SNA Perspective

Volume 12, Number 1
January 1991
ISSN 0270-7284

The single source, objective monthly newsletter covering IBM's Systems Network Architecture

SAA is Dead. Long Live SAA.

SAA is metamorphosing from its marketecture adolescence. A pragmatic, credible, and compelling architectural scheme is beginning to emerge from what appeared, for over three years, to be an overly ambitious and extemporaneous market place-holder.

The Hubble telescope of the computer industry, SAA has had its focus corrected and now looks ahead at the next millennium with a clear mission. IBM has repositioned SAA from being a context for portable applications to an environment analogous to a virtual machine for cooperative processing. The success of the major SAA applications has been mixed to date, but *SNA Perspective* believes that recent announcements flesh out the SAA vision. Users must understand, however, that the vision is long range—the measurable benefits of SAA will emerge over a decade, not a year. This article examines this new vision for SAA, especially with regard to SNA and communications.

(continued on page 2)

Internetworking and SNA, Part II

Last month, *SNA Perspective* introduced a series on internetworking and SNA. The series will be presented over several months as a debate comparing the different approaches of SNA and leading multivendor networking protocols, such as TCP/IP and OSI, to different elements of internetworking. This installment discusses internetworking issues at layers two and three of the OSI reference model.

The challenge of internetworking is to meet the *business* needs for information exchange and the *people* needs for autonomy with solutions that are robust, cost-effective, and minimally disruptive and can be used to build domains and combined with other domains into arbitrarily large networks.

To meet today's need for internetworking, SNA is becoming more heterogeneous and TCP/IP is becoming functionally richer.

Because of the breadth of the IBM product line and the diversity of its customer base, IBM more than any other manufacturer gets involved in all these topics. Its challenge is to support the fewest number of options that meet customer requirements.

(continued on page 13)

In This Issue:

(Continued from page 1)

Clarifying the Vision

The September 1990 System/390 blockbuster announcement suite heralded a new SAA era, albeit rather vicariously. IBM's unwavering commitment to SAA was confirmed by the announcements of APPC/MVS, distributed relational database access (DRDA), resource recovery, SAA application connection services, and SystemView, all of which have significant SAA ties. They also provided a glimpse of what is emerging as a credible implementational support strategy for SAA and cemented SAA's commercial significance.

APPC/MVS and the companion common programming interface for communications (CPI-C) announcements are the most tangible endorsement of SAA to date. APPC/MVS is an LU 6.2-based interprogram communications service that will be a standard feature on IBM's flagship MVS/ESA operating system. SAA's CPI-C will be the application programming interface (API) to this key strategic service. Figure 1 shows the interrelationship between APPC/MVS, CPI-C, and SAA applications.

CPI-C: SAA's First Beachhead

By mid-1991, a standard LU 6.2 service with the same CPI-C API will be available across MVS, VM, CICS, IMS, and OS/400. It is safe to assume that a CPI-C API will also be offered on OS/2, the other SAA platform, within the next twelve months. Seven years after the debut of LU 6.2, an LU 6.2 service callable by C, PL/I, COBOL, and FORTRAN with the same standard, formally architected API will finally be available on all the prime IBM distributed processing–oriented platforms. Availability dates for CPI-C on each SAA platform are provided in Table 1.

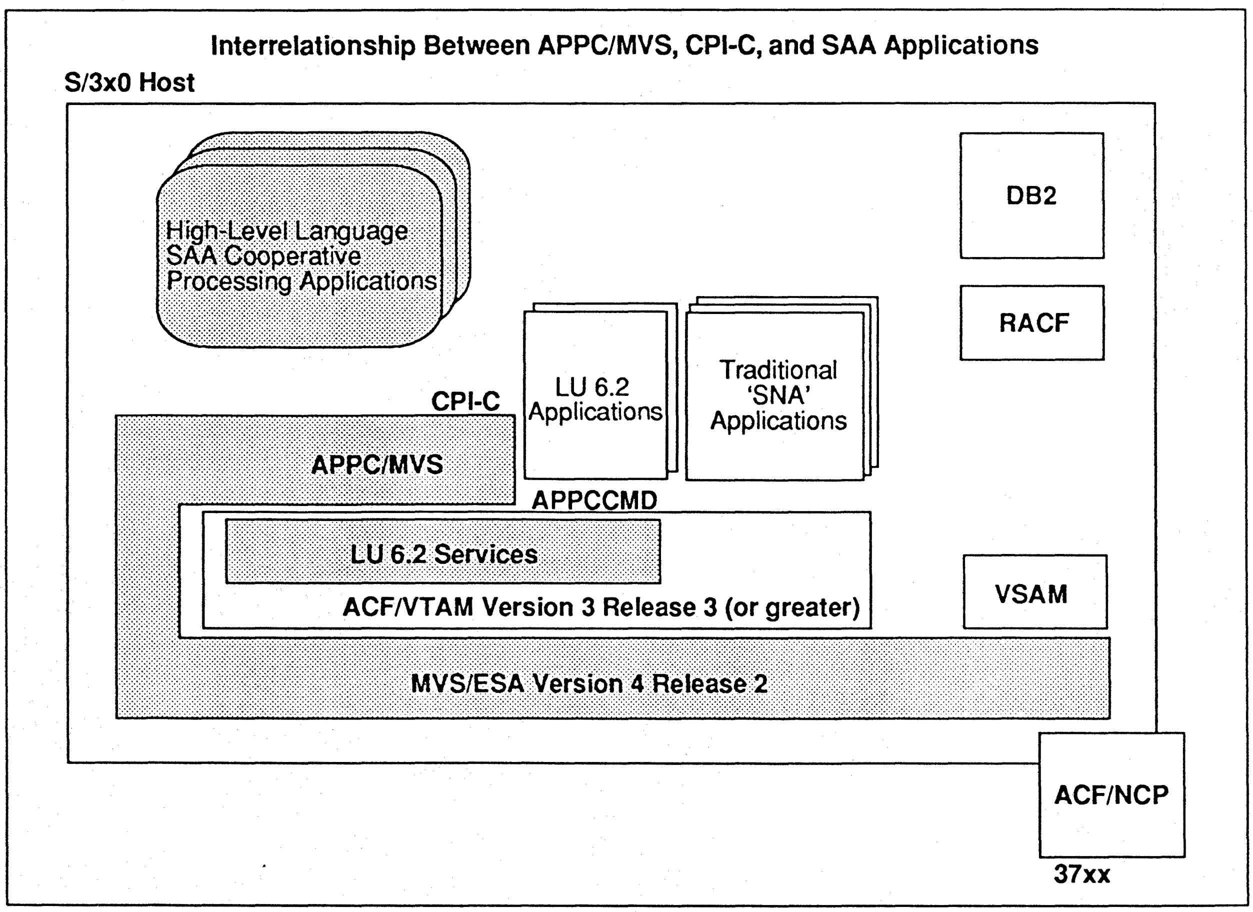

Figure 1

CPI-C Availability

MVS	*APPC/MVS* in MVS/ESA Version 4, Release 2	March 1991
VM	*VMLIB* Service Library in VM/SP Release 6	Shipping
CICS	CPI-C Interface in CICS/ESA Version 3, Release 2	June 1991
IMS	*APPC/IMS* in IMS/ESA Transaction Manager Version 3, Release 2	June 1991
AS/400	CPI-C interface in OS/400	2nd half 1991

Table 1

CPI-C's presence across the SAA platforms and its high-level language support ensure that it will become the *de facto* API for LU 6.2 interactions. IBM's intention for CPI-C to be a common API for both LU 6.2 and OSI distributed transaction processing (DTP) further strengthens SAA's hold. CPI-C will become synonymous with what LU 6.2 was intended to become. In this way, SAA has managed to maintain control of a crucial utility application service—program-to-program communications—which is the basis for cooperative processing.

