Schussel's
DOWNSIZING JOURNAL
October 1993

In This Issue:

Technologies for Business Re-engineering
Part II of II

Contributing Editor Herbert Edelstein discusses the new technologies behind the downsizing and business re-engineering movements 1

What a Fabulous Two Years!

*Editor Schussel summarizes many of the trends that **SDJ** has covered over the past two years, how far we've come in this industry in the 1990s, and where the future of computing is likely heading* 1

Upcoming Downsizing Events 16

Technologies for Business Re-engineering

Herbert Edelstein
Euclid Associates
Part II of II

Pen Computing—Both hardware and software vendors recognize that this technology is not so much for the present time, but for the future. Just this spring, IBM announced a new pen-based computer that is going to replace their older model, and what they did to stir interest was previously unheard of in the hardware world. IBM announced that everyone who owned the old version could receive a free upgrade to the new pen-based machine. Knowing what a charitable organization IBM is, this tells

(continued on page 12)

What a Fabulous Two Years!

How far we've come in just two years! In 1991 when ***Schussel's Downsizing Journal*** started publishing, downsizing was a clear trend for the future. The distance we've come since then, however, is an indication of this field's dynamic nature. For example, just consider the following amazing things that have happened in the last two years:

- ☑ IBM's stock price as been cut in half, and for a short while, Microsoft, a $3 billion corporation, had a greater market capitalization than IBM. IBM brought in an out-

(continued on next page)

Fabulous Two Years...

(continued from front page)

sider as the new CEO—an individual with experience in marketing cigarettes and food.

- ☑ The dollar sales volume and number of mainframe computers is on a downward trend for the foreseeable future.
- ☑ All of the leading dBASE/Xbase companies, Ashton-Tate, Nantucket, and Wordtech, have been acquired and have since disappeared.
- ☑ Apple Computer has just announced a $200 million loss for its last quarter, and has a plan to lay off 2,500 people.
- ☑ AT&T, the nurturer and marketer of UNIX, has sold its UNIX operation to Novell, the leading PC networking company.

Well, you get the picture. It's been a pretty amazing two years. While the downturn in mainframe sales could have been (and was) forecast, the appointment of someone like Lou Gerstner as IBM's new president certainly has to rank as a first class surprise. For some fun, and some reflection on how far we've come, following are synopses of some other amazing events from the past few years, and the likely consequences.

The Intel 486 chip has become an industry standard by being cheap, plentiful, and fast.

In 1991, the 386 and 386SX chips were the standard for office automation. Very few DOS applications, even databases, could challenge the processing power that this class of chip provided. Of course, I remember back to the 1980s when the 386 was labeled "a mini-computer within a PC" and we all strained to imagine how to use all of that power.

Well, GUIs, Windows, and Workplace Shell came along and, all of a sudden, the 386 was a dog. Without a question, quick, clean execution of Windows applications requires a 486 processor. Word for Windows can be slow and dogged when processing certain jobs on a 386. Lotus Freelance for Windows is almost painful to use on the older chip.

Having lost its exclusive franchise on the 386 to competition from AMD, Cyrix, and others, Intel pushed pricing down on the 486 faster than expected. This has resulted in a $1,500 desktop computer that has proven to be a dynamite Windows machine. Windows (and client/server computing) just wouldn't have been as big of a success as it is if the pricing model for the client machine was more in line with the likes of Macintoshes rather than cheap PCs.

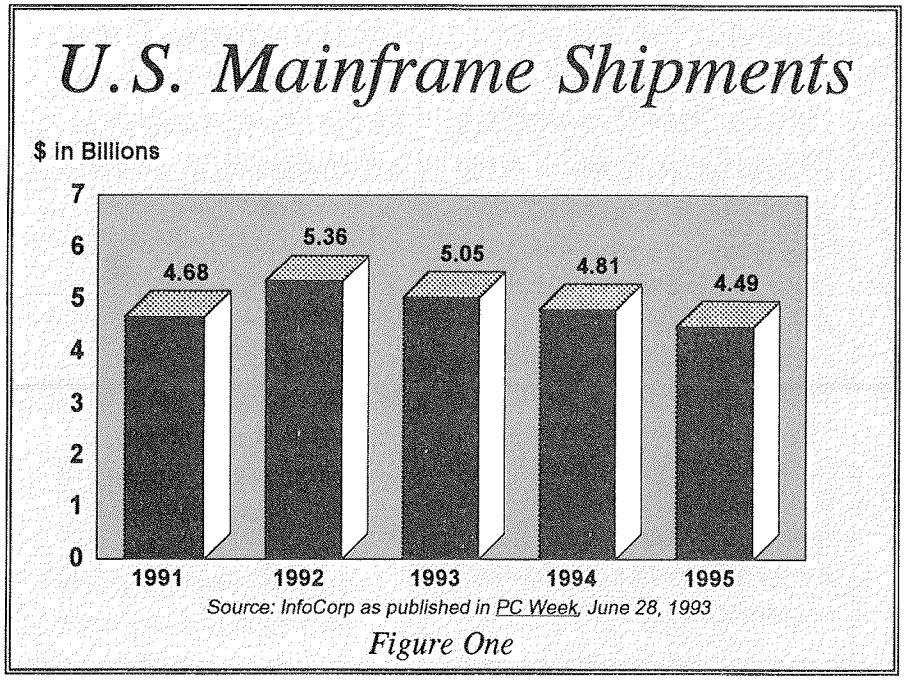


Figure One

I've recently talked to NEXTSTEP users who were absolutely sold on the power of the NEXTSTEP environment, but couldn't imagine installing hundreds or thousands of NeXT workstations at the $10,000 price point for the machine and its software. NEXTSTEP on the 486 benchmarks faster than the NeXT cube. With a large color monitor, sound, and all of the goodies—including the $995 NEXTSTEP software licensing fee—a 486 NEXTSTEP workstation prices out at $3,000. That's a 70% price difference!

Windows has emerged as the new GUI standard.

