hAng ON!

The Monthly Newsletter of the Houston Area NeXT Users Group

Volume I, Issue 2 September, 1991 Only a Buck!

Thanks!!!

Steve Sarich III

The response to our first issue of hAng ON! was tremendous. We received responses from all over the country, as well as from Europe, Asia, and even the kingdom of NeXT. What really shocked us was the overwhelmingly positive nature of the mail. We received only one flaming letter, but there's one in every crowd, right? Thanks to this response, issue 2 has doubled in size.

Special thanks go the dedicated staff here, most of whom are full time developers, who have given up what few free hours they have to put out a high-end publication. Members of our staff will be attending Seybold, Comdex, and NeXTPO (Vancouver), and we will be giving you first hand accounts of new products, as well as more cool secret stuff from the Schlep.

We've got only one bone to pick this month. The cost of producing and mailing the newsletter must be paid by someone. It's time our group got organized! We must be able to collect dues in order to rent projectors, print newsletters, and put on more interesting user group meetings. Over the next month our staff will work with BANG, and other user groups, to form a plan to make our group more informative, entertaining, and financially self-sufficient. For this to work, however, we need input from you, the members! Up to this point, hAng members have not been quick to respond. Of all the "top ten" lists we received this month, NOT ONE came from within our own group! User groups have power. Believe me, NeXT does pay attention. Let's hear from you this month.

Steve Sarich III - Editor

Emergency hAng Committee Meets

An emergency session of the hAng User Group's Central Party Committee was held when it was discovered that President John Glover had fallen ill at his country estate in Katy. Chief Minister Sarich presided over the emergency meeting on Labor Day, after being informed by the hAng Internal Security Ministry that the President had fallen unconscious at his NeXT after reading the preliminary version of the second hAng ON! newsletter. Minister Sarich, along with General Loewenstern and Chief Party Ideologist Nasypany, immediately issued a declaration of grave emergency, and ordered security forces to secure the Houston area from "the forces of liberalism seeking to overthrow the good (Labor Day) party."

Minister Sarich assumed control over both the Party and the User Group, proclaiming an era of "general glee, happiness, and let's all get drunk and do silly things with our NeXT's." Within hours, proclamations of allegiance to the new government were received from Texas A&M, West Texas Polytechnics University of Southwest Abilene, and InUG (Iraqi NeXT Users Group). Unfortunately, many of the emergency committees actions were misinterpreted, and a large number of confused hAng members were inadvertently run over by several tanks. The special committee expressed regret over these incidents, and released a bulletin asking members to not lie in the streets in front of the tanks. Fortunately, the emergency declaration was quickly lifted when it was discovered that President Glover had actually fallen asleep, and was not actually ill.

Inside this issue...

The Amazing Schlepkin 2

Stupid Rumors 2

Top Ten List 2

Letters to the Editor 3

Rick Reynolds - Interview 3

C-Threads 4

JPL and the NeXT 5

Developer Camp - Review 6

You Can't Get There From Here 8

The Maxim Gun 9

What Makes a NeXT a NeXT 10

Voice Recognition on the NeXT 10

Help Wanted 13

Special Thanks 13

The All Seeing The All Knowing The Amazing Schlepkin!

The one piece of software that I'm sure all NeXT users are looking forward to is NeXT OS release 3.0. In this issue the Schlep will focus his psychic powers on new features for OS 3.0!

The most important question is not what will we get with OS 3.0, but *when* will we get OS 3.0! Although it is too early to tell for mere mortals, the Schlep has focused psychic vibrations (first thought to be gas...) that point to a release date in June. Now on to what new things you can expect to see in OS 3.0...

PostScript level 2! Finally, a tighter integration between pure PostScript and Display PostScript. With enhanced performance to boot.

The Pantone Matching System built into the color picker! I have personally seen this. Although other, lesser, platforms have software that supports PMS, only the NeXT has it built into the OS which allows *any* application to use it. With the miracle that is PostScript you can, of course, switch modes and alter the CMYK, RGB, or HSB values of any particular Pantone color. Also look for support of the CIE color model!

CD-ROM file-systems! Expect to see the ISO-9660 standard with the first release and the RockRidge extensions when they are finalized. Also look for lots of cool NeXT CD-ROM titles in the near future.

New networking standards! Software based plug-and-play solutions for Ethertalk and Novell networks.

For everyone thirsting for Renderman, relief is on the way. Although NeXT is not entirely sure how everything will be implemented, things point towards a RenderMan View class and possibly some other miscellaneous objects for manipulating RIB files.

A SQL Database kit! An array of objects for communicating with SQL databases. One of the Schlep's Astral Projections revealed that the kit might be tailored to either Sybase or Oracle.

A New and Improved Interface Builder! Besides support for the new classes, not much is know about what is new and improved, but the Schlep will keep you up-to-date.

Probably the most exciting development at NeXT is the NeXT Color Laser Printer to go along with Pantone Matching System support. Based on the Canon phase-change engine, it will use the Display PostScript interpreter as a RIP much like the current laser printer does. It will have another thing in common with it's black and white brother - it will be very cheap.•

Stupid Rumors

Inexpensive New Monitors

NeXT is getting ready to introduce two new inexpensive monitors. The standard model will be a 13" green-screen and will retail for approximately $150. The deluxe model will have a switch to go between green and amber modes and will retail for $250.

Apple][Emulator

Rumor has it that a third-party will introduce an Apple][emulator card for the cube to help NeXT penetrate the lower-ed market. A source was quoted saying "I believe that people will shell out a whole lot more than what they do for a Mac LC if they can keep the considerable investment they have made in Apple][educational software.

Apple/NeXT

Secret meetings between Steve Jobs and Apple CEO John Sculley have resulted in a joint project to release a new and innovative computer. Code named "ZERO," the new machine will run a cross between System 7.0, OS/2, and Windows.

Mip of the Month

Mip of the Month, Debbie Guy, was recently photographed for this month's centerfold. However, while staff artist Randy Warren was retouching the photograph, Icon crashed and we were forced to wash out our hard-drive with soap.•

Top Ten List

This month's Top Ten in no particular order:

1) Merging Edit with Emacs version 19 - the best of both worlds
2) Mail.app integrated on top of an industrial command-line mailer (like MH). It will be able to exchange files with CMU's Andrew Mail.
3) Another vendor supporting NeXTStep, preferable one smart enough to already have GCC, Mach, and Display PostScript on their 'normal' machines. HP is the best bet to date.
4) A (legal) Macintosh emulator, running System 7 as fast as a Mac II.
5) An X emulator that uses Workspace for the WM, so each X window has its own NeXTStep window. Allows full cut & paste between.
6) Third-Party keyboards and mice.
7) Rebundling at least a mini-database and Lisp runtime.
8) At least 8-bit color standard on NeXT motherboards - you can just plug in whatever level monitor you can afford or need.
9) NeXT will support GCC 2.0 with an integrated debugging and class browser

environment.
10) A 16mb SIMM solution.

The hAng ON! Top Ten is compiled from submissions to hAng ON!. Unfortunately, we got a pathetic number of Top Ten submissions last month. The more Top Ten submissions we get, the more accurate the one that we publish. Send your Top Ten list to hAng_ON@cubetech.com. •

Letters to the Editor

Date: Thu, 8 Aug 91 14:41:51 CDT
From: bill@gothamcity.jsc.nasa.gov (Bill Shirley [CSC] 282-1714)
To: hang_on@cubetech.com
Subject: comment...

