Schussel's

DOWNSIZING JOURNAL

June 1993

In This Issue:

Report from SOFTWARE WORLD

Part I of II

SOFTWARE WORLD, held annually in Toronto, is the largest computer software-oriented conference and exposition—this year there were over 7,000 attendees. In May, I co-chaired this event with Ed Yourdon of *American Programmer*, and Roger Burlton of SRI. I also managed to put in a few days of conference attendance and so for this article, I am acting as a journalist, reporting highlights from some of the sessions. As it happens, there were so many interesting people and points to talk about, that we have to run this article over two issues. In this issue are the synopses of Will Zachmann,

(continued on page 10)

Operating Systems Wars—The Latest Status

Part II of II

The dominating computing architecture for the remainder of this decade is client/server. This means that there are at least two different types of environments in which operating systems (O/S) will compete for market dominance. Those are: 1) the client or desktop, single user side, and 2) the server multi-user side. The requirements for success and the contenders differ somewhat in these two different markets.

On the desktop/client side there are four principal contenders—Microsoft, IBM, Apple, and the UNIX

(continued on next page)

Operating Systems Wars...

(continued from front page)

community. In the battle for dominant O/S on the server side, it remains the same competition with the exception of Novell replacing Apple.

Microsoft is the clear, current dominant power on the desktop. With the combination of DOS and Windows, Microsoft software is currently running on about 100 million machines. New installation rates are going up, not down, for Microsoft O/Ss. The rousing success of Windows has driven Microsoft's current success, since the simple DOS product wasn't enough to confer industry leadership. Microsoft's apparent victory in its potential fight with the Federal Trade Commission, its continuing mastery of marketing, and its rich plans for a variety of future enhancements to various O/Ss, probably insure that its domination of the desktop will continue for quite a while.

The battle for control of the server is much more complex (and interesting). For starters, at the time this article is being written, Microsoft has no server operating system product. Yes, it does sell the combination of OS/2 and LAN Manager, but there is no desire (on Microsoft's part) to continue selling or supporting this combination; since OS/2 has become IBM's product, support for it will detract from Windows NT. This means that SQL Server for OS/2 or LAN Manager for OS/2 users, for example, would be well advised to plan a migration path to other applications for OS/2 or to a Microsoft O/S for their Microsoft systems applications.

...The integration between Microsoft's Windows NT and the Microsoft/Sybase SQL Server means truly superior performance out of this combination....

In the area of open server O/Ss, Novell and the UNIX community are the dominant players. Novell's NetWare has been the dominant LAN O/S. With the emergence of NetWare 4.0, it has become a serious contender for enterprise computing types of environments.

UNIX has become the standard bearer for high-end, open systems computing. The addition of X-computing support, higher levels of security, the Motif GUI, transaction monitors, non-stop computing, and symmetric multiprocessing support, have brought UNIX to the point where it is being used successfully as an alternative to highly robust proprietary systems like VMS and MVS. The acquisition of AT&T's USL (the organization that builds the UNIX kernel) by Novell is likely to enhance the appeal of UNIX on the server for many companies. Let's talk in more detail about some of these companies' strategies.

Microsoft MS-DOS

This is the cash cow. It was here that the Federal Trade Commission was potentially going to make a dent in Microsoft's marketing machine. At issue are the deals that Microsoft cuts with hardware vendors to pay a DOS royalty on every machine they ship, regardless of whether or not it contains DOS. The hardware vendors buy into this arrangement because it gives them much lower royalty rates. And, of course, once you're already paying for an O/S to be on a computer, that removes much of the incentive for using an alternative O/S.

The primary competition for DOS is Novell's DR DOS, which is technologically one release (or about 18 months) ahead of MS-DOS. However, DR DOS's technical advantages haven't been significant enough to allow Novell to capture more than 5% of market share. As a straight DOS product, DR DOS just isn't going to significantly impact Microsoft's market share. Progress for DR DOS may come through bundling the product in complete network solution packages which may include other Novell products such as UNIX and NetWare.

Version 7 of MS-DOS is likely to be released sometime in 1994. It will be a major revision to the product (as version 5 was and as version 6 was not). Microsoft is now talking about splitting Windows 4.0 from DOS 7.0. In other words, desktop Windows will have its own underlying O/S and won't need a separately licensed DOS. DOS 7.0 will feature 32 bit computing support, the ability to run applications in protected modes, and the ability to run multiple DOS sessions. All of these capabilities will come through DOS's ability to use the enhanced mode of the 386/486 processors.

Microsoft Windows

Shipping at a *claimed rate* of over one million copies per month, Windows 3.1 is the star of the Microsoft show right now. Version 3.1 was a nice incremental improvement over 3.0, but

(continued on next page)

