

Schussel's

DOWN SIZING JOURNAL

February 1993

In This Issue:

Novell, UNIX, and The New World Order

Editor Schussel takes a look at the potential repercussions and significance of Novell's recent acquisition of USL1

The Last Thing IBM Did Right... *Contributor Jeff Tash sheds some insight on IBM's reactions to "open" computing and downsizing over the past five years*1

George's Visit to Santa Teresa *The second installment in a three part series by Editor Schussel on some of IBM's most recent developments at the Santa Teresa Labs*7

Event Driven Client/Server Development (EDC/SD), Part I *DCI's Senior Consultant Jim Davey gives an overview of a new client/server software development methodology that he has been developing*13

Upcoming Downsizing Events16

Novell, UNIX, and the New World Order

I was in Summit, NJ on Tuesday, December 22, visiting UNIX System Laboratories (USL) on the day that Novell's acquisition of USL was publicly announced. Novell's interest in UNIX as a counterweight to Microsoft's impending Windows NT had not been a secret. Prior to this merger, Novell and USL had formed a jointly held subsidiary, Univel, to create and market a standard desktop version of UNIX that could be sold through the Novell distribution channel. Also, friends of mine at USL had kept me informed about Ray Noorda's (President of Novell) overtures towards USL.

(continued on page 9)

The Last Thing IBM Did Right...

by Jeff Tash
President
Database Decisions

The last thing IBM did right was DB2. Even though RDBMS products such as Oracle and Ingres were introduced long before IBM's entry into the market, the truth is that relational technology never really soared in popularity until IBM released DB2. The same phenomenon occurred with PCs; it was the introduction of the IBM PC that legitimized the concept of personal computing. Although Apple, Commodore, and Radio Shack had earlier blazed the trail for PC pioneers, it wasn't until the advent of the

(continued on next page)

The Last Thing IBM...

(continued from front page)

IBM PC that the computing landscape was changed forever.

Historically, the industry has always looked to IBM for leadership. Over the past few years, the most pressing issue that IBM has been asked to respond to is the challenge of "open" computing. Once again, IBM has attempted to assert its natural role as industry leader. The result has transformed IBM in what can best be described as *corporate perestroika*—industry imitating politics. Suddenly, the world's most proprietary computer company has become a powerful advocate for *open systems*. In its rush to be as *open* as anybody, IBM has devised a series of grandiose plans which Gartner has affectionately dubbed *marketectures*. These began with SAA, which was designed to provide full portability across all of IBM's diverse proprietary products, and AD/Cycle which promised to automate automation by pulling together all software development tools under the umbrella of a single core repository. These were followed by SystemView, which promised to provide a framework that would integrate all system management functions for an enterprise-wide, distributed

computing environment, and Information Warehouse, which was supposed to enable end-users to access information from anywhere to anywhere.

In essence, if you've followed IBM over the past decade, you'll agree that the message has been, "we're a customer-driven company—we're focused on the customer, we're listening to the customer, and we're going to respond to the customer." Perhaps it was too much attentiveness to customers that has led to IBM's great downfall. They've been so busy listening and promising to be everything to everyone, that IBM has forgotten how to lead. Today, one of IBM's biggest problems is that it has *set completely unrealistic expectations*. Now, after years of failing to deliver on impossible promise after impossible promise, IBM has a severe credibility problem. No one believes in them anymore. The result is a company that is in a deep funk—IBM's role in the industry is diminishing, its size shrinking, its stock tumbling, and the company is struggling.

To examine what's happen to IBM, we should begin by asking the question, what is *open* computing? Virtually every computer company today is selling what can best be described as *proprietary open solu-*

tions. No one is selling *true open* systems because there is no such thing. The reality is that customers don't "buy" *open* systems, they "build" *open* systems by selecting products that can work together effectively.

Open is a marketing buzzword. How could any company possibly be opposed to *open*? After all, what is the opposite of *open*? Can you imagine a company actively promoting *closed* computing? One of the main reasons why *open* is so popular today is because it is the only proven way to compete successfully against IBM.

For many years, IBM was viewed as untouchable. Originally, the computer industry was known as *Snow White and the Seven Dwarfs*. Snow White was IBM, and the seven dwarfs were GE, RCA (both of which eventually quit the computer business), Burroughs, Univac, NCR, Control Data, and Honeywell, all of whom failed to win much market share against the mighty IBM. The only company that ever truly succeeded in combating IBM was DEC with its VAX/VMS systems. DEC found a weakness in IBM's strength, and attacked it. IBM's strength was its controlling share of 70% of the mainframe market, a 70% share in PCs, and a respectable share, albeit not as

dominant, in the mid-range market. IBM's weakness was that all three market segments used radically different architectures. DEC countered IBM with a brilliant marketing strategy based on *one architecture and one system—from the desktop to the data center.*

SAA: the answer to open or just more FUD?

IBM's response to DEC's great market success with VAX/VMS was SAA. IBM promised to make its proprietary systems *open*. SAA's approach employed a time-tested IBM marketing strategy known as FUD—Fear, Uncertainty, and Doubt. The FUD principal was based on the old adage that no one ever lost their job by recommending IBM products. FUD was often cited as the key to IBM's global success. Many CIOs achieved their position by dutifully following IBM's lead, and more than a few believe that they owe their successful careers to IBM. IBM brought along an incredible number of people with them as they rose to greatness.

