Alex. Brown & Sons
Incorporated

Research

Communications &
Information Systems Group

Joseph Payne
(301) 783-3271

August 1989

THE COMMERCIAL COMPUTER SYSTEMS INDUSTRY

On-Line Transaction Processing
In Transition to
On-Line Enterprise Computing

The Battle Lines Are Forming
For The
Next Megamarket

Tandem Computers, Inc.
International Business Machines, Inc.
Digital Equipment Corp.

THE COMMERCIAL COMPUTER SYSTEMS INDUSTRY

TABLE OF CONTENTS

INVESTMENT THESIS

Background

Three of the most visible companies in the commercial systems industry, Tandem Computers, Inc., International Business Machines, Inc. and Digital Equipment Corporation, are jockeying for position at the vanguard of the "battle lines" forming over the next megamarket in computer technology. What has been a niche market for processing transactions into specialized, critical-information storage is about to move upscale, quickly, into the mainstream. Production data processing is going on-line, a transition made possible by distributable database technology.

The ability to conduct business is becoming inexorably more time-critical, whether restricted to local operations or those sprawled around the world. The spreading of data-analysis and decision-support assignments throughout an organization reinforces further the trend toward the immediate processing of all available data. Communications and computing technologies, now available in distributable databases, have advanced such that corporate information can be input and made available to any point within a worldwide organization virtually simultaneously. Even production data processing, now usually an overnight process, must go on-line. Technologies that made possible on-line production data processing (first developed and refined for on-line transaction processing [OLTP]) are being redefined and organizational information structures reformed into on-line enterprise computing (OLEC) systems.

Long an industry leader in OLTP systems, Tandem is just one quarter away from completing its transitional phase into a major (and probably its most significant) new product cycle. Momentum at Digital Equipment is slowing as the company begins its entry into a similar transition phase. Furthermore, Digital Equipment faces the even greater challenge of overcoming its heritage in time-sharing design. IBM seems to be mostly focused on mainframe, minimainframe and end-user computer markets, once again directing its primary efforts toward this new market based on its traditional strengths in designing upgraded host processors and operating systems.

On-line transaction and inquiry processing in its tightest definition represents a 10-20% per year growth market. It is presently sized at $5 billion (U.S.) per year within the overall $30 billion, big-systems market. This market mostly comprises transportation clients supplied by IBM, and bank cash-machine support systems supplied by Tandem. To some extent, all computer systems process transactions, but we are focusing on the transition of OLTP from its former niche position into the mainstream of the commercial computer systems industry as OLEC. We estimate that this market segment can now achieve a 20-30% per-year rate of growth, becoming a $20 billion (U.S.) per year market in the 1993-1995 time frame.

The market for **distribution systems** spans the markets for mainframes managing the corporate-wide, distributable database and the end-user. The mainframe market segment has grown to more than $30 billion in size and its most dynamic part, IBM-Operating Systems Compatible (IBM-OSC) systems, is growing at 12% per year in support of new distributable-database functionalities. The end-user segment, for microcomputers running those distributed applications and supporting applications interfaces, has grown to at least a $20 billion per year market size, expanding at an estimated 10-20% annually.

While every company in the computer industry wants (and in one fashion or another believes that it is prepared) to participate in this new market-expansion phase, we see only three companies as best positioned at this time to convert plans and products in order to gain substantial market shares: Tandem, IBM and Digital. In terms of meaningful, incremental impact on investment-performance measures as derived from participation within the distribution-systems market and the transition to OLEC, we prefer Tandem's strategy.

INDUSTRY OVERVIEW

An Explanation of Where We Are

Western European commercial computer-systems market demand has remained strong, expanding at a higher sustainable rate than in North America, because the European market is in a replacement phase while the North American market is in upgrade phase. The Western European market has traditionally focused on midrange, or small business systems, and lagged behind by one technology cycle. That market is now experiencing a massive, multiyear replacement, moving up to mainframe-based, multinational, transaction-processing systems. It is also prepared to move further upscale into on-line production systems, and is, at least technologically, in sync with patterns of domestic usage.

Systems installed in the North American market over the second half of the 1980s support 7/24 data processing (7-day-a-week/24-hour-a-day operations, including on-line transaction/query and overnight batch-production processing). New systems technologies are now available to begin the full conversion from 7/24 to on-line production/transaction/query data processing, or OLEC. The availability of systems software for both the emerging, mainframe-supported database market segment and for microcomputer-supported distributed applications depends on the resolution of gating[1] issues.

[1]Gating – A circuit with one output and two or more inputs whose output is energized only when certain input conditions are satisfied.

A truism about software: to get better, it must get bigger (and more complex). It never gets smaller. Better software requires bigger memories. Disks don't transfer data anywhere nearly fast enough to go fully on-line, enterprise-wide. ESA/370 may be the world's greatest paging (swapping) system, but it too needs the high-speed, 4Mbit dRAM in Expanded Storage. (Those 4X density chips are scheduled for market shipments this fall.) While software availability constitutes the gating issue, bigger physical memory constitutes the timing issue that will permit faster market growth.

Much has been written about the ESA/370 system-software implementation ramp and whether or not it is much too slow. In our view, ESA/370 is right on its three-year schedule and it will be to the advantage of investors that it not be "pushed" too rapidly down the "throats" of IBM's installed base, so to speak. Not even the most aggressive MIS manager would jeopardize production data processing by crossing over to a new operating system faster than the three-year implementation phase (test, applications development, then production). The first year of a transition tests for incompatibilities against installed-vendor, third-party, and home-grown software systems. The second year (summer of 1989 through summer of 1990) concentrates on limited production and applications-program development. The third year brings full production and the movement to complete the transition to on-line data processing. Thus, the decade of the on-line enterprise is just now beginning.

Table 1 outlines the market availability of IBM systems software necessary to complete the transition to OLEC.

System software has to be the first, critical element in the reinvigoration of demand growth, closely followed by applications software, which must follow the successful introduction and implementation of new system software functionalities. Those new or improved functionalities are grouped into four categories, as follows:

(1) **Data-Management Software**, including both distributabale database management (the logical organization of central storage) and automated storage hierarchy management (the physical organization of central storage);

(2) **Transaction-Processing Software**--IBM and Digital employ monitors, a type of second or overlay operating system while Tandem deploys its functionalities as integrated (in fact intrinsic) characteristic of its operating system technologies;

(3) **Networked Transaction Protection** creates confidence within the using organization (and its customers) that whatever the underlying system structure, databases contain error-free data (the essential function is known in the trade as a "two-phase commit," a more advanced networking and multifaceted database update concept than journaling, queueing, or a simple commit); and,

Table 1

Availability of IBM Systems Software

Package	Function	Market Availability
MVS/ESA (MVS/SP 3.0) Support Modules	Control monitor Job Entry Subsystems, etc	February 1988 February 1989
DF/SMS (Data Facility/Storage Management Subsystem)	(MVS/DFP V3.1) Automated Storage Mgmt.	June 1989
PR/SM	Logical partitioning	February 1988
DFP (Data Facility Product)	Data management, device support, program library mgmt., end user system catalog	Summer 1989
New microcode for 3990 controller for high-speed data transfer	Eventually leading to 100 MBytes/second channels by 1992	September/ November 1989
CSP (Cross System Product)	Application development on 3090S for AS/400 & PS/2	Summer 1989
DB2 v.2 r.2 and Database Repository	Distributable database	Summer 1989 Spring 1990
CICS/ESA & DFSORT	Most advanced communications and transaction processing monitor for on-line cooperative processing, extended to all IBM systems products including PS/2	Summer 1990
OS/2-EE1.1 & PM	Presentation Manager - graphics interface	November 1988
OS/2-EE 1.2 & QM	Query Manager - integrated database	September 1989
OfficeVision	Fully available applications sets	Summer 1989- Summer 1990

(4) **Office Automation Frameworks** for advanced, end-user computing applications based on access to central, distributable databases with analysis, decision support and accounting, all based on transaction-processing technologies.

Transaction-processing systems designers at all the key vendors (yes, even Digital) agree that the corporate database must be centralized, but available through distribution to end-user applications for access and update. Even under the client-server (or, more precisely, requestor-server) model of transaction-to-(distributable) database computing, the database must be "centrally" located for control and cost reasons. That centralization is, in fact, a logical construction, not a physical one. The database may actually be spread physically across a country, or even around the world, but logically it must be a single, interconnected and, therefore, both distributed and distributable structure.

Those capabilities are going to be available, but are not yet in the marketplace. Tandem is the single exception in categories 1, 2 and 3, with those functionalities available now. With only the promise for the future--and in the absence of those functionalities--in essence, without something new for customers to do with "new" compute capacity, but to compute more data in the same "old" way, the markets for the industry-at-large are degenerating into "price-war" conditions likely to continue for the rest of 1989. In other words, the good news is that better times are coming; the bad news is that they are not here now.

What has Distributed Processing Done to the Computer Industry?

By the middle of the 1980s, the era of simpler, time-sharing computer architecture had evolved to maturity. Time-sharing systems were no longer in demand, nor being designed for the mainframe markets and were beginning to fade as a design focus for minicomputers. The host-intelligent, terminal-serving computer design of the 1970s-1980s began to yield market share to its replacement: end-user computing.

Under time-shared designs, the applications interface resided in local-terminals or cluster-controllers, but applications processing was accomplished by central computer power. When ready to compute (process), the user's terminal (or terminal cluster-controller, if terminals were being bunched together into a batch process) would initiate a terminal service call (request) which interrupted (halted) an ongoing (active) application in the host computer. Since processor intelligence in the 1970s and early 1980s was the expensive item in computing, it had to be shared in this fashion. So-called "dumb" terminals were the commodity.

By the mid-1980s, processor intelligence, especially in less-complex forms than those required for central computing--a limited number of routine or predictable applications--began to become mere commodities. Microprocessors, packaged as desk-top microcomputers, ran applications (as well as applications interfaces) locally--and economically. Applications interfaces were elevated into system-wide interfaces. Those entry-points, or "windows," into what is now being called the "enterprise system," make all of the resources of an entire organization's computing and communications system available instantaneously to qualified users. Thus, the bell tolled for time-sharing.

The host computer has already evolved into an even more important element within the new, enterprise-wide, system technology: it is now the keeper of the database, the central repository of all an organization's information. To be valid as an asset to the whole organization, databases must be "shareable." Simple communications failures cripple the utility of separate databases spread around the enterprise. Therefore, the database must be centralized and the distribution system must not be exposed to high-risk, single points of failure, i.e., it must be fault-tolerant. What MIS really needs is a shareable (distributable), centrally-maintained database. In order to provide the greatest benefit, the database must be free of errors--even speed and security rank in importance after freedom from error. To achieve that status, mainframes and mainframe-supervised storage systems are now being designed as parallel structures, just as Tandem's *NonStop* systems have been since the company's inception. Functionally, the mainframe complex becomes a macrocosm "server," to the whole organization's "clients"--its end-user computing structure of distributed applications.

Between the distributable database and distributed applications must fit the distribution system. The present market for distribution systems is fragmented by conceptual approaches, stretching between, but embracing both computing and communications technologies. For instance, the distribution-systems market could be defined as "backbone," or "trunk networking." An example of this approach is IBM's SNA. It could also be defined as embracing Local Area Networks (LAN), such as Ethernet or Token Ring. Both of these definitions, however, are more comprehensively labeled as transport mechanisms. If we extrapolate this definition to a higher, value-added level, to include error-free information flow and fault tolerance with no single point of failure, then the focus of the definition shifts to on-line transaction processing systems that are also "transport mechanisms" of the highest functional value. In our opinion, among all direct and indirect competitors, Tandem serves best the distribution-systems market emerging between distributable databases and distributed applications.

A Tale of Two Strategies to Capture Shares of the Database Market From IBM

IBM's Position

IBM "owns" the corporate (central) database located within what the industry calls the "glass house," or data center with an estimated

60-80% of the mainframe market (either defined as all mainframe corporate users or the faster-growing IBM-OSC segment, respectively) and defends that market by deploying its most technologically-sophisticated interfaces. The distributable database product of choice for those users who are dedicated to implementing IBM technology will be DB2-Version 2, Revision 2. While this product has been "brought closely" into interaction with IBM's mainframe operating systems, it is not as tightly integrated to the operating system as the imbedded database managers employed on the System/38-AS/400, or even the OS/2-EE for PS/2. IBM's mainframe system software designers have left the "hooks" in place to implement a third-party relational database product such as Oracle's as a hub for those users sophisticated enough to build multivendor networks. We call this a "one-only" vendor plus "one-many" vendor strategy. One-many vendor installations facilitate the collection of already-installed equipment and applications under the umbrella concept of sophisticated users themselves building multivendor networks; nevertheless, IBM remains inevitably in control of the network complex and, therefore, in control of the primary account relationship.

The Most Sophisticated Interfaces

One of IBM's most complex, but valuable, interfaces is called LU6.2 (logical unit--an interconnection or protocol between parts of the whole system buried within system software; devices are called PU, or physical units in IBM's systems-software lexicon). It is this Systems Application Architecture (SAA) protocol that facilitates cooperative processing from mainframes (ES/3090S) and minimainframes (AS/400) out to microcomputers (PS/2).

The LU6.2 also allows the interconnection of equal-class computers, or computers that are not IBM system-software compatible, as "peers." In fact, LU6.2 is often referred to as the standard mechanism for peer-to-peer communications. It not only is the beginning of a computer-clustering facility, but one of the important vehicles for joining data, wherever physically located, to form the common corporate database. Another facility that should come along about 1991-1992 is 1GByte/sec. data pathway between physically separate, Expanded Storage units that will create the physical, high-speed storage that holds the collective distributable database.

Cooperative processing is IBM's term for distributable database/ distributed applications computing. Some observers have also taken to calling it "cleaved applications," implying that applications can be split and assigned to whichever computer runs a particular task more efficiently, whether mainframe or microcomputer. An important corollary is that neither class of computer is exclusive, but rather mutually interdependent. This explains why IBM has ES/3090S products down to $500,000 price tags (and AS/400s under that), completing the full host product line, as well as large, medium and small PS/2s for distributed applications, including office-automation file servers for local support of distributable databases.

Manipulation of the distributable database is a critical example of cooperative processing. While the database manager (DB2 v.2 r.2) is itself classifiable as "system software," it is also an application. Consider the functions necessary to change the most important type of account in any enterprise: accounts receivable. Nothing is more important than having error-free records of who owes the firm money--it defines a critical dataset, both financially and legally.

Receivables collections can occur physicially anywhere within an organization and, increasingly, the posting updates to centrally-maintained records are sent to the "glass house" from somewhere other than central data processing. DB2 running on the mainframe is the controller of the update entry. It rolls out the particular receivables for update on immediate receipt of a collection (immediate--not overnight--because cash is tight). A remote microcomputer sends the entry updating the record from information contained in its OS/2-EE database maintained in local storage. Cooperatively with the mainframe, the DB2 "controller" confirms that all the changes have been received and are error free. Only on mutual assurance by the controller database and sender database that the change is right does the controller execute a "two-phase commit" to the change. If error is suspected, all changes are rolled out, the record is restored to its original status and retry is attempted. Imagine this process occurring with the entire corporate storehouse of information covering all aspects of the business for accounting or analysis--on line, i.e., immediately as changes occur. The key concepts are: all data is critical data; all databases must be error free; all communications and computing must be invulnerable to any single point of failure.

IBM does not yet offer fault-tolerant computing, but it is building toward that, certainly in the physical and logical construct of the central data base that is one of the company's three strengths. The second strength: IBM's mainframes (and within maybe a year, its minimainframes) are parallel-computing structures. The third strength: whatever IBM offers works or the company will fix it until it does. Customers can count on that, and at least 60% of the worldwide, commercial-computer market has.

Although computer industry analysts are captivated with the notion that there is some precision value in ratio calculations such as dollars per millions of instructions per second (MIPS), or dollars per transaction per second (TPS), real buyers make decisions according to a substantially broader set of functional criteria. IBM has rarely won a mano-a-mano competitive evaluation on the basis of MIPS or TPS, yet it wins business most of the time. Its systems overall run performance specifications that can be truly awesome. Mainframe systems can process thousands of transactions per second, and do so for the airline, car rental and hotel reservation database set of users. For banks and other organizations, running substantially more complex transactions and applications--one to three hundred transactions per second--is common. Minimainframe AS/400s, which were designed specifically for general-purpose computing have design targets of up to 600 interactive office automation users (at less than $1,000 per user) and up to 20 transactions per second at roughly $19,000 per transaction per second. We should keep these measures in mind as we compare the performance of Digital and Tandem equipment more accurately.

IBM will not only vigorously defend the "glass house": it may, in fact, be experiencing a modest increase in market share through capture of new corporate account central sites from other, non-IBM Operating System Compatible (IBM-OSC) mainframe vendors worldwide. The Company is also aggressively moving out from the central site and downward throughout its client organizations, nowhere more aggressively than in the office automation market. Its new graphics-interface-based OfficeVision and 486 technology products are intended to be the "assault vehicles" used in the attempt to gain the leading market share position against the present number one, Digital's older, character-based All-in-1 and VAX/MicroVAX office systems.

Digital's Case

On the other hand, Digital has a 1990-1991 offensive in its own "battle plans." We see that company as focused, not only against IBM, but also the weaker positions of the non-IBM-OSC mainframe companies. Presently, somewhere between 50% to 60% of Digital's revenues are derived within the commingled IBM-Digital market place, so Digital is broadly engaged across the whole marketplace.

Digital's product lineup will soon include a Halloween announcement (and early 1990 release) of its largest ever, single processor system, the long anticipated *Aridus*, or VAX 9000 series. At 30 VUPS (VAX Units of Processing, or multiples of 1977-vintage VAX 11/780 processing power as unity), it should have more than four times the processing power of its previous, 1987-vintage top machine, the VAX 8810. The generally-accepted equivalent (relative to the performance of an IBM single-processor mainframe): one VUP equals approximately one-half IBM-equivalent MIPS (or Millions of Instructions Per Second). The *Aridus*, therefore, falls about midway between the IBM 3090 models 150S and 170S (at an estimated 13.5 MIPS and 17.5 MIPS, respectively). We expect the initial VAX 9200 single processor to be priced in the neighborhood of $1 million versus $1.65 million and $2.1 million for the IBM systems. Apparently, *Aridus* will have both multiple (initially two, eventually four) processor and dual-vector, add-in features. Much as with IBM mainframe design, *Aridus* will be offered directly to the commercial market as a transaction-processing computer, and, with the vector option, to the technical markets as a minisupercomputer. Single-processor, transaction-processing performance by *Aridus* is likely to fall into the low-end mainframe range of 50 to 100 transactions per second.

While an important upgrade step for the typical Digital installation, especially those employing VAXclusters in engineering and manufacturing support, the *Aridus* now pits Digital against an IBM product line that has already been in the market for four years and, at that, into the mainstream of its competitive preserve. Within two years, the single (seed) processor in an IBM parallel computer will advance from the 22 MIPS of the 3090 model 180S (at a clock of 15 nsec) versus the estimated 15 MIPS (at 16 nsec) of the *Aridus*, to an estimated IBM "Summit" clock of sub-10 nsec and probably more than 30 MIPS per seed processor. That specification could also jump to 50-60 MIPS if IBM introduces its modified RISC implementation on the first "Summits" expected to be ready for shipment in early 1992.

Digital's other new system release, already in volume shipment, is the VAX 6000 model 400 series, roughly comparable to Tandem's CLX 700 and IBM's AS/400 product lines. A more direct, Model 400-to-CLX 700 comparison is shown in Table 2.

The VAX 6000 concept differs, at least initially, from its time-shared, high-performance counterparts that we believe will be labeled VAX 9200 (or possibly, VAX 9000 model 200). While *Aridus* may "initially" be limited to two (2) processor configurations, the VAX 6000's are available in symmetrical strings of up to six processors per system. Someday, systems are likely to be ganged together into fault-tolerant, multiple-unit configurations. According to the rumor mill, Digital is preparing a low-end, VAX 6000, model-300-based, fault-tolerant design for market entry some time in mid-1990. We would compare it to the two-year old CLX 600 product line. Although the hardware path being followed is generally similar to that pioneered by Tandem, the approach taken by these two competitors is quite different beyond a superficial similarity.

The VAX 6000 models 420-460 series of two-to-six processor systems employ a symmetrical multiprocessing (SMP) version of Digital's VMS operating system that is still, in our opinion, somewhat limited in its functionalities. All 2-6 processor systems run under one copy of the operating system, but we detect no single-image capabilities yet. When next available, the separate processors can back one another up and each one accesses a common main-memory as well as a disk-storage pool; however, they cannot intermesh harmoniously to work on one large, unified problem (single-image). While not at all a limitation in performing multiple small jobs, the VAX 6000 cannot be compared, in aggregate, to a single-image-capable, multiprocessor mainframe for general-purpose computing.

The multiprocessor factor (MP) penalty is very much in line with that of multiprocessor mainframe systems from IBM and Amdahl, running 86% according to the statistics released by Digital shown in Table 2. IBM and Amdahl have released ballpark multiples of 80% to 90% as an MP factor. As Digital evolves beyond VMS

Table 2
Digital Equipment Versus Tandem Computers
New Midrange Systems Comparisons

Digital VAX 6000	Model 410	Model 420	Model 430	Model 440	Model 450	Model 460
Number of Processors	1	2	3	4	5	6
CPU Performance						
VUP (2) - multiples of VAX 11/780	7	up to 13	up to 19	up to 25	up to 31	up to 36
Multiprocessor Factor (MP)	1.00x	1.86x	2.71x	3.57x	4.43x	5.14x
$ per VUP	$34,143	$30,692	$25,737	$25,080	$22,581	$20,889
Debit/Credit TPS	12.0	21.3	-	38.5	-	44.0
$ per TPS	$19,917	$18,732	-	$16,286	-	$17,091
Office Automation Timesharing Users (OATS)	216	368	-	600	-	656
$ per OATS	$1,106	$1,084	-	$1,045	-	$1,146
MAX I/O Bandwidth	60 MB/sec	60 MB/sec	60 MB/sec	60 MB/sec	40 MB/sec	40 MB/sec
Systems Price (VMS)	$239,000	$399,000	$489,000	$627,000	$700,000	$752,000

Tandem CLX 700	720	740	760	780			
Number of Processors	2 (3)	4 (3)	6 (3)	8 (3)	(4)	(4)	(4)
Fault-Tolerant MP based on TPS	1.00X	2.00X	3.00X	4.00X			
TPS	7.4	14.8	22.2	29.6			
$ per TPS	$14,189	$13,851	$13,738	$13,682			
Systems Price Guardian 90 XL	$105,000	$205,000	$305,000	$405,000			

Footnotes

(1) More statistics are provided for Digital's 6000 than for the Tandem 700 because the 6000 is intended to be general purpose and also is aimed at both the technical and commercial markets, while the 700 is focused on transactions.

(2) VUP - VAX Units of Processing; a proprietary measure of performance that only has meaning in comparisons within the VAX product line.

(3) Each CLX 700 CPU consists of two cross-coupled processors functioning as a single central processing unit for continuous fault checking in OLTP applications, thus featuring both hardware and software fault tolerance characteristics.

(4) For increasingly larger systems, multiple numbers of individual Tandem systems may be interconnected via functionalities under the Guardian 90XL operating system.

version 5.x, single-image capabilities may be available. We guess within (or just beyond) the next two years.

In contrast, Tandem systems are not intended to be symmetrical multiprocessors. Each computer in a CLX or VLX system is a discrete processor complex with its own main-memory in each computer's configuration. Each works on a discrete application, which may be either repetitively the same, or entirely different per computer. Each additional processor complex added to (or within) an interconnected CLX or VLX system adds one full unit of processor power to the aggregate system's throughput. Tandem transaction-processing systems are at the leading edge of networked OLTP applications. Tandem has published audited test results showing that when networking overhead is added to the processing complex, the Tandem "effective" MP factor is reduced to 90% from 100%. To our knowledge, no other system has yet achieved Tandem's functionalities via networked OLTP capabilities, so it is impossible to assess at this time what additional "effective penalty" networking would apply to either a Digital VAX or IBM 3090 complex. The remaining (and as yet unexplained) issue raised by the Digital data in Table 2 is the curious 33% decline in internal bus bandwidth for models 450 and 460 versus models 410-440.

The Critical Issue: When Will New Software Functions Be Released?

Digital's VMS 32-bit proprietary operating system was derived from its predecessor, the RSX-11M 16-bit, time-sharing progenitor. It has evolved away from its exclusively time-sharing roots to become a mid-range system supporting a production-data-processing operating system, but as yet without built-in (imbedded) OLTP functionalities. VMS, in our opinion, is being backed into an on-line transaction supporting role as an adjunct to its dual, general-purpose, commercial and technical computing focus.

IBM achieved a similar technological evolution in the late 1970's and overlayed onto its commercial, virtual-memory mainframe, production data-processing operating system, a communications and transaction processing monitor (basically a second specialized operating system) called CICS (Customer Information Control System). IBM's mainframes have always had the processor horsepower to handle all of the system software necessary to fulfill simultaneously a complete set of diverse, system-software and applications requirements. Clearly one of these is allowing the user to build systems with multiple operating systems and multiple database managers of choice, while intermixing input systems of all kinds (e.g., "dumb" terminals, PS/2s, cash machines, factory-floor devices). With the *Aridus's* performance potential, Digital has a computer that is entering, for the first time, the lower part of that performance region.

Presently, Digital offers two separate monitors, ACMS (Application Control and Monitoring System) which was developed in-house and introduced in 1984, and DECintact (which was acquired and added to the product line on 1988). Why two? We view ACMS as an add-on-function, transaction-processing monitor to a general purpose departmental system. DECintact had all the features that Digital would need to develop into a future monitor, like automatic host rollover (which backs up a failing computer without losing transactions) and it was available, having been written for VMS. Therefore, it was acquired. DECintact presently works with Digital's higher-performance, RMS flat-file manager, but will be improved to interface to Rdb (Digital's relational database manager). We estimate that this development will take one to two years. A future (DECintact) monitor with ACMS integrated characteristics will be merged into (and imbedded within) a future version of VMS. We estimate this development will take two to five years. Within one year, Digital's present queueing-transaction protection approach (a simple commit format) should be upgraded to full networked transaction protection using a two-phase commit.[2]

Some observers believe that a multiplicity of monitors, promised developments and the early state (but improving via Digital summer school retraining) of in-house marketing and sales knowledge of OLTP is confusing the targeted customers who are outside of a company's captive, departmental-computing installed base.[3] Within its ancillary OLTP products approach to date, Digital has, however, managed to build one of the larger transaction-processing niche businesses in the industry. We estimate it to represent 15% of revenues.

Much of the DECtp story is either new, with new computers, or in the "to be introduced" status, as with software and features. For example, the *Aridus* is being readied for release without the benefit of field (or beta) test time, which may slow the product's initial acceptance and volume-shipment rates for possibly up to a year. Much of the company's VAX architecture approach already available to DECtp designers is well conceived. VAX products easily lend themselves, to modular requestor (client)-server-storage management designs. These evolutionary steps, however, will consume a great deal of time.

[2]According to Dave Zwicker, Digital OLTP Consultant Relations Manager, a "Two phase commit, simply put, lets you know that the transaction has reached the database, or file structure, by coming back to the transaction generator from the repository with a commit message. It's more reliable than a queue management message because the latter goes only to the queue, not to the database. Every other functionality now in the monitors, with the exception of transaction control and applications-development style, will eventually become part of VMS, or a kernel of VMS. Thus, from Digital's OLTP effort, VMS will gain capabilities for recovery, journaling, queuing and remote procedures." *DEC Professional*, Vol. 8, No. 3, March, 1989.

[3]Reference: *Datamation*, "Back To The Drawing Boards," Vol. 35, No. 13; July 1, 1989, pp 57-61.

William R. Demmer, Digital Vice President for Mid-Range Systems, was recently interviewed by *Digital Review* (May 22, 1989) and quoted as saying:

> The goal I'm working toward is to replace that nice, single time-sharing system we all know, love and understand with a database that is available to the whole enterprise. We'd like to replace the entire organization's computing capability and still have it look the same from the desktop as it did when it was based on a time-sharing architecture. We are between five and 10 years away from providing that transparency.

Digital appears to be planning for its capturable customers a strategic conversion from both IBM database sites (and even entire businesses still dependent on non-IBM-OSC mainframe systems) into VAX/VMS based OLTP and production data-processing systems. Over the next two years its principal competitive edge appears to us to be cost-per-transaction and system lifetime cost-of-ownership advantages. While the non-IBM-OSC companies may have installed-base vulnerabilities, the type of customer of these companies was not necessarily the swiftest to implement new DP features and the most eager to spend lots of new monies on system change. The counter-IBM strategy has been made more difficult because IBM has made all the right moves, like SAA, and few if any mistakes in its defense of the "glass house", other than the elongated introduction and implementation phases of its new software-based features, especially against incursion and substitution by an attacker coming up from the departmental-computing direction. While Digital's position has uncanny similarities to Tandem's already successful strategy, Tandem is well into its execution phase while Digital, in our opinion, has at least another year of DECtp development, internal education, and transition to complete. But complete it they will, and Digital too will once again be an OLEC force for competitors to reckon with.

Tandem's Case

Tandem has crafted what we believe is a more sophisticated (and potentially more successful) strategy of competitive coexistence within, and then capture of, a segment of the IBM corporate (central) database market. Tandem-as-competitor brings not only different, but rather the **highest functional values** and performance to the market for distributed transaction and query systems, stretching from the "glass house" interface to the system's entry-points. Tandem has crafted (from **in-house development efforts**) connections to IBM's host complex as a full-functioning host, or peer, in both the developed IBM-SNA connectivity technologies as well as the emerging, international OSI connectivity technologies. It has developed and offers a superior relational database product that is **tightly integrated** to the Tandem *Guardian 90* operating system. *NonStop SQL* has just been substantially upgraded to support on-line production data processing as well as on-line transaction and inquiry processing. Given its development thrust upward into mainframe-class technologies, Tandem will undoubtedly extend its superior, message-based operating system technology via asyncronous, high-speed fiber-optical, data-pathway technology to cover an entire organization's distribution-system requirements. *Guardian 90XL* now interconnects multiple levels of Tandem's computer-systems products (CLX, VLX, as well as the soon-to-be-released "Cyclone") to the Tandem-supported part of the critical database, and then the Tandem host to the IBM host; Tandem will likely also be moving quickly in the direction of "nonstop-to-the-desktop" local-area-networking technology at the user-level (or entry marketplace) now that it has absorbed its acquired Ungermann-Bass product lines and development efforts.

At present, Tandem coexists with either of IBM's one-only or one-many strategies, but market-share capture opportunities can (and should) coalesce toward the one-one alternative, in our view, based on a comprehensive product edge over all of its competitors; Tandem probably accomplishes what Digital hoped to do--not through replacement, but through attachment. Over time, Tandem may be one of the few vendors able to leverage its present strategy into a proprietary-user base and it should be extremely successful, over both the short and long term, in developing the distribution arms of big-system implementations through the development of what we are calling the **distribution-systems** market. In our opinion, Tandem's enhanced financial performance depends on completion of the current product transition, likely in 4Q 1989.

The Evolution of Tandem

Matrix 1 displays the salient differences that we see now in the Tandem-versus-Digital product positions:

Matrix 1

Tandem Versus Digital

Product Positions

Tandem Product	Our Comments	Digital Product	Our Comments
(1) "Cyclone"	2nd generation ECL upgrade of VLX. Intensive I/O upgrade.	"Aridus"	2nd generation ECL upgrade of VAX 8000. upgrade of VAX 8000. Intensive computational (engineering) upgrade.
(2) NonStop SQL	Integrated DB into Guardian XL90.	Rdb	Separate DB module now offered for free, not integrated into VMS.
(3) IBM-TDM	SNAX/CDF LU 6.2/PU 2.0 Subhost (highest connectivity) developed by TDM.	IBM-DEC	Mostly third-party efforts directed by customer base.
(4) FOX II	Very high speed fiber optical backbone (trunk).		
(5) EXPAND	Software interlink of up to 255 NonStop systems.		
(6) CLX 700/600	Hardware AND Software fault-tolerance.	VAX 6000	Redundant symmetrical multiprocessing; physical partitioning only.
(7) Access/One	"Nonstop to the Desktop" under development. Full user-to-error free database connectivity.		
(8) Applications Architectures	Tandem integrated Mfg. Wholesale Banking, and Intelligent Networking--with leading-edge System integrator partnerships.		

Tandem's initial *NonStop* product line, released in 1978, introduced "fault-tolerance," a new concept, to the computer-systems marketplace. Much like similar technologies employed in the telephone-equipment marketplace, *NonStop* computers bypass failing components and subsystems. Add-on boards expand so that system elements can be installed (or boards removed, repaired, and replaced) while the system remains in operation. Because it need never be shut down, the system operates literally "non-stop."

What was the underlying selling point (and primary benefit) to the user's organization that sold Tandem's *NonStop* systems quickly, especially to the sophisticated, large corporate audience? *NonStop* was the first system ever to generate and support an "error-free" database. Only when error-free could the database be considered a high-value corporate asset (and, therefore, an asset to the MIS chief's career development) and become the new foundation for competitively strategic systems.

Error-free databases and OLTP are mutually dependent concepts at the core of strategic systems technology. Transaction processing could only be considered on-line (that is real-time) if the transport and computing system never failed completely. Further transactions cannot be processed (while the system remains on-line) unless the results of processing are completed and made available literally as fast as the transaction occurs. Furthermore, the processing of transactions in real time was only of value so long as there was the highest level of confidence that the complete information repository was virtually error-free. It is impossible to have true OLTP without all of those factors as imbedded functionalities in the product offering. Anything less is not OLTP, and what OLTP is all about is **confidence**. Why else would customers literally "walk up to a wall," ask for cash, and accept the accounting of their bank accounts? Why else would the bank put cash "in the wall" unless all parties were **100% confident** in the underlying system?

Tandem's *NonStop* has a collateral advantage: capacity to process more transactions expands linearly with the addition of more processors. *NonStop* represented one of -- if not the industry's first -- loosely-coupled, parallel-computer systems. (The addition of one processor to a two-processor installation increased the unit of processing capacity from two-fold to three-fold: this is the computer-systems definition of linear expansion.) Only processing power expands, since nothing else changes. The same systems and applications software, plus peripherals, run on the expanded system, whether upgraded with more of the same processors, or with higher-power processors up the product line.

Tandem engineers had actually designed a new **networking technology**. At its core, Tandem's *Guardian 90* proprietary operating system is a messaging structure, one that includes the *NonStop* system of multiple computers and each constituent computer as well as the external network from entry points to database. The subsystems within the computer are tested by "health" messages. "Health" messages are also sent between computers, under the supervision of each computer's copy of the *Guardian* operating system. Transactions and queries are themselves also messages. New service opportunities such as automatic cash machines and tellerless banking offices became practical. Tandem quickly became accepted within the international banking community, later by the business community at large. In a computer-architecture sense, Tandem captures interfaces from the entry device to the "glass house" connection. In the eyes of corporate-user decision-makers, the stage was set early for "friendlier" competition (Tandem) with the host vendor (IBM) than those who would choose to conduct a frontal assault on the host market itself (Digital). Tandem was (and still is) unique; the niche that it opened has not only become substantially larger, but is expanding subsequently into the mainstream of both commercial and technical data processing. The issue is no longer acceptance, rather Tandem's ability to continue successful execution of a leading-edge product strategy and maintain competitive differentiation.

DEFINITION OF SERVICES PERFORMED

On-Line Transaction Processing

On-line transaction processing (OLTP) is generally accepted as characterizing a computerized mode of applications processing in which messages received from systems-entry devices are processed into a common database (information repository) in real time (without apparent delay). Messages can be either transactions or queries. A transaction changes records contained within the database. The response to a query reports, but does not change information contained within the database. Complex transactions involve the simultaneous change of a large number of records, while simple transactions involve only a few. This simultaneity complicates the interpretation of simplistic measures like number of transactions performed per second (TPS). Organization of database information (records, files) conforms to two basic types of structures: hierarchical, or tree-structure (communicates rapidly); and relational, flat or table-structure (slowest, but easiest to construct and getting faster with increasing availability of computer-hardware power).

Transaction-processing-rate benchmarks are timings of the TPS. Attempts have been made by both vendors and independent agencies to derive useful product comparisons from competitive trials. While the task would seem to be relatively simple--measure executions and divide by seconds--variations in task criteria, systems configurations, differing device characteristics per vendor even within similar systems configurations, and all manners of other variations make the idea of precise comparisons misleading, however seductive. We suggest that transactions per second (TPS) as a way of comparing systems performance between different vendors be interpreted only in a very general sense. In fact, such statistics tend to be useful, in our opinion, only when arraying a single vendor's different classes of products for pricing purposes.

If transaction and query messages are at the heart of OLTP, communications computer power is more important than MIPS (millions of instructions processed per second). The *Guardian 90* operating system, which is messaging based, is efficient at the process of queuing (sequencing) a very large number of concurrent jobs. As Tandem releases more powerful *NonStop* computers, with full capacity--not only for the execution of transactions and queries--but also for on-line "batch" or production support, the Company should finally be accepted as a fully-qualified, production-class vendor and be leveraged up into a larger marketplace. That should be a bigger "ball game" for Tandem. In our opinion, that development will also more clearly differentiate *NonStop* market-share capture opportunities versus its most direct competitors. Tandem satisfies mission-critical applications requirements as an essential stepping-stone to OLEC while Digital is still, in our opinion, focused on departmental computing applications, which are not clearly evolving toward OLEC. Tandem should also capture market share against IBM as commercial users move critical database, general-purpose applications onto *NonStop* mainframe systems.

THE INTERNAL CONNECTIVITY

The Tandem Vision

Messages are passed from the application through the file system and disk process to the database in the operation of any one computer within a *NonStop* system. A *NonStop* systems architect recently said that the Company had dedicated 14 years of R&D to make the message interface between an application and the files as "skinny" (efficient) as possible. The message interface layer and system-software structure within the *NonStop* system, consisting of from two to 16 separate computers and then two banks of up to 16 computers in fault-tolerant configuration, contribute the interconnection between those computers, in effect defining the *NonStop* system. Internal systems communications takes place over dual (parallel) Dynabuses, which extend conceptually outward to interconnect entry devices to the *NonStop* and also interconnect *NonStop* as a "peer host" to the IBM mainframe "glass house."

Tandem's messaging system consists of the "work" of its computers plus "health" messages. This latter aspect of the system's design is what facilitates *NonStop* fault-tolerance. As each computer processor's power expands through design cycles, and "health" messaging remains a relatively constant load factor, *NonStop* systems' performance and functionality expand beyond on-line transaction processing and query handling to on-line batch or production processing. Thus, the normal evolution of processor development should facilitate transition from front-end niche to general-purpose data processing. In our opinion, Tandem engineering is fully capable of advancing the state-of-the-art in both internal and external communications technology, quickly moving forward with the transition from typical minicomputer, synchronous-bus operation to high-speed, asynchronous "mainframe-class" (intersystem) communications, based on fiber-optical technology.

Tandem hopes to make a fully successful transition from its original fault-tolerance-based niche market position to becoming a vendor of products with systemwide, modular expandability and a high availability of error-free data. Such an achievement should propel the company toward capturing a unique, Tandem-installed proprietary base. In the pursuit of this transition, it is important to note that the proprietary (non-standard) nature of the company's *Guardian 90* operating system technology is not an issue in the valuation of *NonStop* technology by the marketplace. Criticisms of Tandem and *NonStop* do, however, exist. *NonStop* requires (or at least has required) some degree of application-code "customization." *NonStop* systems are loosely coupled versus the tight coupling achieved by symmetrical multiprocessing systems. Loose coupling requires that each computer have its own, not a shared copy of the operating system, and systems employing that configuration are more difficult to load-balance.

Tandem responds that symmetrical multiprocessing doesn't really help in solving the OLTP applications problem of queuing (sequencing) a large number of concurrent jobs. This is a communications problem that *NonStop* solves very well. *Guardian 90XL* (and ultimately Tandem's interpretation of UNIX) are being made as transparent to the user as possible, so that other than license-fee replication, obtaining multiple copies of the operating system should not cause a performance problem. While OLTP does not present a processor-power problem, but rather a communications-intensive task, messaging tasks are what *NonStop* does best by its intrinsic design: more robust processing power results from the user's ability to intermix small (CLX 600 series), medium (CLX 700 series), and large (VLX 800 series) *NonStop* computers within the same system. Users will soon have very large "Cyclone" systems to choose as well. Units of work will be "scaled" (matched) to a systems' capability. *Non-Stop* will, therefore, feature scalability and load-balancing features common to parallel processing. Ultimately, systems software should be transparent to the end user (by reducing or eliminating requirements to consider Tandem's systems in any way different from any other computer--i.e., requiring little or no customization of applications software). Conventions such as "loosely" or "tightly coupled" should have no practical operating meaning to *NonStop* users as an impled limitation versus other vendors' symmetrical multiprocessor products (e.g., Digital VAX 6000 series).

Now that Tandem offers (and customers are implementing) *NonStop SQL* databases, applications programming is based on knowledge of two industry-standard programming languages, COBOL and SQL. Availability of Tandem's systems has, therefore, passed from the difficult-to-program state to the substantially larger universe of industry-standard, commercial-applications programmers. This is one of the BIG changes in the Tandem story.

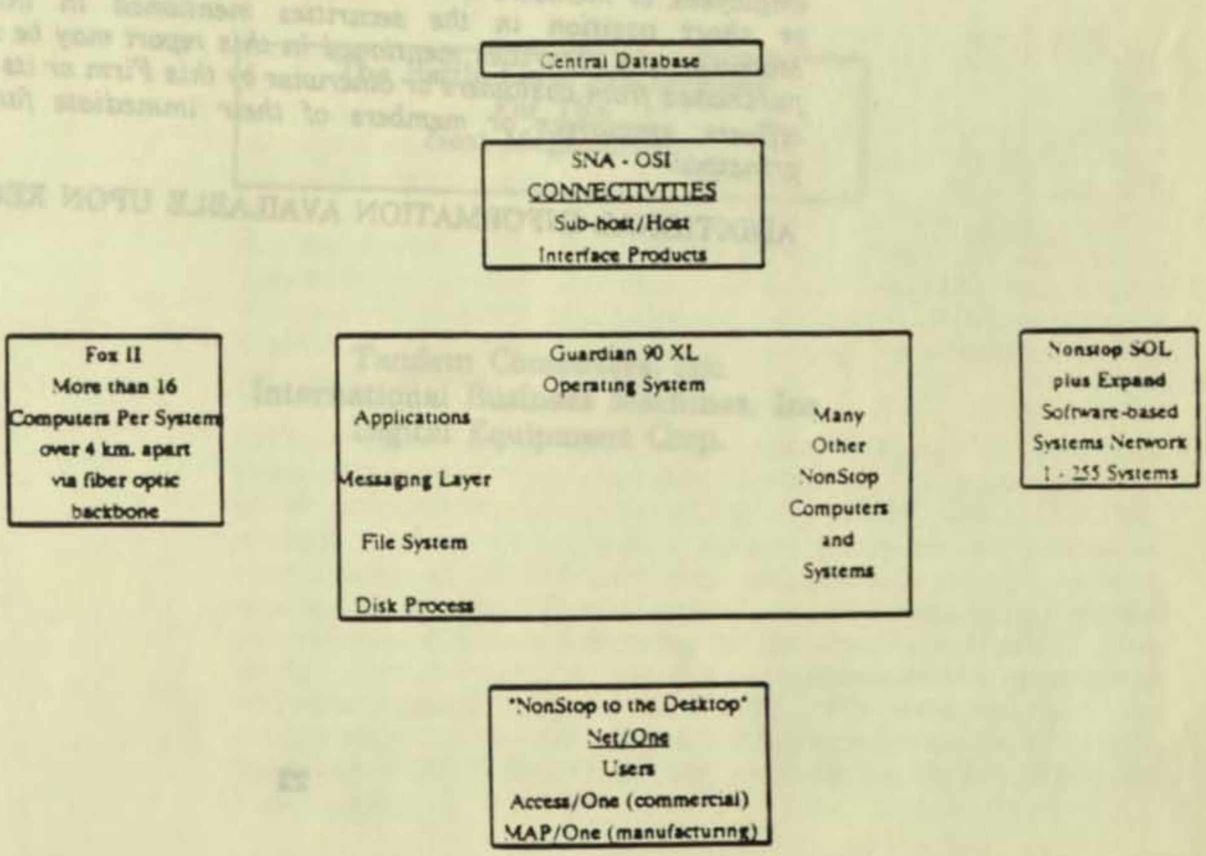
THE EXTERNAL CONNECTIVITY

Software Stands Out

While messaging is the internal and external "business" of Tandem's *NonStop* computers, today, new, more powerful computers (plus the ability to intermix and scale the entire spectrum of the processor product line into a single-system configuration) will make the company's "business" messaging plus production-data processing. The continuous, internal-external nature of *NonStop* differentiates Tandem from Digital.

Diagram 1 displays the structure and names of the present systems software and communications/connectivity product line. Externally, the messaging-transport capabilities extend the *NonStop* system up to the "glass house" interface products, to Tandem's own integral SQL-based database management system and outward via "nonstop-to-the-desktop" levels under its Access/One product banners. This is the "seamless" network that all observers have envisioned as the preferred system structure of the future.

Diagram 1
Tandem Computers
Completed Product Structure
As of Fiscal 1989



While Digital is being pushed by clients into IBM-coexistence/connectivity, in most cases through third-party products employed on already-installed VAX systems, Tandem willingly embraced the concept of IBM-coexistence/connectivity. It has developed and offers a broad range of the deepest (highest level) SNA and OSI interface products. Worth noting as a fundamental difference between the two companies: Tandem had the vision to take this position willingly, while Digital, in our opinion, is changing in order to protect its installed base. Tandem, therefore, represents our real growth story.

CONCLUSION

Tandem is like the high surfer on a wave that is turning into a tsunami (tidal wave). The only question is not whether the wave is going to get bigger, but whether or not Tandem can handle the wave. The wave has been a long time coming and we believe that Tandem management can indeed handle that wave.

Alex. Brown & Sons
Incorporated

Research

Communications & Information Systems Group

Joseph Payne
(301) 783-3271

August 1989

THE COMMERCIAL COMPUTER SYSTEMS INDUSTRY

On-Line Transaction Processing
In Transition to
On-Line Enterprise Computing

The Battle Lines Are Forming
For The
Next Megamarket

Tandem Computers, Inc.
International Business Machines, Inc.
Digital Equipment Corp.

THE COMMERCIAL COMPUTER SYSTEMS INDUSTRY

TABLE OF CONTENTS

INVESTMENT THESIS

Background

Three of the most visible companies in the commercial systems industry, Tandem Computers, Inc., International Business Machines, Inc. and Digital Equipment Corporation, are jockeying for position at the vanguard of the "battle lines" forming over the next megamarket in computer technology. What has been a niche market for processing transactions into specialized, critical-information storage is about to move upscale, quickly, into the mainstream. Production data processing is going on-line, a transition made possible by distributable database technology.

The ability to conduct business is becoming inexorably more time-critical, whether restricted to local operations or those sprawled around the world. The spreading of data-analysis and decision-support assignments throughout an organization reinforces further the trend toward the immediate processing of all available data. Communications and computing technologies, now available in distributable databases, have advanced such that corporate information can be input and made available to any point within a worldwide organization virtually simultaneously. Even production data processing, now usually an overnight process, must go on-line. Technologies that made possible on-line production data processing (first developed and refined for on-line transaction processing [OLTP]) are being redefined and organizational information structures reformed into on-line enterprise computing (OLEC) systems.

Long an industry leader in OLTP systems, Tandem is just one quarter away from completing its transitional phase into a major (and probably its most significant) new product cycle. Momentum at Digital Equipment is slowing as the company begins its entry into a similar transition phase. Furthermore, Digital Equipment faces the even greater challenge of overcoming its heritage in time-sharing design. IBM seems to be mostly focused on mainframe, minimainframe and end-user computer markets, once again directing its primary efforts toward this new market based on its traditional strengths in designing upgraded host processors and operating systems.

On-line transaction and inquiry processing in its tightest definition represents a 10-20% per year growth market. It is presently sized at $5 billion (U.S.) per year within the overall $30 billion, big-systems market. This market mostly comprises transportation clients supplied by IBM, and bank cash-machine support systems supplied by Tandem. To some extent, all computer systems process transactions, but we are focusing on the transition of OLTP from its former niche position into the mainstream of the commercial computer systems industry as OLEC. We estimate that this market segment can now achieve a 20-30% per-year rate of growth, becoming a $20 billion (U.S.) per year market in the 1993-1995 time frame.

The market for **distribution systems** spans the markets for mainframes managing the corporate-wide, distributable database and the end-user. The mainframe market segment has grown to more than $30 billion in size and its most dynamic part, IBM-Operating Systems Compatible (IBM-OSC) systems, is growing at 12% per year in support of new distributable-database functionalities. The end-user segment, for microcomputers running those distributed applications and supporting applications interfaces, has grown to at least a $20 billion per year market size, expanding at an estimated 10-20% annually.

While every company in the computer industry wants (and in one fashion or another believes that it is prepared) to participate in this new market-expansion phase, we see only three companies as best positioned at this time to convert plans and products in order to gain substantial market shares: Tandem, IBM and Digital. In terms of meaningful, incremental impact on investment-performance measures as derived from participation within the distribution-systems market and the transition to OLEC, we prefer Tandem's strategy.

INDUSTRY OVERVIEW

An Explanation of Where We Are

Western European commercial computer-systems market demand has remained strong, expanding at a higher sustainable rate than in North America, because the European market is in a replacement phase while the North American market is in upgrade phase. The Western European market has traditionally focused on midrange, or small business systems, and lagged behind by one technology cycle. That market is now experiencing a massive, multiyear replacement, moving up to mainframe-based, multinational, transaction-processing systems. It is also prepared to move further upscale into on-line production systems, and is, at least technologically, in sync with patterns of domestic usage.

Systems installed in the North American market over the second half of the 1980s support 7/24 data processing (7-day-a-week/24-hour-a-day operations, including on-line transaction/query and overnight batch-production processing). New systems technologies are now available to begin the full conversion from 7/24 to on-line production/transaction/query data processing, or OLEC. The availability of systems software for both the emerging, mainframe-supported database market segment and for microcomputer-supported distributed applications depends on the resolution of gating[1] issues.

[1]Gating – A circuit with one output and two or more inputs whose output is energized only when certain input conditions are satisfied.

A truism about software: to get better, it must get bigger (and more complex). It never gets smaller. Better software requires bigger memories. Disks don't transfer data anywhere nearly fast enough to go fully on-line, enterprise-wide. ESA/370 may be the world's greatest paging (swapping) system, but it too needs the high-speed, 4Mbit dRAM in Expanded Storage. (Those 4X density chips are scheduled for market shipments this fall.) While software availability constitutes the gating issue, bigger physical memory constitutes the timing issue that will permit faster market growth.

Much has been written about the ESA/370 system-software implementation ramp and whether or not it is much too slow. In our view, ESA/370 is right on its three-year schedule and it will be to the advantage of investors that it not be "pushed" too rapidly down the "throats" of IBM's installed base, so to speak. Not even the most aggressive MIS manager would jeopardize production data processing by crossing over to a new operating system faster than the three-year implementation phase (test, applications development, then production). The first year of a transition tests for incompatibilities against installed-vendor, third-party, and home-grown software systems. The second year (summer of 1989 through summer of 1990) concentrates on limited production and applications-program development. The third year brings full production and the movement to complete the transition to on-line data processing. Thus, the decade of the on-line enterprise is just now beginning.

Table 1 outlines the market availability of IBM systems software necessary to complete the transition to OLEC.

System software has to be the first, critical element in the reinvigoration of demand growth, closely followed by applications software, which must follow the successful introduction and implementation of new system software functionalities. Those new or improved functionalities are grouped into four categories, as follows:

(1) **Data-Management Software**, including both distributabale database management (the logical organization of central storage) and automated storage hierarchy management (the physical organization of central storage);

(2) **Transaction-Processing Software**--IBM and Digital employ monitors, a type of second or overlay operating system while Tandem deploys its functionalities as integrated (in fact intrinsic) characteristic of its operating system technologies;

(3) **Networked Transaction Protection** creates confidence within the using organization (and its customers) that whatever the underlying system structure, databases contain error-free data (the essential function is known in the trade as a "two-phase commit," a more advanced networking and multifaceted database update concept than journaling, queueing, or a simple commit); and,

Table 1

Availability of IBM Systems Software

Package	Function	Market Availability
MVS/ESA (MVS/SP 3.0) Support Modules	Control monitor Job Entry Subsystems, etc	February 1988 February 1989
DF/SMS (Data Facility/Storage Management Subsystem)	(MVS/DFP V3.1) Automated Storage Mgmt.	June 1989
PR/SM	Logical partitioning	February 1988
DFP (Data Facility Product)	Data management, device support, program library mgmt., end user system catalog	Summer 1989
New microcode for 3990 controller for high-speed data transfer	Eventually leading to 100 MBytes/second channels by 1992	September/ November 1989
CSP (Cross System Product)	Application development on 3090S for AS/400 & PS/2	Summer 1989
DB2 v.2 r.2 and Database Repository	Distributable database	Summer 1989 Spring 1990
CICS/ESA & DFSORT	Most advanced communications and transaction processing monitor for on-line cooperative processing, extended to all IBM systems products including PS/2	Summer 1990
OS/2-EE1.1 & PM	Presentation Manager - graphics interface	November 1988
OS/2-EE 1.2 & QM	Query Manager - integrated database	September 1989
OfficeVision	Fully available applications sets	Summer 1989- Summer 1990

(4) **Office Automation Frameworks** for advanced, end-user computing applications based on access to central, distributable databases with analysis, decision support and accounting, all based on transaction-processing technologies.

Transaction-processing systems designers at all the key vendors (yes, even Digital) agree that the corporate database must be centralized, but available through distribution to end-user applications for access and update. Even under the client-server (or, more precisely, requestor-server) model of transaction-to-(distributable) database computing, the database must be "centrally" located for control and cost reasons. That centralization is, in fact, a logical construction, not a physical one. The database may actually be spread physically across a country, or even around the world, but logically it must be a single, interconnected and, therefore, both distributed and distributable structure.

Those capabilities are going to be available, but are not yet in the marketplace. Tandem is the single exception in categories 1, 2 and 3, with those functionalities available now. With only the promise for the future--and in the absence of those functionalities--in essence, without something new for customers to do with "new" compute capacity, but to compute more data in the same "old" way, the markets for the industry-at-large are degenerating into "price-war" conditions likely to continue for the rest of 1989. In other words, the good news is that better times are coming; the bad news is that they are not here now.

What has Distributed Processing Done to the Computer Industry?

By the middle of the 1980s, the era of simpler, time-sharing computer architecture had evolved to maturity. Time-sharing systems were no longer in demand, nor being designed for the mainframe markets and were beginning to fade as a design focus for minicomputers. The host-intelligent, terminal-serving computer design of the 1970s-1980s began to yield market share to its replacement: end-user computing.

Under time-shared designs, the applications interface resided in local-terminals or cluster-controllers, but applications processing was accomplished by central computer power. When ready to compute (process), the user's terminal (or terminal cluster-controller, if terminals were being bunched together into a batch process) would initiate a terminal service call (request) which interrupted (halted) an ongoing (active) application in the host computer. Since processor intelligence in the 1970s and early 1980s was the expensive item in computing, it had to be shared in this fashion. So-called "dumb" terminals were the commodity.

By the mid-1980s, processor intelligence, especially in less-complex forms than those required for central computing--a limited number of routine or predictable applications--began to become mere commodities. Microprocessors, packaged as desk-top microcomputers, ran applications (as well as applications interfaces) locally--and economically. Applications interfaces were elevated into system-wide interfaces. Those entry-points, or "windows," into what is now being called the "enterprise system," make all of the resources of an entire organization's computing and communications system available instantaneously to qualified users. Thus, the bell tolled for time-sharing.

The host computer has already evolved into an even more important element within the new, enterprise-wide, system technology: it is now the keeper of the database, the central repository of all an organization's information. To be valid as an asset to the whole organization, databases must be "shareable." Simple communications failures cripple the utility of separate databases spread around the enterprise. Therefore, the database must be centralized and the distribution system must not be exposed to high-risk, single points of failure, i.e., it must be fault-tolerant. What MIS really needs is a shareable (distributable), centrally-maintained database. In order to provide the greatest benefit, the database must be free of errors--even speed and security rank in importance after freedom from error. To achieve that status, mainframes and mainframe-supervised storage systems are now being designed as parallel structures, just as Tandem's *NonStop* systems have been since the company's inception. Functionally, the mainframe complex becomes a macrocosm "server," to the whole organization's "clients"--its end-user computing structure of distributed applications.

Between the distributable database and distributed applications must fit the distribution system. The present market for distribution systems is fragmented by conceptual approaches, stretching between, but embracing both computing and communications technologies. For instance, the distribution-systems market could be defined as "backbone," or "trunk networking." An example of this approach is IBM's SNA. It could also be defined as embracing Local Area Networks (LAN), such as Ethernet or Token Ring. Both of these definitions, however, are more comprehensively labeled as transport mechanisms. If we extrapolate this definition to a higher, value-added level, to include error-free information flow and fault tolerance with no single point of failure, then the focus of the definition shifts to on-line transaction processing systems that are also "transport mechanisms" of the **highest functional value**. In our opinion, among all direct and indirect competitors, Tandem serves best the distribution-systems market emerging between distributable databases and distributed applications.

A Tale of Two Strategies to Capture Shares of the Database Market From IBM

IBM's Position

IBM "owns" the corporate (central) database located within what the industry calls the "glass house," or data center with an estimated

60-80% of the mainframe market (either defined as all mainframe corporate users or the faster-growing IBM-OSC segment, respectively) and defends that market by deploying its most technologically-sophisticated interfaces. The distributable database product of choice for those users who are dedicated to implementing IBM technology will be DB2-Version 2, Revision 2. While this product has been "brought closely" into interaction with IBM's mainframe operating systems, it is not as tightly integrated to the operating system as the imbedded database managers employed on the System/38-AS/400, or even the OS/2-EE for PS/2. IBM's mainframe system software designers have left the "hooks" in place to implement a third-party relational database product such as Oracle's as a hub for those users sophisticated enough to build multivendor networks. We call this a "one-only" vendor plus "one-many" vendor strategy. One-many vendor installations facilitate the collection of already-installed equipment and applications under the umbrella concept of sophisticated users themselves building multivendor networks; nevertheless, IBM remains inevitably in control of the network complex and, therefore, in control of the primary account relationship.

The Most Sophisticated Interfaces

One of IBM's most complex, but valuable, interfaces is called LU6.2 (logical unit--an interconnection or protocol between parts of the whole system buried within system software; devices are called PU, or physical units in IBM's systems-software lexicon). It is this Systems Application Architecture (SAA) protocol that facilitates cooperative processing from mainframes (ES/3090S) and minimainframes (AS/400) out to microcomputers (PS/2).

