arizona

Phoenix NeXT User Group
(602) 869-0316
jsoft!ggf@uunet.UU.NET

Tucson NeXT User Group
University of Arizona
(602) 621-2284
layhe@rcnext1.rc.arizona.edu

California

BaNG*
Bay Area NeXT User Group
Stanford University
BaNG-request@
meta-x.stanford.edu

Berkeley Mac User Group,
NeXT SIG
(415) 549-2684
bfd@meta-x.stanford.edu

CP-NUG
Cal Poly NeXT User Group
(805) 541-6879
mengel@data.acs.calpoly.edu

JPL/Caltech NeXT User Group
(818) 356-9258
carlos@eeyore.caltech.edu

NeXTUCLA
(213) 824-1174
denison@euphemia.math.ucla.edu

Nuggets
California State University, Los
Angeles
(213) 343-2400

Orange County NeXT Group
(714) 938-8850

Santa Barbara NeXT User Group
(805) 968-5584
erone%pumpkin@hub.ucsb.edu

SCaN*
Southern California NeXT Group
California State University,
Long Beach
(213) 985-1550
mahoney@grafix.cse.csulb.edu

SNuG*
San Diego NeXT User Group
(619) 481-7535
tfinn@next.com

STuN
Stanford Users of NeXT
Stanford University
louiex2@portia.stanford.edu

UC Riverside NeXT User Group
(714) 787-3883
plowe@ucrac1.ucr.edu

Colorado

rmNUG*
Rocky Mountain NeXT Users Group
(303) 530-2560
davehieb@boulder.colorado.edu

CSU NeXT Computer Users Club
Colorado State University
(303) 491-6477
dpattie@kontiki.cfnr.colostate.edu

District of Columbia

NeXT Special Interest Group*
(703) 938-NeXT
joel@next.com

Naval Research Labs NeXT User
Group
(202) 767-3524
pitre@ccf.nrl.navy.mil

Florida

Miami NeXT User Group
(305) 854-8954
mgilula@miasun.miami.edu

Georgia

BuzzNUG*
Georgia Institute of Technology
(404) 352-5551
erica%kong@gatech.edu

Illinois

Argonne NeXT User Group
Argonne National Laboratory
(708) 972-5963
henderson@mcs.anl.gov

ChiNUG*
Chicago NeXT User Group
Northwestern University
(708) 491-5368
bill_ parod@nwu.edu

Kentucky

Kentucky NeXT User Group
University of Kentucky, Lexington
(606) 258-8655
neil@s.ms.uky.edu

Massachusetts

Boston Computer Society*
BCS NeXT Group
(617) 969-6555
nextwrld!dlavin@uunet.uu.net

StrataNUG
Stratus NeXT User Group
(508) 460-2915
edw@es.stratus.com
Fax: (508) 485-8818

Michigan

Michigan State University NeXT User Group
(517) 353-9114
bonduku@ibm.cl.msu.edu

Minnesota

Minnesota NeXT User Group
Carleton College
(507) 663-4067
mtie@carleton.edu

Missouri

St. Louis NeXT User Group
(314) 343-4996
71511.125@compuserve.com

New Jersey

Princeton NeXT User Group
(609) 258-8683
jwjames@pucc.princeton.edu

New Mexico

Albuquerque NeXT User Group
Sandia National Laboratories
(505) 846-2613
jnjortn@cs.sandia.gov

Los Alamos NeXT Users Group
Los Alamos National Laboratory
(505) 667-8870
dmb@lanl.gov

New York

GUN*
Gotham Users of NeXT
(212) 227-6767
treed@object.com
Group email: gun@object.com

North Carolina

SAS NeXT User Group
SAS Institute
(919) 677-8000
pranav@unx.sas.com

Ohio

Columbus NeXT User Group*
The Ohio State University
(614) 292-4843
dyer-c@osu-20.ircc.ohio-state.edu

