Schussel's
DOWNSIZING JOURNAL
JANUARY 1993

In This Issue:

George's Visit to IBM, Santa Teresa

This past November, I was afforded the opportunity of spending a day at the famous IBM software development laboratory in Santa Teresa, California (STL). The purpose of the briefing was to bring me up-to-date on a number of software technologies I have been following for the last few years.

STL, one of the principal software development laboratories operated by IBM, has traditionally been, and still is where worldwide, data architecture decisions and plans are developed and managed for IBM. For example, IBM's IMS and DB2 database management systems were developed and are maintained

(continued on page 10)

Downsizing at Motorola for Competitive Advantage

by
William D. Connor
Corporate Vice President
Director of IT, General Systems Sector
Motorola

The title of this article was chosen with specific purpose. The downsizing program, which has been nearly completed by Motorola General Systems Sector, was strongly motivated by our need to maintain competitive leadership in the businesses which compose the Sector. Motorola is a worldwide electronics company with revenues in 1991 of $11,342 million. The General Systems Sector is a major business unit of Motorola

(continued on next page)

Motorola...

(continued from front page)

consisting of three businesses: the Cellular Infrastructure Group, the Cellular Subscriber Group, and the Computer Group. In 1991, the General Systems Sector had revenues of $2,847 million and operating profits of $330 million. The Sector has major manufacturing facilities in the United States and Europe, and employs over 10,000 people.

Information system needs

The business imperatives affect every organization within the Sector. Every person, group, department, or major operation must align itself with the overall business goals if we are to be successful. MIS is no exception. As a result, we have several fundamental goals which are inherent in all MIS planning efforts. First, to be responsive to our fast changing environment, we need to significantly reduce the cycle time required to provide, adapt, and replace information systems. The one to two year development cycles which characterize the mainframe computing world are simply impossible to live with in our competitive environment. By the time a system is designed, developed, and installed, the new system is likely be out-of-date in some important aspect. Dramatically shorter order times were an absolute requirement.

It is mandatory that MIS contribute as much as it can to the enormous cost reduction efforts which are required to support businesses with dramatic and continual price reductions. This is not just an issue of reduced product cost. This is a requirement across the entire organization in the way we serve our customers. Therefore, MIS must be a leader in reducing operating costs, both in the operating departments and within the MIS department in particular. In addition, we felt that MIS must avoid all proprietary hardware and software solutions, no matter how attractive they appeared at the time. Every proprietary product carries a hidden tax which must be paid in time. We wanted to eliminate encumbrances despite any inviting circumstances for the short run. At Motorola, we also needed to place the highest emphasis on the quality of application software, computer availability, and network performance; we know very well that quality is not free but in fact is the only way to achieve low costs over the long run.

Downsizing benefits

In order to meet our business requirements, we knew we must depart from mainframe data processing. There were several reasons which drove us to this decision:

- The price/performance advantage of silicon computers (microprocessor-based products).
- The competitive benefits resulting from a move to open systems computing.
- The emergence of Unix from the engineering laboratory environment into the world of business.

Basic strategies

In building our downsizing strategy, there were several facets important to Motorola's overall success. For servers, we decided on multi-user Unix micros since they provide the price/performance advantage of silicon computers. The Unix micros gave us access to the rapidly emerging body of Unix application software. We wanted to avoid networked PCs and their concomitant data integrity, control, and security problems. Closely related to our choice of Unix environment was the decision to use X terminals instead of workstations or PCs as the desktop device. The major expense of the desktop is in the display; the use of X terminals reduces the rate of obsolescence significantly. It was not necessary to update the desktop device for every upgrade in processors since X

software can be maintained at a constant revision level.

For applications to be developed internally, it was important to achieve shorter cycle times, improved quality, and greater productivity. We decided to employ a CASE development environment to the greatest extent practical. We used 4GL by Informix to reduce the amount of code written. We added ancillary tools such as report writers, screen generators, file updaters, etc., as they became available. We intend to employ IDE's Software through Pictures to aid the system design process.

While we opted not to use PCs as desktop displays, we did want to provide our users with the same elements of personal involvement and control PCs have introduced to the non-MIS world. Therefore, our plan called for providing tools that permit user-initiated queries and report generation without the intervention (or obstruction) of MIS. These tools typically provide limited ability to generate simple applications to manipulate the data. We have solved the data corruption problem by providing daily copies of our selected business files for use in local report generation. If the files are inadvertently corrupted, it doesn't matter since they will be replaced by the next day's copy.

Motorola's plan of action

Our action plan was based upon several key concepts. The first and most important was to establish strong, clear objectives. We decided that we were going to completely eliminate all mainframe processing. Nothing would be left behind. At first, this may seem unnecessarily severe; some people felt that certain applications might be better left on the mainframe. We knew, however, that unless our objective was simple and uncompromising, we would spend endless hours debating the future of each application. As experience has shown, we have not found a single application that should have remained on the mainframe.

Another key concept was the maintenance of a broad band access to application software. We knew that the real challenge in moving to purchased application software was to find application packages that closely approximated our actual or intended business practices. While the number of applications offered under Unix is growing rapidly, we wanted to remain as open as possible to all available products (assuming that the products are not proprietary). Therefore, we did not standardize on one database for our applications. Instead, we have developed and/or purchased applications for Informix, Oracle, Progress, Sybase, and Unify, and have also purchased two products which incorporate internal file structures.