The LU6.2 also allows the interconnection of equal-class computers, or computers that are not IBM system-software compatible, as "peers." In fact, LU6.2 is often referred to as the standard mechanism for peer-to-peer communications. It not only is the beginning of a computer-clustering facility, but one of the important vehicles for joining data, wherever physically located, to form the common corporate database. Another facility that should come along about 1991-1992 is 1GByte/sec. data pathway between physically separate, Expanded Storage units that will create the physical, high-speed storage that holds the collective distributable database.

Cooperative processing is IBM's term for distributable database/distributed applications computing. Some observers have also taken to calling it "cleaved applications," implying that applications can be split and assigned to whichever computer runs a particular task more efficiently, whether mainframe or microcomputer. An important corollary is that neither class of computer is exclusive, but rather mutually interdependent. This explains why IBM has ES/3090S products down to $500,000 price tags (and AS/400s under that), completing the full host product line, as well as large, medium and small PS/2s for distributed applications, including office-automation file servers for local support of distributable databases.

Manipulation of the distributable database is a critical example of cooperative processing. While the database manager (DB2 v.2 r.2) is itself classifiable as "system software," it is also an application. Consider the functions necessary to change the most important type of account in any enterprise: accounts receivable. Nothing is more important than having error-free records of who owes the firm money--it defines a critical dataset, both financially and legally.

Receivables collections can occur physicially anywhere within an organization and, increasingly, the posting updates to centrally-maintained records are sent to the "glass house" from somewhere other than central data processing. DB2 running on the mainframe is the controller of the update entry. It rolls out the particular receivables for update on immediate receipt of a collection (immediate--not overnight--because cash is tight). A remote microcomputer sends the entry updating the record from information contained in its OS/2-EE database maintained in local storage. Cooperatively with the mainframe, the DB2 "controller" confirms that all the changes have been received and are error free. Only on mutual assurance by the controller database and sender database that the change is right does the controller execute a "two-phase commit" to the change. If error is suspected, all changes are rolled out, the record is restored to its original status and retry is attempted. Imagine this process occurring with the entire corporate storehouse of information covering all aspects of the business for accounting or analysis--on line, i.e., immediately as changes occur. The key concepts are: all data is critical data; all databases must be error free; all communications and computing must be invulnerable to any single point of failure.

IBM does not yet offer fault-tolerant computing, but it is building toward that, certainly in the physical and logical construct of the central data base that is one of the company's three strengths. The second strength: IBM's mainframes (and within maybe a year, its minimainframes) are parallel-computing structures. The third strength: whatever IBM offers works or the company will fix it until it does. Customers can count on that, and at least 60% of the worldwide, commercial-computer market has.

Although computer industry analysts are captivated with the notion that there is some precision value in ratio calculations such as dollars per millions of instructions per second (MIPS), or dollars per transaction per second (TPS), real buyers make decisions according to a substantially broader set of functional criteria. IBM has rarely won a mano-a-mano competitive evaluation on the basis of MIPS or TPS, yet it wins business most of the time. Its systems overall run performance specifications that can be truly awesome. Mainframe systems can process thousands of transactions per second, and do so for the airline, car rental and hotel reservation database set of users. For banks and other organizations, running substantially more complex transactions and applications--one to three hundred transactions per second--is common. Minimainframe AS/400s, which were designed specifically for general-purpose computing have design targets of up to 600 interactive office automation users (at less than $1,000 per user) and up to 20 transactions per second at roughly $19,000 per transaction per second. We should keep these measures in mind as we compare the performance of Digital and Tandem equipment more accurately.

IBM will not only vigorously defend the "glass house"; it may, in fact, be experiencing a modest increase in market share through capture of new corporate account central sites from other, non-IBM Operating System Compatible (IBM-OSC) mainframe vendors worldwide. The Company is also aggressively moving out from the central site and downward throughout its client organizations, nowhere more aggressively than in the office automation market. Its new graphics-interface-based OfficeVision and 486 technology products are intended to be the "assault vehicles" used in the attempt to gain the leading market share position against the present number one, Digital's older, character-based All-in-1 and VAX/MicroVAX office systems.

Digital's Case

On the other hand, Digital has a 1990-1991 offensive in its own "battle plans." We see that company as focused, not only against IBM, but also the weaker positions of the non-IBM-OSC mainframe companies. Presently, somewhere between 50% to 60% of Digital's revenues are derived within the commingled IBM-Digital market place, so Digital is broadly engaged across the whole marketplace.

Digital's product lineup will soon include a Halloween announcement (and early 1990 release) of its largest ever, single processor system, the long anticipated *Aridus*, or VAX 9000 series. At 30 VUPS (VAX Units of Processing, or multiples of 1977-vintage VAX 11/780 processing power as unity), it should have more than four times the processing power of its previous, 1987-vintage top machine, the VAX 8810. The generally-accepted equivalent (relative to the performance of an IBM single-processor mainframe): one VUP equals approximately one-half IBM-equivalent MIPS (or Millions of Instructions Per Second). The *Aridus*, therefore, falls about midway between the IBM 3090 models 150S and 170S (at an estimated 13.5 MIPS and 17.5 MIPS, respectively). We expect the initial VAX 9200 single processor to be priced in the neighborhood of $1 million versus $1.65 million and $2.1 million for the IBM systems. Apparently, *Aridus* will have both multiple (initially two, eventually four) processor and dual-vector, add-in features. Much as with IBM mainframe design, *Aridus* will be offered directly to the commercial market as a transaction-processing computer, and, with the vector option, to the technical markets as a minisupercomputer. Single-processor, transaction-processing performance by *Aridus* is likely to fall into the low-end mainframe range of 50 to 100 transactions per second.

While an important upgrade step for the typical Digital installation, especially those employing VAXclusters in engineering and manufacturing support, the *Aridus* now pits Digital against an IBM product line that has already been in the market for four years and, at that, into the mainstream of its competitive preserve. Within two years, the single (seed) processor in an IBM parallel computer will advance from the 22 MIPS of the 3090 model 180S (at a clock of 15 nsec) versus the estimated 15 MIPS (at 16 nsec) of the *Aridus*, to an estimated IBM "Summit" clock of sub-10 nsec and probably more than 30 MIPS per seed processor. That specification could also jump to 50-60 MIPS if IBM introduces its modified RISC implementation on the first "Summits" expected to be ready for shipment in early 1992.

Digital's other new system release, already in volume shipment, is the VAX 6000 model 400 series, roughly comparable to Tandem's CLX 700 and IBM's AS/400 product lines. A more direct, Model 400-to-CLX 700 comparison is shown in Table 2.

The VAX 6000 concept differs, at least initially, from its time-shared, high-performance counterparts that we believe will be labeled VAX 9200 (or possibly, VAX 9000 model 200). While *Aridus* may "initially" be limited to two (2) processor configurations, the VAX 6000's are available in symmetrical strings of up to six processors per system. Someday, systems are likely to be ganged together into fault-tolerant, multiple-unit configurations. According to the rumor mill, Digital is preparing a low-end, VAX 6000, model-300-based, fault-tolerant design for market entry some time in mid-1990. We would compare it to the two-year old CLX 600 product line. Although the hardware path being followed is generally similar to that pioneered by Tandem, the approach taken by these two competitors is quite different beyond a superficial similarity.

The VAX 6000 models 420-460 series of two-to-six processor systems employ a symmetrical multiprocessing (SMP) version of Digital's VMS operating system that is still, in our opinion, somewhat limited in its functionalities. All 2-6 processor systems run under one copy of the operating system, but we detect no single-image capabilities yet. When next available, the separate processors can back one another up and each one accesses a common main-memory as well as a disk-storage pool; however, they cannot intermesh harmoniously to work on one large, unified problem (single-image). While not at all a limitation in performing multiple small jobs, the VAX 6000 cannot be compared, in aggregate, to a single-image-capable, multiprocessor mainframe for general-purpose computing.

The multiprocessor factor (MP) penalty is very much in line with that of multiprocessor mainframe systems from IBM and Amdahl, running 86% according to the statistics released by Digital shown in Table 2. IBM and Amdahl have released ballpark multiples of 80% to 90% as an MP factor. As Digital evolves beyond VMS

Table 2
Digital Equipment Versus Tandem Computers
New Midrange Systems Comparisons

Digital

VAX 6000	Model 410	Model 420	Model 430	Model 440	Model 450	Model 460
Number of Processors	1	2	3	4	5	6
CPU Performance						
VUP (2) - multiples of VAX 11/780	7	up to 13	up to 19	up to 25	up to 31	up to 36
Multiprocessor Factor (MP)	1.00x	1.86x	2.71x	3.57x	4.43x	5.14x
$ per VUP	$34,143	$30,692	$25,737	$25,080	$22,581	$20,889
Debit/Credit TPS	12.0	21.3	-	38.5	-	44.0
$ per TPS	$19,917	$18,732	-	$16,286	-	$17,091
Office Automation Timesharing Users (OATS)	216	368	-	600	-	656
$ per OATS	$1,106	$1,084	-	$1,045	-	$1,146
MAX I/O Bandwidth	60 MB/sec	60 MB/sec	60 MB/sec	60 MB/sec	40 MB/sec	40 MB/sec
Systems Price (VMS)	$239,000	$399,000	$489,000	$627,000	$700,000	$752,000

Tandem

CLX 700	720	740	760	780			
Number of Processors	2 (3)	4 (3)	6 (3)	8 (3)	(4)	(4)	(4)
Fault-Tolerant MP based on TPS	1.00X	2.00X	3.00X	4.00X			
TPS	7.4	14.8	22.2	29.6			
$ per TPS	$14,189	$13,851	$13,738	$13,682			
Systems Price Guardian 90 XL	$105,000	$205,000	$305,000	$405,000			

Footnotes

(1) More statistics are provided for Digital's 6000 than for the Tandem 700 because the 6000 is intended to be general purpose and also is aimed at both the technical and commercial markets, while the 700 is focused on transactions.

(2) VUP - VAX Units of Processing; a proprietary measure of performance that only has meaning in comparisons within the VAX product line.

(3) Each CLX 700 CPU consists of two cross-coupled processors functioning as a single central processing unit for continuous fault checking in OLTP applications, thus featuring both hardware and software fault tolerance characteristics.

(4) For increasingly larger systems, multiple numbers of individual Tandem systems may be interconnected via functionalities under the Guardian 90XL operating system.

version 5.x, single-image capabilities may be available, we guess within (or just beyond) the next two years.

In contrast, Tandem systems are not intended to be symmetrical multiprocessors. Each computer in a CLX or VLX system is a discrete processor complex with its own main-memory in each computer's configuration. Each works on a discrete application, which may be either repetitively the same, or entirely different per computer. Each additional processor complex added to (or within) an interconnected CLX or VLX system adds one full unit of processor power to the aggregate system's throughput. Tandem transaction-processing systems are at the leading edge of networked OLTP applications. Tandem has published audited test results showing that when networking overhead is added to the processing complex, the Tandem "effective" MP factor is reduced to 90% from 100%. To our knowledge, no other system has yet achieved Tandem's functionalities via networked OLTP capabilities, so it is impossible to assess at this time what additional "effective penalty" networking would apply to either a Digital VAX or IBM 3090 complex. The remaining (and as yet unexplained) issue raised by the Digital data in Table 2 is the curious 33% decline in internal bus bandwidth for models 450 and 460 versus models 410-440.

The Critical Issue: When Will New Software Functions Be Released?

Digital's VMS 32-bit proprietary operating system was derived from its predecessor, the RSX-11M 16-bit, time-sharing progenitor. It has evolved away from its exclusively time-sharing roots to become a mid-range system supporting a production-data-processing operating system, but as yet without built-in (imbedded) OLTP functionalities. VMS, in our opinion, is being backed into an on-line transaction supporting role as an adjunct to its dual, general-purpose, commercial and technical computing focus.

IBM achieved a similar technological evolution in the late 1970's and overlayed onto its commercial, virtual-memory mainframe, production data-processing operating system, a communications and transaction processing monitor (basically a second specialized operating system) called CICS (Customer Information Control System). IBM's mainframes have always had the processor horsepower to handle all of the system software necessary to fulfill simultaneously a complete set of diverse, system-software and applications requirements. Clearly one of these is allowing the user to build systems with multiple operating systems and multiple database managers of choice, while intermixing input systems of all kinds (e.g., "dumb" terminals, PS/2s, cash machines, factory-floor devices). With the *Aridus's* performance potential, Digital has a computer that is entering, for the first time, the lower part of that performance region.

Presently, Digital offers two separate monitors, ACMS (Application Control and Monitoring System) which was developed in-house and introduced in 1984, and DECintact (which was acquired and added to the product line on 1988). Why two? We view ACMS as an add-on-function, transaction-processing monitor to a general purpose departmental system. DECintact had all the features that Digital would need to develop into a future monitor, like automatic host rollover (which backs up a failing computer without losing transactions) and it was available, having been written for VMS. Therefore, it was acquired. DECintact presently works with Digital's higher-performance, RMS flat-file manager, but will be improved to interface to Rdb (Digital's relational database manager). We estimate that this development will take one to two years. A future (DECintact) monitor with ACMS integrated characteristics will be merged into (and imbedded within) a future version of VMS. We estimate this development will take two to five years. Within one year, Digital's present queueing-transaction protection approach (a simple commit format) should be upgraded to full networked transaction protection using a two-phase commit.[2]

Some observers believe that a multiplicity of monitors, promised developments and the early state (but improving via Digital summer school retraining) of in-house marketing and sales knowledge of OLTP is confusing the targeted customers who are outside of a company's captive, departmental-computing installed base.[3] Within its ancillary OLTP products approach to date, Digital has, however, managed to build one of the larger transaction-processing niche businesses in the industry. We estimate it to represent 15% of revenues.

Much of the DECtp story is either new, with new computers, or in the "to be introduced" status, as with software and features. For example, the *Aridus* is being readied for release without the benefit of field (or beta) test time, which may slow the product's initial acceptance and volume-shipment rates for possibly up to a year. Much of the company's VAX architecture approach already available to DECtp designers is well conceived. VAX products easily lend themselves, to modular requestor (client)-server-storage management designs. These evolutionary steps, however, will consume a great deal of time.

[2]According to Dave Zwicker, Digital OLTP Consultant Relations Manager, a "Two phase commit, simply put, lets you know that the transaction has reached the database, or file structure, by coming back to the transaction generator from the repository with a commit message. It's more reliable than a queue management message because the latter goes only to the queue, not to the database. Every other functionality now in the monitors, with the exception of transaction control and applications-development style, will eventually become part of VMS, or a kernel of VMS. Thus, from Digital's OLTP effort, VMS will gain capabilities for recovery, journaling, queuing and remote procedures." *DEC Professional*, Vol. 8, No. 3, March, 1989.

[3]Reference: *Datamation*, "Back To The Drawing Boards," Vol. 35, No. 13; July 1, 1989, pp 57-61.

William R. Demmer, Digital Vice President for Mid-Range Systems, was recently interviewed by *Digital Review* (May 22, 1989) and quoted as saying:

> The goal I'm working toward is to replace that nice, single time-sharing system we all know, love and understand with a database that is available to the whole enterprise. We'd like to replace the entire organization's computing capability and still have it look the same from the desktop as it did when it was based on a time-sharing architecture. We are between five and 10 years away from providing that transparency.

Digital appears to be planning for its capturable customers a strategic conversion from both IBM database sites (and even entire businesses still dependent on non-IBM-OSC mainframe systems) into VAX/VMS based OLTP and production data-processing systems. Over the next two years its principal competitive edge appears to us to be cost-per-transaction and system lifetime cost-of-ownership advantages. While the non-IBM-OSC companies may have installed-base vulnerabilities, the type of customer of these companies was not necessarily the swiftest to implement new DP features and the most eager to spend lots of new monies on system change. The counter-IBM strategy has been made more difficult because IBM has made all the right moves, like SAA, and few if any mistakes in its defense of the "glass house", other than the elongated introduction and implementation phases of its new software-based features, especially against incursion and substitution by an attacker coming up from the departmental-computing direction. While Digital's position has uncanny similarities to Tandem's already successful strategy, Tandem is well into its execution phase while Digital, in our opinion, has at least another year of DECtp development, internal education, and transition to complete. But complete it they will, and Digital too will once again be an OLEC force for competitors to reckon with.

<u>Tandem's Case</u>

Tandem has crafted what we believe is a more sophisticated (and potentially more successful) strategy of competitive coexistence within, and then capture of, a segment of the IBM corporate (central) database market. Tandem-as-competitor brings not only different, but rather the **highest functional values** and performance to the market for distributed transaction and query systems, stretching from the "glass house" interface to the system's entry-points. Tandem has crafted (from **in-house development** efforts) connections to IBM's host complex as a full-functioning host, or peer, in both the developed IBM-SNA connectivity technologies as well as the emerging, international OSI connectivity technologies. It has developed and offers a superior relational database product that is **tightly integrated** to the Tandem *Guardian 90* operating system. *NonStop SQL* has just been substantially upgraded to support on-line production data processing as well as on-line transaction and inquiry processing. Given its development thrust upward into mainframe-class technologies, Tandem will undoubtedly extend its superior, message-based operating system technology via asyncronous, high-speed fiber-optical, data-pathway technology to cover an entire organization's distribution-system requirements. *Guardian 90XL* now interconnects multiple levels of Tandem's computer-systems products (CLX, VLX, as well as the soon-to-be-released "Cyclone") to the Tandem-supported part of the critical database, and then the Tandem host to the IBM host; Tandem will likely also be moving quickly in the direction of "nonstop-to-the-desktop" local-area-networking technology at the user-level (or entry marketplace) now that it has absorbed its acquired Ungermann-Bass product lines and development efforts.

At present, Tandem coexists with either of IBM's one-only or one-many strategies, but market-share capture opportunities can (and should) coalesce toward the one-one alternative, in our view, based on a comprehensive product edge over all of its competitors; Tandem probably accomplishes what Digital hoped to do--not through replacement, but through attachment. Over time, Tandem may be one of the few vendors able to leverage its present strategy into a proprietary-user base and it should be extremely successful, over both the short and long term, in developing the distribution arms of big-system implementations through the development of what we are calling the **distribution-systems** market. In our opinion, Tandem's enhanced financial performance depends on completion of the current product transition, likely in 4Q 1989.

The Evolution of Tandem

Matrix 1 displays the salient differences that we see now in the Tandem-versus-Digital product positions:

Matrix 1

Tandem Versus Digital

Product Positions

Tandem Product	Our Comments	Digital Product	Our Comments
(1) "Cyclone"	2nd generation ECL upgrade of VLX. Intensive I/O upgrade.	"Aridus"	2nd generation ECL upgrade of VAX 8000. upgrade of VAX 8000. Intensive computational (engineering) upgrade.
(2) NonStop SQL	Integrated DB into Guardian XL90.	Rdb	Separate DB module now offered for free, not integrated into VMS.
(3) IBM-TDM	SNAX/CDF LU 6.2/PU 2.0 Subhost (highest connectivity) developed by TDM.	IBM-DEC	Mostly third-party efforts directed by customer base.
(4) FOX II	Very high speed fiber optical backbone (trunk).		
(5) EXPAND	Software interlink of up to 255 NonStop systems.		
(6) CLX 700/600	Hardware AND Software fault-tolerance.	VAX 6000	Redundant symmetrical multiprocessing; physical partitioning only.
(7) Access/One	"Nonstop to the Desktop" under development. Full user-to-error free database connectivity.		
(8) Applications Architectures	Tandem Integrated Mfg. Wholesale Banking, and Intelligent Networking--with leading-edge System integrator partnerships.		

Tandem's initial *NonStop* product line, released in 1978, introduced "fault-tolerance," a new concept, to the computer-systems marketplace. Much like similar technologies employed in the telephone-equipment marketplace, *NonStop* computers bypass failing components and subsystems. Add-on boards expand so that system elements can be installed (or boards removed, repaired, and replaced) while the system remains in operation. Because it need never be shut down, the system operates literally "non-stop."

What was the underlying selling point (and primary benefit) to the user's organization that sold Tandem's *NonStop* systems quickly, especially to the sophisticated, large corporate audience? *NonStop* was the first system ever to generate and support an "error-free" database. Only when error-free could the database be considered a high-value corporate asset (and, therefore, an asset to the MIS chief's career development) and become the new foundation for competitively strategic systems.

Error-free databases and OLTP are mutually dependent concepts at the core of strategic systems technology. Transaction processing could only be considered on-line (that is real-time) if the transport and computing system never failed completely. Further transactions cannot be processed (while the system remains on-line) unless the results of processing are completed and made available literally as fast as the transaction occurs. Furthermore, the processing of transactions in real time was only of value so long as there was the highest level of confidence that the complete information repository was virtually error-free. It is impossible to have true OLTP without all of those factors as imbedded functionalities in the product offering. Anything less is not OLTP, and what OLTP is all about is **confidence**. Why else would customers literally "walk up to a wall," ask for cash, and accept the accounting of their bank accounts? Why else would the bank put cash "in the wall" unless all parties were **100% confident** in the underlying system?

Tandem's *NonStop* has a collateral advantage: capacity to process more transactions expands linearly with the addition of more processors. *NonStop* represented one of -- if not the industry's first -- loosely-coupled, parallel-computer systems. (The addition of one processor to a two-processor installation increased the unit of processing capacity from two-fold to three-fold; this is the computer-systems definition of linear expansion.) Only processing power expands, since nothing else changes. The same systems and applications software, plus peripherals, run on the expanded system, whether upgraded with more of the same processors, or with higher-power processors up the product line.

Tandem engineers had actually designed a new **networking technology**. At its core, Tandem's *Guardian 90* proprietary operating system is a messaging structure, one that includes the *NonStop* system of multiple computers and each constituent computer as well as the external network from entry points to database. The subsystems within the computer are tested by "health" messages. "Health" messages are also sent between computers, under the supervision of each computer's copy of the *Guardian* operating system. Transactions and queries are themselves also messages. New service opportunities such as automatic cash machines and tellerless banking offices became practical. Tandem quickly became accepted within the international banking community, later by the business community at large. In a computer-architecture sense, Tandem captures interfaces from the entry device to the "glass house" connection. In the eyes of corporate-user decision-makers, the stage was set early for "friendlier" competition (Tandem) with the host vendor (IBM) than those who would choose to conduct a frontal assault on the host market itself (Digital). Tandem was (and still is) unique; the niche that it opened has not only become substantially larger, but is expanding subsequently into the mainstream of both commercial and technical data processing. The issue is no longer acceptance, rather Tandem's ability to continue successful execution of a leading-edge product strategy and maintain competitive differentiation.

DEFINITION OF SERVICES PERFORMED

On-Line Transaction Processing

On-line transaction processing (OLTP) is generally accepted as characterizing a computerized mode of applications processing in which messages received from systems-entry devices are processed into a common database (information repository) in real time (without apparent delay). Messages can be either transactions or queries. A transaction changes records contained within the database. The response to a query reports, but does not change information contained within the database. Complex transactions involve the simultaneous change of a large number of records, while simple transactions involve only a few. This simultaneity complicates the interpretation of simplistic measures like number of transactions performed per second (TPS). Organization of database information (records, files) conforms to two basic types of structures: hierarchical, or tree-structure (communicates rapidly); and relational, flat or table-structure (slowest, but easiest to construct and getting faster with increasing availability of computer-hardware power).

Transaction-processing-rate benchmarks are timings of the TPS. Attempts have been made by both vendors and independent agencies to derive useful product comparisons from competitive trials. While the task would seem to be relatively simple--measure executions and divide by seconds--variations in task criteria, systems configurations, differing device characteristics per vendor even within similar systems configurations, and all manners of other variations make the idea of precise comparisons misleading, however seductive. We suggest that transactions per second (TPS) as a way of comparing systems performance between different vendors be interpreted only in a very general sense. In fact, such statistics tend to be useful, in our opinion, only when arraying a single vendor's different classes of products for pricing purposes.

If transaction and query messages are at the heart of OLTP, communications computer power is more important than MIPS (millions of instructions processed per second). The *Guardian 90* operating system, which is messaging based, is efficient at the process of queuing (sequencing) a very large number of concurrent jobs. As Tandem releases more powerful *NonStop* computers, with full capacity--not only for the execution of transactions and queries--but also for on-line "batch" or production support, the Company should finally be accepted as a fully-qualified, production-class vendor and be leveraged up into a larger marketplace. That should be a bigger "ball game" for Tandem. In our opinion, that development will also more clearly differentiate *NonStop* market-share capture opportunities versus its most direct competitors. Tandem satisfies mission-critical applications requirements as an essential stepping-stone to OLEC while Digital is still, in our opinion, focused on departmental computing applications, which are not clearly evolving toward OLEC. Tandem should also capture market share against IBM as commercial users move critical database, general-purpose applications onto *NonStop* mainframe systems.

THE INTERNAL CONNECTIVITY

The Tandem Vision

Messages are passed from the application through the file system and disk process to the database in the operation of any one computer within a *NonStop* system. A *NonStop* systems architect recently said that the Company had dedicated 14 years of R&D to make the message interface between an application and the files as "skinny" (efficient) as possible. The message interface layer and system-software structure within the *NonStop* system, consisting of from two to 16 separate computers and then two banks of up to 16 computers in fault-tolerant configuration, contribute the interconnection between those computers, in effect defining the *NonStop* system. Internal systems communications takes place over dual (parallel) Dynabuses, which extend conceptually outward to interconnect entry devices to the *NonStop* and also interconnect *NonStop* as a "peer host" to the IBM mainframe "glass house."

Tandem's messaging system consists of the "work" of its computers plus "health" messages. This latter aspect of the system's design is what facilitates *NonStop* fault-tolerance. As each computer processor's power expands through design cycles, and "health" messaging remains a relatively constant load factor, *NonStop* systems' performance and functionality expand beyond on-line transaction processing and query handling to on-line batch or production processing. Thus, the normal evolution of processor development should facilitate transition from front-end niche to general-purpose data processing. In our opinion, Tandem engineering is fully capable of advancing the state-of-the-art in both internal and external communications technology, quickly moving forward with the transition from typical minicomputer, synchronous-bus operation to high-speed, asynchronous "mainframe-class" (intersystem) communications, based on fiber-optical technology.

Tandem hopes to make a fully successful transition from its original fault-tolerance-based niche market position to becoming a vendor of products with systemwide, modular expandability and a high availability of error-free data. Such an achievement should propel the company toward capturing a unique, Tandem-installed proprietary base. In the pursuit of this transition, it is important to note that the proprietary (non-standard) nature of the company's *Guardian 90* operating system technology is not an issue in the valuation of *NonStop* technology by the marketplace. Criticisms of Tandem and *NonStop* do, however, exist. *NonStop* requires (or at least has required) some degree of application-code "customization." *NonStop* systems are loosely coupled versus the tight coupling achieved by symmetrical multiprocessing systems. Loose coupling requires that each computer have its own, not a shared copy of the operating system, and systems employing that configuration are more difficult to load-balance.

Tandem responds that symmetrical multiprocessing doesn't really help in solving the OLTP applications problem of queuing (sequencing) a large number of concurrent jobs. This is a communications problem that *NonStop* solves very well. *Guardian 90XL* (and ultimately Tandem's interpretation of UNIX) are being made as transparent to the user as possible, so that other than license-fee replication, obtaining multiple copies of the operating system should not cause a performance problem. While OLTP does not present a processor-power problem, but rather a communications-intensive task, messaging tasks are what *NonStop* does best by its intrinsic design: more robust processing power results from the user's ability to intermix small (CLX 600 series), medium (CLX 700 series), and large (VLX 800 series) *NonStop* computers within the same system. Users will soon have very large "Cyclone" systems to choose as well. Units of work will be "scaled" (matched) to a systems' capability. *Non-Stop* will, therefore, feature scalability and load-balancing features common to parallel processing. Ultimately, systems software should be transparent to the end user (by reducing or eliminating requirements to consider Tandem's systems in any way different from any other computer--i.e., requiring little or no customization of applications software). Conventions such as "loosely" or "tightly coupled" should have no practical operating meaning to *NonStop* users as an impled limitation versus other vendors' symmetrical multiprocessor products (e.g., Digital VAX 6000 series).

Now that Tandem offers (and customers are implementing) *NonStop SQL* databases, applications programming is based on knowledge of two industry-standard programming languages, COBOL and SQL. Availability of Tandem's systems has, therefore, passed from the difficult-to-program state to the substantially larger universe of industry-standard, commercial-applications programmers. This is one of the **BIG** changes in the Tandem story.

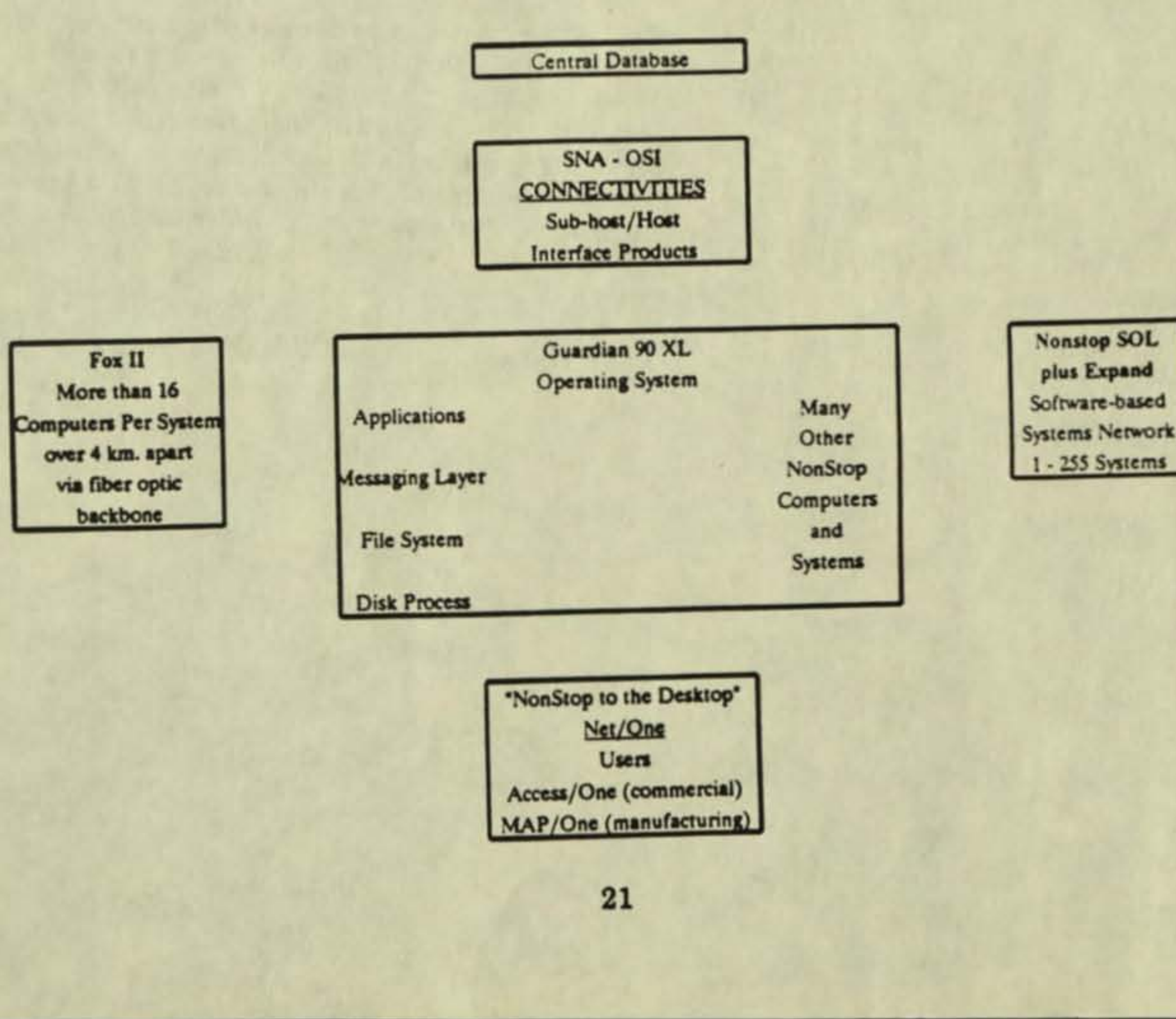
THE EXTERNAL CONNECTIVITY

Software Stands Out

While messaging is the internal and external "business" of Tandem's *NonStop* computers, today, new, more powerful computers (plus the ability to intermix and scale the entire spectrum of the processor product line into a single-system configuration) will make the company's "business" messaging plus production-data processing. The continuous, internal-external nature of *NonStop* differentiates Tandem from Digital.

Diagram 1 displays the structure and names of the present systems software and communications/connectivity product line. Externally, the messaging-transport capabilities extend the *NonStop* system up to the "glass house" interface products, to Tandem's own integral SQL-based database management system and outward via "nonstop-to-the-desktop" levels under its Access/One product banners. This is the "seamless" network that all observers have envisioned as the preferred system structure of the future.

Diagram 1
Tandem Computers
Completed Product Structure
As of Fiscal 1989



While Digital is being pushed by clients into IBM-coexistence/connectivity, in most cases through third-party products employed on already-installed VAX systems, Tandem willingly embraced the concept of IBM-coexistence/connectivity. It has developed and offers a broad range of the deepest (highest level) SNA and OSI interface products. Worth noting as a fundamental difference between the two companies: Tandem had the vision to take this position willingly, while Digital, in our opinion, is changing in order to protect its installed base. Tandem, therefore, represents our real growth story.

CONCLUSION

Tandem is like the high surfer on a wave that is turning into a tsunami (tidal wave). The only question is not whether the wave is going to get bigger, but whether or not Tandem can handle the wave. The wave has been a long time coming and we believe that Tandem management can indeed handle that wave.

Printed By: ZINKER_SELMA @TSII
SENT: 87-05-01 06:08
FROM: MELENDEZ_WILL @DENMARK
TO: ZINKER_SELMA @TSII
SUBJECT: More Authors....
Forward of: 87-05-01 05:58 FROM MELENDEZ_WILL @DENMARK : Authors...

Selma,

In my earlier message , the date of part II was: Volume 5,number 2 1986.

Both issues were published around OCT. '86/Nov. 86.

Will

------------------------- REPLY ATTACHMENT -------------------------

SENT: 87-05-01 05:58
FROM: MELENDEZ_WILL @DENMARK
TO: ZINKER_SELMA @TSII
SUBJECT: Authors...

The article name: "THE UPPER LAYERS OF THE ISO/OSI REFERENCE MODEL"

Part I. published in Volume 5,number 1 of 1986.

Part II. published in Volume x,number x of 1987.

The Journal : COMPUTER STANDARDS & INTERFACES
published by North Holland

The Authors: WILFRED A. MELENDEZ (I am now at TANDEM)

Erik Lorenz Petersen (NOT a TANDEM Empoyee)

Will Melendez

------------------------- ORIGINAL ATTACHMENT -------------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------- TEXT ATTACHMENT -------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.
Selma Zinker

```
Printed By: ZINKER_SELMA @TSII
   SENT: 87-05-01  05:58
   FROM: MELENDEZ_WILL @DENMARK
     TO: ZINKER_SELMA @TSII
SUBJECT: Authors...
In Reply to: 87-04-28  14:49  FROM ZINKER_SELMA : 2:Tandem authors

The article name:  "THE UPPER LAYERS OF THE ISO/OSI REFERENCE MODEL"

      Part I. published in Volume 5,number 1 of 1986

      Part II. published in Volume x,number x of 1987.

The Journal :    COMPUTER STANDARDS & INTERFACES
                   published by North Holland

The Authors:    WILFRED A. MELENDEZ  (I am now at TANDEM)

                Erik Lorenz Petersen (NOT a TANDEM Empoyee)

Will Melendez

------------------------ ORIGINAL  ATTACHMENT ------------------------
   SENT: 87-04-28  14:49
   FROM: ZINKER_SELMA
     TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT  ATTACHMENT ----------------------------
   SENT: 87-04-28  14:47
   FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors.
If you have had a paper published please respond to this message by
listing the title, journal name and date of the issue in which it
appeared.
  Thanks for the cooperation.
   Selma Zinker
```

9-61 to 9-62
Graduate Student University of California
Berkeley

MSEE

9-59 to 6-61
Undergraduate Student San Jose State University

BSEE

3-56 to 9-59
Customer Engineer IBM Corporation, San Francisco

Field Service Representative in San Francisco

2-54 to 3-56
Student San Francisco City College

AA Degree in Electronics

Tom Collins
(Collins_TW)

{new

PUBLICATIONS - A
---

1. 1964 Thomas W. Collins. ANALYSIS OF A ONE-TUNNEL-DIODE BISTABLE CIRCUIT WITH A COUNTING PROPERTY. Solid State Design, pp. 27-31.

2. 1969 Thomas W. Collins. TWO DIMENSIONAL ANALYSIS OF A BIPOLAR DEVICE. IEDM Digest.

3. 1969 Thomas W. Collins. COLLECTOR CAPACITANCE VERSUS COLLECTOR CURRENT FOR A DOUBLE-DIFFUSED TRANSISTOR. Proc. of the IEEE, Vol. 57, No. 5, pp. 840-841.

4. 1969 Thomas W. Collins. MINORITY CARRIER INJECTION INTO THE EMITTER-BASE JUNCTION SIDEWALL REGION OF A DOUBLE-DIFFUSED BIPOLAR TRANSISTOR. IEDM Digest, 43 pages.

5. 1973 Thomas W. Collins. EXACT MODELING OF TIME-DEPENDENT PHENOMENA IN AN MOS STRUCTURE. IEDM Digest, pp. 342-345.

6. 1974 F. E. Holmstrom, T. W. Collins, and J. N. Churchill. MEASURED RADIATION EFFECTS IN MOS CAPACITORS WITH A PROPOSED NEW MODEL. Applied Physics Letters, Vol. 24, No. 10, pp. 464-466.

7. 1974 John N. Churchill, Thomas W. Collins, Fred E. Holmstrom. ELECTRON IRRADIATION EFFECTS IN MOS SYSTEMS. IEEE Transactions on Electron Devices, Vol. ED-21, No. 12, pp. 768-777.

8. 1975 Thomas W. Collins. EXACT MODELING OF THE TRANSIENT RESPONSE OF AN MOS CAPACITOR. IEEE Transactions on Electron Devices, Vol. ED-22, No. 3, pp. 909

9. 1975 Thomas W. Collins, J. Gazdag, and K. Ju. STABILITY OF PARALLEL STRIPE DOMAINS. IEEE Transactions on Magnetics, Vol. Mag-11, No. 5, pp. 1088-1090; Proc. 13th Int. Magn. Conf., 1088-1090.

10. 1976 Karl R. Hense and Thomas W. Collins. LINEAR CHARGE-COUPLED DEVICE SIGNAL-PROCESSING TECHNIQUES. IEEE Journal of Solid- State Circuits, Vol. SC-11, No. 1, pp. 197-202; and IEEE Transactions on Electron Devices, Vol. ED-33, No. 2, pp. 265-270.

11. 1976 J. N. Churchill, F. E. Holmstrom, and T. W. Collins. A LINEAR TWO-LAYER MODEL FOR FLAT-BAND SHIFT IN IRRADIATED MOS DEVICES. Solid-State Electronics, Vol. 19, No. 4, pp. 291-296.

12. 1976 T. W. Collins and R. W. Cole. EFFECTS OF ABRUPT CHANGES IN FILM THICKNESS ON MAGNETIC BUBBLE FORCES. IBM Journal of Research and Development, Vol. 20, No. 2, pp. 132-137.

13. 1978 M. S. Cohen, G. W. Beall, M. H. Kryder, N. J. Mazzeo, and T. W. Collins. INFLUENCE OF NiFE QUALITY ON BUBBLE DEVICES. IEEE Transactions on Magnetics, 14 pages.

14. 1978 T. W. Collins, J. N. Churchill, F. E. Holmstrom, and A. Moschwitzer. MODELING OF THE TRANSIENT RESPONSE OF AN MIS CAPACITOR. Advances in Electronics and Electron Physics, Vol. 47, pp. 267-329.

15. 1979 M. S. Cohen, G. W. Beall, M. H. Kryder, N. J. Mazzeo, and T. W. Collins. INFLUENCE OF NiFE QUALITY ON BUBBLE DEVICES. Journal of Applied Physics, Vol. 50, No. B3, pp. 2201-2203. (See also 13 A)

16. 1979 J. N. Churchill, F. E. Holmstrom, and T. W. Collins. DYNAMIC MODEL FOR E-BEAM IRRADIATION OF MOS CAPACITORS. Journal of Applied Physics, Vol. 50, No. 6, pp. 3994-4002.

17. 1979 T. W. Collins, F. E. Holmstrom, and J. N. Churchill. CHARGE DISTRIBUTIONS IN MOS CAPACITORS FOR LARGE IRRADIATION DOSES. IEEE Transactions on Nuclear Science, Vol. NS-26, No. 6, pp. 5176-5179.

18. 1980 R. D. Holmes, L. J. Tao, and T. W. Collins. DESIGN AND OPERATION OF 8-MICRON PERIOD BUBBLE DEVICES. IEEE Transacions on Magnetics, Vol. MAG-16, No. 5, pp. 858-860.

19. 1980 T. W. Collins and R. W. Cole. A MAGNETIC-BUBBLE-DEVICE SIMULATOR. IEEE Transactions on Magnetics, Vol. MAG-16, No. 5, pp. 846-851.

20. 1982 J. N. Churchill, F. E. Holmstrom, and T. W. Collins. MODELING OF IRRADIATION-INDUCED CHANGES IN THE ELECTRICAL PROPERTIES OF METAL-OXIDE-SEMICONDUCTOR STRUCTURES. Advances in Electronics and Electron Physics, Vol. 58, pp. 1-79.

PATENTS
----

21. 1971 Thomas W. Collins. RESISTIVELY ISOLATED INTEGRATED CURRENT SWITCH. United States Patent # 3,569,800.

22. 1979 Thomas W. Collins and Karl R. Hense. OPERATING CIRCUITRY FOR SEMICONDUCTOR CHARGE-COUPLED DEVICES. United States Patent # 4,156,818.

23. 1979 Thomas W. Collins. SERRATED Y-BAR MAGNETIC BUBBLE SWITCH. United States Patent # 4,175,289.

24. 1980 Thomas W. Collins. MAGNETIC BUBBLE PASSIVE REPLICATOR. United States Patent

25. 1981 Thomas W. Collins and Kay B. Mehta. MAGNETIC BUBBLE Y-BAR CORNER. United States Patent # 4,261,045.

26. 1981 Thomas W. Collins. HIGH PERFORMANCE BUBBLE CHIP ARCHITECTURE. United States Patent

{new

TECHNICAL REPORTS - F
----

1. 1963 Thomas W. Collins. A ONE-TUNNEL-DIODE BISTABLE CIRCUIT. IBM Technical Disclosure Bulletin.

2. 1965 Thomas W. Collins. INHIBIT SCHEME FOR A 2-D MEMORY. IBM Technical Disclosure Bulletin.

3. 1966 Thomas W. Collins. ELIMINATING MAGNETOSTRICTIVE RINGING. IBM Technical Disclosure Bulletin.

4. 1967 Thomas W. Collins. EFFECTING HIGH COLLECTOR DOPING GRADIENTS. IBM Technical Disclosure Bulletin.

5. 1967 Thomas W. Collins. TRANSISTOR-COLLECTOR CLAMP. IBM Technical Disclosure Bulletin.

6. 1967 Thomas W. Collins. TRANSISTOR STRUCTURE. IBM Techinal Disclosure Bulletin.

7. 1970 Thomas W. Collins. VARYING DELAY OF INTEGRATED LOGIC CIRCUITS. IBM Technical Disclosure Bulletin.

8. 1970 Thomas W. Collins. MOS HALL DEVICE. IBM Technical Disclosure Bulletin.

9. 1970 Thomas W. Collins. INTEGRATED READ PREAMPLIFIER. IBM Technical Disclosure Bulletin.

10. 1970 Thomas W. Collins. HALL DEVICE AND PACKAGE FOR MAGNETIC SENSOR. IBM Technical Disclosure Bulletin.

11. 1970 Thomas W. Collins KEYBOARD USING HALL EFFECT DEVICES. IBM Techinical Disclosure Bulletin.

12. 1970 Thomas W. Collins. METHOD OF ELIMINATING D.C. OFFSET VOLTAGE. IBM Technical Disclosure Bulletin.

13. 1971 Thomas W. Collins. MOS FET HALL EFFECT DEVICE FEEDBACK CIRCUIT. IBM Technical Bulletin.

14. 1971 Thomas W. Collins. METHOD OF INCREASING AND AMPLIFYING SIGNALS. IBM Technical Disclosure Bulletin.

15. 1972 Thomas W. Collins. LOW NOISE-HIGH FREQUENCY BIPOLAR TRANSISTOR. IBM Technical Disclosure Bulletin.

16. 1973 Thomas W. Collins. TIME DEPENDENT NUMERICAL ANALYSIS OF METAL- OXIDE-SEMICONDUCTOR DEVICES. Ph. D. Dissertation, UCD, 254 pages.

17. 1976 B. R. Brown and T. W. Collins. SINGLE-LEVEL COLUMN ACCESS STRUCTURE FOR BUBBLE LATTICE FILE. IBM Technical Disclosure Bulletin.

18. 1976 T. W. Collins. GAITING A BUBBLE FROM A LATTICE TO AN ISOLATED REGION. IBM Technical Disclosure Bulletin.

19. 1977 T. W. Collins. CHANGING LOCAL BIAS FIELDS IN MAGNETIC BUBBLE DEVICES. IBM Technical Disclosure Bulletin.

20. 1977 T. W. Collins. H-BAR TRANSFER SWITCH. IBM Technical Disclosure Bulletin.

21. 1978 T. W. Collins. DRIVE-TO-TRANSFER STRIPPING TRANSFER SWITCH. IBM Technical Disclosure Bulletin.

22. 1979 T. W. Collins. SERPENTINE CONDUCTOR TRANSFER SWITCH. IBM Technical Disclosure Bulletin.

23. 1979 T. W. Collins. METHOD FOR OBTAINING MINI-BLOCKS ON A BUBBLE CHIP. IBM Technical Disclosure Bulletin.

24. 1980 T. W. Collins. ASYMMETRIC Y-BAR SWITCH. IBM Technical Disclosure Bulletin.

25. 1981 T. W. Collins. CURRENT-CONTROLLED DISK REPLICATOR. IBM Technical Disclosure Bulletin.

26. 1981 T. W. Collins. A SWITCH DESIGN FOR BIDIRECTIONAL PROPAGATION. IBM Technical Disclosure Bulletin.

27. 1981 T. W. Collins. BUBBLE PROPAGATOR FOR HIGH DENSITY CIRCUIT. IBM Technical Disclosure Bulletin.

{new
ORAL PRESENTATIONS - G

1. 1973 Thomas W. Collins. EXACT MODELING OF TIME-DEPENDENT PHENOMENA IN AN MOS STRUCTURE. 19th International Electron Devices Meeting, Washington, D. C., December 3-5, 1973. (See also 5A)

2. 1975 Thomas W. Collins, Jeno Gazdag, and Kochan Ju. STABILITY OF PARALLEL STRIPE DOMAINS. 13th International Magnetics Conference, Imperial College of Science and Technology, London, England, April 14-17, 1975. (See also 9A)

3. 1979 T. W. Collins, F. E. Holmstrom, and J. N. Churchill. CHARGE DISTRIBUTIONS IN MOST CAPACITORS FOR LARGE IRRADIATION DOSES. IEEE Annual Conference on Nuclear and Space Radiation Effects, Santa Cruz, CA., July 17-20, 1979. (See also 17A)

4. 1980 R. D. Holmes, L. J. Tao, and Thomas W. Collins. DESIGN AND OPERATION OF 8 MICRON PERIOD BUBBLE DEVICES. International Magnetics Conference(Intermag '80), Boston, Mass., April 21-24, 1980. (See also 18A)

5. 1980 Thomas W. Collins and Robert W. Cole. MAGNETIC-BUBBLE-DEVICE SIMULATOR. International Magnetics Conference(Intermag '80), Boston, Mass., April 21-24, 1980. (See also 19A)

{new

ACCOMPLISHMENTS OF IMPORTANT SCIENTIFIC OR ENGINEERING WORK

11-69	Two-Dimensional Analysis of Bipolar Transistors.
3-75	First Complete Time-Dependent Analysis of MOS Capacitors.
9-80	First Published Interactive Magnetic Bubble Device Simulator.
1979	First Complete Analysis of Radiation Effects of MOS Devices.

TEACHING EXPERIENCE

1965-1969	Engineering Education:	Asst. Professor-San Jose State University
1971-1978	Engineering Education:	Engineering Education-IBM Corporation, San Jose

COLLEGE RELATIONS

1971-1981	Engineering Recruitment:	Corporate Engineering Recruitor, IBM Corporation San Jose, CA.
1981-Present	" "	Corporate Engineering Recruitor, Tandem Computers Cupertino, CA.

PROFESSIONAL AWARDS

1969	Outstanding Contribution Award	IBM Corporation, San Jose
1980	Third Level Invention Award	IBM Corporation, San Jose
1981	Senior Membership	IEEE

PROFESSIONAL ENGINEERS LICENSE

P. E. State of California # E 6702

PROFESSIONAL SOCIETY AFFILIATIONS

Senior Member	IEEE
Member	Eta Kappa Nu
Member	Sigma Xi

REFERENCES

DR. C. Denis Mee	IBM Corporation, San Jose, CA.
Dr. John Churchill	University of California, Davis, CA.
Dr. Glenn C. Bacon	IBM Corporation, San Jose, CA.

------------------------ ORIGINAL ATTACHMENT -----------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ----------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it

appeared.
Thanks for the cooperation.
Selma Zinker

```
Printed By: ZINKER_SELMA @TSII
   SENT: 87-04-30  11:51
   FROM: SINNAMON_LYN @EASY
     TO: ZINKER_SELMA @TSII
     CC. HANLON_ROBERT @EASY,DECRE_MARTY @EASY
SUBJECT: Tandem authors
In Reply to: 87-04-30  11:19  FROM ZINKER_SELMA : Tandem authors

"ADA in a Fault-Tolerant Networking Environment"
by Robert F. Hanlon, Lynwood D. Sinnamon, and Martin A. Decre
Tandem Computers, Federal Systems Marketing

Appearing in:U.S. Army Information Systems Engineering Command
                     Technology Strategies '87
                       February 9-12, 1987

------------------------ ORIGINAL  ATTACHMENT ------------------------
   SENT: 87-04-30  11:19
   FROM: ZINKER_SELMA
     TO: SINNAMON_LYN @EASY
SUBJECT: Tandem authors
In which proceedings was the paper published? Yes, it qualifies if it was
written while at Tandem.
Thanks for the response
Selma
```

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-30 10:56
FROM: STAAS_GARY
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Selma,

1. "New Display Station Offers Multiple Screen Windows and Dual Data Communication Ports", Hewlett-Packard Journal, March, 1981.
2. "TDL: A Hardware/Microcode Test Language Interpreter", The Seventeenth Annual Microprogramming Workshop, October 30-November 2, 1984.

Gary

------------------------ ORIGINAL ATTACHMENT ------------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.
Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-28 15:23
FROM: BORR_ANDREA
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

"Transaction Monitoring in Encompass: Reliable Distributed
Transaction Processing", Proceedings of VLDB, 1981.
"Robustness to Crash in a Distributed Database: A Non-Shared-Memory
Multi-Processor Approach", Proceeding of VLDB, 1984.

Andrea Borr

------------------------ ORIGINAL ATTACHMENT ------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors.
If you have had a paper published please respond to this message by
listing the title, journal name and date of the issue in which it
appeared.
Thanks for the cooperation.
Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-28 16:10
FROM: PETTIT_GHERY @COMM2
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Selma,

I don't know how far back you need information, but John Grebenkemper and I wrote a paper for the 1985 IEEE International EMC Symposium on the calibration of our semi-anechoic chamber. The full details are:

"Calibration of a Semi-Anechoic Chamber for FCC Open Field Site Radiated Emissions Measurements"

G. S. Pettit and C. J. Grebenkemper

1985 IEEE International Symposium on Electromagnetic Compatibility. Wakefield, Massachusetts, August 20-22, 1985.

If you already have this, sorry to make you look it up to find out.

Ghery S. Pettit

------------------------ ORIGINAL ATTACHMENT ------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.
Thanks for the cooperation.
Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-28 16:25
FROM: MADSEN_KENT @CASG
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Kent Madsen and David Foley, "When the shoe's on the vendor's foot: A look at the Tandem corporate network," Data Communications, August 1985.

Kent Madsen and David Foley, "How multiprocessor nodes can become more sociable," Data Communications, September 1985.

------------------------ ORIGINAL ATTACHMENT ------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.

Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 09:00
FROM: BECKER_PAT @MKT
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Selma:

I authored an article for Auerbach. "An Overview of Fault-Tolerant Systems", 1986 Auerbach Publishers Inc, Data Processing Management. It appeared around December of last year. I have a copy if that would help.

Pat

------------------------ ORIGINAL ATTACHMENT ------------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.

Selma Zinker

```
Printed By: ZINKER_SELMA @TSII
   SENT: 87-04-28  17:48
   FROM: NOLLEY_BARBARA @WORLD
     TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28  14:49  FROM ZINKER_SELMA : 2:Tandem authors
```

Selma,

I had a paper published in the April 1984 issue of Mental Retardation entitled "Microcomputer Data Analysis at the Clinical Mental Retardation Site. The paper was co-authored with Dr. David A. Nolley.

Barb

```
------------------------ ORIGINAL  ATTACHMENT -----------------------
   SENT: 87-04-28  14:49
   FROM: ZINKER_SELMA
     TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT  ATTACHMENT ----------------------------
   SENT: 87-04-28  14:47
   FROM: ZINKER_SELMA
```

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.

Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-28 19:00
FROM: CHOU_TIMOTHY
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

------------------------- ORIGINAL ATTACHMENT -------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------- TEXT ATTACHMENT -------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.
Selma Zinker

------------------------- TEXT ATTACHMENT -------------------------
SENT: 87-04-28 18:59
FROM: CHOU_TIMOTHY

T.C.K. Chou and J.A. Abraham, "Distributed Control of Computer Systems, " IEEE Transactions on Computers, June 1986.

R.W. Horst and T.C.K. Chou, "An Architecture for High Volume Transaction Processing," IEEE Proceedings of the 12th Annual International Symposium on Computer Architecture, Boston Massachusetts, June 1985.

T.C.K. Chou and J. Gray, "Transaction Acceleration," IEEE Quarterly Bulletin on Database Engineering, Vol 8. No. 1, March 1985.

T.C.K. Chou , P.N. Oleinick and A. Singh, "Language-Directed Modeling of On-Line Transaction Processing Systems," Proceedings of the Seventeenth Annual Hawaii International Conference on System Sciences, January 1984.

T.C.K. Chou and J.A. Abraham, "Load Redistribution in Distributed Systems, " IEEE Transactions on Computers, vol. C-32. No. 9, pp.799-808, September 1983.

T.C.K. Chou and J.A. Abraham, "Load Balancing in Distributed Systems, " IEEE Transactions on Software Engineering, vol SE-8, No. 4, pp. 401-412, July 1982.

T.C.K. Chou and J.A. Abraham, "Performance/Availability Model of Shared Resource Multiprocessors," IEEE Transactions on Reliability, vol R-29, No. 1, pp. 70-75, April 1980.

T.C.K. Chou, "Distributed System Protection" Tandem Systems Review, vol 2, No. 2, pp. 2-9, January 1986.

R.W. Horst and T.C.K. Chou, Contribution to Fault-Tolerant Systems edited by Eiichi Watanabe, McGraw-Hill / Japan

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 08:01
FROM: LOWENTHAL_BRUCE
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Hi Selma:

The paper I just had published is:

Title: Integration of Optical Disk into Mainframe System Software.
Author: Bruce Lowenthal
Journal: Digest of Papers; Spring Compcon 87 (Library of Congress #86-83313)

Bruce

------------------------ ORIGINAL ATTACHMENT ------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors.
If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.
Thanks for the cooperation.
Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 08:19
FROM: SIDOW_JIM @CASG
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Hi Selma;

I had the following titled article published in the January 1987 issue of Data Training:

"Getting Your System Operators Up and Running".

If you require an abstract or topic description, I can provide that.

Jim Sidow

------------------------- ORIGINAL ATTACHMENT -------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------- TEXT ATTACHMENT -------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.
Thanks for the cooperation.
Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 09:49
FROM: HORST_BOB @TSB
TO: ZINKER_SELMA @TSII
SUBJECT: List of papers

Selma -

Attached is a list of my papers which have been published.

-- Bob

------------------------ TEXT ATTACHMENT ----------------------------
SENT: 87-04-29 09:48
FROM: HORST_BOB @TSB
\poff 8
\new

PUBLICATIONS

Horst, R. and Metz, S., "New System Manages Hundreds of Transactions/Second." ELECTRONICS, pp. 147-151, April 19, 1984. Also Tandem Computers TR 84.1. Also reprinted in Tutorial: Fault Tolerant Computing, IEEE Computer Society Press, 1987.

Horst, R. and Chou, T., "An Architecture for High Volume Transaction Processing", Proceedings of 12th Annual International Symposium on Computer Architecture, Boston MA, June 17-19, 1985.

Horst, R. and Chou, T., "The Hardware Architecture and Linear Expansion of Tandem NonStop Systems", Tandem Computers TR 85.3. Also translated to Japanese and included in Fault Tolerant System, McGraw-Hill Book Company Japan, Ltd, 1986.

Bartlett, J., Gray, J. and Horst, R., "Fault Tolerance in Tandem Computer Systems", in The Evolution of Fault-Tolerant Computing, Springer-Verlag, 1986.

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 12:08
FROM: BATE_SIMON @SNAX
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

Selma,

Does my book "VAX/VMS Internals and Data Structures" count?

"VAX/VMS Internals and Data Structures", Lawrence Kenah and Simon Bate, 1984, Digital Press, Burlington, Mass.

S

---------------------- ORIGINAL ATTACHMENT ------------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

---------------------- TEXT ATTACHMENT ------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.
Selma Zinker

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 15:26
FROM: HAUSMAN_JOHN @PRUNE
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

"The Derived Type SIZE_T" by John M. Hausman in "The C Journal" dated Spring 1987

------------------------ ORIGINAL ATTACHMENT ------------------------
SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT ATTACHMENT ------------------------
SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.

Selma Zinker

These were written before employee joined Tandem

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 09:15
FROM: SHEEDY_CHRIS @PRUNE
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem author info
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA: 2:Tandem authors

1. "Measurement of Response Time Performance in Small Time-Sharing Systems", J.P. Penny and C.R. Sheedy, Australian Computer Journal, Vol 12, No 1, February 1980

2. "A Routing Scheme for Integrated Networks", Chris Sheedy, Computer Communications Review, Vol 12, No 1, January 1982

(Honestly, both the above are vol 12, no 1 !)