I would argue that the emergence of Windows as the GUI standard is the single most important happening in the downsizing phenomenon of the past two years. It may be easy to forget, but before Windows 3.0 was released, there was a widespread disinterest in GUIs (exception for with the Macintosh crowd). Sun was pushing Open Look, but that was only on a UNIX platform, and UNIX required expensive workstations. UNIX on the desktop hasn't been a real possibility for most business users in the recent past.

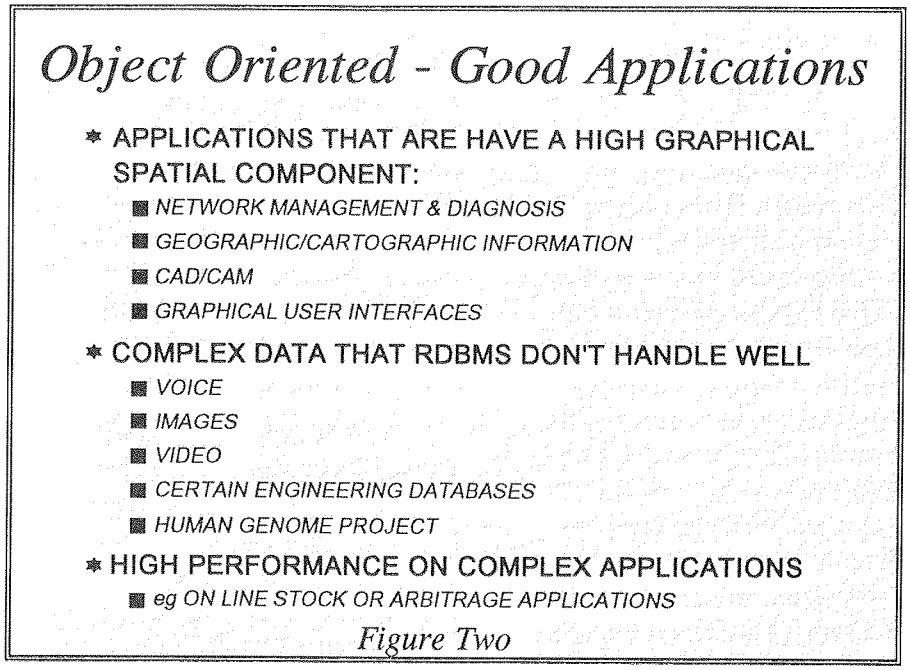


Figure Two

The combination of OS/2 and Presentation Manager was expensive, and, in addition, lacked printer drivers and had no native applications.

The Macintosh was a real possibility and Apple was successful in convincing millions of people of its virtues. The big drawback to a Macintosh has been it's proprietary nature. In the past, Apple has sued anyone who even threatened to bring out a Macintosh clone. That proprietary approach has saved Apple some sales over the last two years, but I would argue that it cost Apple the kingdom that could have been theirs had they built more of a following than the 12% market share they have now.

What Windows 3.0 represented was a way for the normal DOS shop to easily, and somewhat inexpensively, move into the era of graphical computing. Before Windows 3.0, there was no obvious operating system waiting to inherit DOS's 50 million users. After Windows 3.0, it was immediately clear that Windows would become the standard desktop metaphor.

A new generation of application building tools and Windows 4GLs is starting to make a serious impact on the building of new applications.

Client/server database approaches, especially Sybase, have been around since 1987. Until now, however, the field has only grown modestly because there was no standard

(continued on next page)

Fabulous Two Years...

(continued from previous page)

client-side environment. This meant that companies wanting client/server computing could buy something like a Sybase database engine which would offer high-level programming support on the server side, but they had to write their application in C on the client side. Writing applications that generate their own video in C is hard. The difference between regular (character) programming and Windows programming is the difference between straight-line programming and event-driven programming. Except for TSRs, the program drives the user. In Windows, it's the other way around.

Because events can be invoked from all parts of the screen (menu, toolbar, dialog box, etc.), a Windows programmer has to anticipate all potential actions of a user and in all potential orders. And, other programs may also be running and interacting with this program. As a result, it is clear that developing such programs in C should be avoided, unless there is no other choice available.

Once Windows became an accepted standard, then the tools makers focused on that environment for building "Windows 4GLs." We are starting to see the result of this in 1993.

This is the beginning of an entirely new generation of FoxPro and Paradox for Windows applications. It is relatively easy to develop the client side of an application with these new tool sets. And, that is what is really allowing the client/server computing explosion that we're seeing now.

Object Oriented programming has been recognized as an essential ingredient of the migration to a client/server world.

Object orientation, a structure which allows for the efficient combination of data and process, and creates effective mechanisms for maintenance through an inheritance process, has been gaining momentum. Object oriented languages such as Smalltalk and C++ have been around for years. It's only been over the last two years, however, that objects have gained a major following.

Certain types of applications (see *Figure Two*) are know to be particularly amenable to solution by object oriented systems techniques. As the popularity of Windows and GUI programming has spread over the last two years, the real need for object oriented programming has become more visible to interface developers. When the requirement is to understand hundreds of different potential mouse movements or keyboard inputs within the context of several different—but simultaneously running—programs, there is no other current technology that works as well.

The object revolution is going to work its way

Benefits of Object Development

- Programming by component assembly
- Formalism for interoperability & modularity
- Stronger type checking
- Enhances reusability
- Systems can closely model real world
- Simpler maintenance
- Can handle complex or unusual data types
- Solves certain tough performance problems

Figure Three

through the tools industry first. Windows 4GLs and CASE products are being recast on top of object approaches. IBM's AD/Cycle and MVS Repository Manager technologies were failures, significantly because of performance problems that arose from dependency on SQL, instead of object oriented database approaches. The dBASE/Xbase industry's products (dBASE, FoxPro, Clipper) are in the process of being completely redeveloped. New versions of these products will use embedded object technology and will support object oriented development features for users.

By the end of 1994, a reasonably wide selection of object development tools will be on the market. So the mid-1990s, then, will witness the retraining of end-users and corporate developers in an object world. As a side note, it should be mentioned that this coming object revolution doesn't signal the end of SQL or relational systems. Exactly how object oriented approaches and relational systems will interact isn't yet completely understood.

Client/server is now recognized as the best architecture for distributed computing and moving into the new generation.