When we purchase an application, we look to the supplier of that product to resolve any application problems caused by the database software. Our support staff handles all databases with only a modest effort. Applications developed internally are all running on the Informix database due to our preference for the Informix CASE tools and the excellent performance of the database engine.

When we started our downsizing effort we did not build a staff of people in parallel to our normal MIS organization to do this work. After all, our key objective was to reduce cost, not to increase it. We simply started using our staff on one application at a time, and have continued this approach ever since. This allowed us to train a small team which was to grow and gain experience in manageable increments.

In sequencing application work, the key concept was to achieve immediate cost payback. We chose applications which were good fits and easy to migrate, that once relocated would save substantial costs and be brought

(continued on next page)

Motorola...

(continued from previous page)

up-to-date to current business practices. We were able to achieve immediate cost payback by starting with applications running on corporate data centers which function as service bureaus for Motorola. We informed the corporate MIS organization of our efforts so they could implement mainframe capacity adjustments accordingly. If we had been operating our own data centers as many companies our size do, we would have started our downsizing effort by outsourcing our computer operations. Although this step would have resulted in some short term increase in cost, it would have transformed our processing costs from fixed to variable–a necessary step in the downsizing effort if early cost-savings are a goal.

In choosing how to transition an application to the Unix environment, we had three choices: migrate the application using available tools, rewrite the application with CASE tools, or purchase an application to replace the existing application. These three alternatives are commonly referred to as: reverse engineering, re-engineering, and replacement. Our mainframe applications ranged from five to twenty years in age, the oldest software being practically unmaintainable. Most of our other systems were showing increasing mismatch with our current ways of running the business. We decided that our primary approach to downsizing would be to replace existing software with packages purchased from outside software vendors. Consequently, we undertook a comprehensive search for commercial replacements. We wanted to avoid special custom versions because while customization would solve some of our problems, it would result in Motorola's paying 100% of support costs. In addition, software suppliers have tremendous difficulty in maintaining a large number of product variations.

We found one software supplier, FourGen, who has developed an object oriented application approach. The application is developed in object modules in which the basic functionality is self-contained with simple and constant interfaces between modules. This permits addition or modification of individual modules without creating an entire custom version. It even allows new releases of the basic package without destroying the linking of the custom modules that have been inserted.

Approach taken and current progress

We are currently three-quarters of the way through with our downsizing effort. By the end of next year, all of our mainframe computing at the Motorola General Systems Sector will have been eliminated. We have found no applications in our environment that should remain on the mainframe. All of our financial systems are being replaced by purchased, off-the-shelf applications. These applications are from four different vendors, yet we close our books in two working days after the end of each month. This includes closing each of the general ledgers in every country and consolidating into a General Systems Sector general ledger. The only weak link currently remaining in this cycle, is the mainframe general ledger which will be replaced in the next several months.

In the sales area, the new Unix systems consist of both purchased replacements and internal developments. The Cellular Subscriber Group developed their order entry system because they had several unique requirements. The other two groups at Motorola General Systems Sector purchased off-the-shelf order entry systems.

Key issues concerning downsizing vendors

There are several key issues we encountered in our downsizing effort which deserve special discussion. First is the software vendor selection and management. Pur-

chasing organizations in large companies are experienced in the selection and qualification of vendors. MIS organizations are not. MIS has typically relied upon their mainframe vendor in addition to a small number of mainframe software suppliers. In the Unix world, the scene is considerably different. We now deal with over 50 applications software suppliers. One of MIS's principals roles is to identify software products which can be evaluated as candidate applications for the operational departments. This is a very important job.

This qualification process can only be undertaken by someone capable of assessing the company's technical competence as well as their financial and organizational strengths and weaknesses. This qualifier should first examine the vendor's approach to quality. There needs to be a strong commitment to quality and quality processes/metrics if the product is to be successfully installed. Many Unix vendors are quite small and may need help in developing quality programs. Another significant factor to consider is the ability of the vendor to customize products. It was discussed earlier that we strongly prefer not to make large scale modifications. And indeed, many vendors will not modify their products except for minor reporting changes and data entry layouts.

The vendor's long term viability is another serious issue to be considered. Considerations such as the vendor's financial strength, additional capital availability, management motivation and goals, and sales growth history all must be considered. Provisions for the source code to be placed in some form of escrow should be considered. Invocation of escrow instructions should be conditional not only in case of bankruptcy, but also if controlling interest in the vendor company passes to outside hands.

What happens to the people?

At Motorola, we spent a great deal of time considering the training requirements for MIS personnel as we moved from mainframe COBOL programming to Informix 4GL development and Unix. Most of our employees were very apprehensive of the changeover. Two people elected to leave the company rather than lose their professional affiliation with the mainframe world. We were also aware that there has been precedence that new technology often resulted in large dropout rates among data processing personnel. We wanted to avoid this, if possible. The first step we took was to make all necessary training available to our MIS people. However, it was not necessary to teach everyone the intricacies of Unix any more than it is for everyone in the mainframe environment to know the operating system software. The training programs were highly successful. Today, the only morale problems relate to the handful of programmers still supporting the remaining mainframe applications. They feel that they are being left behind. They fret that they are not learning the necessary new skills. They also all realize that the professional skills in great demand today are in Unix programming.