Beginning with SAA, IBM changed its FUD marketing strategy to FUDGE—Fear, Uncertainty, and Doubt combined with *Great Expectations*. This pattern was repeated with AD/Cycle, SystemView, and Information Warehouse. For instance, with

SAA, IBM promised consistency, compatibility, usability, portability, cooperative processing, and common applications. So where did SAA go wrong? The problem is that SAA turned five years old on March 17, 1992, and about the only success that IBM could point to was DB2. The glitter had faded from all the great promises of the past, such as SAA's OfficeVision. Today, SAA is out, *open* is in. AD/Cycle looks poised for the next great fall from grace, especially now that Bachman and Intersolv have dropped their support.

IBM's "Save the Mainframe Foundation"

A lot of people believe that SAA was really the "Save the Mainframe Foundation." The problem is that nobody wants to be the last one left on the mainframe, responsible for turning off the machine. This has created a great dilemma for IBM as customers have lined up in droves to abandon their mainframe systems. This crisis is staggering for IBM, who still heavily depends on mainframe sales because of their huge profit margins. The result has been a massive drop in earnings.

Helping to fuel the rapid migration away from mainframes is the use of zero-based, cost-sharing, charge-back algorithms. Charge-

backs are commonly designed to share the cost of mainframe computing among all users. But, smart users have discovered that they can escape recurring monthly charges by moving their processing onto PCs and LANs, or inexpensive UNIX machines. Whenever this happens, the remaining users on the mainframe are left with bills that go up each month because the costs must be spread across an ever-shrinking user base. The poor guy who gets stuck as the last one on the mainframe is going to get socked with an enormous charge!

As more and more users move off the mainframe, the pressure for others to downsize increases enormously. Today, we are facing a stampede mentality. People want to migrate off the mainframe, ASAP. They think that they can put LANs together to share data, applications, networks, and devices. Moreover, they're convinced that downsizing is going to save lots of money instantaneously. There's a real problem, though, with this scenario. The reality is that getting to a downsized utopia is extremely difficult. There are a host of issues that first must be addressed with regard to security, integrity, recovery, backup, budgeting, licensing, sup

(continued on next page)

The Last Thing IBM...

(continued from previous page)

port, performance tuning, capacity planning, and all of the things that we've done for years with mainframes. It's one of those situations where I've got some good news and some bad news. The good news is that if you come from an I/S background, you understand the requirements. The bad news is that few, if any, of the solutions that worked so well in the glass house data centers are applicable when it comes to managing distributed environments.

Moving from mainframes to distributed systems is a very difficult transition. And, there's this perception that people who downsize are going to immediately save money, which is really kind of a joke. When you look at the total cost of computing, you find that hardware is only one of many different expenses. Also, before you can begin to benefit from downsizing, you must first be prepared to make substantial investments in infrastructure, such as networks, servers, and new software. You really need to inform everyone that downsizing is going to cost you money before it will save you money. You also need to examine organizational roles and responsibilities. Imagine the cost to an organization if you have

500 Excel users, each spending three hours installing a new product release. In today's distributed environments, we find repetitions of functionality like this going on everywhere. Real total savings won't accrue from downsizing until the entire I/S organization is re-engineered and restructured.

There is more to downsizing than just potential cost savings. What is really tempting people to downsize and move to client/server is the *graphical user interface* (GUI). People love a GUI—it's seductive—you ought to think of it as *Win-tendo!* After moving to a GUI environment, most people would rather take a cut in pay before they'd consider going back to working with character-based terminals. What GUIs provide are subsecond response times, local autonomy, and maximum flexibility.

IBM's stand on enterprise computing and downsizing

In the old days, this industry was simple. When we talked about the enterprise, we talked about the big "E"—mainframes, minis, and micros. Everybody could relate to where they and their machines stood on this chart. The problem today is, what's a mainframe, what's a mini, and what's a micro? A

much more appropriate way to classify our industry is in terms of functionality. I/S does three things: batch processing, transaction processing, and end-user interactive computing. This last area, end-user interactive computing, is where client/server is having its biggest impact. Client/server is mainly being used to implement two types of applications: on-line complex processing (OLCP) and decision support systems (DSS), also known as executive information systems (EIS).

On-line complex processing is distinct from on-line transaction processing (OLTP). For one thing, the audiences these systems target are very different. OLTP systems are mainly used by clerical workers. OLCP systems are designed for knowledge workers. OLTP systems usually involve high-volume, simple transactions where each transaction is measured in seconds; transactions are highly uniform with many people doing the same thing at the same time, and most transactions are highly repetitive. OLTP systems are mainly concerned with the price per transaction per second (\$/TPS). Character-based terminals designed for heads-down data entry are still often the most cost-effective input devices for OLTP applications. OLCP

systems, on the other hand, are frequently very complex, with transactions lasting for minutes or even hours. Also, there are few people performing the same task at the same time. OLCP systems must provide maximum flexibility, and benefit tremendously from GUI technology. Because of their complexity, OLCP systems are frequently designed and built using a prototyping approach known as RAD—Rapid Application Development.

Historically, IBM has owned the batch and OLTP markets. The reason why most of the world's batch and OLTP continues to run on IBM mainframes is because there's been no better or more cost-effective environment than MVS for running these types of applications. However, I do see this beginning to change, especially now that versions of CICS that can run on cheap and powerful RISC machines are becoming available. The ability to move CICS applications off expensive mainframes and onto inexpensive micro-based platforms is creating an opportunity called RAM—Rapid Application Migration. This will quicken the pace of downsizing for many mainframe users.