Ref 2. also appeared in the Proceedings of the Fifteenth Hawaii International Conference on System Sciences, 1982.

------------------------- ORIGINAL ATTACHMENT -------------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------- TEXT ATTACHMENT -------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.
Selma Zinker

```
Printed By: ZINKER_SELMA @TSII
   SENT: 87-04-29  13:09
   FROM: WESLING_PAUL @TSB
     TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors...
In Reply to: 87-04-28  14:49  FROM ZINKER_SELMA : 2:Tandem authors

Selma --
  I have a few, though none while here at Tandem.  Here are a couple:

"Packaging for the 580-Series Amdahl Computer", P. Wesling, IEEE
  Electronic Components Conference (5/81); republished in the IEEE
  Computer Society "VLSI Support Technologies" Tutorial.

"Peripheral Equipment as 'Components'", P. Wesling, IEEE/CHMT "NEWSLETTER",
  Sept. 82.

"Evaluation Checklist for Peripheral Equipment", P. Wesling, presented
   at PERIPHERALS '82 conference.

------------------------ ORIGINAL  ATTACHMENT ------------------------
   SENT: 87-04-28  14:49
   FROM: ZINKER_SELMA
     TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------ TEXT  ATTACHMENT ----------------------------
   SENT: 87-04-28  14:47
   FROM: ZINKER_SELMA
The Corporate Information Center is updating our files on Tandem authors.
If you have had a paper published please respond to this message by
listing the title, journal name and date of the issue in which it
appeared.
  Thanks for the cooperation.
   Selma Zinker
```

Printed By: ZINKER_SELMA @TSII
SENT: 87-04-29 07:35
FROM: GILBERT_GARY @CENTDIV
TO: ZINKER_SELMA @TSII
SUBJECT: Tandem authors
In Reply to: 87-04-28 14:49 FROM ZINKER_SELMA : 2:Tandem authors

David Stamper is a former Tandem employee (1978-86) who recently had a book published:

BUSINESS DATA COMMUNICATIONS
Benjamin/Cummings Publishing Co 1986
isbn 0-8053-9104-5

------------------------- ORIGINAL ATTACHMENT -------------------------

SENT: 87-04-28 14:49
FROM: ZINKER_SELMA
TO: DL.ALL_TANDEM @TSII
SUBJECT: 2:Tandem authors

------------------------- TEXT ATTACHMENT -------------------------

SENT: 87-04-28 14:47
FROM: ZINKER_SELMA

The Corporate Information Center is updating our files on Tandem authors. If you have had a paper published please respond to this message by listing the title, journal name and date of the issue in which it appeared.

Thanks for the cooperation.
Selma Zinker

The Nucleus of a Multiprogramming System

Per Brinch Hansen
A/S Regnecentralen, Copenhagen, Denmark

This paper describes the philosophy and structure of a multiprogramming system that can be extended with a hierarchy of operating systems to suit diverse requirements of program scheduling and resource allocation. The system nucleus simulates an environment in which program execution and input/output are handled uniformly as parallel, cooperating processes. A fundamental set of primitives allows the dynamic creation and control of a hierarchy of processes as well as the communication among them.

KEY WORDS AND PHRASES: multiprogramming, operating systems, parallel processes, process concept, process communication, message buffering, process hierarchy, process creation, process removal
CR CATEGORIES: 4.30, 4.31, 4.32, 4.41

1. Introduction

The multiprogramming system developed by Regnecentralen for the RC 4000 computer is a general tool for the design of operating systems. It allows the dynamic creation of a hierarchy of processes in which diverse strategies of program scheduling and resource allocation can be implemented.

For the designer of advanced information systems, a vital requirement of any operating system is that it allow him to change the mode of operation it controls; otherwise his freedom of design can be seriously limited. Unfortunately, this is precisely what present operating systems do not allow. Most of them are based exclusively on a single mode of operation, such as batch processing, priority scheduling, real-time scheduling, or conversational access.

When the need arises, the user often finds it hopeless to modify an operating system that has made rigid assumptions in its basic design about a specific mode of operation. The alternative—to replace the original operating system with a new one—is in most computers a serious, if not impossible, matter because the rest of the software is intimately bound to the conventions required by the original system.

This unfortunate situation indicates that the main problem in the design of a multiprogramming system is not to define functions that satisfy specific operating needs, but rather to supply a system nucleus that can be extended with new operating systems in an orderly manner. This is the primary objective of the RC 4000 system.

In the following, the philosophy and structure of the RC 4000 multiprogramming system is explained. The discussion does not include details of implementation; size and performance are presented, however, to give an idea of the feasibility of this approach. The functional specifications of the multiprogramming system are described in detail in a report [1] available from Regnecentralen.

2. System Nucleus

Our basic attitude during the designing was to make no assumptions about the particular strategy needed to optimize a given type of installation, but to concentrate on the fundamental aspects of the control of an environment consisting of parallel, cooperating processes.

Our first task was to assign a precise meaning to the process concept, i.e. to introduce an unambiguous terminology defining what a process is and how it is implemented on the actual computer.

The next step was to select primitives for the synchronization and transfer of information among parallel processes.

Our final decisions concerned the rules for the dynamic creation, control, and removal of processes.

The purpose of the system nucleus is to implement these fundamental concepts: simulation of processes; communication among processes; creation, control, and removal of processes.

3. Processes

We distinguish between internal and external processes, roughly corresponding to program execution and input/output.

More precisely, an *internal process* is the execution of one or more interruptable programs in a given storage area. An internal process is identified by a unique process name. Thus other processes need not be aware of the actual location of an internal process in the store, but can refer to it by name.

A sharp distinction is made between the concepts program and internal process. A *program* is a collection of instructions describing a computational process, whereas an internal process is the execution of these instructions in a given storage area.

In connection with input/output, the system distinguishes between peripheral devices, documents, and external processes.

A *peripheral device* is an item of hardware connected to the data channel and identified by a device number. A *document* is a collection of data stored on a physical medium, such as a deck of punched cards, a printer form, a reel of magnetic tape, or a file on the backing store.

An *external process* is the input/output of a given document identified by a unique process name. This concept implies that internal processes can refer to documents by name without knowing the actual devices on which they are mounted.

Multiprogramming and communication between internal and external processes are coordinated by the system nucleus—an interrupt response program with complete control of input/output, storage protection, and the interrupt system. We do not regard the system nucleus as an independent process, but rather as a software extension of the hardware structure, which makes the computer more attractive for multiprogramming. Its function is to implement our process concept and primitives that processes can invoke to create and control other processes and communicate with them.

So far we have described the multiprogramming system as a set of independent, parallel processes identified by names. The emphasis has been on a clear understanding of relationships among resources (store and peripherals), data (programs and documents), and processes (internal and external).

4. Process Communication

In a system of parallel, cooperating processes, mechanisms must be provided for the synchronization of two processes during a transfer of information.

Dijkstra has demonstrated that indivisible lock and unlock operations operating on binary semaphores are sufficient primitives from a logical point of view [3]. We have been forced to conclude, however, that the semaphore concept alone does not fulfill our requirements of safety and efficiency in a dynamic environment in which some processes may turn out to be black sheep and break the rules of the game.

Instead we have introduced message buffering within the system nucleus as the basic means of process communication. The system nucleus administers a common pool of *message buffers* and a *message queue* for each process.

The following primitives are available for the communication between internal processes:

send message (receiver, message, buffer),
wait message (sender, message, buffer),
send answer (result, answer, buffer),
wait answer (result, answer, buffer).

Send message copies a message into the first available buffer within the pool and delivers it in the queue of a named receiver. The receiver is activated if it is waiting for a message. The sender continues after being informed of the identity of the message buffer.

Wait message delays the requesting process until a message arrives in its queue. When the process is allowed to proceed, it is supplied with the name of the sender, the contents of the message, and the identity of the message buffer. The buffer is removed from the queue and made ready to transmit an answer.

Send answer copies an answer into a buffer in which a message has been received and delivers it in the queue of the original sender. The sender of the message is activated if it is waiting for the answer. The answering process continues immediately.

Wait answer delays the requesting process until an answer arrives in a given buffer. On arrival, the answer is copied into the process and the buffer is returned to the pool. The result specifies whether the answer is a response from another process or a dummy answer generated by the system nucleus in response to a message addressed to a nonexisting process.

The procedure wait message forces a process to serve its queue on a first-come, first-served basis. The system, however, also includes two primitives that enable a process to wait for the arrival of the next message or answer and serve its queue in any order.

This communication scheme has the following advantages.

The multiprogramming system is dynamic in the sense that processes can appear and disappear at any time. Therefore a process does not in general have a complete knowledge of the existence of other processes. This is reflected in the procedure wait message, which makes it possible for a process to be unaware of the existence of other processes until it receives messages from them.

On the other hand, once a communication has been established between two processes (i.e. by means of a message) they need a common identification of it in order to agree on when it is terminated (i.e. by means of an answer). Thus we can properly regard the selection of a buffer as the creation of an identification of a conversation. A happy consequence of this is that it enables two processes to exchange more than one message at a time.

We must be prepared for the occurrence of erroneous or malicious processes in the system (e.g. undebugged programs). This is tolerable only if the system nucleus ensures that no process can interfere with a conversation between two other processes. This is done by storing the identity of the sender and receiver in each buffer and checking it whenever a process attempts to send or wait for an answer in a given buffer.

Efficiency is obtained by the queueing of buffers, which enables a sending process to continue immediately after delivery of a message or an answer, regardless of whether or not the receiver is ready to process it.

To make the system dynamic, it is vital that a process can be removed at any time, even if it is engaged in one or more conversations. In this case, the system nucleus leaves all messages from the removed process undisturbed in the queues of other processes. When these processes answer them, the system nucleus returns the buffers to the common pool.

The reverse situation is also possible: during the removal of a process, the system nucleus finds unanswered messages sent to the process. These are returned as dummy answers to the senders.

The main drawback of message buffering is that it introduces yet another resource problem, since the common pool contains a finite number of buffers. If a process were allowed to empty the pool by sending messages to ignorant processes, which do not respond with answers, further communication within the system would be blocked. Consequently a limit is set to the number of messages a process can send simultaneously. By doing this, and by allowing a process to transmit an answer in a received buffer, we have placed the entire risk of a conversation on the process that opens it.

5. External Processes

Originally the communication primitives were designed for the exchange of messages between internal processes. Later we also decided to use *send message* and *wait answer* for communication between internal and external processes.

For each kind of external process, the system nucleus contains a piece of code that interprets a message from an internal process and initiates input/output using a storage area specified in the message. When input/output is terminated by an interrupt, the nucleus generates an answer to the internal process with information about actual block size and possible error conditions. This is essentially the implementation of the external process concept.

We consider it to be an important aspect of the system that internal and external processes are handled uniformly as independent, self-contained processes. The difference between them is merely a matter of processing capability. A consequence of this is that any external process can be replaced by an internal process of the same name if more complex criteria of access and response become desirable.

External processes are created on request from internal processes. *Creation* is simply the assignment of a name to a particular peripheral device. To guarantee internal processes exclusive access to sequential documents, primitives are available for the *reservation* and *release* of external processes.

Typewriter consoles are the only external processes that can send messages to internal processes. The operator opens a conversation by pushing an interrupt key and typing the name of the internal receiver followed by a line of text.

A file on the backing store can be used as an external process by copying a description of the file from a catalog on the backing store into the system nucleus; following this, internal processes can initiate input/output by sending messages to the file process.

Real-time synchronization of internal processes is obtained by sending messages to a clock process. After the elapse of a time interval specified in the message, the clock returns an answer to the sending process.

In general, external processes can be used to obtain synchronization between internal processes and any signal from the external world. For example, an internal process may send a message to a watchdog process and receive an answer when a magnetic tape is mounted on a station. In response, the internal process can give the station a temporary name, identify the tape by reading its label, and rename the station accordingly.

6. Internal Processes

A final set of primitives in the system nucleus allows the creation, control, and removal of internal processes.

Internal processes are created on request from other internal processes. *Creation* involves the assignment of a name to a contiguous storage area selected by the parent process. The storage area must be within the parent's own area.

After creation, the parent process can load a program into the child process and *start* it. The child process now shares computing time with other active processes including the parent process.

On request from a parent process, the system nucleus waits for the completion of all input/output initiated by a child process and *stops* it. In the stopped state, the process can still receive messages and answers in its queue. These can be served when the process is restarted.

Finally, a parent process can *remove* a child process in order to assign its storage area to other processes.

According to our philosophy, processes should have complete freedom to choose their own strategy of program scheduling. The system nucleus only supplies the essential primitives for initiation and control of processes. Consequently, the concepts of program loading and swapping are not part of the nucleus. Time-sharing of a common storage area among child processes on a swapping basis is possible, however, because the system does not check whether internal processes overlap each other as long as they remain within the storage areas of their parents. Swapping from process A to process B can be implemented in a parent process as follows: stop(A); output(A); input(B); start(B).

7. Process Hierarchy

The idea of the system nucleus has been described as the simulation of an environment in which program execution and input/output are handled uniformly as parallel, cooperating processes. A fundamental set of primitives allows the dynamic creation and control of processes as well as communication among them.

For a given installation we still need, as part of the system, programs that control strategies of operator communication, program scheduling, and resource allocation; but it is essential for the orderly growth of the system that these *operating systems* be implemented as other programs. Since the difference between operating systems and production programs is one of jurisdiction only, this problem is solved by arranging the internal processes in a *hierarchy* in which parent processes have complete control over child processes.

After initial loading, the internal store contains the system nucleus and a basic operating system, S, which can create parallel processes, A, B, C, etc., on request from consoles. The processes can in turn create other processes, D, E, F, etc. Thus while S acts as a primitive operating system for A, B, and C, these in turn act as operating systems for their children, D, E, and F. This is illustrated by Figure 1, which shows a family tree of processes on the left and the corresponding storage allocation on the right. This family tree of processes can be extended to any level, subject only to a limitation of the total number of processes.

In this multiprogramming system, all privileged functions are implemented in the system nucleus, which has no built-in strategy. Strategies can be introduced at the various higher levels, where each process has the power to control the scheduling and resource allocation of its children. The only rules enforced by the nucleus are the following: a process can only allocate a subset of its own resources (including storage and message buffers) to its children; a process can only start, stop, and remove its own children (including their descendants). After removal of a process, its resources are returned to the parent process.

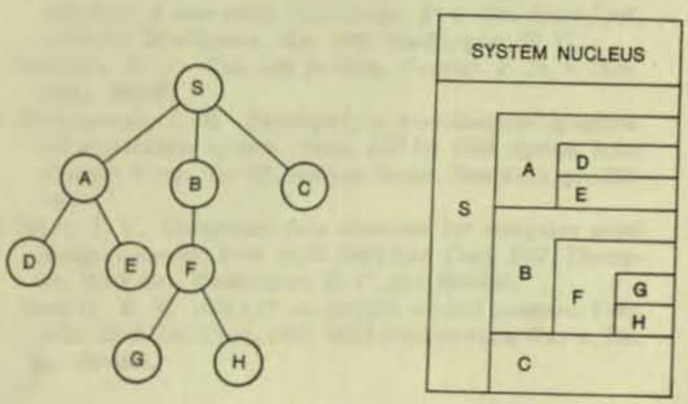


FIG. 1

Initially all system resources are owned by the basic operating system S. For details of process control and resource allocation, the reader should consult the manual of the system [1].

We emphasize that the only function of the family tree is to define the rules of process control and resource allocation. Computing time is shared by round-robin scheduling among active processes regardless of their position in the hierarchy, and each process can communicate with all other processes.

Regarding the future development of operating systems, the most important characteristics of the system can now be seen as the following.

1. New operating systems can be implemented as other programs without modification of the system nucleus. In this connection, we should mention that the ALGOL and FORTRAN languages for the RC 4000 contain facilities for calling the nucleus and initiating parallel processes. Thus it is possible to write operating systems in high-level languages.

2. Operating systems can be replaced dynamically, thus enabling an installation to switch among various modes of operation; several operating systems can, in fact, be active simultaneously.

3. Standard programs and user programs can be executed under different operating systems without modification, provided there is common agreement on the possible communication between parents and children.

8. Implementation

The RC 4000 is a 24-bit, binary computer with typical instruction execution times of 4 microseconds [2]. It permits practically unlimited expansion of the internal store and standardized connection of all kinds of peripherals. Multiprogramming is facilitated by program interruption, storage protection, and privileged instructions.

The present implementation of the system makes multiprogramming feasible with a minimum store of 16K–32K words backed by a fast drum or disk. The system nucleus includes external processes for a real-time clock, typewriters, paper tape input/output, line printer, magnetic tape, and files on the backing store. The size of the nucleus and the basic operating system is as follows:

	words
primitives	2400
code for external processes	1150
process descriptions and buffers	1250
system nucleus	4800
basic operating system	1400
	6200

The communication primitives are executed in the uninterruptable mode within the system nucleus. The execution times of these set a limit to the system's response to real-time events:

	msec
send message	0.6
wait answer	0.4
wait message	0.4
send answer	0.6

An analysis shows that the 2 milliseconds required by a complete conversation (the sum of the four primitives) are used as follows:

	percent
validity checking	25
process activation	45
message buffering	30

This distribution is so even that one cannot hope to increase the speed of the system by introducing additional, ad hoc machine instructions. The only realistic solution is to make the hardware faster.

The primitives for creation, start, stop, and removal of processes are implemented in an anonymous internal process within the system nucleus to avoid intolerably long periods in the uninterruptable mode. Typical execution times for these are:

	msec
create process	3
start process	26
stop process	4
remove process	30

(Continued on page 250)

The analysis presented here suggests that spatial domains are the primitive element of this particular graphic language. In this light, the common assumption that line segments are the primitives of many graphic languages may require revision.

Received June, 1969; revised October, 1969

REFERENCES

1. Gross, Maurice, and Nivat, Maurice. A command language for visualization and articulated movements. In *Computer and Information Sciences II*, Julius T. Tou (Ed), Academic Press, New York, 1967.
2. Nilsson, Nils J. A mobile automaton: An application of artificial intelligence techniques. Proc. Int. Joint Conf. Artificial Intelligence, May 1969, Washington, D. C.
3. Eastman, Charles M. Explorations of the cognitive processes of design, Dep. of Comput. Sci., Carnegie-Mellon U., Feb. 1968, ARPA Rep. DDC No. AD671158, Clearinghouse, Springfield, VA 22151.
4. Eastman, Charles M. Cognitive processes and ill-defined problems: A case study from design, Proc. Int. Joint Conf. Artificial Intelligence, May 1969, Washington, D. C.
5. Howden, W. E. The sofa problem. *Comput. J. 11*, 3 (Nov. 1968), 299–301.
6. Sutherland, I. E. Sketchpad: a man-machine graphical communication system. Proc. AFIPS 1963 Spring Joint Comput. Conf., Vol. 23, Spartan Books, New York, pp. 329–346.
7. Gray, J. C. Compound data structure for computer aided design: a survey, Proc. ACM 22nd Nat. Conf. 1967, Thompson Book Co., Washington, D. C., pp. 355–365.
8. Thomas, E. M. GRASP—a graphic service program. Proc. ACM 22nd Nat. Conf., 1967, MDI Publications, Wayne, Pa., pp. 395–402.
9. Armour, Gordon C., and Buffa, Elwood. A heuristic algorithm and simulation approach to relative location of facilities. *Man. Sci.* (Jan. 1963), 244–309.
10. Lee, R. B. and Moore, J. M. CORELAP—computerized relationship layout planning, *J. Indust. Eng., 18*, 3 (Mar. 1967) 195–200.
11. Simpson, M. G., et al. The planning of multi-story buildings: a systems analysis and simulation approach. Proc. European Meeting on Statistics, Econometrics and Management Science, Amsterdam, Sept. 1968.
12. Barken, Robert. A set of algorithms for automatically laying out hybrid integrated circuits. Internal working doc., Bell Telephone Lab., Holmdel, N. J., Aug. 1968.
13. Nilsson, N. J., and Raphael, B. Preliminary design of an intelligent robot. In *Computer and Information Sciences II*, Julius T. Tou (Ed.), Academic Press, New York, 1967.
14. Rosen, C. A., and Nilsson, N. J. Application of intelligent automata to reconnaissance. SRI Project 5953, Third Interim Report, Rome Air Develop. Center, Rome, N. Y., Dec. 1967.
15. Fair, G. R., Flowerdew, et al. Note on the computer as an aid to the architect. *Comput. J. 9*, 1 (June 1966).
16. Griswold, R., Poage, J., and Polonsky, I. The SNOBOL4 programming language. Bell Telephone Lab., Holmdel, N. J., Aug., 1968.
17. McCarthy, John, et al. *LISP1.5 Programmer's Manual*. MIT Press, Cambridge, Mass., 1965.
18. Moran, Thomas. Structuring three-dimensional space for computer manipulation. Dep. Comput. Sci. working paper, Carnegie-Mellon U., Pittsburgh, Pa., June, 1968.
19. Moran, Thomas. A model of a multi-lingual designer. In *Emerging Methods in Environmental Design and Planning*, G. Moore (Ed.), MIT Press, Cambridge, Mass. (in press).
20. Wylie, C. Romney, et al. Halftone perspective drawings by computer. Tech. Rep. 4-2, Comput. Sci. Dep., U. of Utah, Salt Lake City, Utah, Feb. 1968.

Hansen—cont'd from page 241

The excessive times for the start and removal of an internal process are due to the peculiar storage protection system of the RC 4000, which requires the setting of a protection key in every storage word of a process.

9. Conclusion

Ideas similar to those described here have been suggested by others [4–6]. We have presented our system because we feel that, taken as a whole, it represents a systematic and practical approach to the design of replaceable operating systems. As an inspiration to other designers, it is perhaps most important that it illustrates a sequence of design steps leading to a general system nucleus, namely, the definition of the process concept, the communication scheme, and the dynamic creation and structuring of processes.

We realize, of course, that a final evaluation of the system can only be made after it has been used to design a number of operating systems.

Acknowledgments. The design philosophy was developed by Jørn Jensen, Søren Lauesen, and the author. Leif Svalgaard participated in the implementation and testing of the final product.

Regarding fundamentals, we have benefited greatly from Dijkstra's analysis of cooperating sequential processes.

Received July, 1969; Revised January, 1970

REFERENCES

1. *RC 4000 Software: Multiprogramming System*. P. Brinch Hansen (Ed.). A/S Regnecentralen, Copenhagen, 1969.
2. *RC 4000 Computer: Reference Manual*. P. Brinch Hansen (Ed.), A/S Regnecentralen, Copenhagen, 1969.
3. Dijkstra, E. W. Cooperating Sequential Processes. Math. Dep., Technological U., Eindhoven, Sept. 1965.
4. Harrison, M. C., and Schwartz, J. T. SHARER, a time sharing system for the CDC 6600. *Comm. ACM 10*, (Oct. 1967) 659.
5. Huxtable, D. H. R., and Warwick, M. T. Dynamic supervisors—their design and construction. Proc. ACM Symp. on Operating System Principles, Gatlinburg, Tenn., Oct. 1–4 1967.
6. Wichmann, B. A. A modular operating system. Proc. IFIP Cong. 1968, North Holland Pub. Co., Amsterdam, p. C48.

The Nucleus of a Multiprogramming System

Per Brinch Hansen
A/S Regnecentralen, Copenhagen, Denmark

This paper describes the philosophy and structure of a multiprogramming system that can be extended with a hierarchy of operating systems to suit diverse requirements of program scheduling and resource allocation. The system nucleus simulates an environment in which program execution and input/output are handled uniformly as parallel, cooperating processes. A fundamental set of primitives allows the dynamic creation and control of a hierarchy of processes as well as the communication among them.

KEY WORDS AND PHRASES: multiprogramming, operating systems, parallel processes, process concept, process communication, message buffering, process hierarchy, process creation, process removal
CR CATEGORIES: 4.30, 4.31, 4.32, 4.41

1. Introduction

The multiprogramming system developed by Regnecentralen for the RC 4000 computer is a general tool for the design of operating systems. It allows the dynamic creation of a hierarchy of processes in which diverse strategies of program scheduling and resource allocation can be implemented.

For the designer of advanced information systems, a vital requirement of any operating system is that it allow him to change the mode of operation it controls; otherwise his freedom of design can be seriously limited. Unfortunately, this is precisely what present operating systems do not allow. Most of them are based exclusively on a single mode of operation, such as batch processing, priority scheduling, real-time scheduling, or conversational access.

When the need arises, the user often finds it hopeless to modify an operating system that has made rigid assumptions in its basic design about a specific mode of operation. The alternative—to replace the original operating system with a new one—is in most computers a serious, if not impossible, matter because the rest of the software is intimately bound to the conventions required by the original system.

This unfortunate situation indicates that the main problem in the design of a multiprogramming system is not to define functions that satisfy specific operating needs, but rather to supply a system nucleus that can be extended with new operating systems in an orderly manner. This is the primary objective of the RC 4000 system.

In the following, the philosophy and structure of the RC 4000 multiprogramming system is explained. The discussion does not include details of implementation; size and performance are presented, however, to give an idea of the feasibility of this approach. The functional specifications of the multiprogramming system are described in detail in a report [1] available from Regnecentralen.

2. System Nucleus

Our basic attitude during the designing was to make no assumptions about the particular strategy needed to optimize a given type of installation, but to concentrate on the fundamental aspects of the control of an environment consisting of parallel, cooperating processes.

Our first task was to assign a precise meaning to the process concept, i.e. to introduce an unambiguous terminology defining what a process is and how it is implemented on the actual computer.

The next step was to select primitives for the synchronization and transfer of information among parallel processes.

Our final decisions concerned the rules for the dynamic creation, control, and removal of processes.

The purpose of the system nucleus is to implement these fundamental concepts: simulation of processes; communication among processes; creation, control, and removal of processes.

3. Processes

We distinguish between internal and external processes, roughly corresponding to program execution and input/output.

More precisely, an *internal process* is the execution of one or more interruptable programs in a given storage area. An internal process is identified by a unique process name. Thus other processes need not be aware of the actual location of an internal process in the store, but can refer to it by name.

A sharp distinction is made between the concepts program and internal process. A *program* is a collection of instructions describing a computational process, whereas an internal process is the execution of these instructions in a given storage area.

In connection with input/output, the system distinguishes between peripheral devices, documents, and external processes.

A *peripheral device* is an item of hardware connected to the data channel and identified by a device number. A *document* is a collection of data stored on a physical medium, such as a deck of punched cards, a printer form, a reel of magnetic tape, or a file on the backing store.

An *external process* is the input/output of a given document identified by a unique process name. This concept implies that internal processes can refer to documents by name without knowing the actual devices on which they are mounted.

Multiprogramming and communication between internal and external processes are coordinated by the system nucleus—an interrupt response program with complete control of input/output, storage protection, and the interrupt system. We do not regard the system nucleus as an independent process, but rather as a software extension of the hardware structure, which makes the computer more attractive for multiprogramming. Its function is to implement our process concept and primitives that processes can invoke to create and control other processes and communicate with them.

So far we have described the multiprogramming system as a set of independent, parallel processes identified by names. The emphasis has been on a clear understanding of relationships among resources (store and peripherals), data (programs and documents), and processes (internal and external).

4. Process Communication

In a system of parallel, cooperating processes, mechanisms must be provided for the synchronization of two processes during a transfer of information.

Dijkstra has demonstrated that indivisible lock and unlock operations operating on binary semaphores are sufficient primitives from a logical point of view [3]. We have been forced to conclude, however, that the semaphore concept alone does not fulfill our requirements of safety and efficiency in a dynamic environment in which some processes may turn out to be black sheep and break the rules of the game.

Instead we have introduced message buffering within the system nucleus as the basic means of process communication. The system nucleus administers a common pool of *message buffers* and a *message queue* for each process.

The following primitives are available for the communition between internal processes:

send message (receiver, message, buffer),
wait message (sender, message, buffer),
send answer (result, answer, buffer),
wait answer (result, answer, buffer).

Send message copies a message into the first available buffer within the pool and delivers it in the queue of a named receiver. The receiver is activated if it is waiting for a message. The sender continues after being informed of the identity of the message buffer.

Wait message delays the requesting process until a message arrives in its queue. When the process is allowed to proceed, it is supplied with the name of the sender, the contents of the message, and the identity of the message buffer. The buffer is removed from the queue and made ready to transmit an answer.

Send answer copies an answer into a buffer in which a message has been received and delivers it in the queue of the original sender. The sender of the message is activated if it is waiting for the answer. The answering process continues immediately.

Wait answer delays the requesting process until an answer arrives in a given buffer. On arrival, the answer is copied into the process and the buffer is returned to the pool. The result specifies whether the answer is a response from another process or a dummy answer generated by the system nucleus in response to a message addressed to a nonexisting process.

The procedure wait message forces a process to serve its queue on a first-come, first-served basis. The system, however, also includes two primitives that enable a process to wait for the arrival of the next message or answer and serve its queue in any order.

This communication scheme has the following advantages.

The multiprogramming system is dynamic in the sense that processes can appear and disappear at any time. Therefore a process does not in general have a complete knowledge of the existence of other processes. This is reflected in the procedure wait message, which makes it possible for a process to be unaware of the existence of other processes until it receives messages from them.

On the other hand, once a communication has been established between two processes (i.e. by means of a message) they need a common identification of it in order to agree on when it is terminated (i.e. by means of an answer). Thus we can properly regard the selection of a buffer as the creation of an identification of a conversation. A happy consequence of this is that it enables two processes to exchange more than one message at a time.

We must be prepared for the occurrence of erroneous or malicious processes in the system (e.g. undebugged programs). This is tolerable only if the system nucleus ensures that no process can interfere with a conversation between two other processes. This is done by storing the identity of the sender and receiver in each buffer and checking it whenever a process attempts to send or wait for an answer in a given buffer.

Efficiency is obtained by the queueing of buffers, which enables a sending process to continue immediately after delivery of a message or an answer, regardless of whether or not the receiver is ready to process it.

To make the system dynamic, it is vital that a process can be removed at any time, even if it is engaged in one or more conversations. In this case, the system nucleus leaves all messages from the removed process undisturbed in the queues of other processes. When these processes answer them, the system nucleus returns the buffers to the common pool.

The reverse situation is also possible: during the removal of a process, the system nucleus finds unanswered messages sent to the process. These are returned as dummy answers to the senders.

The main drawback of message buffering is that it introduces yet another resource problem, since the common pool contains a finite number of buffers. If a process were allowed to empty the pool by sending messages to ignorant processes, which do not respond with answers, further communication within the system would be blocked. Consequently a limit is set to the number of messages a process can send simultaneously. By doing this, and by allowing a process to transmit an answer in a received buffer, we have placed the entire risk of a conversation on the process that opens it.

5. External Processes

Originally the communication primitives were designed for the exchange of messages between internal processes. Later we also decided to use *send message* and *wait answer* for communication between internal and external processes.

For each kind of external process, the system nucleus contains a piece of code that interprets a message from an internal process and initiates input/output using a storage area specified in the message. When input/output is terminated by an interrupt, the nucleus generates an answer to the internal process with information about actual block size and possible error conditions. This is essentially the implementation of the external process concept.

We consider it to be an important aspect of the system that internal and external processes are handled uniformly as independent, self-contained processes. The difference between them is merely a matter of processing capability. A consequence of this is that any external process can be replaced by an internal process of the same name if more complex criteria of access and response become desirable.

External processes are created on request from internal processes. *Creation* is simply the assignment of a name to a particular peripheral device. To guarantee internal processes exclusive access to sequential documents, primitives are available for the *reservation* and *release* of external processes.

Typewriter consoles are the only external processes that can send messages to internal processes. The operator opens a conversation by pushing an interrupt key and typing the name of the internal receiver followed by a line of text.

A file on the backing store can be used as an external process by copying a description of the file from a catalog on the backing store into the system nucleus; following this, internal processes can initiate input/output by sending messages to the file process.

Real-time synchronization of internal processes is obtained by sending messages to a clock process. After the elapse of a time interval specified in the message, the clock returns an answer to the sending process.

In general, external processes can be used to obtain synchronization between internal processes and any signal from the external world. For example, an internal process may send a message to a watchdog process and receive an answer when a magnetic tape is mounted on a station. In response, the internal process can give the station a temporary name, identify the tape by reading its label, and rename the station accordingly.

6. Internal Processes

A final set of primitives in the system nucleus allows the creation, control, and removal of internal processes.

Internal processes are created on request from other internal processes. *Creation* involves the assignment of a name to a contiguous storage area selected by the parent process. The storage area must be within the parent's own area.

After creation, the parent process can load a program into the child process and *start* it. The child process now shares computing time with other active processes including the parent process.

On request from a parent process, the system nucleus waits for the completion of all input/output initiated by a child process and *stops* it. In the stopped state, the process can still receive messages and answers in its queue. These can be served when the process is restarted.

Finally, a parent process can *remove* a child process in order to assign its storage area to other processes.

According to our philosophy, processes should have complete freedom to choose their own strategy of program scheduling. The system nucleus only supplies the essential primitives for initiation and control of processes. Consequently, the concepts of program loading and swapping are not part of the nucleus. Time-sharing of a common storage area among child processes on a swapping basis is possible, however, because the system does not check whether internal processes overlap each other as long as they remain within the storage areas of their parents. Swapping from process A to process B can be implemented in a parent process as follows: stop(A); output(A); input(B); start(B).

7. Process Hierarchy

The idea of the system nucleus has been described as the simulation of an environment in which program execution and input/output are handled uniformly as parallel, cooperating processes. A fundamental set of primitives allows the dynamic creation and control of processes as well as communication among them.

For a given installation we still need, as part of the system, programs that control strategies of operator communication, program scheduling, and resource allocation; but it is essential for the orderly growth of the system that these *operating systems* be implemented as other programs. Since the difference between operating systems and production programs is one of jurisdiction only, this problem is solved by arranging the internal processes in a *hierarchy* in which parent processes have complete control over child processes.

After initial loading, the internal store contains the system nucleus and a basic operating system, S, which can create parallel processes, A, B, C, etc., on request from consoles. The processes can in turn create other processes, D, E, F, etc. Thus while S acts as a primitive operating system for A, B, and C, these in turn act as operating systems for their children, D, E, and F. This is illustrated by Figure 1, which shows a family tree of processes on the left and the corresponding storage allocation on the right. This family tree of processes can be extended to any level, subject only to a limitation of the total number of processes.

In this multiprogramming system, all privileged functions are implemented in the system nucleus, which has no built-in strategy. Strategies can be introduced at the various higher levels, where each process has the power to control the scheduling and resource allocation of its children. The only rules enforced by the nucleus are the following: a process can only allocate a subset of its own resources (including storage and message buffers) to its children; a process can only start, stop, and remove its own children (including their descendants). After removal of a process, its resources are returned to the parent process.

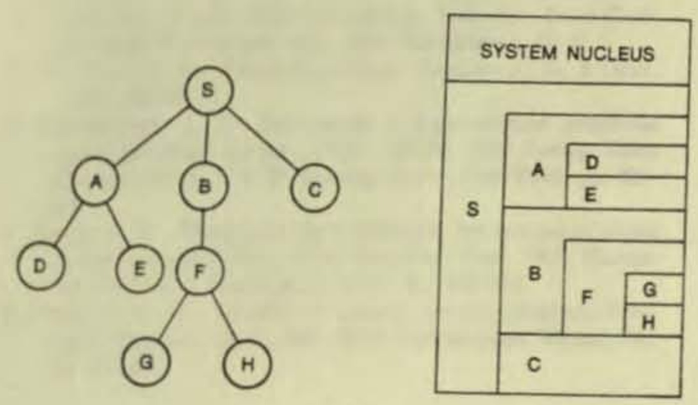


FIG. 1

Initially all system resources are owned by the basic operating system S. For details of process control and resource allocation, the reader should consult the manual of the system [1].

We emphasize that the only function of the family tree is to define the rules of process control and resource allocation. Computing time is shared by round-robin scheduling among active processes regardless of their position in the hierarchy, and each process can communicate with all other processes.

Regarding the future development of operating systems, the most important characteristics of the system can now be seen as the following.

1. New operating systems can be implemented as other programs without modification of the system nucleus. In this connection, we should mention that the ALGOL and FORTRAN languages for the RC 4000 contain facilities for calling the nucleus and initiating parallel processes. Thus it is possible to write operating systems in high-level languages.

2. Operating systems can be replaced dynamically, thus enabling an installation to switch among various modes of operation; several operating systems can, in fact, be active simultaneously.

3. Standard programs and user programs can be executed under different operating systems without modification, provided there is common agreement on the possible communication between parents and children.

8. Implementation

The RC 4000 is a 24-bit, binary computer with typical instruction execution times of 4 microseconds [2]. It permits practically unlimited expansion of the internal store and standardized connection of all kinds of peripherals. Multiprogramming is facilitated by program interruption, storage protection, and privileged instructions.

The present implementation of the system makes multiprogramming feasible with a minimum store of 16K–32K words backed by a fast drum or disk. The system nucleus includes external processes for a real-time clock, typewriters, paper tape input/output, line printer, magnetic tape, and files on the backing store. The size of the nucleus and the basic operating system is as follows:

	words
primitives	2400
code for external processes	1150
process descriptions and buffers	1250
system nucleus	4800
basic operating system	1400
	6200

The communication primitives are executed in the uninterruptable mode within the system nucleus. The execution times of these set a limit to the system's response to real-time events:

	msec
send message	0.6
wait answer	0.4
wait message	0.4
send answer	0.6

An analysis shows that the 2 milliseconds required by a complete conversation (the sum of the four primitives) are used as follows:

	percent
validity checking	25
process activation	45
message buffering	30

This distribution is so even that one cannot hope to increase the speed of the system by introducing additional, ad hoc machine instructions. The only realistic solution is to make the hardware faster.

The primitives for creation, start, stop, and removal of processes are implemented in an anonymous internal process within the system nucleus to avoid intolerably long periods in the uninterruptable mode. Typical execution times for these are:

	msec
create process	3
start process	26
stop process	4
remove process	30

(Continued on page 250)

The analysis presented here suggests that spatial domains are the primitive element of this particular graphic language. In this light, the common assumption that line segments are the primitives of many graphic languages may require revision.

RECEIVED JUNE, 1969; REVISED OCTOBER, 1969

REFERENCES

1. GROSS, MAURICE, AND NIVAT, MAURICE. A command language for visualization and articulated movements. In *Computer and Information Sciences II*, Julius T. Tou (Ed), Academic Press, New York, 1967.
2. NILSSON, NILS J. A mobile automaton: An application of artificial intelligence techniques. Proc. Int. Joint Conf. Artificial Intelligence, May 1969, Washington, D. C.
3. EASTMAN, CHARLES M. Explorations of the cognitive processes of design, Dep. of Comput. Sci., Carnegie-Mellon U., Feb. 1968, ARPA Rep. DDC No. AD671158, Clearinghouse, Springfield, VA 22151.
4. EASTMAN, CHARLES M. Cognitive processes and ill-defined problems: A case study from design, Proc. Int. Joint Conf. Artificial Intelligence, May 1969, Washington, D. C.
5. HOWDEN, W. E. The sofa problem. *Comput. J. 11*, 3 (Nov. 1968), 299–301.
6. SUTHERLAND, I. E. Sketchpad: a man-machine graphical communication system. Proc. AFIPS 1963 Spring Joint Comput. Conf., Vol. 23, Spartan Books, New York, pp. 329–346.
7. GRAY, J. C. Compound data structure for computer aided design: a survey, Proc. ACM 22nd Nat. Conf. 1967, Thompson Book Co., Washington, D. C., pp. 355–365.
8. THOMAS, E. M. GRASP—a graphic service program. Proc. ACM 22nd Nat. Conf., 1967, MDI Publications, Wayne, Pa., pp. 395–402.
9. ARMOUR, GORDON C., AND BUFFA, ELWOOD. A heuris... algorithm and simulation approach to relative locati... facilities. *Man. Sci.* (Jan. 1963), 244–309.
10. LEE, R. B. AND MOORE, J. M. CORELAP—computeriz... relationship layout planning, *J. Indust. Eng., 18*, 3 (Mar. 1967) 195–200.
11. SIMPSON, M. G., ET AL. The planning of multi-story buildings, a systems analysis and simulation approach. Proc. Europ... Meeting on Statistics, Econometrics and Manage... Science, Amsterdam, Sept. 1968.
12. BARKEN, ROBERT. A set of algorithms for automatic... laying out hybrid integrated circuits. Internal working p... Bell Telephone Lab., Holmdel, N. J., Aug. 1968.
13. NILSSON, N. J., AND RAPHAEL, B. Preliminary design of... intelligent robot. In *Computer and Information Sciences II*, Julius T. Tou (Ed.), Academic Press, New York, 1967.
14. ROSEN, C. A., AND NILSSON, N. J. Application of intellig... automata to reconnaisance. SRI Project 5953, Third Inter... Report, Rome Air Develop. Center, Rome, N. Y., De... 1...
15. FAIR, G. R., FLOWERDEW, ET AL. Note on the comp... an aid to the architect. *Comput. J. 9*, 1 (June 1966).
16. GRISWOLD, R., POAGE, J., AND POLONSKY, I. The SNOBOL... programming language. Bell Telephone Lab., Holm... N. J., Aug., 1968.
17. MCCARTHY, JOHN, ET AL. *LISP1.5 Programmer's Man...* MIT Press, Cambridge, Mass., 1965.
18. MORAN, THOMAS. Structuring three-dimensional space... computer manipulation. Dep. Comput. Sci. work... paper, Carnegie-Mellon U., Pittsburgh, Pa., June, 1968.
19. MORAN, THOMAS. A model of a multi-lingual designer. In *Emerging Methods in Environmental Design and P...*, G. Moore (Ed.), MIT Press, Cambridge, Mass. (in press).
20. WYLIE, C. ROMNEY, ET AL. Halftone perspective drawings... computer. Tech. Rep. 4-2, Comput. Sci. Dep., U. of Utah, Salt Lake City, Utah, Feb. 1968.

Hansen—cont'd from page 241

The excessive times for the start and removal of an internal process are due to the peculiar storage protection system of the RC 4000, which requires the setting of a protection key in every storage word of a process.

9. Conclusion

Ideas similar to those described here have been suggested by others [4–6]. We have presented our system because we feel that, taken as a whole, it represents a systematic and practical approach to the design of replaceable operating systems. As an inspiration to other designers, it is perhaps most important that it illustrates a sequence of design steps leading to a general system nucleus, namely, the definition of the process concept, the communication scheme, and the dynamic creation and structuring of processes.

We realize, of course, that a final evaluation of the system can only be made after it has been used to design a number of operating systems.

Acknowledgments. The design philosophy was developed by Jørn Jensen, Søren Lauesen, and the author. Leif Svalgaard participated in the implementation and testing of the final product.

Regarding fundamentals, we have benefited greatly from Dijkstra's analysis of cooperating sequential processes.

RECEIVED JULY, 1969; REVISED JANUARY, 1970

REFERENCES

1. *RC 4000 Software: Multiprogramming System.* P. Brinch Hansen (Ed.). A/S Regnecentralen, Copenhagen, 1969.
2. *RC 4000 Computer: Reference Manual.* P. Brinch Hansen (Ed.) A/S Regnecentralen, Copenhagen, 1969.
3. DIJKSTRA, E. W. Cooperating Sequential Processes. Math. Dep., Technological U., Eindhoven, Sept. 1965.
4. HARRISON, M. C., AND SCHWARTZ, J. T. SHARER, a time sharing system for the CDC 6600. *Comm. ACM 10*, (Oct. 1967) 659.
5. HUXTABLE, D. H. R., AND WARWICK, M. T. Dynamic supervisors—their design and construction. Proc. ACM Symp. Operating System Principles, Gatlinburg, Tenn., Oct. 1–4, 1967.
6. WICHMANN, B. A. A modular operating system. Proc. IFIP Cong. 1968, North Holland Pub. Co., Amsterdam, p. C48.

THE CIRRUS BANKING NETWORK

The CIRRUS banking network makes coast-to-coast automatic banking transactions possible. The system will soon be able to handle international currency transactions and point-of-sale transactions in stores.

DAVID GIFFORD and ALFRED SPECTOR

Interview 1. Bruce Burchfield, president of CIRRUS Systems, Inc., talks about the development of the CIRRUS network, its present scope, and plans for its future.

DG. **Bruce, could you start by telling us something about the history of the CIRRUS network?**
Burchfield. I'd be glad to. Bankers have traditionally provided customers with personalized service. That's the nature of the business. By the 1970s, most banks had automated their "back-room" operations, but were still using people for all their customer interaction. During the 1970s, though, both technology and customer demographics changed. On the one hand, automated teller machines (ATMs) were becoming reliable and cost effective, and on the other, customer demographics were changing and banks were under pressure to provide more convenient access to their services.

Bankers couldn't really afford to provide essential full-branch service for 16 or 18 hours a day, so they installed ATMs to keep certain essential services available around the clock. The next step was to situate these ATMs away from existing bank branches. The choice for bankers was either to build $1 million branches that would cost a half a million a year to operate, or to install 24-hour ATMs for about a $100 thousand each and then $50 thousand a year for maintenance. From the banker's perspective, ATMs are a very cost-effective way of making their services available in more places and at more times.

Then, around 1976, bankers realized they could reduce their costs even further by sharing ATMs. Since there's usually a certain amount of excess capacity on an ATM that's only serving one bank's customers, some banks began to sell their unused capacity by sharing their resources with client banks and then charging a set fee for every transaction processed. Reciprocal sharing arrangements also provide participating banks with better distribution channels, which can mean a competitive advantage.

These developments have allowed banks to reduce staffing and cut back on hours in their regular branches, and to deal with competitive threats from companies like Sears and American Express that are entering the financial-services business with a cost structure that would drive unmodernized banks out of business. It's a traditional paradigm—when the structure of an industry changes, the old leaders are displaced if they can't keep pace.

DG. **So by the late 1970s you knew not only that ATMs were going to be important, but that ATM sharing was also going to be important. How did the idea of CIRRUS begin to take shape?**
Burchfield. Banks had already formed local and regional sharing arrangements. For example, I had developed a shared automated-teller-machine network in metropolitan Chicago called Cash Station. The next logical step was to provide access for customers regardless of where in the country they happened to be.

CIRRUS was started by 10 banks that wanted to expand their regional networks into a national network. These included BayBank of Boston, Manufacturers Hanover of New York, Mellon Bank of Pittsburgh, and First Chicago. The first discussions were held in December of 1981, and by July of 1982 CIRRUS was incorporated. The initial service that CIRRUS set out to provide was cash access for people away from home.

We knew we needed a certain critical mass to make the network successful, and so we worked to build the network up quickly. As of March 1985, we have 46 states covered. In building this network, we've created a technical infrastructure that's unprecedented in the banking industry. The number of member banks has

now grown to 1425, although only 16 of them are directly connected to the CIRRUS switch. For example, there are over 300 banks in Texas on CIRRUS, but they're all connected to a single node. The Texas regional network handles all intrastate traffic and forwards the remainder to CIRRUS via the Texas CIRRUS node. There are probably 150 million to 200 million transactions a year that are processed by CIRRUS member banks, and yet only a fraction of that goes through the CIRRUS switch. Thus, in many cases, CIRRUS is a brand name as opposed to a processing function. Our node-based architecture has worked out well, and we don't think the politics or the structure of the ATM market will ever be conducive to connecting individual banks directly.

DG. **What other services do you now support besides cash access?**
Burchfield. We support withdrawals and balance inquiry from checking, savings, and line-of-credit accounts, and we are also developing direct debit point-of-sale services that will allow us to put machines in retail outlets. This summer we're connecting to a Canadian ATM network, and we're planning to do automatic currency conversion for international transactions.

We're also considering the possibility of taking deposits over CIRRUS. Interbank deposit taking already exists on a regional level—on the Chicago network I was involved with, the bank that received a deposit would prove and verify the deposit envelope, process the checks, and return bounced checks to the depositor's bank. The actual transfer of funds between banks is handled electronically.

DG. **When CIRRUS first started, did you analyze all the different exposures of the system and try to put together a study that could convince all of your participating banks that CIRRUS was going to be secure?**
Burchfield. To resolve the security issue, we assigned liability for all of the things that could possibly go wrong. For example, all of our connection nodes have financial and data security responsibility for their traffic. Extensive logs are kept for audit purposes to help assign liability. This gives each CIRRUS node an active interest in running a secure system. The employees of each bank are bonded, and the switch is operated by a bank with extremely strong audit functions. We also have provisions in our contracts that prohibit banks from using transaction data for competitive analysis.

AS. **With all this delegation, it sounds as though the CIRRUS organization itself doesn't have to worry about liability.**
Burchfield. That's correct. For example, the recent failure of certain savings and loan organizations in Ohio didn't cause us any concern because their CIRRUS attachment node, Central Trust in Cincinnati, is liable for their activity. If the savings and loans went defunct, CIRRUS would not be out of any money because Central Trust is responsible for their transactions.

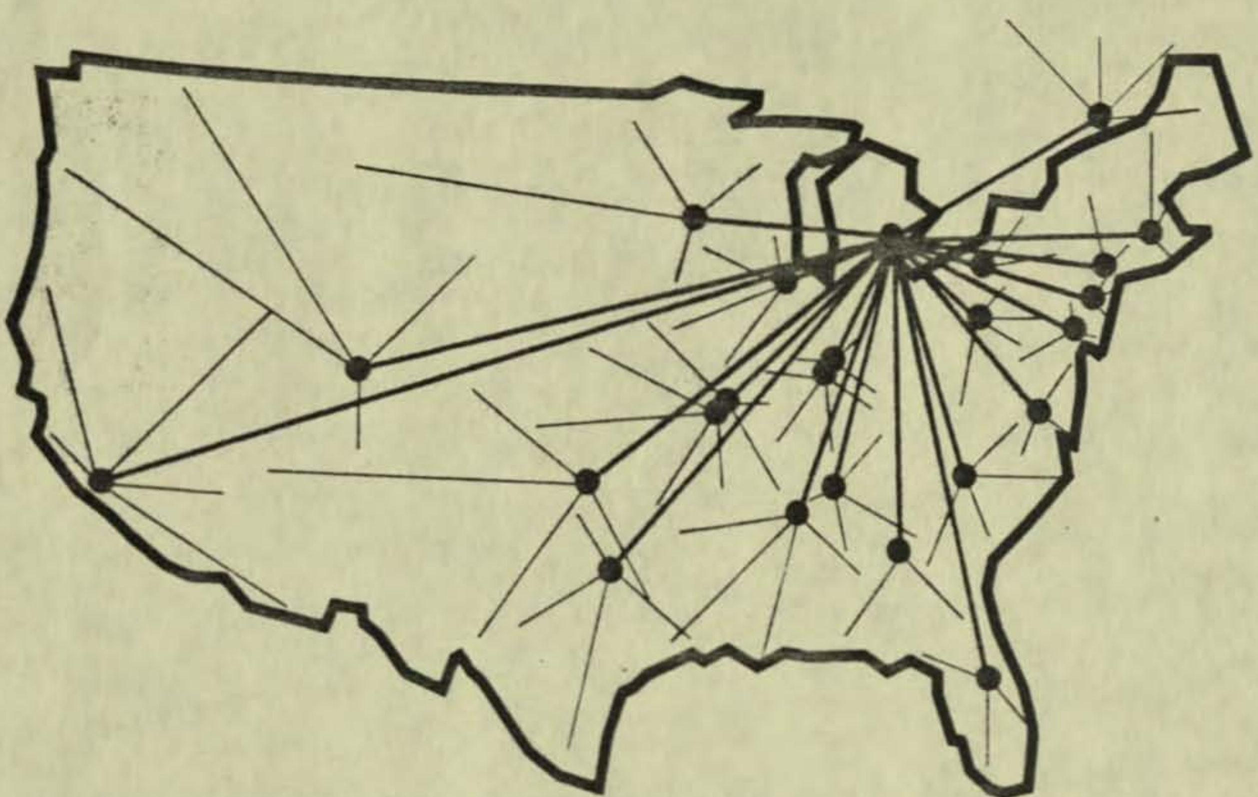
BRUCE BURCHFIELD
Bruce Burchfield is president and chief operating officer of CIRRUS Systems, Inc. He was formerly vice-president and manager of the electronic-banking service division at the First National Bank of Chicago, where he was responsible for the development and management of new electronic-banking services, including electronic funds transfer equipment and First Chicago's shared ATM network, Cash Station. Before joining First Chicago, Burchfield was an engineer for Reynolds Aluminum.

Central Trust actually shut their CIRRUS access off as soon as there was a problem.

DG. **How are consumers protected against the misuse of their cards?**
Burchfield. They're protected by Federal Regulation E, which says that a customer is not liable for the use of lost or stolen cards as long as the loss is reported within 2 days. When a lost card is reported within 60 days, but after 2 days—that is, within the amount of time it takes to get a bank statement—liability is limited to $50. The issuing bank is responsible for any ATM-related problems that a customer may have. That's what we tell customers when they call us—to contact their issuing bank. That's really all we can do, since we have no authority relative to accounts.

DG. **Do you provide any services directly to consumers?**
Burchfield. Only one—an 800 number that consumers can call to locate a nearby CIRRUS machine. The database of ATM locations is a geographical matrix that's accessed by area code and telephone exchange. The 800 operator asks for this information and then provides the locations of some nearby terminals. This service is subcontracted to an outside firm.

DG. **If I ran a bank and felt that I was at a competitive disadvantage because I didn't belong to an ATM network, and I wanted to join CIRRUS, how would you deal with my request?**
Burchfield. To answer that question, let me start by telling you something about our structure (see Figure 1). We have three types of members—Principal, Associate, and Corresponding members. Principal and Associate members have direct links to the CIRRUS switch and



The CIRRUS network consists of 16 4.8 kbit/second lines that radiate from the CIRRUS switch at the National Bank of Detroit (NBD) to the various nodes. Principal and Associate members are directly connected to the CIRRUS switch at NBD. A Principal member has exclusive marketing rights for a particular area, and an Associate member has direct access, but not exclusive rights. Corresponding members are connected to the switch through Principal or Associate members.

FIGURE 1. Topology of the CIRRUS Network

the right to franchise CIRRUS to other financial institutions. Principals and Associates are primarily large financial institutions that operate regional networks.

These Principal and Associate members can bring other banks into the network as Corresponding members. These are generally smaller institutions for which a direct connection would not make economic sense. Therefore, whenever a small bank approaches CIRRUS directly, I refer it to the Principal or Associate member that offers CIRRUS in the bank's state.

***DG.* What would happen if Bank X joined a regional ATM network like Cash Stream, without actually joining CIRRUS, and tried to send transactions through the CIRRUS switch via the Cash Stream CIRRUS node?**
Burchfield. Well, first of all, Cash Stream could sign up Bank X as a member of their regional network, and therefore as a member of CIRRUS, but if Cash Stream actually did send the CIRRUS switch a transaction that originated at Bank X, without working through a regional network, that transaction would be rejected because Bank X would not be in the financial institution table at the CIRRUS switch. It wouldn't be a valid transaction.

***AS.* Who are your direct competitors?**
Burchfield. There's the Plus network, which is run by a group of banks, and Master Teller, which is operated by Master Card. As of the end of 1984, Plus had 4700 ATMs, Master Teller had 3250, and CIRRUS had 6500. Plus has a competing networks rule that prevents their members from joining another network. Master Teller doesn't have such a rule and neither do we, so some banks belong to both CIRRUS and Master Teller. Master Teller is very big on credit functions, of course, but their card base is limited as far as debit transactions or checking accounts, so there's a certain advantage for customers in being connected to the two networks.

***DG.* So if there were two banks that were connected to both CIRRUS and Master Teller, they would have two different ways of routing transactions between themselves. Does that make transaction switching a competitive business?**
Burchfield. Transaction switching is not competitive right now because the bank that decides which network to use is not the bank that pays for the transaction. The destination bank has to pay for the transaction and would thus like to receive it over the more

economical network; the acquiring bank, on the other hand, wants to issue the transaction in the most convenient way, which is probably the least favorable to the user.

AS. **How do CIRRUS member banks recover the cost of processing transactions for other member banks?**
Burchfield. The issuing bank pays the deploying bank 50 cents for a withdrawal and 25 cents for a balance inquiry. The switch gets an additional 25 cents per transaction. Right now, most CIRRUS member banks offer CIRRUS services for free, but my assumption is that members may eventually pass their costs on to the customer. Customers will then have to compare the cost and convenience of CIRRUS with traveler's checks and other alternatives.

AS. **Do you foresee the establishment of gateways for connecting CIRRUS, Master Teller, and Plus?**
Burchfield. We don't think there's any need for connecting ATM services. We see our independence as a competitive advantage. For point-of-sale services, though, it makes sense to allow merchants to accept different kinds of cards. When point-of-sale machines are connected to more than one network, it becomes logical to standardize fees and message formats. These issues are already being addressed by the American Bankers Association and the Electronic Funds Transfer Association (a multi-industry trade association for electronic funds transfer). CIRRUS, Plus, and Master Teller are already meeting to talk about technical standards, to try to build a common gauge railroad, so to speak, so that point-of-sale transactions won't happen in a fragmented way. We're all very competitive, but we realize that it's not going to work unless there are standards.

Interview 2. Jay Levy, director of systems and operations for CIRRUS Systems, Inc., talks about the design and implementation of the network.

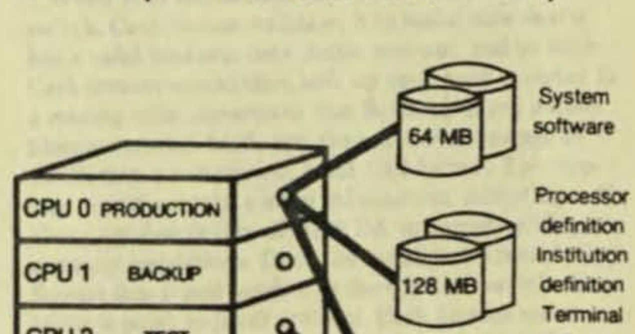
AS. **Jay, we'd like you to take us over some of the technical details of the CIRRUS system. Could you start by giving us an idea of how it's organized?**
Levy. Sure. CIRRUS is a star network, with a message switch at the National Bank of Detroit that currently connects to 16 nodes via 4.8 kbit/second dedicated lines. If a line fails, it's replaced by a dial-up circuit that runs at the same speed. The communications protocols are bisync point to point. The CIRRUS switch is the secondary point on each line, so each node processor has primary access for the bid on its line.

The switch itself is a Tandem NONSTOP II system with four CPU's (See Figure 2). Each CPU has 2.4 megabytes of main memory. The system has a pair of 256-Mbyte drives, a pair of 128-Mbyte drives, and a pair of 64-Mbyte drives. These are shared by all of the CPUs. Most of the switching functions are handled by one Tandem CPU with another acting as backup. When we run tests, we take two of the four CPUs and configure a test system.