We also paid a good deal of attention to the training of end-users. We wanted to ensure that the benefits of a Unix environment were enjoyed. We now have a tremendous support in the user community.

A different key issue in downsizing involves the need for willingness on the part of your company's operating departments to accommodate adjustments to purchased applications software. Certainly, the tail should not wag the dog. The business should not be expected to significantly change its practices to accommodate the software. However, remember that the

(continued on next page)

Motorola...

(continued from previous page)

savings benefits achieved with purchases software are worth some operational accommodation.

Accomplishments to date

We are not finished in our downsizing program, but our accomplishments to date have been very satisfying. The first major goal we set out to accomplish–to dramatically shorten our development cycle for responsiveness to the fast nature of our businesses–has been successfully accomplished. System development which once took one to two years on the mainframe are now being accomplished in four to six months. As a result, not only our systems more flexible to the necessary changes, but our applications are both more current and maintainable.

The quality of our software has dramatically improved. The purchased software is well documented with good training materials. The code is typically well-designed and well-written since we require all of our vendors to assure us of their top-level quality products. There is nothing like competition to ensure that discipline. At the same time, our internally developed software is substantially better. Through the use of CASE tools, we are designing and writing higher quality systems. For us, these quality improvements directly translate into lower cost and higher user satisfaction.

The last and most important accomplishment-to-date has been the tremendous reduction in our business computing costs. We recently performed a study comparing our total data processing costs against several benchmark companies of similar size in related fields. We included the hardware and software costs of all mainframes, distributed Unix computers, workstations, and personal computers involved in business computing. We added all costs of new system development, support of existing applications, computer operations, system administration, and network management. Also included were all data communication network costs. In short, all business computing costs were included. We found that Motorola was running at a level of 1.4% of sales compared to more than 3% of sales for the benchmarked mainframe installations. (Note: These percentages are only meaningful for large manufacturing companies. Each industry will have significant variations here.) Furthermore, as we complete our downsizing in 1993 with the installation of improved system administration and network management tools, we fully expect to have costs less than 1% of sales. With the significant sales growth being experienced by the General Systems Sector, this will result in a savings by 1993 of $90 million per year.

...We decided that we were going to completely eliminate all mainframe processing....As experience has shown, we have not found a single application that should have remained on the mainframe....

At Motorola, we do not need any convincing regarding the benefits of downsizing into the Unix world. We have seen problems in the transition, but they were minimal and well-managed. We have also witnessed improved business operations through better software and more responsive MIS support. At the same time, we have managed to cut our computing costs in half! Our only regret is that we waited so long. *WC*

DATABASE WORLD and CLIENT/SERVER WORLD, Chicago 1992

McCormick Place on Chicago's south side was the venue for this winter's DATABASE WORLD and CLIENT/SERVER WORLD Conferences and Exposition. There were a large number of newsworthy presentations and announcements that highlighted the most successful database-oriented event I've ever attended. An EDA/SQL exposition floor demonstration, comprised of over 40 exhibitor participants, served to illustrate that the era of enterprise-wide database connectivity is now upon us. In mid-December, the trade press coverage of this show was ample. For example, lead cover story of the December 14, 1992 issue of ***Computerworld*** was about the various announcements IBM made at DATABASE WORLD.

Because of the many meetings and activities that go on at DATABASE WORLD, I don't often have the opportunity to listen to gurus such as Chris Gane and Nat Goodman. But, this time I made up my mind not to miss two presentations by Mike Stonebraker and Chris Date.

Stonebraker's "Cold Turkey"

Stonebraker is one of my all time favorite database teachers and consultants. While teaching at UC Berkeley, he has trained some of the industry leaders. He also had a large hand in founding Ingres. Now, Stonebraker has founded a new company, Miro Systems of Emeryville, CA, to bring to the commercial market the POSTGRES work his Berkeley group has accomplished over the last few years.

At the show, Stonebraker talked about the problems in migrating legacy applications into modern relational or object oriented DBMS. The problem is that major industries have spent many millions of dollars (each) on building production systems that are locked into proprietary hierarchical DBMS or file systems running on mainframes. These systems can be very efficient for the purposes initially intended, but are horribly inflexible and a nightmare to maintain. Expanding their scope to add modern capabilities such as multimedia is only a distant dream for the owners of such applications.

The most obvious approach to migrating into the modern, distributed, downsized, flexible, and multimedia world, is to redevelop and redeploy old applications. When the replacement application is ready, then, the process is to convert the older data in a batch process to be cut on the new system. This is a process that Stonebraker calls *"Cold Turkey."* The problem is that in large, real time production systems, *Cold Turkey* fails more often than it succeeds. Failure means that you spend several millions of dollars building the replacement, but fail to have it operational as a replacement for the original target. *(Editor's note: IBM understands this dilemma well and has spent enormous sums of money in developing dual database and propagation approaches that allow users to have multiple copies of data in different DBMS with synchronous control being handled by two phase commit technology.)*

Based on his consulting work in this area, Stonebraker has concluded that it is necessary to develop an incremental strategy in migrating such applications. His presentation covered

(continued on next page)

DATABASE WORLD...