The New Computing Model

Though it's the most obvious and striking, the CPI-C interprogram communications service is not the only important utility application service being molded by SAA. As shown in Figure 2, SAA currently encompasses the services and APIs for the following crucial application services:

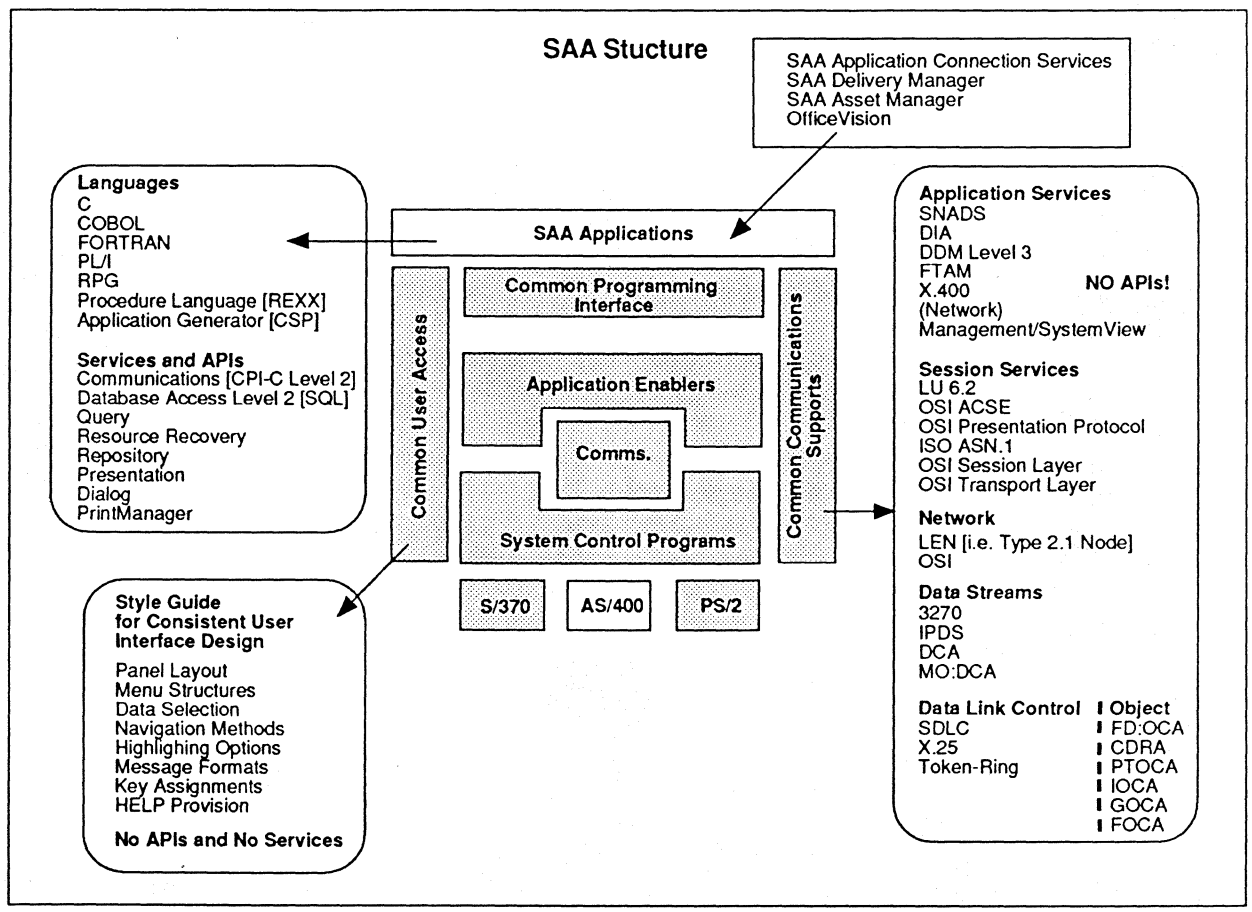

Figure 2

- local, remote, or distributed relational database access
- resource and operation cataloguing (i.e., Repository)
- resource recovery
- graphical presentation
- advanced-function printing

SAA has started to build a foundation for a comprehensive cooperative computing environment that is analogous to similar environments being identified for the open systems UNIX world by X/Open and OSF. This was clarified by the S/390 announcement suite. Several elements are still missing from SAA, however, as shown in Table 2.

SAA, despite its much-derided lack of initial steadfast objectivity, will be the grand unification plan for IBM's bold new computing model for the future based on distributed data and cooperative processing. This model has S/390 mainframes hosting large, pan-enterprise DB2 databases catalogued via Repository Manager acting as ultrahigh performance, high-availability data servers and total system management focal points for a new generation of highly visual, cleaved-function, cooperative processing applications that exploit the processing, storage, and graphical presentation of OS/2 and DOS/Windows 3.0 workstations. SAA goes a long way toward defining a coherent and workable scheme for exactly this type of cooperative processing environment.

SAA—A Pragmatic Definition

The SAA architecture is spelled out in twenty SAA-specific interface reference guides, five SAA design guides, two SAA services summaries and an SAA overview document. (Several of these documents are listed in the sidebar, "SAA Documents.") The architecture thus documented is a generic model for hardware-independent and operating system–independent application

SAA—What is Missing

Architecture

- APIs to the common communications support (CCS) component application services; e.g., SAA-specified standard APIs to SNADS, DIA, FTAM, X.400 and network management
- A true, intermediate-node routing, multilink, peer-to-peer SNA networking scheme; i.e., an APPN derivate as opposed to the current LEN
- Location independent APIs; i.e., the ability to interact with a service using the same API irrespective of whether the service is located locally or remotely
- More distributed computing oriented services: e.g., authentication, naming, directory
- Support for object oriented programming techniques

Implementation

- Comprehensive and consistent support for the SAA application execution environment on all four SAA platforms: MVS, VM, OS/400 and OS/2
- True SAA-compliant SAA applications (not OfficeVision!)

Table 2

development. It defines a set of high-level, generic criteria for the development of consistent application programs, capable of being executed on disparate computing platforms. The criteria attempt to ensure that future application programs will be developed around a common set of standard backbone support services and that these applications will routinely exhibit a common look and feel.

To achieve platform-independent, application-level consistency, the SAA criteria define:

- a standard program execution environment
- a standard repertoire of communications functions
- a standard style guide for user interfaces

SAA, true to its name, is therefore a genuine and intrinsic architecture for application program development methodology. The attributes of SAA applications are given in Table 3.