JAY LEVY
Jay Levy is director of systems and operations for CIRRUS Systems, Inc. Prior to joining CIRRUS, Levy was an assistant vice-president at the National Bank of Detroit (NBD), in charge of design and development pertaining to NBD's operation of the CIRRUS ATM interchange switch. He was also responsible for developing the ATM program from a proprietary system to a regional network, and assisted in the development of telephone-bill payment services, videotex banking applications, and international retail banking operations.

The 256-Mbyte disks are used for daily transactions—one disk drive for primary and one for backup. The 128-Mbyte disks are used for control files, processor status files, terminal files, institution-level files, and holiday records. They also contain the routing tables and the system definition file, although these are kept in main memory when the system is in operation. The 64-Mbyte disks contain system files.

The switch sits next to another Tandem system at the National Bank of Detroit that could be reconfigured to take over the CIRRUS load if the primary system failed. About the only thing that would cause a real disaster is a power failure, since there is no backup



The CIRRUS switch uses redundant CPUs and disk drives to maintain a high degree of availability and reliability.

FIGURE 2. CIRRUS Switch Configuration

power supply at the switch site. However, NBD does have a backup site that wouldn't be affected by a power failure in Detroit.

DG. **How much log data do you accumulate on a day-to-day basis?**
Levy. A log record is 411 bytes. There are two log records written for each transaction. One is written into the log file that corresponds to the transaction's anticipated date of settlement. The log file for a Monday would contain between 24,000 and 40,000 records. The switch stops accepting activity for a current date at 8 P.M. Detroit time, which means cutting over to the new log file. Just before cutover, approximately 99 percent of our traffic is already going to the new date of settlement, as specified by the terminal owner.

The other log record is to a backup transaction file that's in time-stamp order. We've been fortunate in that we've never had to use the backup log file.

AS. **What do you do with the log that you've accumulated each day?**
Levy. The log is the source of the batch reports that we send to each node bank's data-processing organization each day. A node's report lists all of its activity for the day in time-stamp order. The report has separate sections for inbound and outbound traffic. The batch run is also used to verify the on-line settlement totals that indicate the balance of each node with the switch. If the batch run produces totals that are consistent with the on-line totals, settlement is effected by transferring funds between "due from" and "due to" accounts that are maintained by each node bank at NBD. The actual transfers are accomplished by the money-management departments of member banks. We only do settlement on a node-to-node basis—each node must deal separately with its member banks.

AS. **What happens when the batch node balances don't agree with the on-line node balances?**
Levy. We look for the reason. Typically we'll be out 25 or 50 cents for a missed transaction fee somewhere. We've never had a serious discrepancy in a consistency check.

AS. **What is the peak transaction load on the switch?**
Levy. The switch was designed to handle 2 transactions per second, which is 16 messages a second. I don't think we've ever achieved that peak—we have never exceeded 500,000 transactions a month. The total elapsed time for a transaction averages between 12 and 15 seconds from request to cash dispense. This includes the processing time at a node bank for approval and debit.

AS. **Suppose I had an account at BayBank and that I went to an ATM run by Landmark Savings and Loan in Pittsburgh, which happens to be a member of the Cash Stream network. Mellon runs Cash Stream and is a node on the CIRRUS network. What happens when I put my BayBank card in the Landmark ATM and request $100?**
Levy. You would get a *customer lead through* on the screen that would look exactly like the lead through for regular Landmark customers. The ATM wouldn't know who you were yet, and so its processor would ask you for your personal identification number (which it would temporarily buffer), what type of account you wanted to access (i.e., savings or checking), and the dollar amount of your transaction.

By this time the Landmark ATM would have already read Track II of the magnetic strip on the back of your BayBank card, and determined your account number, expiration date, and some other variable data that are dependent on BayBank. The first 3 to 11 digits of an account number specify the issuing bank. Track II holds 40 bytes, although we're only using 19 of them right now. Tracks I and III we don't use at all—Track I is for the airline industry, and Track III is used in off-line banking transactions, but not in the interchange environment.

To get back to your transaction, though, the host processor at Landmark has not been involved at all up to this point. Once the ATM has collected all of the data it needs for your $100 withdrawal, it sends a message to its host at Landmark, which determines that the account number on your BayBank card is not a Landmark account, and creates a message to Cash Stream. The Landmark host also writes a suspense record to its log so that it can time out the transaction and free the ATM if Cash Stream does not reply after a certain amount of time. The Landmark host would probably give Cash Stream about 34 seconds before doing this. Most ATMs also have internal timers—we suggest that ATMs abort transactions and return consumers' cards if they do not hear from their hosts in 45 seconds.

When your transaction arrives at the Cash Stream switch, Cash Stream validates it to make sure that it has a valid business date, dollar amount, and so forth. Cash Stream would then look up your bank number in a routing table, determine that BayBank is not a Cash Stream member bank, and then create a message to CIRRUS in a variation of ANSI X9.2 format. The message would contain a lot of information, including a 40-character description of the ATM, to comply with government regulations. Once the message is created, Cash Stream logs it and sends it to the CIRRUS switch via a bisynch point-to-point protocol. Cash Stream waits up to 24 seconds for a response before aborting the transaction.

CIRRUS receives the message, looks up the issuing bank number in its routing table, and then forwards the message to the CIRRUS node that the issuing bank is connected to, which in this case is BayBank. The CIRRUS switch sets a timer for 20 seconds and then waits for the response from BayBank. BayBank receives the message, verifies that the personal identification number (PIN) you supplied is correct, and then checks your balance. If you check out, BayBank writes a memo

post record on your account indicating that a $100 withdrawal is in progress, and then sends an approval message back through CIRRUS, Mellon, and Landmark to the ATM, which dispenses your $100.

The memo post record that BayBank has written is simply noting that a withdrawal is in progress. It will *not* cause $100 to be deducted from your account. That happens with the second round of messages, which I haven't yet discussed.

When the ATM has actually dispensed cash into your hands, it sends a message to Landmark saying that it has given you $100. The completion message goes from the ATM through Landmark and Mellon to CIRRUS, and then CIRRUS sends a message back to Landmark acknowledging the completion message. CIRRUS also sends the completion message on to BayBank, which uses it to create a "postable" record that is used to deduct the money from your account. A completion acknowledgment then goes back from BayBank to CIRRUS. Altogether, your transaction has required four CIRRUS messages—two for the initial approval and two for the completion.

If CIRRUS does not receive a completion acknowledgment from BayBank after 120 seconds, the switch continues sending until it does receive one. If for some reason BayBank cannot deal with the completion message, they call us and we purge it from our file.

Approximately half of our member banks use this protocol. The other half use a shorter protocol that eliminates the second two messages. In our example, this would mean that the CIRRUS switch would still get the completion message from the ATM, but that it would not forward it to BayBank. When this shorter protocol is used, the issuing bank must simply assume that a withdrawal has actually taken place. Obviously, if there's a problem then a message must be sent to the issuing bank instructing it to compensate for the debit it posted.

AS. What would happen if BayBank had sent an approval to the Landmark ATM and I had received the $100, but the completion message had not returned from the ATM to BayBank because the Mellon Cash Stream system had failed?

Levy. If the CIRRUS switch does not get a completion message from Mellon within 120 seconds, then Mellon would be marked down by the switch and would be placed off line for 10 seconds. CIRRUS would send a message to BayBank saying that the cash had not been dispensed. After the 10 seconds had elapsed, we would bring Mellon on line by issuing a series of network-management messages. If the completion message from Mellon came in after the 120-second time had expired, but within the same settlement day, the CIRRUS switch would interrogate the log file to see if there was a record for that message. If there was, the switch would automatically build adjusting entries to both sides of the transaction to ensure that the settlement was handled properly. If the late completion message came in after the settlement cutover for the day, or never came in at all, Landmark, as the ATM operator, would have to present an off-line paper adjustment to the switch, along with some kind of physical evidence of the transaction, like an audit tape or an internally produced report indicating that the transaction had occurred, and a CIRRUS report indicating that they hadn't received value for the transaction. Without that documentation they wouldn't get their money. This kind of thing is not uncommon—some ATMs don't automatically send completion messages to the host. The only way to discover what's happened is to get a status of prior transactions after a new transaction starts.

DG. Could you review the way the time-out intervals on your system interact?

Levy. We have a 20-second timer at the switch for messages to nodes; we advise sending nodes to set a 24-second timer to allow some transmission time. A bank that's not directly connected to the switch, like Landmark in your previous example, may wait as long as 34 seconds to allow the intermediary bank some processing time. We typically tell the ATM issuers to set their timers for about 45 seconds; if the terminal doesn't get a response within that time, it shuts down and returns the customer's card.

DG. The two message protocols you described—the four-message protocol and the two-message protocol—make radically different assumptions about when to deduct money from an account, don't they?

Levy. Traditionally, the U.S. banking industry won't ever deduct anything immediately. We have very few real-time posting, checking, or savings applications in the U.S. That convention doesn't always hold true for other countries, though. In Canada, for instance, there are systems that use real-time posting, which means that a single customer transaction can post several items on a cardholder's account. In the U.S., banks use on-line systems for maintaining account balances. Posted records from a system are merged with other instrument activity (such as checks) during batch processing. The daily batch run is what actually deducts the money from accounts. The result of a daily batch run is a file containing the starting account balances for the system for the next day, as well as records for pending transactions.

DG. From what you've been saying, there seems to be little concern that any of the participating banks might be doing anything dishonest. What would happen if a member bank immediately acknowledged that an ATM had dispensed cash when in fact it hadn't. Wouldn't that bank be credited for $100 while the customer was left standing in the rain without any money in his hand?

Levy. Customers would let us know if there was anything of that nature going on. Also, there's a significant amount of federal regulation that protects customers in

those situations, so it's hardly in the member bank's best interest to do such a thing.

AS. **But the safety and security of your network depend very much on the safety and security of its constituent parts.**

Levy. Absolutely. CIRRUS was developed as a pipeline between banks that had been doing this kind of thing for 10 or 12 years before we came along. We're another link of potential authorization for the banks. All of the policies and procedures that banks had evolved to protect themselves from error and fraud were carried over directly to the CIRRUS environment.

DG. **You mentioned before that you partition any part of your network that doesn't respond within a certain amount of time. Is that an important part of your system strategy?**

Levy. Yes. I'm probably one of the few proponents of this kind of partitioning. The more popular strategy is to keep resending messages until a response is received. I believe that the financial implications of transaction processing in the ATM environment require this kind of scrupulousness. It has proved easier for us to balance the network by guaranteeing that transactions flow out of the store-and-forward process in a logical sequence. We've also found that if you give a node time to recover you can often prevent more serious problems.

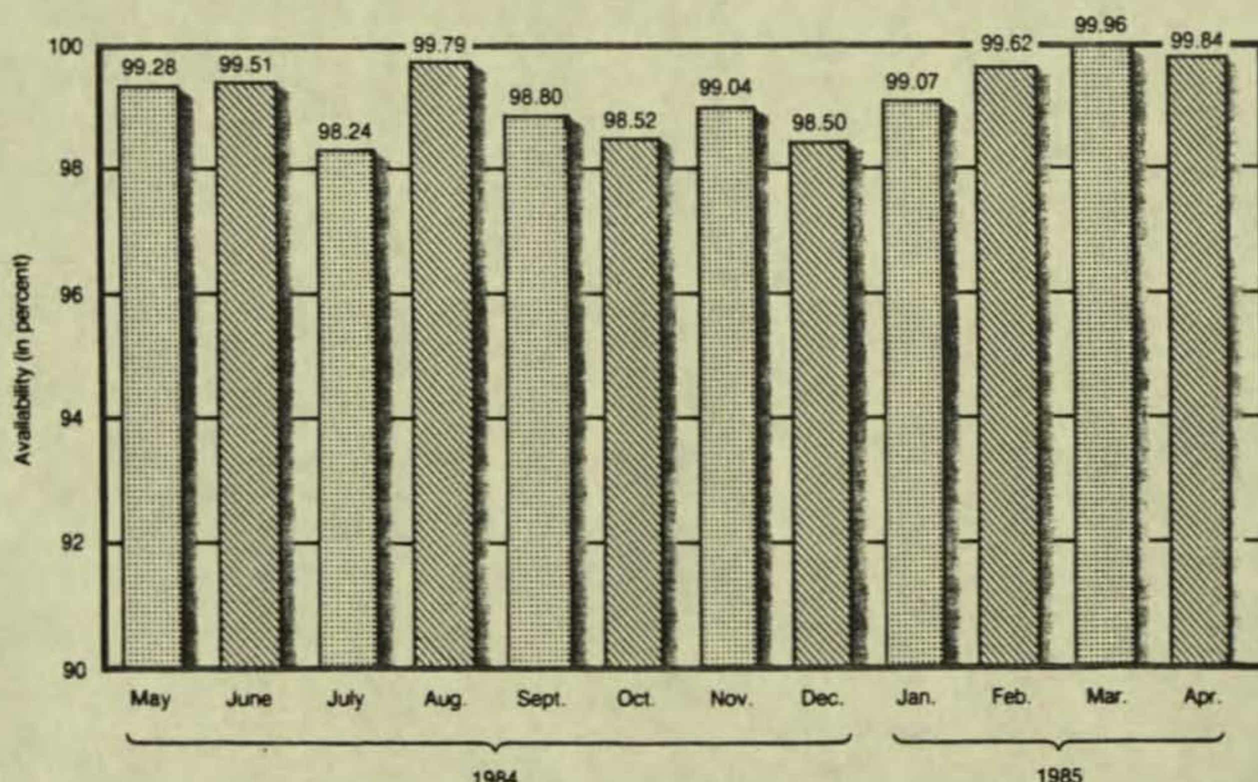
DG. **How often do banks go off line?**

Levy. We're working to get that rate down to a minimum. We have a standard that no bank should be off line more than five percent of the time, averaged over a month. Some of our members are well above that mark, and some are well below it, but most are off line no more than five percent of the time. Nearly all of our member banks use highly available systems, but as I mentioned earlier, if a node does not get a completion message in time from an ATM, CIRRUS may take the entire node off line. Of course in the case of a delayed completion, a message goes out 10 seconds after a node goes off line to bring it back up again. These short ups and downs have a minimal effect on transaction processing.

DG. **CIRRUS itself could also go down, couldn't it?**

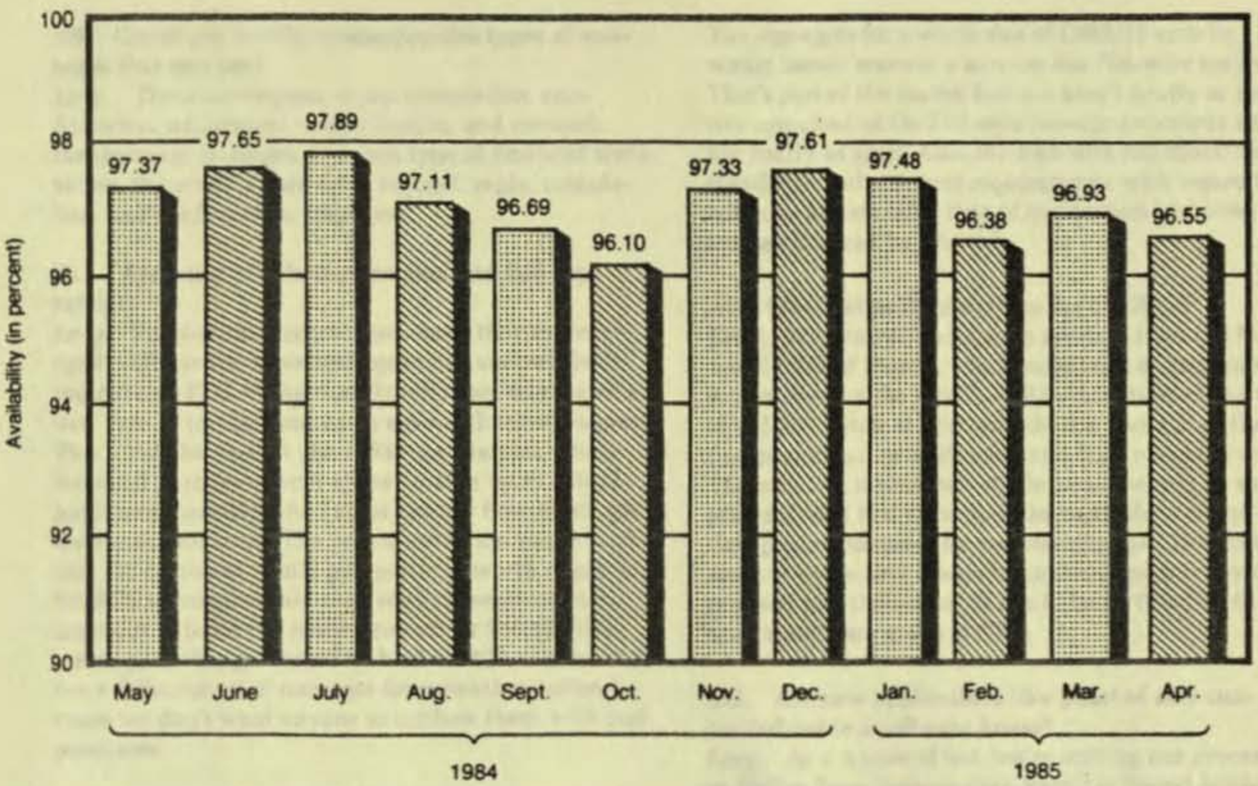
Levy. It's not likely. The average availability of the CIRRUS switch exceeds 98 percent on any given day (see Figure 3). We do not allow scheduled outages of the CIRRUS switch. Actually, we allow one a year, but it still goes against that 98 percent. Tandem equipment does not require outages for hardware maintenance.

DG. **Considering communication-line failures, central-site failures, and node failures, what's the probability that one node will be able to access another at any given time?**



The average uptime on the CIRRUS switch is between 98 and 100 percent over a 24-hour period. Scheduled downtime is counted as unavailable time.

FIGURE 3. Average Availability on the CIRRUS Switch



Network availability is computed as the probability that one CIRRUS node can access another at any given time. Network availability can be affected by failures in any part of the network, from the switch to the communication lines to the nodes.

FIGURE 4. Average Availability on the CIRRUS Network

Levy. About 97 percent (see Figure 4). In February, for example, the average network uptime was 96.3 percent, which covered a range for individual nodes from 90.13 to 99.6. The 90.13 node was either having a line problem or a processor problem. The mean daily average availability for the switch during February was 99.62 percent.

***AS.* What kept you off the other 0.38 percent of the time during February?**
Levy. The backup file just got incredibly large over the three-day Washington's Birthday weekend, and we had to shut the system down for a minute or two (or fifteen) to reallocate some file space. There was also a date problem because some of our members wanted to process on that Monday.

***DG.* The switch was closed?**
Levy. The switch observes the seven holidays that all 12 Federal Reserve offices observe as federal holidays. Because the National Bank of Detroit isn't open, there's no way to move the money, and if the Fed's not open, there's no way to fund an account at NBD. Certain banks that close for Lincoln's Birthday stay open for Washington's, though, and they were trying to present transactions when there wasn't a valid log file open, and that probably led to some downtime.

***AS.* What fraction of the transactions that reach CIRRUS have to be rejected because they're not destined for a member bank?**
Levy. About 15 percent.

***AS.* Could you describe the message protocol you use?**
Levy. It's called the ANSI Interchange Message Specification for Debit and Credit Card Messages on Financial Institutions—ANSI X9.2-1980. It's a guideline, as all standards are guidelines. It defines 64 fields that are usable in a financial message, and it indicates which fields are mandatory for the type of message being sent. At CIRRUS we've enhanced this standard to meet our particular needs. We use all of the mandatory fields, along with some additional ones. There's a bitmap at the head of every message that shows which fields are being used. We have a fixed set of about 40 message formats that we use. The smallest uses about 170 bytes; the largest, a reversal message, over 400. Reversals are long because they contain a great deal of data from the original transaction. For each type of message, we have both a long and a short format. The long format has all of the fields filled in, and the short one requires that missing fields be filled in by the switch.

AS. **Could you briefly summarize the types of messages that you use?**
Levy. There are request, reply, completion, confirmation, adjustment, reconciliation, and network-management messages. For each type of financial transaction, there's an entire set of request, reply, completion, and confirmation messages.

DG. **Are point-of-sale transactions handled separately?**
Levy. Point-of-sale transactions, once they're developed, will use the same messages with various fields indicating a POS transaction. For POS we've created a new type of transaction that is a preauthorized request. This would be used at places like gas stations where intended purchases don't always match up to actual purchases (because who knows exactly how much gas their tank can hold). The preauthorization might be for $20; the customer would pump $18.50 worth of gas, and the POS terminal would then send subsequent messages out to build the posting record for $18.50. The memo post, though, would be held for $20. We want to use a different set of messages for preauthorization because we don't want anyone to confuse them with real purchases.

AS. **Is preauthorization similar to the failure condition that occurs when an ATM can only distribute part of a requested sum?**
Levy. Yes, except here the discrepancy is the norm instead of the exception.

DG. **What are your message latency goals at the switch, and how can you maintain them as your volume increases?**
Levy. We don't allow the switch any more than 2 seconds for processing a message. The actual time is usually somewhere between 0.7 and 1 second. The switch is modular enough that if we ever needed to we could open up a mirror of the application so that there could be multiple processors running the application simultaneously, although there would only be one transaction running in any processor at any time. I think we've actually done this at peak processing time.

AS. **Do you spool log data onto magnetic tape for permanent safekeeping?**
Levy. All the important files, including the log files, are backed up every night. Once a week we back up the entire system, including the application program, and transport it to an off-site storage location. We also put a digested form of the log on microfiche and keep it for five to seven years, to meet regulatory requirements.

DG. **How would you compare your electronic funds transfer network to the Fed-wire?**
Levy. Well, the Fed-wire is a wholesale-level EFT network, and CIRRUS is a retail-level EFT network. The average dollar value per transaction on the Fed-wire is probably a million times what we handle on CIRRUS. The aggregate for a whole day of CIRRUS activity would barely warrant a wire on the Fed-wire system. That's part of the reason that we aren't nearly as security conscious as the Fed-wire is—our exposures are not nearly as great. Also, the Fed-wire has much more significant and stringent requirements with regard to who can initiate what type of transaction and how transactions can be effected.

AS. **Who writes the programs for CIRRUS?**
Levy. We contract the system services from the National Bank of Detroit, which maintains a programming staff to support the switch. NBD subcontracted the writing of the switch to a software house, but did all the design and testing on its own, and NBD owns the code. The software is relatively stable now. I would be surprised if NBD had more than the equivalent of one full-time person budgeted for maintaining the CIRRUS software. When a new release of applications is ready to be released into the system, it has to be certified by the system software group at NBD.

DG. **Are new applications like point of sale contracted out to a software house?**
Levy. As a matter of fact, we're moving our processing to Mellon Bank because they were the lowest bidder for the POS product. Mellon is going to have to redevelop the switch software.

DG. **How many lines of code are in your on-line switch application?**
Levy. The on-line switch software, which includes message routing and editing, is approximately 9000 lines of TAL, which is the Tandem application language.

DG. **How big is your routing table?**
Levy. Currently it consists of about 3500 records, with about 20 bytes per record. A routing-table entry includes a bank's prefix, which is from 3 to 11 digits, and an identification of the processor that services the bank. The routing table is called the FIT, for *financial-institution table*. We send an up-to-date copy of the FIT to our member banks every day. It's our intention to build an on-line message that keeps FIT copies up-to-date. The problem is that bringing a new member on can mean bringing as many as 100 new banks on with it. A hundred new FIT entries at one time is a data-management problem that we haven't solved yet.

AS. **Do you test a member bank's system before letting it on the CIRRUS network?**
Levy. Yes, we certify nodes with a battery of test cases. Every situation we can think of is attempted prior to putting a node on the network. The book that we keep the test cases in is over three inches thick—I believe it contains over 2500 test transactions. When we certify a node, we ask the candidates to have an ATM available in a test environment. We send them plastic cards that run against a simulated card proces-

sor at the switch in Detroit. The switch can also send simulated ATM activity out to them. We find that using an actual ATM is the best way to test.

***AS.* How is a typical test case described?**
Levy. In great detail. This is a *very* critical part of our operation. I think part of the success of the network, in fact, can be attributed to the stringency of our testing. We send candidate banks a list of cards and a set of conditions that the cards have to meet. Some of these cards specify accounts that don't exist or that have incorrect balances. A test case instructs the candidate bank to use a particular card for a particular transaction, and then indicates the expected result. We test weekend and holiday processing and all the exception conditions. We also keep our set of test cases up-to-date as we add new functions to the switch.

***AS.* What percent of your budget eventually goes for testing?**
Levy. I would say that as a percentage of overall development, testing is easily 25 percent. In an environment like ours, we have to be very careful.

***AS.* Have you had any situations where a member bank did not perform properly?**
Levy. Yes, we've had member banks that stopped sending completion messages for one reason or another. On one occasion we had 1500 adjustments for one day's activity. These had to be presented on paper. This swamped their operation center as well as ours. We have also had requests for adjustments that came in as much as 50 business days after a transaction. It's difficult to accept activity that late.

***DG.* Could you discuss your basic strategy for security?**
Levy. Within the banking environment, there's a great deal of physical security in addition to application-level security. Most any data that are maintained in a bank's data-processing center are considered confidential and secure. The only part of the system that we don't normally have physical control over is our communication lines, which are protected with link encryption provided by the Racal-Milgo DataCrypter II with a public key option. This is a commercial product that uses DES for encrypting messages. It includes a public key algorithm that allows us to distribute new master keys from time to time.

***AS.* Have your communication lines ever been actually attacked?**
Levy. Not that I know. The biggest area for fraud in ATM banking is card and PIN compromise. Either a family member takes a card and cleans out an account, or someone obtains a card and its PIN by force. It's as easy to defraud an ATM that way as to tap the line.

***AS.* Are you interested in "smart cards"?**
Levy. They would allow us to put a great deal more data on plastic and would let us do away with PIN verification at the issuing bank. They would also allow us to maintain value on cards, so that customers would be able to put, say, a thousand dollars worth of value on a card that could then be spent without on-line verification. The thing that concerns me is how could we get the postable information to the cardholder's bank and actually deplete his account balance. I see the biggest application of chip cards in this country for things like food stamps where the value is depleted but the transactions don't have to be posted. It's not clear what application smart-card technology will have in banking, which may be why it's taking so long to be developed here. There's also the problem of refitting hundreds of thousands of terminals, and there would have to be a standard for where to put the contacts on the cards.

***DG.* Have you considered digital signature verification for point-of-sale transactions?**
Levy. I have my doubts. Handwriting can be compromised. And it's hard to say what kinds of controls we're going to need until we really find out how easy it's going to be to deploy POS. I'm not convinced that signature technology is that much more secure than PINs.

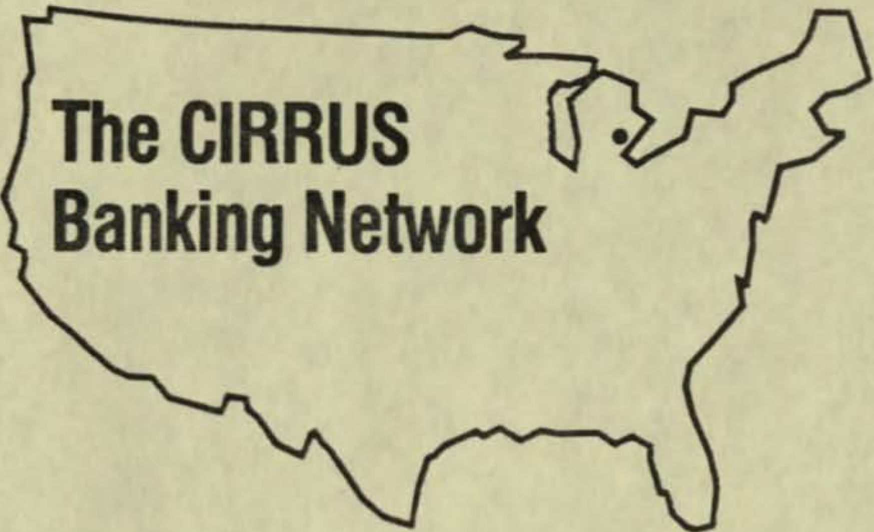
***DG.* What's the largest transaction that can be issued over the CIRRUS network for a cash withdrawal?**
Levy. Theoretically, the field size would allow for a $100,000 transaction, but I don't know of a machine in the country that would let you do that. There are machines in Las Vegas that would probably let you take $1000 out in one shot.

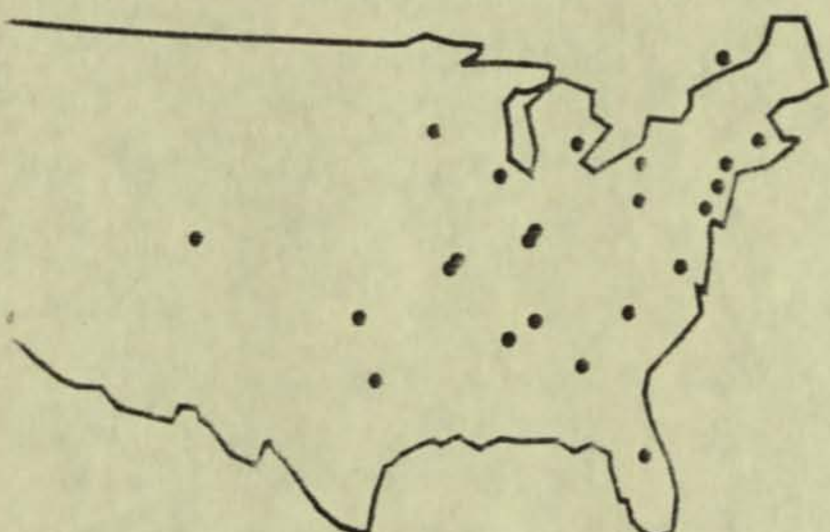
***AS.* Will future ATMs have change-making capabilities?**
Levy. Yes, there's a machine that makes change, and five-bill dispensers that can give ones, fives, tens, twenties, and fifties or hundreds. The more these machines can do, the more truly they'll be able to replace human tellers, which is really valuable in some applications. There may not be a need for change-making capability at O'Hare Airport, but if someone's going to an ATM regularly for their everyday banking business, it should be able to cash their paycheck. Not everyone's paid in increments of $10.

***DG.* I think that's all the questions we have for you. Bruce, Jay, thanks for telling us about the CIRRUS system.**

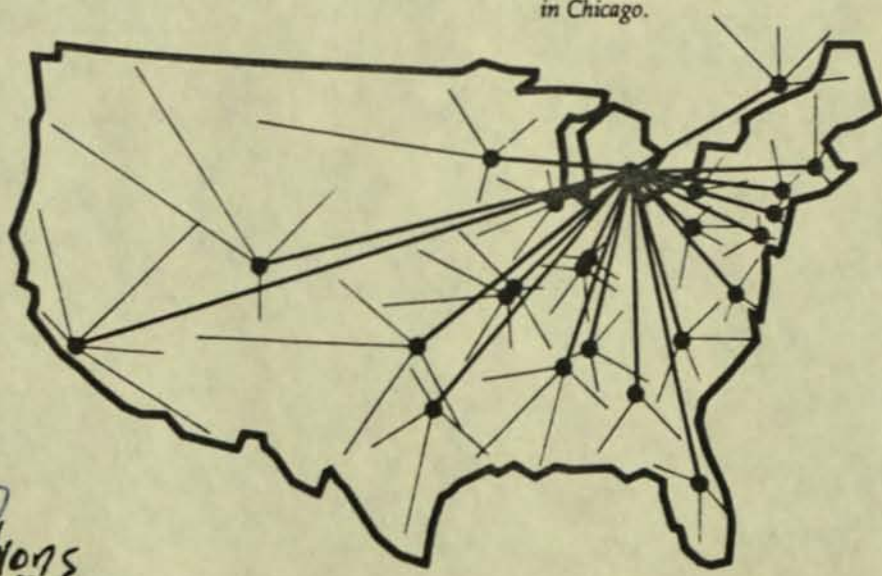
CR Categories and Subject Descriptors: C.4 [**Performance of Systems**]: *reliability, availability, and serviceability*; D.2.9 [**Software Engineering**]: Management—*software quality assurance*; D.4.0 [**Operating Systems**]: General; J.1 [**Administrative Data Processing**]: *financial*; K.6.3 [**Management of Computing and Information Systems**]: Software Management
General Terms: Management, Reliability, Verification
Additional Key Words and Phrases: ATM, CIRRUS, EFT systems

Case Study: The CIRRUS Banking Network





This is the third case study to appear in Communications. *This time, case-study editors David Gifford and Alfred Spector interviewed the president and the director of systems and operations at CIRRUS Systems, Inc., a funds transfer network, to see how distributed systems are being employed in the banking industry. As the following transcript shows, the CIRRUS network provides security, lost message recovery, database consistency via atomic actions, recovery after network partition, and a framework for interorganization cooperation in a distributed system. The CIRRUS system is not the most complex computer system to have been described in these case studies; rather, it is an example of how a few well-chosen ideas can be the basis of a simple and useful system. The interviews were conducted on March 29, 1985, at the headquarters of CIRRUS Systems, Inc., in Chicago.*



THE CIRRUS BANKING NETWORK

The CIRRUS banking network makes coast-to-coast automatic banking transactions possible. The system will soon be able to handle international currency transactions and point-of-sale transactions in stores.

DAVID GIFFORD and ALFRED SPECTOR

Interview 1. Bruce Burchfield, president of CIRRUS Systems, Inc., talks about the development of the CIRRUS network, its present scope, and plans for its future.

***DG.* Bruce, could you start by telling us something about the history of the CIRRUS network?**

Burchfield. I'd be glad to. Bankers have traditionally provided customers with personalized service. That's the nature of the business. By the 1970s, most banks had automated their "back-room" operations, but were still using people for all their customer interaction. During the 1970s, though, both technology and customer demographics changed. On the one hand, automated teller machines (ATMs) were becoming reliable and cost effective, and on the other, customer demographics were changing and banks were under pressure to provide more convenient access to their services.

Bankers couldn't really afford to provide essential full-branch service for 16 or 18 hours a day, so they installed ATMs to keep certain essential services available around the clock. The next step was to situate these ATMs away from existing bank branches. The choice for bankers was either to build $1 million branches that would cost a half a million a year to operate, or to install 24-hour ATMs for about a $100 thousand each and then $50 thousand a year for maintenance. From the banker's perspective, ATMs are a very cost-effective way of making their services available in more places and at more times.

Then, around 1976, bankers realized they could reduce their costs even further by sharing ATMs. Since there's usually a certain amount of excess capacity on an ATM that's only serving one bank's customers, some banks began to sell their unused capacity by sharing their resources with client banks and then charging a set fee for every transaction processed. Reciprocal sharing arrangements also provide participating banks with better distribution channels, which can mean a competitive advantage.

These developments have allowed banks to reduce staffing and cut back on hours in their regular branches, and to deal with competitive threats from companies like Sears and American Express that are entering the financial-services business with a cost structure that would drive unmodernized banks out of business. It's a traditional paradigm—when the structure of an industry changes, the old leaders are displaced if they can't keep pace.

***DG.* So by the late 1970s you knew not only that ATMs were going to be important, but that ATM sharing was also going to be important. How did the idea of CIRRUS begin to take shape?**

Burchfield. Banks had already formed local and regional sharing arrangements. For example, I had developed a shared automated-teller-machine network in metropolitan Chicago called Cash Station. The next logical step was to provide access for customers regardless of where in the country they happened to be.

CIRRUS was started by 10 banks that wanted to expand their regional networks into a national network. These included BayBank of Boston, Manufacturers Hanover of New York, Mellon Bank of Pittsburgh, and First Chicago. The first discussions were held in December of 1981, and by July of 1982 CIRRUS was incorporated. The initial service that CIRRUS set out to provide was cash access for people away from home.

We knew we needed a certain critical mass to make the network successful, and so we worked to build the network up quickly. As of March 1985, we have 46 states covered. In building this network, we've created a technical infrastructure that's unprecedented in the banking industry. The number of member banks has

now grown to 1425, although only 16 of them are directly connected to the CIRRUS switch. For example, there are over 300 banks in Texas on CIRRUS, but they're all connected to a single node. The Texas regional network handles all intrastate traffic and forwards the remainder to CIRRUS via the Texas CIRRUS node. There are probably 150 million to 200 million transactions a year that are processed by CIRRUS member banks, and yet only a fraction of that goes through the CIRRUS switch. Thus, in many cases, CIRRUS is a brand name as opposed to a processing function. Our node-based architecture has worked out well, and we don't think the politics or the structure of the ATM market will ever be conducive to connecting individual banks directly.

DG. **What other services do you now support besides cash access?**
Burchfield. We support withdrawals and balance inquiry from checking, savings, and line-of-credit accounts, and we are also developing direct debit point-of-sale services that will allow us to put machines in retail outlets. This summer we're connecting to a Canadian ATM network, and we're planning to do automatic currency conversion for international transactions.

We're also considering the possibility of taking deposits over CIRRUS. Interbank deposit taking already exists on a regional level—on the Chicago network I was involved with, the bank that received a deposit would prove and verify the deposit envelope, process the checks, and return bounced checks to the depositor's bank. The actual transfer of funds between banks is handled electronically.

DG. **When CIRRUS first started, did you analyze all the different exposures of the system and try to put together a study that could convince all of your participating banks that CIRRUS was going to be secure?**
Burchfield. To resolve the security issue, we assigned liability for all of the things that could possibly go wrong. For example, all of our connection nodes have financial and data security responsibility for their traffic. Extensive logs are kept for audit purposes to help assign liability. This gives each CIRRUS node an active interest in running a secure system. The employees of each bank are bonded, and the switch is operated by a bank with extremely strong audit functions. We also have provisions in our contracts that prohibit banks from using transaction data for competitive analysis.

AS. **With all this delegation, it sounds as though the CIRRUS organization itself doesn't have to worry about liability.**
Burchfield. That's correct. For example, the recent failure of certain savings and loan organizations in Ohio didn't cause us any concern because their CIRRUS attachment node, Central Trust in Cincinnati, is liable for their activity. If the savings and loans went defunct, CIRRUS would not be out of any money because Central Trust is responsible for their transactions.

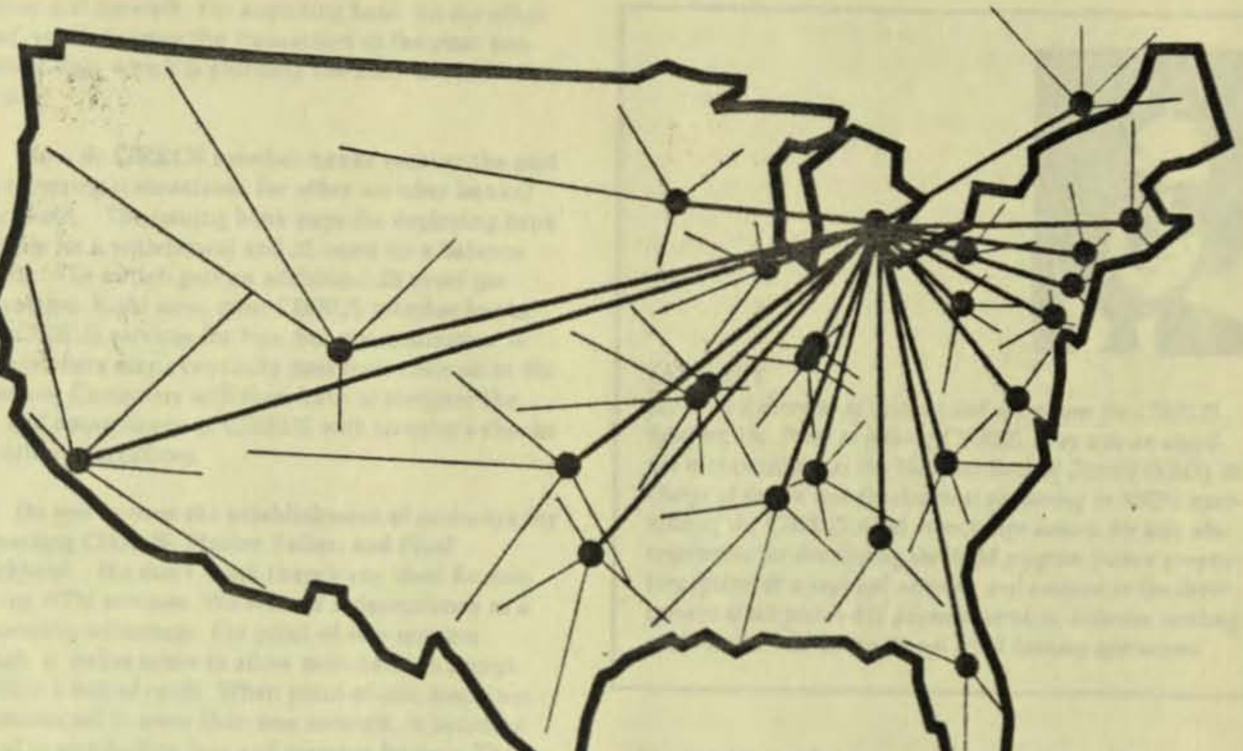
BRUCE BURCHFIELD
Bruce Burchfield is president and chief operating officer of CIRRUS Systems, Inc. He was formerly vice-president and manager of the electronic-banking service division at the First National Bank of Chicago, where he was responsible for the development and management of new electronic-banking services, including electronic funds transfer equipment and First Chicago's shared ATM network, Cash Station. Before joining First Chicago, Burchfield was an engineer for Reynolds Aluminum.

Central Trust actually shut their CIRRUS access off as soon as there was a problem.

DG. **How are consumers protected against the misuse of their cards?**
Burchfield. They're protected by Federal Regulation E, which says that a customer is not liable for the use of lost or stolen cards as long as the loss is reported within 2 days. When a lost card is reported within 60 days, but after 2 days—that is, within the amount of time it takes to get a bank statement—liability is limited to $50. The issuing bank is responsible for any ATM-related problems that a customer may have. That's what we tell customers when they call us—to contact their issuing bank. That's really all we can do, since we have no authority relative to accounts.

DG. **Do you provide any services directly to consumers?**
Burchfield. Only one—an 800 number that consumers can call to locate a nearby CIRRUS machine. The database of ATM locations is a geographical matrix that's accessed by area code and telephone exchange. The 800 operator asks for this information and then provides the locations of some nearby terminals. This service is subcontracted to an outside firm.

DG. **If I ran a bank and felt that I was at a competitive disadvantage because I didn't belong to an ATM network, and I wanted to join CIRRUS, how would you deal with my request?**
Burchfield. To answer that question, let me start by telling you something about our structure (see Figure 1). We have three types of members—Principal, Associate, and Corresponding members. Principal and Associate members have direct links to the CIRRUS switch and



The CIRRUS network consists of 16 4.8 kbit/second lines that radiate from the CIRRUS switch at the National Bank of Detroit (NBD) to the various nodes. Principal and Associate members are directly connected to the CIRRUS switch at NBD. A Principal member has exclusive marketing rights for a particular area, and an Associate member has direct access, but not exclusive rights. Corresponding members are connected to the switch through Principal or Associate members.

FIGURE 1. Topology of the CIRRUS Network

the right to franchise CIRRUS to other financial institutions. Principals and Associates are primarily large financial institutions that operate regional networks.

These Principal and Associate members can bring other banks into the network as Corresponding members. These are generally smaller institutions for which a direct connection would not make economic sense. Therefore, whenever a small bank approaches CIRRUS directly, I refer it to the Principal or Associate member that offers CIRRUS in the bank's state.

***DG.* What would happen if Bank X joined a regional ATM network like Cash Stream, without actually joining CIRRUS, and tried to send transactions through the CIRRUS switch via the Cash Stream CIRRUS node?**

Burchfield. Well, first of all, Cash Stream could sign up Bank X as a member of their regional network, and therefore as a member of CIRRUS, but if Cash Stream actually did send the CIRRUS switch a transaction that originated at Bank X, without working through a regional network, that transaction would be rejected because Bank X would not be in the financial institution table at the CIRRUS switch. It wouldn't be a valid transaction.

***AS.* Who are your direct competitors?**

Burchfield. There's the Plus network, which is run by a group of banks, and Master Teller, which is operated by Master Card. As of the end of 1984, Plus had 4700 ATMs, Master Teller had 3250, and CIRRUS had 6500. Plus has a competing networks rule that prevents their members from joining another network. Master Teller doesn't have such a rule and neither do we, so some banks belong to both CIRRUS and Master Teller. Master Teller is very big on credit functions, of course, but their card base is limited as far as debit transactions or checking accounts, so there's a certain advantage for customers in being connected to the two networks.

***DG.* So if there were two banks that were connected to both CIRRUS and Master Teller, they would have two different ways of routing transactions between themselves. Does that make transaction switching a competitive business?**

Burchfield. Transaction switching is not competitive right now because the bank that decides which network to use is not the bank that pays for the transaction. The destination bank has to pay for the transaction and would thus like to receive it over the more

economical network; the acquiring bank, on the other hand, wants to issue the transaction in the most convenient way, which is probably the least favorable to the user.

AS. **How do CIRRUS member banks recover the cost of processing transactions for other member banks?**
Burchfield. The issuing bank pays the deploying bank 50 cents for a withdrawal and 25 cents for a balance inquiry. The switch gets an additional 25 cents per transaction. Right now, most CIRRUS member banks offer CIRRUS services for free, but my assumption is that members may eventually pass their costs on to the customer. Customers will then have to compare the cost and convenience of CIRRUS with traveler's checks and other alternatives.

AS. **Do you foresee the establishment of gateways for connecting CIRRUS, Master Teller, and Plus?**
Burchfield. We don't think there's any need for connecting ATM services. We see our independence as a competitive advantage. For point-of-sale services, though, it makes sense to allow merchants to accept different kinds of cards. When point-of-sale machines are connected to more than one network, it becomes logical to standardize fees and message formats. These issues are already being addressed by the American Bankers Association and the Electronic Funds Transfer Association (a multi-industry trade association for electronic funds transfer). CIRRUS, Plus, and Master Teller are already meeting to talk about technical standards, to try to build a common gauge railroad, so to speak, so that point-of-sale transactions won't happen in a fragmented way. We're all very competitive, but we realize that it's not going to work unless there are standards.

Interview 2. Jay Levy, director of systems and operations for CIRRUS Systems, Inc., talks about the design and implementation of the network.

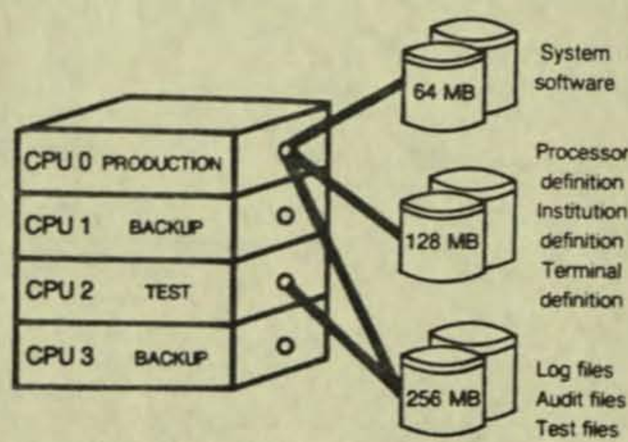
AS. **Jay, we'd like you to take us over some of the technical details of the CIRRUS system. Could you start by giving us an idea of how it's organized?**
Levy. Sure. CIRRUS is a star network, with a message switch at the National Bank of Detroit that currently connects to 16 nodes via 4.8 kbit/second dedicated lines. If a line fails, it's replaced by a dial-up circuit that runs at the same speed. The communications protocols are bisync point to point. The CIRRUS switch is the secondary point on each line, so each node processor has primary access for the bid on its line.

The switch itself is a Tandem NONSTOP II system with four CPU's (See Figure 2). Each CPU has 2.4 megabytes of main memory. The system has a pair of 256-Mbyte drives, a pair of 128-Mbyte drives, and a pair of 64-Mbyte drives. These are shared by all of the CPUs. Most of the switching functions are handled by one Tandem CPU with another acting as backup. When we run tests, we take two of the four CPUs and configure a test system.

JAY LEVY
Jay Levy is director of systems and operations for CIRRUS Systems, Inc. Prior to joining CIRRUS, Levy was an assistant vice-president at the National Bank of Detroit (NBD), in charge of design and development pertaining to NBD's operation of the CIRRUS ATM interchange switch. He was also responsible for developing the ATM program from a proprietary system to a regional network, and assisted in the development of telephone-bill payment services, videotex banking applications, and international retail banking operations.

The 256-Mbyte disks are used for daily transactions—one disk drive for primary and one for backup. The 128-Mbyte disks are used for control files, processor status files, terminal files, institution-level files, and holiday records. They also contain the routing tables and the system definition file, although these are kept in main memory when the system is in operation. The 64-Mbyte disks contain system files.

The switch sits next to another Tandem system at the National Bank of Detroit that could be reconfigured to take over the CIRRUS load if the primary system failed. About the only thing that would cause a real disaster is a power failure, since there is no backup



The CIRRUS switch uses redundant CPUs and disk drives to maintain a high degree of availability and reliability.

FIGURE 2. CIRRUS Switch Configuration

power supply at the switch site. However, NBD does have a backup site that wouldn't be affected by a power failure in Detroit.

DG. How much log data do you accumulate on a day-to-day basis?

Levy. A log record is 411 bytes. There are two log records written for each transaction. One is written into the log file that corresponds to the transaction's anticipated date of settlement. The log file for a Monday would contain between 24,000 and 40,000 records. The switch stops accepting activity for a current date at 8 P.M. Detroit time, which means cutting over to the new log file. Just before cutover, approximately 99 percent of our traffic is already going to the new date of settlement, as specified by the terminal owner.

The other log record is to a backup transaction file that's in time-stamp order. We've been fortunate in that we've never had to use the backup log file.

AS. What do you do with the log that you've accumulated each day?

Levy. The log is the source of the batch reports that we send to each node bank's data-processing organization each day. A node's report lists all of its activity for the day in time-stamp order. The report has separate sections for inbound and outbound traffic. The batch run is also used to verify the on-line settlement totals that indicate the balance of each node with the switch. If the batch run produces totals that are consistent with the on-line totals, settlement is effected by transferring funds between "due from" and "due to" accounts that are maintained by each node bank at NBD. The actual transfers are accomplished by the money-management departments of member banks. We only do settlement on a node-to-node basis—each node must deal separately with its member banks.

AS. What happens when the batch node balances don't agree with the on-line node balances?

Levy. We look for the reason. Typically we'll be out 25 or 50 cents for a missed transaction fee somewhere. We've never had a serious discrepancy in a consistency check.

AS. What is the peak transaction load on the switch?

Levy. The switch was designed to handle 2 transactions per second, which is 16 messages a second. I don't think we've ever achieved that peak—we have never exceeded 500,000 transactions a month. The total elapsed time for a transaction averages between 12 and 15 seconds from request to cash dispense. This includes the processing time at a node bank for approval and debit.

AS. Suppose I had an account at BayBank and that I went to an ATM run by Landmark Savings and Loan in Pittsburgh, which happens to be a member of the Cash Stream network. Mellon runs Cash Stream and is a node on the CIRRUS network. What happens when I put my BayBank card in the Landmark ATM and request $100?

Levy. You would get a *customer lead through* on the screen that would look exactly like the lead through for regular Landmark customers. The ATM wouldn't know who you were yet, and so its processor would ask you for your personal identification number (which it would temporarily buffer), what type of account you wanted to access (i.e., savings or checking), and the dollar amount of your transaction.

By this time the Landmark ATM would have already read Track II of the magnetic strip on the back of your BayBank card, and determined your account number, expiration date, and some other variable data that are dependent on BayBank. The first 3 to 11 digits of an account number specify the issuing bank. Track II holds 40 bytes, although we're only using 19 of them right now. Tracks I and III we don't use at all—Track I is for the airline industry, and Track III is used in off-line banking transactions, but not in the interchange environment.

To get back to your transaction, though, the host processor at Landmark has not been involved at all up to this point. Once the ATM has collected all of the data it needs for your $100 withdrawal, it sends a message to its host at Landmark, which determines that the account number on your BayBank card is not a Landmark account, and creates a message to Cash Stream. The Landmark host also writes a suspense record to its log so that it can time out the transaction and free the ATM if Cash Stream does not reply after a certain amount of time. The Landmark host would probably give Cash Stream about 34 seconds before doing this. Most ATMs also have internal timers—we suggest that ATMs abort transactions and return consumers' cards if they do not hear from their hosts in 45 seconds.

When your transaction arrives at the Cash Stream switch, Cash Stream validates it to make sure that it has a valid business date, dollar amount, and so forth. Cash Stream would then look up your bank number in a routing table, determine that BayBank is not a Cash Stream member bank, and then create a message to CIRRUS in a variation of ANSI X9.2 format. The message would contain a lot of information, including a 40-character description of the ATM, to comply with government regulations. Once the message is created, Cash Stream logs it and sends it to the CIRRUS switch via a bisynch point-to-point protocol. Cash Stream waits up to 24 seconds for a response before aborting the transaction.

CIRRUS receives the message, looks up the issuing bank number in its routing table, and then forwards the message to the CIRRUS node that the issuing bank is connected to, which in this case is BayBank. The CIRRUS switch sets a timer for 20 seconds and then waits for the response from BayBank. BayBank receives the message, verifies that the personal identification number (PIN) you supplied is correct, and then checks your balance. If you check out, BayBank writes a memo

post record on your account indicating that a $100 withdrawal is in progress, and then sends an approval message back through CIRRUS, Mellon, and Landmark to the ATM, which dispenses your $100.

The memo post record that BayBank has written is simply noting that a withdrawal is in progress. It will *not* cause $100 to be deducted from your account. That happens with the second round of messages, which I haven't yet discussed.

When the ATM has actually dispensed cash into your hands, it sends a message to Landmark saying that it has given you $100. The completion message goes from the ATM through Landmark and Mellon to CIRRUS, and then CIRRUS sends a message back to Landmark acknowledging the completion message. CIRRUS also sends the completion message on to BayBank, which uses it to create a "postable" record that is used to deduct the money from your account. A completion acknowledgment then goes back from BayBank to CIRRUS. Altogether, your transaction has required four CIRRUS messages—two for the initial approval and two for the completion.

If CIRRUS does not receive a completion acknowledgment from BayBank after 120 seconds, the switch continues sending until it does receive one. If for some reason BayBank cannot deal with the completion message, they call us and we purge it from our file.

Approximately half of our member banks use this protocol. The other half use a shorter protocol that eliminates the second two messages. In our example, this would mean that the CIRRUS switch would still get the completion message from the ATM, but that it would not forward it to BayBank. When this shorter protocol is used, the issuing bank must simply assume that a withdrawal has actually taken place. Obviously, if there's a problem then a message must be sent to the issuing bank instructing it to compensate for the debit it posted.

***AS.* What would happen if BayBank had sent an approval to the Landmark ATM and I had received the $100, but the completion message had not returned from the ATM to BayBank because the Mellon Cash Stream system had failed?**

Levy. If the CIRRUS switch does not get a completion message from Mellon within 120 seconds, then Mellon would be marked down by the switch and would be placed off line for 10 seconds. CIRRUS would send a message to BayBank saying that the cash had not been dispensed. After the 10 seconds had elapsed, we would bring Mellon on line by issuing a series of network-management messages. If the completion message from Mellon came in after the 120-second time had expired, but within the same settlement day, the CIRRUS switch would interrogate the log file to see if there was a record for that message. If there was, the switch would automatically build adjusting entries to both sides of the transaction to ensure that the settlement was handled properly. If the late completion message came in after the settlement cutover for the day, or never came in at all, Landmark, as the ATM operator, would have to present an off-line paper adjustment to the switch, along with some kind of physical evidence of the transaction, like an audit tape or an internally produced report indicating that the transaction had occurred, and a CIRRUS report indicating that they hadn't received value for the transaction. Without that documentation they wouldn't get their money. This kind of thing is not uncommon—some ATMs don't automatically send completion messages to the host. The only way to discover what's happened is to get a status of prior transactions after a new transaction starts.

***DG.* Could you review the way the time-out intervals on your system interact?**

Levy. We have a 20-second timer at the switch for messages to nodes; we advise sending nodes to set a 24-second timer to allow some transmission time. A bank that's not directly connected to the switch, like Landmark in your previous example, may wait as long as 34 seconds to allow the intermediary bank some processing time. We typically tell the ATM issuers to set their timers for about 45 seconds; if the terminal doesn't get a response within that time, it shuts down and returns the customer's card.

***DG.* The two message protocols you described—the four-message protocol and the two-message protocol—make radically different assumptions about when to deduct money from an account, don't they?**

Levy. Traditionally, the U.S. banking industry won't ever deduct anything immediately. We have very few real-time posting, checking, or savings applications in the U.S. That convention doesn't always hold true for other countries, though. In Canada, for instance, there are systems that use real-time posting, which means that a single customer transaction can post several items on a cardholder's account. In the U.S., banks use on-line systems for maintaining account balances. Posted records from a system are merged with other instrument activity (such as checks) during batch processing. The daily batch run is what actually deducts the money from accounts. The result of a daily batch run is a file containing the starting account balances for the system for the next day, as well as records for pending transactions.

***DG.* From what you've been saying, there seems to be little concern that any of the participating banks might be doing anything dishonest. What would happen if a member bank immediately acknowledged that an ATM had dispensed cash when in fact it hadn't. Wouldn't that bank be credited for $100 while the customer was left standing in the rain without any money in his hand?**

Levy. Customers would let us know if there was anything of that nature going on. Also, there's a significant amount of federal regulation that protects customers in

those situations, so it's hardly in the member bank's best interest to do such a thing.

***AS.* But the safety and security of your network depend very much on the safety and security of its constituent parts.**

Levy. Absolutely. CIRRUS was developed as a pipeline between banks that had been doing this kind of thing for 10 or 12 years before we came along. We're another link of potential authorization for the banks. All of the policies and procedures that banks had evolved to protect themselves from error and fraud were carried over directly to the CIRRUS environment.

***DG.* You mentioned before that you partition any part of your network that doesn't respond within a certain amount of time. Is that an important part of your system strategy?**

Levy. Yes. I'm probably one of the few proponents of this kind of partitioning. The more popular strategy is to keep resending messages until a response is received. I believe that the financial implications of transaction processing in the ATM environment require this kind of scrupulousness. It has proved easier for us to balance the network by guaranteeing that transactions flow out of the store-and-forward process in a logical sequence. We've also found that if you give a node time to recover you can often prevent more serious problems.