(continued from previous page)

several such strategies, one of which I will detail below. However, the bottom line with all of these is an understanding that large, integrated systems simply cannot be replaced in one fell swoop with today's technologies. If the original systems developers used modular logic, and therefore, pieces of the system can be pulled away from others one at a time, then the replacement process will be tremendously easier. Stonebraker recommended doing as much pulling off of such pieces as possible. He also went on to recommend that each individual user be viewed as a module so that cutover to the new environment be planned 1) by module and 2) by individual user.

There is not much question that such a modular, phased cutover will be more expensive than a one time total system replacement. But, since the *Cold Turkey* approach fails so often (and nothing is more expensive than total failure) this methodology should be taken into consideration when building a conversion plan.

Stonebraker's first step in building a modular conversion plan is illustrated in *Figure 1.* The idea illustrated here is that in moving a character based hierarchical or flat file application to the new GUI world with RDBMS, an emulator should be built that allows the old logic and procedures to continue to operate on top of data that has already been migrated forward into the RDBMS. Such a technique allows new capabilities to be built on top of the new DBMS while continuing the older operation. The cost here is of building the DML (data manipulation language) emulator for the older applications to be able to continue operations.

Once this phase is under control, the application logic for a single module can be rewritten into its new form. This is illustrated by the GUI application running at the left side of *Figure 2.* That application is running correctly while the older applications and their users continue to operate with their old procedures (but against the new DBMS) as shown at the right side of *Figure 2.*

Once this type of architecture is in place, the entire system can be migrated forward in the most modular of approaches– where one looks at each application and each individual, and migrates the system piece by piece.

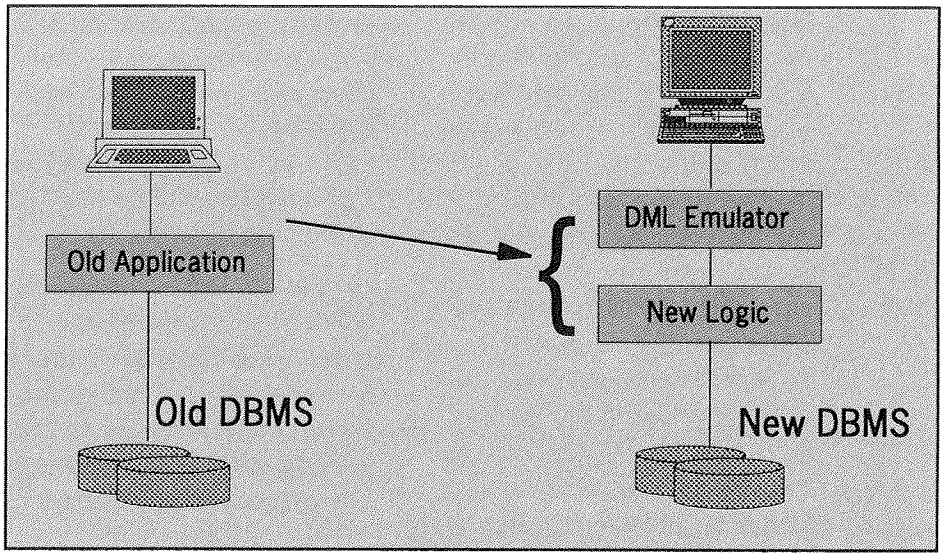


Figure 1

Date's relations

Chris Date is one of the best known educators in the field of relational database. His many textbooks are standard college fare for this subject all over the world. Date's lecture style is challenging, but superbly clear. If one needs a *real* understanding of rela-

tional subjects, then listening to Date is a must.

Date, as expected, talked about the continuing and ever increasing importance and relevancy of the relational model. He started by commenting on the new conventional wisdom that the relational model has become unimportant in an era of object orientation and multimedia. Disagreeing strongly, Date set out to prove that the relational model was the best underpinning for migrating forward into these new technology areas. He also strongly criticized the SQL language in its current formulation as "failing egregiously" in many important requirements. Points that Date stressed were:

☑ Key capabilities of relational algebra include the fact that the system supports closure. This means that the result of a relational operation on sets is a set. It in turn can be operated on further using the same algebra. Closure is a property of most mathematics and offers many benefits.

☑ Every relation has both a heading and body. The heading includes column names. Many SQL systems don't provide information on the column headings. These are necessary and should be included in information managed and displayed by SQL.

☑ SQL is not able to recognize that different SQL syntax may be identical in semantics. *(Editor's note: This is a well known failure of SQL systems that intelligent software optimizers may partly overcome.)* If SQL were to support automated key inheritance in derived views, then SQL systems would be able to recognize semantic equivalents of differing syntax.