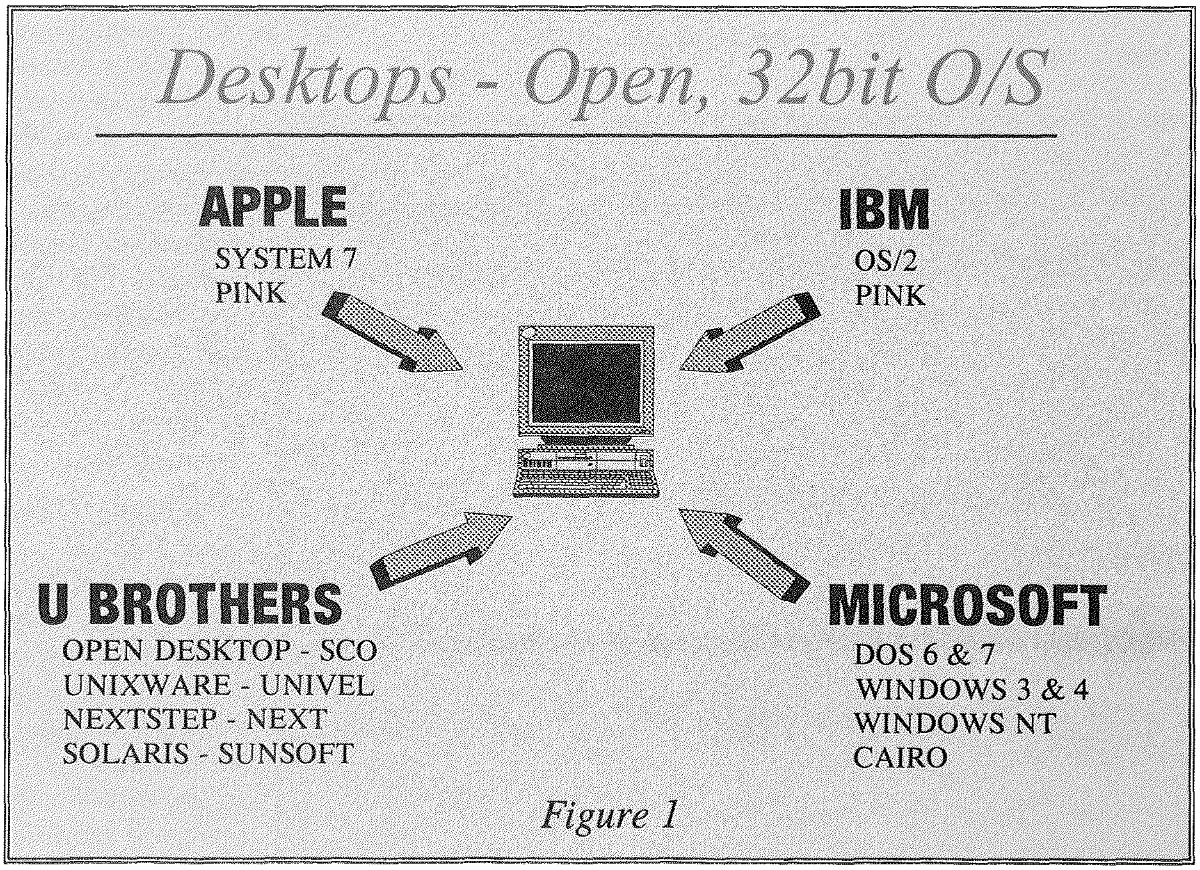


Figure 1

Operating Systems Wars...

(continued from previous page)

didn't represent a new leap in approach. Windows 4.0, due in 1994, seems like it will be another major leap forward. It will forever leave the 286 world as it is a full 32 bit operating environment. Like other real, full-scale O/Ss, it will offer a pre-emptible scheduler along with multi-tasking and multi-threading. Windows for Workgroups functionality will probably be incorporated, and there may no longer be a Word for Windows product.

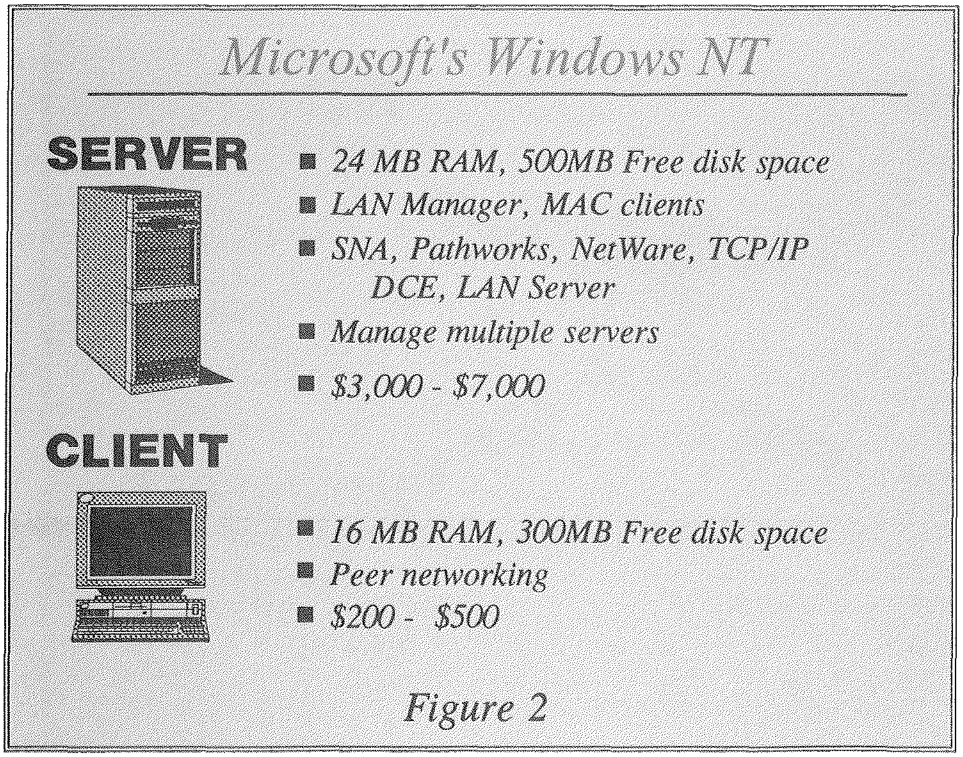
For application builders, Microsoft has committed to making the Windows 4.0 API compatible with Windows NT's. That means that most Windows NT applications will run unaltered without recompilation on Windows 4.0. And, that in turn, means that Windows 4.0 will be the dominant client side computing platform of the mid-1990s.

Microsoft Windows NT

This is the hottest news story of 1993. With an estimated 60,000 (!!) beta users, and generally excellent reviews, it doesn't seem that Windows NT can fail. Microsoft has promised a late May delivery and about the time you read this it should be clear whether that date has held. Many industry analysts are predicting a fall 1993 shipment date. Windows NT is Microsoft's high-end operating system. It will compete with UNIX (mostly) as well as other high-end, general purpose systems like DEC's VMS and IBM's MVS. The built-in support for multi-processing is going to mean that Windows NT will be very rapidly scalable upwards in power and will be comparable or greater in power than the largest mainframe environments.

I am not going to bore you by repeating all of the things that Windows NT will be capable of (protected, pre-emptible, etc.), but I will point out that the integration between Microsoft's Windows NT and the Microsoft/Sybase SQL Server means truly superior performance. Beta sites are already talking about mainframe performance capabilities out of this combination and that's before the System 10 (major) enhancements have been delivered.