As the industry evolves away from centralized, host-based data centers to distributed, network-based

computing, there is a need for a framework that allows users to tie all of the disparate pieces together into an integrated environment. IBM refers to this as *enterprise computing*. It entails three components: databases, networks, and systems management. This last category can be thought of as including nearly every word in the English language that ends with the letter "y" such as security, integrity, recovery, reliability, repository, etc.

IBM has recently announced some new products referred to as CID—Configuration Installation Distribution, which promise to move forward the notion of enterprise computing. Collectively, these tools, which operate within the SystemView framework, provide an *open* scheme for managing networks. These are some outstanding new products, and IBM deserves to be credited for taking a strong leadership position.

IBM on interoperability

When people use the term *open*, generally they mean one of two things. First, *open* to many people simply means the opportunity to buy cheap hardware. Customers don't want to be dependent on any single proprietary hardware platform. The second thing that *open* means is interoperabil-

ity. However, when you step back and look at interoperability, what you'll discover is that it's really three-dimensional.

The first dimension of interoperability involves cooperative processing. The objective is to enable end-users to run applications that span multiple processes. The goal is to provide what is known as a single system image—where an application executes on multiple machines simultaneously, but provides the appearance to the user as if it were running on a single machine.

There's been a lot of confusion between the differences in cooperative processing and client/server computing. IBM has often stressed cooperative processing based on the use of LU 6.2 protocols known as APPC, Advanced Program-to-Program Communication; don't let the word advanced in APPC fool you. APPC may have been advanced back in the 1970s when it was first introduced, but by today's standards, few would consider it terribly advanced. What APPC provides is a set of protocols, or verbs, that essentially allow the simulation of a telephone conversation. There is an LU 6.2 verb that corresponds to picking up a phone and listening for a dial tone. There

(continued on next page)

The Last Thing IBM...

(continued from previous page)

is another verb that let's you pick-up a ringing telephone. There's a verb associated with speaking, and another that lets you listen. Finally, there's an LU 6.2 verb for hanging up the phone when you're done. Using these verbs, applications can be written which can execute on multiple machines simultaneously by engaging in program-controlled conversations.

You get into the difference between APPC and client/server with the concept of a single system image. In essence, client/server is a subset of cooperative processing. With APPC, the only one who sees a single system image is the end-user. For the poor programmer, life's never been more complex. When we talk about client/server, what we mean is that we're shifting the boundary so that both end-users and application programmers see a single system image. This is accomplished through the use of APIs, Application Programming Interfaces. The key to interoperability is to select a set of compatible APIs which allow applications to be pulled apart and plugged together seamlessly over a network.

The second dimension of interoperability pertains

to CASE. Application development can be thought of as a type of manufacturing. As is true of all manufacturing, there is a need for a bill-of-materials, which describes the components and subassemblies which go into a manufactured product, and a well-defined process definition which defines precisely the steps to be followed in the manufacturing process. With respect to CASE, the bill-of-materials corresponds to the repository, and the process definition refers to the development methodology. The goal of CASE is to allow multiple tools to interoperate during the development of applications.

The third dimension of interoperability deals with managing systems across multiple environments. In the distributed environment, what we need is to provide the same level of reliability that we have traditionally offered in glass house data centers. We need to be able to manage our distributed environment as effectively as we have managed the mainframe. The difference is that now the network is the system. We need a full range of tools for managing this distributed environment as a service business. Given the special importance of databases and networks to enterprise computing, it is not surprising to find that there is a

special need for tools to help manage, control, monitor, and administer databases and networks. IBM promotes NetView as its solution for managing all the physical resources scattered throughout the enterprise network. And, Information Warehouse has the responsibility for managing the logical resources by describing what data exists, what it means, who has access to it, and so forth.

What should IBM do?

What should IBM do? They've got to go back to basics. I remember my very first day when I went to work for IBM. I was issued a pencil and a notepad which both boldly displayed the word "THINK." I haven't seen that slogan from IBM lately, and I certainly haven't noticed it in its behavior. The company must stop trying to be everything to everyone.

Right now IBM is actively sending out far too many conflicting messages to its customers. For example, if you ask IBM for guidance on how to build open client/server systems, be prepared to come away very confused. For instance, if you ask IBM for advice on which client platform to buy, "When should I buy OS/2?, When should I choose AIX?," and "When does it make sense for me to

(continued on page 15)

George's Visit to Santa Teresa

This is the second article in a three part series (expanded from two) on some of IBM's developments at their Santa Teresa Laboratory. Next month's article will cover: DRDA, AD/Platform, and object oriented research.

The Information Warehouse

When customers come to IBM's Santa Teresa Labs (STL) for briefings, the most frequently discussed topic is IBM's Information Warehouse (I/W). I/W is an overall architecture for providing access to distributed files and databases across a multi-vendor, distributed environment. It is intended for decision support types of applications, and normally wouldn't be appropriate for transaction processing, production, or update intensive types of applications. What makes I/W so interesting is its inclusionary, not exclusionary, approach to participants. A very large number of end-user tools such as Lotus 1-2-3, Focus, and Nomad, are supported to operate against a long list of potential databases and files. Although

SQL plays an important part in the I/W architecture, I/W's access is not limited to relational databases. One of the principal enabling technologies of I/W is Information Builder's EDA/SQL which offers access to dozens of different multi-vendor database and file systems.