Client/server computing and the new generation of GUIs on the desktop are intimately involved. Stand alone PCs, after all, aren't really able to solve corporate computing problems. What client/server allows is the centralized control of data and its integration with desktop computers—the clients. Time sharing with dumb terminals doesn't allow for the desirable distribution of processing to where it makes most sense.

One alternative architecture that is receiving some attention is the X architecture, originally developed at the Massachusetts Institute of Technology. The X architecture allows time sharing off of servers with supported GUIs at the desktop. X computing is intimately identified with the UNIX world, and isn't likely to widely populate business computing. Most often, commercial sites are adopting the client/server approach by using computers such as PCs (vs. dumb terminals) at the client sites.

A client/server architecture is shown in *Figure Five*. Notice that it allows for a split of processing between the client and the server. Typically computing that is oriented to the database is handled on the server, while process-oriented functions are largely handled on the client. Some

(continued on next page)

Advantages of Client/Server Computing

- **Windows environment is consistent across applications:**
  - Easier training for users
  - Users with more applications
  - Developer API has built in functions for coding interface
  - Interoperability with desktop applications
  - Develop applications with more junior people
- **Downsizing's Advantages:**
  - Cost savings
  - Industrial strength/robustness/control
- **Greater choice & flexibility:**
  - Multiple tools & servers - both RISC & CISC
  - Modular upgrades, scalability

Figure Four

Fabulous Two Years...

(continued from previous page)

tools provide for an automated split of this function, but all serious development tools allow the developer to create code to run wherever it is most efficient.

The advantages of client/server computing are listed in *Figure Four*. No other computing architecture has ever offered more flexibility than client/server. Whether you use a relational or object DBMS, and regardless of what combination of toolsets you choose, client/server has the flexibility to be configured to any desire.

It's very likely that the remainder of this decade will find most companies rebuilding their computer systems in conformance with one or another version of client/server computing.

Microsoft has emerged as a general computer company competing with the likes of IBM, Sun, and Apple.

Microsoft has intelligently used the success it's had in DOS and Windows to build (or attempt to build) new business franchises. It has tried (unsuccessfully so far) to enter Novell's networking turf with LAN Manager. That effort failed, but Microsoft is trying again by embedding LAN Manager into Windows NT. I believe that Windows NT is very likely to have significantly more networking popularity with the embedded LAN Manager than LAN Manager did by itself.

Microsoft paid about $200 million (in stock with a 60/1 P/E!) to acquire Fox Software and its FoxPro product line. That was a very wise investment since it coincided with Borland's incredible slippage in delivering a dBASE for Windows product (will we see it in mid-1994?). Microsoft is out on the street now with FoxPro for Windows (an interim Windows face on a DOS product) and is eating Borland's lunch. In the Xbase sessions at DCI's DATABASE WORLD this past June, there were more Fox developers than dBASE users in attendance.

Again, with it's high stock price, Microsoft has gone out into the industry and bought up the best people. People such as DEC's Dave Cutler who now heads the development of Windows NT. When stock options can make an employee a millionaire in two years, a modest salary is adequate to get you to move to Redmond!

Enterprise computing is probably the most visible new business that Microsoft is attempting to enter. Windows NT is the mechanism that Microsoft is using as it attacks IBM, DEC, and Hewlett Packard's corporate kingdoms. Windows NT is a micro kernel-based client/server operating system designed to operate

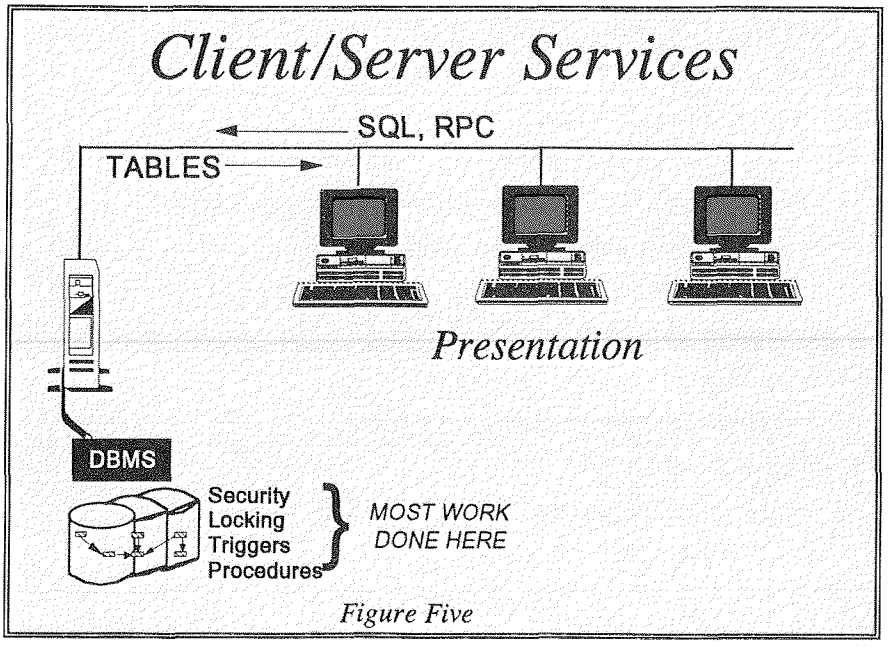


Figure Five

enterprise-wide systems. It's nothing like Windows 3.1, but it uses the same interface as Windows and therefore offers an easier training curve for anyone familiar with the interface.

Windows NT running on large symmetric multiprocessing (SMP) servers should be capable of supporting hundreds or thousands of client computers. Such an architecture would be able to challenge the largest computer systems. Of course, operating systems take some time to mature. Before companies are going to trust their mission critical systems to an operating system, it will have to earn a reputation for reliability and robustness.

While no operating system can start that way, Windows NT is the most serious new challenge to the dominance of the current computer behemoths that we've ever seen.

It is going to take two years for us to start to see results, but Microsoft in the glass house is a real potential for the future.

A new generation of 32 bit operating systems is about to take over from the older 32 bit proprietary operating systems such as MVS and VMS, and also from 16 bit operating systems like DOS. As part of this megatrend, Novell has taken major steps in migrating from the LAN world to becoming a company that markets server operating systems.

