

VM and the VM Community: Past, Present, and Future

Melinda Varian

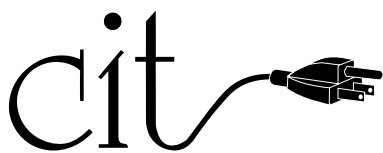
Office of Computing and Information Technology
Princeton University
87 Prospect Avenue
Princeton, NJ 08544 USA

—.—
Email: Melinda@princeton.edu
Telephone: (609) 258-6016

SHARE 89
Sessions 9059-9061
August 1997

(c) Copyright Melinda W. Varian 1989, 1990, 1991, 1997. All Rights Reserved.

Permission is granted to SHARE Inc. to copy, reproduce, or republish this document in whole or in part for SHARE activities only.



**Operating Systems
Computing and Information Technology
Princeton University**

VM and the VM Community: Past, Present, and Future

I. INTRODUCTION

I will be talking today about the past, present, and future of VM, with an emphasis on the influence of the VM community on the growth of the VM product over the past twenty-five years.

An earlier version of this paper was presented at Australasian SHARE/GUIDE in Melbourne in 1989. My husband Lee and I had a delightful time at ASG and are most grateful to ASG for being our host in Australia and to SHARE for giving us the opportunity to represent it there.

When I spoke at ASG, I began by conveying greetings from the then President of SHARE, Cecilia Cowles. I will do that again today, because the pictures are too good not to use again.

In the past, when I'd spoken at SHARE and SHARE Europe, my talks had been strictly technical. This talk was the first time I'd been asked to give my opinions, so you may find that you get more opinion than you wanted.

I must also ask you in advance to forgive me my ethnocentricity. Though I speak of "*the* VM community", I realize that there are actually several overlapping communities of VM people, located in different parts of the world, both inside and outside of IBM. For the most part, I will be speaking of the community of all VM people, but may sometimes lapse into using that term for the smaller community that grew up around the VM SHARE electronic conference. This community overlaps heavily with SHARE and the former SHARE Europe, with the annual VM Workshops in North America, and with various regional VM user groups. It includes many participants from IBM.

I'll be showing you pictures of some members of this community, but because there is not nearly enough time to show all the people who have made outstanding contributions to VM and to the VM community, my choice of who to show was semi-random, depending a lot on which pictures I was able to get. I owe thanks to many photographers who lent me their pictures, but especially to Joe Morris of SHARE, Stuart McRae of SHARE Europe, and Pam Christina of IBM.¹ I am also indebted to Sandra Hassenplug and John Hartmann for their assistance in preparing slides, as well as to several of my colleagues at Princeton.

I should probably also explain the iconography I'll be using. For many years, the SHARE VM Group lamented the fact that VM had no symbol, no totem. A couple of attempts were made to select one, but they fell flat, because, of course, such things can't be mandated. Meanwhile, the

¹ I am grateful to the many people who succumbed good-naturedly when I badgered them for photographs. I wish particularly to thank Bob Creasy, Adenah DeAngelis, Jerry DePass, Walt Doherty, Lyn Hadley, Ed Hendricks, Peter and Carol Jobusch, Ted Johnston, Ken Holt, John Shaw, Dave Tuttle, Lee Varian, John Wagner, Lynn Wheeler, Rich Wiggins, and Joan Winters.

MVS Group had the turkey (which they chose of their own volition), and they went around wearing turkey hats and putting turkey stickers on elevator doors, and so on. The legend is that the MVS Performance Project began using the turkey as a symbol in the early days, when MVS performance was definitely a turkey, and the symbol soon got extrapolated to the whole MVS Group.

With VM's amazing growth, the VM Group in SHARE had always had a problem making newcomers feel at home, simply because they always outnumbered the oldtimers. In 1983, the Group was going through yet another attempt to overcome this problem, and it was decided that at SHARE 60 we would hand out little square yellow stickers to newcomers to the VM Group and little square blue stickers to oldtimers, with the idea that if they all put the stickers on their badges, the oldtimers could identify the newcomers and help make them feel at home. The problem with that, of course, was that nobody could remember which sticker was which, so it didn't work out at all. A couple of days into that week, however, Carol Jobusch bought a few hundred teddy bear stickers, with the idea of affixing them to the cuddler of the oldtimers so that the newcomers would know that here was a warm cuddly person who ran *the* warm cuddly system and who could be counted on to be friendly if approached. Within hours, the teddy bear had become the *de facto* symbol for VM, and everybody in the VM Group, old or new, cuddly or prickly, was wearing a teddy bear on his badge. (The Jobusches subsequently got a 50-KB roll of stickers, to keep SHARE well supplied.)

One rather strange result of all this has been that the offices of many hard-bitten VM system programmers are now full of teddy bears. They're even on our wedding cakes. However, even without being reminded of it by the MVS Group, we would have been careful not to let our arctophilia degenerate into icky sweetness.

Not surprisingly, soon after VM adopted the teddy bear, the MVS Group decided that the turkey was no longer an appropriate symbol for MVS, and opted instead to use the eagle. But, of course, such things can't be mandated.

I should also warn you that you may notice in my presentation a few slides that indicate a certain rivalry between the VM and MVS Groups. I hope that none of you will take offense at our banter, for I assure you that the rivalry is a good-natured one and only skin deep. In fact, most installations in the SHARE VM Cluster run MVS, and most Cluster members use MVS every day, although, of course, very few of us use TSO.

II. A BRIEF HISTORY OF VM

I owe many kind VMers thanks for having shared their memories and their memorabilia with me so that I could share them with you.² I regret that this account will leave out many people who have made good and lasting contributions to VM, but this is inevitable, given our time constraints and my ignorance of much of what has happened inside IBM during the past thirty years. I would be delighted to hear any corrections or additions you may have to story I'm about to tell.

A. CTSS

In the beginning was CTSS, the "Compatible Time-Sharing System". CTSS was written by a small group of programmers³ at the Massachusetts Institute of Technology (MIT) in Cambridge, Massachusetts, under the leadership of Professor Fernando Corbató. One of the CTSS programmers was Robert Creasy, who was later to become the leader of the CP-40 project.

Papers discussing the idea of a time-sharing system began being published about 1959.⁴ There followed a period of experimentation at MIT and other institutions. An early version of CTSS was demonstrated on an IBM 709 at MIT in November, 1961. From that beginning, CTSS evolved rapidly over the next several years and taught the world how to do time-sharing.⁵

² I've managed to locate most of the people mentioned in this chapter. Without exception, they have been extremely generous with their time and assistance, regaling me with delightful tales of their days in VM and patiently enduring my endless questions. I am grateful to them all. I am also indebted to the people who have searched out physical evidence for me: the system programmers at Brown University, who donated an intact PID shipment of CP-67 Version 3; Bill Frantz, Scott Tyree, and Walt Hutchens, who sent me stacks of early manuals; Chuck Rodenberger, who sought out dozens of old "blue letters"; Alan Greenberg and Dewayne Hendricks, who sent me their archives from SHARE and GUIDE activities in the CP-67 and early VM/370 days; David Walker and Jacques Myon, who unearthed some amazing artifacts; Bob Cox, Gabe Goldberg, and Chuck Morse, who sent slide sets from VM demonstrations and announcements; Dave Tuttle, who wrote out his memoirs and also lent me an astonishing collection of VM relics; Fernando Corbató, Les Comeau, and Don Wagler, who made me videotapes of rare old films; and Stu Madnick, who sent me the contents of his attic.

³ Marjorie Merwin-Daggett, Robert Daley, Robert Creasy, Jessica Hellwig, Richard Orenstein, and Lyndalee Korn.

⁴ C. Stratchey, "Time Sharing in Large Fast Computers," *Proceedings of the International Conference on Information Processing*, Paper B.2.19, UNESCO, New York, June, 1959.

⁵ From the preface to the "candy-striped manual" (F.J. Corbató, M.M. Daggett, R.C. Daley, R.J. Creasy, J.D. Hellwig, R.H. Orenstein, and L.K. Korn, *The Compatible Time-Sharing System: A Programmer's Guide*, The MIT Press, Cambridge, Mass., 1963):

The only other general purpose time-sharing system known to be operating presently, that of the Bolt, Beranek and Newman Corporation for the PDP-1 computer, was recently described by Professor John McCarthy at the 1963 Spring Joint Computer Conference. Other time-sharing developments are being made at the Carnegie Institute of Technology with a G20 computer, at the University of California at Berkeley with a 7090, at the Rand Corporation with Johnniac, and at MIT (by Professor Dennis) with a PDP-1. Several systems resemble our

CTSS was developed on a series of IBM processors. In the 1950s, IBM's president, T.J. Watson, Jr., had very shrewdly given MIT an IBM 704 for use by MIT and other New England schools.⁶ Then, each time IBM built a newer, bigger processor, it upgraded the system at MIT.⁷ The 704 was followed by a 709, then by a 7090, and finally by a 7094. IBM also gave MIT the services of some highly skilled Systems Engineers and Customer Engineers, who formed its MIT Liaison Office, which was housed at the MIT Computation Center.

As CTSS evolved, Professor Corbató and his students and colleagues began to encounter problems that they knew were better addressed by hardware than by software, so they asked IBM for modifications to their processor. The IBMers in the Liaison Office had the job of finding engineers in Poughkeepsie to build the hardware extensions that MIT had determined were necessary to do time-sharing properly. By the time CTSS was in full production in 1963, the 7090 at MIT had been modified to have a second memory bank (32K words), an address relocation register, and memory protection. With these extensions to the hardware, Corbató's group was able to build CTSS into the system that became the exemplar for time-sharing systems.

own in their logical organization; they include the independently developed BBN system for the PDP-1, the recently initiated work at IBM (by A. Kinslow) on the 7090 computer, and the plans of the System Development Corporation with the Q32 computer.

To establish the context of the present work, it is informative to trace the development of time-sharing at MIT. Shortly after the first paper on time-shared computers, by C. Strachey at the June 1959 UNESCO Information Processing Conference, H.M. Teager and J. McCarthy at MIT delivered an unpublished paper *Time-Shared Program Testing* at the August 1959 ACM Meeting. Evolving from this start, much of the time-sharing philosophy embodied in the CTSS system has been developed in conjunction with an MIT preliminary study committee (initiated in 1960), and a subsequent working committee. The work of the former committee resulted, in April 1961, in an unpublished (but widely circulated) internal report. Time-sharing was advocated by J. McCarthy in his lecture, given at MIT, contained in *Management and the Computer of the Future* (MIT, 1962). Further study of the design and implementation of man-computer interaction systems is being continued by a recently organized institution-wide project under the direction of Professor Robert M. Fano.

⁶ “Our customers often complained that the most difficult thing about having a computer was hiring somebody who could run it. They'd ask for help, we couldn't provide all those technicians ourselves, and there was not a single university with a computer curriculum. Sometimes we even found ourselves in a position where we had to hold back from taking a customer's order. So I went up to MIT in 1955 and urged them to start training computer scientists. We made a gift of a large computer and the money to run it, and they shared that machine with ten other schools in the Northeast. For the 650, we adapted a very aggressive college discount program that existed for our punch-card machines: you could get 40 percent off for setting up a course in either business data processing or scientific computing, and 60 percent off for setting up courses in both. I put these education policies near the top of the list of IBM's key moves, because within five years there was a whole new generation of computer scientists who made it possible for the market to boom.” (T.J. Watson, Jr., *Father, Son, and Co.: My Life at IBM and Beyond*, Bantam Books, New York, 1990, pp. 244-5.)

⁷ It appears that (without a clear directive from Corporate management) IBM's Cambridge Branch Office decided to interpret Watson's original grant to MIT as authorization for them to upgrade the system at MIT whenever IBM produced a more powerful computer.

A CMS user would find the log of a CTSS session fairly easy to follow. Commands were composed of 6-character, blank-delimited tokens. Files were referenced by their file name and file class. File attributes, such as permanent, read-only, *etc.*, were determined by the file mode. Some commands, such as “START”, “LOAD”, “RENAME”, and “LISTF”, would be quite familiar. The system typed “READY” when it completed the processing of a command. (It also typed “WAIT” when it started the processing of a command, so response time was obviously not what it is today.)

Nor would the system internals (as described in the 1963 CTSS user’s guide) be entirely strange:

The consoles of CTSS form the foreground system, with computation being performed for the active console users in variable-length bursts, on a rotation basis, according to a scheduling algorithm. The background system is a conventional programming system (slightly edited for the time-sharing version) which, at the least, operates whenever the foreground system is inactive, but which may also be scheduled for a greater portion of the computer time. The entire operation of the computer is under the control of a supervisor program which remains permanently in the 32,768 word A-bank of core memory; all user programs are either kept in the 32,768 word B-bank of core memory, or swapped in and out of the disk (or drum) memory as needed.⁸

The supervisor program ... functions include: handling of all input and output for the consoles; scheduling of console-initiated (foreground) and offline-initiated (background) jobs; temporary storage and recovery of programs during scheduled swapping; monitoring of all input and output from the disk, as well as input and output performed by the background system; and performing the general role of monitor for all foreground jobs. These tasks can be carried out by virtue of the supervisor’s direct control of all trap interrupts, the most crucial of which is the one associated with the Interval Timer Clock.⁹

By trapping interrupts, the CTSS supervisor controlled and isolated users in a manner very similar to the way the VM Control Program does this same thing today. CTSS users could request supervisor services by causing a protection exception, in much the same way that we use the CMS SVC and CP DIAGNOSE instructions today.

B. The Births of System/360, Project MAC, and the Cambridge Scientific Center

While CTSS was being developed in Cambridge, in Poughkeepsie IBM was designing the new family of computers on which it had staked its future, System/360. MIT by then was committed to time-sharing and was providing CTSS services to several other New England universities as well as to its own users. At MIT, it was “no longer a question of the feasibility of a time-sharing system, but rather a question of how useful a system [could] be produced”.¹⁰ The IBMers in the MIT Liaison Office and the Cambridge Branch Office, being well aware of what was happening at MIT, had become strong proponents of time-sharing and were making sure that the System/360

⁸ Corbató *et al*, *Ibid.*, p. 2.

⁹ Corbató *et al*, *Ibid.*, p. 8.

¹⁰ F.J. Corbató, in P.A. Crisman, ed., *The Compatible Time-Sharing System: A Programmer’s Guide*, second edition, MIT Press, Cambridge, Mass., 1965, p. 1.

designers knew about the work that was being done at MIT and understood the purpose of the modifications to the 7090. They arranged for several of the leading System/360 architects to visit MIT and talk with Professor Corbató. However, inside IBM at that time there was a strong belief that time-sharing would never amount to anything and that what the world needed was faster batch processing. MIT and other leading-edge customers were dismayed, and even angered, on April 7, 1964, when IBM announced System/360 *without* address relocation capability.

The previous Fall, MIT had founded Project MAC¹¹ to design and build an even more useful time-sharing system based on the CTSS prototype. Within Project MAC, Corbató and others were to draw on the lessons they had learned from CTSS to build the Multics system. The basic goal of the Multics project “was to develop a working prototype for a computer utility embracing the whole complex of hardware, software, and users that would provide a desirable, as well as feasible, model for other system designers to study.”¹² At the outset, Project MAC purchased a second modified 7094 on which to run CTSS while developing Multics. It then requested bids for the processor on which Multics would run.

In February of 1964, IBM had sent Norm Rasmussen to Cambridge to establish what became the Cambridge Scientific Center (CSC). Cambridge in the 1960s was an exciting place, full of ferment. It was a congenial place for Rasmussen, who was very much a man of that era, and it was a congenial assignment, as well, because Rasmussen was eager to demonstrate that science has a role to play in the building of good software.

Rasmussen arranged for space for the Scientific Center in the same building as Project MAC, 545 Technology Square.¹³ (For many years after that, the Scientific Center programmers and the Project MAC programmers would remain on friendly terms and would occasionally get together in the bar on the ground floor of that building after work.)

All of IBM’s contractual relationships with MIT were turned over to the new Scientific Center to administer. The Scientific Center was also expected to take the lead in making IBM “respectable” to the academics. So, only weeks after his arrival in Cambridge, Rasmussen had to deal with MIT’s very negative reaction to System/360. Within days of the System/360 announcement, the chief S/360 architect, Gene Amdahl, came to Cambridge to meet with Professor Corbató and his colleagues, but that meeting seems only to have made matters worse.

As a loyal IBMer, Rasmussen was deeply embarrassed by IBM’s failure to heed the advice of such an important customer, and he became determined to make things right, to do whatever was necessary to make System/360 right for MIT and other customers. To do that, he knew that he would need very talented people, so he set about attracting the best people he could find. He was fortunate to be able to start by taking over the staff of IBM’s MIT Liaison Office. From the Liaison Office came two very skilled Systems Engineers, Les Comeau and John Harmon, as well as a quiet, unassuming Customer Engineer named Fritz Giesin, who would come to be treasured

¹¹ “MAC” may have been an acronym for “Machine-Aided Cognition” or “Multiple-Access Computing” or “Man and Computer”.

¹² F.J. Corbató, in E.I. Organick, *The Multics System: An Examination of Its Structure*, The MIT Press, Cambridge, Mass., 1972, p. ix.

¹³ The building was a hotbed of time-sharing activity. “At one time in the mid-60’s, I counted ten or fifteen time-sharing systems being coded or tested or accessed in Tech Square.” (J.B. Harmon, private communication, 1989.)

by generations of programmers at the Center.¹⁴ Next came another excellent IBM programmer, Ron Brennan, from the Federal Systems Division. Shortly after that, one of the seven CTSS authors, Lyndalee Korn, left MIT to join the Center.

One of the first jobs for the staff of the new Center was to put together IBM's proposal to Project MAC. In the process, they brought in many of IBM's finest engineers to work with them to specify a machine that would meet Project MAC's requirements, including address translation. They were delighted to discover that one of the lead S/360 designers, Gerry Blaauw, had already done a preliminary design for address translation on System/360.¹⁵ Address translation had not been incorporated into the basic System/360 design, however, because it was considered to add too much risk to what was already a very risky undertaking.

The machine that IBM proposed to Project MAC was a S/360 that had been modified to include the "Blaauw Box". This machine was also bid to Bell Labs at about the same time. It was never built, however, because both MIT and Bell Labs chose another vendor. MIT's stated reason for rejecting IBM's bid was that it wanted a processor that was a main-line product, so that others could readily acquire a machine on which to run Multics. It was generally believed, however, that displeasure with IBM's attitude toward time-sharing was a factor in Project MAC's decision.

Losing Project MAC and Bell Labs had important consequences for IBM. Seldom after that would IBM processors be the machines of choice for leading-edge academic computer science research. Project MAC would go on to implement Multics on a GE 645 and would have it in general use at MIT by October, 1969. Also in 1969, the system that was to become UNIX¹⁶ would be begun at Bell Labs as an offshoot and elegant simplification of both CTSS and Multics, and that project, too, would not make use of IBM processors.

But getting back to the summer of 1964: Norm Rasmussen had just begun his fight to make System/360 acceptable to the academics and was not having an easy time of it. During that summer, Professor Corbató (a man widely known for his gentlemanliness) published a Project MAC Report containing a devastating analysis of the weaknesses of the S/360 as a machine on which to implement a time-sharing system.¹⁷ Other customers were also expressing concern about the lack of time-sharing capability in System/360. In August, SHARE's Advanced Planning Division presented a survey of the currently operating on-line programming systems. One of the speakers was Fernando Corbató, who emphasized the potential for growth of the computing industry due to time-sharing.¹⁸

¹⁴ For the next twenty-five years, Giesin would perform daily hardware miracles in support of the Cambridge programmers. "Fritz was a stalwart of the Center, very dependable, Santa Claus in his goodie lab!" (J.B. Harmon, private communication, 1990.)

¹⁵ G.A. Blaauw, *Relocation Feature Functional Specification*, June 12, 1964. "Nat Rochester (one of the designers of the 701) told us, 'Only one person in the company understands how to do address translation, and that's Gerry Blaauw. He has the design on a sheet of paper in his desk drawer.'" (R.J. Brennan, private communication, 1989.)

¹⁶ UNIX is a registered trademark, licensed exclusively by X/Open Company Ltd.

¹⁷ F.J. Corbató, *System Requirements for Multiple Access, Time-Shared Computers*, Project MAC Report MAC-TR-3, undated (probably August, 1964).

¹⁸ "Corbató noted that FORTRAN probably added a factor of 10 to the number of computer users

Rasmussen's response to all this was to decide that the Cambridge Scientific Center would write a time-sharing system for System/360.

Meanwhile, inside Project MAC, Bob Creasy was upset by the inability of his colleagues to come to terms with IBM. He was impressed by the promise of machine upward compatibility offered by S/360,¹⁹ and he wanted Multics to be a mainstream system. When he heard that Rasmussen intended to build a time-sharing system based on S/360 and needed someone to lead the project, Creasy also left MIT to move to the Scientific Center.

Inside IBM, losing the Project MAC bid was immediately recognized as a serious problem. A corporate task force was formed to get the company into a position to be able to win bids for time-sharing systems.²⁰ The task force was composed of the most knowledgeable time-sharing people from around the company. CSC was represented by Rasmussen, Harmon, Creasy, and Comeau. Other task force members included Andy Kinslow, who had written an experimental time-sharing system called BOSS ("Big Old Supervisory System") for the 7090, and John Moressey, the author of Quicktran. The products of this task force were rough specifications for a new processor, which would incorporate the Blaauw Box and be called the S/360 Model 67, and for a new operating system, which would be called TSS (the Time-Sharing System). IBM's management accepted the recommendations of the task force and put Andy Kinslow in charge of TSS Development. Rasmussen and his staff soon prepared a successful proposal for a 360/67 for MIT's Lincoln Laboratory.²¹

and that time sharing will at least equal this factor of 10 and probably surpass it." (*Proceedings of SHARE XXIII*, August, 1964, p. 3-18.) The systems described in addition to CTSS were JOSS, Quicktran, and SDC's system for the Q32.

¹⁹ Creasy had, of course, spotted the most important aspect of the System/360 announcement, that programs written for one model of S/360 would run on any other model as long as they contained no timing-dependent code. From the System/360 "blue letter" (April 7, 1964):

Whatever your customer's data handling requirements are now, whatever they will be in the foreseeable future, the System/360 can be custom-fitted to perform his job. In fact, this amazing new system makes possible, for the first time in the industry, a truly long-range growth plan for all customers. For it can accommodate virtually any combination of processing and computing functions. And can expand in easy, economical steps as the customer's needs change—with little or no reprogramming, no retraining of personnel, no disruption of service.

The decision to make System/360 a family of processors with the same instruction set throughout can safely be said to have made IBM's fortune.

²⁰ The task force was known as the "Flewellin House Task Force", for the building on the grounds of IBM Research at Yorktown Heights in which it met.