More information about the I/W will be available later in 1993 when IBM publishes the architecture's specifications. In fact, it is currently known that there are two different basic approaches to the architecture. IBM is offering customers a fundamental choice (both can be used, however) between access requiring dual databases or direct access to single data sources. While direct access initially seems to be simpler to implement and manage, IBM staffers asserted that the dual database approach will produce better performance characteristics for many customers.

In *Figure 1*, these two approaches are illustrated by the higher and lower horizontal arrows on the right side of the diagram. IBM's technology and DRDA (a further explanation of this technology is to follow) are the underlying linchpins for the upper approach, the dual database concept. The idea here is that production data is maintained in an IMS, DB2 or other high performance, high availability database. The information that is desired for

decision support purposes is extracted out of that database and a copy is then maintained (in DB2, or any other DRDA DBMS) for query and browse purposes. Once the extraction has been made, the original and copy databases are kept in synch by an approach IBM calls "Propagation" which uses a 2-phase commit to insure that both copies of the necessary data are kept current and accurate. Direct access to a single database certainly seems easier to implement, but many of IBM's customers prefer to extract from operational databases and to create a second informational database. This second, informational database can be merge of data from a number of sources. People who prefer this approach usually are very concerned about performance issues, which can be monitored and more closely controlled with the dual approach.

The lower approach in *Figure 1* illustrates direct access to the production database or file, and is based on the EDA/SQL technology developed by Information Builders (IBI). EDA/SQL is only a read (not updatable) technology. Another issue for potential users is whether performance will be adequate. For example, many IS managers would not want end-users doing unanticipated browsing or sequential

(continued on next page)

George's Visit to...

(continued from previous page)

reads through the production databases.

While many of the pieces for I/W are already in place, IBM has yet to deliver the connectivity and management facilities that allow I/W to be used by those without superior technical staffs to integrate the diverse pieces. Important upcoming additions are:

1. **Propagation:** While a limited version of propagation is now available (IMS to DB2, on the same machine), fuller propagation facilities will be delivered later in 1993 and early in 1994. Part of this delivery will be support for asynchronous (lazy) updates.
2. **Copy Management:** Automation of management of multiple copies of data should be at least partially available across the DRDA architecture before the end of '93.
3. **Business Information Location:** This refers to a distributed capability of accessing metadata, or business definition definitions about the data that is located in the I/W.
4. **DRDA support:** Full integration of the I/W with the DRDA architecture is likely to be delivered in 1994.

IDAPI

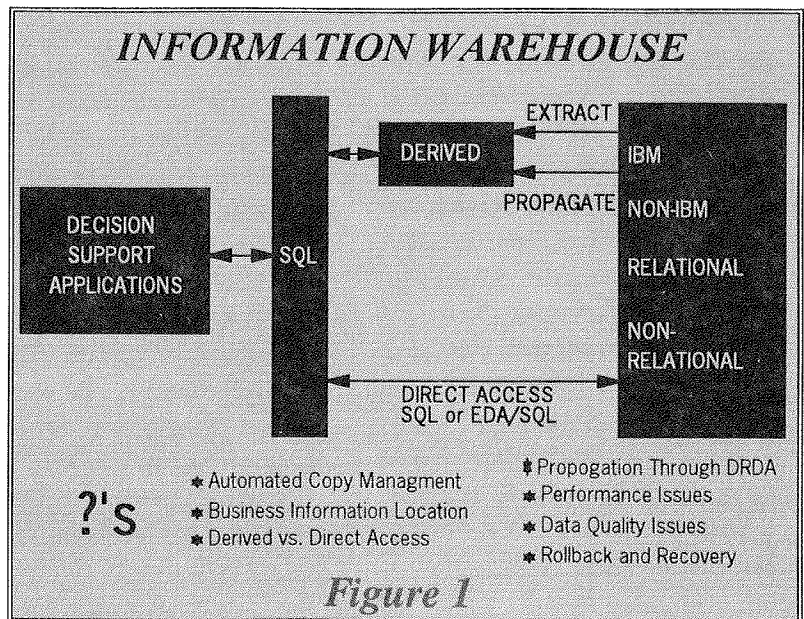
On the day that I visited Santa Teresa, IBM in combination with Borland, Novell, and WordPerfect, announced their joint proposal for a client side interface to local, remote or distributed databases. IDAPI is a derivative of and replacement for Borland's previously announced ODAPI (Object Database API). Like Microsoft's ODBC, it is an implementation and extension of the SQL Access Group's CLI (Call Level Interface) specification. CLI defines a simple interface for having various applications talk to a remote database. IDAPI provides a facility for mapping requests from one database language (e.g. Paradox) to requests in SQL. Applications constructed against existing databases, can thus access other databases without change to the application by acquiring the driver for that database. Ac-

cessing different databases, then, becomes as easy as changing printers on your PC and choosing a different driver for your word processor.

In the same way that has happened with every other technology announcement over the last two years, Microsoft and IBM end-up on opposing teams for the CLI. Users, however, can have the best of both worlds since there is nothing incompatible between ODBC and IDAPI. It certainly will be possible to install drivers of both types on your client PCs for access to miscellaneous databases.

A lot of details on IDAPI weren't available at the time of the announcement, but there are going to be differences between IDAPI and ODBC in at least the following senses:

(continued on page 15)



Novell, Unix, and...