Windows NT is just one of a new generation of 32 bit operating systems that is slowly, but definitely, going to change the way computing is done on millions of stand-alone and (primarily) networked PCs, workstations, and servers. And that new generation of operating systems is going to be one of the primary differences in the way we compute in the mid-1990s from what we did in the 1980s.

(continued on next page)

Microsoft's Windows NT

- First and foremost, Windows NT is a member of the Windows famly with the same Windows interface that users know, and that is easy for Macintosh users to learn.
- Microkernel-based, 32 bit, Modular, Client/Server
- WOSA and its drivers means that application development is easier and more portable, both across architectures (MIPS, Intel, Alph, etc.) and across the Windows product line.
- Pre-emptive, Multitasking, Multi-threaded, Symmetric Multi-Programming, C2 Level Security
- Compatible or easy ports for applications: Windows 3.1, Unix, OS/2, DOS (some)
- Inexpensive when compared to many Netware or Unix environments.

Figure Six

Fabulous Two Years...

(continued from previous page)

It doesn't really make sense to talk about which operating systems are going to dominate in a general sense. Client/server architectures are going to predominate in the new style of computing. The type of functions that one needs on the client are different from those on the server. The server has to have extremely high performance and offer support for large numbers of different users doing different things. A server operating system has to have the highest level of security since there are going to be many different people accessing it. In addition, a server has to have the reliability of the best mainframe since when it's down, none of the clients are going to be served.

A client operating system, on the other hand can go down occasionally, but it can't be different or difficult to learn. Pampering of the user is the highest priority for the client operating system.

DOS, as we now know, just doesn't cut it for either client or server processes. DOS is a 16 bit operating system. Since the advent of the 386 chip, which is 32 bit, a full 32 bit operating system has been a requirement to really take advantage of the available hardware.

When IBM and Microsoft introduced OS/2 in the 1987 time frame, these two companies (and many industry pundits) predicted that sales of OS/2 would surpass those of DOS by 1991. Those forecasts were laughably wrong, but now we can look into the future and understand that DOS's lifetime is limited.

On the client, DOS's lifetime is limited because its vendor, Microsoft, plans to abandon DOS in the next major release of Windows (code-named *Chicago*, but the product may be called Windows 4.0.) Windows 4.0 is going to be a big step up in robustness from the current DOS/Windows 3.1 combo. It will be pre-

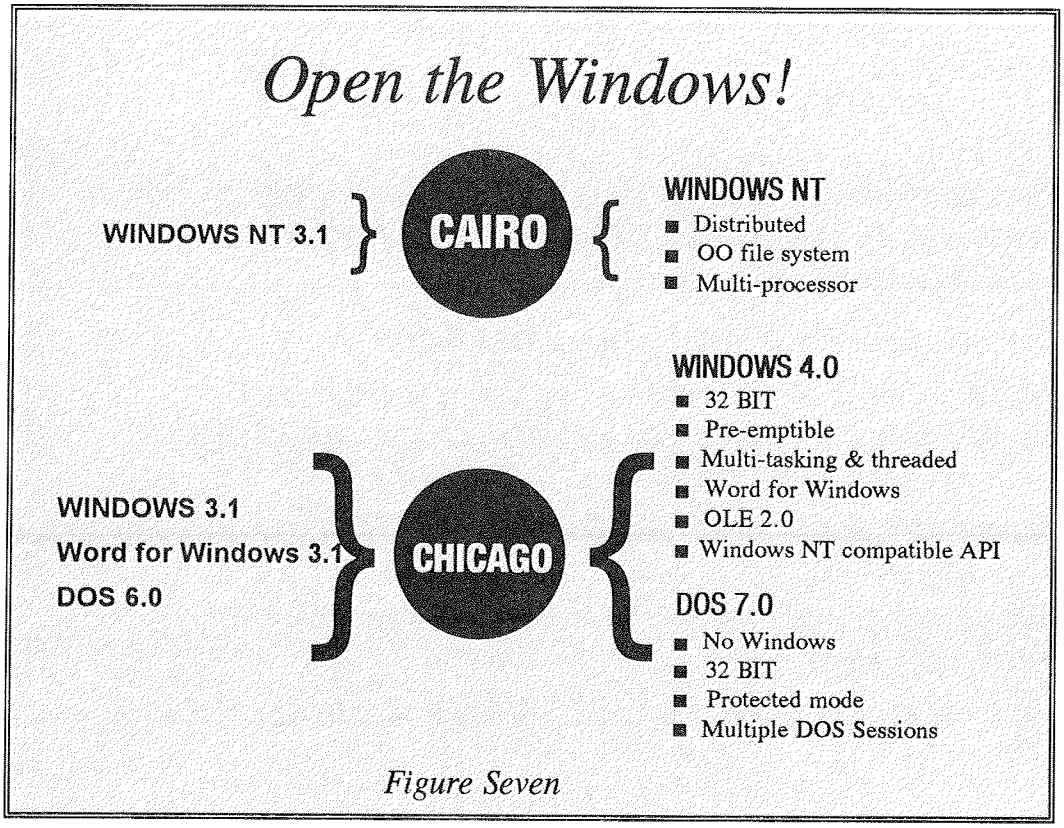


Figure Seven

emptible and multi-tasking, for example. It is Windows 4.0, not Windows NT, that is likely to take over the desktop in the 1995 time frame.

Of course, there will still be DOS over the next few years. Microsoft has talked about DOS 7.0 becoming a 32 bit operating system. But, more and more computing on the client is going the graphical route and whether it's Windows, Workplace Shell, or Motif, GUI computing is going be 32 bit.

On the client side Microsoft's dominant position is being challenged by three major groups—the UNIX brothers, IBM's OS/2 and Apple with its System 7.