Microsoft has better VAR relationships than any other company in computer history (with the possible exception being Novell) and Windows NT will be delivered with more


Microsoft's Windows NT

SERVER

- 24 MB RAM, 500MB Free disk space
- LAN Manager, MAC clients
- SNA, Pathworks, NetWare, TCP/IP DCE, LAN Server
- Manage multiple servers
- $3,000 - $7,000

CLIENT

- 16 MB RAM, 300MB Free disk space
- Peer networking
- $200 - $500

Figure 2

applications than any other new O/S in history. In the next century, I expect business historians to look back on Windows NT's introduction as redefining how to take a market by storm. I expect that there will be a significant number of defections from OS/2, UNIX, and VMS (especially in commercial types of accounts) to Windows NT.

I also believe that Windows NT won't come close to challenging the total number of Windows 3.1 users. It's requirement for 16 MB of RAM and other high-end features means that Windows NT will find a home on power desktops and servers. Microsoft's own Windows 4.0 will take a lot of the market that might otherwise have migrated to Windows NT—it will get the users who realize that the migration to Windows 4.0 simpler and less expensive.

Windows NT will come in two versions, one for the desktop and the other for the server. The difference is that the server version has multi-user communication functions, will include all of the functions now found in LAN Manager, and will have the ability to run directly on top of multi-processor hardware engines.

Naturally, the server version will require a larger machine. With an absolute minimum of 16 MB RAM (more typically 32 MB) and with minimum disk capacity approaching one gigabyte (including some applications), it will be hard to keep a straight face when talking about running Windows NT on a PC! The desktop version will be priced in the (probably low) hundreds, and the server version will be priced in the (mid-single digits, probably) thousands of dollars.

> *...Although much more complex than DOS, OS/2 is a simpler environment to manage than UNIX which is the only other protected environment that's been available for the Intel platform....*

IBM's OS/2

After a painful and slow multi-year start, OS/2 is gaining respect as a PC operating system. Its primary advantage is its stability which is provided by a mainframe-style architecture that supports multiple protected program regions. With a pre-emptible scheduler and support for multi-tasking, OS/2 is really a sophisticated O/S that has good DOS compatibility and runs on PCs.

The happiest users are application developers who have found that the OS/2 environment has good support for new program creation. Its environment, facilities, and selection of tools are all very useful in this way. Although much more complex than DOS, OS/2 is a simpler environment to manage than UNIX which is the only other protected environment that's been available for the Intel platform.

IBM claims that over two million copies of OS/2 version 2 have been shipped. While that's a very respectable number, it's small potatoes when compared to Microsoft's claim of over one million copies of Windows shipping every month. In spite of OS/2's technical quality, it is not likely to emerge as dominant in its market for a variety of reasons:

1. Most importantly, there aren't many native applications available for OS/2. That means that it sells as an environment

(continued on page 16)

CASE is Dead. Long Live CASE!

Jim Davey
Senior Consultant, DCI

As can be seen in various sectors of the information technology (IT) industry, CASE has been caught up in the platform paradigm shift and downsizing movement. The demise of IBM's mainframe Repository Manager product has led a number of IT consultants to declare that "CASE is dead." What is actually true is that CASE tools are leading in the downsizing of application development but, seem to have missed in the downsizing of application deployment. Just one year ago, most of the mainline commercial CASE products: 1) were high-priced, 2) supported traditional structured analysis and design methodologies, 3) were waiting for IBM's AD/Cycle mainframe repository, and 4) generated code for mainframe deployment. In the world of real-time and engineering CASE, most of the products were high-priced UNIX workstation-based tools of little interest to the commercial market.

Today, the CASE industry is more diversified. There are a number of high quality, low-priced CASE tools from companies such as Evergreen CASE, Computer Systems Advisors, Popkin Software & Systems, and Visible Systems. Traditional real-time and engineering CASE vendors, including Cadre Technologies and Interactive Development Environments, are now moving into the "open systems" commercial market. There also exists a number of object-oriented development companies including Protosoft, IntelliCorp, ObjecTime, Object International, and S/Cubed, that see their products as part of the new, broader CASE market. "Brand name" CASE vendors such as Knowledgeware, Texas Instruments, Bachman, LBMS, and other "big six" types are reorienting their products away from mainframe central system development and directing them towards client/server development.

What has been traditionally known as the CASE market is currently splitting into two new major market segments: software engineering and application development. The difference between these two markets is the participants: there are software engineers who design and build computer programs, and application developers who focus on the understanding of subject matter and the development of data models for user interfaces. Application developers are getting away from traditional programming and are now using high-level DBMSs and GUI development tools.

The new breed of software engineers

Today's software engineer builds real-time systems, system software, embedded systems, and data processing applications. They CASE tools they use are based on structured methods and other software engineering principals and concepts. Often, these tools run on UNIX workstations or high-end PCs. The developed systems can be deployed on a variety of platforms ranging from microprocessors, UNIX boxes, super-minis to mainframes.

Software engineers tend to generate programs in C, Ada, or COBOL. Currently on the market are object-oriented software engineering tools with which implementation objects are created and reused as components in the process of building programs. Also available are object-oriented software engineering tools that can be used to create implementation objects which can then be reused as components in the program building process. A part of this market segment consists of products that support a specific development or deployment task such as project management, testing, or performance estimating. Such products tend to come from small companies started by an expert in the field.

Software engineering technology is needed in commercial application development for both large batch support systems and high performance transaction processing applications. In addition, new forms of interactive problem solving applications will be developed using a new application development style.

Application development and client/server

Traditional commercial application development used a "weak" form of software engineering; structured analysis and design helped to organize problems into hierarchical structures that could be transformed into a program control structure. In this model, control was centralized. The end result was that the computer program controlled the user; therefore, the end-users were generally clerks, hired to do inputs at computer terminals.

The new application development style has been heavily influenced by the personal computer (PC) model. While the PC supports personal productivity, the user, now a subject matter specialist, is in control of the computer. This new form of application development—the result of the marriage of traditional central system development and PCs—works well in the client/server model. Application development, in the client/server context, is based on data modeling and GUI development rather than programming. Application logic is associated with data events and implemented as event procedures in the client. These event procedures can be triggered by user interface events or as stored procedures in the server.