²¹ "Lincoln had a role in the design of the time-sharing machine. I have a copy of IBM's response to Lincoln's Request for Quotation, which specified a Model 66. This machine was later to become the 360/67, but I don't know why the model number changed. A group of six sites (Lincoln Lab, University of Michigan, Carnegie University, Bell Labs, General Motors, and Union Carbide, I believe) had a non-disclosure agreement for the development of the 360/66. This group was called the 'Inner Six'. At one meeting in Yorktown Heights, we met with IBM people to discuss relocation hardware. We discussed whether an address should be

C. CP-40 and CMS

In the Fall of 1964, the folks in Cambridge suddenly found themselves in the position of having to cast about for something to do next. A few months earlier, before Project MAC was lost to GE, they had been expecting to be in the center of IBM's time-sharing activities. Now, inside IBM, "time-sharing" meant TSS, and that was being developed in New York State. However, Rasmussen was very dubious about the prospects for TSS and knew that IBM must have a credible time-sharing system for the S/360. He decided to go ahead with his plan to build a time-sharing system, with Bob Creasy leading what became known as the CP-40 Project.

The official objectives of the CP-40 Project were the following:

1. The development of means for obtaining data on the operational characteristics of both systems and application programs;
2. The analysis of this data with a view toward more efficient machine structures and programming techniques, particularly for use in interactive systems;
3. The provision of a multiple-console computer system for the Center's computing requirements; and
4. The investigation of the use of associative memories in the control of multi-user systems.²²

The project's real purpose was to build a time-sharing system, but the other objectives were genuine, too, and they were always emphasized in order to disguise the project's "counter-strategic" aspects.

Rasmussen consistently portrayed CP-40 as a research project to "help the troops in Poughkeepsie" by studying the behavior of programs and systems in a virtual memory environment. In fact, for some members of the CP-40 team, this was the most interesting part of the project, because they were concerned about the unknowns in the path IBM was taking. TSS was to be a virtual memory system, but not much was really known about virtual memory systems. Les Comeau has written:

Since the early time-sharing experiments used base and limit registers for relocation, they had to roll in and roll out entire programs when switching users....Virtual memory, with its paging technique, was expected to reduce significantly the time spent waiting for an exchange of user programs.

31 or 32 bits. We eventually voted and recommended 31 bits. We also discussed the design of the relocation register and had some heated discussions with the IBM team. The Inner Six met with IBM representatives behind closed doors at a SHARE meeting. We six sites discussed various features of TSS and made recommendations to IBM. This was the beginning of the SHARE TSS Project." (J.M. Winett, private communication, 1990.)

²² R.J. Adair, R.U. Bayles, L.W. Comeau, and R.J. Creasy, *A Virtual Machine System for the 360/40*, IBM Cambridge Scientific Center Report 320-2007, Cambridge, Mass., May, 1966.

What was most significant was that the commitment to virtual memory was backed with no successful experience. A system of that period that had implemented virtual memory was the Ferranti Atlas computer, and that was known not to be working well. What was frightening is that nobody who was setting this virtual memory direction at IBM knew why Atlas didn't work.²³

Creasy and Comeau spent the last week of 1964²⁴ joyfully brainstorming the design of CP-40, a new kind of operating system, a system that would provide not only virtual memory, but also virtual *machines*.²⁵ They had seen that the cleanest way to protect users from one another (and to preserve compatibility as the new System/360 design evolved) was to use the System/360 *Principles of Operations* manual to describe the user's interface to the Control Program. Each user would have a complete System/360 virtual machine (which at first was called a "pseudo-machine").²⁶

The idea of a virtual machine system had been bruited about a bit before then, but it had never really been implemented. The idea of a virtual S/360 was new, but what was really important about their concept was that nobody until then had seen how elegantly a virtual machine system could be built, with really very minor hardware changes and not much software.

²³ L.W. Comeau, "CP-40, the Origin of VM/370", *Proceedings of SEAS AM82*, September, 1982, p. 40.

²⁴ Creasy had decided to build CP-40 while riding on the MTA. "I launched the effort between Xmas 1964 and year's end, after making the decision while on an MTA bus from Arlington to Cambridge. It was a Tuesday, I believe." (R.J. Creasy, private communication, 1989.)

²⁵ R.J. Creasy, *General Description of the Research Time-Sharing System with Special Emphasis on the Control Program*, IBM Cambridge SR&D Center Research Time-Sharing Computer Memorandum 1, Cambridge, Mass., January 29, 1965. L.W. Comeau, *The Philosophy and Logical Structure of the Control Program*, IBM Cambridge SR&D Center Research Time-Sharing Computer Memorandum 2, Cambridge, Mass., April 15, 1965.

²⁶ For the first few weeks, the CSC people referred to their concept as a "pseudo-machine", but soon adopted the term "virtual machine" after hearing Dave Sayre at IBM Research use it to describe a system he had built for a modified 7044. Sayre's M44 system was similar to CP-40, except for the crucial difference of not providing a control program interface that exactly duplicated a real machine. The CP-40 team credited Sayre with having "implanted the idea that the virtual machine concept is not necessarily less efficient than more conventional approaches." (L. Talkington, "A Good Idea and Still Growing", *White Plains Development Center Newsletter*, vol. 2, no. 3, March, 1969.) "The system built by Dave Sayre and Bob Nelson was about as much of a virtual machine system as CTSS—which is to say that it was close enough to a virtual machine system to show that 'close enough' did not count. I never heard a more eloquent argument for virtual machines than from Dave Sayre." (R.J. Creasy, private communication, 1990.)

²⁷ "Dick Bayles was not only a great programmer, he was also the fastest typist I have ever seen." (W.J. Doherty, private communication, 1990.) "When Dick Bayles sat down [at a keypunch], he wrote code as fast as it could punch cards. Yes, the machine was slower than Bayles composing code on the fly." (R.J. Creasy, private communication, 1989.)

Creasy and Comeau were soon joined on the CP-40 Project by Dick Bayles,²⁷ from the MIT Computation Center, and Bob Adair, from MITRE. Together, they began implementing the CP-40 Control Program, which sounds familiar to anyone familiar with today's CP. Although there were a fixed number (14) of virtual machines with a fixed virtual memory size (256K), the Control Program managed and isolated those virtual machines in much the way it does today.²⁸ The Control Program partitioned the real disks into minidisks and controlled virtual machine access to the disks by doing CCW translation. Unit record I/O was handled in a spool-like fashion. Familiar CP console functions were also provided.

This system could have been implemented on a 360/67, had there been one available, but the Blaauw Box wasn't really a measurement tool. Even before the design for CP-40 was hit upon, Les Comeau had been thinking about a design for an address translator that would give them the information they needed for the sort of research they were planning. He was intrigued by what he had read about the associative memories that had been built by Rex Seeber and Bruce Lindquist in Poughkeepsie, so he went to see Seeber with his design for the "Cambridge Address Translator" (the "CAT Box"),²⁹ which was based on the use of associative memory and had "lots of bits" for recording various states of the paging system.

Seeber liked the idea, so Rasmussen found the money to pay for the transistors and engineers and microcoders that were needed, and Seeber and Lindquist implemented Comeau's translator on a S/360 Model 40.³⁰ Comeau has written:

Virtual memory on the 360/40 was achieved by placing a 64-word associative array between the CPU address generation circuits and the memory addressing logic. The array was activated *via* mode-switch logic in the PSW and was turned off whenever a hardware interrupt occurred.

The 64 words were designed to give us a relocate mechanism for each 4K bytes of our 256K-byte memory. Relocation was achieved by loading a user number into the search argument register of the associative array, turning on relocate mode, and presenting a CPU address. The match with user number and address would result in a word selected in the associative array. The position of the word (0-63) would yield the high-order 6 bits of a memory address. Because of a rather loose cycle time, this was accomplished on the 360/40 with no degradation of the overall memory cycle.³¹

The modifications to the 360/40 would prove to be quite successful, but it would be more than a year before they were complete. Dick Bayles has described the process that he and Comeau and Giesin went through in debugging the modifications:

²⁸ R.A. Meyer and L.H. Seawright, "A Virtual Machine Time-Sharing System", *IBM Systems Journal*, vol. 9, no. 3, 1970, pp. 199-218.

²⁹ "The CAT was a jewel." (R.J. Creasy, private communication, 1989.)

³⁰ A.V. Lindquist, R.R. Seeber, and L.W. Comeau, "A Time-Sharing System Using an Associative Memory", *Proceedings of the IEEE*, vol. 54, no. 12, December, 1966, pp. 1774-9. The Center actually wanted a 360/50, but all the Model 50s that IBM was producing were needed for the Federal Aviation Administration's new air traffic control system.

³¹ Comeau, *op. cit.*, p. 41.

One of the fun memories of the CP-40 Project was getting involved in debugging the 360/40 microcode, which had been modified not only to add special codes to handle the associative memory, but also had additional microcode steps added in each instruction decoding to ensure that the page(s) required for the operation's successful completion were in memory (otherwise generating a page fault).

The microcode of the 360/40 comprised stacks of IBM punch card-sized Mylar sheets with embedded wiring. Selected wires were "punched" to indicate 1's or 0's. Midnight corrections were made by removing the appropriate stack, finding the sheet corresponding to the word that needed modification, and "patching" it by punching a new hole or by "duping" it on a modified keypunch with the corrections.³²

Back during that last week of 1964, when they were happily working out the design for the Control Program, Creasy and Comeau immediately recognized that they would need a second system, a console monitor system, to run in some of their virtual machines. Although they knew that with a bit of work they would be able to run any of IBM's S/360 operating systems in a virtual machine, as contented users of CTSS they also knew that they wouldn't be satisfied using any of the available systems for their own development work or for the Center's other time-sharing requirements. Rasmussen, therefore, set up another small group under Creasy to build CMS (which was then called the "Cambridge Monitor System"). The leader of the CMS team was John Harmon.³³ Working with Harmon were Lyndalee Korn and Ron Brennan.

Like Multics, CMS would draw heavily on the lessons taught by CTSS. Indeed, the CMS user interface would be *very* much like that of CTSS.

Since each CMS user would have his own virtual machine, CMS would be a single-user system, unlike CTSS. This was an important factor in the overall simplicity and elegance of the new system.³⁴ Creasy has written that one of the most important lessons they had learned from their CTSS experience was "the necessity of modular design for system evolution. Although [CTSS was] successful as a production system, the interconnections and dependencies of its supervisor design made extension and change difficult."³⁵

³² R.U. Bayles, private communication, 1989. "The Model 40 was a Hursley (UK) product, announced in 1964. It used the first programmable ROS (invented by Tony Proudman, I believe) called Transformer Read-Only Storage (developed in 1961/2). In the Model 40 the circuit on the Mylar sheets wound around 60 cores, hence allowing the storage of 60-bit words; the Model 40 had 4096 60-bit words. It was this re-programmable storage that made the Model 40 modifiable, as you describe." (M.F. Cowlshaw, private communication, 1990.)

³³ J.B. Harmon, *General Description of the Cambridge Monitor System*, IBM Cambridge SR&D Center Research Time-Sharing Computer Memorandum 3, Cambridge, Mass., May 12, 1965.

³⁴ Bob Creasy has commented, "Simplicity was important because of our limited resource. I didn't expect the design [of CMS] to hold for more than a couple of years. We recognized the importance of multiple processes in a single-user environment, but we could not afford the complexity. To put it another way, we weren't smart enough to make it simple enough." (R.J. Creasy, private communication, 1990.)

³⁵ R.J. Creasy, "The Origin of the VM/370 Time-Sharing System", *IBM Journal of Research and Development*, vol. 25, no. 5, September, 1981, p. 485.

CP-40 would be far more modular than CTSS, in that it would be divided into two independent components. In the words of Bob Creasy:

A key concept of the CP/CMS design was the bifurcation of computer resource management and user support. In effect, the integrated design was split into CP and CMS. CP solved the problem of multiple use by providing separate computing environments at the machine instruction level for each user. CMS then provided single user service unencumbered by the problems of sharing, allocation, and protection.³⁶

As the weeks went by and the real power of the virtual machine concept unfolded before them, their excitement grew. In discussing the decision to create exact replicas of real machines, Les Comeau has written, “It seems now that the decision to provide a Control Program interface that duplicated the System/360 architecture interface was an obvious choice. Although it was, given our measurement objective, it wasn’t, given our in-house interactive system objective.”³⁷ He credits “the strong wills and opinions of the group” for providing further motivation for selecting such a well-defined interface³⁸ between the CP and CMS components:

I think that most designers recognize the need for good separation of function in programming system design, but compromise becomes the rule very early in the effort. With the particular group assembled to build CP/CMS, the personalities reinforced that design principle, rather than compromising it.

The choice of an architected interface, the System/360, ... turned out to have been most fortunate. It permitted simultaneous development of CP and CMS; it allowed us to measure non-virtual systems, OS and DOS, in a virtual memory environment, and it also provided a high level of integrity and security.³⁹

CMS at this stage was rudimentary, consisting of structural elements on a blackboard (which Harmon updated daily).⁴⁰ “At that time the file system was viewed as supporting the needs of VM as well as the rest of CMS. Performance and simplicity were seen as major goals, because there was much concern about the overhead that would be introduced by the time-sharing function. Performance was achieved by keeping the function minimal and the path lengths short.”⁴¹

³⁶ Creasy, *op. cit.*, p. 485.

³⁷ Comeau, *op. cit.*, p. 43.

³⁸ “The CP group would not discuss what they were doing with us. They just said, ‘Read the *Principles of Ops.*’” (J.B. Harmon, private communication, 1989.) “I credit Bob Adair as the non-compromising standard bearer of [this aspect of] the CP design. Later, Ed Hendricks would join that very small group.” (R.J. Creasy, private communication, 1990.)

³⁹ Comeau, *op. cit.*, pp. 42-43.

⁴⁰ “John, understanding operating systems and programmers, constantly applied the KISS principle.” (R.J. Creasy, private communication, 1990.)

⁴¹ R.J. Brennan, private communication, 1989.

The file system design was clearly crucial. If CMS were to be attractive to use, it had to have “a convenient and simple method” for reading and writing disks. Ron Brennan left to work on TSS in April, 1966, but before he left Cambridge he had completed specifications for the CMS file system⁴² (which was then called the “Disk Service Program”) and had begun the implementation. In producing his design, Brennan drew upon his own knowledge of CTSS and upon Adair’s knowledge of the file system for STRETCH.⁴³ By September of 1965, file system commands and macros already looked much like those we are familiar with today: “RDBUF”, “WRBUF”, “FINIS”, “STATE”, *etc.* Many of the decisions that were to be key to the elegance of the CMS file system had already been made:

- CMS would use the simple filename-filetype-filemode naming convention, rather than using OS-like file names;
- Records would be mapped to fixed-size blocks;
- Records could be read or written by relative record number;⁴⁴
- From the user’s view, a file would be created simply by writing to it; and
- In many commands, the filemode could be defaulted, in which case the disks would be searched in a fixed order.

Another major focus of the CMS team was to determine the nature of the command language:

It was clear, based upon the experience gained with CTSS, that a user-friendly command language was key. Another thing we had learned was that the system had to be very forgiving, and although options were desirable, default-mode, non-required parameters were to be a paramount design consideration in CMS.⁴⁵

While his staff worked at designing the new system, Rasmussen sought ways to pay for it, taking advantage of every bit of luck that came his way. When IBM gave the 7094 to the MIT Computation Center, it retained the night shift on that machine for its own use. So, because the Scientific Center had inherited IBM’s contracts with MIT, Rasmussen “owned” eight hours of 7094 time per day. He traded part of that time to the Computation Center for CTSS time for his programmers to use in doing their development work. He “sold” the remainder to IBM hardware developers in Poughkeepsie, who badly needed 7094 time to run a design automation program that was critical for S/360 hardware development. With the internal funds he acquired this way, he paid for the modifications to the Model 40. Although he could not use these funds to pay for regular “head-count” employees, he could use them to pay for part-time employees, mainly MIT students, and to pay the salaries of IBMers who came to Cambridge to work on the system while remaining in the “head count” of some other part of the company. This method of funding the project with “unbudgeted revenues” had the advantage of allowing it to keep a very low profile.

⁴² R.J. Brennan, *Disk Service Program User Specifications*, IBM Cambridge SR&D Center Research Time-Sharing Computer Memorandum 4, Cambridge, Mass., September 16, 1965.

⁴³ “I gave Ron full responsibility for the file system and he did all the work himself—an excellent job.” (R.J. Creasy, private communication, 1990.)

⁴⁴ “Interestingly, some of the byproducts of this simple design, such as mapping the file records to fixed-size blocks on disk storage and retrieving the imbedded records by a relative block number, were precursors of the ‘fixed block architecture’ and the ‘direct file access method’ used by so many of IBM’s later systems.” (R.J. Brennan, private communication, 1989.)

⁴⁵ Comeau, *op. cit.*, p. 43.

Another bit of luck during the first year was a surprise visit to the Scientific Center by IBM's much-feared President, T. Vincent Learson (known informally as "The Hatchet Man"). Rasmussen was at lunch when Learson arrived, so Lyndalee Korn was given the task of entertaining him.⁴⁶ Lyndalee, unaware of Learson's reputation, charmed him thoroughly as she demonstrated CTSS and explained their plans for implementing a similar system on a S/360.⁴⁷ Following his visit, Learson arranged for a Model 40 to be diverted to CSC for the programmers to use while awaiting the arrival of their modified Model 40. Rasmussen sold the spare time on that machine, too.

Implementation of CP and CMS was begun in mid-1965, and the design continued to evolve rapidly during the implementation. Much of the early programming was done under CTSS using a S/360 assembler for code generation and a S/360 emulator for testing. Before Learson's Model 40 arrived, the programmers scavenged what time they could on various S/360 machines in Cambridge and Boston and on their modified Model 40 while it was sitting on the factory floor in Poughkeepsie. The CMS developers worked under the very early System/360 operating system, BPS, until they got enough of CMS together so that they could IPL it standalone from a real card reader (using a deck of the BPS loader followed by a deck of the CMS nucleus).⁴⁸ Once

⁴⁶ "I can remember Rasmussen saying, 'Oh, my God, I'm wearing a blue shirt!'" (L.K. Korn, private communication, 1990.) Bob Creasy has described his own encounter with Learson:

Wham!! The door to my oversized office burst open! Recognizing the large, rude man, I vaulted my fancy desk and immediately introduced him to my life insurance, an M.I.T. student on a work-study program with whom I was conducting an evaluation.

"Mr. Learson, welcome to Cambridge. What can I do for you?", I asked.

T. Vincent Learson, who as a leader in IBM struck fear into many an impure heart, was angry but now contained as he strolled across the carpet and looked out one of the windows at Technology Square. "Nice place you have here", he said as he fixed his eyes on the deep hole being dug for a new building. "Where's Rasmussen?", he asked.

"At lunch.", I replied, knowing where he wanted Norm to be at that moment. Learson then queried me as to why Norm was at lunch. Not knowing what was going on, I simply stated that in Cambridge we ate lunch around noon.

"I'll be back.", he said as he turned abruptly and left the office.

The M.I.T. student and I got back to business. Only a few minutes later I learned that earlier our receptionist had thrown gasoline on the smoldering Learson and pointed him toward my closed office door.

⁴⁷ CTSS was the only thing they had to demonstrate at that point. Lyndalee remembers that she showed Learson the interactive debugging function of CTSS (which she had written), explaining to him how much more quickly programs could be debugged interactively.

⁴⁸ "We went to loading from tape early. The first time I felt CMS was viable was when we had a card reader not working and the developers could keep working with CMS, but OS could not run without the reader." (J.B. Harmon, private communication, 1989.)

CMS was reasonably stable running standalone on a real machine, the developers worked under it as much as possible. Later, when CP became runnable, they moved CMS into a virtual machine and continued their development there.⁴⁹

The programming was done by the designers and a few other people. Claude Hans, who came to Cambridge on loan from a customer installation in France, did some of the early work on CMS, including getting enough of the OS simulation together to allow a FORTRAN compiler to run under CMS. His wife Danielle also worked on the system and on the documentation.

Another key participant was a 21-year-old MIT student named Stu Madnick, who began working on CMS in June of 1966. His first project was to continue where Brennan had left off with the file system. Drawing upon his own knowledge of the CTSS and Multics file systems, Stu extended the design of the file system⁵⁰ and got it up and running. He continued working part-time during the following school year and added several other important functions to CMS, including the first EXEC processor, which was originally called the COMMAND command. He had written a SNOBOL compiler for S/360, so he got that working under CMS, too. He needed a word processor to use to prepare papers for his courses, so he wrote Script, which was inspired by the CTSS Runoff program.⁵¹ Stu had been told that Dick Bayles (whom everybody acknowledged to be a brilliant programmer) had written the CMS editor in a week, so Stu wrote Script in a week. In 1968, he designed a new file system for CMS that anticipated important features of the UNIX file system, but that was never implemented. Stu was to continue working on CMS until 1972, when he finished school and had to get a real job. He is now a professor at MIT.

CP-40 and CMS were put into production with a regular user schedule in January, 1967. OS/360 had been made to run in a virtual machine by then,⁵² and the CSC staff was beginning to recognize and study some of the unpleasant truths about the behavior of virtual memory systems,

⁴⁹ “Bob Adair said to me, ‘If we can run two virtual machines, we can run *n*.’, but we actually had problems running the third one. After that, though, it worked.” (J.B. Harmon, private communication, 1990.)

⁵⁰ S.E. Madnick, *A Guided Tour Through the CMS File System, Part I—Data Bases*, June 1, 1967.

⁵¹ Madnick started with a program written by Dick Bayles that was very closely modelled on CTSS’s Runoff. The early Script system printed a startup message “SEM—Version x”. After Script had migrated to several sites, people started asking what “SEM” meant. Although they were told that it meant “Script Environment Module”, it was no coincidence that Madnick’s initials are “SEM”. Even today, Madnick’s name appears in the source for Waterloo Script. The comment “MADNICK IS NARCISSITIC” [*sic*] appears on a PRINT NOGEN statement added by somebody who was tired of seeing Madnick’s name displayed in every macro expansion.