UNIX, in its various forms, hasn't had a large volume of the desktop client market. UNIX on the desktop has principally been a solution for scientific and engineering workstations. One of the reasons behind this has been the very fact that there were many different flavors of UNIX, and it's anywhere from an annoyance to a deal killer for vendors and users to have to port between the versions. With Novell's acquisition of USL, Ray Noorda has made convergence in the UNIX community a top priority—we'll see if that has any impact on its market share. There is a lot of NIH (not invented here) feeling in various parts of the UNIX community, so Noorda is going to have his hands full.

Beyond this multiple personality issue for business users, UNIX has had other problems:

- UNIX integration with PCs has been only so-so.
- UNIX is a complex operating system, and is difficult to learn and administer as compared with DOS.
- The combination of UNIX software and hardware costs has meant that UNIX as a solution is much costlier than commodity PCs.

Because of these various problems, UNIX hasn't had the wide distribution and channel support of popular PC products such as NetWare and Windows.

Now that Novell has acquired USL and merged it with Novell's Univel subsidiary, some of these problems may be fixed. Novell can certainly fix UNIX channel distribution problems. However, it is unlikely that UNIX is going to be able to capture a major portion of the client side desktops—that is unless Windows NT is much less of a success than it appears likely to be.

The conclusion is that while UNIX is currently, and is likely to remain, a major factor on the server, I don't think that it is going to be an important player on the client side—not in the U.S. anyway.

(continued on next page)

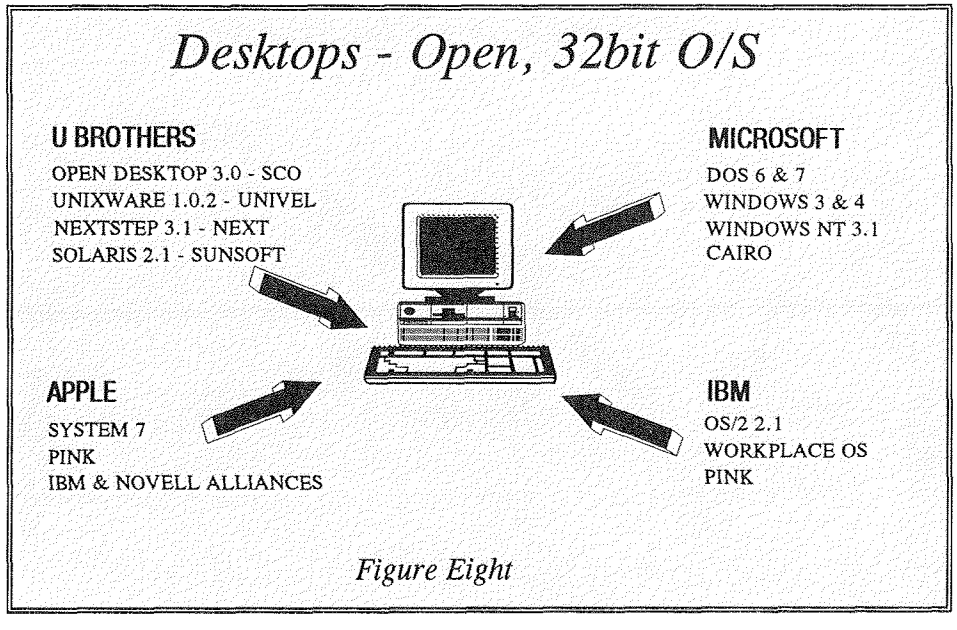


Figure Eight

Fabulous Two Years...

(continued from previous page)

IBM is shipping approximately two million copies of OS/2 a year now that it has matured into a stable production environment. The major problem with OS/2 is the lack of applications for its native environment. OS/2 2.1 now runs Windows 3.1 applications as well as DOS and native OS/2 applications. OS/2 is going to continue to evolve as a competitor to Windows NT and as a partner with IBM's AIX. The 10% of the market for 32 bit operating system that OS/2 owns now is secure. But, it will be a major surprise if IBM can regain desktop market share from its competitors.

The same story applies for Apple. The Macintosh holds about 12% of the PC market and the company's interface is reasonably common in corporate environments. However, Apple has been viewed as a proprietary environment—even though the company talks about opening up its software to clone vendors, nothing definite has happened in this direction yet. And, if and when it does, it will be too late for Apple to regain the mind share and market share that now belongs to Microsoft. The key here is that application developers understand that Windows has the largest market share and they are developing for Windows instead of other GUIs. More and better available applications simply reinforce the lead that Windows has.

When we look at the operating systems that will dominate on the server side, the situation is somewhat different from the client. IBM, UNIX, and Microsoft are players on the server side, but Apple isn't a major contender. Novell, whose DR DOS product has only a small role on the desktop, is the major supplier of network LAN software. Novell has been aggressively changing its technology and marketing to grow NetWare into the role of general purpose server operating system. The opening of NetWare 3.x to NLM applications and the release of NetWare 4.0 with its global directory services are important steps in that direction. In addition, Novell's excursion into the UNIX market gives an indication of how serious it is in becoming a general competitor to Microsoft