Since Steve has proclaimed the NeXT "savior of trees", I think the least that we could do is print _hAng_ON_ on both sides of the paper.
I don't know if you intended to do this, but couldn't for the copies at the meeting last night, or not. Not to mention that 4 pages will mail with one stamp and eight pages won't.
Just thought I'd add my two cents.

-wasting away on command line mail :-(

Bill Shirley
shirley@washington.fmnet.jsc.nasa.gov

We appreciate your comment, and you will find that this month's hAng ON! is printed on both sides of the paper.

Date: Thu, 22 Aug 91 09:58:47 EDT
From: bdbusch@nxt32.darpa.mil (Brian Busch)
To: hang_on@cubetech.com
Subject: Help for NeXT (hAng ON! - August Issue)

Editor:
I read with interest your "flaming" on the predicament for "un-schooled" developers trying to get help from NeXT. I have not been able to go to the camp as well and would like to get info about printing on the NeXT. Looking at the Projects from the camp I see the only thing camp would buy me is access to the people who could answer my questions.

Thanks for the help!

Brian Busch - Computing Analysis Corp, (DARPA Contractor)
(703)527-5527
bdbusch@nxt32.darpa.mil (NeXT mail OK)

To date, no one from NeXT has responded to our developer camp editorial - but we don't give up easily.

Date: Wed, 14 Aug 91 16:03:22 EDT
From: fernwood!uunet.UU.NET!-lissie!abc (Alex Cone)
To: hang_on@cubetech.com
Subject: NeXT Developer in NY offers help & review copies of software...

Hi!
I'm in the middle of a million things, but I just read hAng ON! 1.1 between compiles & had to drop you a line. Excellent job! You can 1) write me just for grins, 2) send me any questions you cannot get answered elsewhere - we will try to figure stuff out if noone else has a clue, 3) plead with me for info/features (who knows, maybe I will have free time at some point...), 4) get demo copies of the current IB Palettes we sell (See AppWrapper for our best add to date), 5) wait for the new versions, currently under development, 6) get a review copy of OTProvide, the Sybase/Improv connection tool (DBKit version to be released with the release of DBKit) 7) get me to beta test stuff & report on it (see 3)...

Gotta run! More later!

Alex Cone
President
abc@object.com

PS Rah! Rah! NewsGrazer! Yeah!

Date: Thu, 8 Aug 1991 11:03:25 PDT
From: Lennart Lovstrand <Lovstrand@europarc.xerox.com>
To: hang_on@cubetech.com (hAng ON!)
Subject: Re: hAng ON! - August Issue - Truncated

Ha, ha, ha, you made my day, guys! Great stuff. Special kudos to the Amazing Schlepkin? I just wish he really is right (except for the RISC stuff ;-). I especially liked the "hardware and documentation delivery by former Dominos Pizza employees" suggestion. And free beer (would never say no to free beer? actually, would never say no to beer. Period.)

Cheers,
--Lennart <Lovstrand@Europarc.Xerox.COM>
Rank Xerox EuroPARC, 61 Regent St., Cambridge, CB2 1AB, UK.

Date: Fri, 9 Aug 91 23:49:55 GMT+0100
From: Gerhard_Moeller <fernwood!proximus.north.de!proximus.north.de!gemoe>
To: hang_on@cubetech.com (hAng ON!)
Subject: Re: hAng ON! - August Issue - Truncated

>So send us your thoughts (if
>you only have one you should
>probably keep it!). You can
>reach/flame us at hang_on@cubetech.com.
>This is your newsletter! We
>want to hear from you.o

I just want to know exactly what to flame... ;-)

Thanks, Gerhard.

PS.: Up to now, I love it. (Is Debbie Guy good-looking?)•

Rick Reynolds - Interview

Steve Nasypany

This months interview was with Rick Reynolds, executive director of the Bay Area Next Users Group (BANG). BANG was founded by Robert Nielsen, after asking Steve Jobs at an early developers camp dinner "Why are there no user groups?" After the dinner, Robert was asked to start

the first user group. NeXT provided some support, but then hired Robert and sent him to the London office. Robert was followed by Eric Ly, who ported Tetris to the NeXT. Finding himself saturated by school and managing BANG, Eric approached BMUG about merging the two groups. Rick, then with BMUG, was asked by its board of directors to check BANG out. "I got a cube with an early beta of 2.0 in June of 1990 - it took about a month, as I was really busy," when asked about how long it took him to become a NeXT convert. At the same time he was learning about the NeXT, Rick was dealing with Apple support "I'd get no help from them, but when I called NeXT, they were just wonderful."

BANG's core consists of about ten leaders, "who do most of the work," as Rick puts it. BANG's meetings regularly attract 100 to 200 members, depending on the topic and speaker. "We want to have something different than just another product demonstration at our meetings. You need to have lots of resources. Sometimes you need to get more new people to a meeting. Sometimes you need to encourage developers. Sometimes you need to get more money!" said Rick, whose degree is in (of all things) Modern Chinese History.

BANG's first goal is aggressive user support. BANG members will show up at local resellers in their group T-shirts and demonstrate the machine and software. BANG'ers even accompany NeXT salesmen, to provide a users view of the machine to potential buyers. Rick says that while NeXT hires top-of-the-line people, "sometimes we can sell the machines better than the salesman." Essentially, BANG's leaders are inexpensive advocates, and an indispensable resource to NeXT. "We give back ten times what they give us," he says.

Rick also places a special emphasis on communications. When a new business acquires a NeXT, Rick will offer to drive across town just to set them up with e-mail. Rick sees everyone as a user - from the lone single-app user, to developers, resellers, and even NeXT itself. Rick has even kept close ties with the Mac community, and now serves on the board of directors for BMUG - which has grown into a million dollar operation with 10,000 members. Along with his duties managing BANG, Rick has also writes manuals for several firms. Finally, Rick also serves as an editor for NeXTWORLD, and jokes that "NeXTWORLD has reached the heights of Mac Desktop Publishing."

When asked his opinion on the Apple/IBM deal, Rick quoted BaNG's most recent speakers - Guy Kawasaki and Stewart Alsop. "Guy asked if we'd seen the Spike Lee movie Jungle Fever," a movie involving an inter-racial affair. "They know that the relationship isn't going to work, their not even sure it's love - it's just jungle fever." Stewart Alsop said, "Apple has a vision, but no technology... IBM has the technology, but no vision."

Rick thinks the current application areas which seem to be most successful are those which are information and time sensitive. Although he couldn't be explicit, Rick says that NeXT recently defined a "top ten" list of priorities for the business, higher education, and graphics markets. We hope to have a copy of this list by the next issue, and if Rick doesn't get it for us, we'll make up our own and blame him. Rick has also seen Top Draw 2.0, and says that it shouldn't be judged by its predecessor. His closing advice to those many struggling NeXT developers is, "All you have to do is survive for the next 5 years, and you'll make a lot of money." •

C-Threads

Mike Barthelemy

When it comes to programming on the NeXT computer we all know about the extensive on-line documentation, but there are a few places where it is deficient. One of those places is C-Threads. C-Threads is a library of functions which allow you to use the power of the Mach kernel to its fullest for concurrent programming. In the future when NeXT offers multi- processor machines C-Threads will become more widely used, but even when limited to one processor it is an excellent tool for concurrent programming.

NeXT's documentation on C-Threads is practically identical to Carnegie Mellon Universities documentation of the package. While it explains what tools there are out there, it does not sufficiently explain how to use them.