⁵² “One of the milestones in the CP-40 Project was getting pre-release P (or was it Q?) of OS/360 to boot under CP for the first time—on the shop floor at the Boardman Road laboratories in Poughkeepsie. We discovered after a lot of trial and error that OS/360 violated one of the design principles of System/360—that no operating system software was to be timing dependent. It turned out that a portion of the boot code for OS/360 was timing dependent, and we had to ‘kludge’ up a two-stage simulation of the S/360 IPL process for it to boot.” (R.U. Bayles, private communication, 1989.)

such as page thrashing.⁵³

D. 360/67 and TSS

In August, 1965, IBM announced the System/360 Model 67 and TSS, the Time Sharing System.⁵⁴ TSS was an elegant and very ambitious system, but the customers who bought the early 67s soon found that TSS had serious stability and performance problems,⁵⁵ for it had been snatched from its nest too young. Then, unfortunately, IBM attempted to address the problems in TSS by “throwing bodies” at them, an approach that had already been found to be highly counter-productive in the case of IBM’s primary System/360 operating system, OS/MVT.⁵⁶

⁵³ “One afternoon, we weren’t getting any response from the 40. Three or four of us went to the Machine Room and saw that the disks were going like mad. At first, we didn’t understand it at all.” (J.B. Harmon, private communication, 1990.) Les Comeau later spent some time studying thrashing (L.W. Comeau, “Operating System/360 Paging Studies”, *IBM Storage Hierarchy System Symposium*, December, 1966) and concluded that thrashing had been the basic problem with the Ferranti Atlas machine.

⁵⁴ From the August 16, 1965, “blue letters”:

We are pleased to announce that the special bid restrictions have been removed from the System/360 Model 67—a system designed to let many remote users share time on the same high-performance computing facility. With its own powerful operating system, Model 67 provides the user with virtually instantaneous access to and response from the computer and, through multiprocessor configurations, with a high degree of system availability. Time-sharing provides a closer working relationship between the man with the problem and the computing power he needs to solve it.

IBM System/360 Model 67 is supported by a Time Sharing System monitor (TSS) that will take advantage of the unique capabilities of a multiprocessor system. The monitor performs dynamic relocation of problem programs using the dynamic address translation facilities of the 2067 Processing Unit, permitting response, within seconds, to many simultaneous users.

⁵⁵ “As a member of Lincoln Lab, I had the assignment of going to Kingston to run TSS before it was delivered to Lincoln. I worked the second and third shift using a 360/67 and ran TSS. It took a long time to IPL and did not stay up very long. In the Fall of 1967, a team of ten IBMers came to Lincoln to help us get TSS running in a production environment. They stayed for about three months before we converted to CP-67/CMS.” (J.M. Winett, private communication, 1990.)

⁵⁶ F.P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley Publishing Co., Inc., Reading, MA, 1975.

E. CP-67 and CMS

In September, 1966, without having access to a Model 67, the folks in Cambridge began converting CP and CMS to run on the 67. CMS was relatively straightforward to move to the 67, but it was also being enhanced rapidly. The CMS work was done by several people. Claude Hans worked on it for a while but left to return to France in 1967,⁵⁷ at which time he was replaced by Tom Rosato. Tom was to be the lead CMS developer for the next several years; it is to him that we owe much of what CMS is today. Another of the CMS developers of that era was a brilliant programmer named John Seymour, whose initials, “JAS”, are still to be found here and there throughout CMS today.⁵⁸

CP was more work to move to the 67 than CMS was, because the address translation hardware on the 67 was rather different from that on the 360/40. The CP design was also generalized substantially, to allow a variable number of virtual machines, with larger virtual memories. The first four programmers on CP-67 were Dick Bayles, Dick Meyer, Harit Nanavati, and John Seymour (who worked on CP for a few months before going over to CMS). Bayles managed the CP group and was the primary architect of CP-67.

In CP-40, the control blocks describing the virtual machines had been a hard-coded part of the nucleus. For CP-67, Bayles designed a new control block structure and added the concept of free storage, so that control blocks could be allocated dynamically. The inter-module linkage was also reworked, and the code was made re-entrant.

⁵⁷ Hans would continue to influence VM, however. He became the leader of a very strong group of VM people at the Grenoble Scientific Center. Rich Kogut, an American who worked for IBM France for ten years and who became a much-esteemed IBM representative to the SEAS VM Project, has described IBM France’s contributions to VM:

The University of Grenoble ran CP-67, which explains the interest of the Grenoble Scientific Center. In addition to Claude’s early work, another keystone was the work Alain Auroux did while on assignment to Cambridge. In the mid-70’s many people from Claude’s group influenced VM. Claude went to Endicott on assignment as a manager and was responsible for convincing IBM to work on and release XEDIT and EXEC2. He arranged for assignments for people from his former group: Xavier de Lamberterie, of course; Maurice Bellot, who did a lot of work on VM/BSE R1 CMS; Amine Zhiri who worked on an ill-fated project to combine VM and VS1; Jean-Pierre LeHeiget, who codeveloped the MSS prototype during a short assignment at Yorktown; Pierre Sauvage, who did a lot of the original 370/138 and 370/148 ECPS implementation. I participated in small projects; I remember helping out the Change Team for two weeks when the number of outstanding APARs skyrocketed, in 1976 I believe, and I somehow got hornswoggled into writing the 3340 alternate track support, which is still in the product. All of the early IBM reps to the SEAS VM Project were from IBM France, first Claude Hans, then Jean-Pierre LeHeiget, then myself. (R.M. Kogut, private communication, 1989.)

⁵⁸ For example, in DMSSVN:

```
SVC      X'CA'          (JAS -- 23 AUGUST 1967)
```

(Note that 202 is said to have been chosen for use as the CMS SVC because in hexadecimal that number forms the first two letters of the word “Cambridge”.)

Because Cambridge didn't yet have a Model 67, the developers had to modify CP-40 to simulate a Model 67, including the address translation hardware and the unique instructions in the Model 67's instruction set. One of these unique instructions was Search List (SLT). Bayles had designed the CP-67 control block structure to take advantage of SLT, so SLT was one of the instructions that CP-40 was modified to simulate. Early in 1967, having gotten a "CP-67" system together on the Model 40, the developers dumped the system to tape and took it to Yorktown, where they'd been allocated some Saturday test time on a real Model 67. They IPLed the system and watched it immediately flame out with an opcode exception on an SLT instruction. When they told the CE who was standing by that SLT was broken, he replied, "What's an SLT?" It was then that they discovered that the SLT instruction was an RPQ.⁵⁹ Soon after that, they began testing CP-67 on the Model 67 at MIT's Lincoln Laboratory, which did have SLT.⁶⁰

Lincoln's was one of the earliest 360/67s, and Lincoln was having severe problems with TSS. It was said to take ten minutes after an IPL to get the first user logged on, but the system's mean time to failure was less than that. So, when Lincoln's computer center manager, Jack Arnow, saw Dick Bayles IPL CP on the Lincoln machine and have the consoles up in less than a minute, he told IBM that he wanted that system. This demand rocked the whole company,⁶¹ but IBM was so desperate to keep a system at MIT that it would deny Lincoln nothing,⁶² so Lincoln was given CP and CMS, which they had in daily operation by April, 1967. There was speculation that the whole affair had been engineered by Norm Rasmussen,⁶³ who was known to have used various

⁵⁹ "The 360/67 SLT instruction RPQ was designed at Lincoln by Jack Nolan. He was interested in using it for database list processing. Once it was implemented, IBM found use for it to process lists in the CP nucleus. I don't know if it was ever used by TSS or for any applications program." (J.M. Winett, private communication, 1990.)

⁶⁰ The SLT simulation code was moved into CP-67, so that CP-67 would run on machines that didn't have the SLT instruction, but CP-67 customers were advised to order the instruction. Many didn't, however, because a processor could not have both 7090 emulation and the SLT instruction, and many customers needed 7090 emulation. SLT was gradually phased out of the system over the next few years.

⁶¹ "Throughout 1967 and very early 1968, IBM's Systems Development Division, the guys who brought you TSS/360 and OS/360, continued its effort to have CP-67 killed, sometimes with the help of some IBM Research staff. Substantial amounts of Norm Rasmussen's, John Harmon's and my time was spent participating in technical audits which attempted to prove we were leading IBM's customers down the wrong path and that for their (the customers'!) good, all work on CP-67 should be stopped and IBM's support of existing installations withdrawn. Luckily, SDD's own development efforts (on both the TSS and OS fronts) were so behind on function, performance, and schedule that we got away with it. But only because the customers—Lincoln, WSU, University of Alberta, Lockheed, and others—were so vocal and ultimately influential." (R.U. Bayles, private communication, 1989.)

⁶² "Lincoln Lab was untouchable." (N.L. Rasmussen, private communication, 1989.)

⁶³ The early CP and CMS developers were to acknowledge Rasmussen's role repeatedly over the years: "The Center Director, N.L. Rasmussen, is to be congratulated for creating and maintaining an atmosphere conducive to advanced systems research." (J.B. Harmon, *CMS User's Guide*, IBM Cambridge Scientific Center, Cambridge, Mass., March 16, 1967.) "The existence, and success so far, of the Cambridge Scientific Center VMCP, the CP/67 system, is due to the foresight of the Center's manager, Mr. Norman Rasmussen,..." (M.S. Field,

subterfuges to protect his “counter-strategic” CP/CMS project;⁶⁴ nevertheless, the Data Processing Division found the money to fund further development of CP-67 to provide temporary relief to Lincoln until TSS could be stabilized.

With the additional funding, the CSC staff grew. Among the new staff members acquired in 1967 were Mike Field, an SE from England who was to do good things to CP, such as adding support for “remote” consoles, so that people didn’t have to share the single system console, and Ed Hendricks, who had gained fame at MIT as the author of a really good “SpaceWar” game. Hendricks was to go on to do good work in graphics and in support of guest systems under CP, before becoming co-author of RSCS. (He also became reknowned for developing slick little multi-tasking monitors, each of which was said to be a descendant of his “SpaceWar” game.) In 1968, an almost twenty-year-old MIT student named Dave Tuttle began working at CSC, at first working with Ed Hendricks on various communications projects and on running guest systems, and later becoming a CP developer.

Lincoln had highly skilled system programmers of its own, who began enhancing CP and CMS as soon as they were delivered. The Lincoln and Cambridge people worked together closely and exchanged code on a regular basis. Lincoln programmers, including Frank Belvin, Hal Feinleib, and Bob Jay,⁶⁵ made a number of fundamental contributions to CP and CMS, contributions as basic as the GET and PUT commands in the editor.⁶⁶

One of the programmers at Lincoln during that period was Jim March. (Before I go any further with this story, I must assure you that Jim is a very good programmer, whose contributions include introducing the concept of stacked console input and designing the dual directory scheme of the CMS file system jointly with John Harmon.) One evening in 1968, Jim was working late and found he needed to sort a list of a few hundred items, so he threw together a “quick and dirty”

Multi-Access Systems: The Virtual Machine Approach, IBM Cambridge Scientific Center Report 320-2033, Cambridge, Mass., September, 1968.) “The complete support of N.L. Rasmussen, manager of the Cambridge center, was crucial to the success of the entire project.” (Creasy, *op. cit.*, p. 490.) “Norm provided a great environment for us.” (R.J. Creasy, private communication, 1990.)

⁶⁴ “Norm was told several times to kill CP-67, but he kept it alive under one guise or another. He really wanted to do the right thing for the company from a time-sharing viewpoint.” (J.B. Harmon, private communication, 1990.) “You have to have leadership. You have to be able to take risks. Being aware of quality is something they don’t teach you in business school.” (N.L. Rasmussen, private communication, 1989.)

⁶⁵ An IBM Service Bureau employee on assignment to Lincoln to work on TSS. (His surname has since been restored to Jesurum.)

⁶⁶ Joel Winett’s description of one of Lincoln’s many CP enhancements, a waiting list for logging on to the system, gives one an idea of the scarcity of computing resources in those days: “CP could support only a fixed number of users, but we had more terminals and users than could be serviced. So we allowed users to be logged on but not given running status. Each user was given a number in a queue (like a bakery number), and when someone logged off the next person was allowed to run. You could query the system to find out how far down in the queue you were. Sometimes you were as high as twentieth in the queue and had to wait hours to be enabled to run. (This was implemented by Rose O’Donnell.)” (J.M. Winett, private communication, 1990.)

sort/merge program and sorted his list and forgot all about it. Shortly after that, he left Lincoln to go to Interactive Data Corporation and didn't have the opportunity to use a "vanilla" CMS system again until 1977, when he moved to Bank of America. The Bank sent him to a GUIDE meeting at which there was much complaining about CMS SORT, so when he got home he printed off a listing and sat down to take a look. To his horror, he immediately recognized that he was the author of the reviled CMS SORT command. He was so embarrassed that he wrote a good CMS SORT and distributed it to all the members of the GUIDE VM Project.

It is important to keep in mind that the CP-40 and early CP-67 work was experimental. The people at Cambridge and Lincoln were simply creating a system for their own use (albeit with the hope that it might later become useful to others). Because of this, the environment in which the developers worked was conducive to experimentation, learning, and creativity. Creasy has described it as follows:

The design of CP/CMS by a small and varied software research and development group for its own use and support was, in retrospect, a very important consideration. It was to provide a system for the new IBM System/360 hardware. It was for experimenting with time-sharing system design. It was not part of a formal product development. Schedules and budgets, plans and performance goals did not have to be met. It drew heavily on past experience. New features were not suggested before old ones were completed or understood. It was not supposed to be all things to all people. We did what we thought was best within reasonable bounds. We also expected to redo the system at least once after we got it going. For most of the group, it was meant to be a learning experience. Efficiency was specifically excluded as a software design goal, although it was always considered. We did not know if the system would be of practical use to us, let alone anyone else. In January, 1965, after starting work on the system, it became apparent from presentations to outside groups that the system would be controversial. This is still true today.⁶⁷

At about the same time that Lincoln decided to run CP-67, another influential customer, Union Carbide, made the same decision.⁶⁸ In February, 1967, Union Carbide sent two of its system programmers, Bob Seawright and Bill Newell, to Cambridge to assist in the development of the system.⁶⁹ They both subsequently made important contributions to CP. Union Carbide's IBM SE, Love Seawright, was sent to Cambridge at the same time to learn to support the system. Love tackled the job of documenting the system, figuring out how it worked by using it and reading the listings. As her temporary assignment kept being extended, she worked at documenting, testing, debugging, and giving demonstrations. Later, she would package Version 1 of CP-67 and then help to support it by teaching courses, answering the hotline, and editing the *CP-67 Newsletter*.

By September, 1967, CP-67 was also running at IBM Research in Yorktown. Stu Toledano tells us that he used to drive to Cambridge from Yorktown every week or two to pick up the latest version of CP-67, which consisted of about 6,000 cards. He would talk with the people to make sure that what he was being given would really assemble and would then drive back to Yorktown.

⁶⁷ Creasy, *op. cit.*, p. 487.

⁶⁸ This decision is supposed to have been the result of a Union Carbide Vice President living next door to an IBM Vice President, who had told him about CP-40.

⁶⁹ Union Carbide also paid for the Model 67 that was later installed at Cambridge.

On one rainy trip back, his station wagon developed a leak and the 6,000 cards got warped, so he spent an evening using all the keypunches at Yorktown to duplicate CP and CMS. That was to be the beginning of a very strong CP/CMS tradition at Yorktown, which subsequently produced much excellent function that made its way into the VM product. It also led to the unleashing upon the world of the great Walt Doherty,⁷⁰ who was to go forth from Yorktown over and over again to convince customers' managers and IBM's of the economic value of sub-second response time.

In January, 1968, CP and CMS were installed on another Model 67, at Washington State University (WSU). At that time, WSU's primary system programmer was an undergraduate named Lynn Wheeler. Wheeler made numerous enhancements to CP and CMS while at WSU, particularly in the area of performance. In fact, he managed to reduce the CP overhead of running MFT under CP-67 by eighty percent, by such means as reducing path lengths, adding fast paths, replacing SVC linkages between CP modules with BALR linkages, improving the page replacement algorithm, and making CP modules pageable. After he graduated in February, 1970, Wheeler drove through snow and sleet straight to Cambridge, where he began a career that proved to be enormously beneficial to VM.⁷¹

By the time SHARE XXX convened in February, 1968, there were eighteen 360/67s installed.⁷² Most of these machines were running TSS (or trying to). SHARE's TSS Project had great *esprit de corps*, much as the VM Project later would. It was working very closely with IBM to improve

⁷⁰ “[In mid-1965, I] was assigned to be T.J. Watson’s man in TSSland at Mohansic. While there, I participated in a number of design meetings and met Lee [Varian], Ted Dolotta, Oliver Selfridge, Jack Arnow, Frank Belvin, and Joel Winett. The last four were at Lincoln Labs. Jack Arnow was Director of Computing there. Frank Belvin and Joel Winett worked for him. Oliver Selfridge was in the Psychology Department. Oliver suggested that I come work with them for a while on an editor project, called the Byte Stream Editor.... I went up to Lincoln for about a year. During that time I came to know Norm Rasmussen and the Cambridge Scientific Center. Sometime early in 1966, Pete Markstein and I visited the Cambridge Scientific Center to see the projects that were going on there at that time. CP-40 was one of those projects.... Peter suggested to Norm that he redo CP-40 for the Model 67 since it was becoming clear to us that an alternative to TSS might be required. But also, it was clear that the virtual machine concept was a very interesting concept for experimentation. Peter also offered my help to Norm to do that. Norm turned us down, saying that he wanted to build up the reputation of the Cambridge Scientific Center and wanted to do it himself.... I doubt if we were the first ones to make that suggestion to Norm, but it didn’t hurt.... When my time at Lincoln was up, I returned to Mohansic and became more convinced than ever that TSS wouldn’t work well, and certainly not on schedule.... Just before the August, 1967, SHARE it became obvious to me that TSS would never be viable in anything like its promised schedule, so I was asked by Bill Dorn (our Director of Computing at T.J.’s Place) and by Peter Markstein to bring CP/CMS from Cambridge down to Yorktown. Stu Toledano was assigned to this as well. We made many late night trips to Cambridge.... Mike Field and Harit Nanavati were very helpful to Stu and me.... So, in the Fall of 1967, CP/CMS was running on the Model 67 at Yorktown for about a hundred people. It was doing quite well.” (W.J. Doherty, private communication, 1990.)

⁷¹ Wheeler moved to AIX Development in August, 1987, and later left IBM.

⁷² *Proceedings of SHARE XXX*, 1968, p. 6-292.

TSS and was devoting tremendous energy to supporting the system.⁷³ On Monday of that SHARE week, IBM released a “blue letter” announcing the decommittal of TSS.

On Wednesday of that week (February 28), Dick Bayles and Harit Nanavati made a presentation on CP-67 and CMS to the TSS Project. The CP/CMS session was well attended and generated some interest, but most of the Project’s energy that week was devoted to protesting the decommittal, which had come along just when it appeared that TSS was finally beginning to achieve acceptable levels of stability and performance. One result of the CP/CMS presentation was that an IBM SE named Dick Newson, who had himself given a TSS presentation that week, stopped by Cambridge on his way home to Alberta, and became hooked on CP. Newson made his way back to Cambridge in February, 1969, where he began a very distinguished career as a CP developer.

Version 1 of CP-67 was released to eight installations in May, 1968, and became available as a TYPE III Program⁷⁴ in June. Almost immediately after that, two “spinoff” companies were formed by former employees of Lincoln Lab, Union Carbide, and the IBM Cambridge Scientific Center, to provide commercial services based on CP/CMS. Dick Bayles, Mike Field, Hal Feinleib, and Bob Jay went to the company that became National CSS.⁷⁵ Harit Nanavati, Bob Seawright, Jack Arnow, Frank Belvin, and Jim March went to IDC (Interactive Data Corporation). Although the loss of so many talented people was a blow, the CSC people felt that the success of the two new companies greatly increased the credibility of CP-67.⁷⁶ Rip Parmelee

⁷³ The TSS users may have been the most thoroughly organized group of customers that IBM has ever had to deal with. The following is from a November, 1966, letter from Oliver Selfridge, of Lincoln Lab, to other TSS users proposing a formal collaboration between TSS sites (which was subsequently implemented, except for the networking):

This memo is concerned with the development of a collaborating community of TSS/67 systems. We propose that each organization possessing such a machine resolve to follow these guidelines in building and using software systems. We propose that the 67s form a *de facto* net for testing and exchanging software systems on-line. We are sure that a little planning and a fair amount of cooperation can lead to a vast increase in the power of our software and hardware, and thus in our own powers and capabilities.

Each organization should coordinate its plans for software systems so as to avoid duplication. Some clearinghouse should be set up to communicate such plans and to aid coordination. It is strongly urged that coordination be started too early rather than too late. It will be found that doing so will not hinder experimentation but will in fact tend to make it more useful in the long run.

⁷⁴ “CP-67/CMS is a Type III (IBM employee contributed) Program, and has not been submitted to any formal test. Type III Programs are provided by the IBM Corporation as part of its service to customers, but recipients are expected to make the final evaluation as to the usefulness of the programs in their own environment. There is no committed maintenance for Type III Programs, nor does IBM make any warranty, expressed or implied, as to the documentation, function or performance of such programs.” (*An Introduction to CP-67/CMS*, IBM Cambridge Scientific Center Report 320-2032, May, 1969.)

⁷⁵ One of the founders of National CSS was Dick Orenstein, one of the authors of CTSS.

became the manager of CP development when Dick Bayles left.

The TSS decommitment was rescinded in April, 1969, as the result of vigorous protests by customers, but by then some TSS sites had switched to CP/CMS. SHARE had formed a CP/CMS Committee under the leadership of Joel Winett, of Lincoln Lab,⁷⁷ and CP-67 was running at fifteen accounts. VM had begun what later came to be known as its “Doubtful Decade”.

Dick Meyer assumed responsibility for both CP and CMS Development in 1969; he was to remain in that position until he, too, left to go to IDC in 1974. Version 2 of CP-67 was released in June of 1969. The authors of Version 2 were Diane Boyd, Clark Frazier, Liz Levey, Stu Madnick, Dick Meyer, Dick Newson, Tom Rosato, Love Seawright, and John Seymour.⁷⁸ Version 2 included PL/I support and a new scheduler developed by Bob Jay at Lincoln Lab, which was a real improvement over the Version 1 scheduler, which had had no means of controlling page thrashing.⁷⁹

CP at this stage was still quite primitive. For example, it didn't have dynamic page slot allocation. When a user logged on, space for his entire virtual machine (typically 256K) was allocated on the paging drum, if there was room, or on disk if the drum was full. Those slots were then his until he logged off. This had the effect of causing users to get to work earlier and earlier, in order to acquire drum page slots. Bruce Marshall⁸⁰ was one of the earliest members of the SHARE CP-67 Project and remembers vividly how good it felt when he succeeded in

⁷⁶ “While it may sound self-serving, I believe IBM's belated decision to announce relocation hardware on the S/370 series was influenced in some part by the commercial success of NCSS and IDC. By the time the first relocating S/370 (the 370/148) was shipped, these two companies had about 50% of IBM's installed base of 360/67 systems—11 systems at NCSS in three data centers on two coasts and several at IDC.” (R.U. Bayles, private communication, 1989.)