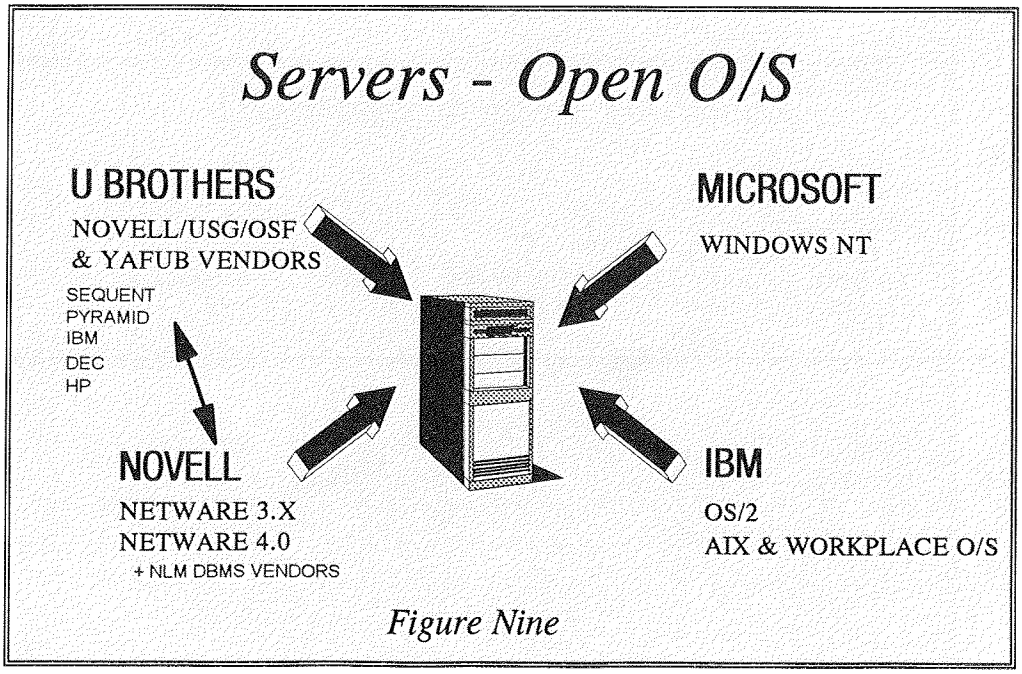


Figure Nine

UNIX on the server is a common and successful environment at this time. That is a major part of the market that Windows NT is going to aggressively attack. Since first releases of operating systems have always taken a few years to reach maturity, robustness, and fullness of

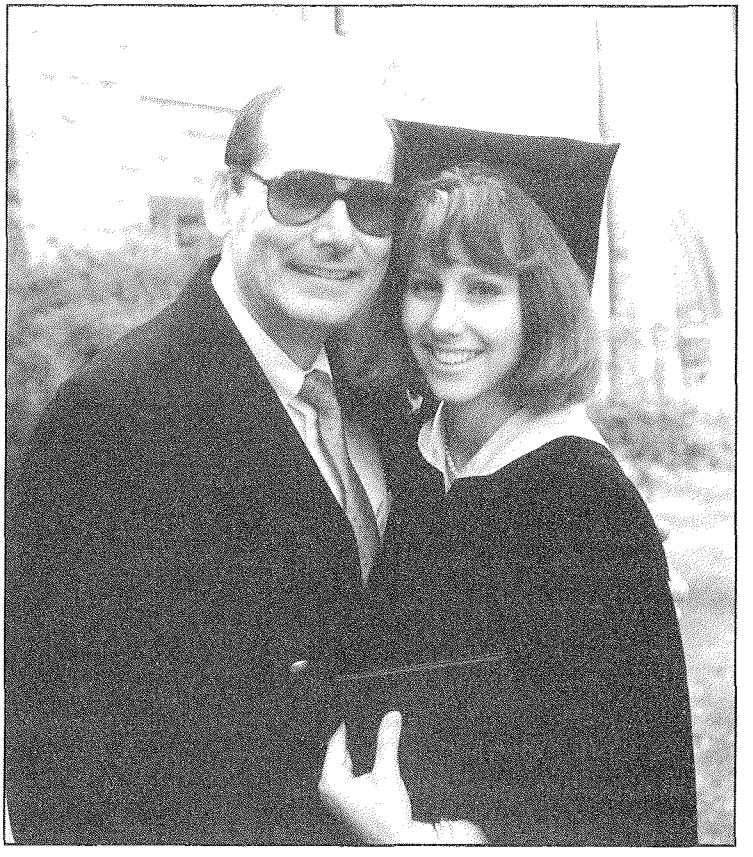
function, it would take the combination of a Herculean effort by Microsoft and a serious mistake by the UNIX vendor community for UNIX to lose the advantage that it holds at the current time. I put the probability of that combination of events happening at 40%. Most likely, UNIX will continue to grow and adjust to the presence of Windows NT as another operating system on both clients and servers.

Notebook computers with color screens, 20 MB memories and 200–400 MB hard drives are starting to replace desktop computers.

More than ever, it is now clear that PCs, and not workstations, will proliferate and dominate in client side computing. The reason is form factor: notebook-sized computers are functionally superior to the majority of desktop-sized dinosaurs that are now in use. The machine that I am writing this essay on is a Compaq 486 notebook that is very portable because of its 7 lb. weight and small size. This machine has a color screen, 209 MB hard drive and 12 MB of memory. For communications, I use a Xircom Ethernet adapter or a Worldport data/fax modem. I no longer use the desktop computer sitting in my office—except for backing up my notebook's hard drive. Even when I'm in the office, I use my notebook to connect to a network or printer rather than using the installed desktop machine.

The portability and "take home-ability" of this type of machine will mean that portable computing means never having to say you're sorry—about screen visibility, capacity, connectivity, or speed. My ideal machine for the mid-1990s is a 5 lb. notebook with active matrix color screen, embedded pointing device, 800 MB hard drive and 32 MB of RAM. Pre-installed with Windows software, this machine will revolutionize commuting to work as it will allow you to work and stay in touch through satellite video conferencing or any of a dozen other options. We're 80% of the way to having such machines right now.

I can't wait to see where the next few years will take us. *GS*



Editor George Schussel and Managing Editor Stacey Griffin preparing for Ms. Griffin's 1995 MIT graduation...

Technologies for business re-engineering...

(continued from front page)

us that they did not end-up giving away many of these machines. There are a number of different styles of pen-based computers. First is the electronic clipboard which essentially replaces traditional clipboards: you can take notes and enter them into your computer system directly. What is also great is that pen-based computers can store signatures with forms.

There are pen computers that function as general purpose computing machines; Grid and NEC are now shipping a pen-based computer that you can use either with a pen interface or with a keyboard. And, there are also hand-held computers, day timers, personal organizers, or specialized machines for inventorying, etc.

All of this mobile technology *alone* is worse than the paper, pencils, and clipboards that they replace. Mobile technologies are only good when implemented within an infrastructure of business processes that take advantage of their special characteristics. I see people all of the time moving to imaging or pen computing without first asking themselves, "why do I want it?" or even better, "why do I *need* it?"

Prerequisites to success

In order to succeed with business process re-engineering, the first thing you must have—and you can't succeed without it—is a senior management commitment. You also need to be sensitive to the concerns of the participants because business re-engineering is an intrinsically political topic. Lets say that you've been working for a company for twenty years and, all of a sudden, someone comes in and tells you that they are going to change the way you do your job—how are you going to react?