First we have to watch our global data to ensure it doesn't become corrupted by multiple threads manipulating the same pieces of data at the same time. This can be achieved by using mutex variables and locking them before inspecting or altering the data which they protect. Let's take for example that we have a program which takes a counter and decrements it. Now, take it one step further and say that there are two threads decrementing this counter and they try to do it at the same time. What would happen without locking, if the original number were 5, is that the first thread would fetch the value of 5 and decrement it to 4 and the second thread would also fetch the value of 5 and decrement it to 4 also. This is clearly an error since the value should be 3.

What needs to be done is that the first thread has to lock the mutex for the variable, decrement it and then unlock the variable. This is achieved by using the mutex_lock() and mutex_unlock() C-Threads functions. The mutex variable is not really associated with the variable, it is entirely up to the programmer to make the associations. Here is an example of how the above threads could be written:

```
mutex_t m_counter;
int counter;

void thread_1() {
  mutex_lock(m_counter);
  counter--;
  mutex_unlock(m_counter);
  cthread_exit(0);
}

void thread_2() {
  mutex_lock(m_counter);
  counter--;
  mutex_unlock(m_counter);
  cthread_exit(0);
}
```

In this next example the second thread only wants to check the lock on the mutex and if it doesn't get the lock it continues working on other things. Specifically in this example we will have the second thread increment a counter for the number of tries it takes to get the mutex.

```
mutex_t m_counter;
int counter;
void thread_1() {
  mutex_lock(m_counter);
  counter--;
  mutex_unlock(m_counter);
  cthread_exit(0);
}

void thread_2() {
```

```
  int attempts;
  while(1) {
    attempts++;
    if
(mutex_try_lock(m_counter)) {
        counter--;
        break;
     }
  }
  mutex_unlock(m_counter);
  cthread_exit(0);
}
```

The situation may come up that you want a thread to execute after another completes a task. C-Threads provides condition variables for this which allow you to signal the completion of a threads task. Once it signals the it's done the thread waiting for the signal wakes up and then should check to make sure the data is ok for it to then complete its task.

In the example below I have 5 threads which are forking from main, which exits after forking the threads. The first thread forked off is the one which waits until the completion of the decrement_counter() threads. Then main forks off four decrement_counter() threads, and exits. At that point the one of the decrement_counter() threads takes control of the mutex for the counter, decrements the counter, and then signals finished_output() and then releases the mutex on the counter. finished_output() then locks the mutex on the counter and checks to see if the counter is zero. If it is not zero it releases the mutex on the counter and then waits for another signal.

Since you don't see any mutex_lock() or mutex_unlock() calls inside of the while() loop in finished_output, you probably wonder how it releases and locks the mutex. condition_wait() takes two arguments, the first one is the condition we're waiting for, and the second one is the mutex which we're locking and unlocking.

```
#import <cthreads.h>
#import <stdio.h>

condition_t c_counter;
mutex_t m_counter;
int counter;

void decrement_counter() {
  mutex_lock(m_counter);
  counter--;
  condition_signal(c_counter);
  mutex_unlock(m_counter);
  cthread_exit(0);
}

void finished_output() {
  mutex_lock(m_counter);
  while (counter) {
    condition_wait(c_counter,
m_counter);
  }
  printf("The program is
finished\n");
  cthread_exit(0);
}

void main() {
  int i;
  cthread_init();
  m_counter = mutex_alloc();
  c_counter = condition_alloc();
  counter = 4;

cthread_detach(cthread_fork((ct
hread_fn_t)finished_output,
NULL));
  for (i = 0; i < 4; i++)
cthread_detach(cthread_fork((ct
hread_fn_t)decrement_counter,
NULL));
  cthread_exit(0);
}
```

The above example also demonstrates the initialization necessary for using the C-Threads package. cthread_init() needs to be called once at the start of the program for the C-Threads initialization. The condition variables need to be condition_alloc()'ed and the mutices need to be mutex_alloc()'ed for either to work properly. cthread_detach() detaches a thread, and lets the program know that the thread will never be joined by another. Finally, cthread_fork() forks off a thread, as you would suspect, which is the first argument and the second argument can be used to pass something to the thread.

When writing a program using C-Threads you may experience it 'hanging' on you, this most likely is deadlock. Deadlock is when you have a mutex locked, which will never be unlocked, and you have another thread waiting for that mutex. Another way for deadlock to occur would be if you had two threads and the first thread locks mutex_1, then thread two locks mutex_2, and then thread one waits for the lock on mutex_2 and thread two waits for the lock on mutex_1. In that example they're both waiting for threads which are not going to be released until sometime after they obtain the second lock, thus neither of them execute. Special care has to be taken when using multiple mutices at the same time to ensure that deadlock does not occur.

One of the final decisions which needs to be made is how many mutices you're going to have. You could have one mutex for all of the shared data, or taking it to the opposite end of the spectrum, a mutex for every piece of shared data. The finer the granularity the greater concurrency, but with finer granularity it also takes up more overhead for mutex manipulation.

When having multiple threads work on the same input or output 'files' you must also have a mutex or mutices for them. This is to ensure that one thread does not attempt an I/O operation on the same 'file' at the same time that another thread does, which can give unpredictable results. The example above did not need to have a mutex for output to the tty because it only had one thread doing output to it, but if you had two threads doing output you could do it like this:

```
mutex_t m_ttyp1;

void thread_1() {
  mutex_lock(m_ttyp1);
  printf("This is a test of
mutex locking output files\n");
  mutex_unlock(m_ttyp1);
  cthread_exit(0);
}

void thread_2() {
  mutex_lock(m_ttyp1);
  printf("This is the second
threads output\n");
  mutex_unlock(m_ttyp1);
  cthread_exit(0);
}
```

I hope that I have done a better job than the on-line documentation available with the NeXT for describing how to use the basics of the C-Threads package. If you have any questions please feel free to ask.●

JPL and NeXT

Mission Operations and the Sir-C Project

Victor B. Taylor

JPL. Since the beginning of the space age, these three letters have been synonymous with the United States' unmanned space effort. Nestled in the foothills of the San Gabriel mountains, just north of Los Angeles, NASA's Jet Propulsion Labora-

tory has always been on the cutting edge of technology synthesis. Originally conceived as a rocket test range by a small group from the California Institute of Technology, the Lab has gained worldwide recognition as a premiere research institute in several scientific areas. Through research in electronics, mathematics, metallurgy, advanced computing systems and communications, and a host of other disciplines, JPL's mandate has been the successful robotic exploration of the solar system. Now, JPL is slowly becoming involved in yet another exploration, this time much, much closer to home: the world according to Steve Jobs.

Although boasting less than forty machines throughout the entire Lab and Cal Tech campus, those directly involved with any NeXT machine are nevertheless fanatical about the possibilities of "interpersonal computing", but also fully aware of the problems which have to be overcome before the flame can be carried to profitability and success throughout the world. One of these people, Jay Trimble, was one of the first JPLers to jump on the bandwagon with the release of system 1.0, roughly one and a half years ago. Yes, it has been quite a while hasn't it?!

Jay Trimble is currently in charge of mission operations for the NASA SIR-C project which is being managed by the Lab. SIR-C, or Shuttle Imaging Radar C for those who like to keep track of acronyms, is the third in a series of synthetic aperture radars (SAR) designed to be carried on a shuttle pallet for earth observation and imaging experiments. Currently scheduled for its first launch in 1993, the sensor will provide a reusable platform for investigators interested in the geological dynamics of Earth as a whole; these goals being somewhat similar to the objectives of the Magellan SAR which is currently in orbit around Venus and returning high-resolution images of the Venusian surface.