New tools to support this development approach, which include PowerBuilder, Object Vision, Visual Basic, and Ellipse, are generally referred to as Windows 4GLs or Windows Application Development Environments. However, vendors of such products do not want to be confused with the traditional CASE tools vendors. But, they are, in fact, competing for much of the same market.

There are also a number of high quality CASE tools, at all price levels, for data modeling and database design. Client/server applications can now be quickly built using CASE tools for

(continued on page 19)

Doing Client/Server Right—The First Time

Part II of II

Richard Finkelstein
Performance Computing, Inc.

This is the second article in a two-part series on client/server systems. In the first article, published in the May 1993 issue of *SDJ*, Finkelstein discussed architectural objectives, the reasons for going to a client/server system, and different hardware specifications for both clients and servers. In this month's article, Finkelstein discusses how to choose between operating systems, and shares his tips for doing client/server right, the first time.

Choosing your operating system

It is within the realm of operating systems that the real controversy exists. I want to put forth a glossary of basic terminology:

...It is within the realm of operating systems that the real controversy exists....

☑ **Protection**: Operating system *protection* means that application memory spaces are *protected* from intrusions by other applications. This facility prevents one application from crashing another application located in the same memory space. Protection is extremely important—it means that you will not experience time-variant bugs or situations in which two applications intrude on each other. You eliminate the situation in which one application can crash an entire system. Because, if such a crash occurs, there will be no way to replicate what happened, and therefore, no way to isolate and fix the problem. Keep in mind that one of these intrusive applications can be the operating system. So even if you only have one application running over the operating system, that application can crash the operating system if it is not running in protected mode.

☑ **Pre-emptive Scheduling**: The concept of *scheduling* was really the original reason for having an operating system; the advent of operating systems came out of the need for a facility which could schedule multiple tasks. *Pre-emptive scheduling* allows the operating system to prioritize and schedule the processing of tasks based upon the mix of tasks in the environment at any given time. Scheduling is important because the system must be in control of the allocation of memory and processing for tasks. If the operating system isn't in control, then the applications are in control. Since an application can not be written to adjust for external factors, it will hold onto resources for as long as it wants, and will release resources when it wants. Such a scenario will cause varying response times. All of this, of course, breeds unpredictability. You can not develop mission critical applications if you can not control the response time and behavior of your environment.

☑ **Virtual paging**: *Paging* allows the operating system to use external disks to extend memory on demand so that programmers, developers, and users do not have to worry about running out of space. *Virtual paging* is necessary when handling a mixture of batch and on-line transaction processing applications. You also need paging to handle the new types of fields for BLOBs, which are filled with two gigabytes or more of data. It is very difficult to put two gigabytes of data in a 16 Mb machine! In order to manage such large amounts of complex data, you really need paging. Paging also makes it much easier to program and develop because you're not worrying about available memory.

How the O/Ss rate

DOS is a very basic operating system which I judge to be stable for client/server work on the client. If you deploy DOS clients, you're going to have a safe environment and fewer problems than in other environments.

Windows, I give the rating of medium maturity, but I don't really think that it's even at a medium level—it is absolutely riddled with problems. In fact, I know of more files that have been corrupted by Windows than by the Michelangelo virus. However, while it is not a very stable platform, it is going to be almost impossible to avoid Windows. Your challenge in designing a client/server environment is to lower your risk. Beware that Windows is especially unstable in networked environments.

OS/2 is actually very mature for both client and server types of work. Unfortunately, there are no applications available to run on OS/2. So, it's a good operating system, but you can't really use it. Of my clients who have used OS/2 on the server and the workstation, their applications usually run very smoothly.

Unix is a very good, mature platform. However, be careful, not all Unix operating systems are the same. Since every Unix is actually a proprietary version, there exists no real interoperability. A good, solid Unix such as SCO Unix, SUN OS, or HP UX will go a long way towards speeding application development.

NetWare can be used on the server, but it has chronic problems that stem from the fact that it is not a protected environment. NetWare was originally developed to manage file and printer sharing. Now, companies want to employ NetWare applications as database servers. Why do they want to do this? Well, because they don't want to introduce another operating system into their environment. They understand, either implicitly or explicitly, that heterogeneity causes problems. However, as far as homogeneity is good, NetWare has problems. I don't know what Novell is saying currently about these issues—at first they said that what was important was that NetWare was running fast. Hey, I can make a Sherman Tank run very fast if I re-

(continued on page 19)

Report from SOFTWARE WORLD...

(continued from front page)

Canopus Research, and his visions for the software industry in the next decade, Mary Loomis, Versant Object Technology, with her explanation of the importance of object-oriented technology, John Tarbox, Caanan Analytics, with his views on Windows NT vs. OS/2, and Tim Lister, Atlantic Systems Guild, on what it takes to be a good software engineer. Next month's issue will cover two sessions, one that was headed by Ed Yourdon who spoke about the silver bullets of software engineering, and the other by Larry DeBoever of DeBoever Architectures who concentrated on the convergence of re-engineering and downsizing.

Will Zachmann—software in the year 2000

Zachmann, a highly visible industry consultant, is probably best know for his fight with Ziff Publishing over editorial freedom. He resigned a highly lucrative PC Week column due to pressure from the publisher to tone down his attacks on Microsoft. Zachmann still hasn't changed his feelings about Microsoft—but more on that later.

Zachmann is now focusing his current research and analysis on software trends for the future decade. During his session, he explained that much of his research is based on reading, talking, and (largely) just thinking about what's most likely to happen. For Zachmann, the phrase *"our research shows"* is a dead give away that an analyst is trying to con you into thinking that a piece of prognostication is based on something more sophisticated than the above mentioned techniques.

Zachmann began his talk by debunking the popular myth that it is applications that sell the hardware platform. He suggested that, in fact, there is no evidence to support such a statement at all. Zachmann argued that what happens in the industry is actually the opposite; a popular platform will attract application writers to produce software, some of which will become very popular.