Attributes of SAA Applications

Standalone Applications

Written in:	C, COBOL, FORTRAN, PL/I[, RPG, CSP]
File I/O:	Programming language I/O (and DDM)
Presentation:	Programming language I/O or SAA presentation or dialog services
Printing:	Programming language I/O or SAA printing service
Database access:	SAA database service (SQL-based), SAA query service or DDM
IPC:	SAA CPI-C
Data distribution:	(SAA common communications support services)

Cooperative Processing Applications

Host or Server Component

Written in:	C, COBOL, FORTRAN, PL/I, (RPG, CSP)
File I/O:	Programming language I/O (and DDM)
Presentation:	*CPI-C to client workstation*
Printing:	Programming language I/O, SAA printing service or *workstation printing*
Database access:	SAA database service (SQL Based), SAA query service or DDM
IPC:	SAA CPI-C
Data distribution:	(SAA common communications support services)

Workstation or Client Component

Written in:	C, COBOL, FORTRAN, PL/I, (CSP)
File I/O:	Programming language I/O, (and DDM)
Presentation:	Programming language I/O, Presentation Manager (Windows 3.0, Motif ?)
Printing:	Programming language I/O, SAA printing service or *host printing*
Database access:	SAA database service [SQL Based], SAA query service, DDM, or *CPI-C to Host Component*
IPC:	SAA CPI-C
Data distribution:	(SAA common communications support services)

Table 3

The SAA POPs

The SAA program development methodology can be thought of as identifying a standard, universal, virtual target system. The SAA architecture is the principles of operations (POPs) for this virtual machine in much the same way that the renowned S/370 POPs describe the workings of S/370 architecture machines, as shown in Figure 3. The S/370 POPs enabled the multibillion dollar S/370 plug-compatible host market, now dominated by Amdahl, Hitachi Data Systems (HDS, née NAS), and Fujitsu, to flourish in the 1970s. The SAA architecture, if innovatively exploited, will permit an even bigger market for SAA-compatible systems and applications to be realized in the 1990s. Of course, that is a big "if."

The program development methodology advocated by SAA is specified in terms of a collection of programming languages, programming interfaces, other architectures, protocols, and conventions. These various, and somewhat disjoint, specificational components and techniques are not applied uniformly across all the constituent parts of SAA in a complementary manner. Neither do they methodically describe the architecture in terms of a framework of hierarchically layered and interrelated sets of functions or services, as is commonly expected in a climate influenced by the OSI seven-layer model. Instead, these specificational devices are pragmatically targeted at defining distinct aspects of the overall SAA application development and execution model. Nonetheless, even this incomplete and somewhat idiosyncratic

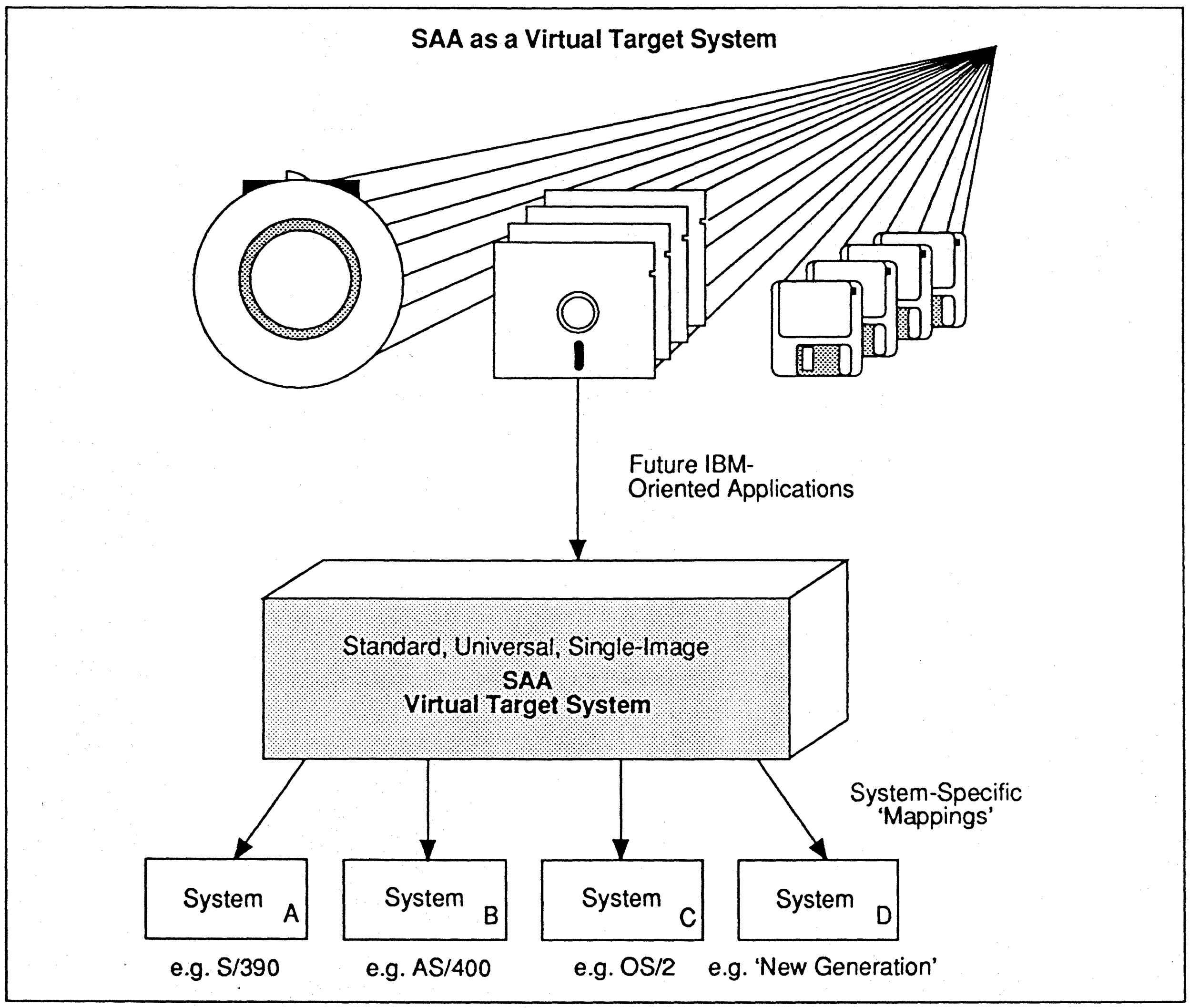

Figure 3

specification is still one of the most far-reaching and practical schemes for platform-independent application development. The UNIX-specific X/Open portability standard and the OSF Motif/ distributed computing environment combo are its only competition.

SAA—The Hidden Agenda

It is becoming clear that SAA's primary objective is to ensure that future application programs destined for the IBM arena can be designed, developed, and used in a standard and consistent manner, irrespective of the eventual target standalone computer system or distributed multiprocessor environment. To realize this, SAA has to insulate application developers as well as end users from the incompatibilities and idiosyncrasies of computer system entities. Such system specific-entities include:

- actual hardware
- operating systems
- access methods
- communications subsystems
- database managers
- presentation managers
- other application-enabling software, such as compilers and link editors

To realize this, SAA creates an SAA-defined protective shell between application programs and system-specific hardware and software entities, as illustrated in Figure 4.

The SAA-defined protective shell appears as the target system on which applications execute. This virtual system insulates the applications and their users from the characteristics of the actual underlying systems hosting the SAA virtual system and the subject applications. In this respect, SAA can be considered as logically extending the virtual system facility provided by IBM's Virtual Machine/370 (VM/370) operating system family since the early 1970s.