The people working on the re-engineering project need to be comfortable with uncertainty and ambiguity because, apriori, it can not be said with 100% assurance that the changes will work. You have to be able to live with an empirical approach—"well, we'll try it and if it's not quite working right, we'll correct the course." If you need to have a rigid, detailed plan that says do A then go to B, then to C, you will fail. One of the reasons you need senior management commitment is that you want them to understand that the implementation of these new types of systems has to be based on a dynamic, evolving plan. "Rigidity in planning, flexibility in execution" is a good motto for these types of projects.

You need to challenge assumptions. *Why are we doing this? Does this make sense? Are we going to make money from this venture?* One of the tales Mike Hammer tells is of Ford Motor Company. Ford had an accounting process that they wanted

Prerequisites to Success

- Secure senior management commitment.
- Be sensitive to the participants concerns.
- Make friends with uncertainty and ambiguity.
- Challenge all assumptions.
- Be bold.
- Solutions must meet minimum acceptable standards for functionality, ease of use, technical requirements, and must work within corporate constraints.

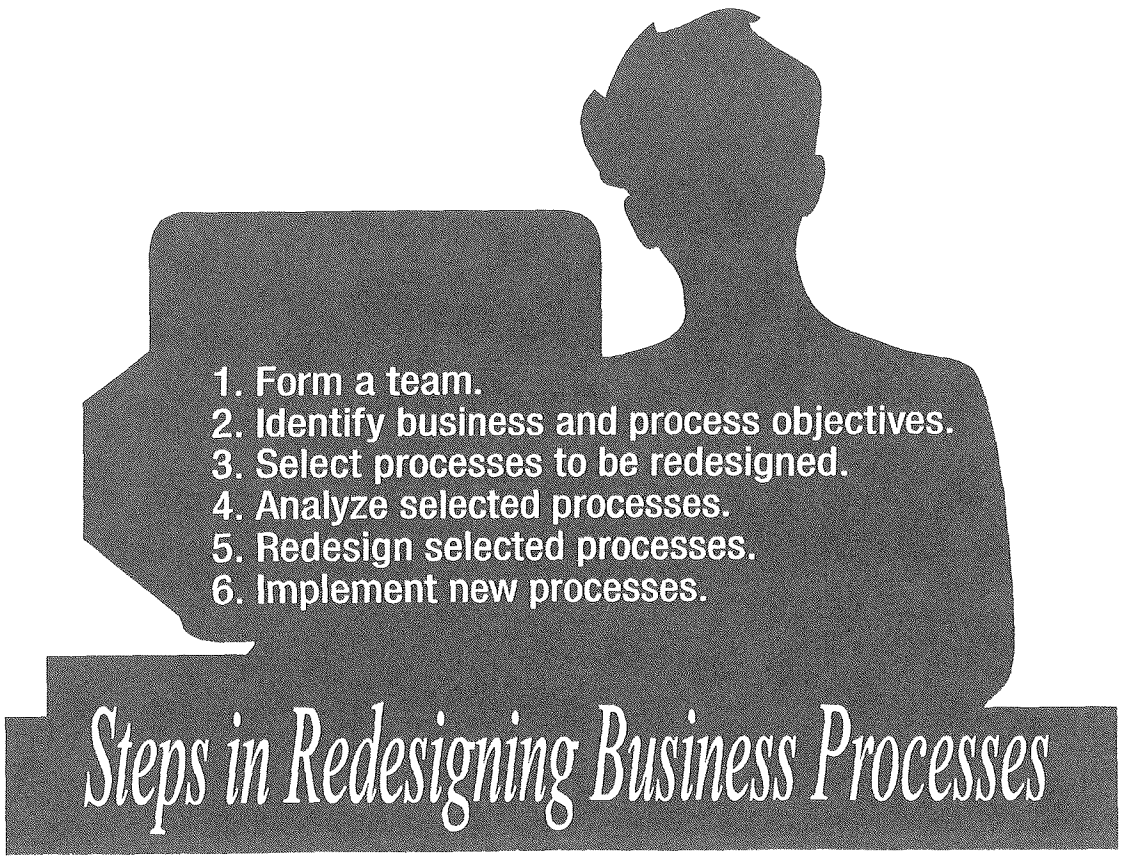
to re-engineer: they had 500 people in their accounts payable department, and wanted to reduce the staff to 400. Ford looked at other companies in the same business, and in the process, examined Mazda Corporation. Mazda has an accounts payable department of five people. Even accounting for size differences, five people in contrast to 400 just didn't seem to add up. So, Ford went back and decided to cut 80% and reduce the staff to 100. They set aggressive goals, and what they had to do to meet those goals was be prepared to think radically differently. In the end, Ford eliminated the requirement that companies invoice them. Now, at the loading dock, when suppliers drive their trucks up, if the incoming shipment is accepted, a check is cut without an invoice.

However, there are downsides to Ford's new procedures. It shifted the burden from the accounts payable department to other people both within and outside of Ford. For example, partial shipments are no longer accepted which raises suppliers' costs. So, when you look at your solution, make sure that it meets minimum standards for functionality, technical requirements, and insure that it works within your *corporate-wide* constraints.

Steps in redesigning business processes

When you design a business process, the first step should be the assembly of a re-engineering team. On your team, you need to have at least one person who represents senior management. You also want to have domain experts—if I'm going to re-engineer my payables, I want to have somebody on board who knows what it means to pay a bill. You'll want to have a financial expert, an IS expert, a re-engineering expert, and industry experts—and more than one of these people may be

(continued on next page)



Technologies for business re-engineering...

(continued from previous page)

external consultants hired on for project guidance.

Once the re-engineering team has been assembled, the team must then identify the objectives. Don't concentrate on making individual tasks better, but focus on the overall corporate objectives and goals. Fit processes into the context of these global objectives. Also, when setting these goals, use corporate metrics not departmental measurements. And, even though it is difficult, I believe that it is important to measure the quality of work life improvement—it is easy to joke about this but a lot of energy at many companies is wasted by people complaining. We want to make sure that the employees have nothing to complain about. And, I really want to emphasize that you should be aggressive in goal setting.