Oregon

Portland NeXT User Group
(503) 624-8499
baker@next.com

Oregon State University NeXT User Group
leach@satchmo.oce.orst.edu

Pennsylvania

IBUG
Interface Builder User Group
Allegheny College
IBUG@music.alleg.edu

Rhode Island

NeXT-BUG
Brown University
(401) 272-2262
agm@cs.brown.edu

Texas

Austin NeXT User Group
(512) 343-1111
pensoft!lorne@cs.utexas.edu

DaNG*
Dallas Area NeXT Group
(214) 385-2991
blackbox!kti!dirk@uunet.UU.net

hAng
Houston Area Next Group
University of Houston
(713) 749-1820
glover@uh.edu

TexNeXT
Texas A&M NeXT User Group
(409) 845-1308
daugher@cs.tamu.edu

Utah

Salt Lake City NeXT User Group
(810) 240-1017
tarbet@chemistry.chem.utah.edu

Washington

Seattle Area NeXT Group
(206) 448-0845

University of Washington NeXT User Group
(206) 543-5611
corey@cac.washington.edu

Washington State University NeXT Group
(509) 334-9594
gerkman@bongo.csc.wsu.edu

* Denotes Golden Nugget Award winners for commitment and service to the NeXT user group community.

NeXT Archives

Here is a list of some of the latest public-domain software, documentation, programming examples, and other resources available for NeXT users at public archive sites. All resources are accessible using the file transfer protocol (ftp) program, and most are available via electronic mail. If you do not know how to use ftp or access the e-mail archives, contact your campus support center.

FTP Archive Sites

Site	*Focus*	*Hostname*	*IP Address*
Oregon State University	General	cs.orst.edu	128.193.32.1
Princeton University	Music	princeton.edu	128.112.128.1
Purdue University	General	nova.cc.purdue.edu	128.210.7.22
San Francisco State University	General	sutro.sfsu.edu	130.212.15.230
University of Virginia	General	uvaarpa.acc.virginia.edu	128.143.2.7
University of Illinois	*Mathematica*	ftp.ncsa.uiuc.edu	128.174.20.50

Electronic Mail Archive Sites

Adobe	PostScript	ps-file-server@adobe.com	na
Purdue	General	archive-server@cc.purdue.edu	na

Resources

Applications

Name	*Site*	*Pathname*	*Description*
Archive	Oregon	pub/next/binaries	File compression and archiving tools
CSound-NeXTv2.0	Virginia	pub/next/Apps	A digital sound synthesis package
Default Manager	Purdue	pub/next/binaries	An interface to the defaults database
Scramble	Purdue	pub/next/binaries	Race the clock to unscramble images
Stuart 2.0	Purdue	pub/next/binaries	vt100 emulator with advanced features
Objects			
DeckofCards	Purdue	pub/next/source	Playing card and deck objects
Executive	Purdue	pub/next/source	Object to manage forked processes
Demos			
TouchTypeDemo	Purdue	pub/next/demos	A large type manipulation package
Utilities and Programming Examples			
DSP	Oregon	pub/next/source	A directory of code for the M56001 DSP
Emacs1.1	Purdue	pub/next/source	Emacs support for the mouse and meta-key
JumpBack	Purdue	pub/next/source	A back-up utility
Newsletters			
BuZZNUG	Purdue	pub/next/BuzzNUG	BuZZings and the NeXT User Journal
NeXTVieW	Purdue	pub/next/NeXTVieW	University of British Columbia journal
OnCampus	Purdue	purdue pub/next/OnCampus	*NeXT on Campus* journal
Tao	Purdue	pub/next/tao	Robert Lin’s tabloid about NeXT
Miscellaneous			
King James Bible	Washington	pub/	The Bible formatted for Digital Librarian
NeXTAnswers	Purdue	pub/next/docs/NextAnswers	Answers to technical questions

**To receive a subscription to *NeXT on Campus*,™
please complete and return this card.
Please print.**

Full Name ....

Title ....

Age: □ 18-25 □ 26-40 □ 41-60 □ 61 or older

Institution Name ....

Department ....

Address ....

City .... State ....

Zip .... Country ....

Daytime Phone ....

Electronic Mail Address ....

Check all that apply:

Faculty: □ Assistant Professor □ Associate Professor

□ Full Professor □ Department Chairperson

□ Other Faculty □ Staff

Student: □ Undergraduate □ Graduate

□ Post Graduate

Other: □ Developer □ Reseller

□ Support Center □ Service Center

Do you currently use a NeXT™ computer?

□ Yes □ No

Describe what you use your NeXT computer for and which applications you use.

What other kinds of computers do you use?

Which computer do you use most frequently and what applications do you use on that computer?

Suggestions, contributions, and subscription requests may also be sent electronically to next_on_campus@next.com