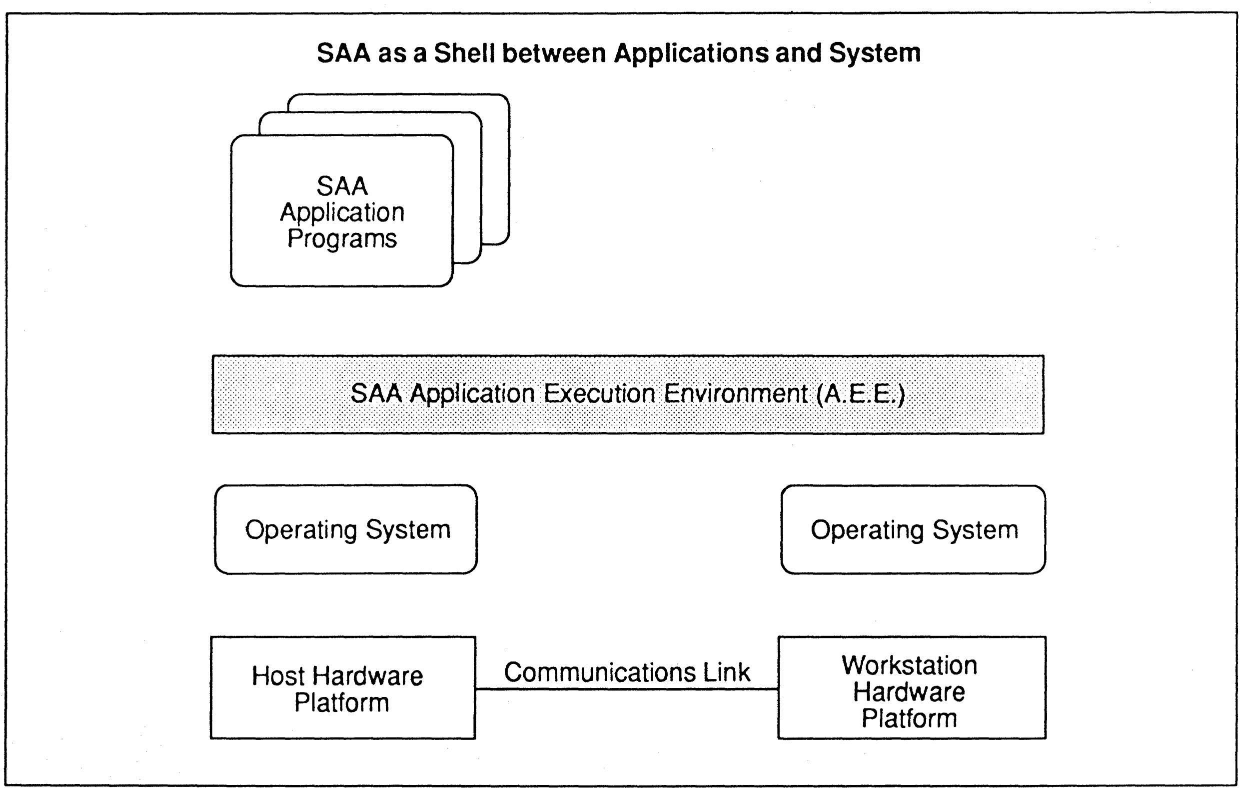

Figure 4

SAA's Similarity to VM

VM allows an installation to create and simultaneously operate multiple virtual S/3x0 machines with differing memory and I/O configurations on a single physical S/3x0 host, as shown in Figure 5. Each virtual S/3x0 thus created appears to be a real S/3x0 host. This enables other standard S/3x0 operating systems, such as MVS or DOS/VSE, as well as VM itself, to be loaded and run unmodified on these virtual systems. An operating system running on a VM-supplied virtual S/3x0 is unaware that it is running on a virtual system and functions in exactly the same way—except for performance—as it would running without VM on a real S/370.

A particular operating system would therefore be able to support the same repertoire of applications whether it is running on a real or virtual S/3x0 host. (The only minor exception to this is the occasional, specialized application that relies on certain S/3x0 host-specific I/O or timing functions.) An installation can transport a complete production system, including all the application programs, from a dedicated, physical S/3x0 host to a VM virtual host

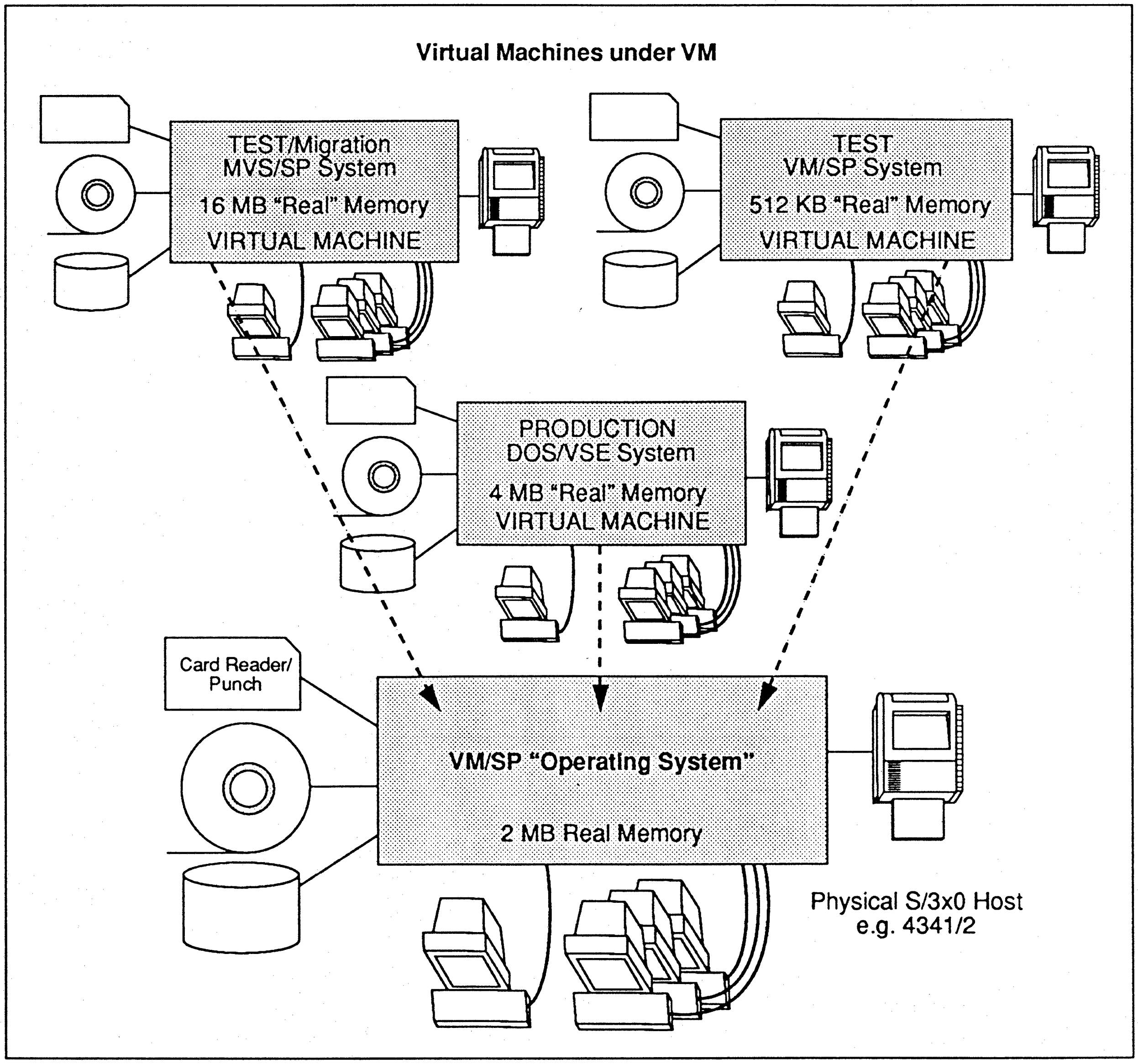

Figure 5

just by "moving" all the disks to and starting up the virtual S/3x0. The applications do not even need to be recompiled or link-edited again on the new system.