⁷⁷ The SHARE and GUIDE CP/CMS Committees were formed at the same time, at the joint SHARE/GUIDE meeting in Atlantic City in the Fall of 1968. Love Seawright helped establish these committees and became the chief IBM liaison to them.

⁷⁸ The *Program Contribution Form* that came with Version 2 listed the authors' names as: D.R. Boyd, H.C. Frazier, Jr., E.H. Levey, S.E. Madnick, R.A. Meyer, S.R. Newson, T.D. Rosato, L.H. Seawright, and J.A. Seymour. And it contained the following acknowledgment:

The CP-67 authors wish to acknowledge the contributions made to the system by R.J. Adair, R.U. Bayles, L.W. Comeau, R.J. Creasy, and J.B. Harmon, who developed the basic virtual machine concept; R.P. Parmelee, who formerly managed the CP-67 effort; M.S. Field, J.A. Kelch, and H.M. Nanavati, formerly with the CP-67/CMS project; E.C. Hendricks, C.I. Johnson, and D. Tuttle for their work in the operation of OS/360 in a virtual machine; as well as several users of the system, in particular the staffs of the MIT Lincoln Laboratory and Washington State University.

⁷⁹ L.H. Wheeler, “VM Performance History”, *Proceedings of SEAS AM86*, October, 1986, pp. 215-224.

⁸⁰ Then at Perkin-Elmer.

modifying CP to do dynamic page slot allocation.

Version 3 of CP-67, the first version with Class A support, was released in November, 1970, to be followed by the last two releases, 3.1 and 3.2, in 1971 and 1972.⁸¹ Version 3 had several important performance enhancements, including free storage subpool support. Under Version 2, processors that didn't have the SLT instruction spent as much as 20 percent of their time on free storage management; Version 3 reduced that by a factor of 10.

Release 3.1 contained the first “Wheeler Scheduler” to be available to customers. It had feedback controls, “fair-share” type priority calculations, dynamically adjusted working set size predictions, and other such magic.

One important change the CMS developers made in Release 3.1 was to replace the SIO instructions in CMS with high-performance DIAGNOSE instructions, so that CP could provide CMS with fast-path I/O. This improved performance dramatically, but it meant that CMS would no longer run on the bare hardware.⁸² For better or for worse, the interface between CP and CMS was no longer defined strictly by the *Principles of Operations*.

CP-67 ultimately ran on 44 processors, about one-fourth of which were internal to IBM.⁸³ By the end of the CP-67 era, much had been learned about making a virtual machine operating system perform well; Version 3.1 supported sixty CMS users on a Model 67.

F. VM/370

On June 30, 1970, IBM announced System/370, *again* without address translation hardware, which was very discouraging to both TSS and CP/CMS customers.⁸⁴

⁸¹ Release 3.2 contained no new function; it was simply a maintenance release.

⁸² Another reason for removing CMS's standalone capability was to avoid the requirement for adding hardware diagnostic support to CMS for Field Engineering.

⁸³ The Cambridge Scientific Center would continue to innovate in CP, CMS, and other components of VM until IBM closed it on July 31, 1992. That date would also mark the retirement from IBM of Bob Creasy, Noah Mendelsohn, Dick Newson, Love Seawright, Tom Rosato, and Lynn Wheeler.

⁸⁴ Rumors abound that up until the last week before the original System/370 announcement, IBM had planned to announce relocation hardware. In fact, at least some S/370 Model 135s were shipped with “DAT” marked under the appropriate lamp in the PSW display.

“With respect to the availability of relocation on the System/370, my personal view is that the TSS debacle coupled with some OS/360 development team management who believed TSO could provide OS-compatible interactive data processing without virtual memory led to a long internal debate. In the end, I think it was CP's performance advantages over TSO coupled with successful non-IBM enhancements to CP that made the (former) DP Division actively press the development people to accede to relocation and VM. It was never an issue of the hardware not having the capability—there were hints of relocation in the 145 as well—but only the internal ‘we want one operating system’ debate that delayed it.” (R.U. Bayles, private communication, 1989.)

In May, 1971, IBM held a meeting of its TSS customers at the Westchester Country Club. At that meeting, IBM permanently decommitted TSS (although a S/370 version was subsequently made available to customers who already had the S/360 version). IBM also revealed its future hardware and software plans, including relocate on the entire S/370 line; two new systems that would use relocate (these became OS/VS1 and OS/VS2); and, in the longer term, an architectural extension to provide 31-bit addressing.⁸⁵

At the Westchester meeting, IBM also said that it hoped that customers would *not* need virtual machines. Despite that hope, development of a S/370 version of CP and CMS had begun in the summer of 1970, in Cambridge. Since nobody in IBM wanted to fund the development of a S/370 version of CP/CMS, an unorthodox approach to getting funding was required. As it happened, Field Engineering had just decided that it had too many Software Field Engineers,⁸⁶ and it was trying to find new IBM positions for one thousand of them. It was willing to pay the first two years of salary for each one who was given a new position, but it didn't want to have to pay for their relocation. The result of this was that CP/CMS Development got thirteen of the sharpest software FEs in the Boston area and didn't have to pay their salaries for two years.⁸⁷ This doubled the size of the development group.⁸⁸ It also made some Boston customers very unhappy. The President of State Street Bank is said to have phoned the President of IBM to complain that they'd lost three of their four software FEs, all of whom they had trained themselves.

CP/CMS Development had been split out of the Scientific Center for legal reasons following the "unbundling" announcement on June 23, 1969. However, the developers were still in the Tech Square building and were still working closely with people in the Scientific Center, including Bob Adair, Rip Parmelee,⁸⁹ Charlie Salisbury, and Alain Auroux, who was at Cambridge on loan from the Grenoble Scientific Center.

By the time VM/370 Release 1 was finished, the CP/CMS Development group, including the documenters, had grown to 110 people. These were the primary developers for VM/370 Release 1:⁹⁰

⁸⁵ TSS/360 had 32-bit addressing, although only 31 bits of address could be used, because of the signed arithmetic used by the BXH and BXLE instructions.

⁸⁶ Later known as PSRs.

⁸⁷ "The new people all arrived for training one day in April or May of 1971. Many of them were really thrilled at getting in on the ground floor of a new operating system." (R.A. Meyer, private communication, 1989.)

⁸⁸ "Without that funding, VM/370 wouldn't have happened." (R.A. Meyer, private communication, 1989.)

⁸⁹ In speaking of his days in Cambridge as a VM developer, Paul Tardif has said, "Rip was one of these magnificent people. He and Bob Adair were upstairs. Whenever we had a difficult problem, we would go upstairs and talk to Adair and Parmelee and get our thinking straightened out." (P. Tardif, private communication, 1989.)

⁹⁰ From an unpublished address by S.R. Newson at the "VM Fifteenth Birthday Party", SHARE 69, August, 1987.

CP Developers

Dick Newson, manager
Charlie Brackett
Larry Estelle
Ray Grein
Tom Heald
Ed Murray
John Seymour
Dave Thibodeau
Dave Tuttle
Charlie Weagle
Clyde Wildes
Carl Young

CMS Developers

Tom Rosato, manager
Fernando Arce
Bob Collins
Bob Downs
Paul Fay
Sharon Koblinsky
Alan Middendorf
Dick Milley
Jim Sullivan
Jim Walsh
Bobbie Wilson
Walt Wisnowski
John Xenakis

(Xenakis was the author of the COPYFILE “compiler” among other things and was known as “Captain Midnight”. Tom Rosato began calling him that because Xenakis was in the habit of working all night and then leaving lists of the new features in CMS taped to his colleagues’ doors where they’d find them when they got to work in the morning.)

While the CP developers concentrated on re-structuring CP, the Scientific Center worked on bootstrapping CP-67 onto a System/370. One of the products of this work was a version of CP-67 that would create System/370 virtual machines on a Model 67. This system became critical to the people who were writing MVS, who were way behind schedule, in part because they had so few prototype 370s on which to test.⁹¹ It may be fair to say that by saving the MVS

⁹¹ An IBM newsletter announced the awards given for the virtualization of System/370 on the 360/67 (“Cambridge Men Modified CP-67, Providing Tool for Developers”, *IBM News*, vol. 9, no. 15, August, 1972, p. 1.):

CAMBRIDGE, MASS.: The work of four men at the Scientific Center here begun almost two years ago had an important role in the development of the operating systems announced this month. The four have received Outstanding Contribution Awards for their work. Two of the men, Dr. Richard Parmelee and Alain Auroux, are now with IBM France in Paris and Grenoble. Charles Salisbury and Robert Adair remain with the Scientific Center staff.

The four extended and enhanced CP-67, the control program that provided virtual machine capability on a Model 67. Their enhancement of the program gave IBM developers access to virtual 370s on a Model 67 as well as a version of CP-67 which would run on System/370.

The central modifications to the program made by the IBMers were: support of the new dynamic address translation facility, additional control registers, and some of the new 370 instructions and features. It became a tool for many IBMers writing the new virtual machine and virtual memory operating systems. Their extensions provided a means of testing 370 programs on Model 67 hardware even before 370 hardware was available internally.

“Moving CP-67 from a 360 base to a 370 base meant that the CP/CMS Development

developers, VM saved itself. Once MVS Development became dependent upon VM to virtualize each new level of the architecture, VM became much harder to kill.⁹²

Alain Auroux did most of the actual coding and testing for the bootstrapping, but Rip Parmelee, Bob Adair, and Charlie Salisbury were also heavily involved in working out the design. When Auroux started, Cambridge was running a 360/67, not a S/370, and that 67 was a production system, so he had to avoid destabilizing it. “Vanilla” CP-67 systems created System/360 virtual machines, but they did not virtualize the 360/67; that is, they did not allow a guest to create its own virtual storage. Auroux’s first step was to modify CP-67 to create virtual 360/67s, which used 4K pages and 1M segments. Once he had convinced the Cambridge Operations Manager to run that as the production system, he could then proceed to develop a CP-67 that virtualized the System/370 architecture.

The System/370 relocation architecture was different from the 360/67 architecture; it allowed both 2K and 4K pages and both 64K and 1M segments. So, Auroux modified his modified CP-67 to support 64K segments and the new System/370 instructions. He ran that system second-level, so he could run a virtual S/370 third-level. He developed a prototype “CP-370” in that third-level virtual machine. Then, to test this CP-370’s virtualization of System/370 virtual memory, he had to run it both third- and fourth-level, with a couple of CMS machines running fifth-level. He remembers doing much of his testing from home at night using an “old, slow, noisy teletype”.⁹³ His prototype CP-370 had been debugged in simulation by the end of 1970. Late in January, 1971, just before Auroux was to return to France,⁹⁴ he and Bob Adair and Rip Parmelee took a

Group working on VM/370 had something to start running on their 145”, explains Dr. William Timlake, Scientific Center manager. “The people in SDD writing VS code also had something to use in developing virtual memory software.” In mid-1970, an SDD advanced system programming team headed by Russ Hamrick asked the four to assist in developing a virtual machine system for the 145. “We obtained the architectural specifications of the 370 advanced functions and together developed the initial code”, notes Adair.

Two prototype systems resulted: one to provide virtual 370s on the 67 (required for testing, since the advanced hardware was not available), the other to provide virtual machines on a System/370. A critical test of the modified program came when Auroux travelled to Endicott to try the program on a prototype Model 145. “That was one of high points”, recalls Salisbury. “Auroux loaded the disk pack onto the engineering model, a computer that had never run an advanced function operating system, and the system ran successfully. It demonstrated that software could be developed on a virtual machine for hardware not yet produced.” After that, use of the modified CP-67 programs spread to several locations where development activity was underway.

⁹² There is a widely believed (but possibly apocryphal) story that anti-VM, pro-MVS forces at one point nearly succeeded in convincing the company to kill VM, but the President of IBM, upon learning how heavily the MVS developers depended upon VM, said simply, “If it’s good enough for you, it’s good enough for the customers.”

⁹³ A. Auroux, private communication, 1989.

⁹⁴ Auroux was on assignment at CSC from July, 1969, to February, 1971. He has described his first few minutes at CSC as follows: “I remember my first day of assignment: when I arrived

copy of his system to Endicott so that they could test it on a prototype 370/145 with relocation hardware. It IPLed the first time.

A real S/370 (a 145 with relocate hardware) was finally delivered to Tech Square during the Fall of 1971 (amidst frantic security precautions for fear that nosy neighbors would figure out that a S/370 being delivered to CSC *must* have relocate hardware).⁹⁵ By then, the CP developers had incorporated Auroux's work into their own enhanced version of CP and were at last able to run their system first-level.⁹⁶ They first IPLed a full-function VM/370 CP one day in February, 1972. They packaged an alpha-test version for internal distribution on July 5, 1972. Thus, working at break-neck speed, the small group in Cambridge managed to get their system ready for IBM's spectacular System/370 Advanced Function announcements on August 2, 1972. On that day, IBM announced:

- Two new computers, the 370/158 and the 370/168;
- Address relocation hardware on all 370s; and
- Four new operating systems:⁹⁷
 - VM/370,
 - DOS/VS, a virtual storage version of DOS,
 - OS/VS1, a virtual storage version of MFT, and
 - OS/VS2, a virtual storage version of MVT.

On announcement day, VM/370 was up and running for demonstration purposes at all of IBM's Field Support Centers, unlike VS2.

At an announcement session at SHARE XXXIX the following week, IBM listed the advantages of virtual storage as:

- Elimination of artificial memory constraints;
- Enhanced processor storage utilization; and
- Enhanced machine accessibility for application program development and maintenance.

Particular emphasis was placed on the productivity gains IBM itself had achieved by doing OS maintenance and testing in virtual machines under CP-67 and VM/370.⁹⁸

at the Center, Norm Rasmussen, who was at the time the Center Manager, brought me in a dark meeting room, with a TV set in it, and told me: 'Sit down, we will discuss later'. I did so, and one minute later I watched on TV Apollo XI leaving for the first trip from Earth to the Moon!" (A. Auroux, private communication, 1989.)

⁹⁵ "Cambridge was not a good place for developing unannounced products. I can remember one evening at a bar in Cambridge having an MIT student tell me that one of his biggest objectives was to find out what IBM was doing with that 145 in 545." (R.A. Meyer, private communication, 1989.)

⁹⁶ At that point, the virtual memory architecture was still changing, and they had to track the changes in the instructions, which changed every couple of months when they got a new microcode load.

⁹⁷ OS/VS2 would ultimately be made available in two versions, which would be known as SVS and MVS. VM/370 would finally be withdrawn from marketing on April 24, 1989.

This is the text that went with the first few slides in the VM/370 announcement package:

To help you take advantage of the real and virtual storage capabilities of System/370, we are going to give you a presentation today on a new IBM product, Virtual Machine Facility/370, or VM/370.

Here is a prism. Consider for a moment what happens when a beam of light falls upon it...many colors evolve from one light source. Let me emphasize that point...many from one.

By way of analogy, think of the beam of light as an IBM System/370; the prism, as Virtual Machine Facility/370. The many colors produced by the prism from the one light source are now many virtual 370s produced by VM/370 from one real 370. And each virtual 370 has the capability to run its own programming system, such as OS, DOS, or CMS. Many from one...many virtual 370s from one real 370. And VM/370 makes it happen!⁹⁹

Further on in this “pitch”, IBM cited keeping your system programmers happy as being one of the big advantages of VM.

VM/370 was announced with two components, CP, the “Control Program”, and CMS, which was now to be called the “*Conversational Monitor System*”. VM/370 was shipped to the first customers at the end of November, 1972.

The design point for VM/370 Release 1 was a 512K 370/145. The largest machine that IBM had announced at that time was an 8M 168. The marketing forecasts for VM/370 predicted that no more than one 168 would ever run VM during the entire life of the product. In fact, the first 168 delivered to a customer ran only CP and CMS. Ten years later, ten percent of the large processors being shipped from Poughkeepsie would be destined to run VM, as would a very substantial portion of the mid-range machines that were built in Endicott. Before fifteen years had passed, there would be more VM licenses than MVS licenses.

Thirty-two CP-67 customers migrated to VM/370 Release 1. All of them blessed the CMS developers for having made the CMS/370 file system upward compatible. Old modules wouldn't work, however, because the user area had been moved from X'12000' to X'20000'.¹⁰⁰

⁹⁸ J.P. Hogan, “Value of Virtual Storage”, *Proceedings of SHARE XXXIX*, August, 1972, pp. 301-329.

⁹⁹ *IBM Virtual Machine Facility/370 (VM/370) Demonstration*, GV20-0388, IBM Corp., 1972.

¹⁰⁰ Another change to CMS in VM/370 Release 1 was the removal of the old “virtual RPQ device”, the pseudo-chronolog at virtual address 0FF, which was replaced by a DIAGNOSE instruction. Under CP-67, I/O to device 0FF returned a buffer containing the date, time of day, and the CPU utilization for the virtual machine. CMS tested for the existence of 0FF to decide whether it was running standalone. (I am told that at one stage when one IPLed CMS standalone or after having detached 0FF, it put out a message that said, “Aren't you embarrassed to be running CMS on a bare machine?”)

An important change to CMS in Release 1 was the introduction of the multi-level update capability in the UPDATE command. This made possible the VM source maintenance procedures we were to use and love for the next fifteen years. Multi-level UPDATE was the joint work of Dave Tuttle, from CP Development, and John Xenakis, from CMS Development.

Unlike CMS, CP was largely restructured for VM/370. Dick Newson designed the new control block structure. Just before Release 1 was announced, Dick drew this diagram of the CP control block logic (which makes an interesting contrast to the MVS control block flow of about the same era). To go with the new control block structure, Newson and Carl Young¹⁰¹ designed new register usage conventions, command scanning routines, and module linkage macros.

The most important new CP function in Release 1 of VM/370 was the ability to run VM under VM. Alain Auroux's virtualization of the 360/67 was never made available to customers, although customers, such as Bruce Marshall, had made the same modification even earlier than Auroux. In VM/370 Release 1, however, it was official. VM system programmers could test VM under VM. They could now go years at a time without having to take standalone test time early on weekend mornings. This became an important factor in instilling the passionate love that was to keep the VM community struggling to save VM through the coming dark years.¹⁰²

¹⁰¹ "Carl Young was responsible for writing vast portions of the code. He owned a third of the major components of the system." (P. Tardif, private communication, 1990.)

¹⁰² In 1973, the ACM held a conference on virtual systems. Robert Goldberg (who had also virtualized the 360/67 and who later became one of the founders of BGS Systems) described the reasons for the increasing interest in virtual systems:

The development of interest in virtual computer systems can be traced to a number of causes. First there has been a gradual understanding by the technical community of certain limitations inherent in conventional time-shared multi-programming operating systems. While these systems have proved valuable and quite flexible for most ordinary programming activities, they have been totally inadequate for system programming tasks. Virtual machine systems have been developed to extend the benefits of modern operating system environments to system programmers. This has greatly expedited operating system debugging and has also simplified the transporting of system software. Because of the complexity of evolving systems, this is destined to be an even more significant benefit in the future.

As a second point, a number of independent researchers have begun to propose architectures that are designed to directly support virtual machines, *i.e.* virtualizable architectures. These architectures trace their origins to an accumulated body of experience with earlier virtual machines, plus a set of principles taken from other areas of operating system analysis. They also depend upon a number of technical developments, such as the availability of low-cost associative memories and very large control stores, which now make proposals of innovative architectures feasible.

A third reason for the widespread current interest in virtual machines stems from its proposed use in attacking some important new problems and applications such as software reliability and system privacy/security. A final point is that IBM has recently announced the availability of VM/370 as a fully supported software

The group developing VM/370 Release 1 produced at least a thousand lines of code per person per month, to the astonishment of IBM's auditors, and they managed to enjoy themselves while they were doing it. Dick Jensen, who was later to earn our gratitude for writing Smart (the VM Real-Time Monitor), was doing "systems assurance". To get even with him for hassling them about their flow charts, the developers added "junk code" to the system to see if he would find it. He did find and remove the code that wished the various developers a happy birthday whenever a system was IPLed on one of their birthdays. He did not delete the code that typed "BONG BONG BONG BONG" across all the terminals at midnight. Dick Newson claims that his group also considered making the DIAL command reply, "Aren't you glad you use DIAL?", but they were afraid that would result in legal hassles.¹⁰³ While having their fun, they developed a good, clean system, with a tight, simple structure.

In the CP-67 days, if a customer had a problem, he simply phoned Cambridge and talked to the person responsible for that part of the system. Customers could also dial into the Cambridge system to download the current fixes. But VM/370 was supported by Field Engineering, so things were now more formal. CP and CMS Level II for the U.S. for VM/370 Release 1 was Lyn Hadley.¹⁰⁴ Lyn continues today to be the one to whom customers turn for help with their really bad bugs. In the early years, Lyn struggled to get debugging tools into the product; the release of the IPCS component was due largely to him.

Eight days after VM/370 was announced, the SHARE VM/370 Project had its first meeting, replacing the CP-67 Project. Six months later, the membership of the Project had trebled, and ten SHARE installations already had VM/370 in production.

product on System/370. With this action, IBM has officially endorsed the virtual machine concept and transformed what had been regarded as an academic curiosity into a major commercial product.

(R.P. Goldberg, *Proceedings of ACM SIGARCH-SIGOPS Workshop on Virtual Computer Systems*, March, 1973, pp. ii-iii.)

The ability to run a system under itself has remained relatively rare, despite the enormous benefits in increased system availability. IBM provided the "PolyASP" capability for the MVT subsystem ASP, but did not carry that capability forward into ASP's successor, JES3. Late in the life of TSS, there was much discussion of adding virtual machines to TSS, but only for running OS, not for running TSS itself. From a system programmer's point of view (at least, from this system programmer's point of view), UNIX's most glaring weakness is in not being virtualized.

¹⁰³ "DIAL" is the name of an American deodorant soap which has long been advertised with the slogan, "Aren't you glad you use DIAL? Don't you wish everyone did?"

¹⁰⁴ He and Terry Gibson, of IBM Canada, had been sent to Cambridge before the announcement to learn the system by reading dumps for the developers. Lyn tells us that one day he was given a really awful dump in which page 0 had been overlaid. He couldn't figure out how he was going to shoot this dump; he had found the trace table without benefit of the overlaid trace table pointers, but had no idea of how to determine the current trace table entry. So, he took the dump to Carl Young, who glanced through the trace table and then pointed out the current entry. When Lyn asked him how he had figured that out, Carl pointed to the next entry and said that that one couldn't logically follow the one before it; therefore, it must be the oldest entry. Lyn never did find out whether Carl was putting him on.

In 1971, as the Technology Square building became more and more crowded, portions of VM Development began moving to the New England Programming Center in Burlington, just outside Boston. The move was completed by January of 1973. The developers thrived in Burlington and delivered new function at an astonishing rate, *via* the service tapes, which were known as “PLC tapes”.¹⁰⁵ Some PLC tapes amounted to new releases. Release 1 PLC 09, for example, added CMS BATCH, console spooling, support for the 370/168 and the 2305 drum, and Carl Young’s “biased scheduler”, which replaced the primitive scheduler in the Release 1 base.