As proof for his thesis, Zachmann cited the fact that the original IBM PC was already on its way to being a huge market success before Lotus 1-2-3 spreadsheet was introduced. A more recent example of a platform that's success attracted application developers is Microsoft's Windows. As both Windows 1.0 and Windows 2.0 were perceived as failures, the applications just weren't there. Now that Windows 3.0 has been reviewed favorably by the press and user community, the application choice in that environment has become extensive (as well as good). Another example Zachmann used was NeXT, which has superb applications, but hasn't become a dominant force because the underlying platform hasn't sold well. *(Editor's note: The availability of NeXTStep on the Intel 486 platform means an effective 60% price cut for this product. It's figured this way: a 68040 NeXT workstation typically configured sells in the $9,000 range. A license to NeXTStep for the Intel 486 or Pentium sells for $995. A 486/50 MHz PC offers comparable performance to the NeXT workstation and sells for under $2,000. So a $9,000 NeXT machine has come down to $3,000 by moving to the more popular Intel platform. I think NeXT will experience more success in its decision to become a software company.)*

Zachmann's discussion of the importance of choosing the correct platform was a natural transition into a discussion of which platforms will dominate over the next few years. The candidates he considered included Intel's x86 and Pentium, SPARC, MIPS, Alpha, and Power. The conventional wisdom is that Intel, with its 90%-plus share of the world-wide market for 32 bit microprocessors, seems invincible. Zachmann is of the opinion, however, that although Intel will remain the dominant force, its Pentium chip has serious technical problems and is likely to be challenged by one of the RISC chips. Here is how Zachmann rates the others:

☑ The SPARC chip has had a reasonable run but hasn't generated an industry dominant position. It probably will continue to hold a small percent of the total microprocessor market.

☑ MIPS has largely become the Silicon Graphics supplier. The other largest potential user of MIPS is DEC (for the DEC Station line of products), and DEC is clearly going to be pushing its own new Alpha processor. It is highly probable that MIPS will continue to play only a very small overall role in the total market picture.

☑ The Alpha chip, to date, sounds good. If Windows NT becomes a huge success (which Zachmann doesn't believe is going to happen), Alpha will have a serious shot at becoming big time.

☑ Finally, Zachmann believes that the Power Consortium (Apple, IBM, Motorola) has the biggest chance of successfully challenging Intel's dominance in microprocessors. He likes the migration strategy that is in place from IBM's AIX and OS/2 and from all of Apple's products.

An attractive alternative for people who want to upgrade from DOS/Windows is going to be IBM's OS/2 version 2.1. This product will be generally distributed over the next few months, and will run DOS and Windows 3.1 applications seamlessly. Actually, OS/2 is more compatible with Windows applications than Windows NT will be, principally because it is far more compatible with DOS. You can boot several different specialized DOS regions in an OS/2 machine which is out of the question with Windows NT because DOS requires an ability to directly manipulate the hardware. Windows NT doesn't allow such direct manipulation because of security reasons.

One particularly interesting prediction that Zachmann made was that Microsoft is going to run into serious trouble within the next year or so. His argument is that Windows NT is going to be late for effective usefulness. The earliest conceivable date for a usable, stable Windows NT is this fall. Even then, the sheer size of machine that it will require (16 MB RAM and 300 MB free disk space) for desktop usage means that it will sell in relatively small quantities until there is an established hardware base of machines capable of running Windows NT.

Zachmann summarized the various risk factors Microsoft will have over the next several months. The first is his Windows NT scenario mentioned above. The second risk factor is the seriously negative image that DOS 6 and its Doublespace compression have generated. Sales of this cash cow are likely to be slower than Microsoft expected. (But what's the alternative? DOS 5?) Next is the apparent final resolution of the Apple lawsuit which may happen over the next few months. A loss on even a single infringing element could cost Microsoft a

(continued on next page)

Report from SOFTWARE WORLD...

(continued from previous page)

major financial settlement. *(Editor's Note: Since Will Zachmann spoke at SOFTWARE WORLD, most of the remaining, large infringement charges have been thrown out by the court. Now only some small issues remain to be resolved.)* And finally, the deadlocked Federal Trade Commission's investigation of Microsoft's pricing structures may be reinstated.

No one needs to be sending Bill Gates any donation checks just yet, however. Zachmann thinks that a good and early (no later than October 1993) beta of Windows 4.0 (code named Chicago) with program protection and multitasking features will generate a lot of positive momentum for Microsoft.

Mary Loomis—the world of objects

The next session I attended was that of Dr. Mary Loomis, head of research for Versant Object Technology which is one of the leading companies in the OODBMS (object-oriented DBMS) market. Loomis started her talk by mentioning that the term "object" has many meanings. It is used to describe various approaches in modeling, programming, database design and other facets of systems building. As opposed to the relational DBMS approach, then, it's important to define what aspect of object-oriented you're talking about in each situation.

From surveys she has seen, Loomis stated that the primary reasons for adopting object-oriented technologies are:

- better/higher quality software, which Loomis feel is achievable for many organizations.
- better/higher productivity (however metrics have not proven this to be true and you should be skeptical about the claims here).
- achieving reusability.
- the ability to solve tougher problems—Loomis said that "Yes, you can achieve more complex systems with higher performance than is otherwise possible."

...Relational DBMSs are good for simple data that will fit in the rectangular relational model.... In contrast, OODBMS are best for complex data—voice, image, complex interrelationships....

Relational DBMSs are good for simple data that will fit in the rectangular relational model. RDBMS also have good ad hoc query and search capabilities. In contrast, OODBMS are best for complex data—voice, image, complex interrelationships. In these situations, you're likely to get better performance from an OODBMS since relationships are built directly into the data model.

Many of Loomis's current customers say the reasons why they use OODBMS include :

1. They are committed to object-oriented programming systems (OOPS) which means it will be easier to integrate an OODBMS rather than a RDBMS. Defining data types and classes in an OOPS maps directly into OODBMS. (Loomis mentioned that OOPS such as Smalltalk or C++ are more popular at this time than OODBMS.)
2. They need OODBMS performance because they have complex data models for which a relational system would require too many JOINs to support the application.
3. The customer has an inherently distributed application. Possibly there

are long transactions. Loomis gave an example of a long transaction being an object that is checked out by an employee to be worked on at a PC for a week. This object would have to be locked for this week and, therefore, needs a non-traditional locking model (supplied by the OODBMS vendors). Both of these characteristics strongly indicate a potential OODBMS application.