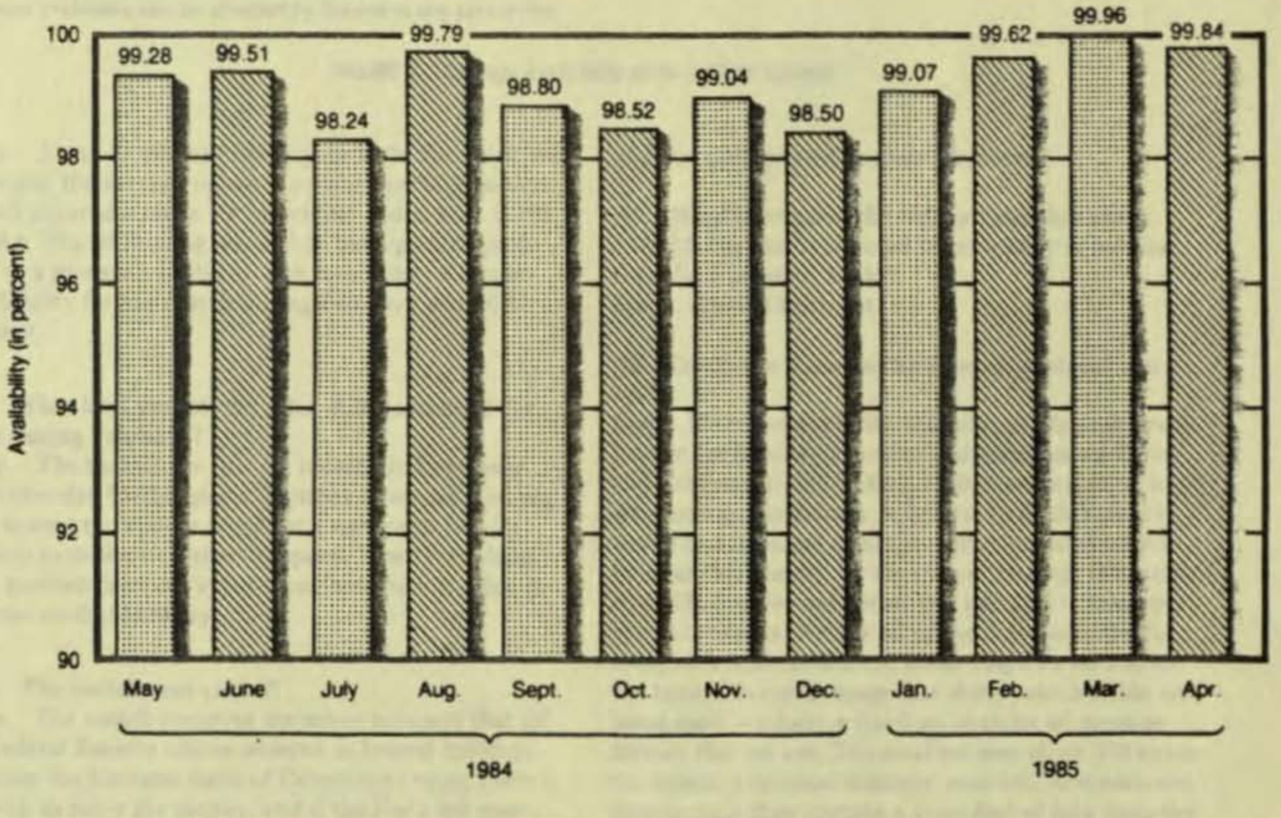
***DG.* How often do banks go off line?**

Levy. We're working to get that rate down to a minimum. We have a standard that no bank should be off line more than five percent of the time, averaged over a month. Some of our members are well above that mark, and some are well below it, but most are off line no more than five percent of the time. Nearly all of our member banks use highly available systems, but as I mentioned earlier, if a node does not get a completion message in time from an ATM, CIRRUS may take the entire node off line. Of course in the case of a delayed completion, a message goes out 10 seconds after a node goes off line to bring it back up again. These short ups and downs have a minimal effect on transaction processing.

***DG.* CIRRUS itself could also go down, couldn't it?**

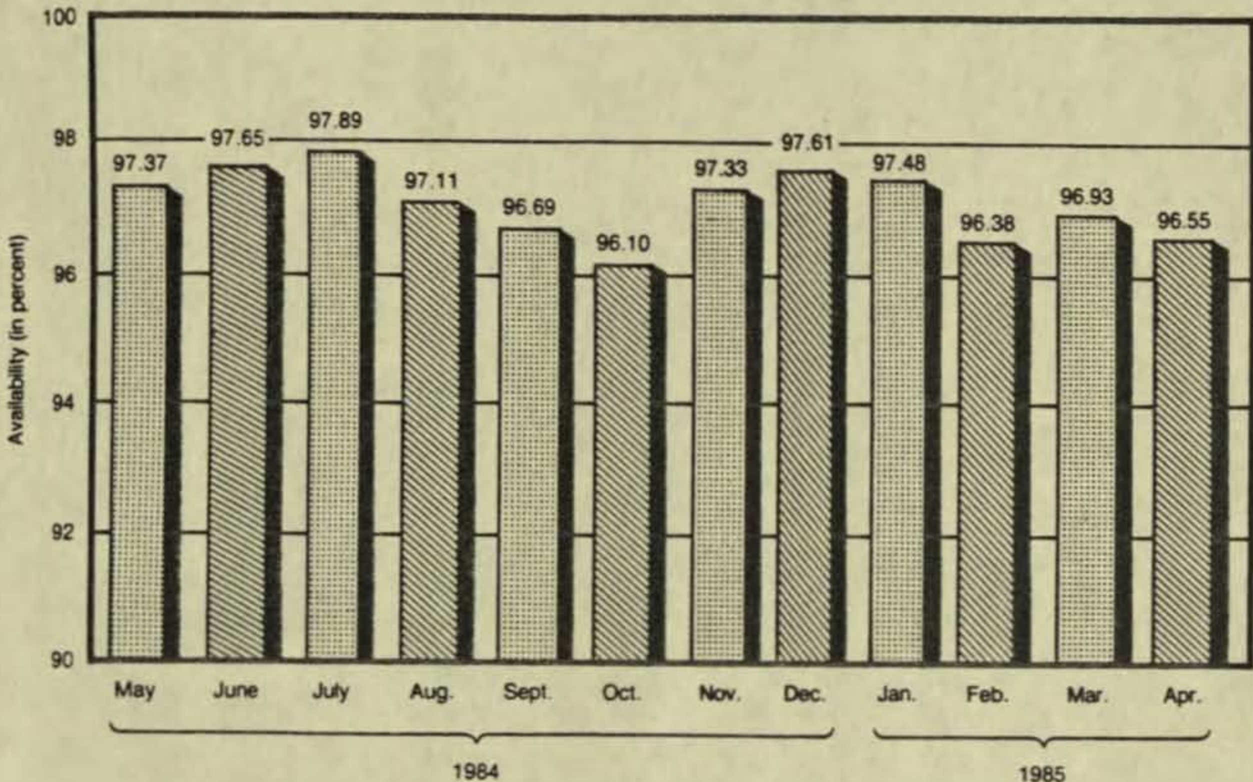
Levy. It's not likely. The average availability of the CIRRUS switch exceeds 98 percent on any given day (see Figure 3). We do not allow scheduled outages of the CIRRUS switch. Actually, we allow one a year, but it still goes against that 98 percent. Tandem equipment does not require outages for hardware maintenance.

***DG.* Considering communication-line failures, central-site failures, and node failures, what's the probability that one node will be able to access another at any given time?**



The average uptime on the CIRRUS switch is between 98 and 100 percent over a 24-hour period. Scheduled downtime is counted as unavailable time.

FIGURE 3. Average Availability on the CIRRUS Switch



Network availability is computed as the probability that one CIRRUS node can access another at any given time. Network availability can be affected by failures in any part of the network, from the switch to the communication lines to the nodes.

FIGURE 4. Average Availability on the CIRRUS Network

Levy. About 97 percent (see Figure 4). In February, for example, the average network uptime was 96.3 percent, which covered a range for individual nodes from 90.13 to 99.6. The 90.13 node was either having a line problem or a processor problem. The mean daily average availability for the switch during February was 99.62 percent.

***AS.* What kept you off the other 0.38 percent of the time during February?**
Levy. The backup file just got incredibly large over the three-day Washington's Birthday weekend, and we had to shut the system down for a minute or two (or fifteen) to reallocate some file space. There was also a date problem because some of our members wanted to process on that Monday.

***DG.* The switch was closed?**
Levy. The switch observes the seven holidays that all 12 Federal Reserve offices observe as federal holidays. Because the National Bank of Detroit isn't open, there's no way to move the money, and if the Fed's not open, there's no way to fund an account at NBD. Certain banks that close for Lincoln's Birthday stay open for Washington's, though, and they were trying to present transactions when there wasn't a valid log file open, and that probably led to some downtime.

***AS.* What fraction of the transactions that reach CIRRUS have to be rejected because they're not destined for a member bank?**
Levy. About 15 percent.

***AS.* Could you describe the message protocol you use?**
Levy. It's called the ANSI Interchange Message Specification for Debit and Credit Card Messages on Financial Institutions—ANSI X9.2-1980. It's a guideline, as all standards are guidelines. It defines 64 fields that are usable in a financial message, and it indicates which fields are mandatory for the type of message being sent. At CIRRUS we've enhanced this standard to meet our particular needs. We use all of the mandatory fields, along with some additional ones. There's a bitmap at the head of every message that shows which fields are being used. We have a fixed set of about 40 message formats that we use. The smallest uses about 170 bytes; the largest, a reversal message, over 400. Reversals are long because they contain a great deal of data from the original transaction. For each type of message, we have both a long and a short format. The long format has all of the fields filled in, and the short one requires that missing fields be filled in by the switch.

AS. Could you briefly summarize the types of messages that you use?
Levy. There are request, reply, completion, confirmation, adjustment, reconciliation, and network-management messages. For each type of financial transaction, there's an entire set of request, reply, completion, and confirmation messages.

DG. Are point-of-sale transactions handled separately?
Levy. Point-of-sale transactions, once they're developed, will use the same messages with various fields indicating a POS transaction. For POS we've created a new type of transaction that is a preauthorized request. This would be used at places like gas stations where intended purchases don't always match up to actual purchases (because who knows exactly how much gas their tank can hold). The preauthorization might be for $20; the customer would pump $18.50 worth of gas, and the POS terminal would then send subsequent messages out to build the posting record for $18.50. The memo post, though, would be held for $20. We want to use a different set of messages for preauthorization because we don't want anyone to confuse them with real purchases.

AS. Is preauthorization similar to the failure condition that occurs when an ATM can only distribute part of a requested sum?
Levy. Yes, except here the discrepancy is the norm instead of the exception.

DG. What are your message latency goals at the switch, and how can you maintain them as your volume increases?
Levy. We don't allow the switch any more than 2 seconds for processing a message. The actual time is usually somewhere between 0.7 and 1 second. The switch is modular enough that if we ever needed to we could open up a mirror of the application so that there could be multiple processors running the application simultaneously, although there would only be one transaction running in any processor at any time. I think we've actually done this at peak processing time.

AS. Do you spool log data onto magnetic tape for permanent safekeeping?
Levy. All the important files, including the log files, are backed up every night. Once a week we back up the entire system, including the application program, and transport it to an off-site storage location. We also put a digested form of the log on microfiche and keep it for five to seven years, to meet regulatory requirements.

DG. How would you compare your electronic funds transfer network to the Fed-wire?
Levy. Well, the Fed-wire is a wholesale-level EFT network, and CIRRUS is a retail-level EFT network. The average dollar value per transaction on the Fed-wire is probably a million times what we handle on CIRRUS. The aggregate for a whole day of CIRRUS activity would barely warrant a wire on the Fed-wire system. That's part of the reason that we aren't nearly as security conscious as the Fed-wire is—our exposures are not nearly as great. Also, the Fed-wire has much more significant and stringent requirements with regard to who can initiate what type of transaction and how transactions can be effected.

AS. Who writes the programs for CIRRUS?
Levy. We contract the system services from the National Bank of Detroit, which maintains a programming staff to support the switch. NBD subcontracted the writing of the switch to a software house, but did all the design and testing on its own, and NBD owns the code. The software is relatively stable now. I would be surprised if NBD had more than the equivalent of one full-time person budgeted for maintaining the CIRRUS software. When a new release of applications is ready to be released into the system, it has to be certified by the system software group at NBD.

DG. Are new applications like point of sale contracted out to a software house?
Levy. As a matter of fact, we're moving our processing to Mellon Bank because they were the lowest bidder for the POS product. Mellon is going to have to redevelop the switch software.

DG. How many lines of code are in your on-line switch application?
Levy. The on-line switch software, which includes message routing and editing, is approximately 9000 lines of TAL, which is the Tandem application language.

DG. How big is your routing table?
Levy. Currently it consists of about 3500 records, with about 20 bytes per record. A routing-table entry includes a bank's prefix, which is from 3 to 11 digits, and an identification of the processor that services the bank. The routing table is called the FIT, for *financial-institution table*. We send an up-to-date copy of the FIT to our member banks every day. It's our intention to build an on-line message that keeps FIT copies up-to-date. The problem is that bringing a new member on can mean bringing as many as 100 new banks on with it. A hundred new FIT entries at one time is a data-management problem that we haven't solved yet.

AS. Do you test a member bank's system before letting it on the CIRRUS network?
Levy. Yes, we certify nodes with a battery of test cases. Every situation we can think of is attempted prior to putting a node on the network. The book that we keep the test cases in is over three inches thick—I believe it contains over 2500 test transactions. When we certify a node, we ask the candidates to have an ATM available in a test environment. We send them plastic cards that run against a simulated card proces-

sor at the switch in Detroit. The switch can also send simulated ATM activity out to them. We find that using an actual ATM is the best way to test.

AS. **How is a typical test case described?**
Levy. In great detail. This is a *very* critical part of our operation. I think part of the success of the network, in fact, can be attributed to the stringency of our testing. We send candidate banks a list of cards and a set of conditions that the cards have to meet. Some of these cards specify accounts that don't exist or that have incorrect balances. A test case instructs the candidate bank to use a particular card for a particular transaction, and then indicates the expected result. We test weekend and holiday processing and all the exception conditions. We also keep our set of test cases up-to-date as we add new functions to the switch.

AS. **What percent of your budget eventually goes for testing?**
Levy. I would say that as a percentage of overall development, testing is easily 25 percent. In an environment like ours, we have to be very careful.

AS. **Have you had any situations where a member bank did not perform properly?**
Levy. Yes, we've had member banks that stopped sending completion messages for one reason or another. On one occasion we had 1500 adjustments for one day's activity. These had to be presented on paper. This swamped their operation center as well as ours. We have also had requests for adjustments that came in as much as 50 business days after a transaction. It's difficult to accept activity that late.

DG. **Could you discuss your basic strategy for security?**
Levy. Within the banking environment, there's a great deal of physical security in addition to application-level security. Most any data that are maintained in a bank's data-processing center are considered confidential and secure. The only part of the system that we don't normally have physical control over is our communication lines, which are protected with link encryption provided by the Racal-Milgo DataCrypter II with a public key option. This is a commercial product that uses DES for encrypting messages. It includes a public key algorithm that allows us to distribute new master keys from time to time.

AS. **Have your communication lines ever been actually attacked?**
Levy. Not that I know. The biggest area for fraud in ATM banking is card and PIN compromise: Either a family member takes a card and cleans out an account, or someone obtains a card and its PIN by force. It's as easy to defraud an ATM that way as to tap the line.

AS. **Are you interested in "smart cards"?**
Levy. They would allow us to put a great deal more data on plastic and would let us do away with PIN verification at the issuing bank. They would also allow us to maintain value on cards, so that customers would be able to put, say, a thousand dollars worth of value on a card that could then be spent without on-line verification. The thing that concerns me is how could we get the postable information to the cardholder's bank and actually deplete his account balance. I see the biggest application of chip cards in this country for things like food stamps where the value is depleted but the transactions don't have to be posted. It's not clear what application smart-card technology will have in banking, which may be why it's taking so long to be developed here. There's also the problem of refitting hundreds of thousands of terminals, and there would have to be a standard for where to put the contacts on the cards.

DG. **Have you considered digital signature verification for point-of-sale transactions?**
Levy. I have my doubts. Handwriting can be compromised. And it's hard to say what kinds of controls we're going to need until we really find out how easy it's going to be to deploy POS. I'm not convinced that signature technology is that much more secure than PINs.

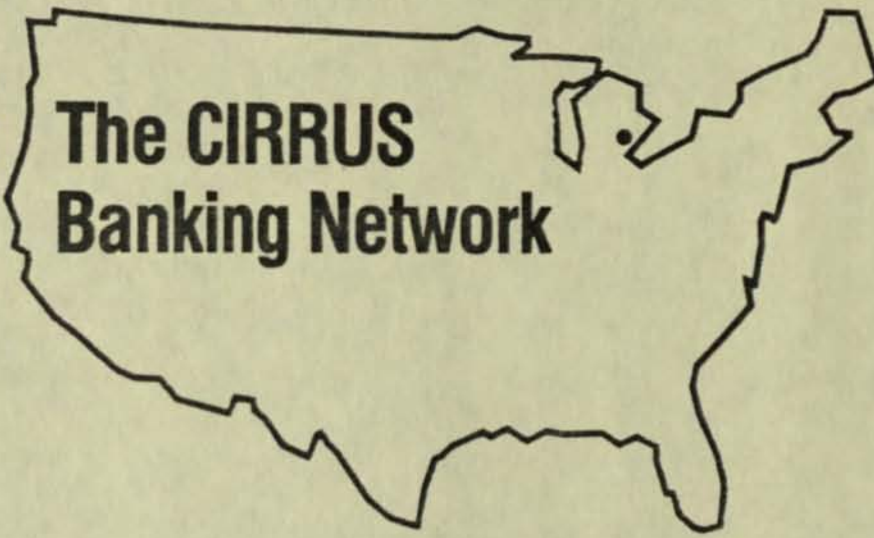
DG. **What's the largest transaction that can be issued over the CIRRUS network for a cash withdrawal?**
Levy. Theoretically, the field size would allow for a $100,000 transaction, but I don't know of a machine in the country that would let you do that. There are machines in Las Vegas that would probably let you take $1000 out in one shot.

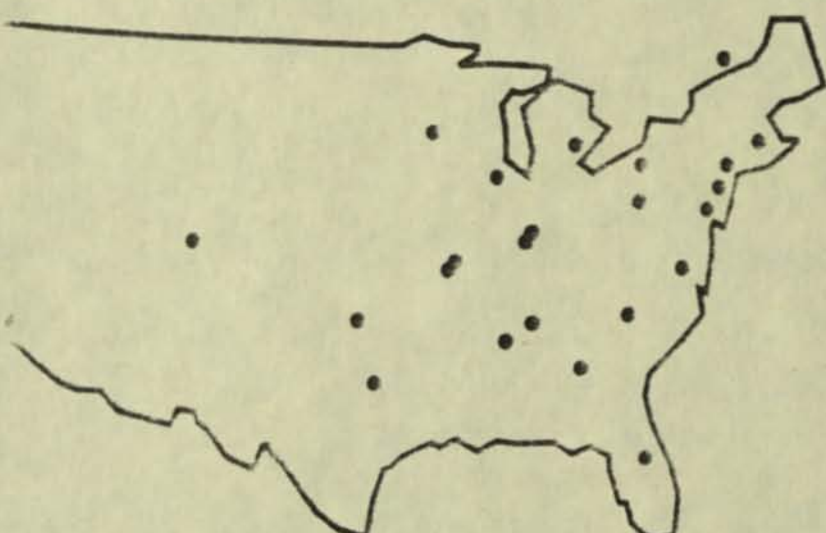
AS. **Will future ATMs have change-making capabilities?**
Levy. Yes, there's a machine that makes change, and five-bill dispensers that can give ones, fives, tens, twenties, and fifties or hundreds. The more these machines can do, the more truly they'll be able to replace human tellers, which is really valuable in some applications. There may not be a need for change-making capability at O'Hare Airport, but if someone's going to an ATM regularly for their everyday banking business, it should be able to cash their paycheck. Not everyone's paid in increments of $10.

DG. **I think that's all the questions we have for you. Bruce, Jay, thanks for telling us about the CIRRUS system.**

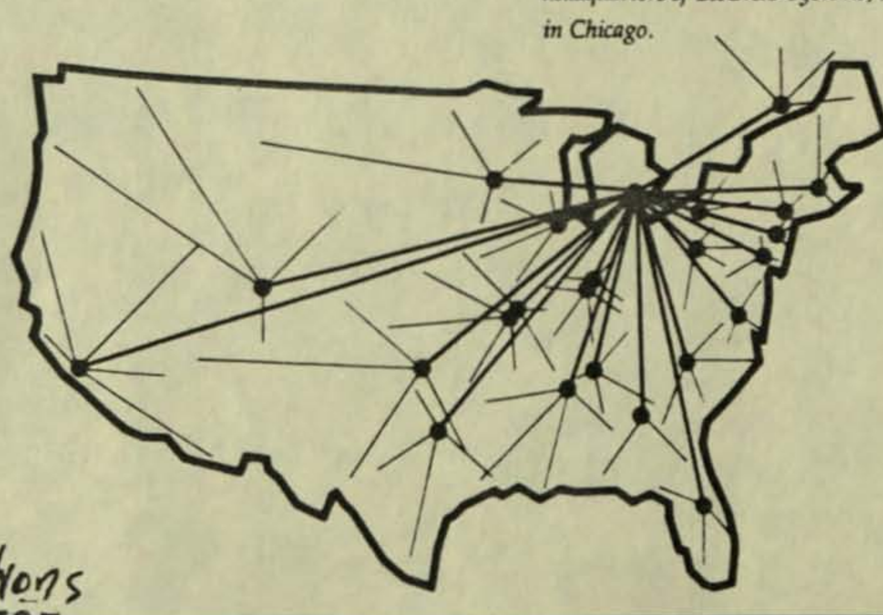
CR Categories and Subject Descriptors: C.4 [**Performance of Systems**]: *reliability, availability, and serviceability*; D.2.9 [**Software Engineering**]: Management—*software quality assurance*; D.4.0 [**Operating Systems**]: General; J.1 [**Administrative Data Processing**]: *financial*; K.6.3 [**Management of Computing and Information Systems**]: Software Management
General Terms: Management, Reliability, Verification
Additional Key Words and Phrases: ATM, CIRRUS, EFT systems

Case Study: The CIRRUS Banking Network





This is the third case study to appear in Communications. *This time, case-study editors David Gifford and Alfred Spector interviewed the president and the director of systems and operations at CIRRUS Systems, Inc., a funds transfer network, to see how distributed systems are being employed in the banking industry. As the following transcript shows, the CIRRUS network provides security, lost message recovery, database consistency via atomic actions, recovery after network partition, and a framework for interorganization cooperation in a distributed system. The CIRRUS system is not the most complex computer system to have been described in these case studies; rather, it is an example of how a few well-chosen ideas can be the basis of a simple and useful system. The interviews were conducted on March 29, 1985, at the headquarters of CIRRUS Systems, Inc., in Chicago.*



Aug 1985 p797,
ACM Communications

ARTICLES

THE CIRRUS BANKING NETWORK

The CIRRUS banking network makes coast-to-coast automatic banking transactions possible. The system will soon be able to handle international currency transactions and point-of-sale transactions in stores.

DAVID GIFFORD and ALFRED SPECTOR

Interview 1. Bruce Burchfield, president of CIRRUS Systems, Inc., talks about the development of the CIRRUS network, its present scope, and plans for its future.

DG. **Bruce, could you start by telling us something about the history of the CIRRUS network?**

Burchfield. I'd be glad to. Bankers have traditionally provided customers with personalized service. That's the nature of the business. By the 1970s, most banks had automated their "back-room" operations, but were still using people for all their customer interaction. During the 1970s, though, both technology and customer demographics changed. On the one hand, automated teller machines (ATMs) were becoming reliable and cost effective, and on the other, customer demographics were changing and banks were under pressure to provide more convenient access to their services.

Bankers couldn't really afford to provide essential full-branch service for 16 or 18 hours a day, so they installed ATMs to keep certain essential services available around the clock. The next step was to situate these ATMs away from existing bank branches. The choice for bankers was either to build $1 million branches that would cost a half a million a year to operate, or to install 24-hour ATMs for about a $100 thousand each and then $50 thousand a year for maintenance. From the banker's perspective, ATMs are a very cost-effective way of making their services available in more places and at more times.

Then, around 1976, bankers realized they could reduce their costs even further by sharing ATMs. Since there's usually a certain amount of excess capacity on an ATM that's only serving one bank's customers, some banks began to sell their unused capacity by sharing their resources with client banks and then charging a set fee for every transaction processed. Reciprocal sharing arrangements also provide participating banks with better distribution channels, which can mean a competitive advantage.

These developments have allowed banks to reduce staffing and cut back on hours in their regular branches, and to deal with competitive threats from companies like Sears and American Express that are entering the financial-services business with a cost structure that would drive unmodernized banks out of business. It's a traditional paradigm—when the structure of an industry changes, the old leaders are displaced if they can't keep pace.

DG. **So by the late 1970s you knew not only that ATMs were going to be important, but that ATM sharing was also going to be important. How did the idea of CIRRUS begin to take shape?**

Burchfield. Banks had already formed local and regional sharing arrangements. For example, I had developed a shared automated-teller-machine network in metropolitan Chicago called Cash Station. The next logical step was to provide access for customers regardless of where in the country they happened to be.

CIRRUS was started by 10 banks that wanted to expand their regional networks into a national network. These included BayBank of Boston, Manufacturers Hanover of New York, Mellon Bank of Pittsburgh, and First Chicago. The first discussions were held in December of 1981, and by July of 1982 CIRRUS was incorporated. The initial service that CIRRUS set out to provide was cash access for people away from home.

We knew we needed a certain critical mass to make the network successful, and so we worked to build the network up quickly. As of March 1985, we have 46 states covered. In building this network, we've created a technical infrastructure that's unprecedented in the banking industry. The number of member banks has

now grown to 1425, although only 16 of them are directly connected to the CIRRUS switch. For example, there are over 300 banks in Texas on CIRRUS, but they're all connected to a single node. The Texas regional network handles all intrastate traffic and forwards the remainder to CIRRUS via the Texas CIRRUS node. There are probably 150 million to 200 million transactions a year that are processed by CIRRUS member banks, and yet only a fraction of that goes through the CIRRUS switch. Thus, in many cases, CIRRUS is a brand name as opposed to a processing function. Our node-based architecture has worked out well, and we don't think the politics or the structure of the ATM market will ever be conducive to connecting individual banks directly.

DG. **What other services do you now support besides cash access?**
Burchfield. We support withdrawals and balance inquiry from checking, savings, and line-of-credit accounts, and we are also developing direct debit point-of-sale services that will allow us to put machines in retail outlets. This summer we're connecting to a Canadian ATM network, and we're planning to do automatic currency conversion for international transactions.

We're also considering the possibility of taking deposits over CIRRUS. Interbank deposit taking already exists on a regional level—on the Chicago network I was involved with, the bank that received a deposit would prove and verify the deposit envelope, process the checks, and return bounced checks to the depositor's bank. The actual transfer of funds between banks is handled electronically.

DG. **When CIRRUS first started, did you analyze all the different exposures of the system and try to put together a study that could convince all of your participating banks that CIRRUS was going to be secure?**
Burchfield. To resolve the security issue, we assigned liability for all of the things that could possibly go wrong. For example, all of our connection nodes have financial and data security responsibility for their traffic. Extensive logs are kept for audit purposes to help assign liability. This gives each CIRRUS node an active interest in running a secure system. The employees of each bank are bonded, and the switch is operated by a bank with extremely strong audit functions. We also have provisions in our contracts that prohibit banks from using transaction data for competitive analysis.

AS. **With all this delegation, it sounds as though the CIRRUS organization itself doesn't have to worry about liability.**
Burchfield. That's correct. For example, the recent failure of certain savings and loan organizations in Ohio didn't cause us any concern because their CIRRUS attachment node, Central Trust in Cincinnati, is liable for their activity. If the savings and loans went defunct, CIRRUS would not be out of any money because Central Trust is responsible for their transactions.

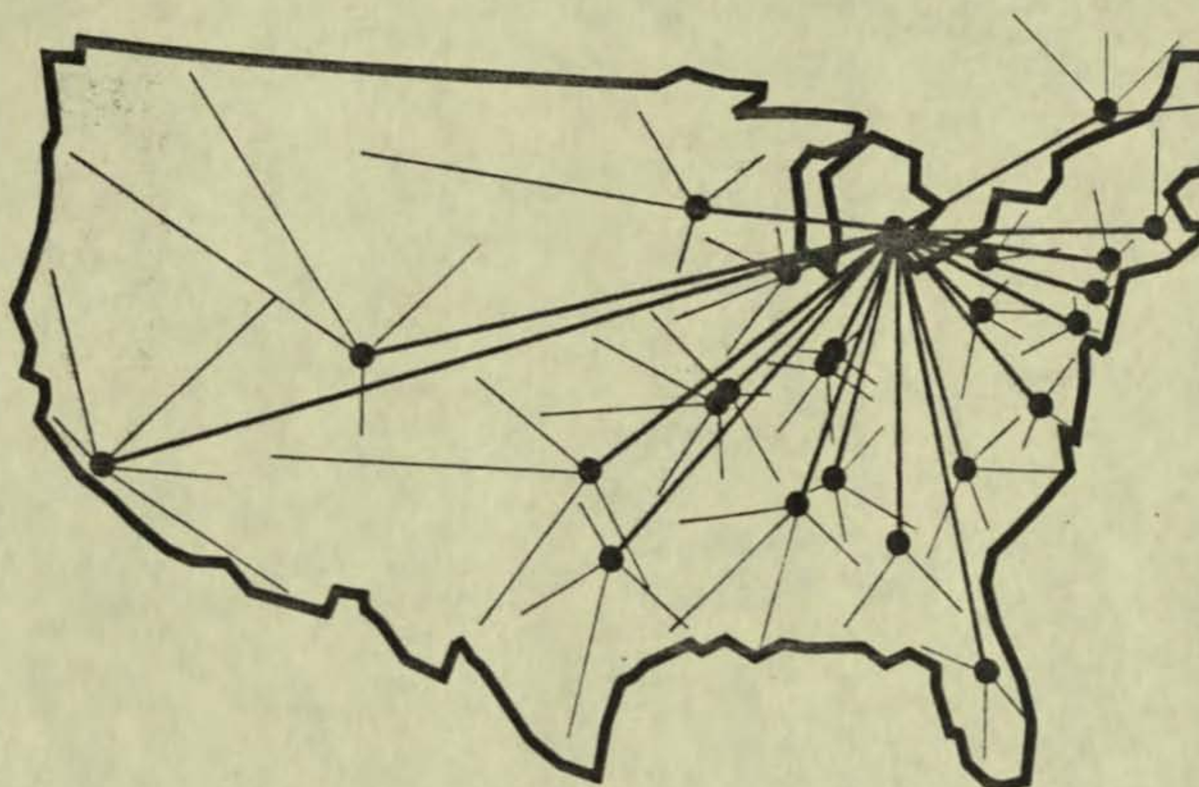
BRUCE BURCHFIELD
Bruce Burchfield is president and chief operating officer of CIRRUS Systems, Inc. He was formerly vice-president and manager of the electronic-banking service division at the First National Bank of Chicago, where he was responsible for the development and management of new electronic-banking services, including electronic funds transfer equipment and First Chicago's shared ATM network, Cash Station. Before joining First Chicago, Burchfield was an engineer for Reynolds Aluminum.

Central Trust actually shut their CIRRUS access off as soon as there was a problem.

DG. **How are consumers protected against the misuse of their cards?**
Burchfield. They're protected by Federal Regulation E, which says that a customer is not liable for the use of lost or stolen cards as long as the loss is reported within 2 days. When a lost card is reported within 60 days, but after 2 days—that is, within the amount of time it takes to get a bank statement—liability is limited to $50. The issuing bank is responsible for any ATM-related problems that a customer may have. That's what we tell customers when they call us—to contact their issuing bank. That's really all we can do, since we have no authority relative to accounts.

DG. **Do you provide any services directly to consumers?**
Burchfield. Only one—an 800 number that consumers can call to locate a nearby CIRRUS machine. The database of ATM locations is a geographical matrix that's accessed by area code and telephone exchange. The 800 operator asks for this information and then provides the locations of some nearby terminals. This service is subcontracted to an outside firm.

DG. **If I ran a bank and felt that I was at a competitive disadvantage because I didn't belong to an ATM network, and I wanted to join CIRRUS, how would you deal with my request?**
Burchfield. To answer that question, let me start by telling you something about our structure (see Figure 1). We have three types of members—Principal, Associate, and Corresponding members. Principal and Associate members have direct links to the CIRRUS switch and



The CIRRUS network consists of 16 4.8 kbit/second lines that radiate from the CIRRUS switch at the National Bank of Detroit (NBD) to the various nodes. Principal and Associate members are directly connected to the CIRRUS switch at NBD. A Principal member has exclusive marketing rights for a particular area, and an Associate member has direct access, but not exclusive rights. Corresponding members are connected to the switch through Principal or Associate members.

FIGURE 1. Topology of the CIRRUS Network

the right to franchise CIRRUS to other financial institutions. Principals and Associates are primarily large financial institutions that operate regional networks.

These Principal and Associate members can bring other banks into the network as Corresponding members. These are generally smaller institutions for which a direct connection would not make economic sense. Therefore, whenever a small bank approaches CIRRUS directly, I refer it to the Principal or Associate member that offers CIRRUS in the bank's state.

***DG.* What would happen if Bank X joined a regional ATM network like Cash Stream, without actually joining CIRRUS, and tried to send transactions through the CIRRUS switch via the Cash Stream CIRRUS node?**
Burchfield. Well, first of all, Cash Stream could sign up Bank X as a member of their regional network, and therefore as a member of CIRRUS, but if Cash Stream actually did send the CIRRUS switch a transaction that originated at Bank X, without working through a regional network, that transaction would be rejected because Bank X would not be in the financial institution table at the CIRRUS switch. It wouldn't be a valid transaction.

***AS.* Who are your direct competitors?**
Burchfield. There's the Plus network, which is run by a group of banks, and Master Teller, which is operated by Master Card. As of the end of 1984, Plus had 4700 ATMs, Master Teller had 3250, and CIRRUS had 6500. Plus has a competing networks rule that prevents their members from joining another network. Master Teller doesn't have such a rule and neither do we, so some banks belong to both CIRRUS and Master Teller. Master Teller is very big on credit functions, of course, but their card base is limited as far as debit transactions or checking accounts, so there's a certain advantage for customers in being connected to the two networks.

***DG.* So if there were two banks that were connected to both CIRRUS and Master Teller, they would have two different ways of routing transactions between themselves. Does that make transaction switching a competitive business?**
Burchfield. Transaction switching is not competitive right now because the bank that decides which network to use is not the bank that pays for the transaction. The destination bank has to pay for the transaction and would thus like to receive it over the more

economical network; the acquiring bank, on the other hand, wants to issue the transaction in the most convenient way, which is probably the least favorable to the user.

AS. **How do CIRRUS member banks recover the cost of processing transactions for other member banks?**
Burchfield. The issuing bank pays the deploying bank 50 cents for a withdrawal and 25 cents for a balance inquiry. The switch gets an additional 25 cents per transaction. Right now, most CIRRUS member banks offer CIRRUS services for free, but my assumption is that members may eventually pass their costs on to the customer. Customers will then have to compare the cost and convenience of CIRRUS with traveler's checks and other alternatives.

AS. **Do you foresee the establishment of gateways for connecting CIRRUS, Master Teller, and Plus?**
Burchfield. We don't think there's any need for connecting ATM services. We see our independence as a competitive advantage. For point-of-sale services, though, it makes sense to allow merchants to accept different kinds of cards. When point-of-sale machines are connected to more than one network, it becomes logical to standardize fees and message formats. These issues are already being addressed by the American Bankers Association and the Electronic Funds Transfer Association (a multi-industry trade association for electronic funds transfer). CIRRUS, Plus, and Master Teller are already meeting to talk about technical standards, to try to build a common gauge railroad, so to speak, so that point-of-sale transactions won't happen in a fragmented way. We're all very competitive, but we realize that it's not going to work unless there are standards.

Interview 2 Jay Levy, director of systems and operations for CIRRUS Systems, Inc., talks about the design and implementation of the network.

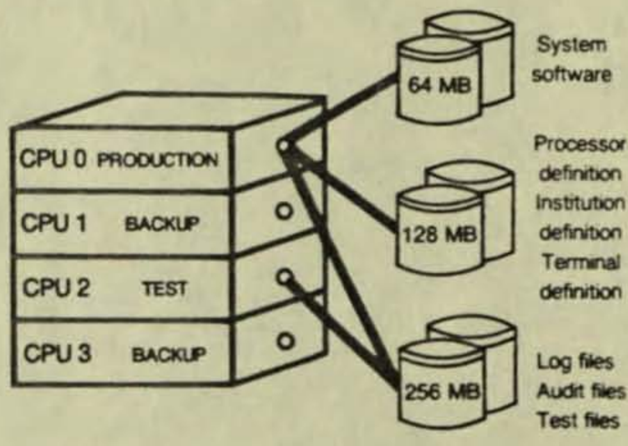
JAY LEVY
Jay Levy is director of systems and operations for CIRRUS Systems, Inc. Prior to joining CIRRUS, Levy was an assistant vice-president at the National Bank of Detroit (NBD), in charge of design and development pertaining to NBD's operation of the CIRRUS ATM interchange switch. He was also responsible for developing the ATM program from a proprietary system to a regional network, and assisted in the development of telephone-bill payment services, videotex banking applications, and international retail banking operations.

AS. **Jay, we'd like you to take us over some of the technical details of the CIRRUS system. Could you start by giving us an idea of how it's organized?**
Levy. Sure. CIRRUS is a star network, with a message switch at the National Bank of Detroit that currently connects to 16 nodes via 4.8 kbit/second dedicated lines. If a line fails, it's replaced by a dial-up circuit that runs at the same speed. The communications protocols are bisync point to point. The CIRRUS switch is the secondary point on each line, so each node processor has primary access for the bid on its line.

* The switch itself is a Tandem NONSTOP II system with four CPUs (See Figure 2). Each CPU has 2.4 megabytes of main memory. The system has a pair of 256-Mbyte drives, a pair of 128-Mbyte drives, and a pair of 64-Mbyte drives. These are shared by all of the CPUs. Most of the switching functions are handled by one Tandem CPU with another acting as backup. When we run tests, we take two of the four CPUs and configure a test system.

The 256-Mbyte disks are used for daily transactions—one disk drive for primary and one for backup. The 128-Mbyte disks are used for control files, processor status files, terminal files, institution-level files, and holiday records. They also contain the routing tables and the system definition file, although these are kept in main memory when the system is in operation. The 64-Mbyte disks contain system files.

The switch sits next to another Tandem system at the National Bank of Detroit that could be reconfigured to take over the CIRRUS load if the primary system failed. About the only thing that would cause a real disaster is a power failure, since there is no backup



The CIRRUS switch uses redundant CPUs and disk drives to maintain a high degree of availability and reliability.

FIGURE 2. CIRRUS Switch Configuration

power supply at the switch site. However, NBD does have a backup site that wouldn't be affected by a power failure in Detroit.

DG. **How much log data do you accumulate on a day-to-day basis?**
Levy. A log record is 411 bytes. There are two log records written for each transaction. One is written into the log file that corresponds to the transaction's anticipated date of settlement. The log file for a Monday would contain between 24,000 and 40,000 records. The switch stops accepting activity for a current date at 8 P.M. Detroit time, which means cutting over to the new log file. Just before cutover, approximately 99 percent of our traffic is already going to the new date of settlement, as specified by the terminal owner.

The other log record is to a backup transaction file that's in time-stamp order. We've been fortunate in that we've never had to use the backup log file.

AS. **What do you do with the log that you've accumulated each day?**
Levy. The log is the source of the batch reports that we send to each node bank's data-processing organization each day. A node's report lists all of its activity for the day in time-stamp order. The report has separate sections for inbound and outbound traffic. The batch run is also used to verify the on-line settlement totals that indicate the balance of each node with the switch. If the batch run produces totals that are consistent with the on-line totals, settlement is effected by transferring funds between "due from" and "due to" accounts that are maintained by each node bank at NBD. The actual transfers are accomplished by the money-management departments of member banks. We only do settlement on a node-to-node basis—each node must deal separately with its member banks.

AS. **What happens when the batch node balances don't agree with the on-line node balances?**
Levy. We look for the reason. Typically we'll be out 25 or 50 cents for a missed transaction fee somewhere. We've never had a serious discrepancy in a consistency check.

AS. **What is the peak transaction load on the switch?**
Levy. The switch was designed to handle 2 transactions per second, which is 16 messages a second. I don't think we've ever achieved that peak—we have never exceeded 500,000 transactions a month. The total elapsed time for a transaction averages between 12 and 15 seconds from request to cash dispense. This includes the processing time at a node bank for approval and debit.

AS. **Suppose I had an account at BayBank and that I went to an ATM run by Landmark Savings and Loan in Pittsburgh, which happens to be a member of the Cash Stream network. Mellon runs Cash Stream and is a node on the CIRRUS network. What happens when I put my BayBank card in the Landmark ATM and request $100?**
Levy. You would get a *customer lead through* on the screen that would look exactly like the lead through for regular Landmark customers. The ATM wouldn't know who you were yet, and so its processor would ask you for your personal identification number (which it would temporarily buffer), what type of account you wanted to access (i.e., savings or checking), and the dollar amount of your transaction.

By this time the Landmark ATM would have already read Track II of the magnetic strip on the back of your BayBank card, and determined your account number, expiration date, and some other variable data that are dependent on BayBank. The first 3 to 11 digits of an account number specify the issuing bank. Track II holds 40 bytes, although we're only using 19 of them right now. Tracks I and III we don't use at all—Track I is for the airline industry, and Track III is used in off-line banking transactions, but not in the interchange environment.

To get back to your transaction, though, the host processor at Landmark has not been involved at all up to this point. Once the ATM has collected all of the data it needs for your $100 withdrawal, it sends a message to its host at Landmark, which determines that the account number on your BayBank card is not a Landmark account, and creates a message to Cash Stream. The Landmark host also writes a suspense record to its log so that it can time out the transaction and free the ATM if Cash Stream does not reply after a certain amount of time. The Landmark host would probably give Cash Stream about 34 seconds before doing this. Most ATMs also have internal timers—we suggest that ATMs abort transactions and return consumers' cards if they do not hear from their hosts in 45 seconds.

When your transaction arrives at the Cash Stream switch, Cash Stream validates it to make sure that it has a valid business date, dollar amount, and so forth. Cash Stream would then look up your bank number in a routing table, determine that BayBank is not a Cash Stream member bank, and then create a message to CIRRUS in a variation of ANSI X9.2 format. The message would contain a lot of information, including a 40-character description of the ATM, to comply with government regulations. Once the message is created, Cash Stream logs it and sends it to the CIRRUS switch via a bisynch point-to-point protocol. Cash Stream waits up to 24 seconds for a response before aborting the transaction.

CIRRUS receives the message, looks up the issuing bank number in its routing table, and then forwards the message to the CIRRUS node that the issuing bank is connected to, which in this case is BayBank. The CIRRUS switch sets a timer for 20 seconds and then waits for the response from BayBank. BayBank receives the message, verifies that the personal identification number (PIN) you supplied is correct, and then checks your balance. If you check out, BayBank writes a memo

post record on your account indicating that a $100 withdrawal is in progress, and then sends an approval message back through CIRRUS, Mellon, and Landmark to the ATM, which dispenses your $100.

The memo post record that BayBank has written is simply noting that a withdrawal is in progress. It will *not* cause $100 to be deducted from your account. That happens with the second round of messages, which I haven't yet discussed.

When the ATM has actually dispensed cash into your hands, it sends a message to Landmark saying that it has given you $100. The completion message goes from the ATM through Landmark and Mellon to CIRRUS, and then CIRRUS sends a message back to Landmark acknowledging the completion message. CIRRUS also sends the completion message on to BayBank, which uses it to create a "postable" record that is used to deduct the money from your account. A completion acknowledgment then goes back from BayBank to CIRRUS. Altogether, your transaction has required four CIRRUS messages—two for the initial approval and two for the completion.

If CIRRUS does not receive a completion acknowledgment from BayBank after 120 seconds, the switch continues sending until it does receive one. If for some reason BayBank cannot deal with the completion message, they call us and we purge it from our file.

Approximately half of our member banks use this protocol. The other half use a shorter protocol that eliminates the second two messages. In our example, this would mean that the CIRRUS switch would still get the completion message from the ATM, but that it would not forward it to BayBank. When this shorter protocol is used, the issuing bank must simply assume that a withdrawal has actually taken place. Obviously, if there's a problem then a message must be sent to the issuing bank instructing it to compensate for the debit it posted.

***AS.* What would happen if BayBank had sent an approval to the Landmark ATM and I had received the $100, but the completion message had not returned from the ATM to BayBank because the Mellon Cash Stream system had failed?**

Levy. If the CIRRUS switch does not get a completion message from Mellon within 120 seconds, then Mellon would be marked down by the switch and would be placed off line for 10 seconds. CIRRUS would send a message to BayBank saying that the cash had not been dispensed. After the 10 seconds had elapsed, we would bring Mellon on line by issuing a series of network-management messages. If the completion message from Mellon came in after the 120-second time had expired, but within the same settlement day, the CIRRUS switch would interrogate the log file to see if there was a record for that message. If there was, the switch would automatically build adjusting entries to both sides of the transaction to ensure that the settlement was handled properly. If the late completion message came in after the settlement cutover for the day, or never came in at all, Landmark, as the ATM operator, would have to present an off-line paper adjustment to the switch, along with some kind of physical evidence of the transaction, like an audit tape or an internally produced report indicating that the transaction had occurred, and a CIRRUS report indicating that they hadn't received value for the transaction. Without that documentation they wouldn't get their money. This kind of thing is not uncommon—some ATMs don't automatically send completion messages to the host. The only way to discover what's happened is to get a status of prior transactions after a new transaction starts.

***DG.* Could you review the way the time-out intervals on your system interact?**

Levy. We have a 20-second timer at the switch for messages to nodes; we advise sending nodes to set a 24-second timer to allow some transmission time. A bank that's not directly connected to the switch, like Landmark in your previous example, may wait as long as 34 seconds to allow the intermediary bank some processing time. We typically tell the ATM issuers to set their timers for about 45 seconds; if the terminal doesn't get a response within that time, it shuts down and returns the customer's card.

***DG.* The two message protocols you described—the four-message protocol and the two-message protocol—make radically different assumptions about when to deduct money from an account, don't they?**

Levy. Traditionally, the U.S. banking industry won't ever deduct anything immediately. We have very few real-time posting, checking, or savings applications in the U.S. That convention doesn't always hold true for other countries, though. In Canada, for instance, there are systems that use real-time posting, which means that a single customer transaction can post several items on a cardholder's account. In the U.S., banks use on-line systems for maintaining account balances. Posted records from a system are merged with other instrument activity (such as checks) during batch processing. The daily batch run is what actually deducts the money from accounts. The result of a daily batch run is a file containing the starting account balances for the system for the next day, as well as records for pending transactions.

***DG.* From what you've been saying, there seems to be little concern that any of the participating banks might be doing anything dishonest. What would happen if a member bank immediately acknowledged that an ATM had dispensed cash when in fact it hadn't. Wouldn't that bank be credited for $100 while the customer was left standing in the rain without any money in his hand?**

Levy. Customers would let us know if there was anything of that nature going on. Also, there's a significant amount of federal regulation that protects customers in

those situations, so it's hardly in the member bank's best interest to do such a thing.

***AS.* But the safety and security of your network depend very much on the safety and security of its constituent parts.**

Levy. Absolutely. CIRRUS was developed as a pipeline between banks that had been doing this kind of thing for 10 or 12 years before we came along. We're another link of potential authorization for the banks. All of the policies and procedures that banks had evolved to protect themselves from error and fraud were carried over directly to the CIRRUS environment.

***DG.* You mentioned before that you partition any part of your network that doesn't respond within a certain amount of time. Is that an important part of your system strategy?**

Levy. Yes. I'm probably one of the few proponents of this kind of partitioning. The more popular strategy is to keep resending messages until a response is received. I believe that the financial implications of transaction processing in the ATM environment require this kind of scrupulousness. It has proved easier for us to balance the network by guaranteeing that transactions flow out of the store-and-forward process in a logical sequence. We've also found that if you give a node time to recover you can often prevent more serious problems.

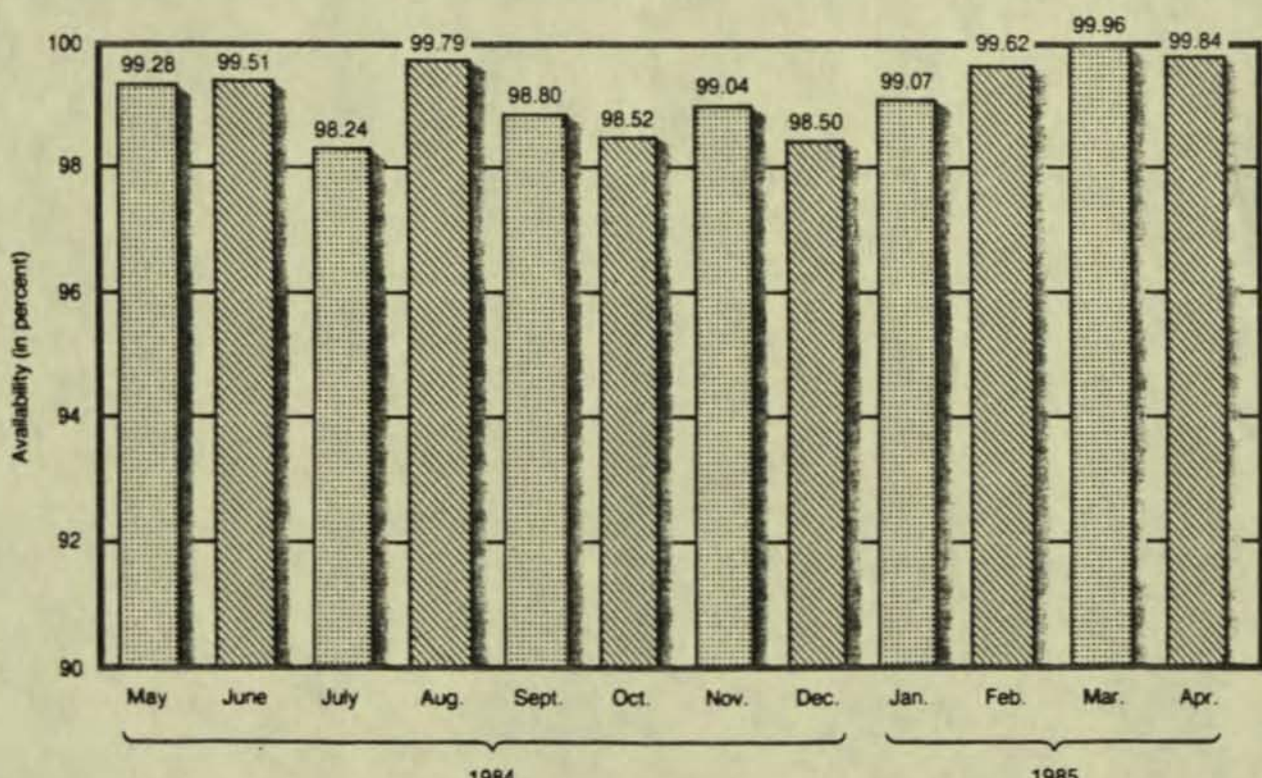
***DG.* How often do banks go off line?**

Levy. We're working to get that rate down to a minimum. We have a standard that no bank should be off line more than five percent of the time, averaged over a month. Some of our members are well above that mark, and some are well below it, but most are off line no more than five percent of the time. Nearly all of our member banks use highly available systems, but as I mentioned earlier, if a node does not get a completion message in time from an ATM, CIRRUS may take the entire node off line. Of course in the case of a delayed completion, a message goes out 10 seconds after a node goes off line to bring it back up again. These short ups and downs have a minimal effect on transaction processing.

***DG.* CIRRUS itself could also go down, couldn't it?**

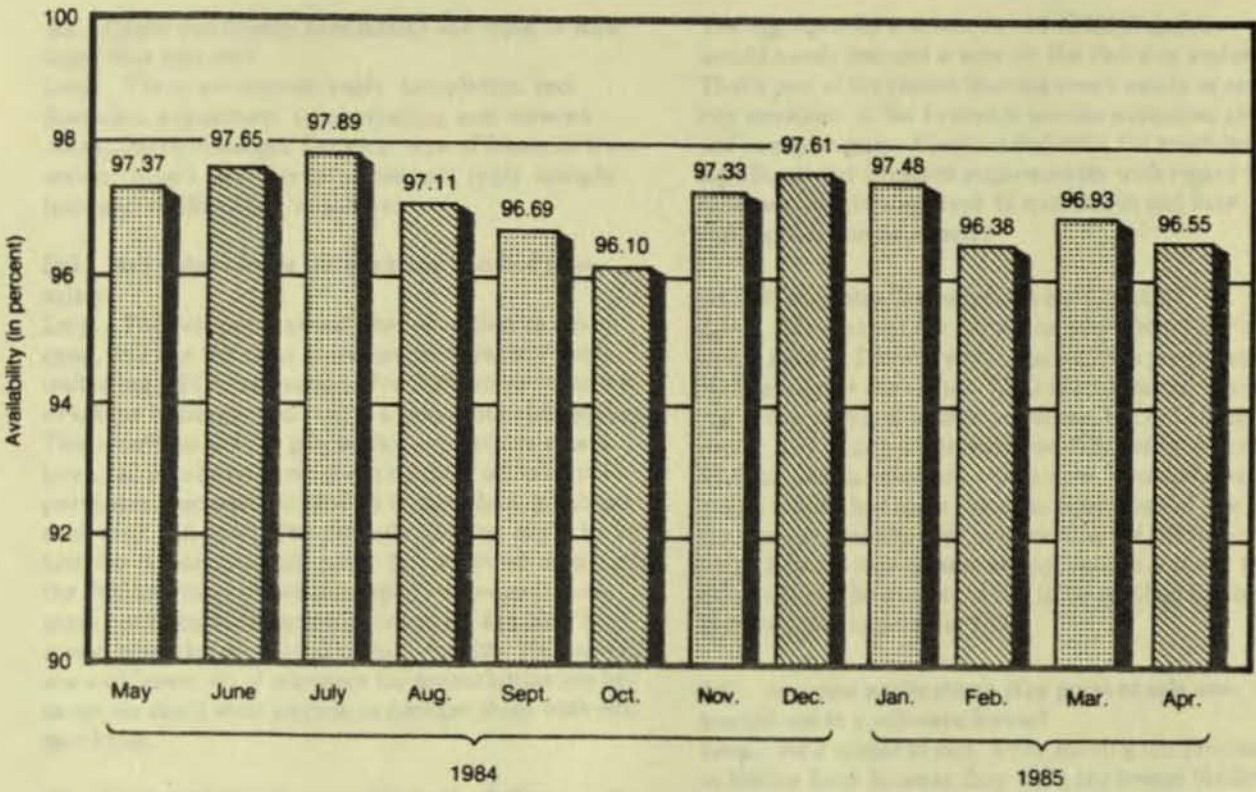
Levy. It's not likely. The average availability of the CIRRUS switch exceeds 98 percent on any given day (see Figure 3). We do not allow scheduled outages of the CIRRUS switch. Actually, we allow one a year, but it still goes against that 98 percent. Tandem equipment does not require outages for hardware maintenance.

***DG.* Considering communication-line failures, central-site failures, and node failures, what's the probability that one node will be able to access another at any given time?**



The average uptime on the CIRRUS switch is between 98 and 100 percent over a 24-hour period. Scheduled downtime is counted as unavailable time.

FIGURE 3. Average Availability on the CIRRUS Switch



Network availability is computed as the probability that one CIRRUS node can access another at any given time. Network availability can be affected by failures in any part of the network, from the switch to the communication lines to the nodes.

FIGURE 4. Average Availability on the CIRRUS Network

Levy. About 97 percent (see Figure 4). In February, for example, the average network uptime was 96.3 percent, which covered a range for individual nodes from 90.13 to 99.6. The 90.13 node was either having a line problem or a processor problem. The mean daily average availability for the switch during February was 99.62 percent.

***AS.* What kept you off the other 0.38 percent of the time during February?**
Levy. The backup file just got incredibly large over the three-day Washington's Birthday weekend, and we had to shut the system down for a minute or two (or fifteen) to reallocate some file space. There was also a date problem because some of our members wanted to process on that Monday.

***DG.* The switch was closed?**
Levy. The switch observes the seven holidays that all 12 Federal Reserve offices observe as federal holidays. Because the National Bank of Detroit isn't open, there's no way to move the money, and if the Fed's not open, there's no way to fund an account at NBD. Certain banks that close for Lincoln's Birthday stay open for Washington's, though, and they were trying to present transactions when there wasn't a valid log file open, and that probably led to some downtime.

***AS.* What fraction of the transactions that reach CIRRUS have to be rejected because they're not destined for a member bank?**
Levy. About 15 percent.

***AS.* Could you describe the message protocol you use?**
Levy. It's called the ANSI Interchange Message Specification for Debit and Credit Card Messages on Financial Institutions—ANSI X9.2-1980. It's a guideline, as all standards are guidelines. It defines 64 fields that are usable in a financial message, and it indicates which fields are mandatory for the type of message being sent. At CIRRUS we've enhanced this standard to meet our particular needs. We use all of the mandatory fields, along with some additional ones. There's a bitmap at the head of every message that shows which fields are being used. We have a fixed set of about 40 message formats that we use. The smallest uses about 170 bytes; the largest, a reversal message, over 400. Reversals are long because they contain a great deal of data from the original transaction. For each type of message, we have both a long and a short format. The long format has all of the fields filled in, and the short one requires that missing fields be filled in by the switch.

AS. Could you briefly summarize the types of messages that you use?
Levy. There are request, reply, completion, confirmation, adjustment, reconciliation, and network-management messages. For each type of financial transaction, there's an entire set of request, reply, completion, and confirmation messages.

DG. Are point-of-sale transactions handled separately?
Levy. Point-of-sale transactions, once they're developed, will use the same messages with various fields indicating a POS transaction. For POS we've created a new type of transaction that is a preauthorized request. This would be used at places like gas stations where intended purchases don't always match up to actual purchases (because who knows exactly how much gas their tank can hold). The preauthorization might be for $20; the customer would pump $18.50 worth of gas, and the POS terminal would then send subsequent messages out to build the posting record for $18.50. The memo post, though, would be held for $20. We want to use a different set of messages for preauthorization because we don't want anyone to confuse them with real purchases.

AS. Is preauthorization similar to the failure condition that occurs when an ATM can only distribute part of a requested sum?
Levy. Yes, except here the discrepancy is the norm instead of the exception.

DG. What are your message latency goals at the switch, and how can you maintain them as your volume increases?
Levy. We don't allow the switch any more than 2 seconds for processing a message. The actual time is usually somewhere between 0.7 and 1 second. The switch is modular enough that if we ever needed to we could open up a mirror of the application so that there could be multiple processors running the application simultaneously, although there would only be one transaction running in any processor at any time. I think we've actually done this at peak processing time.