VM/370 Release 2 came out in April, 1974, and included support for several new devices, as well as the new Virtual Machine Assist (VMA) microcode. The primary author of VMA was Pete Tallman, a young CPU microcoder for the System/370 Model 158. VMA began as the result of a meeting in the office of Don Ludlow, the author of SuperZap, on January 23, 1973. Tallman had crashed the meeting because he thought it might be interesting. The topic was whether anything could be done to address the performance problems due to CP simulating the LRA instructions issued by virtual machines. The conclusion was that microcode enhancements could solve the problem. Working nights and weekends (because it wasn’t his real job), Tallman had a prototype ready to test two weeks after the meeting. That test showed a sixteen percent performance improvement with an OS/VS2 guest. When word of this got out, his telephone started ringing off the hook, as 158 sites throughout the company pleaded for the new microcode. Tallman has written:

We figured we might be on the right track and decided to continue on to other instructions. Tom Gilbert was able to guide my microcode development to the performance hot spots, and we did only the functions that would be relatively self-contained within the microcode. The tough parts were shadow-table update and the storage-key instructions, because of all the tables involved.... The whole VMA package as we know it was finished, with the current interface and set of functions, for an OS/VS2 benchmark on the evening of April 26, 1973. We halved the elapsed time needed to run jobs on the guest.¹⁰⁶

Release 2 also included support for 3270s—the now familiar “MORE” and “HOLDING” and “NOT ACCEPTED”—which was the work of Dick Newson, who had wandered around MIT watching how people worked with tubes and trying to figure out the best human factors. (He regrets not having thought of a RETRIEVE key.)

Release 2 PLC 05 added native support for the 3705, a project of Dave Tuttle’s. Release 2 PLC 11, in January, 1975, added the new component RSCS. Two months later, PLC 13 added VS1 handshaking and the CP monitor, which was the work of Peter Callaway.

In February, 1976, PLC 23 added another new component, IPCS, the Interactive Problem Control System. The primary component of IPCS was Dumpscan. It is difficult now to remember how revolutionary Dumpscan was when it first came out, how tremendously it increased our productivity. For many of us, reading CP dumps was practically a full-time job. Suddenly, we were able to search a dump at electronic speeds and no longer had to struggle with mountains of paper. Dumpscan was the brainchild of the late Dick Seymour. His original intent was not the creation of a tool to be used by customers or even by the Change Teams; Seymour was exploring the possibilities for automatic analysis of system failures. Dumpscan was an unexpected

¹⁰⁵ “PLC” stood for “Program Level Change”.

¹⁰⁶ P.H. Tallman, private communication, 1993.

byproduct of that study. The primary author of Dumpscan was John Shaw.¹⁰⁷ Larry Estelle, a Regional VM Specialist who had been a CP developer, demonstrated Dumpscan's potential as a service tool by dialing into a customer system and using Dumpscan to shoot dumps. Ultimately, Dumpscan was made available to customers, whereupon it quickly became the object of more customer enhancements than almost any other part of VM. Once people saw the usefulness of such a tool, other, far superior dump viewers soon appeared, while IBM allowed Dumpscan to waste away of neglect.

Also in February, 1976, Release 3 of VM/370 became available, including VMCF and support for 3350s and the new ECPS microcode.

Edgar (the "Display Editing System"), a program product full-screen editor written by Bob Carroll, also came out in 1976. Edgar was the first full-screen editor IBM made available to customers, although customers had previously written and distributed full-screen editors themselves, and Lynn Wheeler and Ed Hendricks had both written full-screen editors for 2250s under CMS-67.

G. Supporting the new VM users

By the mid-70s, VM was spreading rapidly, and education of new VM system programmers was becoming a problem. One important tool that helped the new VMers get "up to speed" rapidly was the excellent *Virtual Machine Facility/370 Features Supplement*, which was released in January of 1974, and was, in my view, the best manual IBM ever published. The author was Barbara McCullough.

During this period, one noticed geographic pockets of VM activity that corresponded to the presence of a believer inside IBM who was doing his or her best to promote and support VM. Many of these heroes and heroines should be cited here, but I can mention only a few of them. One especially influential person was Canada's "Mr. VM", Paul Tardif, an SE who had been a VM developer. Claude Hans, Rich Kogut, and others who had been at the Grenoble Scientific Center were a strong pro-VM force in Europe. Kogut was a much esteemed IBM representative to the SEAS VM Project, but left to return to the U.S. in 1982, after which the indispensable Adrian Walmsley succeeded him as SEAS rep.

After IBM closed the New England Programming Center in Burlington, many of the VM developers became SEs, some remaining in the Boston area, others moving elsewhere, and all of them spreading the word about VM. Love Seawright continued to be the primary liaison to the user groups and to play an important role in conveying customer concerns back to the development group. GUIDE rewarded her efforts with a button in her honor.

Despite the efforts of the heroes and heroines inside IBM, much of the support for new installations had to come from the user community. Most IBM branches were openly hostile to VM, and many used extreme measures to discourage customer managements from installing VM. Few branches provided good VM support, so new users were often in trouble.

Existing VM installations were conscious of TSS's fate and knew that VM would die unless it grew, so they built the necessary support structure:

¹⁰⁷ Mike Ness and Tom Pattison also worked on Dumpscan and other parts of IPCS.

- ***The VM Library:*** In 1973, the great guru Romney White volunteered the University of Waterloo to become the home of the VM Library, more familiarly known as “the Waterloo Tape”. Sandra Ward graciously took on the job of supporting this project and was able to report by the end of its first year that the Library had grown to four reels of tape. These tapes of tools, modifications, utilities, cookbooks, *etc.*, were contributed by VM people around the world and became vital to installations getting started in VM.
- ***Regional VM Users Groups:*** In December, 1973, Simcha Druck founded the Metropolitan VM Users Association (“MVMUA”) in New York. MVMUA is still going strong and required two large cakes to celebrate its twentieth anniversary. Other local users groups soon sprang up around the world, and many have been extremely active in supporting their members and new installations. The UK VM Users Group, whose slogan is “*VM Users do it Virtually All the Time*”, was founded in February, 1974. It is still very active and held a two-day celebration for its twentieth birthday a few years ago. In much of Europe, IBM marketing people were even more anti-VM than they were in North America, so the local user groups were even more essential there. *Le club Francais des utilisateurs VM* was founded by Frédéric Roux in 1977, with help and encouragement from Claude Hans. The Canada VM Users Group has been a force for good since its founding in 1978.
- ***VM Workshops:*** In 1977, Romney White came up with another great idea—that the University of Waterloo should hold the first of the annual VM Workshops. The Workshops allow VM people from all over to get together to spend a few days attending technical presentations and discussing their installations’ problems. Since Waterloo started the tradition, many other universities have hosted the Workshop. Many people have devoted their time to making the Workshops work, notably John O’Laughlin, Simcha Druck, Harry Williams, and Martha McConaghy. Martha will be the hostess for next year’s Workshop at Marist College.
- ***The SHARE VM New Users Project:*** In 1977, SHARE formed a project specifically to support new VM installations. For many years after that, the New Users Project contributed greatly to getting new VMers up to speed. Among the most notable of the VMers who worked for the New Users Project were “the fastest wit in the West”, JoAnn Malina, and her successors as New Users Project Manager, Roger Deschner and Donna Walker.

H. Making VM faster

Possibly even more important to new users was the work that the older users were doing to make the system run faster. Not many IBMers believed that VM’s performance problems could be solved, so not much was being done about those problems inside IBM.

Most installations had started running VM so that they could run two flavors of OS on one processor while migrating from DOS or MFT or MVT to a virtual storage system. Typically, then, the system programmer had fallen in love with VM and was doing everything he could to keep it around. He liked being able to test operating system changes during the middle of the day, and he really liked CMS. One of my earliest memories of a SHARE VM Group meeting is the CMS guru Dave Gomberg telling how he had made a mod that got back enough of his CPU that he’d be able to keep VM for four more months. Everyone cheered.

The process of making guest systems perform better began as soon as the customers got their hands on CP. Lynn Wheeler had done a lot of work on this while he was a student at Washington State, but he was by no means the only one who had worked on it. The CP-67 Project had

frequently scheduled sessions in which customers reported on modifications to CP and guest systems to make the guests run better under CP. These customers had measured and monitored their systems to find high overhead areas and had then experimented with ways of reducing the overhead.¹⁰⁸ Dozens of people contributed to this effort, but I have time to mention only a few.

Dewayne Hendricks¹⁰⁹ reported at SHARE XLII, in March, 1974, that he had successfully implemented MVT-CP handshaking for page faulting, so that when MVT running under VM took a page fault, CP would allow MVT to dispatch another task while CP brought in the page. At the following SHARE, Dewayne did a presentation on further modifications, including support for SIOF and a memory-mapped job queue. With these changes, his system would allow multi-tasking guests actually to multi-task when running in a virtual machine. Significantly, his modifications were available on the Waterloo Tape.

Dewayne became the chairman of the Operating Systems Committee of the SHARE VM Project. Under his guidance, the Committee prepared several detailed requirements for improvements to allow guest systems to perform better. At SHARE XLV, in 1975, the Committee presented IBM with a White Paper entitled *Operating Systems Under VM/370*,¹¹⁰ which discussed the performance problems of guests under VM and the solutions that customers had found for these problems. Many of the solutions that Dewayne and others had found, such as PAGEX, made their way into VM fairly quickly, apparently as the result of customers' persistence in documenting them. By SHARE 49, Dewayne was able to state that, "It is now generally understood that either MFT or MVT can run under VM/370 with relative batch throughput greater than 1."¹¹¹ That is to say, they had both been made to run significantly faster under VM than on the bare hardware. Dewayne and others did similar work to improve the performance of DOS under VM. Other customers, notably Woody Garnett and John Alvord, soon achieved excellent results with VS1 under VM.

Having gotten MFT, MVT, DOS, and VS1 to perform well under VM, the user community still had before it the challenge of making SVS and MVS perform well, too. Many people attacked this problem, including Sean McGrath, who succeeded Dewayne as the chairman of the Operating Systems Committee, and who continued for many years to do wonderful things to make MVS run better under VM. The problems in running SVS and MVS under VM were mostly due to CP overhead, rather than to I/O constraints as in the other cases. One of the VM manuals said explicitly that MVS under VM was not a production configuration. And that was true. People were seeing relative batch throughputs as low as 2 percent.

Robert Fisher was a system programmer at Texas Instruments who was faced with the problem of migrating from MVT to SVS to MVS with only one 168. His only choices were either to get MVT and SVS and MVS to perform well under VM or else to stay on MVT forever, so he made the guest systems perform. At the outset, he resolved to make no mods to the guests. However, he did ultimately ZAP one instruction in SVS:

¹⁰⁸ For example, see S. Wecker, "OS/360 as a Preferred Machine Under CP-67", *Proceedings of SHARE XXXVII*, August, 1971, pp. 48-57.

¹⁰⁹ Then at Southern Illinois University.

¹¹⁰ *Proceedings of SHARE XLV*, August, 1975, pp. 493-497.

¹¹¹ D. Hendricks, "Primer on Operating MFT/MVT Under VM/370", *Proceedings of SHARE 49*, August, 1977, p. 475.

LRA R2,0 ==> SR R2,R2

because that Load Real Address was consuming about 15 percent of his processor, which seemed an awful lot just to zero a register.

After a great deal of measuring and experimenting, Fisher concluded that the reasons for poor SVS- and MVS-under-VM performance were shadow table maintenance, CPU timer maintenance, and VM page wait. He wrote some very sophisticated CP modifications to address the shadow table maintenance and timer maintenance problems, introducing multiple segment table origin stacks, “virtual=shadow” support, and fast paths through the program check first-level interrupt handler. Then he put together an impressive presentation in which he described his results and suggested changes to VMA that, together with his changes, could result in MVS under VM running with 90 percent relative batch throughput.

The first foil in Fisher’s presentation said, “VS2 will perform well on VM when...”, and the last foil said, “...when IBM wants it to”. Fisher first gave the presentation at SHARE XLVI, in February, 1976. He continued to give it at SHARE and GUIDE over and over until IBM had to acknowledge his results. He then worked closely with IBM while they developed the Systems Extensions Program Product (SEPP), which implemented a great many of his suggestions.

During this same era, the VM community was working hard to improve CMS performance. IBM had taken the position that CMS was not a production system either, but during VM’s first five years, several customers, such as Pat Ryall and others at Bell Northern Research, successfully established CMS-only data processing centers. Doing this meant adding function that CMS was missing, improving CMS performance, and convincing IBM that VMA could be useful for CMS as well as for guests.

Considerable work was also being put into making CP itself run faster. Jim Best, of Pratt & Whitney, developed the drum multiple exposure modification, three lines of code that trebled the number of pages a system could turn in a given time. Two installations, Cornell and MITRE, wrote new CP schedulers that were a great improvement over the IBM scheduler, and they made their code available to other installations. Larry Brenner,¹¹² of Cornell, implemented drum-to-disk page migration, which greatly improved interactive response time.

The VM Project in SHARE was also working very hard to provide IBM with guidance. The Project produced White Papers on running guests under VM, on the CMS file system, on the CP scheduler, on RSCS, and on other topics. Furthermore, members of the community had already implemented most of the facilities discussed in their White Papers, so they were in a very good position to tell IBM exactly what was needed.

¹¹² Brenner is now in AIX Development at IBM. He made many other important contributions to VM over the years, including the CMS “IPLer” concept, which greatly facilitated supporting multiple versions of CMS.

I. The birth of VMSHARE

The most important step the community took to support the system and unite the users was announced at SHARE XLVII in Montreal in August, 1976. There the great David N. Smith announced the birth of the VMSHARE electronic conference with this foil:

**SHARE
VM Project
Community
Bath
(and Login)**

Dave was the SHARE CMS Committee chairman and was greatly admired for the wonderful things he was doing with CMS, particularly in the area of performance. One of his most famous statements was, “CMS is like a sponge—touch it anywhere and performance squirts out.” Electronic conferences were new back then. A few people had begun playing around with the idea, but not much was happening yet, and Dave hadn’t heard of any other conference. He came up with the idea as the result of Ed Haskell’s having asked him, “Why can’t you do something with that fancy network of yours¹¹³ so that we can communicate between SHARES?”. Dave wrote all the software required for the conference¹¹⁴ and then persuaded his employer, TYMSHARE, to provide the conference and the networking to the VM Project at no charge.

¹¹³ TYMNET.

¹¹⁴ In implementing VMSHARE (in EXEC Classic), Dave had to build one of the earliest CMS “padded cells”, which resulted in one of my favorite of the early VMSHARE files:

<<< MEMO SECURE >>>

Secure memo you can never see!!

If you type the contents of this memo you are entitled to one free drink at SCIDS at Houston. This file is protected by an access list containing only \$AD and thus invisible to you. (Of course!!) Send mail if you crack my ‘security’.

dave smith

*** CREATED 01/06/77 22:51:35 BY TST ***

gotcha!

*** APPENDED 10/10/78 16:35:58 BY AMD ***

Then, Dave went to SHARE and told the VM Project to start using the conference. It can be very difficult to get a conference off the ground, for there is no incentive for anyone to log on until a goodly number of other people are already logging on regularly. But, Dave knew how to take care of that problem. Members of the VM Project soon learned that to go a week without logging on to VM SHARE was to invite a phone call from Dave. Dave's stature within the Project—physical, intellectual, and moral—was such that nobody said “no” to him. As a result, the conference soon blossomed into the heart and soul of the VM community and became essential to people supporting VM systems.

By the following SHARE, in March, 1977, more than half of the Project members had registered to use the conference, and there were some very vigorous discussions taking place. At that SHARE, Dave went around posting banners reading, “VM SHARE: At least once a week!”

John Mort, an IBM representative to the Project, wrote a letter¹¹⁵ the following month commending the Project for setting up VM SHARE and noting that VM SHARE had already saved IBM trouble and expense:

I want to commend the VM/370 Project for their creativity in establishing VM SHARE.

Its benefit to users in avoiding redundant effort is obvious. I would like to comment upon VM SHARE's contribution toward improving the Project's ability to impact the product itself.

Questions pertaining to resolutions are now asked more frequently than previously. Prior to Houston, VM SHARE allowed us to accept two resolutions and avoided our rejecting another through lack of additional information. In another instance, VM SHARE allowed us to provide missing microfiche on a PLC at least 2-3 months earlier than 'normal business'.

The technical dissertation presently going on as a follow-on to a White Paper response at Houston is extremely valuable to us in understanding customer needs and requirements.

To sum it all up, SHARE is now 365 days a year rather than 4 times a year. Please convey my admiration to the developers.

VM SHARE saved IBM from expense and embarrassment over and over in the years that followed, as well as providing customers with expert assistance and early warning of problems. It also allowed more dialogue between IBM and customer technical people than had ever been possible before.

Many of us are convinced that VM would not have survived if Dave hadn't given us VM SHARE.

¹¹⁵ J.R. Mort to E.S. Haskell, April 11, 1977, *SHARE VM/370 Project Minutes*, 1977.

Early in 1979, the next VMSHARE Administrator, Charles Daney (later to gain fame as the author of Personal REXX), extended VMSHARE to Europe, when he arranged for members of the SHARE European Association (SEAS) to participate in VMSHARE, following an ardent campaign for this by Jeff Gribbin, of SEAS.

```
<<< MEMO VMSEAS >>>
```

```
AT LAST I'M LEGAL!!! EVERYBODY PLEASE NOTE MY NEW  
PROJECT CODE IS _CU. ALL VMSEAS USERS MAY BE IDENTIFIED  
BY THE '_' PREFIX ON THEIR USERID. I HOPE THAT WE ARE ALL  
GOING TO BE USEFUL CONTRIBUTORS; THERE'S ANOTHER 16  
USERS COMING UP WITHIN THE NEXT WEEK OR TWO.
```

```
REGARDS, JEFF GRIBBIN (COMMERCIAL UNION ASSURANCE,  
U.K.)
```

```
*** APPENDED 02/20/79 02:48:41 BY _CU ***
```

Jeff, as SEAS VMSHARE Administrator, insisted that SEAS members start using VMSHARE, just as Dave Smith had done with SHARE members two years earlier. Their response was enthusiastic, and soon the VM community had truly become an electronic global village.

J. The birth of VNET

VNET, IBM's internal network, united and strengthened the VM community inside IBM in the same way that VMSHARE united and strengthened the VM community in SHARE and SHARE Europe.

The VNET network, like many of the other good things we have today, was put together “without a lot of management approval”, to quote Tim Hartmann, one of the two authors of RSCS. VNET arose because people throughout IBM wanted to exchange files. It all started with Hartmann, a system programmer in Poughkeepsie, and Ed Hendricks, at the Cambridge Scientific Center. They worked together remotely for about ten years, during which they produced the SCP version of RSCS (which came out in 1975), and the VNET PRPQ (which came out in 1977). After that, RSCS was turned over to official developers.

The starting point for RSCS was a package called CPREMOTE, which allowed two CP-67 systems to communicate *via* a symmetrical protocol. Early in 1969, Norm Rasmussen had asked Ed Hendricks to find a way for the CSC machine to communicate with machines at the other Scientific Centers. Ed's solution was CPREMOTE, which he had completed by mid-1969. CPREMOTE was one of the earliest examples of a “service virtual machine” and was motivated partly by the desire to prove the usefulness of that concept.

CPREMOTE was experimental and had limited function, but it spread rapidly within IBM with the spread of CP-67. As it spread, its “operational shortcomings were removed through independent development work by system programmers at the locations where [new] functions

were needed.”¹¹⁶ Derivatives of CPREMOTE were created to perform other functions, such as driving bulk communications terminals. One derivative, CP2780, was released with VM/370 shortly after the original release of the system.

By 1971, CPREMOTE had taught Hendricks so much about how a communications facility would be used and what function was needed in such a facility, that he decided to discard it and begin again with a new design. After additional iterations, based on feedback from real users and contributions of suggestions and code from around the company, Hendricks and Hartmann produced the Remote Spooling Communications Subsystem (RSCS).

When the first version of RSCS went out the door in 1975, Hendricks and Hartmann were still writing code and, in fact, the original RSCS included uncalled subroutines for functions, such as store-and-forward, that weren’t yet part of the system. The store-and-forward function was added in the VNET PRPQ, first for files, and then for messages and commands.

Once that capability existed, there was nothing to stop a network from forming. Although at first the IBM network depended on people going to their computer room and dialing a phone, it soon began to acquire leased lines. The parts of IBM that were paying for these lines were not always aware of what they were paying for. Since the network grew primarily because the system programmers wanted to talk to one another, a common way of acquiring leased lines for the network was to go to one’s teleprocessing area and find a phone circuit with nothing plugged into it.

The network was originally called SUN, which stood for “Subsystem Unified Network”, but at first it wasn’t actually unified. It was two separate networks that needed only a wire across a parking lot in California and a piece of software (which became the RSCS NJI line driver) to make them one. Hartmann spent some time in California reverse-engineering the HASP NJI protocol, which hadn’t really been written down yet, and finally got that last link up late one evening. Wishing to commemorate the occasion, he transferred some output from a banner printing program running on his system in Poughkeepsie through the network to a printer in San Jose. His co-worker in San Jose, Ken Field, the author of the original HASP NJI code, thought Tim’s output was pretty nifty, so he asked for more copies and taped them up on the walls before finally going home to get some sleep. When Field got back to work late the next morning, he found the place in an uproar over the apparent unionization attempt. The banners had read:

Machines of the world unite! Rise to the SUN!

After that got quieted down, the network began to grow like crazy. At SHARE XLVI, in February, 1976, Hendricks and Hartmann reported that the network, which was now beginning to be called VNET, spanned the continent and connected 50 systems. At SHARE 48, a year later, they showed this map of the network. By SHARE 52, in March, 1979, they reported that VNET connected 239 systems, in 38 U.S. cities and 10 other countries. In August, 1982, VMers celebrating VM’s tenth birthday imprudently attempted to hang the current VNET network map up at SCIDS. By that time, a circuit analysis program was being used to generate the network

¹¹⁶ E.C. Hendricks and T.C. Hartmann, “Evolution of a Virtual Machine Subsystem”, *IBM Systems Journal.*, vol. 18, no. 1, 1979, p. 122.

maps. VNET passed 1000 nodes in 1983 and 3000 nodes in 1989. It currently connects somewhat more than 4000 nodes, sixty percent of which are VM systems. Nobody even attempts to print a map of the network any more.