☑ Nulls continue to be an important area of interest and research. Nulls represent the absence of knowledge (definitely not 0, which is a known quantity). There have been proposals for multi-value nulls, the most prevalent at this time being a three valued null. Date stated that three valued nulls fail catastrophically. First of all, the existing products that he has seen supporting three valued nulls are buggy and return incorrect results. Secondly, the whole three value system, even if working properly, returns answers that are intuitively wrong. He suggested that the best current solution for missing information is to assign default values.

☑ Relational domains (data types) are the exact equivalent of object classes. Date claimed that a proper implementation of domains in relational systems would expand functionality to include objects.

(continued on page 12)

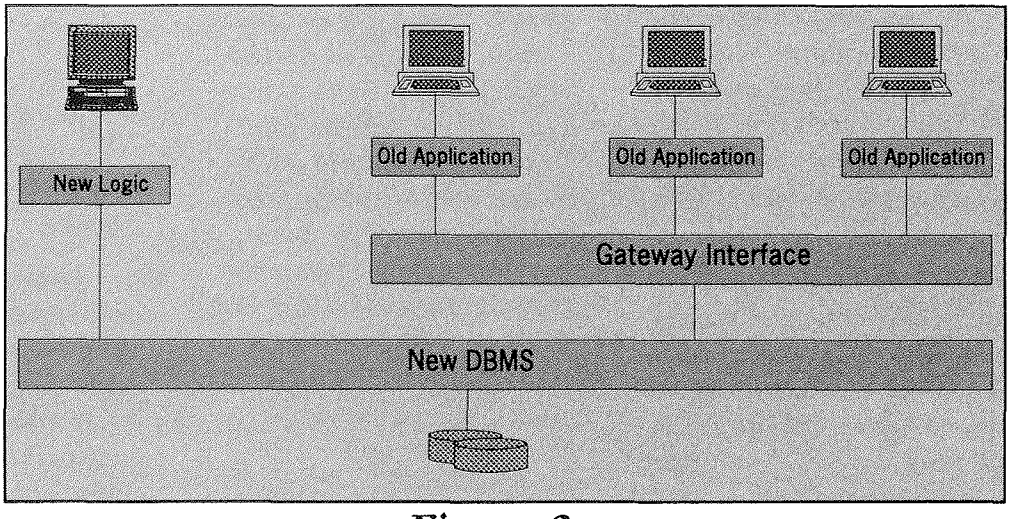


Figure 2

George's Visit to IBM...

(continued from front page)

here, as is the planning and architecture for DRDA (Distributed Relational Data Architecture). While some of IBM's recent products and initiatives haven't been universally acclaimed as world-class, that same statement is definitely not accurate in regards to IBM's database and data management products.

- The first and most popular hierarchical DBMS in the world is IBM's IMS. There is more live production data in IMS than in any other DBMS.
- The relational model for DBMS was developed by Dr. E. F. Codd when he was at IBM.
- The SQL relational DBMS language was developed at IBM.
- IBM's System R* team, in the 1970s developed many of the concepts which defined distributed relational processing as it would be implemented in the 1990s.

The above list could be extended, but I believe that my point is made. Data management software is definitely one technological area in which IBM's efforts are considered top-notch, and where competitors fear and respect IBM's efforts.

Of course, when one company leads in software development, they are bound to also have failures. IBM at STL certainly has had its share. Most recently at STL, the AD/Cycle initiative along with Repository Manager/MVS was planned and developed. Certainly, some aspects of these products could be considered failures (more on this later). A few years ago, a major project to develop a general purpose SQL front-end to the IMS database manager was abandoned after a major effort in building code. The effort was stopped because it was believed that a product of this type could not be both general and robust enough to be widely successful.

Any product initiative IBM attempts in the data management area becomes ultimately judged by its acceptance among IBM's largest customers which includes firms such as Boeing, General Motors, and some of the largest North American banks. For these customers, products must be engineered with adequate security, integrity, documentation, and functionality (robustness) so that the most sophisticated users will find it comfortable. The need for a high level of functionality and robustness means that, often, IBM's software initiatives are late (compared with competitors) or ponderous. Accordingly, IBM is criticized by the press and public, but its largest customers who are often the most loyal, insist on such high levels of functionality and performance. IBM chooses to meet the needs of those customers.

In this article, I will convey the essence of several discussions that were held about topics of interest to the software development community. Those topics are:

1. The new mainframe (hardware) architecture.
2. The AIX relational database management system.
3. The Information Warehouse.
4. Distributed SQL functionality over the DRDA architecture.
5. IDAPI (Integrated Database Application Programming Interface).
6. AD/Platform.
7. IBM's directions for object database support.

The New Mainframe Architecture

I have never been able to understand why the 370/390 architecture hasn't been re-implemented with microprocessor logic. Such a move, if it reduced costs of these processors, would appear to offer a broadening of

the market as a result. Over lunch, Don Haderle, an IBM Fellow, and I talked about the fact that just such a new hardware architecture is due from IBM. When? I'll tell you that later.

Haderle explained that the essential element in moving to microprocessor implementation is changing the logic from bi-polar to CMOS. Historically, IBM mainframes have been built from bi-polar logic, which is fast but runs hot and is expensive in terms of today's microprocessor CMOS circuitry.