AS. Do you spool log data onto magnetic tape for permanent safekeeping?
Levy. All the important files, including the log files, are backed up every night. Once a week we back up the entire system, including the application program, and transport it to an off-site storage location. We also put a digested form of the log on microfiche and keep it for five to seven years, to meet regulatory requirements.

DG. How would you compare your electronic funds transfer network to the Fed-wire?
Levy. Well, the Fed-wire is a wholesale-level EFT network, and CIRRUS is a retail-level EFT network. The average dollar value per transaction on the Fed-wire is probably a million times what we handle on CIRRUS. The aggregate for a whole day of CIRRUS activity would barely warrant a wire on the Fed-wire system. That's part of the reason that we aren't nearly as security conscious as the Fed-wire is—our exposures are not nearly as great. Also, the Fed-wire has much more significant and stringent requirements with regard to who can initiate what type of transaction and how transactions can be effected.

AS. Who writes the programs for CIRRUS?
Levy. We contract the system services from the National Bank of Detroit, which maintains a programming staff to support the switch. NBD subcontracted the writing of the switch to a software house, but did all the design and testing on its own, and NBD owns the code. The software is relatively stable now. I would be surprised if NBD had more than the equivalent of one full-time person budgeted for maintaining the CIRRUS software. When a new release of applications is ready to be released into the system, it has to be certified by the system software group at NBD.

DG. Are new applications like point of sale contracted out to a software house?
Levy. As a matter of fact, we're moving our processing to Mellon Bank because they were the lowest bidder for the POS product. Mellon is going to have to redevelop the switch software.

DG. How many lines of code are in your on-line switch application?
Levy. The on-line switch software, which includes message routing and editing, is approximately 9000 lines of TAL, which is the Tandem application language.

DG. How big is your routing table?
Levy. Currently it consists of about 3500 records, with about 20 bytes per record. A routing-table entry includes a bank's prefix, which is from 3 to 11 digits, and an identification of the processor that services the bank. The routing table is called the FIT, for *financial-institution table*. We send an up-to-date copy of the FIT to our member banks every day. It's our intention to build an on-line message that keeps FIT copies up-to-date. The problem is that bringing a new member on can mean bringing as many as 100 new banks on with it. A hundred new FIT entries at one time is a data-management problem that we haven't solved yet.

AS. Do you test a member bank's system before letting it on the CIRRUS network?
Levy. Yes, we certify nodes with a battery of test cases. Every situation we can think of is attempted prior to putting a node on the network. The book that we keep the test cases in is over three inches thick—I believe it contains over 2500 test transactions. When we certify a node, we ask the candidates to have an ATM available in a test environment. We send them plastic cards that run against a simulated card proces-

sor at the switch in Detroit. The switch can also send simulated ATM activity out to them. We find that using an actual ATM is the best way to test.

AS. **How is a typical test case described?**
Levy. In great detail. This is a *very* critical part of our operation. I think part of the success of the network, in fact, can be attributed to the stringency of our testing. We send candidate banks a list of cards and a set of conditions that the cards have to meet. Some of these cards specify accounts that don't exist or that have incorrect balances. A test case instructs the candidate bank to use a particular card for a particular transaction, and then indicates the expected result. We test weekend and holiday processing and all the exception conditions. We also keep our set of test cases up-to-date as we add new functions to the switch.

AS. **What percent of your budget eventually goes for testing?**
Levy. I would say that as a percentage of overall development, testing is easily 25 percent. In an environment like ours, we have to be very careful.

AS. **Have you had any situations where a member bank did not perform properly?**
Levy. Yes, we've had member banks that stopped sending completion messages for one reason or another. On one occasion we had 1500 adjustments for one day's activity. These had to be presented on paper. This swamped their operation center as well as ours. We have also had requests for adjustments that came in as much as 50 business days after a transaction. It's difficult to accept activity that late.

DG. **Could you discuss your basic strategy for security?**
Levy. Within the banking environment, there's a great deal of physical security in addition to application-level security. Most any data that are maintained in a bank's data-processing center are considered confidential and secure. The only part of the system that we don't normally have physical control over is our communication lines, which are protected with link encryption provided by the Racal-Milgo DataCrypter II with a public key option. This is a commercial product that uses DES for encrypting messages. It includes a public key algorithm that allows us to distribute new master keys from time to time.

AS. **Have your communication lines ever been actually attacked?**
Levy. Not that I know. The biggest area for fraud in ATM banking is card and PIN compromise. Either a family member takes a card and cleans out an account, or someone obtains a card and its PIN by force. It's as easy to defraud an ATM that way as to tap the line.

AS. **Are you interested in "smart cards"?**
Levy. They would allow us to put a great deal more data on plastic and would let us do away with PIN verification at the issuing bank. They would also allow us to maintain value on cards, so that customers would be able to put, say, a thousand dollars worth of value on a card that could then be spent without on-line verification. The thing that concerns me is how could we get the postable information to the cardholder's bank and actually deplete his account balance. I see the biggest application of chip cards in this country for things like food stamps where the value is depleted but the transactions don't have to be posted. It's not clear what application smart-card technology will have in banking, which may be why it's taking so long to be developed here. There's also the problem of refitting hundreds of thousands of terminals, and there would have to be a standard for where to put the contacts on the cards.

DG. **Have you considered digital signature verification for point-of-sale transactions?**
Levy. I have my doubts. Handwriting can be compromised. And it's hard to say what kinds of controls we're going to need until we really find out how easy it's going to be to deploy POS. I'm not convinced that signature technology is that much more secure than PINs.

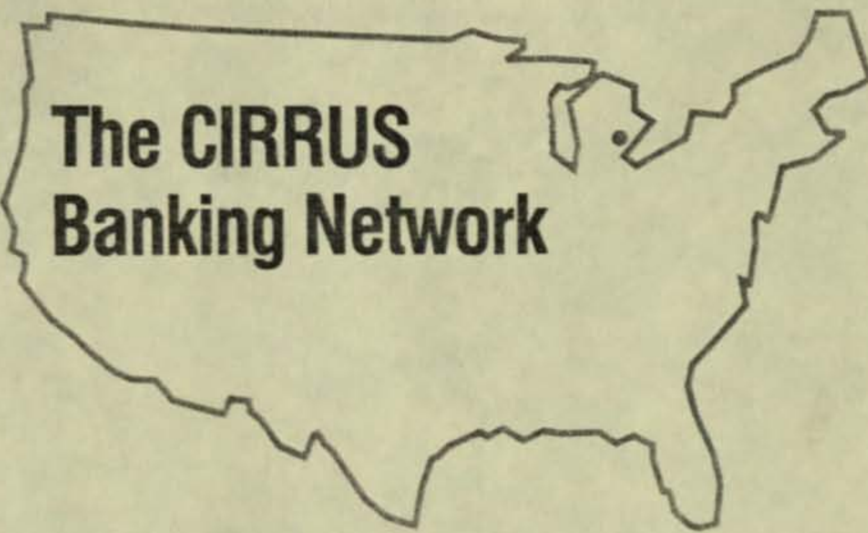
DG. **What's the largest transaction that can be issued over the CIRRUS network for a cash withdrawal?**
Levy. Theoretically, the field size would allow for a $100,000 transaction, but I don't know of a machine in the country that would let you do that. There are machines in Las Vegas that would probably let you take $1000 out in one shot.

AS. **Will future ATMs have change-making capabilities?**
Levy. Yes, there's a machine that makes change, and five-bill dispensers that can give ones, fives, tens, twenties, and fifties or hundreds. The more these machines can do, the more truly they'll be able to replace human tellers, which is really valuable in some applications. There may not be a need for change-making capability at O'Hare Airport, but if someone's going to an ATM regularly for their everyday banking business, it should be able to cash their paycheck. Not everyone's paid in increments of $10.

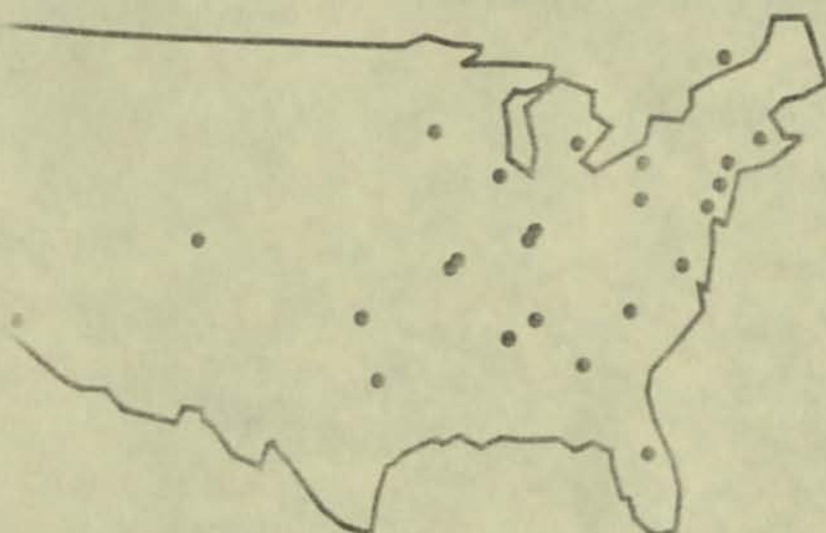
DG. **I think that's all the questions we have for you. Bruce, Jay, thanks for telling us about the CIRRUS system.**

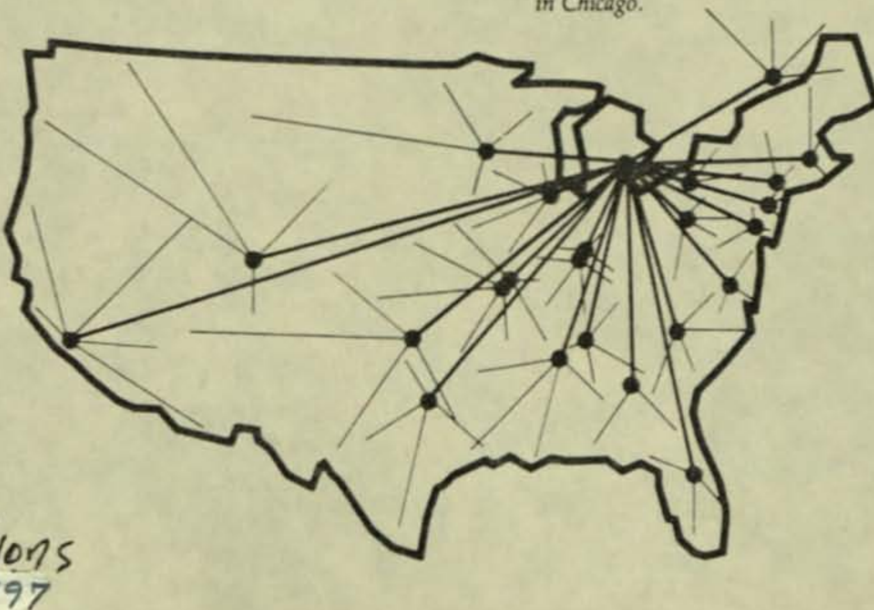
CR Categories and Subject Descriptors: C.4 [**Performance of Systems**]: *reliability, availability, and serviceability*; D.2.9 [**Software Engineering**]: Management—*software quality assurance*; D.4.0 [**Operating Systems**]: General; J.1 [**Administrative Data Processing**]: *financial*; K.6.3 [**Management of Computing and Information Systems**]: Software Management
General Terms: Management, Reliability, Verification
Additional Key Words and Phrases: ATM, CIRRUS, EFT systems

Case Study: The CIRRUS Banking Network



This is the third case study to appear in Communications. *This time, case-study editors David Gifford and Alfred Spector interviewed the president and the director of systems and operations at CIRRUS Systems, Inc., a funds transfer network, to see how distributed systems are being employed in the banking industry. As the following transcript shows, the CIRRUS network provides security, lost message recovery, database consistency via atomic actions, recovery after network partition, and a framework for interorganization cooperation in a distributed system. The CIRRUS system is not the most complex computer system to have been described in these case studies; rather, it is an example of how a few well-chosen ideas can be the basis of a simple and useful system. The interviews were conducted on March 29, 1985, at the headquarters of CIRRUS Systems, Inc., in Chicago.*





ACM Communications
Aug 1985 p. 797

THE CIRRUS BANKING NETWORK

The CIRRUS banking network makes coast-to-coast automatic banking transactions possible. The system will soon be able to handle international currency transactions and point-of-sale transactions in stores.

DAVID GIFFORD and ALFRED SPECTOR

Interview 1. Bruce Burchfield, president of CIRRUS Systems, Inc., talks about the development of the CIRRUS network, its present scope, and plans for its future.

***DG.* Bruce, could you start by telling us something about the history of the CIRRUS network?**

Burchfield. I'd be glad to. Bankers have traditionally provided customers with personalized service. That's the nature of the business. By the 1970s, most banks had automated their "back-room" operations, but were still using people for all their customer interaction. During the 1970s, though, both technology and customer demographics changed. On the one hand, automated teller machines (ATMs) were becoming reliable and cost effective, and on the other, customer demographics were changing and banks were under pressure to provide more convenient access to their services.

Bankers couldn't really afford to provide essential full-branch service for 16 or 18 hours a day, so they installed ATMs to keep certain essential services available around the clock. The next step was to situate these ATMs away from existing bank branches. The choice for bankers was either to build $1 million branches that would cost a half a million a year to operate, or to install 24-hour ATMs for about a $100 thousand each and then $50 thousand a year for maintenance. From the banker's perspective, ATMs are a very cost-effective way of making their services available in more places and at more times.

Then, around 1976, bankers realized they could reduce their costs even further by sharing ATMs. Since there's usually a certain amount of excess capacity on an ATM that's only serving one bank's customers, some banks began to sell their unused capacity by sharing their resources with client banks and then charging a set fee for every transaction processed. Reciprocal sharing arrangements also provide participating banks with better distribution channels, which can mean a competitive advantage.

These developments have allowed banks to reduce staffing and cut back on hours in their regular branches, and to deal with competitive threats from companies like Sears and American Express that are entering the financial-services business with a cost structure that would drive unmodernized banks out of business. It's a traditional paradigm—when the structure of an industry changes, the old leaders are displaced if they can't keep pace.

***DG.* So by the late 1970s you knew not only that ATMs were going to be important, but that ATM sharing was also going to be important. How did the idea of CIRRUS begin to take shape?**

Burchfield. Banks had already formed local and regional sharing arrangements. For example, I had developed a shared automated-teller-machine network in metropolitan Chicago called Cash Station. The next logical step was to provide access for customers regardless of where in the country they happened to be.

CIRRUS was started by 10 banks that wanted to expand their regional networks into a national network. These included BayBank of Boston, Manufacturers Hanover of New York, Mellon Bank of Pittsburgh, and First Chicago. The first discussions were held in December of 1981, and by July of 1982 CIRRUS was incorporated. The initial service that CIRRUS set out to provide was cash access for people away from home.

We knew we needed a certain critical mass to make the network successful, and so we worked to build the network up quickly. As of March 1985, we have 46 states covered. In building this network, we've created a technical infrastructure that's unprecedented in the banking industry. The number of member banks has

now grown to 1425, although only 16 of them are directly connected to the CIRRUS switch. For example, there are over 300 banks in Texas on CIRRUS, but they're all connected to a single node. The Texas regional network handles all intrastate traffic and forwards the remainder to CIRRUS via the Texas CIRRUS node. There are probably 150 million to 200 million transactions a year that are processed by CIRRUS member banks, and yet only a fraction of that goes through the CIRRUS switch. Thus, in many cases, CIRRUS is a brand name as opposed to a processing function. Our node-based architecture has worked out well, and we don't think the politics or the structure of the ATM market will ever be conducive to connecting individual banks directly.

***DG.* What other services do you now support besides cash access?**
Burchfield. We support withdrawals and balance inquiry from checking, savings, and line-of-credit accounts, and we are also developing direct debit point-of-sale services that will allow us to put machines in retail outlets. This summer we're connecting to a Canadian ATM network, and we're planning to do automatic currency conversion for international transactions.

We're also considering the possibility of taking deposits over CIRRUS. Interbank deposit taking already exists on a regional level—on the Chicago network I was involved with, the bank that received a deposit would prove and verify the deposit envelope, process the checks, and return bounced checks to the depositor's bank. The actual transfer of funds between banks is handled electronically.

***DG.* When CIRRUS first started, did you analyze all the different exposures of the system and try to put together a study that could convince all of your participating banks that CIRRUS was going to be secure?**
Burchfield. To resolve the security issue, we assigned liability for all of the things that could possibly go wrong. For example, all of our connection nodes have financial and data security responsibility for their traffic. Extensive logs are kept for audit purposes to help assign liability. This gives each CIRRUS node an active interest in running a secure system. The employees of each bank are bonded, and the switch is operated by a bank with extremely strong audit functions. We also have provisions in our contracts that prohibit banks from using transaction data for competitive analysis.

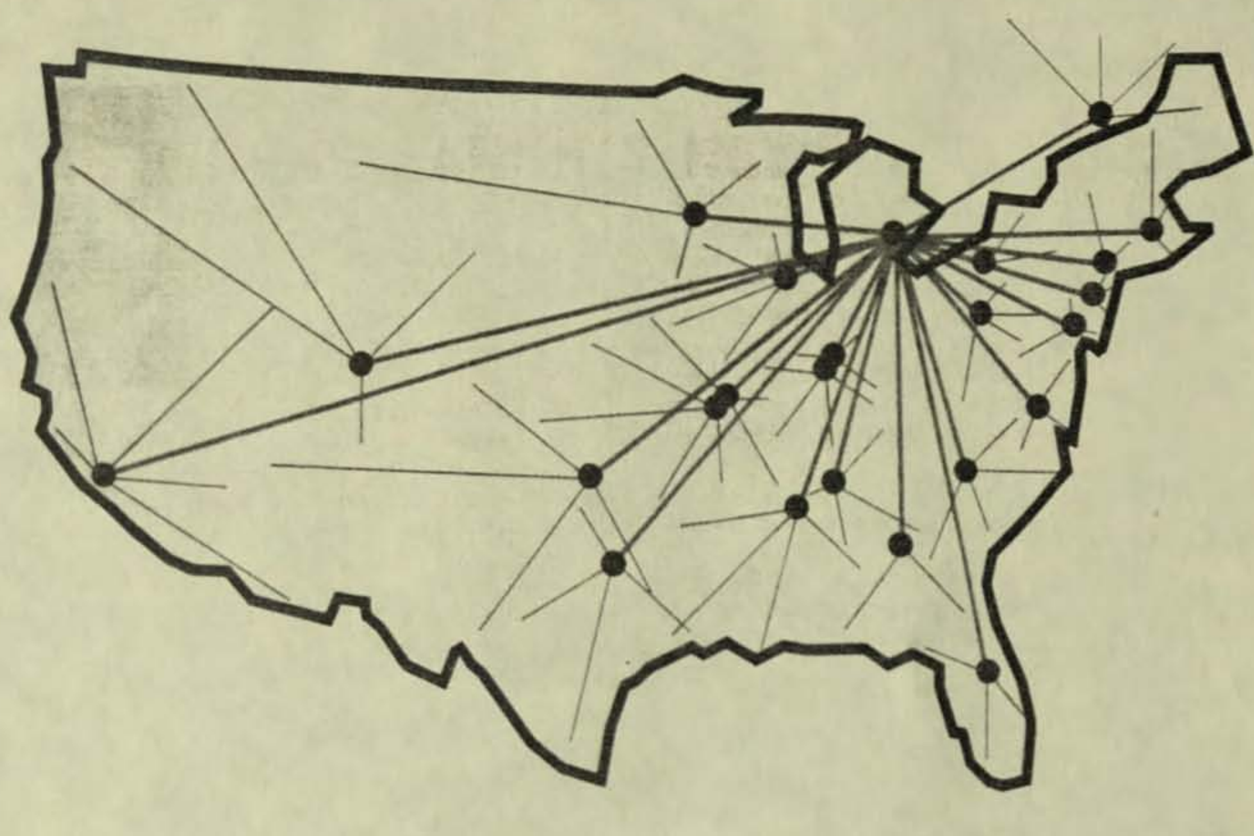
***AS.* With all this delegation, it sounds as though the CIRRUS organization itself doesn't have to worry about liability.**
Burchfield. That's correct. For example, the recent failure of certain savings and loan organizations in Ohio didn't cause us any concern because their CIRRUS attachment node, Central Trust in Cincinnati, is liable for their activity. If the savings and loans went defunct, CIRRUS would not be out of any money because Central Trust is responsible for their transactions. Central Trust actually shut their CIRRUS access off as soon as there was a problem.

BRUCE BURCHFIELD
Bruce Burchfield is president and chief operating officer of CIRRUS Systems, Inc. He was formerly vice-president and manager of the electronic-banking service division at the First National Bank of Chicago, where he was responsible for the development and management of new electronic-banking services, including electronic funds transfer equipment and First Chicago's shared ATM network, Cash Station. Before joining First Chicago, Burchfield was an engineer for Reynolds Aluminum.

***DG.* How are consumers protected against the misuse of their cards?**
Burchfield. They're protected by Federal Regulation E, which says that a customer is not liable for the use of lost or stolen cards as long as the loss is reported within 2 days. When a lost card is reported within 60 days, but after 2 days—that is, within the amount of time it takes to get a bank statement—liability is limited to $50. The issuing bank is responsible for any ATM-related problems that a customer may have. That's what we tell customers when they call us—to contact their issuing bank. That's really all we can do, since we have no authority relative to accounts.

***DG.* Do you provide any services directly to consumers?**
Burchfield. Only one—an 800 number that consumers can call to locate a nearby CIRRUS machine. The database of ATM locations is a geographical matrix that's accessed by area code and telephone exchange. The 800 operator asks for this information and then provides the locations of some nearby terminals. This service is subcontracted to an outside firm.

***DG.* If I ran a bank and felt that I was at a competitive disadvantage because I didn't belong to an ATM network, and I wanted to join CIRRUS, how would you deal with my request?**
Burchfield. To answer that question, let me start by telling you something about our structure (see Figure 1). We have three types of members—Principal, Associate, and Corresponding members. Principal and Associate members have direct links to the CIRRUS switch and



The CIRRUS network consists of 16 4.8 kbit/second lines that radiate from the CIRRUS switch at the National Bank of Detroit (NBD) to the various nodes. Principal and Associate members are directly connected to the CIRRUS switch at NBD. A Principal member has exclusive marketing rights for a particular area, and an Associate member has direct access, but not exclusive rights. Corresponding members are connected to the switch through Principal or Associate members.

FIGURE 1. Topology of the CIRRUS Network

the right to franchise CIRRUS to other financial institutions. Principals and Associates are primarily large financial institutions that operate regional networks.

These Principal and Associate members can bring other banks into the network as Corresponding members. These are generally smaller institutions for which a direct connection would not make economic sense. Therefore, whenever a small bank approaches CIRRUS directly, I refer it to the Principal or Associate member that offers CIRRUS in the bank's state.

DG. **What would happen if Bank X joined a regional ATM network like Cash Stream, without actually joining CIRRUS, and tried to send transactions through the CIRRUS switch via the Cash Stream CIRRUS node?**
Burchfield. Well, first of all, Cash Stream could sign up Bank X as a member of their regional network, and therefore as a member of CIRRUS, but if Cash Stream actually did send the CIRRUS switch a transaction that originated at Bank X, without working through a regional network, that transaction would be rejected because Bank X would not be in the financial institution table at the CIRRUS switch. It wouldn't be a valid transaction.

AS. **Who are your direct competitors?**
Burchfield. There's the Plus network, which is run by a group of banks, and Master Teller, which is operated by Master Card. As of the end of 1984, Plus had 4700 ATMs, Master Teller had 3250, and CIRRUS had 6500. Plus has a competing networks rule that prevents their members from joining another network. Master Teller doesn't have such a rule and neither do we, so some banks belong to both CIRRUS and Master Teller. Master Teller is very big on credit functions, of course, but their card base is limited as far as debit transactions or checking accounts, so there's a certain advantage for customers in being connected to the two networks.

DG. **So if there were two banks that were connected to both CIRRUS and Master Teller, they would have two different ways of routing transactions between themselves. Does that make transaction switching a competitive business?**
Burchfield. Transaction switching is not competitive right now because the bank that decides which network to use is not the bank that pays for the transaction. The destination bank has to pay for the transaction and would thus like to receive it over the more

economical network; the acquiring bank, on the other hand, wants to issue the transaction in the most convenient way, which is probably the least favorable to the user.

AS. **How do CIRRUS member banks recover the cost of processing transactions for other member banks?**
Burchfield. The issuing bank pays the deploying bank 50 cents for a withdrawal and 25 cents for a balance inquiry. The switch gets an additional 25 cents per transaction. Right now, most CIRRUS member banks offer CIRRUS services for free, but my assumption is that members may eventually pass their costs on to the customer. Customers will then have to compare the cost and convenience of CIRRUS with traveler's checks and other alternatives.

AS. **Do you foresee the establishment of gateways for connecting CIRRUS, Master Teller, and Plus?**
Burchfield. We don't think there's any need for connecting ATM services. We see our independence as a competitive advantage. For point-of-sale services, though, it makes sense to allow merchants to accept different kinds of cards. When point-of-sale machines are connected to more than one network, it becomes logical to standardize fees and message formats. These issues are already being addressed by the American Bankers Association and the Electronic Funds Transfer Association (a multi-industry trade association for electronic funds transfer). CIRRUS, Plus, and Master Teller are already meeting to talk about technical standards, to try to build a common gauge railroad, so to speak, so that point-of-sale transactions won't happen in a fragmented way. We're all very competitive, but we realize that it's not going to work unless there are standards.

Interview 2. Jay Levy, director of systems and operations for CIRRUS Systems, Inc., talks about the design and implementation of the network.

AS. **Jay, we'd like you to take us over some of the technical details of the CIRRUS system. Could you start by giving us an idea of how it's organized?**
Levy. Sure. CIRRUS is a star network, with a message switch at the National Bank of Detroit that currently connects to 16 nodes via 4.8 kbit/second dedicated lines. If a line fails, it's replaced by a dial-up circuit that runs at the same speed. The communications protocols are bisync point to point. The CIRRUS switch is the secondary point on each line, so each node processor has primary access for the bid on its line.

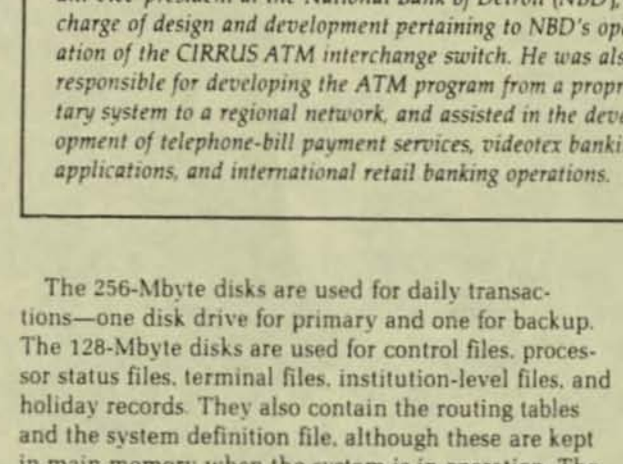
The switch itself is a Tandem NONSTOP II system with four CPUs (See Figure 2). Each CPU has 2.4 megabytes of main memory. The system has a pair of 256-Mbyte drives, a pair of 128-Mbyte drives, and a pair of 64-Mbyte drives. These are shared by all of the CPUs. Most of the switching functions are handled by one Tandem CPU with another acting as backup. When we run tests, we take two of the four CPUs and configure a test system.

JAY LEVY

Jay Levy is director of systems and operations for CIRRUS Systems, Inc. Prior to joining CIRRUS, Levy was an assistant vice-president at the National Bank of Detroit (NBD), in charge of design and development pertaining to NBD's operation of the CIRRUS ATM interchange switch. He was also responsible for developing the ATM program from a proprietary system to a regional network, and assisted in the development of telephone-bill payment services, videotex banking applications, and international retail banking operations.

The 256-Mbyte disks are used for daily transactions—one disk drive for primary and one for backup. The 128-Mbyte disks are used for control files, processor status files, terminal files, institution-level files, and holiday records. They also contain the routing tables and the system definition file, although these are kept in main memory when the system is in operation. The 64-Mbyte disks contain system files.

The switch sits next to another Tandem system at the National Bank of Detroit that could be reconfigured to take over the CIRRUS load if the primary system failed. About the only thing that would cause a real disaster is a power failure, since there is no backup



The CIRRUS switch uses redundant CPUs and disk drives to maintain a high degree of availability and reliability.

FIGURE 2. CIRRUS Switch Configuration

power supply at the switch site. However, NBD does have a backup site that wouldn't be affected by a power failure in Detroit.

DG. How much log data do you accumulate on a day-to-day basis?

Levy. A log record is 411 bytes. There are two log records written for each transaction. One is written into the log file that corresponds to the transaction's anticipated date of settlement. The log file for a Monday would contain between 24,000 and 40,000 records. The switch stops accepting activity for a current date at 8 P.M. Detroit time, which means cutting over to the new log file. Just before cutover, approximately 99 percent of our traffic is already going to the new date of settlement, as specified by the terminal owner.

The other log record is to a backup transaction file that's in time-stamp order. We've been fortunate in that we've never had to use the backup log file.

AS. What do you do with the log that you've accumulated each day?

Levy. The log is the source of the batch reports that we send to each node bank's data-processing organization each day. A node's report lists all of its activity for the day in time-stamp order. The report has separate sections for inbound and outbound traffic. The batch run is also used to verify the on-line settlement totals that indicate the balance of each node with the switch. If the batch run produces totals that are consistent with the on-line totals, settlement is effected by transferring funds between "due from" and "due to" accounts that are maintained by each node bank at NBD. The actual transfers are accomplished by the money-management departments of member banks. We only do settlement on a node-to-node basis—each node must deal separately with its member banks.

AS. What happens when the batch node balances don't agree with the on-line node balances?

Levy. We look for the reason. Typically we'll be out 25 or 50 cents for a missed transaction fee somewhere. We've never had a serious discrepancy in a consistency check.

AS. What is the peak transaction load on the switch?

Levy. The switch was designed to handle 2 transactions per second, which is 16 messages a second. I don't think we've ever achieved that peak—we have never exceeded 500,000 transactions a month. The total elapsed time for a transaction averages between 12 and 15 seconds frôm request to cash dispense. This includes the processing time at a node bank for approval and debit.

AS. Suppose I had an account at BayBank and that I went to an ATM run by Landmark Savings and Loan in Pittsburgh, which happens to be a member of the Cash Stream network. Mellon runs Cash Stream and is a node on the CIRRUS network. What happens when I put my BayBank card in the Landmark ATM and request $100?

Levy. You would get a *customer lead through* on the screen that would look exactly like the lead through for regular Landmark customers. The ATM wouldn't know who you were yet, and so its processor would ask you for your personal identification number (which it would temporarily buffer), what type of account you wanted to access (i.e., savings or checking), and the dollar amount of your transaction.

By this time the Landmark ATM would have already read Track II of the magnetic strip on the back of your BayBank card, and determined your account number, expiration date, and some other variable data that are dependent on BayBank. The first 3 to 11 digits of an account number specify the issuing bank. Track II holds 40 bytes, although we're only using 19 of them right now. Tracks I and III we don't use at all—Track I is for the airline industry, and Track III is used in off-line banking transactions, but not in the interchange environment.

To get back to your transaction, though, the host processor at Landmark has not been involved at all up to this point. Once the ATM has collected all of the data it needs for your $100 withdrawal, it sends a message to its host at Landmark, which determines that the account number on your BayBank card is not a Landmark account, and creates a message to Cash Stream. The Landmark host also writes a suspense record to its log so that it can time out the transaction and free the ATM if Cash Stream does not reply after a certain amount of time. The Landmark host would probably give Cash Stream about 34 seconds before doing this. Most ATMs also have internal timers—we suggest that ATMs abort transactions and return consumers' cards if they do not hear from their hosts in 45 seconds.

When your transaction arrives at the Cash Stream switch, Cash Stream validates it to make sure that it has a valid business date, dollar amount, and so forth. Cash Stream would then look up your bank number in a routing table, determine that BayBank is not a Cash Stream member bank, and then create a message to CIRRUS in a variation of ANSI X9.2 format. The message would contain a lot of information, including a 40-character description of the ATM, to comply with government regulations. Once the message is created, Cash Stream logs it and sends it to the CIRRUS switch via a bisynch point-to-point protocol. Cash Stream waits up to 24 seconds for a response before aborting the transaction.

CIRRUS receives the message, looks up the issuing bank number in its routing table, and then forwards the message to the CIRRUS node that the issuing bank is connected to, which in this case is BayBank. The CIRRUS switch sets a timer for 20 seconds and then waits for the response from BayBank. BayBank receives the message, verifies that the personal identification number (PIN) you supplied is correct, and then checks your balance. If you check out, BayBank writes a memo

post record on your account indicating that a $100 withdrawal is in progress, and then sends an approval message back through CIRRUS, Mellon, and Landmark to the ATM, which dispenses your $100.

The memo post record that BayBank has written is simply noting that a withdrawal is in progress. It will *not* cause $100 to be deducted from your account. That happens with the second round of messages, which I haven't yet discussed.

When the ATM has actually dispensed cash into your hands, it sends a message to Landmark saying that it has given you $100. The completion message goes from the ATM through Landmark and Mellon to CIRRUS, and then CIRRUS sends a message back to Landmark acknowledging the completion message. CIRRUS also sends the completion message on to BayBank, which uses it to create a "postable" record that is used to deduct the money from your account. A completion acknowledgment then goes back from BayBank to CIRRUS. Altogether, your transaction has required four CIRRUS messages—two for the initial approval and two for the completion.

If CIRRUS does not receive a completion acknowledgment from BayBank after 120 seconds, the switch continues sending until it does receive one. If for some reason BayBank cannot deal with the completion message, they call us and we purge it from our file.

Approximately half of our member banks use this protocol. The other half use a shorter protocol that eliminates the second two messages. In our example, this would mean that the CIRRUS switch would still get the completion message from the ATM, but that it would not forward it to BayBank. When this shorter protocol is used, the issuing bank must simply assume that a withdrawal has actually taken place. Obviously, if there's a problem then a message must be sent to the issuing bank instructing it to compensate for the debit it posted.

***AS.* What would happen if BayBank had sent an approval to the Landmark ATM and I had received the $100, but the completion message had not returned from the ATM to BayBank because the Mellon Cash Stream system had failed?**

Levy. If the CIRRUS switch does not get a completion message from Mellon within 120 seconds, then Mellon would be marked down by the switch and would be placed off line for 10 seconds. CIRRUS would send a message to BayBank saying that the cash had not been dispensed. After the 10 seconds had elapsed, we would bring Mellon on line by issuing a series of network-management messages. If the completion message from Mellon came in after the 120-second time had expired, but within the same settlement day, the CIRRUS switch would interrogate the log file to see if there was a record for that message. If there was, the switch would automatically build adjusting entries to both sides of the transaction to ensure that the settlement was handled properly. If the late completion message came in after the settlement cutover for the day, or never came in at all, Landmark, as the ATM operator, would have to present an off-line paper adjustment to the switch, along with some kind of physical evidence of the transaction, like an audit tape or an internally produced report indicating that the transaction had occurred, and a CIRRUS report indicating that they hadn't received value for the transaction. Without that documentation they wouldn't get their money. This kind of thing is not uncommon—some ATMs don't automatically send completion messages to the host. The only way to discover what's happened is to get a status of prior transactions after a new transaction starts.

***DG.* Could you review the way the time-out intervals on your system interact?**

Levy. We have a 20-second timer at the switch for messages to nodes; we advise sending nodes to set a 24-second timer to allow some transmission time. A bank that's not directly connected to the switch, like Landmark in your previous example, may wait as long as 34 seconds to allow the intermediary bank some processing time. We typically tell the ATM issuers to set their timers for about 45 seconds; if the terminal doesn't get a response within that time, it shuts down and returns the customer's card.

***DG.* The two message protocols you described—the four-message protocol and the two-message protocol—make radically different assumptions about when to deduct money from an account, don't they?**

Levy. Traditionally, the U.S. banking industry won't ever deduct anything immediately. We have very few real-time posting, checking, or savings applications in the U.S. That convention doesn't always hold true for other countries, though. In Canada, for instance, there are systems that use real-time posting, which means that a single customer transaction can post several items on a cardholder's account. In the U.S., banks use on-line systems for maintaining account balances. Posted records from a system are merged with other instrument activity (such as checks) during batch processing. The daily batch run is what actually deducts the money from accounts. The result of a daily batch run is a file containing the starting account balances for the system for the next day, as well as records for pending transactions.

***DG.* From what you've been saying, there seems to be little concern that any of the participating banks might be doing anything dishonest. What would happen if a member bank immediately acknowledged that an ATM had dispensed cash when in fact it hadn't. Wouldn't that bank be credited for $100 while the customer was left standing in the rain without any money in his hand?**

Levy. Customers would let us know if there was anything of that nature going on. Also, there's a significant amount of federal regulation that protects customers in

those situations, so it's hardly in the member bank's best interest to do such a thing.

AS. **But the safety and security of your network depend very much on the safety and security of its constituent parts.**

Levy. Absolutely. CIRRUS was developed as a pipeline between banks that had been doing this kind of thing for 10 or 12 years before we came along. We're another link of potential authorization for the banks. All of the policies and procedures that banks had evolved to protect themselves from error and fraud were carried over directly to the CIRRUS environment.

DG. **You mentioned before that you partition any part of your network that doesn't respond within a certain amount of time. Is that an important part of your system strategy?**

Levy. Yes. I'm probably one of the few proponents of this kind of partitioning. The more popular strategy is to keep resending messages until a response is received. I believe that the financial implications of transaction processing in the ATM environment require this kind of scrupulousness. It has proved easier for us to balance the network by guaranteeing that transactions flow out of the store-and-forward process in a logical sequence. We've also found that if you give a node time to recover you can often prevent more serious problems.

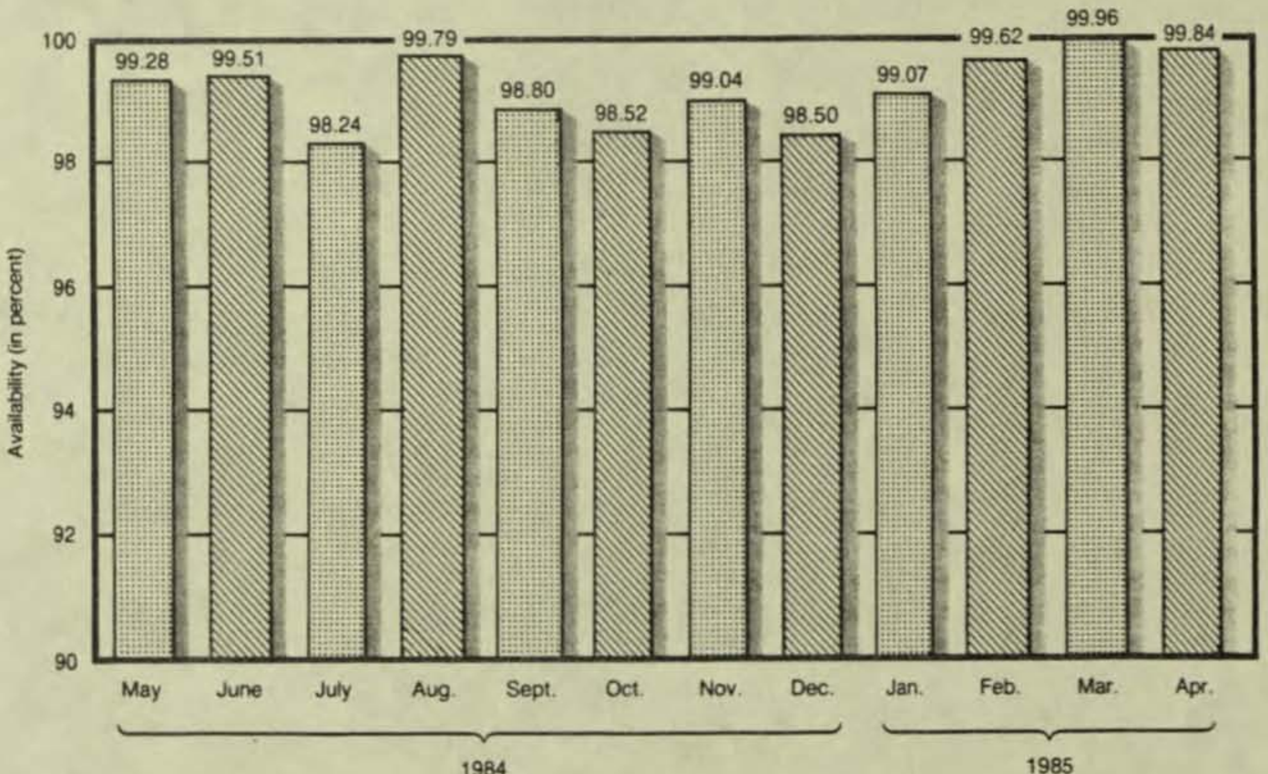
DG. **How often do banks go off line?**

Levy. We're working to get that rate down to a minimum. We have a standard that no bank should be off line more than five percent of the time, averaged over a month. Some of our members are well above that mark, and some are well below it, but most are off line no more than five percent of the time. Nearly all of our member banks use highly available systems, but as I mentioned earlier, if a node does not get a completion message in time from an ATM, CIRRUS may take the entire node off line. Of course in the case of a delayed completion, a message goes out 10 seconds after a node goes off line to bring it back up again. These short ups and downs have a minimal effect on transaction processing.

DG. **CIRRUS itself could also go down, couldn't it?**

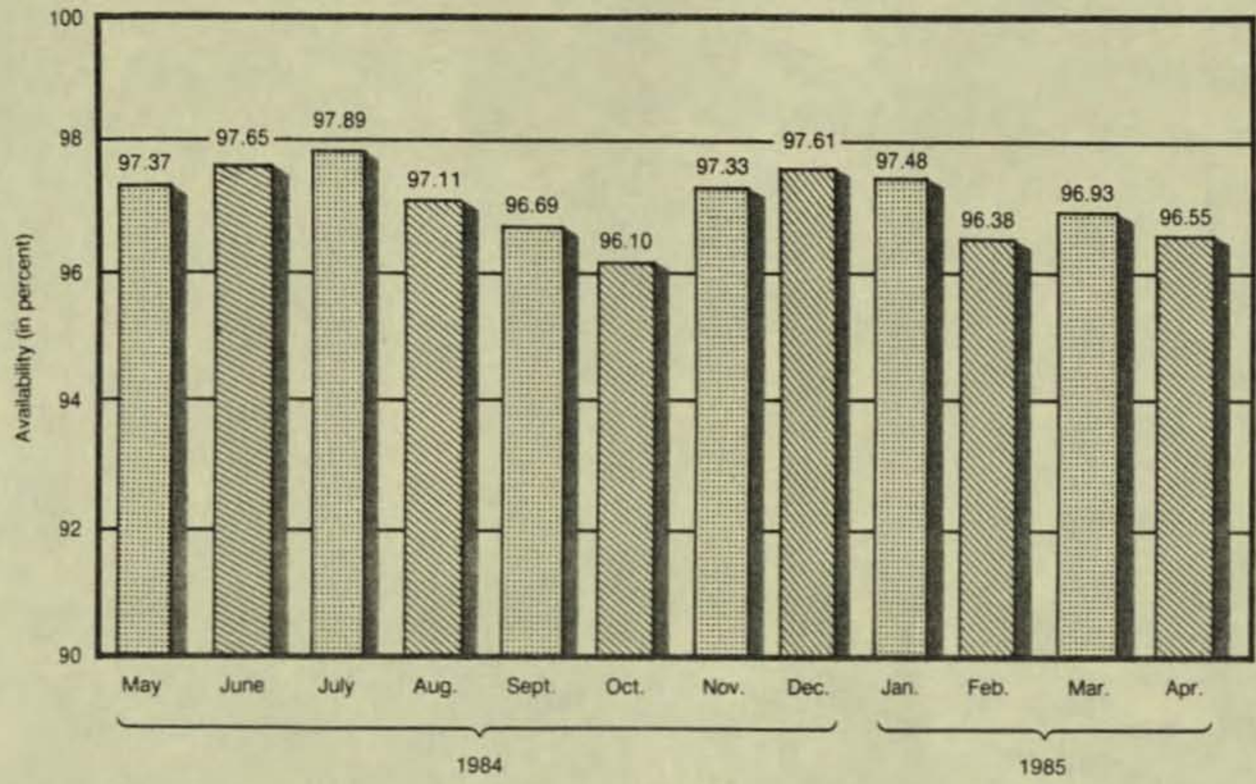
Levy. It's not likely. The average availability of the CIRRUS switch exceeds 98 percent on any given day (see Figure 3). We do not allow scheduled outages of the CIRRUS switch. Actually, we allow one a year, but it still goes against that 98 percent. Tandem equipment does not require outages for hardware maintenance. *

DG. **Considering communication-line failures, central-site failures, and node failures, what's the probability that one node will be able to access another at any given time?**



The average uptime on the CIRRUS switch is between 98 and 100 percent over a 24-hour period. Scheduled downtime is counted as unavailable time.

FIGURE 3. Average Availability on the CIRRUS Switch



Network availability is computed as the probability that one CIRRUS node can access another at any given time. Network availability can be affected by failures in any part of the network, from the switch to the communication lines to the nodes.

FIGURE 4. Average Availability on the CIRRUS Network

Levy. About 97 percent (see Figure 4). In February, for example, the average network uptime was 96.3 percent, which covered a range for individual nodes from 90.13 to 99.6. The 90.13 node was either having a line problem or a processor problem. The mean daily average availability for the switch during February was 99.62 percent.

AS. **What kept you off the other 0.38 percent of the time during February?**
Levy. The backup file just got incredibly large over the three-day Washington's Birthday weekend, and we had to shut the system down for a minute or two (or fifteen) to reallocate some file space. There was also a date problem because some of our members wanted to process on that Monday.

DG. **The switch was closed?**
Levy. The switch observes the seven holidays that all 12 Federal Reserve offices observe as federal holidays. Because the National Bank of Detroit isn't open, there's no way to move the money, and if the Fed's not open, there's no way to fund an account at NBD. Certain banks that close for Lincoln's Birthday stay open for Washington's, though, and they were trying to present transactions when there wasn't a valid log file open, and that probably led to some downtime.

AS. **What fraction of the transactions that reach CIRRUS have to be rejected because they're not destined for a member bank?**
Levy. About 15 percent.

AS. **Could you describe the message protocol you use?**
Levy. It's called the ANSI Interchange Message Specification for Debit and Credit Card Messages on Financial Institutions—ANSI X9.2-1980. It's a guideline, as all standards are guidelines. It defines 64 fields that are usable in a financial message, and it indicates which fields are mandatory for the type of message being sent. At CIRRUS we've enhanced this standard to meet our particular needs. We use all of the mandatory fields, along with some additional ones. There's a bitmap at the head of every message that shows which fields are being used. We have a fixed set of about 40 message formats that we use. The smallest uses about 170 bytes; the largest, a reversal message, over 400. Reversals are long because they contain a great deal of data from the original transaction. For each type of message, we have both a long and a short format. The long format has all of the fields filled in, and the short one requires that missing fields be filled in by the switch.

AS. **Could you briefly summarize the types of messages that you use?**

Levy. There are request, reply, completion, confirmation, adjustment, reconciliation, and network-management messages. For each type of financial transaction, there's an entire set of request, reply, completion, and confirmation messages.

DG. **Are point-of-sale transactions handled separately?**

Levy. Point-of-sale transactions, once they're developed, will use the same messages with various fields indicating a POS transaction. For POS we've created a new type of transaction that is a preauthorized request. This would be used at places like gas stations where intended purchases don't always match up to actual purchases (because who knows exactly how much gas their tank can hold). The preauthorization might be for $20; the customer would pump $18.50 worth of gas, and the POS terminal would then send subsequent messages out to build the posting record for $18.50. The memo post, though, would be held for $20. We want to use a different set of messages for preauthorization because we don't want anyone to confuse them with real purchases.

AS. **Is preauthorization similar to the failure condition that occurs when an ATM can only distribute part of a requested sum?**

Levy. Yes, except here the discrepancy is the norm instead of the exception.

DG. **What are your message latency goals at the switch, and how can you maintain them as your volume increases?**

Levy. We don't allow the switch any more than 2 seconds for processing a message. The actual time is usually somewhere between 0.7 and 1 second. The switch is modular enough that if we ever needed to we could open up a mirror of the application so that there could be multiple processors running the application simultaneously, although there would only be one transaction running in any processor at any time. I think we've actually done this at peak processing time.

AS. **Do you spool log data onto magnetic tape for permanent safekeeping?**

Levy. All the important files, including the log files, are backed up every night. Once a week we back up the entire system, including the application program, and transport it to an off-site storage location. We also put a digested form of the log on microfiche and keep it for five to seven years, to meet regulatory requirements.

DG. **How would you compare your electronic funds transfer network to the Fed-wire?**

Levy. Well, the Fed-wire is a wholesale-level EFT network, and CIRRUS is a retail-level EFT network. The average dollar value per transaction on the Fed-wire is probably a million times what we handle on CIRRUS. The aggregate for a whole day of CIRRUS activity would barely warrant a wire on the Fed-wire system. That's part of the reason that we aren't nearly as security conscious as the Fed-wire is—our exposures are not nearly as great. Also, the Fed-wire has much more significant and stringent requirements with regard to who can initiate what type of transaction and how transactions can be effected.

AS. **Who writes the programs for CIRRUS?**

Levy. We contract the system services from the National Bank of Detroit, which maintains a programming staff to support the switch. NBD subcontracted the writing of the switch to a software house, but did all the design and testing on its own, and NBD owns the code. The software is relatively stable now. I would be surprised if NBD had more than the equivalent of one full-time person budgeted for maintaining the CIRRUS software. When a new release of applications is ready to be released into the system, it has to be certified by the system software group at NBD.

DG. **Are new applications like point of sale contracted out to a software house?**

Levy. As a matter of fact, we're moving our processing to Mellon Bank because they were the lowest bidder for the POS product. Mellon is going to have to redevelop the switch software.

DG. **How many lines of code are in your on-line switch application?**

Levy. The on-line switch software, which includes message routing and editing, is approximately 9000 lines of TAL, which is the Tandem application language.

DG. **How big is your routing table?**

Levy. Currently it consists of about 3500 records, with about 20 bytes per record. A routing-table entry includes a bank's prefix, which is from 3 to 11 digits, and an identification of the processor that services the bank. The routing table is called the FIT, for *financial-institution table*. We send an up-to-date copy of the FIT to our member banks every day. It's our intention to build an on-line message that keeps FIT copies up-to-date. The problem is that bringing a new member on can mean bringing as many as 100 new banks on with it. A hundred new FIT entries at one time is a data-management problem that we haven't solved yet.

AS. **Do you test a member bank's system before letting it on the CIRRUS network?**

Levy. Yes, we certify nodes with a battery of test cases. Every situation we can think of is attempted prior to putting a node on the network. The book that we keep the test cases in is over three inches thick—I believe it contains over 2500 test transactions. When we certify a node, we ask the candidates to have an ATM available in a test environment. We send them plastic cards that run against a simulated card proces-

sor at the switch in Detroit. The switch can also send simulated ATM activity out to them. We find that using an actual ATM is the best way to test.

AS. **How is a typical test case described?**
Levy. In great detail. This is a *very* critical part of our operation. I think part of the success of the network, in fact, can be attributed to the stringency of our testing. We send candidate banks a list of cards and a set of conditions that the cards have to meet. Some of these cards specify accounts that don't exist or that have incorrect balances. A test case instructs the candidate bank to use a particular card for a particular transaction, and then indicates the expected result. We test weekend and holiday processing and all the exception conditions. We also keep our set of test cases up-to-date as we add new functions to the switch.

AS. **What percent of your budget eventually goes for testing?**
Levy. I would say that as a percentage of overall development, testing is easily 25 percent. In an environment like ours, we have to be very careful.

AS. **Have you had any situations where a member bank did not perform properly?**
Levy. Yes, we've had member banks that stopped sending completion messages for one reason or another. On one occasion we had 1500 adjustments for one day's activity. These had to be presented on paper. This swamped their operation center as well as ours. We have also had requests for adjustments that came in as much as 50 business days after a transaction. It's difficult to accept activity that late.

DG. **Could you discuss your basic strategy for security?**
Levy. Within the banking environment, there's a great deal of physical security in addition to application-level security. Most any data that are maintained in a bank's data-processing center are considered confidential and secure. The only part of the system that we don't normally have physical control over is our communication lines, which are protected with link encryption provided by the Racal-Milgo DataCrypter II with a public key option. This is a commercial product that uses DES for encrypting messages. It includes a public key algorithm that allows us to distribute new master keys from time to time.

AS. **Have your communication lines ever been actually attacked?**
Levy. Not that I know. The biggest area for fraud in ATM banking is card and PIN compromise: Either a family member takes a card and cleans out an account, or someone obtains a card and its PIN by force. It's as easy to defraud an ATM that way as to tap the line.

AS. **Are you interested in "smart cards"?**
Levy. They would allow us to put a great deal more data on plastic and would let us do away with PIN verification at the issuing bank. They would also allow us to maintain value on cards, so that customers would be able to put, say, a thousand dollars worth of value on a card that could then be spent without on-line verification. The thing that concerns me is how could we get the postable information to the cardholder's bank and actually deplete his account balance. I see the biggest application of chip cards in this country for things like food stamps where the value is depleted but the transactions don't have to be posted. It's not clear what application smart-card technology will have in banking, which may be why it's taking so long to be developed here. There's also the problem of refitting hundreds of thousands of terminals, and there would have to be a standard for where to put the contacts on the cards.

DG. **Have you considered digital signature verification for point-of-sale transactions?**
Levy. I have my doubts. Handwriting can be compromised. And it's hard to say what kinds of controls we're going to need until we really find out how easy it's going to be to deploy POS. I'm not convinced that signature technology is that much more secure than PINs.

DG. **What's the largest transaction that can be issued over the CIRRUS network for a cash withdrawal?**
Levy. Theoretically, the field size would allow for a $100,000 transaction, but I don't know of a machine in the country that would let you do that. There are machines in Las Vegas that would probably let you take $1000 out in one shot.

AS. **Will future ATMs have change-making capabilities?**
Levy. Yes, there's a machine that makes change, and five-bill dispensers that can give ones, fives, tens, twenties, and fifties or hundreds. The more these machines can do, the more truly they'll be able to replace human tellers, which is really valuable in some applications. There may not be a need for change-making capability at O'Hare Airport, but if someone's going to an ATM regularly for their everyday banking business, it should be able to cash their paycheck. Not everyone's paid in increments of $10.

DG. **I think that's all the questions we have for you. Bruce, Jay, thanks for telling us about the CIRRUS system.**

CR Categories and Subject Descriptors: C.4 [**Performance of Systems**]: *reliability, availability, and serviceability*; D.2.9 [**Software Engineering**]: Management—*software quality assurance*; D.4.0 [**Operating Systems**]: General; J.1 [**Administrative Data Processing**]: *financial*; K.6.3 [**Management of Computing and Information Systems**]: Software Management
General Terms: Management, Reliability, Verification
Additional Key Words and Phrases: ATM, CIRRUS, EFT systems

- Auerbach Reprint on Prototyping -
8/83

Selma - I'm enclosing the folio we

AUERBACH
publishers inc.
6560 n. park dr.
pennsauken
n. j. 08109

609-662-2070
1-800-257-8162
EXT. [redacted]
326

PEGGY MACK
SALES REPRESENTATIVE

discussed. I will talk to you when your trial arrives.

Auerbach Reprint on Prototyping
8/83

INSIDE:
Prototyping and Systems Development Methodologies, Developmental Environment, Impact on Development

Prototyping

PAYOFF IDEA. Prototyping is a method whereby the systems development team puts together a model of the proposed system in conjunction with the user. Although the initial prototype does not contain all possible logic conditions, it clearly shows all functional elements of the system. Prototyping, while still retaining management control, allows a return to the informal approach to systems development that existed prior to formalized development methodologies. This portfolio describes the advantages of prototyping as a method of systems development; it also presents a case study of how one system was developed.

PROBLEMS ADDRESSED

Prototyping has always been an option for systems developers; however, the technological tools that enabled quick and relatively inexpensive development of the prototype as more than a throwaway commodity were not available. In addition, the available coding structure was not sufficiently flexible to allow the constant and radical change often necessary to develop an acceptable model. Although the systems themselves have progressed from batch to real-time, online processing, a batch mentality still applies to the method of actually developing these systems, as attested to by the following:

- Systems requirements are captured at one particular point and are processed under set conditions.
- Inquiry and updating of stated requirements can be made only at predetermined times according to rigid rules.

The Technology

In a recent dramatic change, interactive development tools allowing online design of a prototype have become available. They are generally based on fourth-generation programming languages integrated with a relational data base and data dictionary.

Fourth-generation programming languages are nonprocedural and contain an extremely powerful instruction set; one instruction is equivalent to many procedural instructions in a third-generation programming language (e.g., COBOL, PL/1). Current literature suggests that fourth-generation languages are more than 10 times as productive as third-generation ones.

A relational data base is organized on the basis of a flat file structure in which relationships between records are based on data values and not through the creation of pointers (as is the case in network and hierarchical data base structures). To be related within a relational data base, a common data value must exist within each record. It is generally accepted that the relational data base approach provides extremely flexible access to data. This flexibility is of paramount importance when building a prototype. The definition of fields and record types and, therefore, the relationships between data can be dynamically altered without having to redesign and rewrite the applications code.

The data dictionary is a centrally maintained description of the fields, field attributes, records, and files that make up the data base. To provide the most effective development environment, these prototyping tools (i.e., fourth-generation language, relational data base, and data dictionary) must be fully compatible with one another. This compatibility can be achieved by using a single product (e.g., Computer Corporation of America's data base management system, Model 204) or by creating a totally integrated modular environment (such as those supplied by Tandem Computers).

PROTOTYPING AND SYSTEMS DEVELOPMENT METHODOLOGIES

Systems development methodologies have certainly helped developers focus on the main issues and have contributed greatly to improved project control in terms of time and dollar management (although there is room for further improvement in these areas). Despite the introduction of and adherence to a systems development methodology, however, systems that do not meet user needs are still being developed. Users still complain that the DP department does not provide what is requested; the DP department still contends that users do not know what they want and are given as close an approximation of their needs as the department can determine. There is undoubtedly some truth in both arguments; unfortunately, rather than solving the problem of providing the user with an acceptable system, systems development methodologies can compound it. The following describes some difficulties caused by systems development methodologies and how, when applied, prototyping can eradicate them.

Production Overhead

Following the protocols defined by the methodology consumes valuable project time. The argument used to justify this additional time is that the project will meet user requirements more accurately if the analysis and definition of the project are properly completed and approved. As explained in the following sections, however, the review and sign-off procedures become meaningless because of the volume of paper submitted to the user for review and the timing constraints associated with the sign-off procedure.