(continued from front page)

Benefits and Synergies

USL and Novell officials initially promised that the merger of the two companies would speed the integration and interoperability of their two operating systems by:

- ☐ Creating a common application programming interface that would allow developers to write applications that would operate under both UNIX and NetWare. This is the same type of benefit that Microsoft is promising its Windows developers with the Win32s API—a bigger market with the same effort.
- ☐ Adding Distributed Computing Environment (DCE) support to NetWare—it's already a UNIX staple. This addition would speed developer development and delivery of true distributed database applications.
- ☐ Creating common systems applications such as administration, data security, and network management between NetWare and UNIX.
- ☐ Strengthening UNIX to PC connectivity with NetWare file, database, and host access services.
- ☐ Integrating UNIX's strengths in wide area

and enterprise computing into NetWare.

This list is impressive and clearly, if all of these promises are kept, both UNIX and NetWare users will benefit.

Beyond these explicit, initial promises, an analysis of the merger acquisition seems to promise there are even more benefits to result. From a user's point of view, it's hard to see the downside of this merger/acquisition. Some of the benefits that this industry can anticipate include:

- ☑ Novell's strengths in the distribution and packaging business will give broader exposure to UNIX than it would have received by staying under AT&T's wings.
- ☑ Novell's credibility in commercial markets will only be a favorable feature for businesses that are considering server operating systems—they will know that Novell understands their requirements.
- ☑ Novell is the big-time product in PC/commercial computing, and when its imprimatur is on UNIX, the UNIX penetration into commercial markets can only benefit.
- ☑ The coming (for sure) integration of UNIX and NetWare communication functions and protocols will be a plus for all com-

puter users. A high degree of UNIX/NetWare integration will also open doors for more UNIX sales to sites moving from LANs into true client/server computing. Without this merger, I would say that the upcoming Windows NT option would have been likely to pick up much of the upsizing client/server market. Now, Novell is going to grab a lot of that business.

What did the insiders think?

The reaction to Novell's merger/acquisition proposals at AT&T and USL was mixed. Pioneering the move to open systems in a vendor neutral environment is something that appeals to a lot of UNIXphiles. But, while being a part of Novell has a lot of advantages, some don't view the "vendor neutrality" as one of those advantages.

Likewise, those inside Novell had mixed feelings about the acquisition of USL. With its NLM technology, Novell had already made a serious move into the server operating systems business. "Why do we need UNIX and all that baggage when we can get much of that market for ourselves?"

(continued on next page)

Novell, Unix, and...

(continued from previous page)

UNIX has come a long way

UNIX has come a long way from its incipient days as a time sharing, word processing support system developed at Bell Labs by Dennis Richie and Ken Thompson. When Thompson arrived at UC Berkeley as a visiting professor early in UNIX development, he gave the product much more communications functionality. At the same time, very liberal licensing policies from both Berkeley and AT&T meant that seed copies of UNIX were carried far and wide by some of the best and brightest computer science graduates. Copies of the UNIX license ended-up implemented at computer companies such as Apple and Sun. By 1980, UNIX had become a kind of counter-culture computer cult. When AT&T decided to get into the computer business, UNIX naturally became a rallying point.

USL is not in the retail or distribution business. For the most part, it licenses UNIX base code to computer companies such as Sequent, Pyramid, Apple, and Sun, each who adapts the code for their specific hardware implementations. In this sense, the company's distribution model resem-

bles that of Microsoft. Yes, you can buy DOS at the retail level, but it is mostly sold by the PC companies that manufacture the hardware and provide DOS with the basic machine.

Novell and Microsoft—we're defining the next generation of computer leaders here

Certainly the computer and general business presses were interested in the Novell/USL announcement. However, because it came just before Christmas, I don't think it got the play that it would have received at another time. History will look back on this piece of news as one of 1992's biggest stories. As a matter of fact, I think that it's one of the two most important computer industry news stories of last year. (The other, of course, being the stock market's awakening to the desperate straits of IBM's overall business.)

By mid-January this year, Microsoft's overall market value equaled IBM's. Novell's market value surpassed that of DEC well over a year ago. What we're dealing with here is a fundamental paradigm shift in industry power. IBM and DEC had the power and could create the standards (and reap the wealth) in the past. The decade of 1990 looks like it will belong to

Microsoft and Novell, more than any other company. This is true for a number of reasons:

- ⇒ They are software companies. We've all been saying that software has more value added than hardware for some time, now everyone can see the truth to that statement. The two possible exceptions to this rule are the market's dominant semiconductor companies, Intel and Motorola.
- ⇒ Microsoft owns the desktop. I've had to take a lot of criticism for saying this in the past, but now there can be no doubt in the truth of this statement. I understand that even IBM is having its developers write the Windows API as a priority over Presentation Manager in beginning projects.
- ⇒ Novell owns the LAN market, and is making a serious move into low-end, PC-based server applications with NetWare NLM technology.
- ⇒ UNIX is the anointed "open systems" champion and the world, beginning with the U.S. Government and its procurement practices, have deemed that UNIXness is equal with goodness.
- ⇒ With UNIX in the Novell camp, Novell will fully cover the server operating systems business. They

offer NetWare on the PC and UNIX on larger enterprise computing rigs like Sequent and Pyramid.

Novell's and Microsoft's roles in the open systems market

☛ Novell needs UNIX to have credibility in its competition with Microsoft for the "open operating systems (server side) market." By now everyone understands that the computer business is going through a transition to client/server architectures. Computers on both the client and server side will, of course, need true operating systems. The client side has to look attractive with GUIs for single users, while the server side will be judged on the basis of robustness and performance for multiple users.