Other issues of interest that Loomis mentioned:

- *Environments*—Object-oriented technology principally exists in the UNIX world and soon, Windows NT. There is almost no support in the mainframe MVS or DOS/Windows world.
- One of the issues here is that OODBMS only support C, C++ and Smalltalk. It's awkward to integrate an OODBMS with typical languages like COBOL or FORTRAN.
- Loomis mentioned that the analyst community has predicted growth in the OODBMS market of about 100% per year over the first half of the 1990s. She thinks that this growth has been overstated and that so far the field is growing at only about half that rate.
- Security and tools are much more mature for RDBMS than OODBMS.
- Customers almost never acquire an OODBMS with the intent of replacing an RDBMS. It is always a dual approach, each product being used where it's most appropriate. Gateways that allow connections between OODB-MS and RDBMS are just now becoming available, but there is more need than there are products available at this time.
- Coexisting/interfacing object-oriented and relational technologies can be done in two fashions:

1. The first is putting SQL programs on top of the OODBMS. Very few people want to run relational programs straight on top of object-oriented files or databases. The principal advantage of this approach is that it allows companies to leverage the knowledge that is invested in their (already) trained people. A complicating factor here is that relationships in the relational approach are value-based, while in object-oriented approaches the relationships are hard-wired into the database design. ANSI SQL will, in the future, be extended to acquire some of the functionality that object-oriented systems provide.
2. The co-existence strategy involves putting a OOPS on top of a relational DBMS. Setting a table equal to a class is not the usual mapping, as it more normal that a class be set equal to a collection of tables. While normalization is a well accepted design methodology for relational, there is no comparable database design approach for object-oriented.

...Customers almost never acquire an OODBMS with the intent of replacing an RDBMS. It is always a dual approach, each product being used where it's most appropriate....

John Tarbox on Windows NT vs. OS/2

With some form of Windows NT introduction expected at this spring's Comdex, a session on how Windows NT and OS/2

(continued on next page)

Report from SOFTWARE WORLD...

(continued from previous page)

compare was of particular interest to SOFTWARE WORLD attendees. John Tarbox of Caanan Analytics started his session by stating that it wasn't fair to compare a real product (OS/2) to a vapor product (Windows NT). Nonetheless, the comparison was of interest to the standing room only audience and so Tarbox spoke about his experiences using OS/2 version 2.1 and various Windows NT betas.

In terms of similarities, both products have a strong DOS and Windows heritage. In fact, at the technical and syntax levels, there are more similarities than differences between the two products. Windows NT is a more modern system—at one point it was called OS/2 version 3— and is built on a micro-kernel architecture which makes the software more portable than OS/2.

In overall networking capability, the products are approximately equal in total functionality. However, the methods and products that are used to deliver the functionalities are different. The two products will have no trouble operating together across a LAN. Windows NT now has remote procedure call support that is POSIX compliant. OS/2 has no comparable feature, but IBM is now committed to the delivery of remote procedure calls in the future (no dates available).

For DOS compatibility, OS/2 is by far the better choice. That's because it allows you to boot several DOS image files and allows you to run specialized DOS applications with any associated drivers. However, for this same reason, Windows NT offers better security. A DOS application under OS/2 can directly manipulate hardware. Security is, therefore, very problematic. Windows NT will be delivered with C2 levels of security which is secure enough for almost all shared business types of applications.

Windows NT's file system has commit logic built in which will make developing transaction processing applications easier. With other, older O/Ss (all including OS/2, MVS, etc.), commit logic has to be implemented in applications such as database management systems or transaction processing monitors such as CICS.

Tarbox concluded that anyone making a choice between the two O/S for platforms should (probably most importantly) consider non-technical issues in making a choice. No one knows Windows like Microsoft and no one knows mainframes like IBM—which is more important to you? Which company can better support your site?

Tim Lister—what it takes to be a good software engineer

Tim Lister, a principal of the Atlantic Systems Guild, is one of the founding members of the (unofficial) structured systems old boys network. His lectures are always packed with fundamental and simple insights which he delivers in his enormously entertaining manner.

Lister started his session by noting that presentations are becoming high-tech. *(Editor's note: For example, this editor now uses a 486 color notebook running multiple screen shows and database applications under Windows for his presentations. At DATABASE WORLD in June (Boston) this will be extended to include a complete client/server EDA/SQL network demo.)* As a reaction, Lister used a thick crayon to draw the ten or so foils he used in his keynote. And, as usual, his presentation was second to none.

In the process of software development, Lister said that there are four basic issues:

Scale—This is the first real problem. When you are developing software with a group of three to six programmer/analysts, you can use coffee and donut management techniques. This is where you get together over coffee and discuss what to do that day. When the scope of the project is bigger and you have 240 programmers, then this management technique breaks down.

Speed—This is the second issue, and of course, it's intimately tied in with the issue of scale. The most economical way to develop software is always the way that employs the fewest people. Where the problem arises is when the scope of project is such that development with only a handful of people will take too long. At this time, a decision may be to add more people to the project. In the funny world of software development, however, it may be that adding more people will lengthen, not shorten, the development time.

Practice—Lister commented that from informal surveys of software tools, he found that all tools (yes, even the best) usually fail in the majority of sites installed. And the difference in failure or success rates between the best or worst CASE or software tools wasn't consistent or significant. In other words there must be something else at work in determining which sites succeed and which fail. Lister called this factor *practice*. By this he meant that there are stupid people, but no stupid tools. In this way, tools are analogous to loyal assistants. They don't make any decisions, they just implement your wishes. So, regardless of what tool you use, there is still the need to apply the very best people possible. And, Lister concluded, frequent and serious practice of any approach is better than a dilettante's approach to advanced engineering tools.

...In terms of similarities, both OS/2 and Windows NT have a strong DOS and Windows heritage. In fact, at the technical and syntax level, there are more similarities than differences between the two products....

Beliefs—Commenting on work that has been done by the Software Engineering Institute (SEI), Lister discussed the difference between what management says it's doing, and what it actually does. The SEI principle of determining management's true beliefs is to look only at projects under duress and watch what management does. SEI believes that when things are going well, an organization's true beliefs are hidden. When the pressure is really on—that's when you find out what people really think.

