Stanford University:
The NeXT Computer fosters learning and discovery

To the students and faculty at Stanford University, computing is an integral part of every day life. The university has several computer clusters, a rigorous computer science program, and a new computer science facility. Many of the dorms have terminals and personal computers that students can use to complete assignments, write papers, and logon to the university's network.

In talking with Stanford faculty members and students who are using the NeXT Computer, a common theme arose: Computing is not only about getting day-to-day tasks done more efficiently but, more important, it is about accelerating the process of learning and discovery. These faculty members and students are using the NeXT Computer to simulate, analyze, and visualize data, information, and events. In the process, they are testing old theories, formulating new ones, and uncovering the unexpected.

Researchers at the Stanford Linear Accelerator Center develop a visual statistical analysis tool

In physics, as in many academic disciplines, researchers need to analyze vast amounts of data. Formulating new theories and testing established ones requires being able to look at the data in a variety of ways. Often, the greatest challenge lies in finding a tool that doesn't get in the way of the analysis.

At the Stanford Linear Accelerator Center (SLAC), data about particle collisions is collected from the detectors at the end of the accelerator and then processed on an IBM mainframe computer. A small group of researchers lead by Paul Kunz have developed an application on the NeXT Computer to analyze these complex data sets and to test theories of physics.

The goal of the project is to create a application that will give physics students, professors, and researchers the ability to analyze data visuallyÐin an intuitive manner.

The application lets users set up analysis chains by using the mouse rather than having to write computer code. Users simply select an input (for example, a file that contains data of physics events), determine what type of analysis should be performed, and choose the type of output to be displayed (specifically, the type of histogram, graph, or chart with which to display the analysis). Users can quickly execute a variety of cuts through the data. Because graphs and charts are produced almost instantly, users can immediately see the results of their work.

Kunz and his colleagues have already used their application on many different data sets containing thousands of physics events. ªIt used to be the case that working with a computer involved a trade-off. You'd have an idea of what you wanted to do, but completing the programming for it would often take hours. With our application, there's no programming involved, so analysis takes very little time and more tests get tried,º Kunz says.

In designing the application, Kunz and his colleagues were careful to separate general data analysis functions from from functions that are particular to physics. The application provides a pallet of objects that perform operations such as input, output, looping, and the plotting of histograms. Physics specific objects are organized on a user pallet. Users can also write their own objects to perform any number of functions and include these on the user pallet. As a result, the application can be customized to perform tasks applicable not just for physics but for any field requiring statistical analysis.

"If you remove the barriers and difficulty associated with analyzing the dataÐif you make it easier to manipulate the data, to cut and browse through itÐI believe you'll do better work. You'll discover things that would otherwise been overlooked,º says Kunz. ªYou'll actually understand things better.º

Paul Kunz
pfkeb@slacvm.bitnet

Developing a powerful searching tool for the humanities

ªI'm now pushing workstations in the humanities,º says George Drapeau, a workstation environment specialist at Academic Information Resources (AIR)Ða group of advocates in charge of academic computing at Stanford. ªAt Stanford, it's no problem to give workstations to a computer scientist or engineer,º Drapeau comments. ªWe already know how to do that. What we know less about is building tools for professors and students in the humanities.º

A year ago, a colleague who was pursuing a doctorate in German Studies offered a challenge to Drapeau: Build a powerful text searching tool for the humanitiesÐa tool that would help professors and students analyze text in ways never before possible. Drapeau accepted the challenge. He started by looking for a search engine. After examining quite a few, he recommended that Stanford purchase PATS which was developed at the University of Waterloo.

Using Interface Builder on the NeXT Computer, Drapeau designed an easy-to-use interface that enables even computer novices to perform complex searches across hundreds and thousands of pages of materialÐfor instance, on-line books, articles, and research papers. Drapeau wanted users to be not only able to perform word or phrase searches quickly, but also to perform much more complex searches.

ªSay, for example, you wanted to find all the occurrences of the theme of love in *The Adventures of Sherlock Holmes*,º says Drapeau. ªYou should be able to search for `love' as well as for its synonyms. You should also be able to do Boolean searchesÐfor example, be able to search for `love' and `death.' And you should be able to do proximity searchesÐfor instance, searching for all occurrences of `murder' and `dagger' that appear within 50 characters.

ªThe idea here is that you couldn't do this kind of analysis just by opening a book, even if you were to read it10 times. With a computer, you can,º says Drapeau. By being able to analyze the way text is organized, the types of words an author chose to use, and where certain words and phrases appear in a text, scholars can formulate and test their theories about the work.

´ A graduate student in sociology has used the program to analyze the political changes happening in China. Among other things, the searching program helped him analyze the content of reporting over time, seeing how the topics of conversation changed based on the events that week.

´ A pathology student is in the process of converting all past pathology records into digital form so that they can be searched by Drapeau's program. This will enable pathology students studying a case to easily find related cases by searching by a particular symptom, or by doing a Boolean search for a group of symptoms.