VMers inside IBM became “networked” very early, and that fact was critical for the survival of VM. One of the most important uses of the network in the early days was for the distribution of the *VM Newsletter*, more commonly known as “Capek’s newsletter”, for its editor, Peter Capek. The *VM Newsletter* went through fifty editions in the late 1970s and early 1980s, ultimately attaining a circulation of 10,000. To submit an article to the newsletter was to assure that it would be brought to the attention of most VMers in the company. One especially delightful item from the *VM Newsletter* was the following:

I am happy to announce the recent marriage of Kittredge Cary and Michael Cowlshaw. Since Kittredge had been living in the U.S. and Mike in England, a major portion of the courtship occurred using the network, and has been recorded down to the tiniest coo.¹¹⁷ In order that others may profit from the wooing technology that has been developed, it is the subject of two upcoming “how to” issues of the *VM Newsletter*, one each from the male and female perspective. Congratulations again to Kittredge and Mike!

Capek’s newsletter was ultimately superseded in 1983 by a more interactive use of the network, the electronic conference, IBMVM. IBMVM is much like VMSHARE in that it contains many individual files (or “fora”), each pertaining to a particular topic. Participants may “append” comments to these files using Mike Cowlshaw’s Tools system,¹¹⁸ which posts their appends to the master copy of the conference disk and also keeps “shadow” copies of the conference disk around the world synchronized with the master.

The existence of VNET has greatly facilitated the functioning of the company and the spread of information. By 1986, *Think* magazine estimated that VNET was saving the company \$150,000,000 per year as the result of increased productivity. VNET has made it much more possible for IBMers in different parts of the world to work closely together, so much so that there is a definition in Mike’s IBM jargon dictionary of a word that describes the condition one gets into when working with someone in a distant time zone:

Netlag n. The result of one’s internal (biological) clock being out of synchronisation with local time due to working on an electronic network. For example, there is a tendency among European IBMers to live on the USA clock in order to have a maximum working time overlap with their US colleagues. This means that they constantly look as though they just stepped off an overnight flight.¹¹⁹

¹¹⁷ It has been suggested that I should state explicitly that IBM does not actually record the contents of mail files exchanged between its employees.

¹¹⁸ Bob Cronin took over support for Tools in 1987 and has enhanced it extensively. A subset of the Tools system was released in 1987 as the VM/DSNX product. DSNX synchronized multiple copies of a disk around a network, but unfortunately did not contain the conferencing function from Tools.

¹¹⁹ M.F. Cowlshaw, *IBM Jargon and General Computing Dictionary*, Tenth Edition, May, 1990, p. 36.

Bruce Marshall was the SHARE VM Project Manager then, and Bruce was just about fed up. VMers had waited two years after System/370 was announced before IBM announced a S/370 that CP and CMS would run on. And even now, two years later, VM was still full of holes. Bruce's project was putting much effort into White Papers and requirements, but nothing was coming of them. So, when *no* IBM representative showed up for the Montreal Interim, Bruce flipped out. After the Project Opening session, he walked down the hall, took a dime out of his pocket, and phoned *Computerworld*. That lost him his SHARE ribbon (although he later became a Director of SHARE) and almost lost him his job.

The *Computerworld* article got Jerry a *lot* of attention inside IBM. Years later, Jerry could look back and say that it had been a good thing, because it got him audiences with IBM managers he'd never otherwise have had a chance to talk to about VM, but at the time it hadn't looked so good. However, he never missed another SHARE between then and the time seven years later when he left IBM to become one of the founders of the Adesse Corporation.¹²⁰

The VM people in SHARE were not the only ones who were unhappy. Shortly after this time, the GUIDE CMS Project Manager, Walt Hutchens (the author of the MITRE Scheduler), wrote IBM an angry letter about the lack of new function in CMS.¹²¹ The SEAS VM Project Manager, Francesco Carreras, sent an angry letter to IBM demanding that IBM send the highest person responsible for VM development to the next SEAS meeting.¹²² Then he resigned as Project Manager (for unrelated reasons) and left Hans Deckers holding the bag. Hans led the SEAS VM Project for many years after that and was followed in that job by Iain Stinson, who later became President of SEAS. The VM community owes all of these gentlemen thanks for their tireless efforts on behalf of VM.

After the *Computerworld* blast, Jerry started learning about VM and trying to figure out why the VM customers were so unhappy. He soon found that IBM had no plan for VM. IBM had no intention of committing more resources to VM. The current marketing forecast was that there would never be more than 500 VM customers. Having been convinced by the customers that VM was too good to let die, Jerry set out to convince the rest of IBM of the same thing and to get that forecast changed. He had no staff to help him, but he did have the customers on his side, and many of them could draw on the resources of their own big companies to help him in the fight. He kept struggling, and more than once was warned that he was being reckless. He later described his first few years as VM Product Administrator as "exhilarating—like riding a roller coaster".

SHARE XLVII, in August, 1976, is perhaps a perfect illustration of the roller-coaster that the VM community was on during this period. By that time, there were 300 VM accounts. One hundred fifty people attended the VM Project Opening session, at which IBM made a variety of very encouraging and very discouraging announcements:

¹²⁰ One of his partners at Adesse was Bruce Marshall, who was still marvelling that Jerry held no grudge against him for the *Computerworld* escapade.

¹²¹ Walter A. Hutchens to B. Suzanne Burr, April 19, 1977, *SHARE VM/370 Project Minutes*, 1977.

¹²² Francesco Carreras to the Manager of VM Development, November 12, 1976, *SHARE VM/370 Project Minutes*, 1977.

- IBM reported that Development's move to Poughkeepsie¹²³ had been completed early and that Burlington had closed on August 22. Although an attempt was made to put a good face on this, everyone knew that it was a move to get VM under control.
- IBM reported that the "Wheeler Scheduler" (the Resource Manager PRPQ) had shipped earlier in the month. Unlike the rest of the operating system, the new scheduler was not free, but it was worth paying for. Not only did you get a visit from Lynn Wheeler, you also found that you could suddenly support ten percent more users.
- IBM announced that the VM Project's requirement VM-142-0276 was being closed as rejected. That was the first time that IBM rejected a VM source requirement.
- IBM accepted a requirement to provide handshaking for other SCPs besides VS1 and a requirement for improved computer networking facilities for VM.
- And, to the amusement of all, IBM announced the cancellation of a session that had been scheduled to discuss the notorious *IBM Systems Journal*¹²⁴ article on penetrating a VM system, giving the explanation that the people who had done the study felt that there was nothing more to be said on the subject.

Following the closing of Burlington in 1976, the roller-coaster ride got even rougher, as we began receiving new releases of VM that seemed to have more new bugs than new function. The problem was simply that there were very few people working on the product who actually knew anything about it. Only twenty-four percent of the Burlington personnel had moved to Poughkeepsie when VM was moved.¹²⁵ Most of the people we've been talking about up to now leave the VM story at this point, because Poughkeepsie was seen as a boring place to live and a bureaucratic place to work. Boston and its suburbs provided other opportunities that didn't require the VM developers to move away from the bright lights and disrupt the lives of their families. By the end of 1976, sixteen percent of the former Burlington personnel were working for DEC. Forty-seven percent had found IBM positions that didn't require them to move to Poughkeepsie. To make matters worse, several of the most knowledgeable CP people who did go to Poughkeepsie, including Dick Newson and Per Jonas, were put into a separate group whose purpose was to build a tool that could be used for the development of MVS/XA. Pete Tallman, of VMA fame, also joined what came to be known as the "VM Tool" project as soon as it moved to Poughkeepsie. Starting with VM/370 Release 3, PLC 06, the VM Tool group began building a fast, stripped-down CP that would create XA virtual machines on a real 370 so that the MVS developers could test MVS/XA.

¹²³ Shortly after that, a VM Development effort started up in Endicott, as well, with the objective of supporting the mid-range processors built in Endicott, while the Poughkeepsie group became more and more tied to the objective of supporting the large processors built in Poughkeepsie.

¹²⁴ C.R. Attanasio, P.W. Markstein, and R.J. Phillips, "Penetrating an Operating System: A study of VM/370 Integrity", *IBM Systems Journal*, vol. 15, no. 1, 1976, pp. 102-116. The IBM order number for a reprint of this article is G321-5029.

¹²⁵ *Directory of the Virtual People*, December 17, 1976.

The roller coaster ride continued as 1977 brought us VM/370 Release 5, with the new systems extensions, BSEPP and SEPP, which incorporated the Wheeler Scheduler, as well as replacing some of the most common user modifications, such as accounting to disk (rather than to real punched cards), page migration, swap table migration, and shadow table maintenance enhancements. Release 5 also introduced the new program product VMAP, written by Chuck Tesler, an IBM SE in Los Angeles. VMAP became an extremely important tool as we struggled to keep our systems running efficiently in an era of growing complexity.

L. Getting IBM's attention

At the suggestion of Jerry DePass, the SHARE VM Project had begun working in 1974 to put together a business case for VM that they hoped would be meaningful to IBM management. At about the same time, the GUIDE VM Project, with help and encouragement from Love Seawright, also began working on a business case for VM. SHARE's effort was organized and led by Ed Haskell, who later became President of SHARE. GUIDE's effort was under the leadership of the great innovator Walt Hutchens.

The final form of GUIDE's business case was their presentation *VM/370 in 1978*.¹²⁶ The final form of SHARE's business case was a presentation entitled *Why VM?* that Ed made to the Director of IBM's Poughkeepsie Lab in November, 1977.

The background for all the title slides in the *Why VM?* presentation displayed a list of customers running VM. The presentation began with an overview of VM's history that portrayed the causes for customer dissatisfaction. It then presented case studies of half a dozen VM customers with an emphasis on the growth rates of their VM systems and their plans for future expansion. That was followed by an explanation of why customers value VM:

**VM/370 users
are committed to
interactive computing
and simplicity.**

and a quantification of the business opportunities that IBM was losing because of its inadequate support for VM. The presentation concluded:

**VM/370 is valuable to
IBM customers in ways
that other SCPs are not.**

¹²⁶ *Proceedings of SHARE 51*, August, 1978, pp. 2-26.

**VM users grow rapidly
subject, in part, to
limitations on the availability
of enhancements to the SCP.**

**IBM can reap great benefits
by enhancing VM/370 in ways
that meet customer needs.**

M. After the Doubtful Decade

By the time we celebrated VM/370's fifth birthday in 1977, we were finally able to begin hoping that IBM would listen to us and not kill VM. A few weeks later, the director of IBM's Poughkeepsie lab came to SHARE to listen to our *Why VM?* presentation, and shortly after that IBM began to commit money to VM's future.

By the end of 1978, there were 1,000 VM installations. In 1979, an entire issue of the *IBM Systems Journal*¹²⁷ was devoted to VM. Clearly, the Doubtful Decade was over. Although it took years before we saw much new function, IBM did at last have a plan that included VM.

1979 brought us VM/370 Release 6 and Release 2 of BSEPP and SEPP, with logical device support, the EDF file system,¹²⁸ and the first of many very disappointing implementations of

¹²⁷ Volume 18, number 1. The IBM order number for this issue is G321-0056.

¹²⁸ On January 11, 1978, during the internal negotiations on the design of the new file system, Dave Smith, who had recently left TYMSHARE and joined IBM Research, produced the following classic letter:

I have been studying the file system subject extensively for the last month and I am now prepared to hand down my blessing on the correct and proper path for CMS file systems development.

I have studied a file system written by Chris Stephenson of Yorktown which is called the New File System or NFS for short. I have studied a file system designed by Dale Witt of Endicott which is called the New Control Block File System or NCB for short. I have also studied a third file system, the Antique File System or AFS for short.

NFS is modern, elegant, sparse, tightly constructed, and is available for use. (One

CMS HELP. The RSCS Networking Program Product was announced on the same day as Release 6.

Meanwhile, hidden away in a corner of Poughkeepsie, the VM Tool group, which had grown to a dozen people by 1980, continued to work on building an XA CP system and SIE microcode to support MVS/XA development and testing. By 1977, they had gotten together a system that would support XA guests (in 24-bit addressing mode) on a real S/370 with SIE emulated in software. They first IPLed an XA processor with SIE in microcode in October, 1979. By August, 1980, they had an XA VM system using SIE in production. Although their official purpose was to support MVS development, the group was composed of passionate VM loyalists who very much wanted to create an XA VM for VM customers.

Once it was confident that IBM was prepared to invest real money on VM, the SHARE VM Group withdrew all of its outstanding requirements, which had been written when we knew we shouldn't ask for much, and set about developing new, longer-range requirements.

In 1980, the VM Group's VM/CMS Task Force, under the leadership of Fred Jenkins (whose mother runs TSO), presented IBM with a White Paper entitled *VM/CMS for the Eighties*.¹²⁹ In the presentation accompanying their White Paper, the Task Force again used a list of VM installations for the background of the title slides, but the print had to be finer this time. The Task Force demonstrated VM's growth by a plot of the attendance at VM sessions at SHARE. There followed a discussion of how fast VM would grow if the inhibitors to its growth were removed and a detailed discussion of those inhibitors. The presentation ended with a plea for improved dialogues between developers and customers to speed the process of removing the inhibitors:

is tempted to say that NFS is tannic in flavor and needs several more years of bottle age.)

NCB is traditional in design with modern overtones but with less bulk than NFS. It is still under construction and all I have seen is some initial blueprints.

AFS is a fine example of its period but showing the results of having had to adopt to the requirements of different ages. One can see the dust particles in the newer coats of paint as well as several chips and scratches. Additions to the design have not added to its basic beauty, but this is common in many older pieces.

Despite the availability of newer pieces which might fit the usage pattern more exactly, I feel that it is important to preserve our heritage, and I therefore recommend that AFS be reconditioned. I feel that we cannot condemn the past since that would deny the future the propagation of the best of the present, but that we must deny the present so that we can propagate the past and preserve it for the condemnation of the future.

¹²⁹ *SHARE SSD*, no. 311, February/March, 1981.

**Without dialogues,
if you tell them
you want something real bad,
you will get it real bad.**

Beginning in the 1980s, as the number of VM installations grew dramatically, we began to see the birth of firms devoted to producing VM systems and applications software. The founders of this “cottage industry” were, for the most part, long-time VM gurus from customer shops and IBM, who knew from first-hand experience what function VM needed to make it a complete production system. They set about supplying that function commercially, thus enabling new VM installations to get started with substantially less expertise and initial investment than had been required earlier.

At the same time, we started seeing the results of IBM’s new commitment to VM. VM System Product Release 1 came out late in 1980. VM/SP1 combined all the [B]SEPP function into the new base and added an amazing collection of new function (amounting to more than 100,000 lines of new code): XEDIT, EXEC 2, IUCV, MIH, SUBCOM, MP support, and more.

There can be no question that by releasing XEDIT in 1980, IBM gave CMS a new lease on life. Within no time, programmers and end users were building large, sophisticated applications based entirely on XEDIT, stretching it to its limits and doing things with it that IBM had never envisioned. That they were able to do that was a tribute to XEDIT’s author, Xavier de Lamberterie.¹³⁰ (If you’ve ever wondered where the “X” in “XEDIT” came from, now you know—it was Xavier here.)

It was also fortunate that those XEDIT macros could be written in the new language EXEC 2, which was considerably faster and more powerful than VM’s original EXEC processor. The author of EXEC 2 was Chris Stephenson, of IBM Research.

¹³⁰ When asked what other editors influenced the design of XEDIT, Xavier graciously replied with the following note (X. de Lamberterie, private communication, 1989):

Well, XEDIT comes from a long way. It has been influenced by the editor from CP-67, then some other editors that were developed locally at the Grenoble University (including editors with macro capabilities, which were probably the first ones to have such a concept), and certainly from Ned that we had a long time ago (some XEDIT target commands are inspired from Ned). Then later on, when full-screen displays were available, XEDIT took some ideas from Edgar and ISPF (features like prefix line commands).

But I guess the major feature of XEDIT was to keep the “heart” relatively small and allow users to redefine and/or extend the existing commands *via* EXECs or REXX macros. This was one of the major successes of XEDIT.

VM/SP1 was just amazingly buggy. The first year of SP1 was simply chaotic. The system had clearly been shipped before it was at all well tested, but the new function was so alluring that customers put it into production right away. So, CP was crashing all over the place; CMS was destroying minidisks right and left; the new PUT process was delaying the shipment of fixes; and tempers were flaring. When the great toolmaker Jim Bergsten¹³¹ produced a T-shirt that warned “VM/SP is waiting for you”, his supply sold out immediately.

Again, as so many times in the past, IBM attempted to address the problems in SP1 by “throwing bodies” at them. In this case, many of the bodies were barely lukewarm, and they produced some stunningly awful “fixes”, but eventually IBM and customers working together got the system stabilized.

Another important innovation announced in 1980 was Pass-thru. The original author of the system that became Pass-thru was Noah Mendelsohn. In 1974, Noah was given the task of inventing a way to allow the PSRs and Change Team people using a system in Mohansic to get access to the RETAIN system in Raleigh. His solution was “V6”, a server that provided the basic Pass-thru function in an elegant and extensible form. Bill Anzick, who had advised Noah on V6 from its beginning, took over the project in 1977 and expanded it into the product that was announced in 1980 as Pass-thru.

Although SHARE had not asked for something like Pass-thru, as soon as we saw it, we wanted it badly. As soon as the tapes arrived, they were rushed to our computer rooms, and the product was installed right away, all over the world. Pass-thru had been used extensively inside IBM before it was released, so it had already been fairly well debugged. Many of us who had moderate-sized Pass-thru networks never saw a failure. However, within weeks of its release, Pass-thru started being used on quite large networks, with many more concurrent users than it had ever had before. Inevitably, some algorithms that worked perfectly well in smaller networks didn’t work so well in large networks. But Anzick and Don Ariola, of Field Engineering, spent a year heroically expanding and strengthening Pass-thru, and it was soon able to support the very large networks that customers were building.

VM/SP1 also included the first implementation of SNA for VM, VCNA. VCNA required a guest system, so VM system programmers unhappily set about learning SMP4. The pain of that led Gabe Goldberg to coin the term “intruder virtual machine” to describe a guest whose only purpose is to implement function that should be native to VM.

Late in 1981, IBM released the PROFS PRPQ, which was the Electronic Office System that had been developed jointly by AMOCO and IBM for AMOCO’s use. After the PRPQ proved to be a success, IBM released PROFS as a full program product. By 1987, there were said to be a million PROFS users outside IBM, and IBM itself had become heavily dependent on PROFS.

In October, 1981, IBM announced the System/370 Extended Architecture (XA). At the same time, it announced a rudimentary XA version of VM, the VM/XA Migration Aid. The Migration Aid was based on the VM Tool built for use by the MVS/XA developers.¹³² At the time of the

¹³¹ Then at Amdahl Corporation.

¹³² “We started the design at the time of VM/370 Release 3 PLC 06 in 1976. We basically took the XA architecture and re-designed what had to be changed to make VM run on XA. We redesigned the RIO structure, the VIO structure, CCW translation, virtual storage management, and a bunch of other stuff. We built a system that would run on a vanilla 370

XA announcement, the VM/XA Migration Aid clearly was not considered to be ready for customer use; its general availability date was announced for twenty-six months later.

On that same day in October, 1981, IBM announced the first three releases of the VM High Performance Option (HPO), a new product that would be shipped as updates to VM/SP to enhance it to support high-end S/370 processors running in 370 mode.

All three flavors of VM were to grow and sometimes prosper throughout the 1980s. The details are probably familiar to you all, so I will touch on only a few of the highlights (and lowlights) of that decade.

Although the button IBM was handing out said that the Migration Aid was more than just a migration aid, it was not very much more. Release 1 was intended to be used to run MVS/SP in production while running MVS/XA in test. All virtual machines were uniprocessors, for example. VM/SP guests were also supported, and RSCS and PVM would run, but CMS was supported only for maintaining the system. However, the group in Poughkeepsie was working hard to produce a real VM/XA system. It was not a small job. Pete Tallman, who by then was leading the team developing virtual SIE, has written:

While VM/370 was tight and well functioning, we, of necessity, had to make some significant changes in order to move into 370-XA architecture. For example, the interval timer no longer existed. Now the CPU timer had to serve double duty for both CPU consumption measurement and time slice. This kind of change rippled into some of the rest of CP in subtle ways that were not entirely anticipated. Little changes kept causing ripples, which caused more ripples. We woke up one day with eighty percent of the system rewritten.¹³³

The same week that XA was announced, a new programmer named Joyce Tomaselli joined VM/XA development. Nine years later, she would become the VM Market Support Administrator in time to announce VM/ESA.

The wait for a real XA VM would be longer than anyone then anticipated, and that wait grew less bearable each time Dick Newson was let loose to talk to a VM user group and to tantalize them with descriptions of the goodies the XA developers were creating. As the waiting continued, the SHARE VM Group Manager challenged the XA developers by proposing a small wager:

and simulate an XA (except for 31-bit mode). We used this to develop the real XA operating system. First time we saw an engineering model (early 3081) with real XA on it, we IPLed and ran with no problem!! Virtual machines for you!!!!” (S.R. Newson, private communication, 1989.)

¹³³ P.H. Tallman, private communication, 1993.

Whereas Richard Newson, hereinafter referred to as The Developer, professes confidence in IBM's ability to churn out code, and

Whereas David Farnham, hereinafter referred to as The Customer, harbors doubts about IBM's steadfastness in such endeavors,

Therefore, be it resolved that The Customer and The Developer do hereby wager

One Case of Johnny Walker Black Label Scotch Whiskey

upon the general availability of a version of VM/CMS that supports most of the function present in the current version of VM/System Product while allowing the Control Program to take advantage of the Extended Architecture present in the 308x series of large computers.

Therefore, The Customer shall surrender the wager unto The Developer should the First Customer Ship date for a VM/XA Product be prior to

April Fool's Day in the year 1985

Otherwise, The Developer shall forfeit the wager unto The Customer.

Award of the wager shall occur at the first major SHARE following the general availability of the product, but not later than SHARE 65.

Done in the City of Los Angeles, California, at SHARE 58, on this Nineteenth day of March, Ninteen-hundred and eighty-two.

“The Developer” never paid off, but says that that was because “The Customer” didn't show up for SHARE 65 in August, 1985.

Meanwhile, back in Endicott, in the summer of 1981, just about the time VM/SP Release 1 got stabilized, the VM/SP developers switched from their heavily modified system to a “vanilla” system. Customers were very pleased to hear this, because they knew what the result would be. Sure enough, in the summer of 1982, VM/SP2 brought us the command retrieve function, and the next year SP3 brought us PER, thus finally obsoleting two of the most common user modifications. SP2 also added the “Productivity Tools” (FILELIST, NAMES, NOTE, PEEK, RDRLIST, RECEIVE, SENDFILE, and TELL), which, combined with XEDIT, greatly improved the end user interface. At the suggestion of Claude Hans, who was back in Endicott for a while, a committee of volunteers at IBM Research at Yorktown had refined the Productivity Tools from programs in use at Yorktown. This effort was headed by Bob O'Hara, of whom we shall hear more soon.