However, before the time of the first flight arrives, several man-years will have been spent by Jay and his small group along with their European counterparts scheduling the mission's science objectives in relation to the performance limitations of the Shuttle's hardware and software, the synthetic aperture radar itself, and the earth-based support systems which are used during a shuttle flight. In other words, the un-glamorous, but necessary world of simulation and reams of paperwork are what mission ops is all about.

Back in late 1989, the two things which caught Jay's attention about the NeXT cube was the Interface Builder (no surprise here) and the Digital Librarian. These two software packages caused a twinkle in his eye for several reasons. The Librarian for managing the large amount of paperwork that he knew would be generated during the course of the SIR-C project, and the Interface Builder and its associated object-oriented paradigm for telemetry simulation and testing. After examining other systems such as high-end Macintoshes and entries from Sun Microsystems, he decided to give the NeXT cube a try because it appeared to offer quite a bit of "bang for the buck".

Two years and one 68040 upgrade later, Jay's cadre of networked machines includes his original cube with a 660 MB internal hard drive, an optical drive, and a recent RAM upgrade to 24 MB which he says, "Makes all the difference in the world as far as speed is concerned." His group members also have two base-model grayscale NeXTStations with 105 MB hard drives. Currently on order is another NeXTStation along with a fax modem and a grayscale scanner with associated OCR software. Jay's group plans to use the fax modem to exchange formatted documentation and diagrams with its European cohorts who currently don't have NeXTs. The scanner will be used to acquire essential documents which were not created with the aid of a computer. Down the road a few months, the mission operations group plans to further utilize the NeXT's object-oriented programming environment to create hypertext-based applications for mission operations training and information dissemination to upper-level management and scientists association

One interesting side of the coin to note in the continuing debate over NeXT user and developer support is the fact that Jay (and it appears JPL in general) has received excellent support from the NeXT Southern California sales and support group. A few weeks ago, a representative from NeXT personally came out to JPL to help install Jay's third-party memory upgrade. It seems that NeXT is treating its installed base with respect and efficiency, at least from JPL's viewpoint.

"It's great, I couldn't imagine doing this project without it.", Jay said of the NeXT machine as we drove to the Burbank airport a few days ago. He also notes that he believes (and hopes) NeXT will survive. As far as his wish list is concerned, he says that it would be nice if NeXT didn't forsake the technical market by trying to exclusively sell to the "professional workstation" market. To sum it up, as always, more visible advertising in the nationwide Unix and computer journals, enhanced marketing, and an even faster machine directly aimed at the technical market and as aggressively priced as the NeXTStation grayscale and color models would probably make him believe in Santa Claus again. Although this author doesn't believe in Santa Claus, it would be a nice surprise if this Christmas eve he could look up into the dark sky and see the outline of a black, metallic sleigh bringing gifts and goodwill to believers (and nonbelievers) everywhere.•

Victor Taylor works at Jet Propulsion Laboratory and can be reached at taylor@blacks.jpl.nasa.gov

Developer Camp

Andrew Loewenstern

NeXT developer camp has been a big issue lately. People have been discussing how to offer developer camp to more people in more cities, and how NeXT could offer developer support to more people. Last week I broke down and went out to developer camp in Redwood City.

Hey Ho! Let's Go!

I really went to get the developer support and so NeXT would talk to my company. NeXT doesn't tell you what the class is really like. I went in thinking it would be some boring programmer class - boy was I wrong.

The classroom had thirty Cubes with three more as servers and one for the teacher. Everyone had an '040 with 8 megs of RAM, a 300 meg hard-drive, an optical drive, and a PLI floppy drive. Additional peripherals at our disposal included a fax-modem, three printers, three telephones, and a UUCP link to the internet so we could keep in contact with the outside world through NeXTmail. The only aspect of the provided hardware that needed improvement was more RAM for

the cubes - 12 or 16 megs would do, but when you are developing, you need at least Edit, Interface Builder, and Digital Librarian open at once. Page swapping became really annoying during and after a compile.

Although the class itself was structured, the environment was informal. We were encouraged to communicate with mail and browse through each-other's code. The classroom network was isolated from NeXT's main network and security was nearly non-existent so we could get example code out of the teacher's home directory and share code between ourselves. We were also encouraged to work together when we got stuck.

Our teacher was Randy Nelson. Randy is undoubtedly the best teacher I have ever had in my life. Besides being extremely knowledgeable on the topic he was teaching, he was very funny and had an incredible talent for explaining even the most intricate concepts in a clear and simple manner. Randy also had a seemingly inexhaustible amount of enthusiasm.

Class started officially at 9 a.m. but most of use showed up earlier to get on the machines. A continental breakfast of pastries, fruit, coffee, and assorted beverages (fruit juices, soft-drinks, and mineral water) was in the back of the room. We were encouraged to eat at our machines (as Randy said "Hey, you're programmers. You're supposed to eat at the computer."). Class continued until 12 p.m. when we had lunch. The food was pretty good ("Yes, another DCamp another 10 pounds..." Randy said.). After lunch we kept going until 3 when the "Complex Carbohydrates" arrived - an assortment of chocolate chip cookies, brownies, and fruit. Class officially ended at 5 p.m. when we had dinner. Randy would leave immediately after dinner, but he put his home phone number on the white-board in case we had any problems. Susan Rayl would stay with us until 8 p.m. when we had to leave.

The class started with the assumption that you knew nothing of programming on the NeXT. You could literally show up at the class with only a month of Macintosh experience nearly no C experience and not be too far behind. The objectives of the class were the following:

- Use the NeXT Computer to create and manage files.
- Locate and describe the sources of documentation and examples available online.
- Use Interface Builder to create an interface that complies with NeXTstep guidelines.
- Write a NeXTstep application with a custom object and a custom View, each created in the Objective-C language.

Bob Boeye

The first day consisted of how to use the computer, a few interface guidelines, and the basics of Objective-C. A projector was hooked up to Randy's machine to display "slides" (pages from WriteNow documents...) example code, and his demo applications.

The class continued with lectures on how to use InterfaceBuilder and how to integrate portions of the interface you create with your own code. Our first project was a very simple application that took a fahrenheit temperature in a text field as an input when converted it to a celsius temperature when the user clicked on a button.

Tuesday evening we did the "Calculator Lab" - an example that comes with the extended OS release. It demonstrated how to connect up buttons and text fields to your own custom objects. We were on our own for the lab, but the source to a working version was on all of our hard-drives. Most of us worked in groups of two or three.

Lectures were given on using the View class and some basic Display PostScript. The Text and NXImage classes were loosely covered. Overall, most of the commonly used classes in the AppKit were covered to some extent - more than enough foundation to get us started programming.

Wednesday evening was known as the "Cracker Barrel." For dinner we ate pizza and beer with NeXT employees. Some of the faces from NeXT in attendance were Jayson Adams, Rob Poor, Donna Simonides, and Peter Graffagnino. It gave us a chance to get to know some of the people at NeXT.

Our class was lucky enough to get a demo of the NeXT Dimension by Keith Ohlfs. Besides getting a chance to see the famous "Standard Star-Wars NeXT Dimension Demo" we got to see two of Keith's unreleased applications.

Pixelist was the first application we saw. Pixelist is basically a total rewrite of Icon with more features and without the bugs. Keith did a quick demo of composite lab by taking about 6 or 7 color images and layering them in a window to show off the Alpha Channel capabilities of the NeXT. Keith did the image on the cover of the last NeXTWorld entirely in Pixelist and Adobe Illustrator.