Through prototyping, systems specifications can be developed and refined. The prototype then becomes an evolutionary vehicle that eventually will be expanded to full production status. This eliminates the previous redundancy of writing various levels of specifications before any real development work could begin. In addition, the technology associated with prototyping is demonstrably more productive than prior methods of systems development. The facilities offered by fourth-generation programming languages and the flexibility inherent in a relational data base mean that a prototype can be developed in a relatively short time.

Voluminous User Documentation

Various documents are produced during development as charted by the methodology, which indicates the content framework for each document. Standardizing this paperwork results in voluminous documentation regardless of the size and complexity of the system being designed. Asked to review these documents for completeness and accuracy, the user encounters too much narrative, system-oriented information to be able to comment with any degree of certainty. In addition, it is assumed that user requirements are stationary between the analysis phase and implementation; such an assumption is totally unrealistic in today's business environment. Changes to the documents produced through the methodology (and signed off by the user), are viewed suspiciously by the DP department, however, and are taken as an indication that the user once again does not know what is really wanted in the system.

Rather than producing a mass of paper to describe what and how a system can perform, prototyping allows the user to "see" the proposed system and gain hands-on experience. In this way, the user can easily determine if all features are available and if the operations proposals are practical. Changes required by the user as a result of viewing the prototype can be easily and quickly made using the prototyping technology. Therefore, written documentation presented to the user is confined to operational procedures and any other required information that is not immediately apparent from the prototype (e.g., cost/benefit analysis, interaction with manual procedures).

Contentious Review Process

The review process built into the methodology is usually one of contention between the user and the DP department over issues requiring further work by the DP department. The DP department, however, is typically geared to move immediately to the next phase. One of two possibilities is likely:

- The development is delayed inordinately while the outstanding issues are resolved.
- The DP department continues the development as if the outstanding issues did not exist and presumes that when these issues are resolved, the solution will be incorporated.

The review process is greatly simplified with a prototype. The user has the opportunity throughout the evolution of the prototype to see the system as it is being built and can thus make changes accordingly. The DP department can accommodate user changes because of the flexibility of the prototyping technology and the fact that the specifications and coding associated with the prototyping are the foundation of the production system. The previous problems of having to revise several levels of specifications and then restructure any programming already begun are no longer significant.

Mistaken Priorities

In many installations, adherence to the development methodology begins to be more important than the development itself. Understanding the requirements and intricacies of the methodology becomes as important (or more so) as knowledge of the business needs to be satisfied by the system or the technical skills necessary to develop the system. Failure to develop the system according to user requirements on time and within budget is viewed as the fault of the methodology and not of the development team.

Prototyping strongly emphasizes developmental aspects as opposed to specification aspects. With prototyping, it is not enough to follow the rules of the development methodology and produce the required documentation. A working prototype that satisfies user needs and can be expanded and refined to full production status must be produced.

System Maintenance

The development methodologies do not address the problems of maintaining existing systems but are still a matter of negotiation between the user and the DP department. Typically, fourth-generation programming languages require very few statements for multiple selection, manipulation, and presentation of data. Combined with the standardization and control inherent in a data dictionary and the flexibility of a relational data base, maintenance of a system developed using prototyping technology is substantially simplified.

DEVELOPMENTAL ENVIRONMENT

The environment presented here concerns the development of a prototype for an online, real-time system for the settlement of securities (i.e., stocks and bonds). The components and functions of this environment are described as follows.

Information to Be Processed

Information regarding ledger balance and deliveries (i.e., transactions) that must be processed is available from an existing batch system.

Ledger Balances. Each user of the system has numerous ledger balances and is identifiable by a unique number. Each ledger balance represents the quantity of stocks or bonds held by the user for that particular security. Each security is also identifiable by a unique number.

Deliveries. A user of the system may have numerous transactions that indicate the user is to deliver securities to another user of the system. The delivering participant is credited with a cash payment in return for the delivered securities.

Functions to Be Performed

Ledger inquiry, invoking deliveries, and creating deliveries are the functions to be performed.

Ledger Inquiry. All users of the system should have access to their ledger positions either by the use of specified security number or by starting at the first security for a user and then paging sequentially through all the user's ledger balances.

Invoking Deliveries. Users should be able to access all their deliveries and invoke selected deliveries. These deliveries can be accessed either with a specified security number or by starting at the first security number for a user for which there are delivery transactions and then paging sequentially through all subsequent securities with associated delivery transactions. When an individual delivery is invoked, the system reduces the user's ledger balance by the quantity of the delivery and increases the receiving participant's ledger balance accordingly. Cash amounts are credited to the delivering user and debited from the receiving user. If the delivering user has an insufficient ledger balance to satisfy the quantity on an invoked delivery, the delivery is rejected.

Creating Deliveries. The details on delivery transactions usually are confirmed as correct by both parties; this is done in the existing batch system. Therefore, the deliveries that appear online are confirmed as correct before the user is able to invoke them. A system requirement, however, allows a user to input a delivery transaction online. Before this delivery can be added to the list of delivery transactions supplied by the batch system, it must be approved by the receiving user, thus requiring an online approval function. In addition, the delivering user (i.e., the person inputting the delivery) must be able to determine the status of these created deliveries (i.e., whether confirmed, rejected, or still awaiting action).

Development Methodology

The systems development methodology in effect at the time of this prototype's development was structured as shown in Table 1.

Development System

The development system was provided through the use of hardware and software designed for the development and operation of high-volume, online transaction processing systems. Tandem Computer products were chosen for the development of the application system that forms this case study prototype. Tandem uses a modular approach in constructing its development environment. The basis of this approach is Tandem's provision of the software inherent in the development of online systems (e.g., terminal access, telecommunications protocols); also included is a framework within which application code can easily be written and integrated to form a complete online application system. The software supplied by Tandem revolves around its PATHWAY™ transaction processing system and the data definition language.

Table 1. Systems Development Methodology Used in This Case Study

Phase	Purpose
Project definition	To document a business or a system problem to a level at which management can decide on a strategy to remedy it
General design and feasibility	To prepare a high-level design of a system solution to an identified business or system problem and present a case for adopting such a solution
Detailed design	To expand the general design of an approved system solution to the point at which programming and procedure writing can begin
Program and procedure development	To develop and test all computer programs and manual procedures (i.e., develop the "total" system)
Implementation	To ensure that the system meets operational requirements and is smoothly migrated into the production environment

PATHWAY Transaction Processing System

PATHWAY consolidates the various software components necessary to execute the application system, thereby providing the system with a global view of the processing environment (e.g., valid transactions, the terminals at which these transactions may be originated, the data base to be accessed by these transactions). The software components of primary interest to the systems developer are discussed in the following subsections.

Terminal Control Process (TCP). TCP is responsible for controlling all physical terminal input/output (I/O) and for performing four major application functions: terminal interface, input field validation, output data mapping, and transaction control. The TCP software is supplied as part of PATHWAY.

Screen COBOL (SCOBOL). In order to perform the applications functions described above, the TCP is associated with a SCOBOL program. With SCOBOL, the systems developer defines the screen formats, input and output data mapping, data validation and consistency checks, transaction routing to server programs, and overall application control. The TCP dynamically executes the code generated by the SCOBOL compiler.

Although not specifically marketed by Tandem as such, SCOBOL has all the attributes of a fourth generation language in that it contains a powerful instruction set allowing multiple processing to be defined with very few statements. SCOBOL is also interpretive in that it is dynamically compiled and executed by the TCP at run time.

Server Programs. Server programs provide access to the data base (one server program per flat file within the data base). A server program can perform a maximum of four functions: read, update, add, and delete. Routing to the server programs is from a SCOBOL program via the TCP. The server program is presented with a transaction indicating the function to be performed. It performs the required function and replies to the TCP, which returns control to the SCOBOL program for further processing. Control over the online configuration monitor (PATHCOM™) and the online system monitor (PATHMON™) is supplied as part of the PATHWAY product. Data base management functions are performed by Tandem's ENSCRIBE™, which forms an integral part of the operating system.

As should be apparent from this description, the systems developer had to write only the SCOBOL and server programs. The vendor supplied all other required software. The SCOBOL programs are written in a high-level, powerful programming language, whereas the functions of the server programs are limited, making them easy to write and test. The flexibility and ease of use of this modular environment are ideal for prototyping.

Data Definition Language

The DDL provides three functions to the system developer as follows.

Data Dictionary Creation. The data dictionary is a complete, centralized description of the data structures, records, and files that make up the data base. The DDL provides definition and record statements to create the data dictionary. The definition statement is a data description independent of any file or record. Groups and fields are specified through the definition statement. The record statement specifies record structures using the data groups and fields of a previous definition statement. Record statements also define the attributes of the files that will store these data base records. When a record statement refers to a definition, the groups and fields of the definition are inserted into the record statement at the point at which the definition name is written.

Data Base Build Commands. The DDL can generate the commands used by the data base create utility to build the actual data base files with the specified attributes. This utility can then be used to load the data base from external files as needed.

"Copybook" Source Code. The DDL compiler provides for "Copybook" source code, which can be directly incorporated into the application program code (both SCOBOL and server programs), thereby ensuring consistency between the program and the data base. Syntax checking is performed to ensure that names of groups or fields adhere to the rules for the programming language in the application program.

IMPACT ON DEVELOPMENT

Each phase of the systems development methodology was complied with. The prototype expansion is described through the various phases of the methodology to the point at which the system was placed in production. The impact on each development phase caused by using the prototyping approach is described in the following sections.

Project Definition

The project was initially part of a long-term corporate development plan. A project leader was assigned to develop the objectives and scope of the project. Management was presented with a short paper describing the functions to be developed (as previously defined in this portfolio) and the approximate cost, resources, and time frame for the development. Management approved the proposal submitted by the project leader and allocated a budget and project team consisting of one business analyst and two programmers for the next phase.

General Design and Feasibility

The project leader and business analyst discussed with the primary user the functionality of the system. These discussions made possible the design of the various files and records to make up the data base; it was also possible to complete the initial screen layouts.

The programmers were given the screen layouts, file, and record descriptions as well as a brief description of the system's functional capabilities. With this information, they were able to use the DDL to define all fields, records, and files to the data dictionary and generate the appropriate commands to load the data base.

The programmers then wrote a server program for each file within the data base. The data base contained a file for the ledger, a file for delivery transactions, a cross-reference file between user numbers and their names and corporate titles, and a cross-reference file between security (i.e., stocks, bonds) numbers and their designated descriptions. The server programs were able to access a file in order to read, update, add, or delete.

Using SCOBOL, the programmers wrote the code necessary to demonstrate the functionality of the proposed system. Figure 1 shows the configuration of the development environment for this prototype, since the purpose of this initial prototype was to demonstrate functionality. Only the most common, correct conditions were coded. Meanwhile, the business analyst prepared a system outline for the user, briefly describing the contents of each screen and the functionality available. The business analyst also prepared a formalized presentation to demonstrate the prototype to the user.

When the prototype was ready, the user was invited to view it. The first part of this exercise was the formal presentation by the business analyst. Thereafter, the user was allowed to sit at the terminal and experiment with the prototype. Comments from the user were written down by the business analyst and passed to the programmers, who could quickly change the prototype; the business analyst's system outline was correspondingly altered. The review by the user continued until agreement was reached on the functionality and usability of the system. The revised prototype was presented to management along with the revised cost, resources, and time frame for the remaining phases of development. Approval was given to proceed.

Detailed Design

The detailed design phase consisted of defining the remaining logic conditions (including all error conditions) that were to be incorporated into the

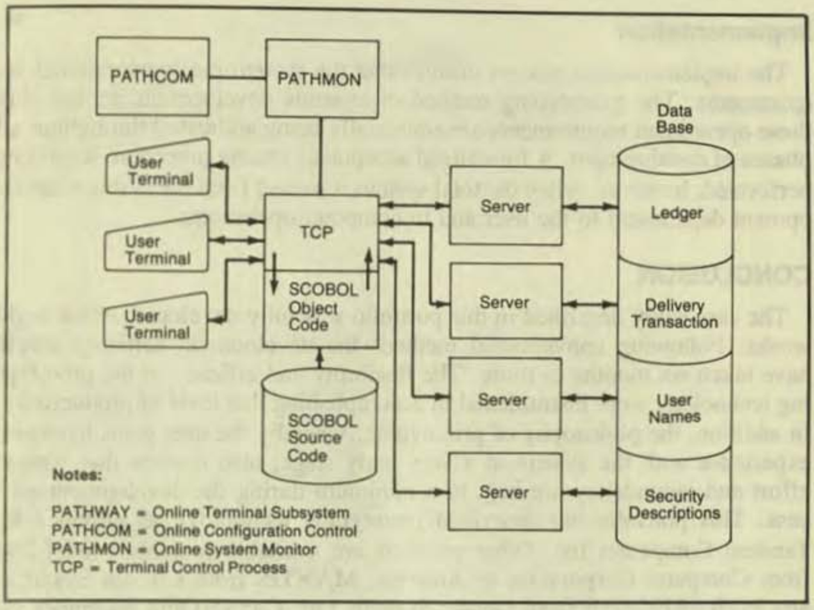


Figure 1. The Development Configuration for the Case Study

prototype. Network planning, backup and recovery, the security, auditability and control aspects, and the interface to the batch systems were fully defined and documented. Although each of these factors must be considered in the development of an initial prototype, the various problems and solutions do not require complete analysis and documentation until the system's functionality has been defined and approved by the user. This analysis and documentation, however, must be completed during the detailed design phase of development. In fact, prototyping technology has already addressed many of these factors.

Management was presented with the detailed design document, along with the revised cost, resources, and time frame for the remainder of the project. From a user standpoint, the only particularly interesting output from the detailed design phase was the cost and time frame for what still had to be done. The user is primarily concerned with the system's functionality, and this had already been defined and approved.

Program and Procedure Development

The prototype was expanded to incorporate those features defined in the detailed design. Batch interface programs were written. Procedural development consisted primarily of incorporating an online help function available to the user at all times.

As the programmers completed testing of individual pieces of code, the users were invited to try the system again. Any amendments were discussed with the business analyst, documented, and quickly incorporated. The flexibility of the technological tools associated with prototyping was very much in evidence throughout this phase.

Implementation

The implementation process ensures that the system meets operational requirements. The prototyping method of systems development implies that these operational requirements are continually being addressed throughout all phases of development. A formalized acceptance testing procedure should be performed, however, when the total system is passed from the systems development department to the user and to computer operations.

CONCLUSION

The case study described in this portfolio was fully developed within eight weeks. Following conventional methods the development certainly would have taken six months or more. The flexibility and efficacy of the prototyping technology were instrumental in accomplishing this level of productivity. In addition, the philosophy of prototyping, whereby the user gains hands-on experience with the system at a very early stage, also ensures that wasted effort and redundancy are kept to a minimum during the development process. This portfolio has described prototyping technology as provided by Tandem Computers Inc. Other products are available (notably Model 204 from Computer Corporation of America, MANTIS from Cincom Systems, and POWERHOUSE from Quasar Systems Ltd, Canada) and run under assorted hardware and operating systems; these products also provide the benefits of prototyping as described here.

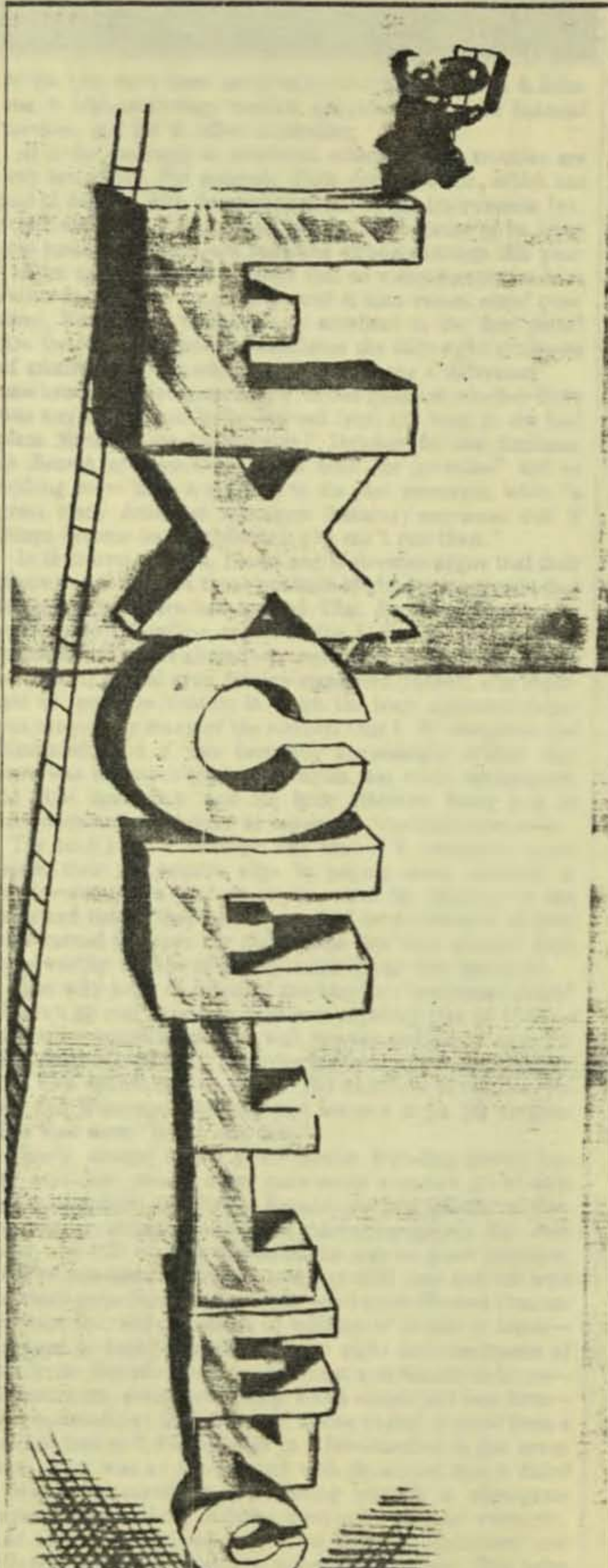
This portfolio was written by Ian A. Gilhooley, Manager, Business Development Department, The Canadian Depository for Securities Ltd, Toronto. Technical assistance was provided by David Clack, Regional Specialist, Tandem Computers Inc.

Bibliography

ACM Rapid Prototyping Conference. *SIGSOFT Notes*. Columbia MD, December 1982.

Martin, James, *Applications Development without Programmers*. Englewood Cliffs NJ: Prentice-Hall Inc, 1982.

NOTES



ILLUSTRATIONS BY MICHAEL WITTE

Cover Story

WHO'S EXCELLENT NOW?

SOME OF THE BEST-SELLER'S PICKS HAVEN'T BEEN DOING SO WELL LATELY

Not long after *In Search of Excellence* zoomed to the top of the best-seller list, co-author Thomas J. Peters gave a speech to a division of Hewlett-Packard Co., one of the star companies in the book. After the speech, as Peters recalls, the division's manager told him: " 'What we should do is call you in to give a speech once a quarter. So we can remember what it was that we were when we were really a great company. And to remind us how damned hard it is to maintain some of those traits once you get big.' "

Judging from Hewlett-Packard's current difficulties, the manager knew what he was talking about. The turmoil and product-development problems plaguing HP—the third-largest computer maker after International Business Machines Corp. and Digital Equipment Corp.—hardly make it look like one of America's most innovative, best-run companies. Although its earnings are still strong, HP has stumbled badly in the critical microcomputer and superminicomputer markets.

DISENCHANTMENT. To regain its stride, HP is being forced to abandon attributes of excellence for which it was praised. Its technology-driven, engineering-oriented culture, in which decentralization and innovation were a religion and entrepreneurs were the gods, is giving way to a marketing culture and growing centralization. The continuing exodus of disenchanted managers—12 have left in just the last six months—tells the story. "The time spent in coordinating meetings has increased by an order of magnitude in the last four years," sighs André Schwager, a former HP general manager who left in September. "It's clear that the culture is beginning to change."

Hewlett-Packard is not the only "excellent company" that is not looking so excellent these days. According to studies by BUSINESS WEEK, management consultants McKinsey & Co., and Standard & Poor's Compustat Services Inc., at least 14 of the 43 "excellent" companies highlighted by Peters and co-author Robert H. Waterman Jr. in their book just two years ago have lost their luster (table, page 78).

If judged on their performance during the last decade, Delta Air Lines, Walt Disney Productions, Eastman Kodak, and Texas Instruments would not pass the financial tests for excellence laid down in the book. In more recent years, nine others—Atari, Avon, Caterpillar Tractor (page 91), Chesebrough-Pond's, DEC, Fluor, Levi Strauss, Revlon, and Dart & Kraft's Tupperware International—have suffered significant earnings declines that stem from serious business problems, management problems, or both. While most outsiders still view them as well-managed and in robust financial health, other members of the elite have been humbled by blunders: Johnson & Johnson in high-technology medical equipment, Dana in financial services, and 3M in office automation.

It is far too early to determine whether these troubles are only temporary. For example, Delta Air Lines Inc., which has had to contend with deregulation, and Texas Instruments Inc., which sustained a staggering loss in 1983 because of its foray into home computers, are reporting strong earnings this year.

Even so, it comes as a shock that so many companies have fallen from grace so quickly—and it also raises some questions. Were these companies so excellent in the first place? Are the eight attributes of excellence the only eight attributes of excellence? Does adhering to them make a difference?

NEW LESSONS? Not surprisingly, critics question whether there was any new lesson to be learned from the book in the first place. Management writer Peter F. Drucker, for one, dismisses *In Search of Excellence* as "a book for juveniles" and as nothing more than a reaction to the last recession, when "a great many American managers [became] convinced that if things become too complicated, you can't run them."

In their own defense, Peters and Waterman argue that their intent was to address those qualities of good management that too many managers had ignored. That *Excellence* has so far sold nearly 2.8 million copies in the U.S.—and hundreds of thousands of copies abroad—proves that it struck a responsive nerve in U.S. and even foreign managers. Indeed, it is important to recall the context in which the book appeared: Japan was conquering many of the markets that U.S. companies had dominated. And it was becoming increasingly evident that there was too much analysis-paralysis, too much bureaucracy, too little innovation, and too little attention being paid to customers and employees at too many American companies.

The book's basic message was that U.S. companies could regain their competitive edge by paying more attention to people—customers and employees—and by sticking to the skills and values they know best. And when virtually all eyes were turned to Japan for the answer, the book showed there were worthy models of management in our own backyard.

Then why have so many of the excellent companies fallen? "There's no real reason to have ever expected that all of these companies would have done well forever and ever," says Peters. Adds Waterman: "If you're big, you've got the seeds of your own destruction in there." The excellent companies, Peters and Waterman contend, just seemed to be big corporations that were "losing less fast."

Clearly, several of the 43 companies, including Revlon Inc. and Atari Inc., should never have made anyone's list of well-managed companies. Charles Revson, the late founder of Revlon, was an entrepreneurial and marketing genius. But even those who still worship him admit he was no great manager.

LOVE OF NUMBERS. Atari, the company that rose and fell with the video-game boom and collapse—and stuck Warner Communications Inc. with hundreds of millions of dollars in losses—managed to break almost all of the eight commandments of excellence. Out-of-control management and bloated fiefdoms—not autonomy, entrepreneurship, and a simple and lean form—were hallmarks at that company, whose payroll zoomed from a few hundred to 7,000 and back to a few hundred in just seven years. Atari was so out of touch with its market that it failed to realize its customers were losing interest in video-game players and switching to home computers—a fatal oversight.

At several companies, a love for product, customers, and entrepreneurship gave way to a love for numbers. This helps explain why Chesebrough-Pond's Inc. stalled in 1981, when sales in its core health and beauty businesses slowed and some of its vaunted acquisitions—notably Bass shoes and Prince tennis rackets—ran into trouble. This was also the case at

THE EIGHT ATTRIBUTES OF EXCELLENCE

1. **BIAS FOR ACTION:** A preference for doing something—anything—rather than sending an idea through endless cycles of analyses and committee reports
2. **STAYING CLOSE TO THE CUSTOMER:** Learning his preferences and catering to them
3. **AUTONOMY AND ENTREPRENEURSHIP:** Breaking the corporation into small companies and encouraging them to think independently and competitively
4. **PRODUCTIVITY THROUGH PEOPLE:** Creating in all employees the awareness that their best efforts are essential and that they will share in the rewards of the company's success
5. **HANDS-ON, VALUE-DRIVEN:** Insisting that executives keep in touch with the firm's essential business and promote a strong corporate culture
6. **STICK TO THE KNITTING:** Remaining with the businesses the company knows best
7. **SIMPLE FORM, LEAN STAFF:** Few administrative layers, few people at the upper levels
8. **SIMULTANEOUS LOOSE-TIGHT PROPERTIES:** Fostering a climate where there is dedication to the central values of the company combined with tolerance for all employees who accept those values

THE 'EXCELLENT COMPANIES' CITED BY PETERS AND WATERMAN

Allen-Bradley	International Business Machines
Amdahl	Johnson & Johnson
Atari	K mart
Avon Products	Levi Strauss
Bechtel Group	Marriott
Boeing	M&M Mars
Bristol-Myers	Maytag
Caterpillar Tractor	McDonald's
Chesebrough-Pond's	Merck
Dana	National Semiconductor
Data General	Procter & Gamble
Delta Air Lines	Raychem
Digital Equipment	Revlon
Dow Chemical	Schlumberger
Du Pont	Standard Oil (Indiana)
Eastman Kodak	Texas Instruments
Emerson Electric	3M
Fluor	Tupperware International
Frito-Lay	Wal-Mart Stores
Hewlett-Packard	Walt Disney Productions
Hughes Aircraft	Wang Laboratories
Intel	

DATA: *IN SEARCH OF EXCELLENCE*

of the elite have been humbled by blunders: Johnson & Johnson in high-technology medical equipment, Dana in financial services, and 3M in office automation.

It is far too early to determine whether these troubles are only temporary. For example, Delta Air Lines Inc., which has had to contend with deregulation, and Texas Instruments Inc., which sustained a staggering loss in 1983 because of its foray into home computers, are reporting strong earnings this year.

Even so, it comes as a shock that so many companies have fallen from grace so quickly—and it also raises some questions. Were these companies so excellent in the first place? Are the eight attributes of excellence the only eight attributes of excellence? Does adhering to them make a difference?

NEW LESSONS? Not surprisingly, critics question whether there was any new lesson to be learned from the book in the first place. Management writer Peter F. Drucker, for one, dismisses *In Search of Excellence* as "a book for juveniles" and as nothing more than a reaction to the last recession, when "a great many American managers [became] convinced that if things become too complicated, you can't run them."

In their own defense, Peters and Waterman argue that their intent was to address those qualities of good management that too many managers had ignored. That *Excellence* has so far sold nearly 2.8 million copies in the U.S.—and hundreds of thousands of copies abroad—proves that it struck a responsive nerve in U.S. and even foreign managers. Indeed, it is important to recall the context in which the book appeared: Japan was conquering many of the markets that U.S. companies had dominated. And it was becoming increasingly evident that there was too much analysis-paralysis, too much bureaucracy, too little innovation, and too little attention being paid to customers and employees at too many American companies.

The book's basic message was that U.S. companies could regain their competitive edge by paying more attention to people—customers and employees—and by sticking to the skills and values they know best. And when virtually all eyes were turned to Japan for the answer, the book showed there were worthy models of management in our own backyard.

Then why have so many of the excellent companies fallen? "There's no real reason to have ever expected that all of these companies would have done well forever and ever," says Peters. Adds Waterman: "If you're big, you've got the seeds of your own destruction in there." The excellent companies, Peters and Waterman contend, just seemed to be big corporations that were "losing less fast."

Clearly, several of the 43 companies, including Revlon Inc. and Atari Inc., should never have made anyone's list of well-managed companies. Charles Revson, the late founder of Revlon, was an entrepreneurial and marketing genius. But even those who still worship him admit he was no great manager.

LOVE OF NUMBERS. Atari, the company that rose and fell with the video-game boom and collapse—and stuck Warner Communications Inc. with hundreds of millions of dollars in losses—managed to break almost all of the eight commandments of excellence. Out-of-control management and bloated fiefdoms—not autonomy, entrepreneurship, and a simple and lean form—were hallmarks at that company, whose payroll zoomed from a few hundred to 7,000 and back to a few hundred in just seven years. Atari was so out of touch with its market that it failed to realize its customers were losing interest in video-game players and switching to home computers—a fatal oversight.

At several companies, a love for product, customers, and entrepreneurship gave way to a love for numbers. This helps explain why Chesebrough-Pond's Inc. stalled in 1981, when sales in its core health and beauty businesses slowed and some of its vaunted acquisitions—notably Bass shoes and Prince tennis rackets—ran into trouble. This was also the case at

THE EIGHT ATTRIBUTES OF EXCELLENCE

1. **BIAS FOR ACTION:** A preference for doing something—anything—rather than sending an idea through endless cycles of analyses and committee reports
2. **STAYING CLOSE TO THE CUSTOMER:** Learning his preferences and catering to them
3. **AUTONOMY AND ENTREPRENEURSHIP:** Breaking the corporation into small companies and encouraging them to think independently and competitively
4. **PRODUCTIVITY THROUGH PEOPLE:** Creating in all employees the awareness that their best efforts are essential and that they will share in the rewards of the company's success
5. **HANDS-ON, VALUE-DRIVEN:** Insisting that executives keep in touch with the firm's essential business and promote a strong corporate culture
6. **STICK TO THE KNITTING:** Remaining with the businesses the company knows best
7. **SIMPLE FORM, LEAN STAFF:** Few administrative layers, few people at the upper levels
8. **SIMULTANEOUS LOOSE-TIGHT PROPERTIES:** Fostering a climate where there is dedication to the central values of the company combined with tolerance for all employees who accept those values

THE 'EXCELLENT COMPANIES' CITED BY PETERS AND WATERMAN

Allen-Bradley	International Business Machines
Amdahl	Johnson & Johnson
Atari	K mart
Avon Products	Levi Strauss
Bechtel Group	Marriott
Boeing	M&M Mars
Bristol-Myers	Maytag
Caterpillar Tractor	McDonald's
Chesebrough-Pond's	Merck
Dana	National Semiconductor
Data General	Procter & Gamble
Delta Air Lines	Raychem
Digital Equipment	Revlon
Dow Chemical	Schlumberger
Du Pont	Standard Oil (Indiana)
Eastman Kodak	Texas Instruments
Emerson Electric	3M
Fluor	Tupperware International
Frito-Lay	Wal-Mart Stores
Hewlett-Packard	Walt Disney Productions
Hughes Aircraft	Wang Laboratories
Intel	

DATA: *IN SEARCH OF EXCELLENCE*

Revlon under Michel C. Bergerac. The ITT Corp. alumnus installed some of the so-called modern management techniques that Revlon needed. But former executives and industry experts say an overemphasis on numbers has dulled much of Revlon's marketing pizazz.

Peters argues that it is virtually impossible to score a perfect 10 on all eight attributes of excellence. Several excellent companies that fell by the wayside overstressed some attributes and ignored others. Disney Productions' employees were so devoted to the clean-taste values established by its founder that it lost its creative flair and failed to respond to changes in moviegoers' tastes. Employees were so wedded to the legacy that many protested the making of *Splash*, one of the company's first efforts in recent years to make a truly contemporary film—and one that turned out to be a hit.

In most instances, the transgressors ran amok by walking away from the principles that had been key to their earlier successes. A slew of companies—TI, Revlon, Fluor, Avon, Johnson & Johnson, Dana, and 3M—did not "stick to their knitting" and are paying the price. Fluor Corp. made the mistake of paying a staggering $2.3 billion for St. Joe Minerals Corp. in 1981—right before metals prices collapsed. Instead of helping Fluor ride out the rough times in its mainstay engineering-construction business, St. Joe has added to Fluor's financial woes. Its earnings plummeted to $27.7 million last year from $158.9 million in 1981. Its stock, which was trading at 71 right before its acquisition, is now hovering around 18.

WHEN STRICTNESS HURTS. In the *Harvard Business Review*, consultant Daniel T. Carroll attacked *Excellence* for ignoring the importance of such factors as proprietary technology, government policy, and national culture. The BUSINESS WEEK and McKinsey studies suggest that the criticism is well-founded. Of the 14 excellent companies that had stumbled, 12 were inept in adapting to a fundamental change in their markets. Their experiences show that strict adherence to the eight commandments—which do not emphasize reacting to broad economic and business trends—may actually hurt a company.

For example, Delta Air Lines, which had flourished by maintaining a low debt and exploiting a close-knit culture to keep costs low, failed to see that deregulation had changed its world. The Atlanta-based carrier was slow in recognizing the importance of computers to keep tabs on ticket prices in different markets. Consequently, Delta first failed to meet competitors' lower prices. Then it overreacted. The result: an $86.7 million loss in its fiscal year ended June, 1983, and a brand-new computer system.

Staying close to the customer can backfire on a company when a market shifts dramatically, leaving the company close to the wrong customer. This is what happened to Avon Products Inc. and to Dart & Kraft's Tupperware unit when the housewives to whom they catered began to pursue careers. Similarly, DEC and HP—companies run by engineers for customers who are engineers—have stumbled in trying to sell to customers without a technical background.

FRUSTRATION. Hewlett-Packard's famed innovative culture and decentralization spawned such enormously successful products as its 3000 minicomputer, the handheld scientific calculator, and the new ThinkJet nonimpact printer. But when a new climate required its fiercely autonomous divisions to cooperate in product development and marketing, HP's passionate devotion to the "autonomy and entrepreneurship" that Peters and Waterman advocate became a hindrance.

To its astonishment, HP found itself frustrated in trying to move into such new high-growth, high-tech markets as superminicomputers, engineering work stations, personal computers, and office automation. Two years after its rollout, the HP 9000 work station that was developed in a $100 million crash effort still lacks competitive software, and HP has been outstripped by a crowd of startups. HP's response: centralizing.

One major lesson from all this is that the excellent companies of today will not necessarily be the excellent companies of tomorrow. But the more important lesson is that good management requires much more than following any one set of rules. *In Search of Excellence* was a response to an era when management put too much emphasis on number-crunching. But companies can also get into trouble by overemphasizing Peters' and Waterman's principles. Says Waterman: "The book has been so popular that people have taken it as a formula for success rather than what it was intended to be. We were writing about the art, not the science of management."

WHY SOME OF THE 'EXCELLENT' COMPANIES HAVE STUMBLED

The commandments of excellence they broke

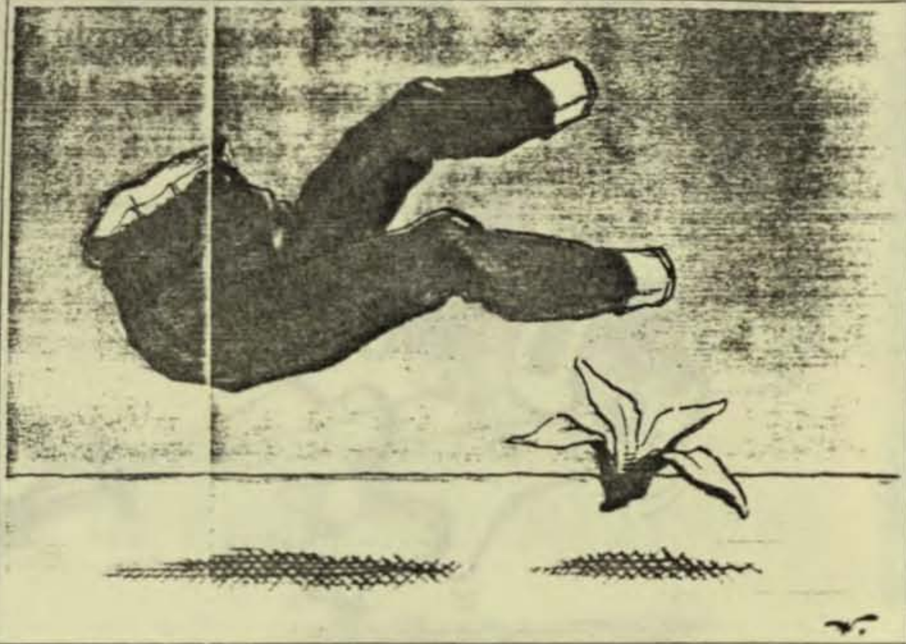
	BIAS FOR ACTION	STAYING CLOSE TO THE CUSTOMER	AUTONOMY AND ENTREPRENEURSHIP	PRODUCTIVITY THROUGH PEOPLE	HANDS-ON, VALUE-DRIVEN	STICK TO THE KNITTING	SIMPLE FORM, LEAN STAFF	SIMULTANEOUS LOOSE-TIGHT PROPERTIES
[illegible]	✓	✓	✓	✓	✓		✓	✓
AVON PRODUCTS (c)		✓		✓		✓		
CATERPILLAR TRACTOR (c)	✓						✓	
CHESEBROUGH-POND'S (c)	✓				✓	✓		
DELTA AIR LINES (a)(c)		✓						
DIGITAL EQUIPMENT (c)	✓		✓				✓	
DISNEY PRODUCTIONS (b)(c)		✓	✓					
EASTMAN KODAK (a)			✓				✓	
FLUOR (c)						✓		
HEWLETT-PACKARD (c)								
LEVI STRAUSS (c)		✓					✓	
REVLON (c)	✓	✓	✓	✓	✓	✓		
TUPPERWARE (b)(c)		✓		✓				
TEXAS INSTRUMENTS (c)	✓	✓	✓			✓	✓	✓

(a) Did not pass Peters' and Waterman's financial-criteria hurdle in 1974-83 (b) Not tested against financial criteria because they were not included in Compustat's data base
(c) Difficulty adapting to fundamental change in market

DATA: BW, STANDARD & POOR'S COMPUSTAT SERVICES INC., McKINSEY & CO.

LEVI'S: THE JEANS GIANT SLIPPED AS THE MARKET SHIFTED

Levi Strauss & Co. developed and built its jeans business by operating as close to its customers as the pockets on a pair of its 501 button-fly jeans. So close, in fact, that a former company president, on returning from a camping trip, ordered that a rivet be removed from the basic jeans line because he and fellow campers had discovered that the tiny fastener burned them as they crouched around the fire.



Unfortunately for Levi's, that kind of sensitivity to customers was lost in the midst of the jeans industry's heady growth in the 1970s. *In Search of Excellence* lauds Levi's as one of the excellent few that are "pushed around by their customers, and they love it." But that analysis now proves to have been sadly out of date. By the early 1980s, Levi's was grappling with diversification, marketing, and fashion dilemmas. Its confusion translated into disappointing performance and retrenchment for the nation's largest garment maker.

INWARD FOCUS. A family-run business for all of its 134 years, Levi's built its reputation on two basic principles: adherence to quality and a one-big-happy-family corporate culture. As the company grew, its managers concentrated on manufacturing enough jeans to supply a seemingly insatiable market. But that inward focus left Levi's unequipped to deal with a slowdown in jeans sales growth and the market's shift toward more fashionable apparel. "For so long, we were [always] sold out. Our time was prioritized on getting more product, new factories, more raw materials. We were internally oriented," notes one executive. The main casualty of Levi's inward focus was the customer. "We let that relationship with our retailers fall into a sad state of disrepair," admits one Levi's insider. As the jeans industry grew 15% annually for most of the past 20 years, retailers could sell every pair of jeans they received from Levi's. That "created tensions in the relationship and unfortunate habits on the part of some of our people," admits President Robert D. Haas, the great-great-grandnephew of founder Levi Strauss.

Retailers saw Levi's as aloof and inflexible. The company spent little on joint local advertising campaigns and did not support in-store promotions. When jeans demand slackened, Levi's lacked a loyal retail customer base from which to launch new products and stave off competitors who were providing better service, such as VF Corp. with its Lee brand and Blue Bell Inc. and its Wrangler line.

In losing touch with its customers, Levi's also distanced itself from the fashion trends of the 1980s and failed to anticipate the consumer shift away from basic goods. Says Robert L. Pugmire, general manager of Seattle-based Bernie's Menswear Inc., a 40-store chain that carries Levi's products: "In fashion, if you have to change your entire business overnight, you do it. Levi's can't react that fast. That's why it's important to have relationships with people who can see the changes coming."

Levi's suffered several false starts when it belatedly tried to broaden its product base. In 1979 the company positioned its David Hunter line against famous designer labels such as Ralph Lauren's Polo brand. But Levi's advertising program failed to convince customers that clothing with the Levi's label was comparable to the designer lines. For the same reason, its Activewear line of sports clothing has failed to challenge the likes of Nike Inc. and Adidas-Sportschuhfabriken. And one attempt to penetrate the European market with lighter-weight jeans failed because consumers wanted the heavy-duty American variety after all.

After a three-year earnings slide beginning in 1980, Levi's rebounded with the economy in 1983, with earnings rising 54% on sales of $2.7 billion. Most of that rise, however, was a result of retailers' restocking barren inventories and Levi's filling the pipeline for Sears, Roebuck & Co. and J.C. Penney Co.'s stores, two new customers. Some retailers, angry that Levi's had turned to such mass merchandisers, dropped the Levi's line entirely. And consumers suddenly decided they wanted more fashionable sportswear, turning away from such old-time favorites as blue jeans. The result: Levi's sales for the first nine months of 1984 slipped 6%, and earnings plummeted 72%.

In response, Haas, who took over as president in April, is moving to end the paralysis that caused Levi's to lose touch with customers. He has expanded Levi's local advertising budget and its participation in special promotions. Levi's co-sponsored a local track meet with Seattle's Bernie's—a marketing gambit that would have been unheard-of a few years ago, says retailer Pugmire.

MORE RESPONSIVE. Levi's is also offering retailers volume discounts for the first time. And the company will exchange unsold goods for other products. In the past, retailers were stuck with anything they could not sell. These changes seem to be having an impact. A merchandise manager for Robinson's Department Stores in Los Angeles now calls Levi's one of his most responsive vendors.

Levi's efforts to improve its rapport with customers are crucial to what insiders see as its most significant management challenge: responding more quickly to shifts in the apparel market. Because the hot market today is higher-priced fashions made in relatively small quantities, "sticking to its knitting" as a jeansmaker is unlikely to serve as a cure for what ails Levi's. But Haas says the basic jeans and corduroy lines—which make up almost two-thirds of sales—will continue as the company's mainstays throughout the 1990s. Levi's, in fact, will spend $36 million advertising its basic 501 button-fly jeans this year.

The big question is: Can Levi's successfully pursue both the basic jeans and higher-fashion markets—something it has failed to achieve in the past? To do so will require a fundamental change. Asserts Alan L. Stein, director of corporate finance at Montgomery Securities and a Levi's director: "Levi's needs to become more a marketing company and less of a production company."

FEWER LAYERS. A critical test will be Haas's current effort to move into high-growth leisure and fashion goods. To facilitate that effort, Haas is creating autonomous units under the new Battery Street Enterprises (BSE) operating division. The division, which will make a push into fashion apparel, is so independent that employees' paychecks do not even bear the Levi's name. One indication of how much importance Haas is placing on such enterprises is the man he picked to be the BSE Fashion Group champion: James A. McDermott. McDermott, who formerly served as Levi's senior vice-president for marketing, is credited with developing a niche market for Levi's women's wear.

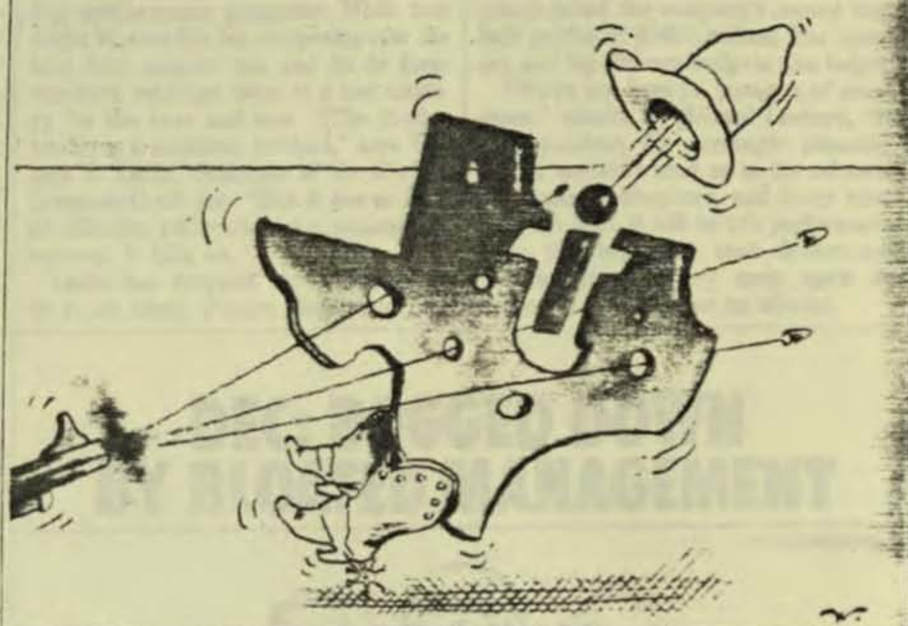
To respond quickly to fashion shifts, BSE has fewer layers of middle management than the jeanswear divisions. And McDermott boasts that his managers "are spending much more time with the retailer than in the past. We're going out with the salesmen and sitting with them... and we're listening, not just giving lip service."

Haas seems determined to make his jeanswear divisions more market-oriented and flexible, too. Sources say the company is searching extensively for top marketing talent to join a management dominated by operations men.

A simple form and lean staff are also companywide goals. In 1982, Levi's violated its proud tradition of lifetime employment and trimmed 2,400 people—including middle managers—from its payroll. And this year, Levi's cut its 38,000-member work force by an additional 12% and closed 19 plants. But in an attempt to preserve its legendary employee loyalty, Levi's offered generous severance and retraining benefits.

All this prompts McDermott, for one, to claim that Levi's has climbed back on the path toward excellence. "We've seen the error of our ways," he says.

TI: SHOT FULL OF HOLES AND TRYING TO RECOVER



Even at the time *In Search of Excellence* was written, its authors realized there were cracks in Texas Instruments Inc.'s armor. But because of the Dallas company's reputation for innovation and its 20-year financial record, they felt they had no choice but to include TI. Given the company's more recent record, however, co-author Thomas J. Peters acknowledges that today "we wouldn't have written about TI."

One could argue that the company has been breaking so many of the "eight commandments" of excellence for so long that even when the book was published in 1982, TI was not one of America's best-managed companies. The most dramatic evidence came last year, when TI folded its home computer business and accepted a $660 million operating loss and write-down. In 1983, TI suffered its first annual loss—$145.5 million on sales of $4.6 billion—in its 54-year history. But TI's home computer fiasco was only the most glaring symptom of a host of fundamental management problems that had taken root:

□ Its engineers, accustomed to industrial customers, lacked expertise in consumer markets. TI's foray into home computers—not to mention its earlier digital watch disaster—amply illustrates this. TI cut prices to create demand for its home computer. But the tactic, borrowed from its industrial chip markets, failed to generate a lasting interest. What TI thought was a revolution turned out to be only the hot Christmas gift of 1982.

□ The company believed it could dictate to its customers rather than listen to them. Indeed, the home computer sprang from management's desire to find a mass market for its newest microprocessor. Even when it became clear that the microprocessor was too costly for the market, TI refused to buy cheaper microprocessors from outsiders.

□ An overly complex management system—including matrix management and numbers-dominated strategic planning—tended to smother entrepreneurship. TI's confusing reporting structures, for instance, delayed the design and production of key new products like large-scale computer memory chips. And headquarters' demand that operations project such factors as floor space and manpower needs made TI's planning process more of a liability than a weapon.

□ The domineering styles of Chairman Mark Shepherd Jr. and President J. Fred Bucy often intimidated product managers, who told them what they wanted to hear—not what was really going on. For example, neither learned of the home computer problems until the company was drowning in inventory.

The TI of Shepherd and Bucy is remarkably different from the TI of Shepherd's legendary predecessor, Patrick E. Haggerty, who headed the company from 1945 to 1976. This was TI's golden age, during which decentralization gave rise to a flood of innovations. Under Haggerty, the company that started in oil-field seismology blossomed into the largest manufacturer of integrated circuits, a huge defense contractor, and a major power in minicomputers.

It was Haggerty who introduced strict financial controls and strategic planning to control TI's rapid growth and increasingly complex business mix. But Haggerty also championed TI's stable of entrepreneurs, understanding that people, not rigid systems, produce innovation.

Shepherd and Bucy had to learn that lesson the hard way. In 1982 they scrapped TI's matrix management system and returned control of products to their managers. Before then, semiconductor product managers had marketing responsibilities but lacked control over the labs that developed chips and the plants that made them. Now, memory chip manager and Senior Vice-President Timothy B. Smith, for instance, is responsible for research and development, manufacturing, and marketing. The result: TI's memory devices grabbed a leading market share and also command the industry's highest average price.

MORE PRAGMATIC. Efforts to rein in strategic planners' demands for ever more data also appear to be paying off. The 15 variables that product managers had to project out 10 years have been reduced to four. This helps explain TI's dramatic rebound in innovation: In the past 18 months the company introduced more new products than in any similar period in its history.

There are also signs that the not-invented-here syndrome that was fatal in home computers is giving way to pragmatism. TI recently initiated semiconductor ventures with Fujitsu Ltd. and National Semiconductor Corp. The company is even using archenemy Intel Corp.'s chips in its Professional Computer.

Moreover, Shepherd, 61, and Bucy, 56, are loosening their grip. They have expanded the membership and scope of TI's executive committee: It now includes more senior operating managers. Another significant move was the appointment of Bucy, who is considered a better delegator than Shepherd, to CEO in 1984. Shepherd, still chairman, now oversees only long-term strategy.

While TI's leaders have atoned for some of their sins, they are not totally repentant: They still cling to their strategy of applying their technological muscle to three key areas: semiconductors, computers, and consumer electronics. And they bristle at suggestions that TI's engineering culture and consumer markets do not mix. But TI's travails in personal computers tell a different story. Although the company has been successful in selling its Professional Computer through its own sales force to commercial markets, its effort to sell it through computer stores is foundering.

When TI launched the computer in 1983, it chose a design that could exchange data with International Business Machines Corp.'s Personal Computer but required customized software. The intent: to position the Professional as a high-performance computer. While that might be sensible for competing over the long haul against IBM and its de facto standard, retailers deem it a bad strategy for the here and now. "[The Professional] is a fantastic product," says William E. Ladin, chairman of the 55-store ComputerCraft Inc. "But it has to have all different software and a separate inventory. It kills us."

Ladin has dropped the product—and he is not alone. Future Computing Inc., a market research firm, estimates that only 13% of the nation's retail computer stores now carry the Professional—down from 20% in January. And, it adds, computer-store sales of the product have slipped from 1,500 per month to 650.

Nonetheless, Shepherd and Bucy have begun reviving TI's entrepreneurial drive. A reinvigorated semiconductor group paced the company's record first-half profits of $165.7 million. The recovery and big defense budgets also helped.

"We're not [now] a paragon of excellence," admits R. Michael Lockerd, TI's vice-president for strategic planning, "but I wouldn't send us to the minors." As Lockerd, Shepherd, and Bucy know only too well, it will be TI's performance over the long run that determines whether the company once again deserves to be picked as an all-star.

DEC: BOGGED DOWN BY BLOATED MANAGEMENT

The most popular name around the offices of Digital Equipment Corp. these days is Venus—the company's brand new superminicomputer, which packs twice the punch of its previous top-of-the-line model. Venus is expected to be a huge success with DEC's existing customers and should help revive the 27-year-old Maynard (Mass.) company's profit margin, which has plummeted to 7%, less than half of the 1981 level.

But Venus, scheduled for introduction on Oct. 31, may fail to work much magic in the broader marketplace because its unveiling is two years late. An addition to DEC's VAX family of computers, Venus is more powerful than the comparable machines of such key competitors as IBM, Data General, and Prime Computer. Its bang for the buck, however, hardly comes close to the supermins of such startups as Elxsi, a $30 million San Jose (Calif.) company that claims to have stolen half its customers from DEC. Says one industry expert: "Will Venus get DEC any new customers? I doubt it."

It is ironic that DEC looked to antiquity for the supermini's code name. In many ways, DEC's recent failure to measure up as an excellent company is the result of founder and President Kenneth H. Olsen's seeming preoccupation with an illustrious past. The company that pioneered the development of the minicomputer now seems incapable of adjusting to another new age in computers—one increasingly dominated by microcomputers and larger superminis.

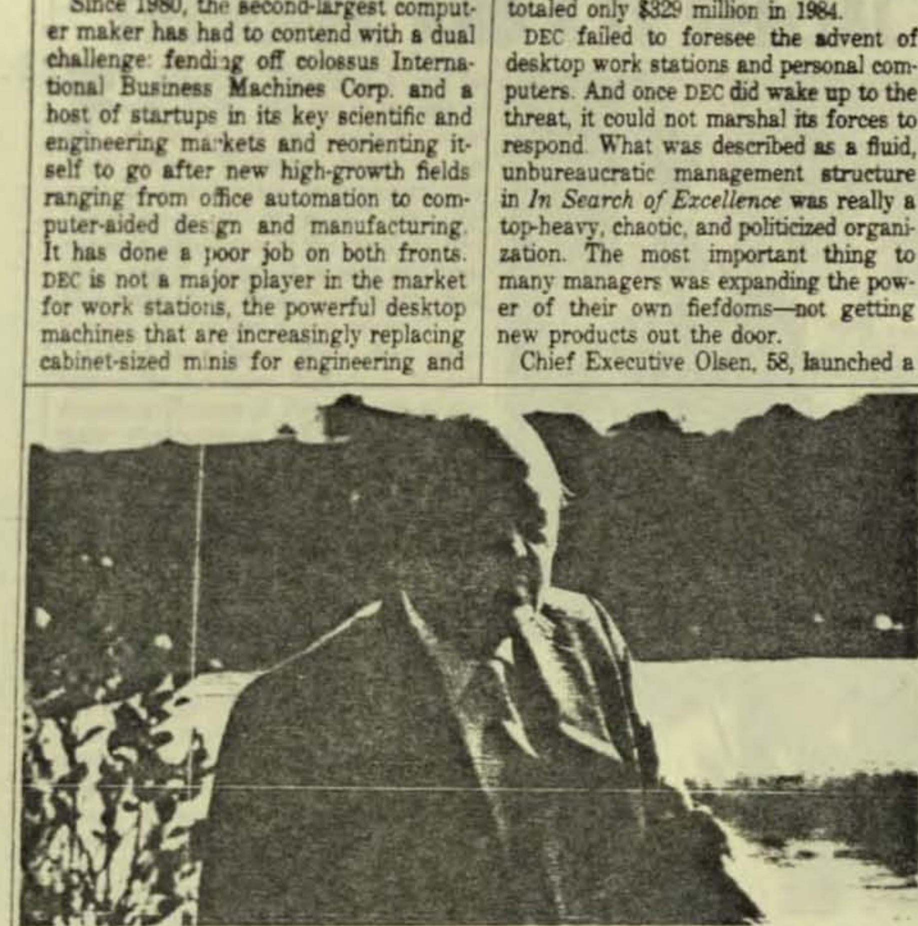
Since 1980, the second-largest computer maker has had to contend with a dual challenge: fending off colossus International Business Machines Corp. and a host of startups in its key scientific and engineering markets and reorienting itself to go after new high-growth fields ranging from office automation to computer-aided design and manufacturing. It has done a poor job on both fronts. DEC is not a major player in the market for work stations, the powerful desktop machines that are increasingly replacing cabinet-sized minis for engineering and scientific applications. And its sales of text-editing equipment totaled only $97 million in 1983, says Dataquest Inc.

DEC's struggles show how hard it can be for a company whose culture and skills are sharply honed for a specific market to adapt to a fundamental change in its world. Until just a few years ago, DEC thrived by building extraordinary ties with its customers: sophisticated data processing managers, scientists, engineers, and original-equipment manufacturers who customized its minis for end-users. DEC counted on these customers to find new applications for its minis. Instead of striving to be first in the market with new technology, DEC's primary emphasis was making highly reliable products.

NEW GAME. As underscored by a spectacular 31% average earnings growth from 1977 to 1982, DEC's approach worked splendidly when the name of the game was technological evolution and nurturing an existing customer base. But when the game changed to one of technological revolution and a need for a different customer base, DEC's playbook was ineffective. The result: After earning a record $417 million in the fiscal year that ended June 30, 1982, DEC's earnings plunged to $284 million in fiscal 1983 and totaled only $329 million in 1984.

DEC failed to foresee the advent of desktop work stations and personal computers. And once DEC did wake up to the threat, it could not marshal its forces to respond. What was described as a fluid, unbureaucratic management structure in *In Search of Excellence* was really a top-heavy, chaotic, and politicized organization. The most important thing to many managers was expanding the power of their own fiefdoms—not getting new products out the door.

Chief Executive Olsen, 58, launched a massive reorganization of his company in early 1983. It was an all-out attempt to streamline a bloated headquarters staff and replace DEC's celebrated "rampant chaos"—as *Excellence*'s authors describe it—with a marketing-oriented, team spirit. Since 1982, 11 of 34 vice-presidents have left the company.



OLSEN CLAIMS DEC NEVER INTENDED TO BE A MAJOR PLAYER IN PERSONAL COMPUTERS

Olsen succeeded in trimming DEC's minicomputer product line and in gearing up efforts to peddle VAX superminis. But there are few signs that the reorganization has accomplished another major purpose: speeding up product development. Indeed, the introduction of DEC's long-awaited 32-bit work station—originally scheduled for this year—has now been rolled back until next March.

Olsen's team seems torn between its loyalty to the traditional minicomputer business and the need to sell such products as personal computers and work stations. Accustomed to selling to data processing and technical professionals, DEC also is still struggling to learn how to sell to nontechnical businesspeople.

DEC had hoped to use personal computers to penetrate the fast-growing commercial and small-business markets. But it has tried three marketing strategies in the past 18 months with little success. In that period, the number of retailers carrying DEC personal computers plunged to 500 from 900. And DEC's total personal computer sales—virtually all of them to its existing 250,000 customers—totaled only $325 million in 1983.

Olsen's ambivalence about personal computers and work stations reflects the essence of DEC's problem. Long a believer that if you had a good product customers would beat down your doors, Olsen, an engineer trained at Massachusetts Institute of Technology, refused to allow aggressive marketing of personal computers. For instance, DEC's outlays for television advertising have been only a tiny fraction of those of IBM and Apple Computer Inc.

Now, Olsen insists that DEC was "sticking to its knitting" in minicomputers and never intended to be a major player in the personal computer game. "We had six PCs in-house that we could have launched in the late '70s. But we were selling so many [VAX minicomputers], it would have been immoral to chase a new market. [PCs and work stations are] the kind of high-growth business we're trying to get out of," he says.