The first half of 1982 also brought us the first two releases of HPO, with Single-Processor Mode, Preferred Machine Assist, V=R Recovery, support for 4K storage keys and segment protect, and a significant speedup in CP free storage management. With all this in place, running guest operating systems became much less burdensome than before.

By the time we celebrated VM/370's tenth birthday at SHARE 59 in New Orleans in August, 1982, IBM had declared VM strategic, and the number of licenses was growing wildly. Our installations had already begun to enjoy some fruits of IBM's new commitment to VM, such as XEDIT, the enhanced file system, and Pass-thru. Although IBM had been to SHARE to discuss the possibility of distributing only object code for its software, the general view was that it would soon realize how unwise that was. In 1982, the VM community had a lot to celebrate. Most of us believed that CMS was about to take over the world, so we gave it a wonderful birthday party.

And we soon got a *very* big reason to celebrate. Early in February, 1983, IBM announced VM/SP Release 3. After years of pleading, we would finally get REXX!

Mike Cowlshaw had made the decision to write a new CMS executor on March 20, 1979.¹³⁴ Two months later, he began circulating the first implementation of the new language, which was then called "REX". Once Mike made REX available over VNET,¹³⁵ users spontaneously formed the REX Language Committee, which Mike consulted before making further enhancements to the language. He was deluged with feedback from REX users, to the extent of about 350 mail files a day. By consulting with the Committee to decide which of the suggestions should be implemented, he rather quickly created a monumentally successful piece of software.

VM customers fell in love with REXX the moment they got it. REXX quickly became the language of choice for CMS applications. In no time, we began seeing REXX applications consisting of hundreds of thousands of lines of code, and few of us wanted to work in any other language.

By the time REXX celebrated its tenth birthday in 1989, it had been designated the SAA Procedures Language and had long since begun spreading to other systems, including MVS. Today, it is available on essentially every significant platform and continues to delight more users

¹³⁴ "It was my good fortune to be in charge of the IBM Hursley Laboratory during the period when Mike Cowlshaw, who at the time was a member of IBM's System Assurance Laboratory, created REXX. I should hasten to say that I claim no credit whatsoever; REXX is the product of a dedicated individual committed to the solution of a problem. He did so not because he was asked to; not because he was expected to; not even because there was any job-related requirement for him to do so, because he was supposed to be wholly dedicated to the evaluation of products developed by others; but he did so because Mike Cowlshaw saw the need and the solution, and got on with it, almost *despite* his management rather than *because* of them." (Sir John Fairclough, in *The REXX Handbook*, Gabriel Goldberg and Philip H. Smith III, eds., McGraw-Hill, 1992, p. xi.)

¹³⁵ "The most important influence on the development of the REXX language was the IBM internal electronic network, VNET. Without the network (and the people who keep it running), there would have been little incentive to start a task of this magnitude; and without the constant flow of ideas and feedback from people using the network REXX would have been a much poorer language. Much credit for the effectiveness of VNET as a communication medium for this sort of work is due to Peter Capek who created the *VM Newsletter* (1973-1983). Today, REXX language design is carried out over the same network almost entirely with the aid of the Tools computer conference system—appropriately enough, a system written in REXX." (M.F. Cowlshaw, *The REXX Language: A Practical Approach to Programming*, Prentice Hall, second edition, 1990, p. x.)

every year.¹³⁶

On February 8, 1983, a few days after the announcement of VM/SP Release 3, IBM announced the Object Code Only (OCO) policy. Much of the heart went out of the VM community on that day. In the years that followed, IBM and its customers lost opportunity after opportunity because of that unfortunate decision.¹³⁷

The VM community devoted enormous effort to attempting to convince IBM's management that the OCO policy was a mistake. Many, many people contributed to the effort in SHARE and in the other user groups. The greatest of SHARE's source heroes was unquestionably Gabe Goldberg, who persevered and maintained hope and a sense of humor in the face of IBM's seemingly implacable position. In SEAS, Hans Deckers was a particularly hard worker in the battle against OCO, and Sverre Jarp, the former President of SEAS, also deserves much praise for his role.

In February, 1985, the SHARE VM Group presented IBM with a White Paper that concluded with the sentence, "We hope that IBM will decide not to kill the goose that lays the golden eggs." Though we had tried to make our White Paper reasonable and business-like, IBM chose not to reply to it.

A few months after the announcement of the OCO policy, IBM released the first OCO version of VM, VM/PC. VM/PC had a number of problems, including poor performance and incorrect or missing or incompatible function. Without source, the users were unable to correct or compensate for these problems, so nobody was surprised when VM/PC fell flat.

IBM continued throughout the decade to divert much of its own energy to closing up its systems, not noticing until too late that the rest of the industry (and many of its customers) were moving rapidly toward open systems.

OCO and other woes drew the VM community ever closer together in the 1980s. The tradition of supporting one another flourished. We all revelled in the generous gifts of other VMers. One especially delightful gift was the amazing TRACK program written in 1984 by my colleague Serge Goldstein. With TRACK, we all suddenly became wizards, able to do real-time trouble-shooting with much greater skill than before.

Equally generous was the work that Bob Cowles at Cornell did during the 1980s to solve problems members of the community had with the IBM scheduler for VM, even though he was himself the author of a competing scheduler product.

¹³⁶ Cowlshaw was made an IBM Fellow in 1990.

¹³⁷ Looking back at VM's early years, Les Comeau evaluated the factors that led to VM's success as follows:

"The success of the CP/CMS system certainly is in no small way attributable to its friendly and forgiving user interface. A second contributor was the clean separation of function in the CP product, which made it easy for the sophisticated user population to remove, add, and substitute functions supplied by IBM, thereby greatly expanding the talent working on enriching the system." (Comeau, *op. cit.*, pp. 45-46.)

Throughout the 1980s, VM SHARE remained the center of the community and the place we went to get help in solving our problems. By 1985, the time had come for us to stop imposing on TYMSHARE's generosity in supporting VM SHARE, so we reestablished VM SHARE on a self-supporting basis,¹³⁸ moving it from California to McGill University in Montreal, where it thrived under the kindly care of Alan Greenberg, the Director of McGill's Computing Centre, and McGill's charming and very skillful system programmer, Anne-Marie Marcoux.

In 1987, members of Australasian SHARE/GUIDE joined us on VM SHARE. ASG's VM Project Manager, Neale Ferguson, encouraged ASG participation in VM SHARE with the same enthusiasm that Dave Smith and Jeff Gribbin had shown in getting SHARE and SEAS members started years earlier. Neale soon endeared himself to the VM SHARE community by initiating one of our all-time great capers, a fictitious around-the-world trip for Gnorman, a neighbor's garden gnome. With the assistance of the VM community, Gnorman made friends around the world, fell in love in Florida, drank beer in Bavaria, and narrowly escaped being arrested in England, all the while sending home postcards promising to return one day.

Meanwhile, IBM was continuing to deliver new VM function. New releases of HPO came out faster than most of us could keep up with. In 1984, HPO 3.4 brought us the swapper. The swapper was a major boon to many installations, as was the page migration function delivered in HPO 4.2 in 1986.

The XA CP developers were also delivering function rapidly, although not rapidly enough to catch up with HPO. Release 1 of the VM/XA System Facility was announced and delivered in 1985. This time, the button said, "More than just a name change", and it was more than that, although a full-fledged XA VM system was still a long way away, and customers were beginning to become alarmed at the large number of bodies being thrown at the XA system.

Just about then, we began hearing fascinating rumors about a guy named Ron Hoffman, a system programmer at an IBM internal site in Rochester, Minnesota, who had single-handedly modified HPO 4.2 to run on an *n*-way XA processor and support XA guests. Nobody seemed ever to have met him, but his manager came to SHARE and told us about his work. By 1986, however, Hoffman had been lured to Endicott, where he and a small group of other very good people, including Sam Drake, Chip Coy, and Bill Fiscofer (the author of AUTOLINK and SVM), had begun working on the Mayflower Project, whose goal was to get a fully XA-capable HPO 5 CP out the door before the hordes in Kingston could ship their HPO-compatible XA. When it began to look as though Mayflower might succeed, it was killed.

By the time VM/SP Release 4 came out late in 1985, there were more than 10,000 VM installations, but as the decade rolled on and SP4 was followed by SP5 and SP6, a very disturbing trend developed. For the first time ever, we saw widespread dissatisfaction with the new function being added to VM/SP:¹³⁹

¹³⁸ Dave Smith's original EXEC implementation of the conferencing software was replaced in 1982 by Charles Daney's PL/I implementation. The Adesse Corporation later acquired Charles' code and marketed it under the name CONTACT. Adesse very generously provided VM SHARE with a free license for CONTACT.

¹³⁹ This dissatisfaction resulted in normally leading-edge customers rapidly retreating to the trailing edge, many of them deciding to remain as long as possible on Release 4, which had come to be known as "the last real release of CMS", because it was the last release before the introduction of CMS Windows. See Larry Chace's paper on why Cornell University

- CMS HELP got worse rather than better, since with every little “improvement” IBM made to it, the source became less available for us to use in alleviating its real problems.
- Although the second implementation of SNA for VM was less foreign than the first, it was not well received. In particular, customers were disappointed that the new component GCS was such a closed system, because they could see that it had great potential. In the years since, the use of GCS has expanded much more slowly than it would have, had full source been available.
- Fullscreen CMS, although it provided some useful infrastructure, was an embarrassment to many of us whose installations were already being taken over by systems with far superior windowing systems.
- Although everybody agreed that National Language Support was badly needed, the VM implementation was viewed as incomplete and awkward to use.
- Although the long-awaited Shared File System¹⁴⁰ appeared to be more promising than most of the other new function, there were concerns about its performance and the fact that it, like CMS Windows, needlessly polluted the CMS command name space, thus causing existing VM installations unnecessary migration problems.
- As more and more of CMS became object-code-maintained, VM installations were encountering long, expensive delays in getting IBM to fix bugs that the customers would have fixed themselves if they had had access to the source.

It wasn't all negative, however. During the mid-80s, a great many limitations were removed from CP, such as the very annoying limit of 10,000 users in the CP directory. At the same time, CMS was receiving numerous small enhancements, such as RLDSAVE, that made it a less difficult application development environment.¹⁴¹ A very fruitful collaboration between Ray Mansell and Jerry Bozman gave us the minidisk directory hashing in SP4 and the faster ACCESS and FST sort in SP5, thus making the CMS file system even faster than before.

For VM/370's fifteenth birthday, in 1987, we had another grand birthday party, at SHARE 69 in Chicago. There were two huge birthday cakes, and several of the most honored old-time VMers gave delightful talks full of funny stories about our long-past struggles to keep VM alive. But underneath these festivities, there was an air of unease. OCO had become real and loomed large. IBM had begun delivering truly unpalatable new function in VM, so that at that SHARE, for the first time ever, we were trying to find out whether we had a mechanism for asking IBM to

decided not to put SP5 CMS into production. (L. Chace, “CMS Release 5 Experiences at Cornell University”, *Proceedings of SHARE 70*, March, 1988, pp. 544-552.)

¹⁴⁰ “My principal ‘if I could do it over again’ item from the CP/67 and CMS days would have been to provide file-sharing extensions to the ‘370 virtual machine’ early on. I don’t believe that any main-line IBM software of the time had these hooks built in, so an implementation would have been one-of-a-kind. I vaguely remember in discussions toward the tail end of 1967 that to try to implement file-sharing would have increased our profile *vis-à-vis* the TSS group and made us appear to be competing with them (which we in fact were), so we didn’t pursue it.” (R.U. Bayles, private communication, 1989.)

¹⁴¹ RLDSAVE was actually first available in VM/PC.

remove new function. And five years after MVS customers had gotten an Extended Architecture version of MVS, we still didn't have a full-function VM/XA.

Throughout the decade, the XA CP developers had been struggling to catch up with HPO. VM/XA steadily acquired function, graduating from "Migration Aid" to "System Facility" in 1985 and then to "System Product" in 1988. VM/XA System Product Release 1, which became available on a limited basis in March of 1988, was called a "full-function" XA VM system. Its CP had been made compatible with HPO 5, and it provided a high-capacity, XA-capable CMS, CMS 5.5. It would be followed by VM/XA SP Release 2 in 1988 and Release 2.1 in 1989, but none of these would support APPC, so their CMS systems would remain incapable of running the Shared File System.

We had been watching the development of XA CP all along, thanks to the frequent reports from Dick Newson, but this "Bimodal CMS" (CMS 5.5) seemed suddenly to appear full-blown from nowhere in 1987. In 1984, an IBM representative had asked the SHARE CMS Project to submit a requirement for an XA CMS. Even in the early 1980s, the high-end engineering and scientific users, especially the chip makers, were suffering from the memory constraints in CMS and, indeed, some were abandoning CMS for that reason. CMS Project members had naturally been assuming that CMS would be made XA-capable, so they were surprised to be told that a requirement was necessary. Nevertheless, the Project dutifully prepared a requirement and then waited three years for the result.

Looking back over VM's history, one sees that CMS has always been somewhat neglected. When the system was being ported from the Model 40 to the Model 67, CP was largely redesigned and the developers hoped to do the same for CMS, but there wasn't time. When the system was being ported from the Model 67 to System/370, CP was largely redesigned and the developers hoped to do the same for CMS, but again there wasn't time. By the 1980s, then, CMS was a tatty old system that was definitely showing its age. For example, it had been designed at a time when a machine with 256K of real memory was impressive, so CMS tended to conserve memory at the expense of all else. And it had never been given proper programming interfaces. To write CMS applications of any sophistication whatsoever, one of necessity had to poke about in the internals of the system; there really was no other choice except not to use CMS at all.

To understand the origins of XA CMS requires more of an understanding of the IBM political climate than of the technical issues. In 1976, when VM Development was moved from Burlington to Poughkeepsie (where IBM's high-end S/370 processors were being built), a few VM Development people were also moved to Endicott (where the low-end S/370 processors were being built). With reorganizations, these two groups came to report up through quite different parts of the corporation, although they were working on the same system. By the time VM/SP Release 1 was shipped, essentially all CMS development was being done in Endicott, while CP development was divided between Endicott and Poughkeepsie. In 1983, the VM developers in Poughkeepsie were moved to Kingston, which is nearby. From then on, VM/SP CP was developed almost exclusively in Endicott, while HPO and XA CP were developed in Kingston.

While IBM management has prided itself on the success of its "contention system" of management, in the case of VM Development, the contention system ran amok during the 1980s, becoming much too contentious. Customers were frequently astonished by the degree of enmity the staffs of the two VM development labs expressed for one another. One saw behavior that seemed unprofessional and, indeed, in some cases, quite childish. It often seemed that almost the only point of agreement between the two labs was their shared resentment of IBM Research, and customers were led to wonder at times whether anyone in IBM really cared as much about VM as about their own silly turf battles.

So, in the early 1980s, XA had been announced, but making CMS XA-capable was not a high priority for CMS Development in Endicott, because Endicott was not building XA-capable processors. Kingston needed an XA CMS for its high-end customers, but had no real CMS expertise itself. (It has been said that in Kingston a CMS expert was anybody who knew what an SVC 202 was.) This situation led to a rather odd tri-partite agreement in which Kingston would fund IBM Research at Yorktown to build an XA CMS that would then be supported by Endicott.

Bob O'Hara was chosen to lead what came to be known as the Turbo CMS Project. Working with him at Yorktown was a small group of very skilled people, including Valerie Nelson, Richard Rynicker, Steve Estes (the author of NAMEFIND), Nick Simicich, and Steve Record, with occasional assists from distinguished visitors, such as Iain Stinson, from the University of Liverpool. Nancy Smith from Research and Steve Jones and Ray Mansell from VM Development in Kingston and a number of others also made significant contributions.

Beginning with VM/SP Release 3 CMS, the Turbo group undertook a major rewrite of CMS with the goal of producing a CMS that would be fully XA-capable and would also run in 370 mode and be mostly compatible with earlier levels of CMS.

At the outset, O'Hara solicited advice from many of IBM's CMS "heavies" and suggested that everyone interested in the project read two papers from VM SHARE, *Is CMS Ready for Extended Architecture?*, by Pat Ryall, and *CMS Architecture and Interactive Computing*, by Charles Daney. Both of these papers contained eloquent descriptions of CMS's strengths (particularly its simplicity and speed)¹⁴² and its weaknesses (lack of multi-tasking, lack of a shared file system, poor storage and contents management, poorly defined programming interfaces, etc.).

O'Hara set up the Turbo CMS group in the Fall of 1982. Their work was scheduled to be completed by the end of 1984, but actually extended about a year beyond that. Over a period of three years, this small group managed to address many of the deficiencies in CMS while keeping the system recognizably CMS, although it was a CMS that would have been as difficult to port old applications to as CMS 5.5 eventually proved to be. By the time Turbo was finished, it provided 31-bit addressing, XA I/O interfaces, improved contents management, well-defined interfaces to CMS system services, general-purpose multi-tasking, a hierarchical file system, and a number of algorithmic improvements. Steve Record has described their work:

Peer multi-tasking was introduced partly because taking full advantage of XA seemed to require it and partly because we all agreed CMS simply needed it. Nick designed a queue manager to provide intertask communication, and I designed a console manager that ran as a service task and was the pre-eminent consumer of the queueing interface. Dick Ryniker and Steve Estes redesigned storage management. Dick rewrote the loader. Dick and Nick, with help from Valerie and Steve Estes, completely reinvented CMS program management. Val brought interrupt handling and much of the rest of the nucleus up to snuff for XA. Everyone worked on OSSIM when other things were done.¹⁴³

¹⁴² For further discussion of CMS's strengths, see these excellent papers by two of VM's brightest stars, Pat Ryall and Stuart McRae: P.R. Ryall, *Why CMS?*, Report RC 13932, IBM Research Division, Yorktown Heights, NY, and S.J. McRae, *CMS and UNIX: What They Can Learn from Each Other*, Document SJM-84.1, Systems & Telecoms Limited, Phoenix House, 1 Station Hill, Reading, Berkshire, RG1 1NB, United Kingdom.

¹⁴³ S.E. Record, private communication, 1994.

It must be admitted that some of Turbo's new features could not possibly be construed as having been necessary to make CMS XA-capable, such as the nifty "YEDIT" extensions to XEDIT that Steve Record had done at Yale and had then brought with him to Yorktown.

But by the time Turbo was finished, it was no longer the CMS that was to be shipped with VM/XA SP. On July 31, 1985, Turbo CMS was run successfully in a 200-megabyte XA virtual machine, but a few weeks later, Kingston and Endicott agreed that they would abandon Turbo and would instead jointly develop an XA CMS based on VM/SP Release 5 CMS.

There is so much speculation about this event that one can hardly hope to get to the truth about it. It seems that the development labs had gotten much more than they bargained for in Turbo, and they were not at all sure that it was what they wanted. What they seem to have wanted was a very bare-bones upgrade of CMS to add 31-bit addressing and XA I/O capability. The spectre of problems supporting customer migration to this sophisticated and somewhat incompatible CMS had apparently scared them off.

The process of building an XA CMS began afresh, this time based on CMS 5 (although CMS 5 was still under development). Some benefit was gained from what had been learned in the Turbo project, but there was little code reuse, in part because implementing general-purpose multi-tasking had resulted in pervasive changes between CMS 3 and Turbo, with the result that some of the Turbo enhancements did not readily fit into CMS 5.

In the joint development project, Endicott had design responsibility and Kingston was in the role of subcontractor. In Endicott, the primary designers were Dave Getson, Joel Farrell, and Rick McGuire. The key technical leaders in Kingston were Gale Fair and Frank Nesogda. The technical people had interesting problems resulting from trying to build on top of a release that was still in development; it was not unusual for them to spend a morning refitting updates that had fitted the day before. There were both technical and political problems resulting from having the project spread across two sites.¹⁴⁴

In 1987, this XA CMS developed jointly by Kingston and Endicott was announced as CMS 5.5, and it was made available on a limited basis with XA/SP Release 1 in March, 1988. It supported 31-bit addressing and the XA I/O architecture. It could run in either 370 mode or XA mode. It had some storage management and programming interface improvements, but it did not have multi-tasking or the Shared File System. Most importantly, it introduced enough incompatibility to be difficult and expensive for users and software vendors to convert to, while providing little in the way of attractive new function to give them an incentive to migrate.

VM development had to put much effort into hastening the conversion of IBM products to CMS 5.5. Steve Jones has written:

The job was far from done after the code was developed. An enormous amount of time and effort was spent on getting program product support. As this project was initiated due to the requirements of chip designers, FORTRAN was critical. In fact, it was the key reason for a 31-bit CMS. FORTRAN wasn't about to support any CMS that wasn't supported by ISPF, its platform for interactive debugging. ISPF initially had little

¹⁴⁴ The development process was greatly facilitated by the use of Mike Cowlshaw's Tools system to maintain the development database. Not too surprisingly, the question of whether to use a Master/Slave or a Peer/Peer configuration in setting Tools up took on political connotations.

interest in supporting another CMS.... At this point, CMS 6 and 9370s were going to take over the world, so it was hard to put together a business case to entice interest in a CMS that was for VM/XA. They didn't have the resources even to do a [code] sizing. It was only after a hacker in Kingston by the name of Al Bruckert was able to get mild success at making ISPF limp on top of CMS 5.5, without any source code and zapping text decks, that a line of communication was opened. I was sent down to Cary to help them with their sizing. The ISPF developer I worked with was a wonderful person by the name of Paul Leone who saw things like me—technical without politics. We decided the best way to do a sizing was to make it work and see how much code it [actually] took....

It was similar with most program products, and support was a hard fought battle. FORTRAN finally did come around, but it was touch-and-go for a while....

The most cooperative program product was GDDM. Its conversion was led by Ray Mansell, who was back in the UK after his years in Kingston. Ray found many problems with CMS 5.5 and fixed many of them himself. He was a true asset and extension to the development team. Later he converted many IUO tools for tool owners who had no access to VM/XA (or had never heard of it) or hadn't time to do the conversion....

Another person who provided much insight as to how CMS 5.5 could be improved was John Hartmann, as he converted his "Toy".... I remember some of the early exchanges we had over the network to this day....