The other application that Keith showed us was Oscar. Oscar is a really neat animation program. You can have multiple actors and interpolate different objects between frames. It has all the features of a good commercial animation program for a Mac or PC and the only drawback lies in the SoundKit. It cannot play mer than one

sound simultaneously, so you must time your sounds to not interfere with eachother or the first sound will just stop when the second starts playing. I wonder what other great applications NeXT has that they haven't released?

All's Well that Ends Well

The class ended on Friday at 12 p.m. and Friday morning's session consisted of "Oh, by the way, you can do this..." lectures. We were also given four ED floppy disk's to back up our home directories and any example code or documentation we wanted. Randy gave out certificates to everyone who had successfully completed the course (ok, we all successfully completed the course...).

For people who knew nothing about programming on the NeXT, the course was great. It explained everything they needed to know in a clear and easily digestible way. However, it seems that for anyone who is more than casually familiar with programming NeXT computers, the course is inadequate. I had been programming for only about two weeks prior to taking the course, and really the only thing I learned new was the proper way to do some things that I was doing incorrectly (Like writing the "Fortran program in any language." - putting everything into one mega-class.).

It was a wonderful opportunity to have Randy as a teacher, but I still would have liked to take it somewhere closer to home. Many people could counter with "Well, you couldn't meet the people at NeXT if you didn't take the course in Redwood City." but I don't feel that limited informal encounters with people at NeXT were extremely productive. True a couple people left to go to dinner or what have you with some of the NeXT team, but for the majority of "campers" who don't have connections at NeXT, their money for a hotel, rental car, and airfare could have been better spent.

If you just look at the class, what it costs, and what you get after the class, it is an incredible bargain. NeXT is famous for it's killer after-camp developer support, and considering what you get from just the class, it's an even sweeter deal.•

You Can't Get There From Here

Rick Reynolds

There are some things you just can't do with a NeXT. And it drives me crazy.

One of these things is, you can't take it with you.

I spend far too much time in my car driving all over the Bay Area. I have worn grooves into the pavement in a circle going from my house to NeXTWORLD Magazine's offices to YES! Graphics & Printing to NeXT and back to my house. Often I schlep my cube but I have hated to do that since the day someone broke into my car and stole the last of my Macintosh computers.

I often run into people fortuitously around Silicon Valley and we have the impromptu meeting. I wish I could take notes. I get an idea for a column and I wish I could frame the idea into an outline on the spot. I get stuck waiting for someone and I wish I could download my Email and handle some of it while I have some time. Very often I find myself out in the field somewhere trying to help someone with a computer problem and I wish I had my own software tools from my workstation back home. A portable NeXT, or some reasonable facsimile thereof, would fill many of these needs.

We aren't an easy group to please though, we NeXT users. We are accustomed to large, sharp displays. We are accustomed to mondo hard drives. Even our floppies, until recent changes, were titans compared to anyone else's. How can NeXT's wonder engineers put together anything with as good a display, as large a hard drive, as fast, as ergonomic, as elegant and attractive as our NeXT workstations, boil them down to a few pounds, and charge under $10,000 list price? They have to or those of us who are most addicted will simply refuse to leave our offices.

I have seen and tried many of the little monsters out there right now. The Macintosh Portable is not so bad. It can't multitask, doesn't run A/UX, costs a fortune, costs another fortune to give it memory, has very limited storage options, has limited power, and weighs in at one and a half sacks of potatoes. In terms of portables, this qualifies as what Jaron Lanier of VPL would call ™two refrigerators of silicon and I still like it. Yeah, I have heard that Apple has something nifty, color, and tiny on the way, but I will evaluate it when I get to Comdex and see it. The Apple PowerBook's are just like most of Apple's products though: overpriced for what you get.

The Outbound System (once known as the Wallaby) by Outbound is a improvement over the Mac Portable. It's faster, less expensive, much smaller and lighter, has a better display, runs longer on batteries, etc., but it too doesn't run A/UX and is basically limited by all the other things that limit a mid-range Macintosh. I find the Mac Portable slightly more aesthetically pleasing too.

Those two are nearly the only ones out there with nice GUI's (graphic user interface). There are some handy, if ugly contenders from the non-GUI world too.

Last count I heard there were something like nearly 200 portables, 15 notebooks, and one super-portable available. Way too many to look at in this article, but if it doesn't run a graphical user interface, can't handle all the most common stuff (like word processing, telecom, spreadsheets, etc.) or isn't very portable, I'm not that interested. Of the remaining, two still stand out.

The Poqet PC, is one pound, nearly fits in your pocket, has a standard qwerty keyboard, last 100 hours on two AA batteries, and runs full MS-DOS. That means I can run a word processor, spreadsheet, even dial out on a modem to log onto the Internet and get my ascii email. Not too shabby at all, even for DOS. Rumor has it that Poqet, like all the other fashion-sensitive Silicon Valley firms, is doing a notebook computer as well. Code-named Eagle, it should sneak out in 1992 some time. From what I have heard, though, it won't push any buttons with me.

There is a lot to be said for the Go PenPoint. The PenPoint, in case you were so tied up in the Iraq thing that you missed the whole news, is a notebook computer redesigned from scratch along a new paradigm. The PenPoint paradigm says that a keyboard is only useful for text-based operations and should just be considered a peripheral like any other–the pen is also as useful. There are situations where the user can interact more directly with the information in the computer by touching it with a pen rather than interacting through the indirect medium of the keyboard.

That is just the surface of the PenPoint though. This, like the NeXT, is an object-oriented development environment that shows all the signs of planning and integration that comes when you sit down to design something from scratch the right way. This is in contrast to Microsoft Corporation's PenWindows‰ which is a modification of a modification of a copy of an operating system designed for a computer that no longer exists.

Go Corporation's technology is easily understood by NeXT-users on a level more subtle than most PC or workstation users. Go doesn't need to explain to us the benefits of the object-oriented approach but can go right on to explaining the complex document architecture, the portability, and so on. The ™religious aspects of both philosophies are quite compatible and they would make a fine marriage, in my opinion. I think a modest courtship overture should be carefully considered.

The best possible answer is for NeXT to do a portable. If you want something done right, do it yourself? I agree. If the resources can't be committed for that, it would be great if we could at least work out something with someone else's notebook computer so we can get our email in the field. That's all I must have.•

Rick Reynolds is Executive Director of BANG. Rick also serves on the board of directors of BMUG, the Berkeley NeXT User Group, and NOIR (NeXT Organizations International). Rick is a contributing editor for NeXTWORLD Magazine and an enthusiastic general NeXT advocate-at-large. He can be reached via email at Rick_Reynolds@BANG.org.

The Maxim Gun

Jack Reynolds

I am a white collar professional. I entered the "Professional workstation market," when I bought a Next computer about a year ago. My reasons for not purchasing another Mac or PC were simple: My competitors and co-workers were using Macs and PCs and I couldn't do anything more with mine then they could with theirs.

I work in a competitive industry and I'm surrounded by competitive co-workers. To give you an idea, most of my piers at the office have read "Marketing Warfare," and "The Art of War," to name a few. Since I am not as smart as some and not as hard working as most, I need better weapons. Competing with a different set of weapons affords me the opportunity to either defeat my adversary with superior tools or to fall on my sword trying.

Evaluating the work habits of my piers provided clues as to how I might defeat similar workers at other companies. Two things became immediately apparent:

1. People were spending less that 1.5 hours per day "using" their computers (although they always seemed to be turned on).
2. Most of the time at the battlefield was spent revising and modifying paper work (usually with the assistance of one our more administrators).