Identify the processes that are candidates for re-engineering, and work with them at the appropriate scale. What may be the appropriate scale for one company may not be appropriate for a different company. The processes that are identified as candidates for re-engineering should have potentially large paybacks: it does not make sense to put in an enormous amount of effort to produce only small paybacks.

When you select a process, measure the current performance as well as is possible—it is important to have a base line from which to work. Information from the processes' customers is essential. I can't tell the quality of the job you're doing simply by asking, "how are you doing?" So, ask the customers—the people who receive the end result of the business process in question.

Focus on performance. Establish aggressive performance targets, and *analyze potential side effects*. I know of people who have put in imaging systems and redesigned work flows,

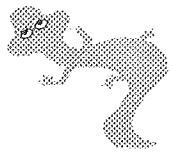
then replaced the filing clerks with people who do nothing all day but sit in front of workstations. This project then has produced expensive side effects of increased workers' sick time. You have to anticipate and design for such occurrences.

People commonly set goals for 20% improvement, but that is very much in conflict with the *"Be Bold"* that I talk about as a pre-requisite for success when restructuring. What tends to happen when the goal is for 20% productivity improvement, only the small, incremental things that can be done are done—20% improvement is always achievable without much pain. However, if you decide to go for a 60%, 70%, or 80% improvement in productivity, you're not going to be able to achieve that goal by tinkering. ***Be bold!***

Some concluding thoughts...

- ☑ Document your selected process and try to have a full comprehension of how your company is currently operating. Include all external interactions. Know what the system boundaries are. Do a personnel, hardware, and information systems inventory. Look at budgeting.
- ☑ When you think about the new processes, a major source of underestimation is always the training time. Allow adequate time for learning, and prepare for people who can't make the adjustment. I can guarantee that if you have 150 COBOL programmers, if 50 of them become competent C programmers by the end of one year, you have one of the best COBOL staffs around.
- ☑ Redesign your processes.
- ☑ Look for opportunities to remove excess steps and processes, and to perform step in parallel.
- ☑ Anticipate new requirements.
- ☑ Start with pilot projects and prototypes. Pilot projects should be processes that are important to the company, but not something that is extremely time constraint.
- ☑ When you're prototyping, you can't prototype three transactions, and then extrapolate the results to 10 million transactions. Keep in mind that computer systems, information systems, and organizations all behave non-linearly. Stress-test your prototyping.
- ☑ Your projects should have real deliverables due within one year. The probably of failure goes up exponentially with the length of the project.
- ☑ Many of the technologies involved in business and IS re-engineering are non-standard. When you're trying to integrate client/server, pen-based computers and imaging, the resulting integration problems can kill the project.
- ☑ Keep in mind that your initial analysis is the key that determines the final scope of the work. *HE*

This article is based on a presentation given by Mr. Edelstein at DOWNSIZING EXPO this past spring. Edelstein is a principal of Euclid Associates, a consulting firm specializing in database management and document image management. Edelstein consults to both computer vendors and users, and teaches professional seminars on a variety of topics. He is consistently rated as one of DCI's top speakers. Edelstein is reachable at (301) 983-9550.



UPCOMING downsizing Events...

DCI's most popular conference/exposition is being held one more time this year: CLIENT/SERVER WORLD, in conjunction with DATABASE WORLD, is returning to Chicago, December 8-10, 1993. There are nine conference tracks between both shows: *Object-Oriented Technology Conference, Database Technologies Conference, DB2/Information Warehouse Conference, Xbase Conference, Database Connectivity Conference, Client/Server Databases Conference, Managing the Client/Server Environment Conference, Client/Server Networking Conference, Building Client/Server Applications Conference.* Keynotes are being given by several renowned industry figures including: Chris Date, Michael Stonebraker, George Schussel, Larry DeBoever, and William Zachmann. This is the only show where all of the issues involved with client/server computing are explored in-depth.

A new seminar which is being held December 16-17, 1993 in Washington, DC, is *Analysis and Design for Client/Server Applications.* Instructor Jim Davey will be covering a new design methodology, event driven client/server development (EDC/SD), that will help to resolve the mainframe/PC culture clash.

One of DCI's most popular seminars has been updated for its fall date; *Cheryl Currid: Implementing Downsizing* will be in Orlando, November 11-12, 1993. In this two day seminar, Currid covers downsizing vs. rightsizing, approaches and strategies for downsizing, the link with re-engineering, downsizing case studies, organizational and political issues, downsizing products and technologies, networking options, and client/server databases.

Finkelstein's Practical Guide to Client/Server DBMS Computing, being held in Ottawa, November 17-18, 1993 has also been recently updated. Course instructor Richard Finkelstein will be covering the topics: building a client/server DBMS, evaluating database servers, database server guidelines, middleware, client/server tools, and merging object oriented and relational technologies.

Herbert Edelstein's two-day seminar, *Implementing Client/Server Applications and Distributing Data*, will also be in Ottawa, November 15-16, 1993. The perfect preamble to *Finkelstein's Practical Guide to Client/Server DBMS Computing*, this seminar covers the topics of: client/server computing, open systems, networks, relational DBMSs & SQL, database integrity, and distributed data.

The three day seminar, *Business Process Reengineering*, teaches attendees how "using information technology to renew the business" can be beneficial to any company's bottom line. In San Francisco, October 25-27, 1993, instructors Roger Burlton and Brett Martensen will cover topics including: case studies, process renewal methodology, enabling technologies, managing workflows, tactics for success, the change implementation phase, and techniques and tools.

For more information on any of these classes or conferences call DCI at (508) 470-3880.

Schussel's DOWNSIZING JOURNAL October 1993 Volume 3 Issue 2	**Editor:** Dr. George Schussel **Managing Editor:** Stacey S. Griffin **Subscription rates:** ● $199 annually for U.S. residents ● US$225 annually elsewhere	**SDJ is published monthly by:** Digital Consulting, Inc. 204 Andover Street Andover, MA 01810 USA 508-470-3870 FAX 508-470-1992