I was told that the 370/390 instruction set is in fact already running as a microprocessor implementation, a four chip set. There's a very large cost savings in pushing such a design down to a single chip implementation: the elimination of the connectors and connector logic, along with the associated propagation delays. A four chip microprocessor implementation might offer a two-to-one price/performance improvement over bi-polar logic design costs. However, a single chip implementation would reduce costs to the point where they are improved by a factor of six-to-one, a compelling reason to move to the new circuit logic.

There's a very simple reason why IBM hasn't made the microprocessor move before now. It has to do with the power necessary to run MVS, IBM's crown jewel. This high end operating system takes about 2 MIPS (IBM mainframe MIPS, not microprocessor MIPS) when just sitting, doing nothing. Before now, the CMOS micro 370 implementations were generating no more than 3 MIPS while idle. The 1 MIPS left over for useful work made it appear to not be worthwhile.

Haderle felt that this implementation would reach 6 MIPS in the near future–about the same time that IBM's ability to implement the logic as a single chip set reached reality. As a 6 MIPS chip, the 4 mainframe MIPS left over would be a reasonable solution for customers. It's expected that this mainframe would be sold as a tightly coupled (SMP shared memory) six-way processor, an architecture already common for ES/9000's. That might be the entry module. Haderle talked about using multiprocessor versions of that basic module (loosely coupled, separate memories) in versions that would have up to 30 or so, basic modules. At that level one begins talking about BIPS (Billion Instructions per Second) not MIPS.

Evidently enough progress has been made on this implementation approach that software engineers like Haderle are already looking at the implementation issues. He thought that large database problems would be solvable through software technique extensions to the hardware. Haderle expects these new super-micro mainframes to have parallel I/O along with huge memories. In such an environment, full scan searches of terabyte databases might be considered a reasonable and useful approach.

A radical reduction in mainframe pricing along with much more power and the existing software base would go a long way to extending and perhaps expanding the mainframe's life. There's no question that there exist many applications that are far beyond the power of current hardware/software techniques. If IBM can deliver on the topics that we discussed over lunch, it's going to make the mainframe world much more interesting. Once the mainframe becomes a micro, even *Schussel's Downsizing Journal* will write about it!

When is all this going to be available for customers? Well the engineers don't make marketing decisions, but the discussions about this technology were clearly

(continued on next page)

George's Visit to IBM...

(continued from previous page)

not about distant futures. I got the impression that two years out would be a reasonable expectation for availability.

The infamous AIX DBMS

When I think about IBM's AIX relational DBMS, the thought of Elvis comes to mind. Loads of sightings, but no real confirmation of current existence. IBM has been very successful with its RS/6000 RISC-based microprocessor line (running AIX, IBM's version of UNIX). Price/performance value ratios on this system compare very favorably with the AS/400 mini-computer line that is IBM proprietary. An important component of the RS/6000 becoming more successful is a proper choice of software for business applications. Recently, IBM has begun delivering CICS for the RS/6000, but the relational DBMS that has been openly discussed and promised for this product line is still nowhere to be seen.

While at STL, I asked, "What's going on here?" AIX DBMS had been under development by IBM's Personal Systems Division at the Austin Texas development laboratory. It seems that the approach of the Austin development was towards a *point solution*–a product that focuses on the market for stand-alone RS systems. IBM STL, on the other hand, wanted an approach that would fit in with distributed and enterprise-wide strategies. It seems that the dispute was resolved by 1) moving responsibility for this product to IBM's Programming Systems Division (headed by Earl Wheeler), 2) by assigning architecture responsibility to STL, and finally 3) by assigning the work to Toronto, where the SQL/DS system is built.

The staffers I spoke to thought that the first version of this product should be out in 1993, and that it will stress compatibility with DB2. I was warned that the functionality of this first release will be limited when compared with more mature products, such as Oracle and Sybase, that are already in the market. As is typical for IBM products, it will focus on performance, reliability, availability, and robustness. The second release of the product will add considerably more functionality. *GS*

This is the first in a two article series on IBM by Editor Schussel. Detailed in next month's issue will be the Information Warehouse, distributed SQL functionality over the DRDA architecture, IDAPI (Integrated Database Application Programming Interface), AD/Platform, and IBM's directions for object database support.

Database World...

(continued from page 9)

One of Date's favorite topics is Nullology. He stated that this is not the study of nulls, but rather the study of nothing at all! As Date made clear, Nullology concerns boundary conditions of the relational model; in other words, sets that consist of one or no objects. This, he defined, was the algebraic equivalence of the numbers one and zero. Testing application systems should always include a step where the logic of the system is tested on the null set. It would be helpful to test SQL assertions on tables with no columns. The problem is that SQL doesn't support such creations.

Date explains the relational model better than anyone else. If you missed him at Chicago, he will be teaching a full day seminar on the subject at DCI's DATABASE WORLD in Boston, June 29–July 1, 1993. *GS*

What's Wrong with Borland?

Subtitled: What's Right with Microsoft?

As I write this in the middle of December, Borland stock is trading at 20, a drop of over 75% from its high of 86 in 1991. Why has there been such a dramatic drop? The answer is that Microsoft has introduced its Windows ACCESS PC DBMS at a $99 price point.