From these observations I drew the rather simple conclusion that in order to be more competitive I had to use my computer more than 1.5 hours per day and use it to do my paperwork as well as the standard applications like spreadsheets. I needed a paperless battlefield.

I explored dos and Mac based weapon systems first. Mac didn't have much in the way of a paperless office that I could find. Dos had a number of very elaborate systems available. However, all of them were way too expensive for my defense budget (under ten grand) and all of them required a pre-defined structure for the handling of the paper flow. My problem was that everything I do in terms of paper work is very fluid and unstructured. I can not define ahead of time what my combat needs are going to be in terms of indexing, document size etc.

I was initially drawn to the NeXT because of price. The things that made the dos based "document management systems" too expensive were the high resolution screens, optical storage devices, and special imaging boards that usually contained a DSP for compression and file viewing. Further, NeXT had integrated these components and others in a way that served my purposes. All this for about one fourth of the cost of a dos based system comprised of parts from many different manufactures. I was half way there.

Now, to find the software. Fortunately, I didn't have to look too hard because there weren't very many programs to look at on the NeXT. In fact, there were only two that did document management. One was $25,000 and one was $595. I decided to look at the cheap one first.

PaperLight, solved my problem of unstructured paper flow in a unique way. It used a hypertext indexing system to attach key words to faxes and scanned documents. These key words could either be entered manually or entered by way of optical character recognition. Any of these words or combination of words could strategically reference any document on the system. Additionally, icons from the browser could be attached and filed with the documents in a similar way to how I had seen icons used with NeXTMail.

I arrived at the office with my new NeXT weapon early on Saturday morning to begin furiously scanning in documents out of my desk and filing cabinets. Dealer applications, credit reports, product literature, correspondence, wire transfers, market analysis, contracts, customer files, competitive information. By Sunday night, I had loaded most of the important munitions. Over 1,500 pages worth. I was pleased to see that because the DSP had performed on the fly group IV compression of the documents, they took up less than 100 MB on optical.

Monday morning I returned to the battlefield. I was a little tired. But, ready for my first skirmish. I got defeated... It was horrible. It was more than twice the work to do battle. I was having to operate in the old manual mode in parallel with my new system of being paperless. It was more work than I could manage. I started falling farther and farther behind. I didn't have the coordination to capitalize on the NeXT multitasking environment, I didn't have the time to scan in all the new paper, and I didn't have the courage to abandon the manual filing system. I was loosing the war.

Just as I thought all was lost, I was rescued by two brand new deadly weapons. One was code named "040." The other "Integrated FAX/Modem Receipt" or IFR for short. The 040 allowed me greater speed in loading ammunition from the scanner, faster searching through documents, and more timely document reconnaissance. The IFR combined with a dedicated communication line provided unattended automatic loading. This along with my increased mission experience

started to turn the tide. The combined power of my weapons system became intoxicating. I abandoned the manual filing system.

My life has been changed forever. Each day I get more deadly at using my weapons. Six active applications? No problem. Simultaneous faxing, scanning, document retrieval, all while talking on the phone? Piece of cake. I am a very dangerous man. And the best part of all is that my weapons are stealth. Totally transparent. My competitors never know what hit 'em.

Here's a typical battle:

My intercom announces, "It's Steve, he wants to know where his order is at."

"Put him through," I snap as I pounder which "Steve" is calling me.

I type in the word "Steve." Twelve entries flash on the screen. I quickly deduce that since he is looking for an order. He must be a customer.

The phone rings as the call is transferred. I click on the word "order." Only three entries now.

The second ring heightens the intensity.

Quickly, glancing at the screen I notice that only one of the three files has had an entry lately. It must be Steve Sharbock. I double click on the record and reach for the phone just in time to choke off the third ring.

"Hi Steve," I say in a calm voice. "What can I do for you."

Before Steve utters his greeting I'm already perusing his most recent faxes and letters. "Hi Jack, we got shorted one unit on our order. We should have gotten three not two."

A smirk appears on my lips as I view the order that he sent on the fax. I return fire, "I've got the fax right here Steve. Yep, you should have gotten three." I type in, "did you ship three?" on the order and launch the modified document off on the fax to the warehouse just as Steve starts small talking.

"Got anything new going on product wise?" he taunts.

I click off some literature to Steve's fax machine as I reply, "Sure, I'll send you some stuff." Knowing full well that he'll get it before we even hang up the phone.

Just as Steve and I end our conversation, the intercom announces, "It was the warehouse. They're going to send the other unit out right away."

As I close Steve's documents, I rest easy knowing that this skirmish has been permanently recorded, expediently handled, and a minimum of my time has been expended.

HILAIRE BELLOC 1870–1953
Whatever happens, we have got
The Maxim Gun, and they have not.
vi•

Jack Reynolds is Vice President of Sales at HSD Microcomputer and can be reached at reynolds@montana.portal.com.

What Makes a NeXT a NeXT?

Ernest Prabhakar

This question has been at the root of a lot of debate recently, so I thought I'd try to push the discussion a little deeper.

Obviously, a NeXT is many things to many people. It is:

- a good bang/buck Unix box.
- a reasonably priced X server
- an overpriced X terminal
- an extremely overpriced PC-AT (i.e., Soft-PC).
- a Mac on steroids
- a nifty multi-media platform

All these are not bad, and if it causes people to buy a NeXT, wonderful.

However, I believe the NeXT is more than that. It is a dream.

A dream of taming the wild stallion that is Unix. A dream of improving interpersonal productivity. A dream of users having the power of programmers. A dream of breath-taking graphics. A dream of applications never before imaginable.

Sure, I'll admit that the NeXT is only a rough approximation of that dream. But it is the embodiment of it. While others talk about the dream admiringly, wistfully, or scornfully, NeXT users (like Jayson Adams) roll up their sleeves and try to mash reality into the shape of their dreams.

And of course NeXT needs to conform wherever possible. No computer is an island. What advantage is there is idiosyncratic independence? But what happens if the dream conflicts with the way things are done?

People say that NeXT, above all, should be compatible. Drop NeXTStep, push X, conform to the standards, toe the line, don't be different. Be a low-cost, X/Unix terminal. If you don't, you will die.

Well, maybe so. But if NeXT did that, the dream would die. Frankly, I'd rather have a live dream and a dead company than the other way around. Because even if NeXT dies, having pushed technology and their luck as far as they could, they would have changed the world.

Look around you. People are trumpeting Motif and GUIDE and System 7.0 as if they were the greatest things since sliced bread. If it weren't for the NeXT, vendors would claim that such was the state of the art, and it was unfair of users to expect any more than that. If NeXT caved in, everybody else could breathe a sigh of relief and be content with the status quo.

[okay, so I'm dramatizing slightly].

People call this a religious attitude. Well, if religious means loving a dream of how things should be, and following it even if it means your death, then yes, I accept the title with pride. And hope NeXT will do the same.

Because it isn't how much you win or lose, it's how much you dare.•

Ernest Prabhakar is the president of CaJUN, the Caltech NeXT User Group. His e-mail address is ernest@pundit.cithep.caltech.edu.

Voice Recognition and the NeXT Computer

Challenges, Opportunities, and Opinions

Paul R. Dietz

Preface

By way of disclaimer, all material contained in this work, except where noted, is based on the opinions and knowledge of the author. For the establishment of credibility in regards to the subject, the author offers the following information.

Toltec Human Services is a registered

NeXT developer, #832. Our primary development focus is medical informatics; that is to say we develop applications on the NeXT platform for use by the medical community both domestic and internationally. Toltec was formed specifically to develop software technology for the NeXT. We do not develop for other platforms and indeed, have no plans to do so. The primary reason for this "attitude" is based on our knowledge and perception of the medical profession and the principle of "least common denominator".