Lister's concluding comments were that *"software development is hard"* and that the practice of any consistent approach is the best way to achieve high levels of productivity.

Operating Systems Wars...

(continued from page 5)

for running the latest applications from DOS and Windows. That's an important disadvantage since supporting both OS/2 and Windows for Windows applications is obviously tougher than running DOS/Windows.

2. While it runs multiple DOS and Windows application regions, it doesn't seem to offer any advantage for those applications (except protection, of course). You can count on these applications running slower (although sometimes it's only slightly).
3. Installing applications is tougher in OS/2 and the availability and management of device drivers (network, printer, video, etc.) is much tougher than in the Microsoft environment. Your support costs will be much higher for OS/2 than for DOS/Windows. That is a cost that must be played against the benefits it provides.

OS/2 has suffered from tough competition at the hands of Microsoft Windows. And that's competition from something that only superficially looks high-end, but is really a low-end product. Over the next year, Microsoft will roll out Windows NT which will do everything that OS/2 does now, and then some (multi-processing, multiple platforms, fast database servers). There are some 60,000 beta sites for Windows NT and I have met a large number of OS/2 users who plan to switch to Windows NT when it ships.

> *...With Microsoft's continuing improvements in the quality and diversity of the Windows environment, I don't believe that Apple will ever be able to significantly increase their 10% share of the desktop market....*

More ominously, I think that Windows 4.0 will be the really tough competition for OS/2. Windows NT is going to require a larger machine than OS/2. At this time, Gates is talking about a minimum 16 MB RAM, even for the Intel version. That size requirement is going to limit Windows NT's use to servers and real power desktops. Windows 4.0 will offer essentially all of OS/2's advantages plus thousands of the most popular applications running in a native mode.

Apple System 7

Apple deserves the credit for popularizing the graphical computing idiom. It's Macintosh (technically and marketing-wise) was one of the most outstanding accomplishments of the 1980s. Now however, it has become increasing difficult to differentiate Apple applications from Windows applications—even in the desktop publishing field.

The Apple environment is still closed without clones. That translates into a price premium for Apple products. And, even though Apple has done at least as good a job as IBM in bringing interesting new products to market (e.g. Powerbooks), it still doesn't match up to the product selection that is available from the combination of IBM, Compaq, Dell, Toshiba, Hewlett Packard, etc., in the Intel world.

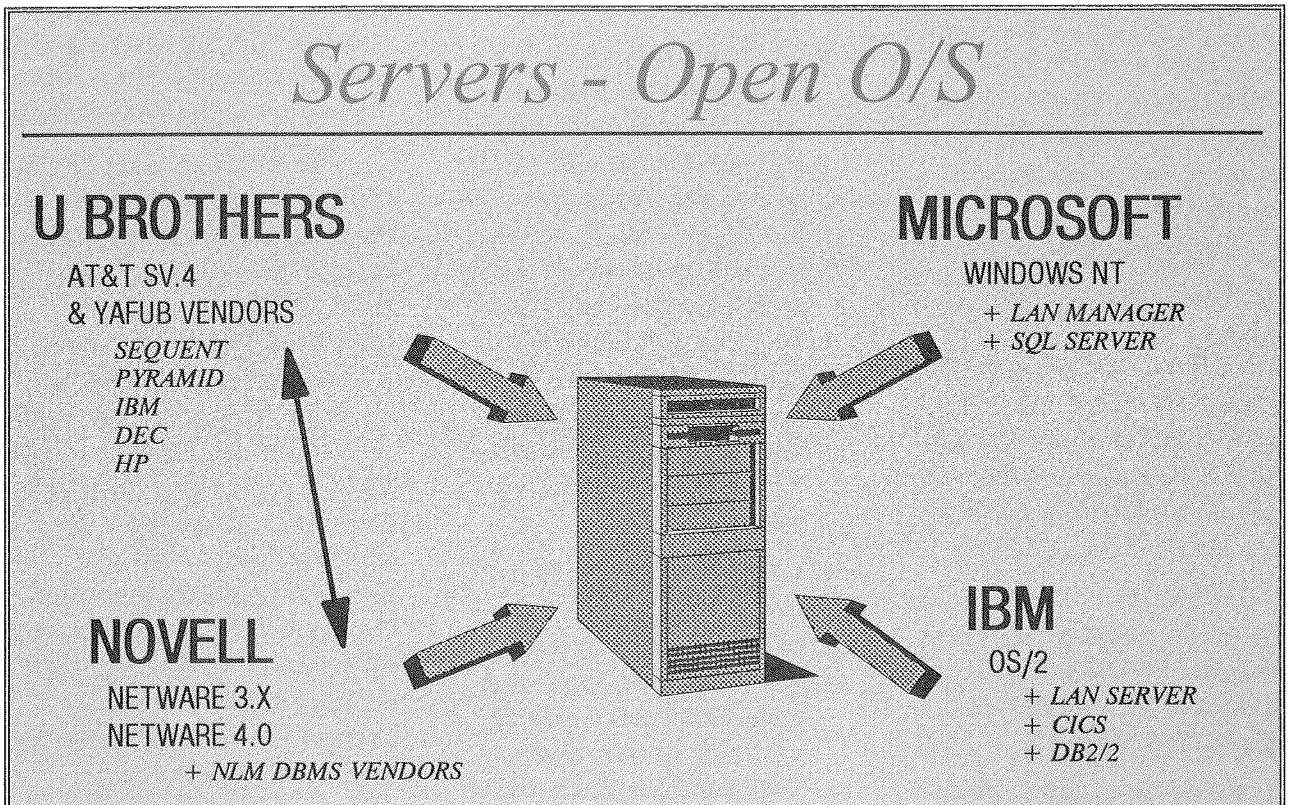
Another Apple problem has been its reliance on Motorola 68000 microprocessors. The much larger volume (and clone competition from AMD, Cyrix, etc.) of the Intel x86 market has driven the price, performance, and application selection on the Intel platform to the top. The Motorola 68000 isn't likely to be able to keep up the pace. Apple has recognized this and by the end of this year, we should see Macintoshes with the Power chip that is being developed by Motorola as a follow up to IBM's RS/6000 microprocessor.