SAA as a Hardware-Independent Logical VM

Though IBM has never explicitly positioned it as such, SAA can be viewed as extending the VM concept of virtual machines, which are unfortunately confined to just S/370 hosts, to embrace a much wider range of hardware and operating systems. This SAA virtual system, in the form of a protective shell, is superimposed on top of the application enablers, operating system, and hardware of the various platforms, as shown in Figure 6.

SAA is an Architecture, Not an Implementation

To avoid any potential confusion about SAA's true capabilities, it is prudent to highlight the fact that SAA is currently a paper architecture and has yet to be implemented in its entirety by any one IBM product. VM, on the other hand, is an existing, well-proven product.

This distinction between an architecture and products that are implementations of that architecture can be most easily demonstrated with SNA. SNA is an architecture that only exists in the form of some 4,500-plus pages of specifications. SNA is not a hardware or software product that can be ordered from IBM. Implementations of SNA can be found in IBM products such as ACF/VTAM, ACF/NCP, and NetView, as well as in the communication subsystems of most contemporary IBM cluster controllers and midrange systems.

Many customers have been disappointed with the long time frame associated with bringing implem-

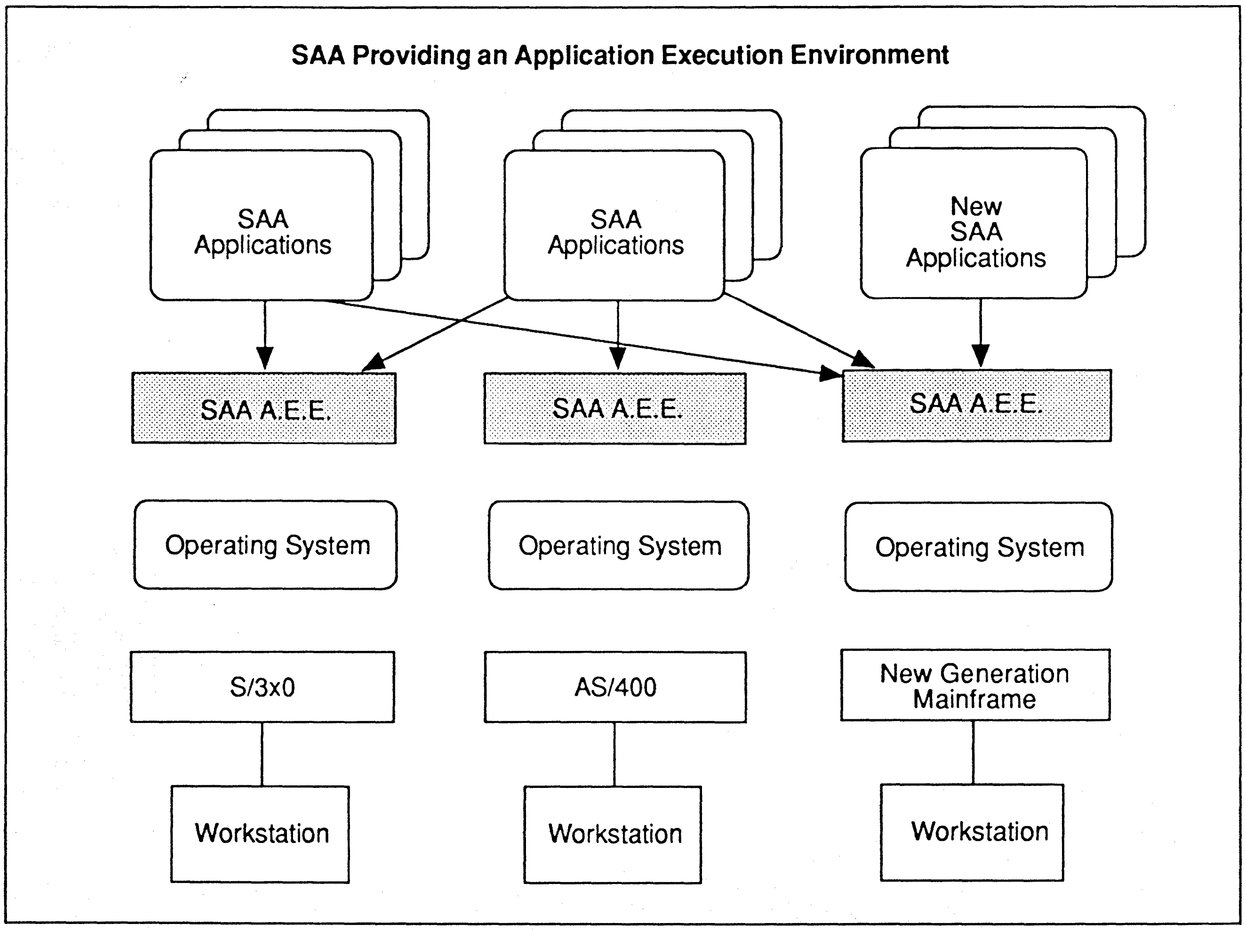

Figure 6

entation of such an extensive architecture as SAA to market. IBM is operating in a more competitive world, with less than half of the market share of the information systems market than it controlled when SNA was introduced. Therefore, SAA has less time to prove its value.

The Case for a New Mainframe Architecture

The S/3x0 architecture which has served IBM and its customers well for many years is now over twenty five years old. (The ES/9000 is not a truly new mainframe architecture, as described in the sidebar "Five Years to the Future.") Even with extensions such as Extended Architecture (XA) which increased its memory addressing mechanism to 31 bits from the original 24, it cannot compete with more recent architectures that exploit the capabilities offered by current technology such as those of the IBM AS/400 and System/6000. For example, just in the area of memory addressing, the System/6000 boasts a 52-bit scheme, which is more than double what the original, non-XA S/370 architecture can support and is still a significant increase to the addressing capability of XA.

Delivering a new host architecture, as *SNA Perspective* expects by the mid-1990s, will not be too difficult for IBM given its vast development resources and the processor design experience it has gained over the last twenty years from developing the S/370 3090, S/38, AS/400, and the RISC-based System/6000. A very difficult challenge, however, is to provide a migration path that will enable customers to painlessly move their existing production applications to the new generation host systems.

During most of the 1980s, there was speculation from some seasoned IBM analysts that S/3x0 migration strategy would revolve around a new generation virtual machine operating system that was independent of hardware architecture. This virtual machine, while supporting both S/370 and midrange systems, would not be tied to a specific underlying hardware architecture. This multiplatform emulation capability was a key concept of the huge and eventually ill-fated Future System advanced research project worked on by most IBM research laboratories in the early 1970s. (Not wasted, the Future System work paved the way for the then radical, object oriented S/38 architecture which, in turn, served as the model for the AS/400 architecture.) Given that an operating system capable of hosting multiple architecture emulations was being looked at even in the early 1970s, it is conceivable that such a system could be perfected thanks to the staggering technological advancements during those twenty years.