´ The secretary of the academic senate has 22 years worth of senate minutes on-line, and would like to be able to analyze themÐsay, for instance, to search for similar discussions, issues, and decisions. He intends to use Drapeau's application.

Drapeau also plans to put a NeXT Computer in the undergraduate library to give all students access to the searching tool. He is also pushing Stanford to buy more on-line text. Stanford has already put the

Bible, the entire collection of works featuring Sherlock Holmes, and numerous news files on-line.

Overall, Drapeau thinks the project is a great success. ªWhen I first came on board here, my boss asked me to figure out the future of workstations at Stanford. Then the NeXT Computer came along, and I told them they should fire me. Because the problem was pretty much solved. Hands down, it's the best for developing applications.º

George Drapeau
drapeau@jessica.stanford.edu

Synthesizing the human singing voice

At the Center for Computer Research in Music and Acoustics (CCRMA), tucked in the hills behind campus, Stanford professors and students are pushing the boundaries of computer music research. Ph.D. candidate Perry Cook's projectÐto use the NeXT Computer to synthesize the human singing voiceÐis a good example of the work they're doing.

Since the early 1950s when Bell Labs began researching how to replicate the human voice, spectral analysis models have been the primary method researchers have used for reproducing the human voice. Commonly, researchers have taken actual voice samples, analyzed them, and then tried to resynthesize the sounds with a computerÐessentially using a computer to mimic what the human ear hears.

Cook's model takes a completely different approach. His project, called SPASM, is based on physical modeling synthesis and wave guide theory. In essence, he is trying to build a computer model that represents the physics of the vocal tract. Cook explains his approach: ªUsing a spectral model, the hardest thing is making the voice sound pleasant. But if you use a physical model, or start with a model that approximates an actual vocal tract, you'll find that you can reproduce the voice much more accuratelyÐand if your model is off, you can make adjustments more easily.º

To duplicate a human vocal tract, Cook began by dividing it into a series of sections and defining the physics of the each section. Using Interface Builder, Cook then built a graphical model of a human head. By using a mouse, the radius of each segment of the vocal tract and the size of the opening into the nasal tract can be controlled.A filtered noise source can be placed at any point within the tract to simulate consonants. Transitions between sounds, for instance between two vowels, can be constructed by specifying initial and final sets of parameters, speed, and an interpolation curve for the transition.

ªIt used to be that researchers would come in at midnight to get run-time on the mainframe, because it would take nine hours just to get four seconds of sound,º says Cook. His model improves the simulation dramatically. A user can generate a voice, and then can manipulate the vocal tract and hear changes in the voiceÐall in real-time. The result is an application that Cook hopes will change the way composers and musicians work, giving them greater flexibility for studying the voice as well as for composing.

Perry Cook
prc@ccrma.stanford.edu

Creating a new computer language for AI

From his office in Margaret Jacks HallÐthe heart of the computer science department at StanfordÐchairman Nils Nilsson advises students on their artificial intelligence and robotics projects. One of these students, undergraduate Jonas Karlsson, sits at a NeXT Computer demonstrating his on-screen simulation of a robot navigating through a dynamic, changing environment. Karlsson types in a ªgoalº for the robot: ªGo to the corner.º Then he places obstacles in the robot's way. The robot slowly moves across the screen, stopping briefly when it encounters an object. Responding to that object, the robot moves around it and continues on its way. The simulation is highly interactive: Karlsson uses the mouse to reposition the robot during the simulation, forcing it to readjust and plot a new course to achieve its goal.

The NeXT Computer enabled Karlsson to test the theories underlying his robotics simulation quickly. Building this simulation on any other computer would have been a daunting task. But by using Interface Builder, Karlsson was able to develop the user interface for his simulation in only a few months. Along the way, he was able to try different ideas about how the simulation should work and make changes to the interface as the project progressed.

One of the most difficult aspects of programming a robot to perform a useful task is that the information you can feed the robot about its environment is often incomplete and is constantly changing. ActNet bridges the fields of artificial intelligence and control theory by providing an easy way for programmers to program a robot in such an environment.

To tell his robot what to do, Karlsson took advantage of ActNetÐa programming language developed by Stanford Ph.D. candidate Rebecca Moore and undergraduate Mark Torrance, also under Nilsson's guidance.

It is based on the concept of action networksÐtrees of logic gates that select actions in response to sensory and stored data. Each gate has binary inputs and thus implements a Boolean function. Essentially, by using ActNet, a programmer is able to define a series of desired actions for a robotÐªgo forward,º ªgo backward,º and ªso forth.º The robot will draw upon these actions to achieve its assigned goal, choosing the appropriate action based on feedback it gets from its environmentÐand then modifying its action based on the progress it makes.

So far, ActNet has worked exceptionally well in simulated settings. The next challenge is to use it to control multiple robots, each depending on other robots to do a portion of work. This, of course, would be directly applicable to real-world applications, for instance controlling robots to build a space station. And in fact, Nilsson is beginning experiments that will use ActNet to control actual robots at Stanford's robotics laboratory.

Professor Nils Nillson
nilsson@cs.stanford.edu