Open can be defined in many ways:

- ☛ Offering users the flexibility to easily change hardware vendors (the software vendors like this definition, of course).
- ☛ Offering users the flexibility to easily change software vendors (the hardware vendors like this one).
- ☛ Allowing customers to easily communicate and interoperate between

multiple systems from multiple vendors.

- ☛ Allowing the customer a choice of software and hardware options in a broad market—not being locked into any one supplier for a majority of systems (hardware or software) products.

Without dealing with any measure of "goodness" here, or with the blessing of any standards bodies, it is clear that Microsoft (on the client) and Novell (on the network) offer the dominant product strategies in each respective area. *That means that the widest range of products and add-ons are available for the wares of these two vendors.* That, in my mind, is the real definition of an open environment for the 1990s.

With the proven UNIX property in its camp, Novell should be in the best position in our industry to compete with Microsoft's upcoming Windows NT. Novell resellers should have an easy time generating fear, uncertainty, and doubt about the yet unproved NT compared to the mature and stable UNIX.

As stated above, I don't see Novell competing with Microsoft's desktop Windows franchise. On the server side, the dual strategies of UNIX and NetWare with their different user bases will offer formidable

competition for Microsoft. There is also one other player in this market, and that's IBM with OS/2. And, Novell's acquisition of USL is **not** good news for the IBM camp:

1. Because Novell will make UNIX a more aggressive product in the commercial market, and that means more competition for IBM's OS/2.
2. Because IBM is on the OSF side of the UNIX wars. It's own AIX and the Mach kernel on which AIX will be redeveloped, have no relation, other than name, to the USL UNIX. Also, Novell's USL acquisition will focus attention towards that UNIX market to insure that Novell is less interested in more joint IBM efforts (such as NetWare for OS/400, etc.).

Microsoft covets Novell's networking franchise, but has failed badly with its strategy of hitting Novell head on (with LAN Manager). Microsoft's logical response was to try an end run and put communications functions into the operating systems it sells. This strategy has a Trojan Horse quality to it—once you have Microsoft Windows, Windows for Workgroups, or Windows NT, their functionalities start to

(continued on next page)

Novell, Unix, and...

(continued from front page)

make NetWare's functionality superfluous. Regardless of the strength of Novell's networking franchise, Microsoft's operating system offensive has the potential of trumping Novell. Novell's strategy has become, "why not forget this *coopetition* (Ray Noorda's word for competing with people by cooperating with them) thing and just go head to head with Microsoft?"

The market that I primarily see Microsoft and Novell battling for is that of server operating systems. Yes, Novell competes in the desktop DOS market with DR DOS, which it acquired with Digital Research, but DR DOS isn't a serious contender for desktop shares—the product has less visibility now than when Digital Research was an independent company. I, frankly, don't understand the logic behind why anyone would buy DR DOS. It seems to me that any technical superiority that it offers over MS DOS is a trivial issue compared to the concerns that would be raised over potential incompatibilities (see side bar).

Conclusion

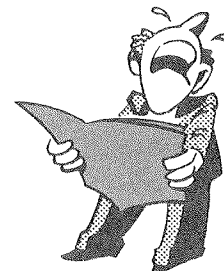
The stock market's initial sell off of Novell stock

was a reaction to the USL acquisition and was reversed within two weeks. The analyst's argument was that at an estimated price of about \$350 million, Novell was overpaying for the acquisition of a company with annual sales of \$80 million and negative cash flow. Four and a half times gross sales is a high price, but if you're paying that price in stock with a high price/earning ratio (P/E), as did Novell, then the payment is made in low-value currency. This acquisition price is just 25% more than Microsoft paid for Fox Software. If Novell and USL follow through on the promises made about this merger, then I would argue that USL's addition to Novell's value is much greater than Fox's potential contribution to Microsoft's. When your growth is aggressive, your stock trades at a high P/E, and one can afford to pay a high price for important acquisitions. There is some risk for Novell with this USL acquisition strategy, but I would argue that it's a tremendously important play that will vault Novell into the big leagues as a rival (in fact the only rival) of Microsoft for computer industry leadership as we move towards the next century.

GS

Editor's Personal Experiences!

I am currently in the process of learning the relatively new Freelance for Windows graphics program from Lotus. The program has some "bugs" or anomalies—in certain import/export circumstances, text font sizes are changed, or portions of text might totally disappear! Lotus acknowledges these problems, but does not know the cause or the solution. It was suggested that I add more RAM to my system (currently 4 MB) in order to *avoid* the bugs. If I were using DR DOS rather than MS DOS, do you think the Lotus technical support people would have used that as a scapegoat? I do. I've also begun to use WinFAX which, even with its new version 3.0 release, won't work with long-distance telephone carrier services such as Sprint or MCI. The Delrina people (WinFAX publishers) are without a clue when it comes to solving user problems, but I'm sure that if I told them that I was using DR DOS that I would be told it was the cause of my problems.



Event Driven Client/Server Development (EDC/SD)

Part I: Introduction

by Jim Davey

The definition of downsizing has evolved over the past year to encompass more than the downscaling of hardware. The processes also entail downsizing the complexity of traditional software development techniques. Today's businesses cannot afford to use traditional mainframe-based development methods. Methodologies such as those supported by the Big Eight (or is it Six?) accounting giants, or government standards such as DOD-STD-2167 require many person-years to develop complex requirements. Often, these years-long projects end-up gathering dust while being housed in three ring binders.