A Hardware-Independent VM

A hardware-independent virtual machine is considered by many to be the most attractive and viable means of migrating existing S/3x0 customers to a

Five Years to the Future

Though presented as being based on the System/390 architecture, the ES/9000 range announced on September 5, 1990 is not the manifestation of a new, mainframe architecture that will finally usurp the venerable, twenty-seven year old, S/360–S/370 architecture. The S/390 architecture offers some significant enhancements in areas of multiple processor interconnection (i.e., Sysplex), high-speed, long-distance channel connections (i.e., ESCON), and integrated cryptography support. However, it is still centered around the 32-bit word, 31-(24-)bit addressing S/370(/XA) architecture.

The S/390 architecture is, unfortunately, not the much anticipated S/370 replacement architecture, but rather just another extension to it. The replacement architecture is unlikely to be announced for at least another four to five years.

When announced, it is likely to offer 44- to 64-bit addressing, object orientation à la AS/400 and S/38, multiple instruction execution per cycle à la RS/6000, even more processor interconnection flexibility, and channel speeds at least an order of magnitude higher than today's 10 Mbytes per second.

new family of IBM mainframes. This new virtual machine, which would run on the new mainframes, would concurrently support both S/3x0 and new generation virtual machines. Customers would be able to deploy their S/3x0-based production systems, including their existing S/3x0 operating system, to a virtual S/3x0 machine on a new mainframe, thus immediately gaining the price and performance advantages offered by these systems. They could then begin a gradual application migration process to one of the new generation virtual machines without the cost of maintaining physical S/3x0 hosts and new generation mainframes. This migration strategy is a variation of IBM's recommendation to customers wishing to migrate from DOS/VSE to MVS—install the virtual machine, run both DOS/VSE and MVS in parallel virtual machines, and move one application at a time from the DOS/VSE virtual machine to the MVS virtual machine.

SAA as the Basis for this Hardware-Independent VM

SAA is not the manifestation of, nor the architecture for, such a hardware-independent VM. SAA, however, is based upon and promotes the same overall concept of a virtual machine for *applications* that is independent of the underlying hardware and operating system of the target computer system hosting the actual applications.

In this sense SAA, in addition to its more mundane and immediate role of being the computing model for cooperative processing, can also be viewed already as being the vanguard of an overall S/3x0 migration strategy. IBM is banking on the new generation of 1990s cooperative processing-oriented applications being written according to the SAA guidelines and will expect to receive all their backbone utility services via SAA-specified CPI services (e.g., CPI-C, SQL, Presentation Manager). IBM is encouraging this by:

- unremitting promotion of SAA
- AD/Cycle programs to expedite SAA application development
- incessant software house acquisitions/ investments by IBM
- compelling SystemView concept of total system management

For these strategic applications, SAA's virtual application execution environment rather than the underlying S/3x0, AS/400, or PS/2, will be their logical host. They will, in effect, execute on an SAA virtual machine.

Implementing an SAA application execution environment (i.e., an SAA VM) on top of a new operating system on a new generation mainframe is not a technically challenging proposition. It can be achieved by writing a mapping layer that maps the APIs and services being provided by the SAA application execution environment onto the appropriate interfaces and services on the new operating system. The new operating system and the underlying new generation mainframe hardware will be transparent to the SAA applications. They will continue to execute on the SAA application execution environment, which will just happen to be on a new generation system. SAA applications could thus be easily moved to a new system purely on the basis of this SAA VM concept.

The ability to move SAA applications between disparate systems is not a new concept. If anything, application portability was initially touted by IBM as the prime goal of SAA when it was introduced in March 1987. This is highlighted (and immortalized) in the first edition of IBM's *SAA—Writing Applications: A Design Guide* which started off on page 1 with a section entitled "The Need for Portability," quickly followed by one on the "Value of Portability" on page 4. Though rarely discussed now by IBM, this inherent VM capability of SAA and the possibility of exploiting it as a migration tool should not be forgotten.

In this context of hardware-independent and operating system–independent, virtual application execution systems, *SNA Perspective* notes that the Open Software Foundation (OSF) advocates a similar approach called architecture neutral distribution format (ANDF). ANDF is intended to provide a UNIX-based virtual environment that will allow a UNIX-based application to be run on any processor supported by ANDF, independent of its hardware architecture or the target system for which the application was originally developed. ANDF would permit an application running on Sun SPARC workstations to be ported to run on an IBM System/6000 without requiring code modifications or recompilation of source code. Though there are fundamental technical differences between SAA and ANDF in the process for transporting applications among disparate systems, their underlying motivations and goals are remarkably similar.

The Future Role of SAA

An SAA application execution environment VM will not be the only migration path to the new generation mainframe nor the only avenue through which applications access its capabilities. Rather, it will serve as one means of migrating a very important pool of production applications.

In addition to porting existing applications across from S/3x0s and possibly even AS/400s, there will also be a clamor for the development of new applications that would fully utilize the capabilities of the new machine. Most likely, SAA, in conjunction with the SAA application execution environment, will play a vital role in facilitating the development of such applications.

It is in vogue for applications to obtain the backbone services they require, especially those involving any kind of I/O, from standard service providers via standard APIs rather than attempting to provide such facilities through direct operating system calls or specific hardware interactions. VTAM, VSAM, DB2, CICS, and APPC/MVS are all service providers in this context.

This trend dovetails perfectly with the SAA philosophy, particularly in relation to any innovative capabilities that might be available on a new generation machine. The service providers would perform the necessary system-specific functions to access these new features. Applications would exploit the features through the relevant service providers via new extensions to their APIs, for example, extensions made to the SAA SQL-based database access API when the two-phase commitment control–based distributed data access capability was recently introduced. The SAA architecture would specify what these new services are and provide the guidelines as to how they can be used, while implementations of the SAA application execution environment would provide the new applications with the appropriate high-level access to the service providers.

The success of the major SAA applications has been mixed to date, but *SNA Perspective* believes that recent announcements flesh out the SAA vision. APPC/MVS and the companion CPI-C announcements are the most tangible endorsement of SAA to date. IBM has repositioned SAA from being a context for portable applications to an environment analogous to a virtual machine for cooperative processing.

Systems Application Architecture Documents

Overview	GC26-4341
Common Programming Interface: Summary	GC26-4675
Common Communications Support: Summary	GC31-6810
Common User Access: Advanced Interface Design Guide	SC26-4582
Common User Access: Basic Interface Design Guide	SC26-4583
Writing Applications: A Design Guide	SC26-4362
AD/Cycle Concepts	GC26-4531
Common Programming Interface:	
Application Generator Reference	SC26-4355
C Reference—Level 2	SC09-1308
COBOL Reference	SC26-4354
Communications Reference	SC26-4399
Database Reference	SC26-4348
Dialog Reference	SC26-4356
FORTRAN Reference	SC26-4357
PL/1 Reference	SC26-4381
Presentation Reference	SC26-4359
Procedures Language Reference	SC26-4358
Query Reference	SC26-4349
Repository Reference	SC26-4684
RPG Reference	SC09-1286

(Continued from page 1)

Bridges versus Routers

Users have several ways of interconnecting subnetworks: bridges, routers or brouters. Figure 7 shows the differences between them.

Routers

Routers were used initially for internetworking. These special systems operate at the third layer of the OSI model—the network layer. End systems send internetwork traffic to routers for relay. Routers process the internetwork protocol information and determine which router should be the next relay. Routers continue to relay traffic until it reaches its destination. Since users today usually have heterogeneous networks, most routers today are capable of routing multiple protocols. Routing protocols are used to provide current information to each router. Routers are currently used in local and wide area networking, with a trend toward connecting different sites across a backbone and internets belonging to different organizations because of the need to partition address spaces.