HOMEGROWN. These comments contradict what Olsen has told BUSINESS WEEK for four years. They could be interpreted as an attempt to downplay DEC's humiliation in personal computers and its difficulty in getting its new work station out the door. But they also reflect Olsen's seeming reluctance to accept that the minicomputer market is maturing. That attitude would explain why DEC reportedly designed its personal computer so that it could not run powerful new software programs, speculates Michael Goldstein, a vice-president at Macmillan Inc.'s publishing group. MacPub, for instance, just introduced a program that transforms a $5,000 personal computer into a laboratory terminal able to tackle jobs that used to require a $35,000 DEC mini. Says Goldstein: "I think they wanted to protect their minis."

Unlike his counterparts at such other computer companies as Data General, Burroughs, and Apple, Olsen does not appear to see the need to recruit top-level managers from other companies to help DEC adjust to the demands of the new marketplace. His reliance on homegrown managers is part of a pattern of management that has tended to increase DEC's dependence on its existing customers. As a result, he has put DEC in a defensive posture that could ultimately relegate the company to an also-ran status in the industry.

'EXCELLENCE INC.': LECTURES, SPINOFFS, AND A TV SPECIAL

For Robert H. Waterman Jr. and his wife, Judith, the smash success of *In Search of Excellence* is one of life's ironies. Waterman began his career as a computer programmer who "hated to go out on stage." His wife, however, often sought the limelight: She once sang in a musical comedy and hosted a radio show. Since then, laughs Waterman, "we've had a role reversal."

Indeed they have. When first interviewed for this story, Waterman, 47, and co-author Thomas J. Peters were in the midst of shooting a Public Broadcasting System special on *In Search of Excellence* that will be aired in January. And Judy Waterman, who now writes software for Apple Computer Inc., was frantically trying to finish a program so she could accompany her husband on a five-week European speaking tour.

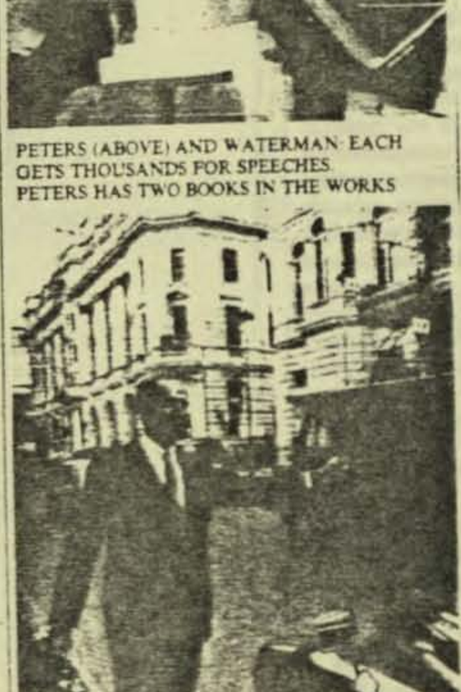
STILL HOT. Waterman, who charges as much as $12,000 per speech, estimates that he has given 250 since *In Search of Excellence* exploded off the charts to become the best-selling management book ever. The story is the same for Peters, whose talks are grossing him close to $1.5 million a year.

Meanwhile, *In Search of Excellence* is still a hot item. After selling 1.3 million hardcover copies—making it the most popular book in Harper & Row Publishers Inc.'s history—it is now doing just as well in paperback. It has sold nearly 1.5 million copies and is still going strong.

"No one in publishing would have predicted you'd sell over 1 million copies of a very serious business book priced at $19.95," says William M. Shinker, a marketing director at Harper & Row, which initially printed fewer than 15,000 copies. And no one foresaw such success at McKinsey & Co., where Waterman is a director and where Peters was a partner before making his self-described "extremely unpleasant" departure in 1981, a year before *Excellence* was published. That explains why McKinsey, which helped fund the research that led to the book, agreed to a 50-50 split with Peters of royalties from all sales over 100,000 copies. As a McKinseyite, Waterman received no cut.

Peters' agreement has made him "economically secure, to put it mildly," he laughs. But the 41-year-old Peters, an energetic man and nonstop talker, has no intention of slowing down and contenting himself with riches. He has become the consummate entrepreneur.

PETERS (ABOVE) AND WATERMAN EACH GETS THOUSANDS FOR SPEECHES. PETERS HAS TWO BOOKS IN THE WORKS



The five Tom Peters Cos. employ 20 people and have revenues of $4 million to $5 million (including *Excellence* royalties). Peters' operations range from seminars to consulting for such companies as People Express Airlines Inc. and Apple. He has a publishing unit—named Not Another Publishing Company Inc. His products include everything from newsletters and a training program to an appointment book that Random House Inc. publishes.

Peters has two co-authored books in the works. *Achieving Excellence*—a book on implementing the attributes of management excellence—will be published this spring. *A New Management*, which will profile 12 to 15 smaller companies, is scheduled for publication a year later. And Peters would like to team up again with Waterman "if he can find the time to do some writing."

That is a sore point for some McKinseyites, who do not agree with Waterman's assertion that the best-seller "reinforces McKinsey's image as a thought leader." Some are dismayed that only 20% to 25% of his time goes to clients. A touch of jealousy?

While Managing Director D. Ronald Daniel acknowledges that *In Search of Excellence* "has enhanced the firm's general reputation," he adds: "This is a firm with a purpose and a strategy. If there's to be another major collaboration—a 'son of *Excellence*'—Bob would have to decide if it made sense to take a leave [or resign]."

Waterman, who hungers to do more writing, is obviously torn between staying at McKinsey, where he has worked for 20 years, and taking Tom Peters' road. Indeed, he acknowledges that if McKinsey had not agreed to adjust his compensation to reflect the book's success, he would have been gone long ago. "It was dicey for a while," he says. "Suddenly you find yourself for at least your brief moment in history worth a great deal on the outside as a star."

A FAD? The big question now is how much longer Peters and Waterman will be stars. Dismissing the *Excellence* craze as nothing more than a fad, management guru Peter F. Drucker gives it no more than another year. Peters disagrees and bristles at any suggestion that his appointment books, speeches, seminars, and the like spring from any but the loftiest of motives: to spread the word to small and private companies.

Has Peters sold out? Absolutely not, he says. All he is doing is what *In Search of Excellence* preaches: staying close to his customers. "We are being dragged by the market," he says. "We are not pushing the market." ■

Business Week November

COMPUTERS: WHEN WILL THE SLUMP END?

NOT BEFORE 1987. AND BREAKNECK GROWTH MAY NO LONGER BE POSSIBLE

Evo Alexandre Jr. is in a quandary. As the head of computer-systems planning for Kaiser Aluminum & Chemical Corp. in Oakland, Calif., he would like to use the latest computer technology to automate Kaiser's plants and let its far-flung office workers trade documents instantly. But there's a problem: The technological and organizational issues surrounding the creation of such information networks, Alexandre says, are too unsettled to let Kaiser move ahead. "Until it's sorted out, we won't be pioneering."

A lot of computer customers these days are trying to avoid the arrows pioneers often get in the back. Suppliers of data processing equipment, who reveled in revenue growth of 30% a year in the early 1980s, saw sales growth fall below 15% last year. It's clear that there won't be a quick return to the go-go times. Instead, after a first quarter of disappointing news from IBM, Burroughs, and Convergent Technologies, it looks as if the slump in sales of computers will persist at least through 1986. Dataquest Inc., the San Jose-based market researcher, expects U. S. computer hardware revenues to rise only 7.4% this year, to $47.5 billion, including dealer markups (chart). Unless a breakthrough product changes the pattern, says Thomas H. Bredt, head of Dataquest's computer research, "we'll just bump along."

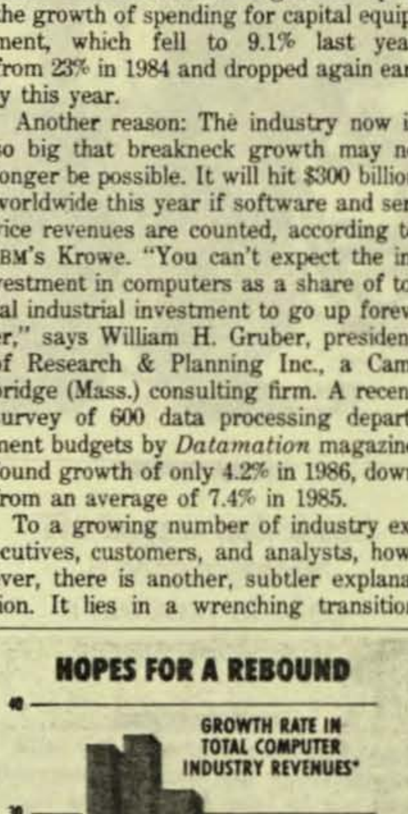
'ANTIQUE BOX.' No one factor explains this extended pause in the country's highest-flying glamour industry. In part, it reflects nothing more than buyers' anticipation of new products, such as a replacement for the basic IBM PC, which many analysts expect this year. "The PC is an antique box," says one major IBM customer. "People have been saying, 'What's next?'" New products already announced or due out soon, such as Hewlett-Packard Co.'s Spectrum line and Burroughs Corp.'s A Series mainframe family, ensure some pickup by 1987.

The state of the economy is also a factor. Allen J. Krowe, IBM's senior vice-president for finance and planning, attributes the slowdown to a general dip in the growth of spending for capital equipment, which fell to 9.1% last year from 23% in 1984 and dropped again early this year.

Another reason: The industry now is so big that breakneck growth may no longer be possible. It will hit $300 billion worldwide this year if software and service revenues are counted, according to IBM's Krowe. "You can't expect the investment in computers as a share of total industrial investment to go up forever," says William H. Gruber, president of Research & Planning Inc., a Cambridge (Mass.) consulting firm. A recent survey of 600 data processing department budgets by *Datamation* magazine found growth of only 4.2% in 1986, down from an average of 7.4% in 1985.

To a growing number of industry executives, customers, and analysts, however, there is another, subtler explanation. It lies in a wrenching transition that is changing the way computers are sold, lengthening the buying process, and creating new organizational and even psychological problems for the people who use them. The catalyst for all this is a move away from simple number-crunching toward information networks. These networks ease the communication of data and put more reliance on small and medium-sized machines than on the giant mainframes that dominated computer strategies in the past. Mainframe sales accounted for 36% of U. S. computer sales just five years ago, according to Dataquest. They will bring in only 19% of industry revenues this year.

HOPES FOR A REBOUND



*TOTAL OUTLAYS FOR COMPUTERS IN THE U.S., INCLUDING DEALER MARKUP

DATA: DATAQUEST INC.

INFORMATION NETWORKS. To some degree, says William F. Zachman, vice-president for office technology assessment at International Data Corp. (IDC), this shift is occurring because smaller systems for the first time offer more power for the dollar than large ones. "Users are taking advantage of the economies of small," he says. But to do this, they must tie together a vast array of often incompatible gear, find software that can operate on many different machines, and develop new strategies for storing and securing information and routing it around their companies.

The biggest impediment to growth, in this view, is the lack of a clearly defined set of hardware and software standards on which to build these networks. Thus the task of designing and installing them is overwhelming, and "we're seeing a much longer planning cycle as users rethink what their information management strategy is," says Robert C. Miller, senior vice-president of Data General Corp. Such delays "do more to explain the demand shortfall than any macroeconomic factor," says David N. Martin, president of National Advanced Systems, a National Semiconductor Corp. subsidiary that sells computers.

The move toward information networks will benefit some companies and punish others. IBM may suffer some erosion in the growth of its mainframe com-

CHART BY HAYES COHEN/BW

puters, add-ons, and software, which together accounted for 45% of its $50 billion in worldwide sales last year. It is racing to plug gaps in its lineup of smaller systems and to spell out networking strategies (page 62). But it faces a daunting task in helping customers link its nine main computer designs, which are incompatible (table, page 63).

By contrast, Digital Equipment Corp. is in high gear. Having developed a continuum of products that can be easily linked to sophisticated networks, it is giving IBM some unaccustomed competition, particularly in Big Blue's commercial computing stronghold. DEC's success has finally vindicated the much maligned leadership of founder Kenneth H. Olsen (page 64). "Suddenly the world is wide open to the kinds of efficiencies we offer," he says.

PRICE-CUTTING. Even for companies less well-positioned than DEC, the slump's had a silver lining: It's forced them to be more efficient. Overblown expectations for growth in 1985 led to inventory pileups and excess hiring that sent profits tumbling an average of 14%, according to IDC. Control Data, Computervision, Wang Laboratories, among others, had big losses, and even IBM could not improve on its 1984 earnings of $6.5 billion.

The biggest task ahead is tying together a vast array of often incompatible gear

Now, with expectations lower, bloated inventories deflated, and thousands of excess workers fired, most of the players expect to see profits improve. "We don't need 50% growth," says John Sculley, chairman and chief executive of Apple Computer Inc. "We can make a lot of money with 20% growth." Apple and others, Sculley adds, have developed more sophisticated forecasting that should help prevent overly ambitious projections in the future. "I'm optimistic because it's a manageable industry now," Sculley declares.

Moreover, most companies will get a boost from Europe, where growth has continued at a steady pace. The precipitous drop in the dollar magnifies these gains for U.S. companies. So analysts at Kidder, Peabody & Co. expect Apple's earnings to double this year, DEC's to climb 30%, and IBM's to advance 10%.

The primary threat to earnings progress is price-cutting that may get out of hand as companies jockey for position. IBM is leading the way. It slashed prices of its newest 3090 mainframes by 10% in February, then on Apr. 2 marked down its PCs as much as 25%. William D. Easterbrook, who follows mainframe computers for Kidder Peabody, figures that IBM's new lineup of midsize 4300-series machines is priced 30% below equivalent earlier versions. "That is a much bigger price cut than IBM traditionally uses within the same family," Easterbrook says. "I see it as a reaction to inroads by DEC and others."

MARKET 'BUBBLES.' Such cutthroat competition is endemic in the computer industry, but slow sales growth magnifies its effect. So to keep from shrinking, many computer makers are now targeting hot market segments rather than trying to be all things to all customers. It's these markets—and others that will appear as the shift to networks gains momentum—that are likely to lead the industry out of its slump sometime in the next two years. Gerald L. Peterson, marketing vice-president of Tandem Computers Inc., calls these niche markets "bubbles." "The trick," says Peterson, "is to figure out where the new

ILLUSTRATIONS BY KEITH BENDIS

bubbles are being formed and tap into them early."

The most important bubble recently has been in the scientific and engineering community, which despite the slump has been busily installing networks of powerful desktop workstations linked to supercomputers and minicomputers. Such networks can greatly enhance productivity for product designers and other technical workers. While business computer sales grew only 3.8% last year, dollar sales of computers used in technical work were up 10.8%, Dataquest says.

This helps explain the recent success of DEC, which dominates the technical market with its Vax superminis, as well as that of Cray Research, the leader in supercomputers, and Apollo Computer and Sun Microsystems, the workstation kingpins. "Virtually everything we sell goes into a network," says Scott McNealy, president of Sun. This growth is likely to continue at least through the end of the decade as engineers explore such new uses of computer power as simulating wind tunnel tests and modeling auto body styles.

Other bubbles haven't quite risen to the surface. Potentially the biggest technical market of all—factory automation—still has a way to go. It's true that a factory network standard—the Manufacturing Automation Protocol, or MAP—has been in place since 1982, when General Motors Corp. informed its shop-floor computer suppliers that it would give priority to those who offered equipment compatible with MAP. And at least 20 companies, including IBM, DEC, and Hewlett-Packard, sell MAP-compatible products. But attaching a robot to a factory computer network still costs more with the newfangled MAP components than it does with the old connections, notes Michael A. Kaminski, manager of the MAP program at GM.

In offices, equally thorny issues are more psychological than technical. Many customers who binged on personal computers and word processors in the last few years and now want to tie them into networks must overcome significant organizational and educational hurdles.

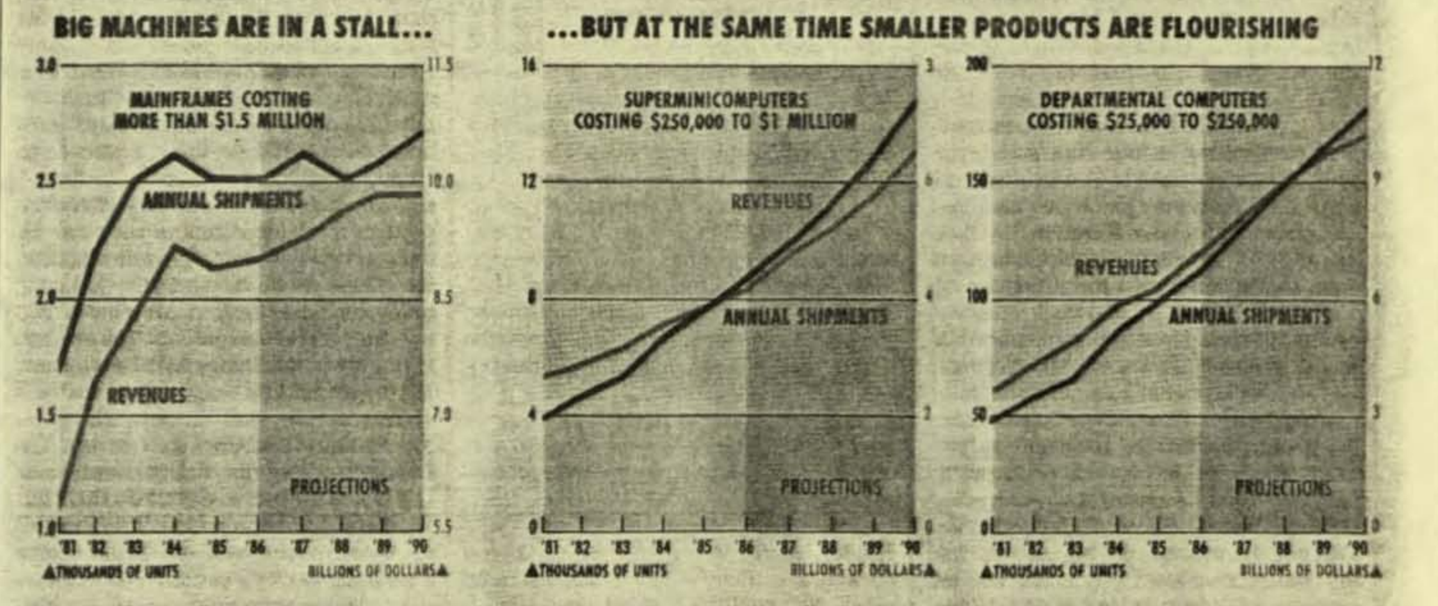
'THE HARDEST PART.' At American Can Co. in Greenwich, Conn., for example, managers from all of the company's financial services units can now use a computer network to tap into a portfolio management system developed by one subsidiary or a corporate tax program from another. But installing such a network in a company as decentralized as American Can, where subsidiaries prize their independence, was complicated. "We had to agree on technical issues, but that wasn't the hardest part," says Susan M. Smalley, who heads technical services at the corporate level. "Dealing with corporate culture issues was harder."

Bank of America is facing similar challenges. Its ambitious goal is to link individuals in work groups such as marketing teams, tie the work groups to departments, and then connect the departments to corporate information banks. Such a network, says R. Dennis Wayson, head of professional support services for the San Francisco-based bank, will let BofA expand its electronic mail system and handle many reporting and personnel tasks at the local rather than corporate level. It also will make possible a move to electronic publishing of internal documents. But "it won't be good for anything until we solve the management and cultural questions it raises," Wayson says. The bank, he adds, will have to answer such questions as who keeps what kinds of information, who has access to it, and who is responsible for service and support.

Because many parts of a company must be involved in putting together a network, the computer buying cycle is getting stretched out, exacerbating the sales slowdown. "They have to set up a committee to look at it," says Douglas C. Chance, vice-president and general manager at Hewlett-Packard Co. "It's a more complex purchase." One result of the internal wrangling, says John J. Connell, executive director of the Office Technology Research Group in Pasadena, Calif., is that IBM is no longer the automatic choice of many large companies. Buyers are more willing to look at new suppliers with ideas that differ from IBMs. Trying to figure out the merits of each can draw out the process even longer.

Assuming that customers can resolve such issues, there will be tempting new bubbles of opportunity in offices as well as factories and engineering labs. Once documents can be passed around a company for comments and refinement, desktop publishing of the finished versions should be cheaper than hiring a printing company. Such publishing systems could be a $3.5 billion industry by 1990, predicts Ajit Kapoor, Data-

THE PARADOX OF



quest's top electronic publishing analyst.

The winners in this and other new office markets will be the companies selling systems and software that handle many tasks and connect effortlessly to other computers in the company. True integrated office systems have become available only in the past couple of years, and DEC and Data General, rather than IBM, have taken the lead in installing them, according to Dataquest. While IBM is catching up quickly, it is clearly no longer unchallenged.

IBM has one overwhelming advantage: As the dominant force in both desktop and mainframe computers, it has been able to set de facto standards for moving data around. Its Systems Network Architecture, which lays down the etiquette for conversations between terminals and mainframes, is "the backbone that pulls it all together," boasts Terry R. Lautenbach, president of IBM's Communication Products Div.

CREATING ACCESS. But IBM's standard-setting authority is being challenged on several fronts. Many of the new engineering networks, for example, are based on the Ethernet connection standard developed by Xerox Corp. as well as on AT&T's Unix operating system. AT&T is lobbying hard for Unix, which can be used on 140 different computer models. "You don't see the other [computer makers] doing connections within their own lines and to everything that's already out there," argues Jack Scanlon, group vice-president for AT&T's Computer Systems Div.

In addition, more than 30 companies have joined in a consortium called the Corporation for Open Systems, which is trying to quickly promulgate a new set of common standards. COS has drawn so much support from customers that even IBM has joined. Competitors are skeptical about the company's motives, but IBM executives say they support the concept of industry standards even if IBM doesn't set them. "I think we'll get more business this way," says Lautenbach, "not less." He reasons that as the industry's lowest-cost producer and distributor, IBM would have an advantage if all equipment used the same standards.

Once the tangle of problems with standards has been cleared away, the era of information networks promises spectacular gains in the usefulness of computers. "What people have really been doing with computers up to now is to create information," says Apple's Sculley. "We've barely scratched the surface of getting access to information." Adds Paul C. Ely Jr., president of Convergent Technologies Inc.: "There will be another millennium of growth, driven by work-group computing."

A bright spot: Most companies will get a boost from Europe, where growth continues

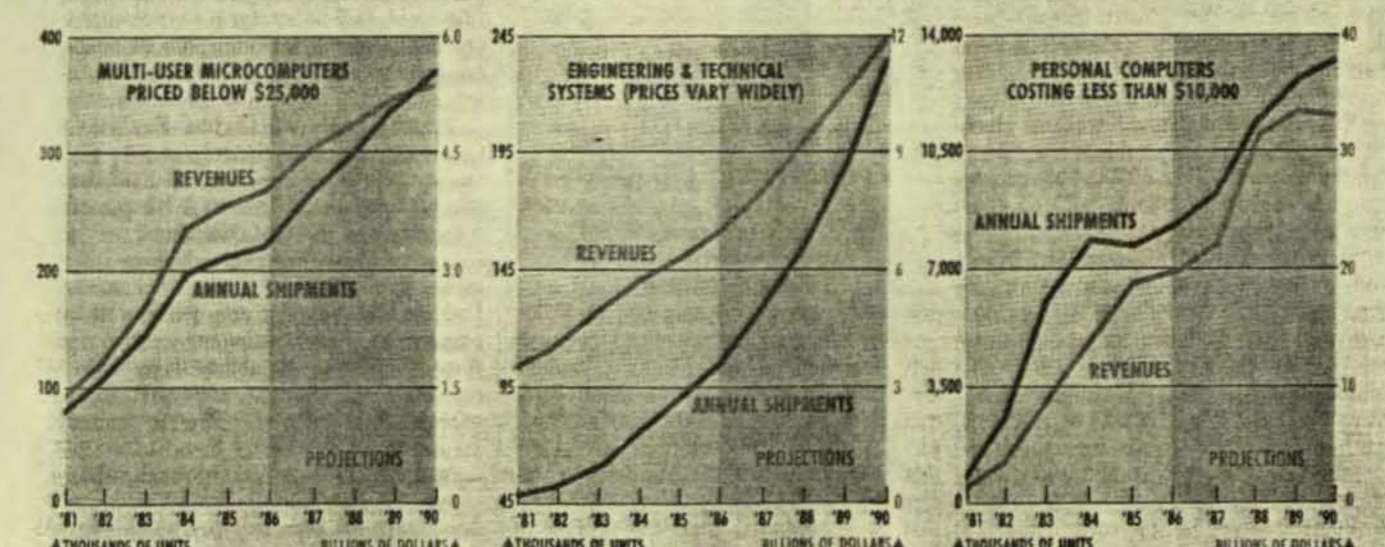
The growth will not necessarily favor the present industry leaders. Traditional competitors such as Burroughs, Sperry (formerly Univac), NCR, Control Data, and Honeywell are "more vulnerable than ever," says Grant S. Bushee, executive vice-president of market researcher InfoCorp. As the job of developing new products becomes more complex, Bushee says, these companies must get better at zeroing in on a few market segments. Larry E. Jodsaas, president of Control Data's Computer Systems and Services Group, claims to be doing just that. Control Data Corp., which lost $568 million last year, is concentrating on just six markets, including weather forecasting and factory automation. NCR Corp. has been building on its traditional strength in transaction processing.

SHORTER CYCLES. Companies that can't master the intricacies of niche marketing will have to scramble to stay abreast of accelerating product cycles and the fierce competition stimulated by network standards. Already, dozens of hot startups that make midrange systems based on the Unix operating system have emerged. These companies—which cater almost exclusively to technical customers—move at a killing pace. As Sun and others start building mainframe power into their desktop boxes, the entire industry will have to speed up. "The minicomputer companies are used to building $500,000 machines with five-year life cycles," says McNealy of Sun. "Our prices are under $50,000, and our product cycles are only 18 months."

The challenge to IBM, DEC, and other established companies will be to protect their proprietary designs from such upstarts and, at the same time, accommodate the standards that customers must have before they fully enter the network era. In such an environment, past performance doesn't guarantee future success for anyone.

By John W. Wilson in San Francisco, with bureau reports

OF COMPUTER SALES



Applying fault tolerant system architectures

Maximizing system uptime means getting the most from your equipment investment. Here's a look at one method of keeping your system up and running.

Jonathan A. Humphry, President
Triplex
Torrance, CA 90501

Manufacturers of industrial automation equipment are continuously challenged by equipment users to help improve manufacturing and process efficiencies. Fault tolerance is an emerging technology available today that offers a way to improve production efficiencies and profitability.

In a control system any element that can fail and bring the system down is called a "single point of failure." If such an element is extremely reliable or has very predictable and controllable failure modes, it may not present a problem in achieving high availability. If, however, it is unreliable or if its failure modes are unpredictable and/or cause catastrophic effects, it will severely reduce system availability.

This article outlines the two competing fault tolerance computer technologies used today – SIFT (software implemented fault tolerance) and HIFT (hardware implemented fault tolerance). It also discusses currently available fault tolerant system architectures, including dual, hot backup, triple modular, and 3-2-1, and touches on fault tolerant I/O and power supplies. It concludes with a look at the economics of fault tolerance and an application example.

Fault tolerance features

A practical fault tolerant computer or programmable controller must have the performance of a conventional, non-fault tolerant system. Plus:

- No single point of failure,
- Hot replacement of all modules,
- 100% fault detection
- High availability and safety,
- Fault detection, isolation, and re-integration of failed modules transparent to the application.

Hardware vs software

The SIFT approach to fault tolerant designs (Fig. 1) uses

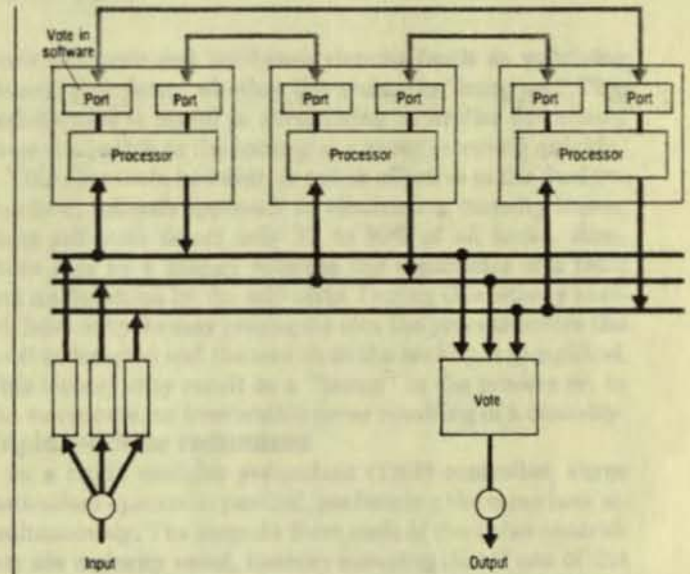


Fig. 1 (right, top): The SIFT approach uses multiple computers running asynchronously tied together by a communications channel. Voting is accomplished in software.

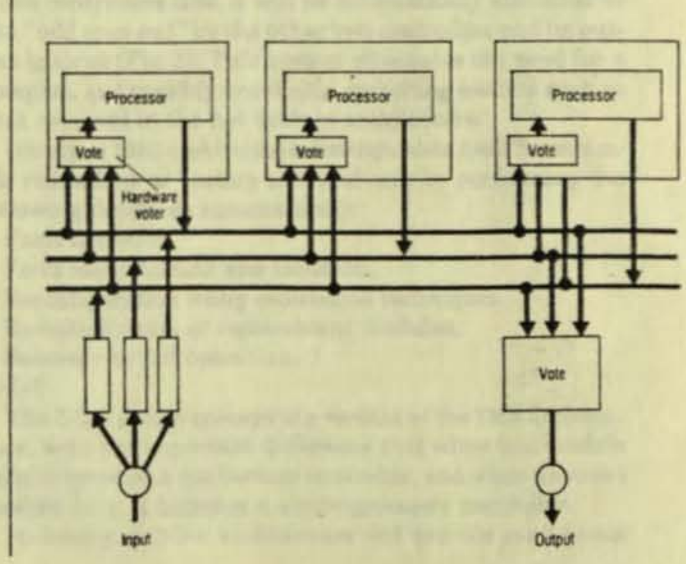
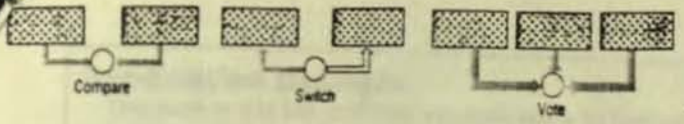
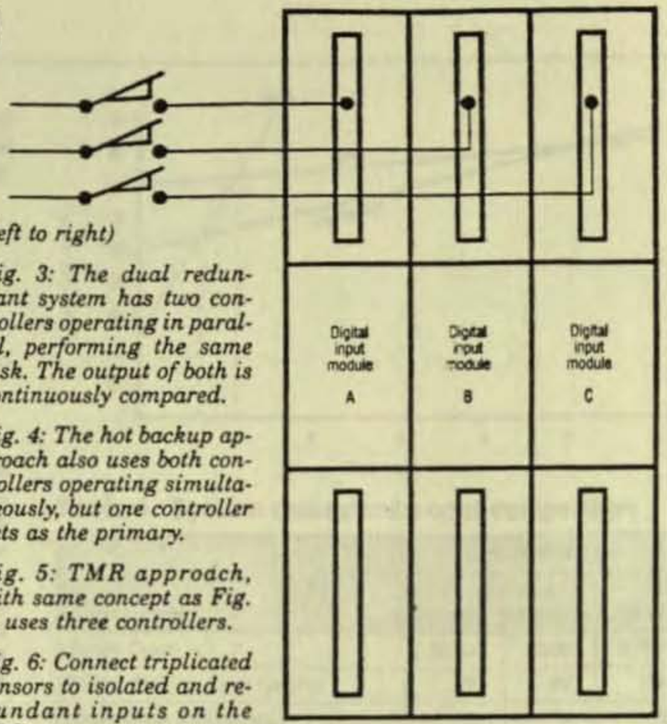


Fig. 2 (right, bottom): The HIFT approach uses lock-step synchronization – providing an identical time base to all redundant processors. Hardware voters interface to each processor.





(left to right)

Fig. 3: The dual redundant system has two controllers operating in parallel, performing the same task. The output of both is continuously compared.

Fig. 4: The hot backup approach also uses both controllers operating simultaneously, but one controller acts as the primary.

Fig. 5: TMR approach, with same concept as Fig. 2, uses three controllers.

Fig. 6: Connect triplicated sensors to isolated and redundant inputs on the PLC or computer.

multiple computers running asynchronously. A special communication channel is used to communicate between pairs of processors, connecting all processors either directly or indirectly through a third unit. In a real time control system, such as a PLC, the communication channels are periodically used to bring the multiple processors into synchronization. Redundant commands intended for control devices (such as valves) are compared and, if a disagreement exists, the majority of the results computed by the multiple processors is used as the final output.

The principle advantage of the SIFT approach is the ability to use standard computer hardware. However, disadvantages also exist. Substantial time and software overhead is involved in synchronizing the processors and performing the majority vote function on each output command. Furthermore, a time delay will exist from the occurrence of a fault until it is detected in the periodic voting process. During this delay time, the machine is vulnerable to becoming confused by a second fault.

The HIFT approach (Fig. 2) relies on lock-step synchronization. HIFT requires a fault tolerant clock providing an identical time base to all redundant processors. In this way, all processors execute exactly the same instruction simultaneously. Simple hardware circuits can be used to determine the majority vote on each instruction execution and data manipulation in all processors. A voter circuit can be implemented in currently available semi-custom integrated circuits. Therefore, the HIFT approach provides instantaneous fault detection and 100% fault coverage. Also, fault handling overhead is reduced in the HIFT system. HIFT only relies on software functions to annunciate faults to maintenance personnel. HIFT does, however, require a significant investment in the design of the custom circuits needed to accomplish the voting and fault detecting tasks.

Dual redundant, fail-safe architectures

A dual redundant fault tolerant design has two controllers operating in parallel, identically performing the same task (Fig. 3). The outputs from the two controllers are continuously compared; as long as they agree, the process continues. If the two controllers ever disagree, the process is stopped or placed in a safe state while the failed device is isolated and repaired or replaced.

The dual redundant, fail-safe approach is effective in eliminating casualty losses, but results in reduced availability as compared to a single controller. This is because the dual system is heavily biased to shut the system down in the event of a fault and, with twice the hardware, it will have twice the failure rate.

Hot backup

The hot backup approach also uses a dual architecture, with both controllers operating simultaneously (Fig. 4). One controller acts as the primary, performing the entire control function. In this way, only one controller interacts with the process at any given moment. When a failure occurs in the primary controller, the backup controller is switched in to take over the control function, and the primary is taken off line.

Failures in the primary controller are detected by software self-tests and hardware circuits (such as watchdog timers) that detect whether the system is "hung up." This architecture is useful in eliminating controller downtime, since the switch to the backup can occur relatively quickly.

This approach, however, is not as effective as the dual redundant, fail-safe approach in eliminating casualty losses, since self-tests detect only 70 to 80% of all faults. Also, there may be a latency between the occurrence of a fault and its detection by the self-tests. During this latency period, false outputs may propagate into the process before the fault is detected and the switch to the backup is completed. This latency may result in a "bump" in the process or, in the worst case, an irreversible error resulting in a casualty.

Triple modular redundant

In a triple modular redundant (TMR) controller, three controllers operate in parallel, performing the same task simultaneously. The outputs from each of the three controllers are majority voted, thereby ensuring that if one of the three controllers fails, it will be automatically identified as the "odd man out" by the other two controllers and its output ignored (Fig. 5). This system eliminates the need for a complex, and possibly unreliable, switching method such as that required in the hot backup architecture.

Ideally, a TMR architecture distinguishes itself from simple redundant or backup architectures by performing the following functions automatically:

- Fault detection,
- Fault identification and isolation,
- Reconfiguration using redundant techniques,
- Re-initialization of replacement modules,
- Recovery to full operation.

3-2-1

The 3-2-1 design concept is a variant of the TMR architecture, with the important difference that when one module fails, it becomes a hot backup controller, and when a second module fails, it becomes a single processor controller.

In theory, a 3-2-1 architecture will provide continuous

Application example

The table in this box provides example costs for fault tolerant and non-fault tolerant systems. The costs incurred are for the programmable controller based system specified below:

- 64 discrete inputs
- 48 discrete outputs
- 32 analog inputs
- 16 analog outputs
- 12 proportional-integral-derivative control loops
- operational 21 shifts/week (2 week/year scheduled maintenance)
- $1000/hr value added
- potential damage costs estimated from $5K to $50K

The graph shown here examines the total cost, over time for the three systems compared in the table. It shows that both the hot backup and triple modular redundant (TMR) controllers become most cost effective than the single controller within one to two years.

The TMR system becomes most effective when a useful life of five years or more is considered. The plots on the graph do not reflect a consideration of the damage costs. A TMR controller is the only system that completely eliminates all single points of failure. Therefore, where damage costs or safety concerns are significant, a TMR system is heavily favored.

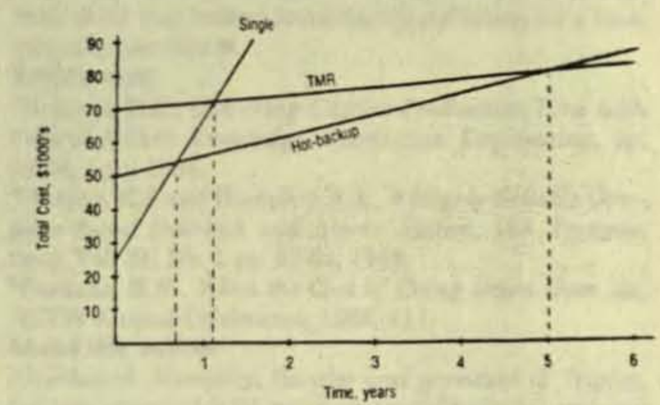


Table 1: System redundancy cost comparison

Cost factor:	Controller type: Single	Hot Backup	TMR
Initial Cost	$25K	$50K	$70K
Maint./Repair Cost (yearly)	2K	2K	2K
Downtime (hours/year)	40	4	0
Yearly Downtime Cost	$40K	$4K	0

availability. Unfortunately this is not altogether true. The greatest problem with a 3-2-1 system is that any time the system is not operating with its full triple redundancy (after the failure of a module, for example), it is prone to all of the faults and potential drawbacks found in a hot backup or single controller system. Therefore, 3-2-1 design sacrifices safety in an attempt to gain a marginally greater degree of availability.

Input/output architecture

A control system's sensors and actuators generally are more prone to failure than the electronics. Therefore, the design of a fault tolerant computer or PLC based system must consider failure modes in the sensors and actuators.

Controls engineers sometimes use triplicated sensors to provide fault tolerance in the measurement, or input, side of control. Limit switches and thermocouples are relatively inexpensive and can be easily triplicated to measure the same physical position or temperature. Ideally, these triplicated signals would be connected to isolated and redundant input channels to the computer or PLC, as shown in Fig. 6. Note in this figure that no single point of failure exists between the point at which the physical parameters are measured and the processor itself. If properly implemented, failures in the sensors and the input modules can be repaired without disturbing the process.

Actuators, such as valves, motors, and solenoids, are generally not replicated due to cost. Therefore, fail-safe or fault tolerant output circuits are required to provide system level fault tolerance from the processor to the actuator. These circuits rely on two switches (usually switched transistors) in series to ensure that the current can be turned off. The effect of current spikes on the output circuit is reduced by connecting the load between the switches. The inductive nature of the load will filter external noise.

This type of output circuit will only provide protection if the transistors can be turned off. To make certain the output transistors aren't shorted, each transistor should be tested by briefly turning them off. Such a self-test can be included in the design of a fail-safe output module. The self-test can be performed by momentarily turning off the drive to the output transistors and observing the resulting current change in the load circuit. This testing can be performed rapidly and, therefore, will have no effect on the control of typical actuators.

Two fail-safe output circuits can then be combined in series with the load to form a fully fault tolerant configuration. This configuration provides proper operation in the event that any one transistor has failed shorted or open.

Power supplies

A successful fault tolerant power supply design must provide for load sharing circuitry so that the load is balanced among the redundant supplies. When determining the number of supplies used in a redundant power supply configuration, fault detection is not a consideration; rather, the output capacity of the supplies is the key factor. To understand this point, consider a situation in which three power supplies are used. Assuming all three are identical, and that a single failure must be tolerated, two supplies must be capable of providing the required power. Therefore, the triplicated power supply system will necessarily have 50% excess capacity. A dual system however, necessarily involves 100% over-capacity, giving the triplicated system a cost-performance advantage.

Economics of fault-tolerance

The initial cost of a fault-tolerant control system is greater than that of a single controller system. However, the initial purchase price is only a portion of the total cost of ownership. Other costs include maintenance and repair of failed components, loss of production due to downtime, and damage to production equipment or material that results from system failure. When these are considered, fault-tolerant controllers can offer the more cost-effective solutions as compared to conventional single controller systems.

With the increased performance and reduction in price of modern electronic components, the initial cost of controllers has declined significantly. Even considering the cost premium of fault-tolerant controllers over conventional equipment, the purchase price of the controller doesn't represent a significant portion of the total cost of ownership.

The ability of fault tolerant controllers to detect, identify and isolate their failed components makes troubleshooting much easier and quicker. Their ability to continue operation in the presence of failures allows for scheduled maintenance, eliminating extra costs of performing these duties during premium rate periods. Therefore, maintenance and repair costs for fault tolerant controllers are generally less than those for conventional controllers.

The reduction in downtime costs is the principal cost advantage for fault-tolerant controllers. Market surveys have shown that typical downtime runs 5%, and about 50% of that downtime is due to failures of the controller equipment components. Depending on the number of shifts per week that the controller is used, the downtime resulting from controller failure can range from 5 to 60 hours per year.

The industrial control systems found in today's manufacturing and process plants show an increasing amount of system integration. This has resulted in greater value-added per hour figures than were observed just a few years ago. Typical value-added per hour figures range from \$200/hr to \$10,000/hr. Even larger values are not uncommon.

Damage costs due to controller failures also favor the use of fault tolerant controllers over single controllers. Often in safety critical areas, or processes where controller failures may cause damage to expensive equipment, the damage costs alone may become immediate justification for a fault tolerant controller.■

References

[1]Hegland, D.E., *Unlocking Captive Production Time with Control failure Knowledge*, Production Engineering, pp. 58-64, April 1981.

[2]Murphy, K.J. and Humphry, J.A., *A Highly Reliable Computer-Based Interlock and Alarm System*, ISA Transactions, Vol. 24, No. 1 pp. 47-52, 1985.

[3]Peterson, H.N., *When the Cost of Going Down Goes Up*, IECON Annual Conference, 1983, 111.

About the author

Jonathan A. Humphry, founder and president of Triplex, has been responsible for many successful projects involving architecture, reliability analysis, and implementation of fault tolerant computing systems with both the U.S. Air Force's spaceborne computer project and Hughes Aircraft Company. Dr. Humphry received his PhD from the University of Southern California and his BSEE and M Eng (electrical) from Cornell University.

The author will be available to answer any questions you may have about this article. He can be reached at (213) 618-1441 on Monday and Tuesday, November 9 and 10, during normal business hours.

If you found this article useful and would like to see more articles on this or similar subjects, circle 162.

Reliability, availability, and fault tolerance

How to evaluate your system reliability needs.

Paul R. Hamilton
Product Marketing Manager
Gould Inc., Industrial Automation Systems
Andover, MA 01810

System reliability is an extremely important concern these days, largely because of dramatic changes in our manufacturing practices, coupled with new economic and social pressures. For example, our increased reliance on automation to cut costs and raise productivity levels, has made industry more vulnerable to control system failures. And new manufacturing concepts, such as just-in-time production/inventory, magnify the impact of system downtime. To all of this, you must add the fact that competitive pressures and potential liability losses have made the cost of lost product and system mishaps virtually unbearable.

A conservative estimate of downtime cost for a typical operation is $20,000 per occurrence, with an average annual production loss of $40,000. Compared to possible losses from a down system, the cost of upgrading the system to improve the reliability is insignificant.

Everyone needs and wants a highly reliable control system. However, it's difficult to determine how much incremental engineering, training, and dollars you must invest in a system before the costs exceed the gains. In general, upgrading the reliability of your control system is best suited to applications where a control failure might result in:

- Loss of production time;
- Creation of a safety hazard;
- Environmental damage;
- Damage to raw materials or products;
- Loss of production data.

Quantifying system requirements

A control system has several time-related factors that measure its effectiveness – many of which are interrelated. Stated simply, a system is judged by its "abilities" – availability, reliability, maintainability, and useability[1].

Before we take a detailed look at each of the abilities, we should point out that, while this evaluation procedure can be applied to various types of control systems, we will use programmable controller (PLC) based systems as examples in all discussions.

Availability

The *availability* of a system, or *uptime*, is the total time it is in use plus the time it is idle or standing by and capable of being used. In considering availability, it is also necessary to consider the *mean time between failures* (MTBF). This is the average time that a device will operate before failure. When a system is designed to be operative 100% of the time, then the system uptime is equal to the MTBF.

Assuming that the system is always operative, downtime is the time spent in active repair (including both diagnosing and fixing the problem, plus time spent waiting for spare parts, paperwork, and so on). In this case, the downtime is the same as the *mean time to repair* (MTTR).

Thus, availability is the percentage of time a system is available, and is typically expressed as the ratio:

$$\text{UPTIME} \div [\text{UPTIME} + \text{DOWNTIME}]$$

or as the ratio:

$$\text{MTBF} \div [\text{MTBF} + \text{MTTR}]$$

You can start assessing availability by answering some specific questions about your control system, including:

- What availability is needed by your system?
- What is the critical time period during which you require high availability?
- Will the system tolerate an unscheduled shutdown?
- What is the downtime cost when availability is needed?
- What is real time as related to your system?

Reliability

If a system never failed, its availability would be 100%. However, systems do occasionally fail. The probability of a system operating without failure for a specific time is its *reliability*. The classic definition is "the probability of a product performing, without failure, a specified function under given conditions for a specified period of time."[2] Reliability is also known as the *probability of survival*.

The required probability of survival is based on a user-specified time period – usually the time you require your system to be operative or available. Although this time may be equal to MTBF, you undoubtedly would like MTBF to be much greater. An uptime requirement equal to the MTBF would yield only a 37% probability of survival over the specific time you require your system to be operative. (See the ıability later in this article.)

‚ıg reliability can be complex because it is sub- ‚de variety of factors such as unanticipated envi- ‚s, lapses in quality control, products flaws, and ‚aintenance. However, answering the following ques- ‚ about reliability should enable you to determine how portant it is to your application:

- What level of operational reliability is required?
- How reliable is the equipment?
- Can your system tolerate faults while it is running?
- To what level?
- Can the system be repaired without shutdown?

Maintainability

Maintainability identifies how easily the system can be serviced and repaired. This includes both preventive or scheduled maintenance, and unscheduled maintenance to restore service after a failure.

Quantifying maintainability is another complex task. Factors such as product design, personnel skills, availability of the proper repair equipment, spare parts inventory, and accessibility of service and support have to be considered. Because these areas also affect the overall system availability, they should also be considered in the original control equipment selection process. When evaluating a system's maintainability, consider the following questions:

- What amount of unscheduled downtime can your system tolerate per operational cycle?
- What is the estimated MTTR of the system?
- What level of diagnostics are available in the system?
- What is the probability of restoring service in your specified time period?
- Is modular on-line replacement possible?
- Are special tools, components, or spare parts necessary to bring the system on-line?
- Is complete training, documentation, and logistical support available as needed?

Useability

Usability is a measure of how easy the system is to operate, taking into consideration the convenience of its design. The answers to the following questions will help you quantify the useability of the system:

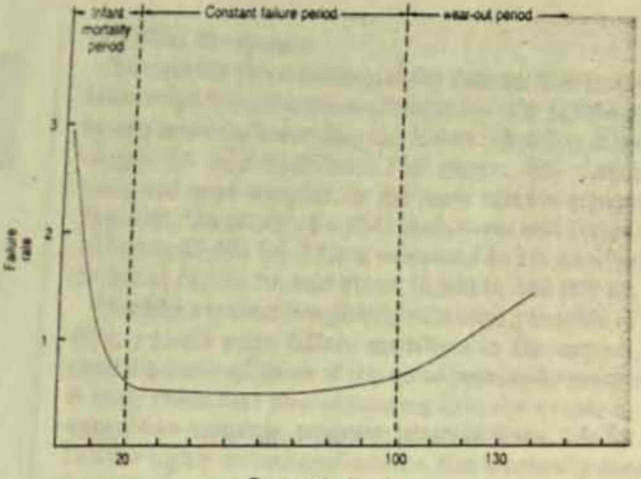
- How much effort is required to install and operate the system?
- How familiar is your current workforce with the equipment and support tools?
- Will incremental training be required?
- How transparent to normal operations is the equipment?
- Will additional hardware or software affect the maintainability of the system?
- Will increasing the system's availability decrease its functionality?

Other considerations

When considering an upgrade of a system to improve its reliability, there are, in addition to the four areas previously discussed, several additional relationships[3] to consider:

1. Assuming equal quality of all components and testing, the components with the most parts will fail first.
2. The probability of survival of a system will always be lower than the lowest probability of failure of a single component of that system.

For example, if the probability of survival (reliability) of one component is 90% for a specified time and another is 95%, the probability of successful operation of the two components as a system is (.9) × (.95), or 85.5% during the same time period. This formula is consistent for any number of components.



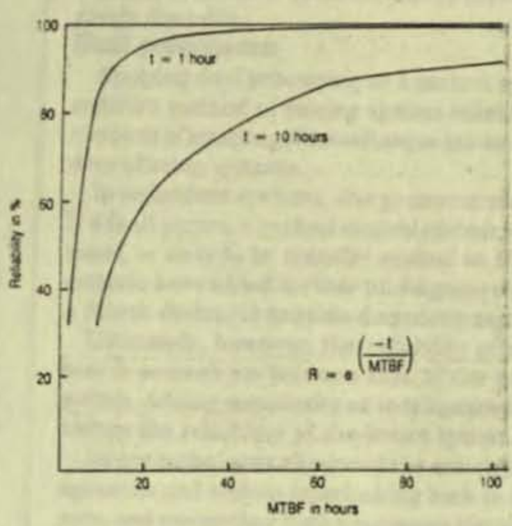


Fig. 1 (top): The "Bathtub Curve" represents typical failure rate of computer systems.

Fig. 2 (bottom): The MTBF of a system must increase exponentially to gain improvements in reliability.

3. The failure rate of complex systems typically follows a pattern known as the "bathtub curve" (Fig. 1). The curve has three distinct zones that differ in both the frequency and the cause of failure. They are the infant mortality period, the constant failure period, and the wear-out period.
4. In an electronic control system, most of the system's life will be in the constant failure period. The probability of survival for a selected time period is an exponential curve (Fig. 2). It is calculated with the formula:

$$\text{Reliability} = e^{-t/\text{MTBF}}$$

where,

t = specific time period or availability required by a system.

The fourth relationship merits some further qualification as it relates to MTBF. An increase in MTBF does not lead to a proportional increase in reliability. If t remains constant, the MTBF must increase exponentially to gain constant increases in reliability.

If a system must be available for one year, for example, a MTBF of 10 years gives a reliability of 90%. Increasing the MTBF to 20 years gives a reliability of 95%. However, to

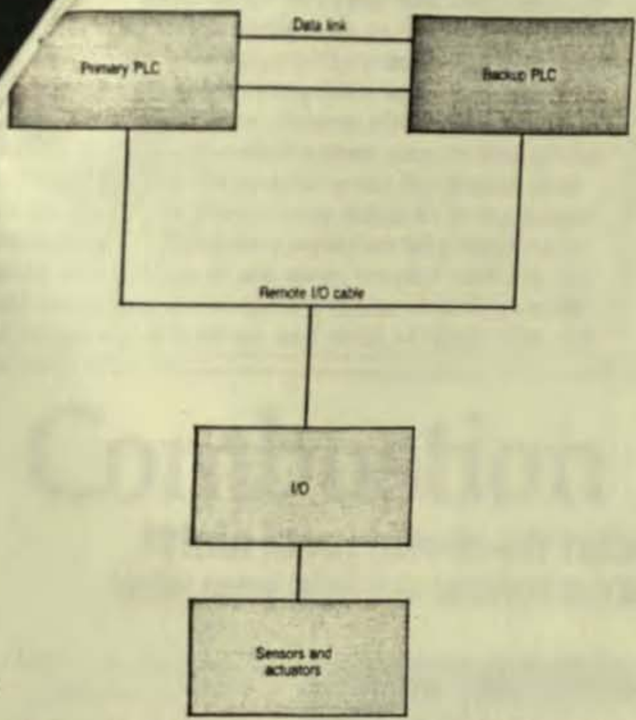


Fig. 3: Typical configuration used to obtain redundancy with programmable controllers.

gain a reliability of 99%, only four percentage points from 95%, the MTBF must be increased to 100 years. Obviously then, you should consider reliability as a function of a specific time or availability requirements rather than using MTBF as a the basis for a decision.

System design

In any system requiring fault tolerance, the two major design criteria are save the plant and save the process.

Every system should be designed to save the plant. If the plant loses all power or ability to operate, it should be designed to go through an immediate, orderly shutdown. Although product or equipment may be lost, the plant shuts down so people and the environment are not endangered. The shutdown can be caused by a failure of a critical component, loss of power, or the out-of-limit operation of a portion of the system.

With the plant and people protected, consider ways to save the process. In this situation, life and environment may not be threatened, but equipment may be damaged or product lost. The system design should allow for loss of critical components, whose failure could cause a system to shut down, or modify the operation to prevent product loss or equipment damage.

When designing a fault-tolerant system, first identify the critical system components. These are the ones that have the worst historical track record and are most likely to fail. In addition, identify the components whose failure when availability is required, would have a catastrophic impact on the process. Once these elements are indentified, isolate them, and then implement hardware or software that will allow the system to operate if they fail.

Redundancy

In the traditional model of, for example, a programmable controller (PLC) system, a random failure striking any of the three areas, (mainframe, I/O, or sensors and actuators) can affect the system.

To quantify the reliability of the system, first isolate the lower reliability subsystems. Remember, the reliability of a system is always lower than the lowest reliability of a single component. In a traditional PLC system, the mainframe, being the most complex, is the least reliable component. Typically, the MTBF of a PLC mainframe will range from 10,000 to 20,000 hr. This is compared to I/O interfaces at 30,000 to 50,000 hr, and I/O at 70,000 to 150,000 hr.

To make a system completely redundant, capable of handling a single point failure anywhere in the system, you must duplicate all three of the areas previously mentioned. A truly redundant manufacturing line, for example, consists of two complete, separate, identical lines, side by side. In a few highly critical applications, this is actually done. As a practical matter, however, this is seldom necessary and rarely desirable.

Dual processors

Applying dual processors to a control system is the most common method of raising system reliability (Fig. 3). The methods of applying this technique are as varied as the vendors offering systems.

In redundant systems, one processor monitors the other. If a fault occurs, a backup control system signals a third element, or switch, to transfer control to the backup. Many systems have added further intelligence to the switch, or to a fourth device, to provide diagnostic capabilities.

Ultimately, however, the reliability of a redundant system is somewhere between that of the processor and the switch. Adding complexity or intelligence to the switch will reduce the reliability of the entire system.

Newer techniques eliminate the switch by moving the diagnostics and system interlinking back to the main processors, and connecting both processors simultaneously to the same I/O communication link. Switchless redundancy involves two systems running concurrently, both systems reading inputs and solving logic, with only the primary system writing to the outputs. Each system contains an independent processor performing health monitoring, diagnostics, scan synchronization, and system state table transfer. In the event of a PLC failure, the backup unit detects the failure via the system diagnostics, pulls the primary PLC off line, and takes control of the system.

A major concern of redundancy has been the "bump" to the process. A bump is the time it takes the backup to:

1. Detect the fault,
2. Implement the switchover,
3. Stabilize the system outputs to the process.

Depending on the design and implementation of the system, the bump could be from 100 ms to several minutes.

If, during a failure, the system is designed to hold the last state of the outputs for a time greater than the bump, the process will maintain its last output state. If there were no changes to any inputs during this time, or the transfer time is sufficiently small compared to normal process change, the bump would be transparent to the process.

Minimizing the three time periods associated with the bump is a function of product design and system implementation. How quickly the system detects a fault depends on the extent of diagnostics in the PLC and the speed of its scan. Switchover speed is a function of design. With a hard switch, it can take 200 to 400 ms; done electronically without a switch can take as little as 20 ms. System stabiliza-

..ing time, can be virtually eliminated by syn- ..the scans of the PLCs and completely transfer- ..primary PLCs state table to the backup PLC on ..can. If a complete copy of the primary PLC state table ..transferred to the backup every scan, the backup PLC ..ld have different timer, counter, and register values. It ..ould take the backup several system scans to read all the inputs and stabilize the outputs to the last process state.

It's possible to further increase reliability by duplicating or triplicating I/O. Redundant inputs are fairly simple to implement, while outputs are more complex (and not discussed here). By connecting three independent drops of discrete inputs to each sensor and using OR gates with the three possible input pairs in user logic, you have essentially created a voting scheme. Analog or register inputs are connected similarly, but must be averaged in user logic.

If the system can recognize the health of each I/O module and I/O interface, the processor can flag the computer if any individual device fails. This allows quick identification of system failure and modification of the process. System reliability is further increased with the use of redundant I/O communication. This should allow you to replace any I/O drop or module while the system is on-line.

Maintainability and useability are two important considerations when selecting a redundant programmable controller system. Replacement of failed systems or components without interrupting the process will increase system uptime. Communication with both the primary and backup PLC without complicated software or hardware switches simplifies maintenance. Built-in software that implements the redundancy and redundancy diagnostics simplifies future system changes. Many systems require extensive user-implemented software specifically designed for your application. This makes any changes to the system software difficult and time consuming. And, finally, remember to consider such things as existing system knowledge, vendor service, and the logistical support available.

There are many alternatives for increasing reliability in a control system. Developing a clearer understanding of what reliability and availability mean will help you make smarter choices in fault tolerant control systems.■

References

[1] Juran, J. M., Gyrna, F. M., Bingham, R.S. "*The Abilities,*" *Quality Control Handbook*, New York; McGraw Hill Book Company, 1951, sect. 2-6 - 2-8.
[2]Ibid, Section 27.
[3]Ibid, Section 22.

About the author

Paul Hamilton has been involved in the sales and marketing of industrial electronics equipment and systems for over ten years. Hamilton earned his BSEE from the University of Pittsburgh. Prior to joining Gould, he was a sales engineer of industrial electronics equipment and systems at General Electric Company.

The author will be available to answer questions you have about this article. He can be reached at (617) 975-2806 on Monday and Tuesday, October 26 and 27 during normal business hours.

If you found this article useful and would like to see more articles on this or similar subjects, circle 161.