Development environments such as IBM's AD/Cycle with its Repository Manager require expensive mainframe hardware and high priced CASE tools. We can no longer afford or justify these approaches; these expensive

and overly complex artifacts of a past era must be abandoned along with the dinosauric mainframes that led to their existence. Mainframe development has evolved into the creation of "complex solutions for simple problems."

On the other hand, the "just do it" approach that has become popular with PC developers today often results in applications that fall short of what is needed. Often, I have seen applications built in this "just do it" environment that need to be reworked time and time again.

These two problems, the mainframe's "complex solutions for simple problems" and the "just do it" failures, are why new methods must be developed to produce specifications that can be used to rapidly develop and deploy new client/server applications.

Event Driven Client/Server Development (EDC/SD)

With our traditional mainframe model, computer systems have supported business functions and enterprise information requirements. Specification models have typically been based on either functional decomposition or data structure analysis. Most Structured Methodologies were based on functional decomposition while Information Engineering Methodology

was based on data structure analysis. And, the selection of which methodology to use always depended on your perception of the computer—is a computer for performing computations or managing data? Most Methodologies used some combination of function and data, but often separated the process modeling from the data modeling.

With the client/server model, a system of computers now supports the knowledge worker. A new method based on business events rather than functions or data structures needs to be developed. The form of specification must be something end-users can understand. In the past, end-users were expected to learn how to use Data Flow Diagrams and Entity Relationship Diagrams. This seldom worked well; these diagrams are not part of an end-users everyday life. In fact, they were usually only seen when MIS types came around to develop new systems.

In contrast, business events are well understood by end-users. Events can be articulated by users in narrative form either in writing or through the interview process. Then, examples of inputs and outputs can be provided or developed. Simple diagrams using bubbles and arrows can be used to model events or

(continued on next page)

Event Driven...

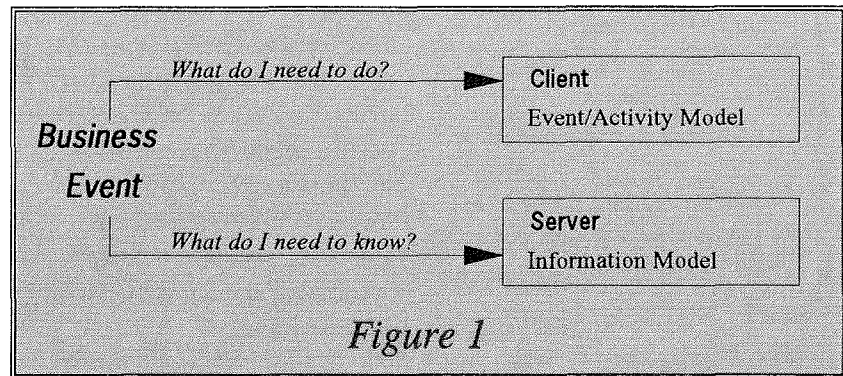
(continued from previous page)

tools using graphical icons and animation could be developed.

The set of events should be used to define the scope of the system, and as events are added, the scope will expand. Both data requirements and processing requirements can be derived from business events. The server requirements that can be defined as an information model are created from the set of events. Client requirements, in the form of user interface and processing requirements, can be defined for each event.

Using this new approach, EDC/SD, you start with a particular user or job title and identify the associated business events. Next, you select those events to be part of the scope of the new system. As more users are studied, you will often find that a single person or job title will perform many roles, or that the same role will be performed by different people or job titles. The defined roles will be the best way to organize events since events tend to be specific to a role. In turn, those roles will be useful for organizing the user interface.

A model of roles and events should be developed to define the scope of the system. For each event, there is a process component, *what*



do I need to do, and an information component, *what do I need to know* (see figure 1).

✪ *What I need to do* can be organized by roles and sequences of events for which activities can be defined. Roles and events are then organized into windows and window objects. Activities can be implemented as event procedures to be executed when a window object is acted upon. And, interactions between the clients and the server can be defined as SQL statements and user views.

✪ *What I need to know* can be expressed as attributes and relationships. These event facts can be used to develop an information model in the form of an entity relationship diagram. Using the information model, a relational database can be developed for the server.

Information modeling

Once the first step using EDC/SD has been performed, data analysis methods, such as Curtice/Jones¹ or NIAM² can be used for in-

formation modeling. These approaches generate an information model from a set of detailed facts and examples—the model and roles developed using EDC/SD. An Entity Relationship model can be developed from the event facts and used to verify the subject matters accuracy. For server development, a fully normalized relational database implementation can also be derived from the event facts.

For client development, user interface and event procedure prototyping seem to work well. Prototyping is most effectively accomplished when the scope of the project is predefined. The set of client roles and business events can be used to define the scope of the project. The information model defines a context for prototype development. It defines the entities or application, defines objects that the clients have in common, and relationships among these entities. With the scope and context defined, the prototyping effort can concentrate on user interface and event procedures.

Summary

This is an overview of a method to develop requirement specifications that can be used to develop client/server applications. This method is based on the use of business events for modeling rather than processing functions or data structures. An information model is derived from the

set of events to be used in the development of the server database. User roles and event sequences are used to define the user interface. Event activities are then used to define event procedures and client/server interaction.

Future articles are planned to define in greater

detail the techniques involved in this approach. *JD*

¹Robert M. Curtice & Paul E. Jones, Jr., **Logical Data Base Design**, VanNostrand Reinhold, New York, 1982.