Bridges

Bridges operate at the data link level, giving them the capability of passing many encapsulated protocols transparently. Bridges have also been used to separate a large LAN into smaller segments in order to restrict heavy traffic within a single segment. Because they do not need to do protocol processing, bridges can process data at a faster rate than routers and are less expensive than routers. Bridges can cause problems as the address space becomes large due to the need for unique addresses and the exponential impact of broadcasts. Routers can be used to solve these problems. Bridges are also required for protocols that cannot be routed, such as SNA, Novell IPX, or Digital Equipment Corporation's LAT protocol. Actually, today, OSI is also a nonroutable protocol. The end system to intermediate system (e.g., PC to router) routing protocol has been defined, but the intermediate system to intermediate system (i.e., router to router) protocol is still under development, so the only solutions today are nonstandard, vendor-developed implementations.

Brouters

Bridging routers, or brouters, are the next generation of internetworking products, a combination of both technologies. Brouters bridge traffic which uses protocols that they don't understand or are otherwise unable to route. Using multiple high-speed processors and high-speed buses, brouters will increase their packet forwarding rates.

SNA Perspective believes that brouters will become an increasingly popular internetworking device. Routers will be used for backbones or for connecting internets belonging to different organizations. Bridges will remain a popular lower cost alternative at the local internetworking level, and will become increasingly intelligent, rapid, and capable.

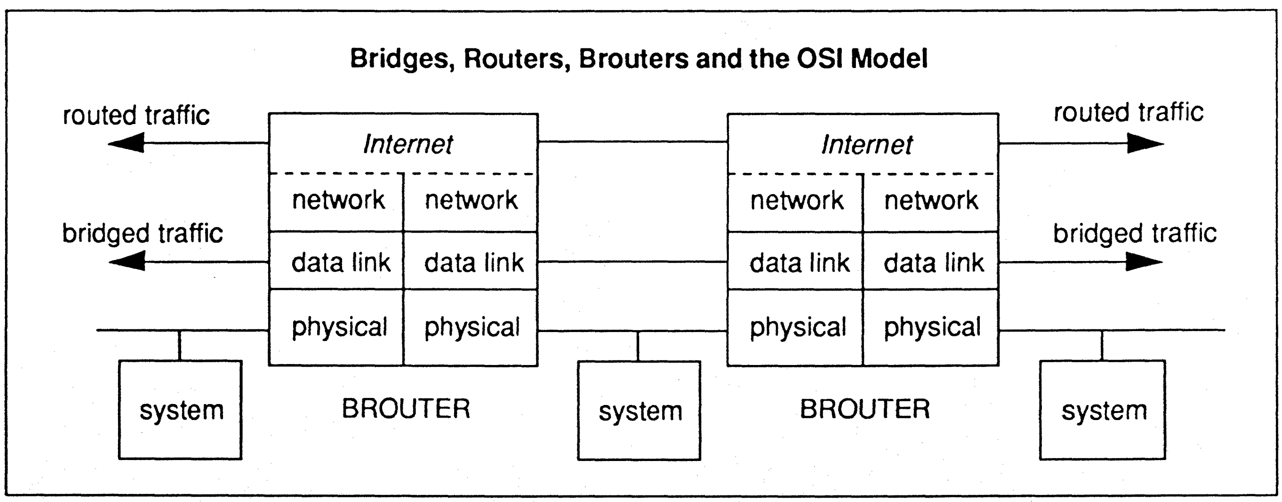

Figure 7

Bridges: Transparent Bridging or Source Routing

Two incompatible strategies have been employed for bridges: the transparent bridging of the Ethernet world and source routing used with the IBM Token-Ring.

Transparent bridges make relay decisions based on destination addresses. These bridges "learn" by monitoring all traffic on the attached subnetworks. Transparent bridges use a spanning tree routing protocol to prevent looping in complex topologies. (Note: spanning tree routing and source routing are second-layer protocols, even though "routing" usually refers to third-layer networking.)

In contrast, source routing bridges follow relay instructions contained in the internetwork traffic itself. End systems broadcast special packets, called discovery packets, which each bridge relays after placing its address inside. When such a packet reaches the destination, a record exists of each bridge used. This record defines a fixed path for traffic between a pair of systems for a given session. A new session can use a different path.

The Institute of Electrical and Electronic Engineers (IEEE) adopted a standard in 1989 that allows bridges to operate in both modes. Each internetwork packet must carry a bit indicating whether source routing information in the packet is used or whether transparent bridging is needed. All previously installed Token-Ring bridge adapters would need to be replaced to support the new standard. IBM has incorporated the new IEEE standard in its 8309 Bridge. Vendors are also offering Token-Ring-to-Ethernet bridges to address this problem.

Network Layer: Connectionless or Connection-Oriented

Maturing internets use a combination of both approaches. The OSI standards for the network layer include options for both connectionless network services (CLNS) and connection-oriented network services (CONS).

TCP/IP's internet protocol (IP) delivery service operates in a *connectionless* or *datagram* mode. This type of protocol treats each piece of traffic as an independent event. Routers do not hold any traffic until it is acknowledged, check the sequence of incoming traffic, or recover from any transmission or protocol error. IP routers have one option when they encounter obstacles: discard the datagram. This inherent unreliability is handled by the transport layer in each system. Each datagram in a given session can follow a different path to the destination (see Figure 8). Connectionless operation ensures that router resources are used to relay traffic as quickly as possible.

The SNA backbone, in contrast, uses a *connection-oriented* or *virtual circuit* protocol. This type of operation requires a concatenated set of connections between nodes relaying traffic (as shown in Figure 9). The sequence of traffic is preserved and recovery from transmission errors is handled by the nodes. This type of operation provides a more reliable service at the expense of using more resources in each relay node. Further, each packet for a given session in a connection-oriented service must follow the same path through the backbone since the nodes maintain information about each virtual circuit. Failures in the path require development of alternate routing and communication with all affected nodes.

There has always been spirited debate between the partisans of each approach. Trade-offs exist for each approach. Connectionless operation uses fewer resources in each router:

- unacknowledged datagrams are not stored
- less processing is required since no protocol states must be maintained
- there are no management messages for acknowledge or retransmission requests

However, each datagram must carry more information since the routers keep no information between datagrams. Connection-oriented protocols require more router resources, particularly at connection setup and take-down, but allow shorter packets.

In essence, connectionless protocols say such functions as sequencing and error recovery are done outside the network; connection-oriented protocols say the network provides the service. IBM believes that all protocol families should provide both, as both are required for different situations. The application type, not debates among partisans, will determine which type is used.