The president and founder of Toltec is Paul D. Orcutt, M.D., FACEP. Dr. Orcutt selected the author to build the company to achieve one purpose; to improve the quality of health-care for AS MANY PEOPLE AS POSSIBLE THROUGH TECHNOLOGY. Our development partners are, for the most part, physicians and medical scientists. The author served in the USAF Medical Corps during the Vietnam Conflict as a paramedic. Following discharge in 1977, the author went on to work in a variety of health-care settings, including hospitals, clinical laboratories, and research facilities. In short, we know something about medicine and medical information systems.

The "least common denominator" principle simply means "what is the least amount of hardware and software resources one can expect to have access to on a desktop computer platform." The NeXT platform, in our minds and experience, has the HIGHEST denominator of any currently available desktop computer. While this is not critically important during the development phase of software, it is ABSOLUTELY critical during the delivery phase. High resolution video and high quality sound can be added to most desktop computers via OPTIONAL hardware and software. Doing so, however, does two things that are perceived as negatives to the end user and the vendor; it raises the cost and complexity of the platform, and lowers the reliability and serviceability of the platform. Common sense tells us that as the number of discrete components in a system increases, so does the incidence of failure. Service and support overhead increase significantly, especially when the components are of "mixed parentage"; i.e., third party hardware boards. For these and other reasons not germaine to this discussion, Toltec chose the NeXT computer as the delivery vehicle for our software technology.

Section One - Challenges

Providing usable voice recognition technology to the end user, in our case medical professionals, presents some significant challenges to success. To understand these challenges we need to know what voice recognition really is: sophisticated signal processing combined with complex algorithms for the recognition task itself. To give the reader an idea of the enormous complexity of the subject, the following paragraph is from an article in the April/May issue of Speech Technology, page 38, by Rudnicky and Stern from the Carnegie Mellon University Speech Research Laboratory regarding the Sphinx project:

"Sphinx uses speech sampled at 16kHz, although recent experiments indicate that nominal 8 kHz sampling, using a mulaw CODEC, can be used with minimal consequences for system accuracy. A 12th-order cepstrum is calculated over a 20-msec window stepped at 10 msec intervals. The cepstral coefficients are transformed to approximate the mel-scale, using a bilinear transform. Three separate vector quantized (VQ) codebooks are generated: the cepstrum itself, a difference cepstrum (calculated from frames 40 msec apart), and a codebook based on energy and differential energy"

If the reader understands the above paragraph, the author suggests that the reader request the CMU Speech Labs' publications for perusal. If the reader got a migraine from the above, don't be concerned; most people get lost rather quickly. This is a typical example of the material written on speech technology.

So as to make this document available to as many people as possible, the author has elected not to embed figures, diagrams, or other graphic elements herein. As such, the author can only describe the Sphinx Recognizer system at CMU. To achieve near real-time performance, voice is presented to a Sun 3/280 system with two special purpose hardware boards. The first board is for signal processing, using a TI TMS 32030 signal processing chip, and the second board is a search accelerator using three Weitek 8032 processors. Both specific details of the Sphinx Recognizer and supporting figures can be found in the above referenced article in Speech Technology.

The author visited the CMU facility and worked with the Sphinx system in April of 1990. To achieve speaker-independent speech recognition on a NeXT computer, the signal is preprocessed on the NeXT and sent via Ethernet to the Sun 3/280 system described above. It works well within certain constraints; specifically, quality and type of the capture instrument (microphone) and ambient noise floor. The author suggests the reader review a white paper entitled "Acoustical Pre-Processing for Automatic Speech Recognition" by Stern and Acero for a detailed discussion of adaptation to changes in microphone and acoustical environment

The primary point is this; the current NeXT hardware in a SINGLE machine; i.e., the 56001 DSP, the 68040 main processor, and the microphone elements in either the monochrome displays or the NeXT Sound Box in the case of color systems, is simply incapable of speaker-independent real-time voice recognition. One of Toltec's contributing development partners is John Holbrook, M.D., FACEP, Vice-President of Medical Affairs at Mercy Hospital in Springfield, Massachusetts. Dr. Holbrook holds degrees in both Medicine and Computer Science and is the primary knowledge engineer for Kurzweil Applied Intelligences' "VoiceMed" product line. VoiceMed is a hardware/software product typically running on Intel-based 386/486 processor platforms for medical dictation purposes. (Toltec owns a Kurzweil VoiceMed system for research and testing purposes.) Dr. Holbrook is also the creator of "ChartChecker", a real-time artificial intelligence text parsing system for medical encounter quality assurance and analysis. Toltec has successfully ported ChartChecker to the NeXT platform and integrated it to our TranScriber medical dictation product. Unlike VoiceMed, TranScriber does NOT even attempt voice recognition; rather we capture voice in a networked environment and present it to human transcriptionists wrapped in a NeXTstep interface. While this may seem "old fashioned" or a step backward in light of products like VoiceMed and Lanier Systems similar "VoiceRad" for radiology dictation, TranScriber is orders of magnitude less in terms of complexity, cost, and end user training requirements.

In discussions with both Dr. Holbrook and various personnel at Kurzweil Applied Intelligence, the specific needs for the NeXT (or any desktop platform for that matter) to achieve the desired result of speaker-independent real-time voice recognition are threefold in terms of hardware: one - a robust DSP chip (the 56001 is not considered such, by the way), two - a main processor capable of 100 MIPS of SUSTAINED performance, not burst performance, and three - a reasonable quality capture instrument; i.e., a good microphone. The Speech Laboratory at CMU typically uses headset-mounted Sennheiser microphones with spurious noise cancellation windscreens installed; the Kurzweil system uses a specially designed Telex hand-held instrument similar to a standard telephone handset. The Kurzweil software has been successfully ported to a Sun machine, so it is a fairly reasonable assumption that their software would "live" on a NeXT. Kurzweil, however, has no current plans to do such a port because their software is optimized to their own proprietary DSP board, which is in fact TI processor based, rather than Motorola processor based. At this point, many of you may be wondering how vendors sell voice systems on Intel platforms, which are nowhere near 100 MIPS in processor performance. There are two answers; one - the above applications are SPECIFIC PURPOSE, rather than general purpose and have optimized vocabulary databases geared to a specific category of end user; i.e., radiologists and, two - none of these systems are true real-time systems. The Kurzweil system, for instance, requires the user to perform a background noise check every single time it is used and further requires the user to "parse" their speech with at least a one quarter second pause between utterances. The end result is a very unnatural speech pattern and cadence that is in some cases, impossible to master. Further, in "noisy" environments like factories and such, these systems fall apart rather dramatically. As the Sphinx software has been on a NeXT since release 0.9/1.0 (of NeXTstep) days, this is probably the best current choice in terms of software technology that would likely be deliverable to a wide audience. The author's impressions and experiences with all of these systems is that the Sphinx system is the best technical choice also as CMU has massive brainpower to apply to the project and virtually zero pressure to "hurry and get it to market". Indeed, companies like DEC, IBM, and Apple have contributed thousands and thousands of dollars to the Speech Consortium at CMU for access to Sphinx. Further, government entities like DARPA and the DoD fund a lot of research at CMU related to speech technology.