While reviews have been mixed, most analysts believe that ACCESS is a good value (this is assuming that Microsoft will clean up most of ACCESS's apparent problems within a short time period). Since PC DBMS products generally sell for around $500, the Microsoft PC DBMS attack should take a serious chunk out of Borland sales. In fact, Microsoft spokespeople have reported of their inability to keep the supply equal to the demand for ACCESS, and that they hope to clear the order backlog by late January 1993.

One problem of Borland's is, of course, the lack of product. Borland has nothing to compete with ACCESS–at any price level. Paradox for Windows and dBASE for Windows are not yet available, although Paradox for Windows ship dates are being projected for the beginning of 1993. The most optimistic estimate that I've heard for a dBASE for Windows delivery date is next summer. At an analyst's briefing in October of 1991, Borland's president, Philippe Kahn, arrived on his motorcycle to give a demonstration of Quattro Pro for Windows. Kahn told the analysts that Quattro Pro for Windows and Paradox for Windows would ship in the first quarter of 1992, and following shortly would be dBASE for Windows. With his forecast, Kahn displayed arrogance in both attitude and content. And when you display attitude, you have to deliver or else suffer the consequences.

...According to Kahn, Borland's commitment to object oriented technology gives them superior development capability. If object oriented technology is so good, why hasn't Borland been able to deliver products to market sooner than rivals?...

The problem, of course, is Borland's technology. According to Kahn, Borland's commitment to object oriented technology gives them superior development capability. If object oriented technology is so good, why hasn't Borland been able to deliver products to market sooner than rivals?

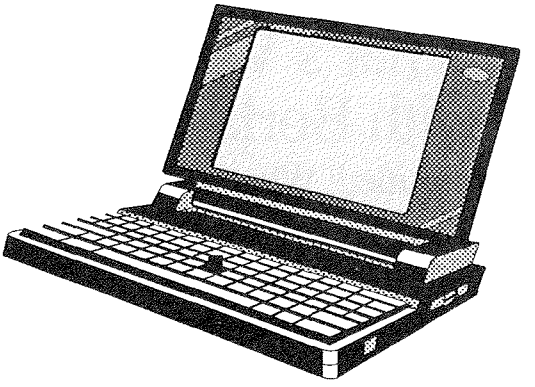
The problem, of course, is Borland's marketing. Borland, the direct mail king, doesn't understand how to convey its message to corporations. Yesterday, I returned from the December DATABASE WORLD Conference and Expo in Chicago, and Borland wasn't there to talk to the 10,000 database developers who attended. Microsoft, IBM, Sybase, Gupta, DataEase, Data Boss, Computer Associates, Ingres, Oracle, Informix, and *every other* DBMS supplier, however, were there. This is not just a solitary example; Borland seems to understand the PC business, but hasn't shown any insight into marketing for enterprise solutions.

(continued on page 15)

Laptops–The Beat Goes On...

Well, I finally placed an order for the machine which is to replace my Toshiba 2200SX. It was DATABASE WORLD and CLIENT/SERVER WORLD in Chicago that motivated me to place that momentous call. At the shows, I used my Toshiba 2200SX notebook for my keynote presentation on database technologies. The presentation was projected directly from the computer onto several video screens in two separate rooms. Actually, the use of such technology proved useful as we had to spill the overflow conference attendees into a physically remote room.

My presentation consisted of two DOS Freelance (from Lotus) screen shows running concurrently with a database application (written in DataEase Express) with OLE links into Excel. Everything was running under Windows for the task switching capabilities. With only 4 MB of RAM memory and a 386SX/20 MHz processor, the strain on the processor was visible. As I thought about my future keynotes, it became clear that my use of screen shows, multimedia, windows 4GLs, and database demos would require a significantly larger amount of both memory and power than the 2200SX possesses.



As I surveyed the portable PC market this December, I felt that the leading notebooks on the market were (this is my opinion, of course) the IBM Thinkpad 700C and the Compaq Lite LTE 486/25C (they need a shorter name for this machine!). Both machines have x486 processors with Intel's SL (power management) capabilities. Both have active color matrix screens, nickel hydride battery technology, and embedded trackballs–the IBM's is actually more of a track button, but it certainly works just as well. I heard from vendors and resellers that both machines were back-ordered by two months, and were currently selling for prices around $4,300 (with base configurations). Interestingly, margins in this business seem to be narrowing as the spread on prices from dealers and mail order suppliers for these machines never exceeded $100 or 2%.

...Why are there so many dual page, full color advertising spreads for the Thinkpad while the production capacity is not there?...

I decided to go ahead and purchase the Compaq. The major reason for this decision was that when I inquired about buying the Windows driver for the IBM embedded track button, not one dealer knew *anything* about how I could obtain the driver. On the other hand, the Compaq machine is delivered with Windows 3.1 pre-loaded with all of the necessary drivers. In addition, it seems that the back-order for the IBM machine is worse than that for the Compaq. This caused me to wonder, why are there so many dual page full color advertising

spreads for the Thinkpad while the production capacity is not there?

I decided not hold back any of the niceties with my Compaq order, and popped for the 209 MB hard drive (not available on the IBM, which has a maximum of 120 MB) and 12 MB of RAM memory. I knew that my dream machine wasn't going to be cheap, and I was right–it cost $4,800. But it seems to me that the PC I always want, when I want it, costs around $5,000 (introductory price).