We missed the boat on a lot of migration and compatibility issues; however, most of the Kingston folk were like me and had never seen a customer before. The first customers I ever met were from Cornell. Ben Schwarz, Larry Chace, and Nick Gimbrone came to Kingston to report on their early experiences with CMS 5.5 prior to its being released. I remember topics on their foils such as, "The Mythical Segment Constraint Relief". They were critical but fair in their evaluation and I was in awe of "real customers", having never seen one before, and Ben was my introduction.¹⁴⁵

Customers soon found that migrating to the new CMS was not an easy job. There were many incompatibilities, not all of which seemed justified. To make matters worse, VM/XA/SP and its bimodal CMS were both heavily infested with object-code-maintained modules. The reluctance of software vendors to move products that required system extensions into a mostly sourceless system heightened the barriers to customer migration. The OCO policy and SES, the new service tool in VM/XA/SP and VM/SP Release 6, dramatically increased the complexity of supporting VM systems. This exacerbated the long-standing problem of VM's growth being held down by the lack of knowledgeable VM system programmers.

By 1989, despite widespread customer disillusionment, VM had reached 20,000 licenses.

Relations between the two development labs had gotten so bad by then that Endicott had actually shipped customers a new release of CMS that contained code that checked early in initialization to see whether it was being IPLed under a Kingston CP and then, if it was, quietly committed suicide by issuing a CP SYSTEM RESET command. Even worse, they had made DMSINI OCO

¹⁴⁵ S.L. Jones, private communication, 1994.

to cover their tracks. Customers were outraged, of course, and rose up in rebellion.¹⁴⁶ A few weeks later, the two labs were put under a single manager, Paul Loftus, who soon made the difficult decision to move all of VM Development to Endicott and then began working to repair the rift between IBM and its VM customers. The merger of the rival labs was a painful process that lost many good people from VM, but the benefit was increased focus on what was good for the system rather than what was good for one or the other lab.

In their insularity, the two competing labs had increasingly resisted bringing function into the system from other parts of the company, despite a long history of customers being delighted with mature function brought into VM from internal sites. One of the first fruits we saw of Loftus's campaign to appear more responsive to customers was the announcement on November 1, 1989, that *CMS Pipelines*, the most significant enhancement to CMS since REXX, could at last be purchased by U.S. customers (after having been available in Europe since 1986).

John Hartmann, of IBM Denmark, the author of *CMS Pipelines*, has described the origin of “Pipes”:

I passed through Peter Capek's office one day. We can't really remember when it was—probably sometime late '80 or early '81. He had a box of the *Bell Systems Technical Journal* issue on UNIX under his table. I saw him slip a copy to someone, so I said gimme! Having read it (and ignoring their remarks about structured data), I ran off shouting from the rooftops and then began coding with both hands and my bare feet.¹⁴⁷

Throughout VM's history, real end users have played an indispensable role in teaching others to use CMS. An interesting example of this is seen in the spread of *CMS Pipelines* within IBM. As it matured, *CMS Pipelines* was distributed throughout IBM via the VNET network. Experienced *CMS Pipelines* users (known as “master plumbers”) soon developed a self-study course to help others learn to use *Pipes*.¹⁴⁸ This self-study course later evolved into the excellent *CMS Pipelines Tutorial* (GG66-3158).

The morning after the U.S. announcement of the *CMS Pipelines* PRPQ, John Lynn, of Mobil Research, one of the authors of the *Tutorial*, expressed his delight by sending this note to friends:

Well, I'm still getting over the pleasurable surprise. Reaction to the announcement has had tumultuous effect here at Mobil. There are huge crowds of application programmers from many different Mobil divisions now pressed against the fence of the Technical Center, as they have been throughout most of the night, chanting over and over “PIPELINE, PIPELINE”. Many of the programmers storming the Center are carrying huge posters of John Hartmann, the cult's alleged leader and creator of the product.

¹⁴⁶ The rebellion took place on VM SHARE, in the file PROB 6SP01. Within three weeks of the first shriek of outrage, IBM had released several APARs to address the problem, including one that restored the source for DMSINI and DMSINQ.

¹⁴⁷ J.P. Hartmann, private communication, 1990.

¹⁴⁸ “Larry Kraines wrote the first eight lessons of the course originally. John Lynn wrote one when Larry got busy elsewhere. Many other people contributed improvements after that.” (J.P. Hartmann, private communication, 1990.)

The mood here has been a roller coaster of emotion. There have been several fires, some arrests for controlled substance abuse, and at least one birth since the crowd started to form after yesterday's announcement. There are some camera crews trying to move into place at the entrance to the Center, so once the massive crowd allows room, we'll get some up-close coverage and perhaps an interview with one of the crazed, blanket-clad rioters. I'll keep you posted...

Once customers got hold of *Pipes*, they continued the tradition of teaching others to use it, with Stuart McRae of Share Europe and the greatly esteemed SHARE CMS Project Manager Chuck Boenheim leading the way.

N. The Nineties

On September 5, 1990, IBM announced System/390 and the Enterprise Systems Architecture (ESA). VM/ESA Releases 1.1.0 and 1.1.1 were announced that same day. Although these new releases of VM were not functionally a terribly significant enhancement over VM/XA System Product Release 2.1, the name was significant and heartening. Also very significant was the fact that VM/ESA 1.1.0 was announced with two features, the ESA Feature and the 370 Feature. Thus, it could be (and was) described as the long-awaited "single VM". That was a fiction, of course, as the Control Program components of the two features had been evolving separately since VM/370 Release 3 fifteen years earlier. But they supported the same CMS, and that was what really mattered. It also mattered that these systems were not vaporware. The 370 Feature became available later that month, and Release 1 of the ESA Feature was only six months away from being shipped.

RSCS Version 3 was announced on the same day as VM/ESA, bringing gladness to the hearts of the RSCS gurus in BITNET and EARN, who were badly in need of the enhancements Version 3 provided for managing large RSCS networks. Unfortunately, the primary author of RSCS Version 3, Oliver Tavakoli, was to leave IBM when the Cambridge Scientific Center, VM's birthplace, closed in the summer of 1992.

By the time VM/ESA was announced, the VM developers and the VM customers had begun really talking to one another again. For example, we noticed a very large increase in IBM's use of VM SHARE, as IBMers logged on to ask questions about our requirements or the ways we were using the system and to answer questions we had raised. We were heartened to see them even finally begin to argue with us, setting us straight when they thought we needed it.

Also very encouraging were the "developer dialogue" sessions that began taking place at SHARE and SHARE Europe and other user group meetings. These sessions allowed the developers to benefit from the experience of knowledgeable customers and provided them with willing guinea pigs to try out their prototypes. Joel Farrell, the lead developer of CMS Multi-Tasking, made particularly effective use of developer dialogue sessions during the evolution of CMS Multi-Tasking. I think it is fair to say that the result would have been far less successful without the customer input and feedback that Joel got in those sessions. Bob Bolch, in particular, should be cited for the effort and knowledge he devoted to this process.

CMS Multi-Tasking was first delivered in 1991 in the form of a no-charge PRPQ called the Server Tasking EnViRonment/VM. The acronym was pronounced "SteveR", a tribute to Steve Record. In the years that followed, Steve was to put as much work into teaching us to use CMS Multi-Tasking as he had put into helping design it. Although STEVR was not a general-purpose multi-tasking capability such as the Turbo Group had built for Turbo CMS, it did provide a

usable infrastructure for building multi-tasking servers. It was shipped as part of CMS beginning with CMS 9 (VM/ESA 1.2.0) in 1993. Subsequent CMS developments, especially in OpenEdition/VM, have made extensive use of the multi-tasking infrastructure.

As more and more customers moved to versions of VM that required SES, the SES situation became rather desperate. Fortunately, a number of people made heroic efforts to make SES safe. One of the greatest of the SES heroes was Mark Cathcart, of IBM UK, who taught us to find a path through the SES jungle (though perhaps a minefield would have been a more apt metaphor). Then, an enthusiastic new SES developer, Alex Feinberg, began coming to SHARE to work with customers to define the problems and the solutions. Many customers worked hard to help with this process, most notably Romney White.

Way back in 1983, the SHARE Software Service Task Force issued an extensive report on the state of IBM's software service process and made many recommendations for improving that process. Among the most strongly felt of the Task Force's recommendations were those for electronic communication between customers and IBMers, electronic distribution of service, and availability of softcopy for publications. Happily, in the 1990s IBM began to implement those recommendations, and when it did, VM Development and VM Information Development were quick to make good use of these new capabilities.

CMS 8, in VM/ESA 1.1.1, which became available at the end of 1991, finally had "indoor plumbing". *CMS Pipelines* was at last a part of CMS, rather than an add-on product.¹⁴⁹ To the great relief of those of us who have too often watched splendid internal tools stagnate after being put into the product, the ESA version of *CMS Pipelines* has continued to track the internal version, as it grows ever more powerful. VM Development is to be commended for the decision to fund on-going development of *CMS Pipelines*.

Like so many other good things from VM, *Pipelines* soon spread to TSO. A TSO implementation was released in 1995 as part of the BatchPipes/MVS product (now called SmartBatch).

Late in 1991, VMSHARE began buzzing with rumors about one of the most exciting VM innovations in this decade—the Personal/370 card, which came from the lab of the IBM Fellow Bill Beausoleil, one of the original System/360 architects (and father of the channel). The P/370 was a real System/370 on a card; it had about the same capacity as a 4381-P13 and ran as a co-processor in a PS/2. Really slick OS/2 programs were used to emulate channels and control units to make PC devices look like mainframe devices to VM on the P/370. (The primary author of these programs was Chuck Berghorn, to whom the VM community was already indebted for WAKEUP and LANRES.) What made the P/370 so exciting was that it ran real S/370 operating systems, such as the VM/ESA 370 Feature, *without modification*.

With Neale Ferguson leading the way, members of the VM community acquired P/370s for development and even production use and were quite successful with them. However, the P/370 had escaped from IBM just a little too late, after most innovative installations had moved on to the S/390 architecture, so many VMers sat on their hands waiting for the 1995 announcement¹⁵⁰

¹⁴⁹ Later, *Pipes* was also incorporated into Release 1.5 of the 370 Feature.

¹⁵⁰ The P/390 announcement letter is unique among IBM announcement letters in this decade in that whenever it mentions VM and MVS together, VM is always listed first. It appears that this anomaly can be attributed to Gary Ehemann.

of the P/390, officially known as the “PC Server/500 System/390”. The P/390 has been quite successful,¹⁵¹ but has, like so many other VM innovations, been more successful in sheer numbers in the MVS world.

The big story of the 1990s throughout the computing industry has, of course, been the growth of the Internet. VM was quite an early participant in the Internet. In 1984, a group at the University of Wisconsin, under the leadership of Professor Larry Landweber, released WISCNET, a VM implementation of what were then still known as “the DOD Internet protocols”. WISCNET rapidly caught hold at VM installations in universities. By 1985, Bob Cowles at Cornell and others around the world were working out ways to keep CP performance from degrading under a heavy WISCNET workload.

WISCNET development had been funded by IBM Research, and a group was soon formed at Yorktown to expand on WISCNET. In 1987 they released a follow-on VM TCP/IP product commonly known as “FAL” (after its product number, 5798-FAL).¹⁵² FAL quickly became the lifeblood of university VM installations, and VMers at universities were grateful for the opportunity to work closely with the FAL group, who supported their own code and engaged in an on-going dialogue with their customers.

Because FAL was distributed with complete source, customers were able to extend and enhance it in significant ways, making their VM systems more and more competitive. The FAL product supplied just a basic suite of Internet applications, but customers quickly built more, innovating to keep their VM systems alive.

Because of FAL, the academic RSCS networks BITNET and EARN continued to grow in parallel with the Internet. In 1989, Princeton released its VMNET product, written by my colleague Peter Olenick. VMNET moved RSCS traffic between VM systems across an IP network. This allowed BITNET to take advantage of the growing speed of the Internet, so suddenly we could use SENDFILE and TELL across T1 lines! A few years later, Kerry Anderson, of the University of Maine, wrote the TCP line driver for RSCS that came out in RSCS Version 3 Release 2.

One of the most significant impacts of VM on the Internet came from the BITNET LISTSERV program developed by Eric Thomas in 1986 to reduce the BITNET traffic from mailing lists. With the development of mail gateways between BITNET and the Internet, the use of LISTSERV lists grew explosively, uniting people with like interests all over the world, including VMers. By the time of its twentieth birthday in 1996, VMSHARE would start becoming disused, after twenty years of intensive activity, as VMers switched to the more convenient and accessible mailing lists.

By the early 1990s, VMers at universities understood only too well that VM’s future depended on its ability to use the Internet. Then, unfortunately, support for FAL was moved from Yorktown to Raleigh. The folks in Raleigh refused to learn to do maintenance in the VM style, with updates and auxfiles, but instead sent out very buggy fixes in the form of complete module replacements. That left customers to deal with the problem of a vital part of their systems being unmaintainable. Years of complaining got them nowhere, and then things got worse, as FAL development was also moved to Raleigh, at which point FAL essentially stopped being enhanced. Meanwhile, the

¹⁵¹ It is interesting to note that we have come full circle in this story, as one of the vendors now selling products based on the P/390 is Gene Amdahl, the father of System/360.

¹⁵² The MVS TCP/IP product, produced by the same group, is known as “HAL” (5735-HAL).

MVS TCP/IP, which had been derived from FAL, continued to be improved.

VM customers did their best to compensate for IBM's short-sightedness. The great toolmaker Arty Ecock gave the community the wonderful gift of REXX support for IP sockets, in his splendid RXSOCKET package. RXSOCKET let other toolmakers go wild developing VM Internet applications. The most notable of these innovators was unquestionably Rick Troth, then at Rice University. Early in 1992, Rick gave us both a Gopher client and a Gopher server for VM, and those competed easily with the Gophers on other systems. Rick's Gopher used RXSOCKET for all of its dealings with the network and also made heavy use of *CMS Pipelines*, becoming the first *Pipes* application widely distributed among customers and opening the eyes of many VMers to the power of *Pipes*.

Two years later, while IBM was still neglecting FAL terribly, Rick gave the world his Webshare Web server for VM, which was also built on RXSOCKET and *Pipelines*. By then, the Internet was becoming visible in the business world, too, and some of VM's most innovative commercial customers, most notably Jeff Savit, of Merrill Lynch, and my colleague Gretchen Thiele (then at 3M), began leading the way to putting VM applications on the Internet and their Intranets. Meanwhile, other VMers had used RXSOCKET and *Pipes* to build Web browsers for CMS.

In 1994 *CMS Pipelines* support for TCP/IP became available experimentally, further facilitating the development of VM-based TCP servers and clients. Rob van der Heij and others soon began spewing out TCP-based applications that used *Pipes* to talk to the network.

When VM/ESA 2.2.0 was announced in 1996, we were relieved to see that IBM had finally come to realize that the Internet is important to VM and had incorporated Arty's RXSOCKET into CMS 13. IBM Endicott has been picking up support for FAL in recent years, and we are hopeful of better times.

As we got into the 1990s, the size of the VM Development group began being reduced dramatically, as IBM management recognized that the dream it had once cherished of hundreds of thousands of VM licenses on low-end Endicott processors was not going to materialize. A new emphasis was placed on enhancing VM's traditional strengths, especially its power as a server. One of the first fruits of this was the extremely successful ADSTAR Distributed Storage Manager (ADSM).

ADSM had its origins in about 1983 as a program called CMSBACK, written by two VM support people at IBM's Almaden Research, Rob Rees and Michael Penner. CMSBACK allowed CMS users to back up and restore their own files. It quite naturally became the basis a few years later for a program to allow workstation users to back up and restore their files, which was announced as the WDSF/VM product in 1990. By that time, VM Development in Endicott had gotten involved, doing much of the work required to bring the new product to market.

The follow-on product, ADSM, was developed jointly by Almaden, the newly minted ADSTAR division, and some of the VM developers in Endicott.¹⁵³ Version 1 of ADSM (which briefly had the name DFDSM) was shipped in 1993. By then, the server had been made to run on MVS as well as VM. The port to MVS was done primarily in Endicott by Chip Coy and Bill Fiscofer.

¹⁵³ To see a list of the server developers for ADSM Version 1, type SHOW DEVELOPERS on your ADSM server.

ADSM has been one of IBM's great success stories in the 1990s, with hundreds of thousands of users. The server now runs on all major platforms, and there are clients for almost every workstation one can name. Sadly, though, ADSM has forgotten its origins; there appears to be no plan to make Version 3 of the ADSM server available on VM.

One of the most astonishing developments in VM in the 1990s has been the extraordinary improvement in code quality. New levels of the system have gone in with unprecedented ease. This had the effect of increasing customer confidence in IBM and of greatly reducing the impact of the SES problems. Obviously, much discipline and hard work were required to reduce system failures so dramatically. Almost certainly, the reduction in the size of the VM Development group was an important factor in achieving the lower error rates.¹⁵⁴

Another factor in the code quality improvement was the introduction of a sophisticated new Assembler that provided excellent diagnostic capabilities. The High-Level Assembler was another gift to IBM from the VM community; many of the ideas it incorporated began with the famous "SLAC mods" to earlier IBM Assemblers. John Ehrman, of IBM's Santa Teresa Labs, had been the primary author of the SLAC mods before he moved to IBM. At John's SHARE presentation shortly after the HLASM announcement in 1992, delighted customers interrupted him with spontaneous applause several times per foil.

New levels of VM have also performed better and better, as IBM, vendors, and customers have evolved a skillful and cooperating group of VM performance gurus, who trade ideas, discoveries, and challenges with one another on a daily basis, embodying the traditional spirit of the VM community in their shared delight at making the system run better. The most cherished member of this group is, of course, Bill Bitner, who has taught himself and us how to make our systems sing and dance.

When Release 1.2.0 of VM/ESA came out in 1992, VM got a fabulous twentieth birthday present in the form of masses of nifty enhancements from the "CP Configurability Group", whose chief members were David Boloker, Rich Corak, and Oliver Tavakoli. These enhancements cleaned up all sorts of "lawn furniture"¹⁵⁵ items and externalized the configuration data from the CP nucleus, making most of the configuration parameters changeable dynamically by CP command.

¹⁵⁴ I would like to believe that every VM development manager is familiar with a comment Les Comeau made about the success of VM:

"It would be extremely gratifying to attribute that success to brilliant design decisions early on in the program, but, upon reflection, the real element of success of this product was that it was not hampered by an abundance of resources, either manpower or computer power." (Comeau, *op. cit.*, p. 38.)

¹⁵⁵ The "lawn furniture" analogy for CMS's weak spots is derived from an append to the VMSHARE file MEMO VMSP5, by Val Breault, of General Motors Research, who was discussing CMS Windows:

It strikes me as somewhat arrogant and thoughtless. Consider... How would your intentions be perceived if you were to surprise your wife with ghastly livingroom wallpaper and clashing drapes when for seven years she had been begging for real livingroom furniture to replace the lawn chairs you had been using? Would you *really* expect to get away with it by saying the furniture is a Future Objective, as you had been saying since 1982?

Boloker's SHARE presentation on these enhancements ended in a standing ovation.

Boloker and Corak went from triumph to triumph as the 1990s continued. VM/ESA V2 brought us their Configurability II, which made CP's I/O devices externally configurable, and Corak's brilliantly designed CP Exits facility, which made us wish that CP had always been like this.

REXX continued throughout the 1980s and 1990s to spread to other platforms and to grow in importance. When the SHARE REXX Project surveyed SHARE membership in 1992, forty-four percent of the installations reported that they used REXX on more than one platform. Sixty-three percent said that they had used REXX to write mission-critical applications, and nearly one-fourth reported that they made extensive use of REXX for such applications.

Meanwhile, however, REXX was being neglected on VM. When CMS 10 came out in ESA 1.2.1 in 1993, VM customers were relieved to see that CMS had finally gotten to Version 4 of the REXX language, well after that level had become available in other REXX implementations. For a decade, VM customers had viewed REXX as the very essence of CMS, but somehow they had never succeeded in making IBM understand that, and IBM had allowed REXX to languish under VM while it devoted development effort to components that were much less significant to its customers. REXX enhancements were slow in coming to CMS, and the new REXX function was definitely buggy, which was especially disappointing in a product that had in its early days been essentially flawless.

Despite the fact that VM customers were using REXX for much of their critical application development, IBM frequently failed to make new system interfaces available to the REXX programmer. In some cases, customers stepped in to provide the REXX interfaces necessary for using VM functions. In 1989, Arty Ecock gave us RXLDEV, which made it easy for REXX programmers to manipulate logical devices. Two years later, his RXSOCKET became key to VM's survival. More recently, Perry Ruitter has given us MTREXX to allow REXX programs to create MT threads.

Despite itself, IBM did take one very significant step to empower the REXX programmer when it released *CMS Pipelines* in 1989. *Pipes* opened up many previously-unavailable VM interfaces to use from REXX.

IBM was neglecting REXX on other platforms, too. The REXX Project spent years submitting well-thought-out requirements for enhancements to REXX, but little new function was forthcoming. In February, 1994, the REXX Project presented IBM with a White Paper entitled *The Future of REXX*.¹⁵⁶ In that White Paper it provided a business case for IBM to upgrade REXX on all IBM platforms, both to provide object-oriented programming constructs and to meet a number of long-outstanding requirements against the classic REXX language. At SHARE 84 a year later, IBM gave that White Paper a very positive response,¹⁵⁷ committing to essentially everything the REXX Project had asked for, agreeing to deliver both the object-oriented constructs and the enhancements to the original language in the new Object REXX language. A year later, however, in a small, closed meeting at SHARE 86, IBM took the completely unprecedented step of rescinding a White Paper response. It informed the REXX Project that there would be *no* S/390 implementations of Object REXX.

¹⁵⁶ *SHARE SSD*, no. 390, April 6, 1994, pp. 26-41.

¹⁵⁷ "REXX: Launching the Next Generation", *Proceedings of SHARE 84*, February, 1995, Session 6519.

The VM community can now only hope that it will soon be able to use Mike Cowlshaw's exciting new NetREXX language on CMS.

Despite the reduction in the size of VM Development (or perhaps because of it), IBM has been able to deliver significant new function in VM, such as OpenEdition/VM, which came out as part of VM/ESA Version 2. The community appears, in fact, to be a bit overwhelmed by OpenEdition, and few of us so far are exploiting it fully. As so often in the past, Neale Ferguson has led the way in exploring this new technology, porting large numbers of UNIX programs to OpenEdition and very generously doing his best to teach the rest of us how to do the same.

We have been heartened in this decade by a quiet slackening of the momentum of the move towards OCO. Indeed, a significant portion of the system has been moved back out of the OCO category. We take this as a sign of a wish on IBM's part to return to a less antagonistic relationship with its customers. The process is not yet complete, however. As long as we are unable to assemble or compile the entire system, we will be hobbled in our use of it and weighted down by the burden of the artificial requisite chains that are the inevitable result of doing object-level maintenance.

There are 14,000 VM licenses today, and where the future will take VM is unclear. There are pessimistic views and optimistic views.

Whatever the future holds, however, all of us here have much to celebrate, as we look back over twenty-five years of accomplishment and camaraderie, twenty-five years of a world-wide community of people giving their best to make something wonderful even better.