Which brings us back to the hardware issues; is the NeXT too "wimpy" to do voice? Quite the contrary, it is an excellent platform. Note that CMU took a "client / server" approach to the problem. (via Ethernet, by the way) Now recall the principle of "least common denominator"; what does every NeXT made come with out of the box? Ethernet, MACH, TCP/IP, NFS and the NeXT ICP (Integrated Channel Processor) firmware for starters. Add to that, tools like Richard Crandall's Zilla (formerly Godzilla) and you can begin to get an idea as to how a NeXT developer or even NeXT themselves could do voice fairly easily on the platform. Ariel makes a 60+ MIP DSP array board, the QuintProcessor, which is available now. They (Ariel) also make the DM-N digital microphone which is capable of 16 kHz or greater capture rates. Finally, NeXT is hardly finished in terms of main processor technology. What do future processors like the MC88110 do for us? The author would not even begin to speculate on what NeXT plans for the future, but experience indicates that whatever they (NeXT) do, it promises to be exciting. So, in summation, the greatest challenge would seem to be correctly assembling currently available hardware and software tools, with some awareness of potential futures, to produce a workable voice recognition system on the NeXT. If this sounds too simple, it is. We have not discussed issues like system training, vocabulary construction, application integration and so on. These issues are also challenging to the developer or manufacturer, but are, happily, foreseeably addressable in the near future.The author will expand on this in Section Three - Opinions.

Section Two - Opportunities

As we have seen from the previous section, there are enormous technical issues in bringing voice recognition to the NeXT. By implication and in fact, it is also very expensive. For instance, the members of the CMU Speech Consortium are required to contribute $100,000 annually to maintain access to the Sphinx technology. Kurzweil estimates it would cost well over $500,000 to port their software technology to the NeXT. Vocabulary database construction is currently very costly due to the large number of samples required to teach a system a single utterance like "hello". It is reasonable to assume that a deliverable voice recognition system for the NeXT could and probably will cost several million dollars to develop to a marketable product.

So why do it? Simply put, we may not have a choice. Toltec has research data on a specific category of user, physicians, which sheds some light on the previous statement. An internal, unpublished study by Toltec asked physicians all over the United States the following question: "What is or would be your preferred method of interaction with a computer system?". Less than 5 percent expressed a preference for a keyboard, slightly over 50 percent wanted a mouse or other pointing device, approximately 20 percent desired a touch screen, and the remaining 20 odd percent expressed voice as their preference. When asked to specifically choose between keyboard, mouse, touch screen, or voice however, over 90 percent of those polled replied voice. Since the author assumes that the majority of people reading this work are computer users, we would pose the same question: if you had a choice, what would it be? Odds are, voice would again dominate the responses. Why is that? Human beings rely on spoken language to communicate with others of our species as well as our pets and even inanimate things like cars, furniture, and so on. Add to that our "experience" of fictional portrayals of human/computer interaction like Star Trek, Star Wars, the current Terminator 2 movie and so on and it becomes clear that we as human beings can easily envision and expect to interact with computer systems vocally at some time in the near future. Here at Toltec we actually refer to that expectation paradigm as the "Star Trek Syndrome".

Consumer expectation itself is not compelling enough to devote resources to a voice recognition project, however. There must be other, more financially tangible reasons for a developer or manufacturer to undertake such a project. Not surprisingly, there are. As most NeXT users know, there is a great deal of "truth", for lack of a better word, in NeXT's philosophy of

"Interpersonal Computing". NeXTstep is far easier to learn and love than DOS or "raw" UNIX. An operating environment like NeXTstep COMBINED with voice recognition and command capabilities would almost certainly "take the market by storm". Market segments that are currently untapped would open up to the vendor of such a system; i.e., the physically challenged segment of our population, for instance. From Toltec's market perspective, for example, consider the following numbers:

- health-care costs in the U.S. account for over 11 percent of the GNP, annually, or approximately 600 billion dollars.
- It is estimated that over 25 cents of every health-care dollar goes for paper administration.
- In 1989, figures show that less than 1 percent of the 600 billion dollars spent on health-care went towards information handling systems.
- Extrapolation of the potential market in HEALTHCARE ALONE is well over 10 billion, annually.

The key to significant market penetration of the health-care market, as well as many others, comes down to user acceptance. The AMA (American Medical Association) has stated that the condition known as "computerphobia" or "cyberphobia" is indeed real and afflicts approximately 80 percent of the general population. Voice capability may very well "cure" this condition in a big way. The author suspects that for a (hopefully) small segment of the population, voice will actually intensify the fear of computers. For the most part, however, voice recognition should break down some barriers to market acceptance, which translates to money, which is of course why anyone makes any product in the first place. So, in summation, we can see there are probably sufficient economic reasons to develop voice recognition systems.

Section Three - Opinions

While the author has probably interlaced the previous sections with some amount of personal opinion, everything contained in this final section should mentally be prefaced with the phrase "In my opinion...".

- Voice recognition will become a reality before the end of this decade.
- The NeXT platform is probably the most viable, but not the only, system to do this on.
- NeXT should devote cycles to a Voice API for a future release of NeXTstep.
- Single machine approaches to the problem are not optimum and probably undoable.
- A client/server approach is better both technically and economically.
- Superior natural language algorithms will be required for success and acceptance.
- Symmetric multiprocessing on the desktop will speed development.
- Any viable solution should be integrated at the OS level, not the application level.
- Vocabulary database construction and manipulation tools for "mere mortals" must exist.
- System training of speech and usage patterns should be machine-intelligent.

The author does not want to expand on any of the above points save the last one, which needs some explanation. By "machine-intelligent", the author means that the pattern recognition algorithms should be closely tied to a continuously optimizing database. Microsoft, for instance, has some code called "Wizard" under development for Windows that supposedly "learns" usage patterns based on system experience with the user and at some point improves productivity and interaction with Windows and Windows applications. Voice systems should have a similar capability. In point of fact, the author has seen and worked with a "machine-intelligent" application on a NeXT recently, so this is certainly a viable technology direction. In closing, the author appreciates the opportunity to present this information. Toltec Human Services would also like to ask that this document and any and all material contained herein not be copied, quoted, or otherwise distributed in any fashion without our express written or electronic permission.•

Paul R. Dietz is the CEO of Toltec Human Services. His can be reached via e-mail at dietz%tsta1@uunet.uu.net

Help Wanted

hAng ON! Needs You!

The response to the first issue of hAng ON! was tremendous. We got some great articles and the newsletter has increased in size from 8 to 13 pages - with less articles by the actual hAng ON! staff.

We could use a lot more submissions from everyone out there, namely:

- More Cartoons. If it has to do with NeXT, put it in EPS format and mail it to us.
- Reviews of new products for the NeXT.
- Editorial style articles on topics dealing with the NeXT.
- Inside information for The Schlepkin.
- Letters to the Editor.
- Beer.•

A Special Thanks To...

Randy Nelson

For a *great* developer camp.

Susan Lopez

For making it *possible* to go to developer camp.

Bob Boeye

For that *great* developer camp cartoon.•

How to Reach Us

General comments and flames: hang_on@cubetech.com

Steve Sarich: steve@cubetech.com

Andrew Loewenstern: andrew@cubetech.com

Mike Barthelemy: mike@cubetech.com•

Who Dunnit?

Editor: Steve Sarich

Assistant Editor: Andrew Loewenstern

Astral Projection Specialist: The Schlep

Staff Reporter: Steve Nasypany

Staff Programmer: Mike Barthelemy

Layout: Andrew Loewenstern

hAng ON! is produced entirely with LetraSet rub-on letters and a Microtek black&white scanner. Just kidding again. This time we did it on a NeXTstation Color and an Atari 1040ST (we only use it as a terminal...) using FrameMaker, Edit, Mail, WordPerfect, and Emacs.•