Having lived with Toshiba products for the last three years, I have reached some conclusions. First of all, there was a noticeable quality drop from the 1989 generation T1600-style of notebooks to the 1992 generation T2200. Several of DCI's Toshiba T2200 machines have had motherboard quality problems and have needed several service calls. The most serious problems encountered to date have involved the PC's external video driver capacity. Toshiba's engineers aren't doing quality work in this area.

In the new product arena, I think that Compaq and IBM are eating Toshiba's lunch. One of the most annoying features of traveling with PCs is the necessity of using mice or trackballs. Apple solved this problem with their embedded keyboard trackballs. Compaq and IBM have both developed similar, satisfying solutions. And, Toshiba's products just don't have the added feature of the embedded trackball.

I am hoping for delivery of my new computer around the middle of January which will give me enough time to put together an EDA/SQL database connectivity demonstration for DOWNSIZING EXPO in Chicago, February 16-18, 1993. Stay-tuned for the next chapter of *Living with Laptops!* GS

What's Wrong with...

(continued from page 13)

The problem, of course, is Borland's product positioning. Borland senior executives were effusive over the acquisition of Interbase as part of the Ashton-Tate purchase. Interbase is a fine example of distributed database/UNIX technology. But Borland hasn't been able to explain to corporate developers (the only reasonable constituency for a product like Interbase) the unique capabilities of Interbase and how it fits into existing corporate legacy DBMS including IMS, DB2, IDMS or corporate standards such as Oracle and Informix. Borland also isn't getting its message across on the role that a combined Interbase/Paradox/dBASE capability could play well within the corporate development structure.

The problem, of course, is Borland management's arrogance. The majority of Ashton-Tate employees were terminated very shortly after the acquisition was completed. The story was that Borland's technology was superior and the work Ashton-Tate employees had done on dBASE wouldn't be relevant for the Windows version. Maybe this is true, but most of the Ashton-Tate marketing personnel I knew were also terminated. Certainly Borland has proven themselves to be less adept at marketing to corporate accounts than Ashton-Tate was. With the announcement of layoffs at the beginning of December, Borland was into their third reorganization this year. Former employees talk about the immaturity of the decision making process inside of Borland–decision making that might be well-suited for a small company, not one with $500 million in annual sales. Based on an

(continued on back cover)

What's Wrong with Borland...

(continued from previous page)

outsider's view of their marketing and product deliveries over the last year, one would have to concur that radical changes in management are necessary or Borland isn't going to survive.

I expect that the stock market has sent Borland a wake-up call. I hope it has because most people don't want more of a Microsoft hegemony than already exists. In order to succeed, Borland is going to need to:

1. Deliver good Windows DBMS products to the marketplace. This shouldn't be a problem since Borland's products generally have an excellent reputation.
2. Deliver that product soon. Now we really get to see how outstanding the Borland object oriented technology base is.
3. Gain maturity in the building of marketing relationships. This will require a change in management and outlook.

I, for one, hope that Borland succeeds. *GS*

Schussel's

DOWNSIZING JOURNAL

January 1993
Volume 2 Issue 5

SDJ is published monthly by:
Digital Consulting, Inc.
204 Andover Street, Andover, MA 01810 USA
(508) 470-3870 FAX (508) 470-1992

Editor: Dr. George Schussel
Managing Editor: Stacey S. Griffin

Subscription rates:

$199 annually for U.S. residents
US$225 annually for foreign addresses

UPCOMING **downsizing** Events...

DOWNSIZING EXPO, being held in Chicago, February 16-18, 1992, will be co-chaired by ***Schussel's Downsizing Journal*** Editor George Schussel, and Larry DeBoever of Tucker/DeBoever Technologies. The conference portion has been expanded to include: The Downsizing Conference, The Interoperability Conference, The Business Re-Engineering Conference. There will be over 100 exhibitors on the exposition floor, as well as several dozen renowned conference keynote speakers including: William Zachmann, John Soyring, and Esther Dyson.

Being offered in both January and February are two of DCI's most popular downsizing seminars: **Cheryl Currid: Managing Downsizing** and Herbert Edelstein's **Implementing Client/Server Applications and Distributing Data**. Currid will be demonstrating how to assess your company for the proper implementation of downsized systems in Washington D.C., February 24-15, 1993. Edelstein's course on the pragmatic "how-to's" involved with client/server systems and distributed data will be in Washington D.C., February 22-23, 1993.

Richard **Finkelstein's Practical Guide to Client/Server DBMS Computing**, in San Francisco, February 23-24 1993, is an in-depth study of the tools and techniques that are necessary in implementing a successful client/server application. Finkelstein will discuss the features of several popular database servers including: Microsoft/Sybase SQL Server, Oracle, IBM Database Manager, SQLBase, Ingres, Informix, Interbase, Informix, NetWare SQL, and XDB.

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

For your enjoyment and ease of reference, like a chameleon, **SDJ** will be changing its colors monthly. Our changing color scheme will make it easy to spot each new issue at a glance, and quick to reference past issues.