²G. M. Nijssen & T. A. Halpin, **Conceptual Schema and Relational Database Design: A Fact Oriented Approach**, Prentice Hall, Sydney, 1989.

George's Visit to...

(continued from page 8)

- ODBC is for Windows primarily (nothing in its technology limits it to Windows, but all of the early deliverables are for Windows only) environment, while IDAPI was announced for DOS, Windows, and OS/2 clients.
- ODBC access drivers were announced for SQL databases, while IDAPI was announced as a technol-

ogy that would have drivers for non-relational databases in addition to relational types. For access to non-relational DBMS, the IDAPI CLI will come with navigational extensions.

Watching the evolution of applications as they move toward multimedia and the emergence of these different competing "marketecture standards," it is becoming clear that client PCs will necessarily be huge by

today's standards if for general purposes. We know companies that are already running multiple protocol stacks in their clients because they want interleaved access to different network servers, e.g. NetWare and Vines. With GUIs, multiple database and network drivers, and multimedia applications, a 486 PC with 16 MB RAM and 300 MB of hard disk is going to seem rather sluggish! *GS*

The Last Thing IBM...

(continued from page 6)

wait for Taligent?" I guarantee that its response will be less than satisfying. Networking is even more confusing. How is a customer supposed to decide which networks to buy—LAN Server, NetWare, TCP/IP, or SNA's APPC/APPN? IBM sells all of them and claims that they're all strategic. The situation is no better when it comes to choosing servers. IBM provides precious little

direction to customers who seek assistance in choosing among OS/2, OS/400, VM, MVS, and AIX. One thing is certain, no one currently thinks of IBM as an industry leader when it comes to client/server computing.

IBM has become a massive and unwieldy bureaucracy that they're now trying to modify and change. Essentially, IBM must stop behaving like General Motors and push the company back to where they can be inno-

vative once again. When you look at some of its new offerings, like CID and DRDA, you can see that they're making some real progress. But, IBM is in the midst of a large crisis. They must formulate a coherent strategy that will successfully lead its existing mainframe customers into the next generation of client/server systems and downsizing. *JF*

UPCOMING downsizing Events...

DATABASE WORLD AND CLIENT/SERVER WORLD are once again being held jointly in Boston, June 14-16, 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 delivered by several renowned industry figures including: Chris Date, Michael Stonebraker, George Schussel, Larry DeBoever, and William Zachmann. In addition, Philippe Kahn of Borland is to be the plenary speaker.

A new addition to DCI's WORLD EVENTS is MOBILE WORLD which is being held this year in conjunction with MOBILE COMPUTING EXPO and

PEN-BASED EXPO. MOBILE WORLD, in Boston, March 3-5, 1993, features eight separate conference tracks: Developing/Programming for Pen Conference, Systems Integration Conference, Pen Hardware Technology Conference, Mobile Communications Conference, Keyboard Machines and Personal Devices Conference, Enabling Technologies Conference, Managing Mobile Technologies Conference, The User Stories Track.

Jointly co-sponsored by DCI and DATAMATION is **The Information Technology Summit—Re-engineering The Future**. The focus of this executive summit is on the competitive advantages that can be gained from the newest computing technologies. The Summit is being held at the PGA National Resort & Spa, Palm Beach Gardens, FL, May 17-19, 1993.

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

| | | |
|--|--|--|
| <p><i>Schussel's</i></p> <p>DOWNSIZING JOURNAL</p> <p>February 1993 Volume 2 Issue 6</p> | <p>Editor: Dr. George Schussel Managing Editor: Stacey S. Griffin Subscription rates:</p> <ul style="list-style-type: none"> ● \$199 annually for U.S. residents ● US\$225 annually elsewhere | <p>SDJ is published monthly by: Digital Consulting, Inc. 204 Andover Street Andover, MA 01810 USA 508-470-3870 FAX 508-470-1992</p> |
|--|--|--|

Copyright © by *Schussel's Downsizing Journal*. This newsletter is fully protected by copyright. Reproduction or photocopying, even for internal use, without the publisher's permission is PROHIBITED BY LAW. Violators risk criminal penalties up to \$100,000 in damages. For permission to photocopy for internal use or to inquire about special reprints, contact Stacey Griffin at (508) 470-3870. *Discounted multiple-copy subscriptions are available.*

| | |
|--|--|
| <h2 style="margin: 0;"><i>Schussel's Downsizing Journal</i></h2> <p><input type="checkbox"/> <i>I want to subscribe. Please send me the next issue.</i></p> <p><input type="checkbox"/> <i>Please RENEW my subscription.</i></p> <p><input type="checkbox"/> Check payable to Digital Consulting, Inc. is enclosed</p> <p><input type="checkbox"/> Bill me</p> <p><input type="checkbox"/> MC # _____ Exp. _____</p> <p><input type="checkbox"/> Visa # _____ Exp. _____</p> <p>Subscription Rates</p> <ul style="list-style-type: none"> ● \$199 for one year in the U.S., US\$225 elsewhere ● All amounts payable in U.S. dollars please <p>To Subscribe: <input checked="" type="checkbox"/> Drop a copy of this form in the mail</p> | <h3 style="margin: 0;"><i>Subscription Form</i></h3> <p>_____</p> <p>Name</p> <p>_____</p> <p>Title</p> <p>_____</p> <p>Company</p> <p>_____</p> <p>Address</p> <p>_____</p> <p>_____</p> <p>Telephone</p> <p>☎ Phone 508-470-3870 📠 FAX 508-470-1992</p> |
|--|--|