For pure quality and consistency of desktop applications, Apple is still the leader. Even with its IBM partnership, however, Apple is still not considered as a viable choice by most corporate application developers that I know. With Microsoft's continuing improvements in the quality and diversity of the Windows environment, I don't believe that Apple will ever be able to significantly increase their 10% share of the desktop market. Nor, however, do I expect the company to lose that share.

UNIX

The use of UNIX on desktops can be mostly found in engineering departments. UNIX is really a minicomputer O/S; the physical and time requirements of supporting it on the desktop have severely limited its numbers (as compared with DOS and Windows). Numbers of

(continued on next page)



Servers - Open O/S

U BROTHERS

AT&T SV.4
& YAFUB VENDORS

SEQUENT
PYRAMID
IBM
DEC
HP

MICROSOFT

WINDOWS NT

+ LAN MANAGER
+ SQL SERVER

NOVELL

NETWARE 3.X
NETWARE 4.0

+ NLM DBMS VENDORS

IBM

OS/2

+ LAN SERVER
+ CICS
+ DB2/2

Figure 3

Operating Systems Wars...

(continued from previous page)

UNIX sites are measured in 10's or possibly 100's of thousands, not 10's of millions. Sun's Solaris and Univel's UnixWare are new attempts by Sun and Novell, respectively, to attack the SCO lock on desktop UNIX.

The NeXTStep environment is also a desktop, not server, UNIX implementation, but it is really optimized for application development. Now that it's about to be delivered on the Intel microprocessor platform, I can envision it being used for developing tough applications that are client and peer to peer oriented. NeXT is too small a company to develop into an industry leader in the O/S wars, though.

Where UNIX is currently dominant is in the area of server and multiuser larger scale computing where the customer wants to be open systems-based. UNIX has a very wide variety of databases and tools such as 4GLs and CASE.

UNIX has had a number of problems in penetrating the mass market to the same extent as DOS and Windows. For example:

1. There are many implementations of UNIX. Those implementations vary depending on whether you're running desktop or server styles of UNIX. Applications must be ported and then recompiled to move between the different versions. UNIX applications can't, therefore, sell on a shrink-wrap basis as DOS and Windows applications do.
2. UNIX is a complex O/S for anyone used to the DOS environment. Its application programming language, C, is relatively tough—UNIX commands make DOS look like a · 4GL. A complete set of UNIX documentation can occupy about 10 feet of shelf space.
3. GUI's such as Open Look and Motif can hide part of the complexity of UNIX, but most users will need to understand some of the UNIX commands to take advantage of the O/S. The split between Open Look and Motif has complicated this problem, although it now looks like the entire UNIX community is lining up behind Motif.
4. UNIX hasn't been in the major leagues of the distribution game. Companies including Microsoft and Novell have refined methods of distributing their products that employ multiple channels. There is no real retail distribution channel for UNIX. The sophisticated VAR arrangements, for example, that Novell uses to push NetWare, are missing in the UNIX world. Now that Novell has purchased USL, the principal builder of the UNIX kernel technology, I expect that Novell's distribution expertise will be used to sell UNIX.
5. The culture of PCs has been very different from UNIX. And, to no surprise, UNIX/PC integration is modest. Here again, the Novell acquisition of USL means that we can expect top notch NetWare/TCP/IP integration. PC networking and UNIX networking will come together over the next couple of years.

The UNIX community is coming together because of the advent of Windows NT. Windows NT, as a modern, open, client/server O/S with the familiar Windows GUI is bound to primarily affect UNIX as the high-end O/S standard. Now, finally, all of the various players in the UNIX community understand that. UNIX is old—that's good, it's tested. A united UNIX community with Novell at its head and with tight NetWare integration is a worthy competitor for Microsoft. Stay tuned, it's only going to get more interesting. GS

Doing Client/Server Right...

(continued from page 9)

move all of its armor, but I'm not quite sure if that is any solution.

VMS is very stable; I really love the environment. The problem here is that VMS is much too expensive. I've gone through a cost justification process for VMS on client/server, and it is almost impossible to convince users to spend the money needed for VAXs. If you can spend the money because good administration and reliability are important issues within your company, then this is a great platform.

Concluding hints to remember

Some final, concluding recommendations to remember when setting out to design a client/server system include:

- ☑ Limit heterogeneity as much as is possible.
- ☑ Try to choose a combination of operating systems for the client and server that were designed to work together and with the network operating system. This may not be possible today for a variety of reasons. In keeping, choose similar hardware systems for both the client and server ends.
- ☑ Look for a single source for hardware and operating systems.
- ☑ Check out the quality and the support cost for both hardware and software.
- ☑ Beware of the *real* costs! Always remember to account for the training of existing personnel and the hiring of new personnel.
- ☑ With hardware, remember that consistency in machines is important. RF

Editor's note: Richard Finkelstein is the top world-wide authority on the specifics of client/server DBMS systems and 4GLs. His expertise is from actual experiences with real world products—not theory. He is consistently one of the most popular presenters at DATABASE WORLD. *His extensive knowledge in implementing client/server systems makes him widely sought out for consulting.*

This article was adapted from a session taught by Mr. Finkelstein at DATABASE WORLD, December 8-10, 1992. For more information on these topics or others taught by Mr. Finkelstein, he is reachable at Performance Computing at (312) 549-4824.

CASE is Dead!

(continued from page 7)

data modeling, and database design and generation, and by using windows development tools to create the GUI and client application logic. CASE vendors are also getting into the client/server application development tool market. They are doing this by building, buying, or forming alliances with GUI development vendors.

Many prominent consultants in the IT field have been saying that "CASE is dead!" Over the next few years, CASE may take on different forms, and CASE as it has been commonly thought of may disappear. But, automated software engineering and automated application development tools will thrive and prosper. JD

Jim Davey is a Senior Consultant for DCI and is an expert in the fields of CASE, DBMS, 4GLs, and structured system analysis. Davey lectures extensively at both public and in-house seminars and is reachable at (508) 470-3870.