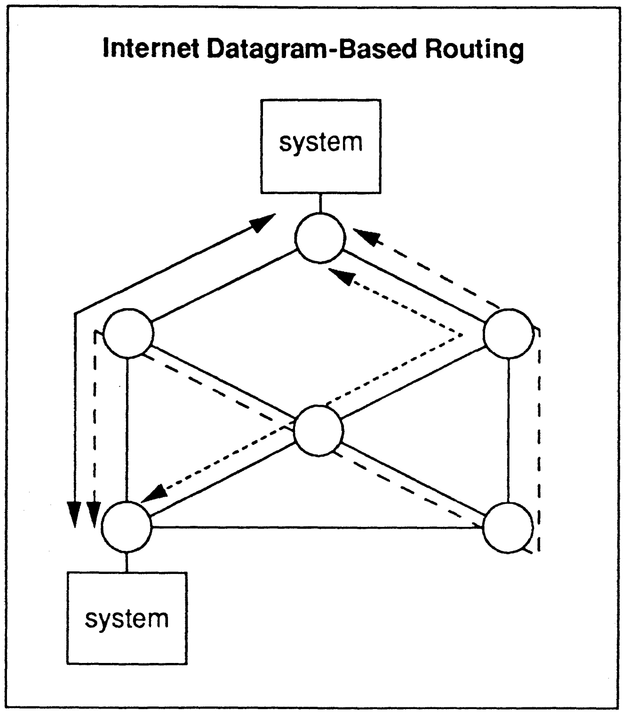

Figure 8

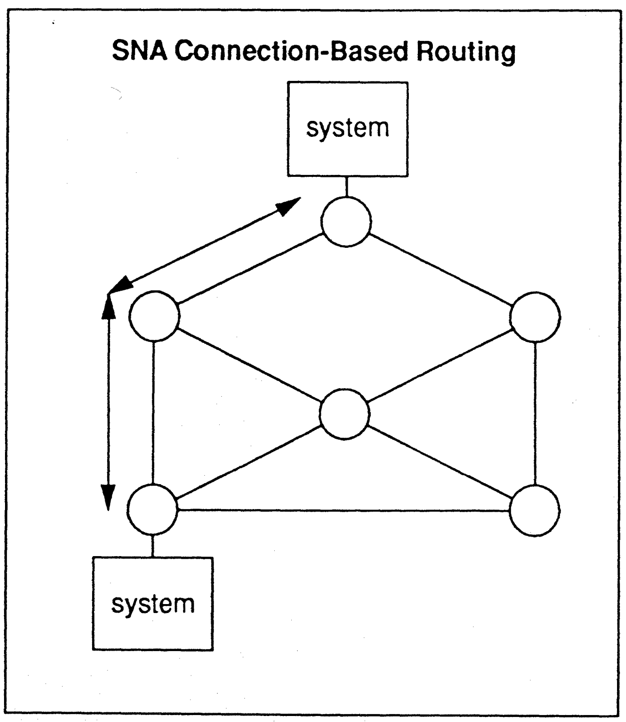

Figure 9

SNA Perspective believes that emerging internets will have backbones that are actually collections of both types. For example, circuit switching with T-1 or T-3 may be more cost-effective for high-volume transfers while the packet-switched X.25 is more suited for widespread coverage and short sessions.

For internetworking, *SNA Perspective* sees SNA going toward connectionless operation at layer three. We see this theme continuing in the long run, as network infrastructures become more rich and protocols at each layer become more robust, that eventually all layers below the session layer will be primarily connectionless.

Summary

SNA Perspective sees bridges and routers both continuing to be valuable tools for internetworking, with brouters also being popular for some environments. All will become increasingly capable and be able to handle increasing loads and larger networks.

The source routing versus transparent bridging controversy was ameliorated to a great extent by the IEEE source routing transparency standard, but the incompatibilities in the current installed base will cause problems for internetworking for some years. *SNA Perspective* believes both protocols will continue to be used, and this controversy will become a nonissue.

SNA Perspective sees protocols at the lower levels increasingly emphasizing connectionless communication in the long term, except for local high-volume applications. We also see SNA headed in that direction at the network level.

The next installment in this series will address additional issues at the network layer, issues at the transport layer, and problems in addressing especially internetwork uniqueness.

Architect's Corner

Things That Go Bump in the Night *or* Management of Simple Systems

by Dr. John R. Pickens

`...news flash: IBM and 3Com announce joint work for Heterogeneous LAN Management (HLM), an OSI-based network management standard for LANs...`

`...news flash: IBM and 3Com announce availability of HLM specifications...`

`...news flash: 3Com/IBM propose HLM to IEEE 802...IEEE 802 halts its 802.1b ballot on LAN network management in order to incorporate elements of HLM...`

Reactions:

- OSI community welcomes the increased scope of network management to cover simple systems including the desktop
- SNMP community withholds support —SNMP is supposed to be simple enough
- Early-partner vendors provide positive support
- Users defer judgement

I've received many calls on these events over the past few weeks. A common question is, given IBM's strong commitment to SNA Management Services, what does this move toward OSI-based LAN management mean? What is HLM, anyway? Ok, here's a brief tutorial and analysis.

HLM contains several elements: protocol, APIs, and MIBs. (See Figure 10.)

Protocol

HLM utilizes CMIP protocol, including the OSI ROSE sublayer (ROSE is the OSI remote operations service element—a simple remote procedure call). Compared to OSI CMIP, HLM CMIP replaces OSI Presentation/Session/Transport with direct use of LLC Type 1 services (connectionless best-effort datagram services). Also, OSI association control is replaced by an architected group-addressed registration function. HLM can be thought of as OSI CMIP with an LLC Type 1 profile. Some call it CMOL (CMIP over LLC), but I prefer CMIP/LLC-profile. (See Figure 11.)

The design center for HLM operation is simple systems. Simple systems are defined as systems with memory constraints or systems which only run low-level protocols. Note: this scheme also works for "broken systems"—an OSI term for systems which are capable of running only lower-layer protocol stacks when broken.

APIs

A set of APIs are defined for manager side and managed system side application development. There are two levels of API, one for use by protocol-based layer management (LME) and another for use by Management Processes and their Managed Entities (MP, ME).

MIBs

A MIB is the specification for management functionality within managed objects. The current HLM specification specifies MIBs for Ethernet,

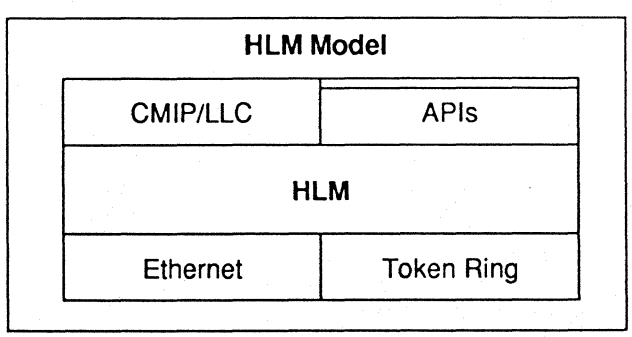

Figure 10

Token-Ring, and intelligent access units. HLM also defines an advanced method for managing counters and thresholds which requires no divide operations. Finally, in consideration of simple system requirements, the MIBs are constrained to use easy-to-parse data structures, only allow simple OSI filters (e.g., a simple compare and set), and only allow an OSI scoping level of one (to read tables).

Analysis

What does this work mean to IBM? First it foreshadows an evolution toward the OSI management paradigm. HLM operates at the link layer, but I would expect future IBM moves to allow OSI CMIP to operate over SNA (APPC).

Does this move signify a decommitment to IBM's SNA Management Services? Well, yes and no.

- Yes—if the HLM model is indicative of future moves, some management functions, e.g., alerts and commands, are going to be obsoleted and replaced by CMIP alternative mappings.
- No—I expect IBM to continue to support and add SNA-specific management functions beyond standards such as security, encryption, microcode distribution, etc.

Can HLM operate beyond the scope of a bridged LAN environment? Yes, but with a CMIP relay (not yet defined). Access beyond a bridged environment (i.e., across an SNA internet) requires some kind of relaying function. This function might be placed in management stations, hubs, bridges, routers, or servers. In an SNA environment, for example, a manager station could then use CMIP/SNA-profile (not yet defined) to cross a routable internet and then the relay could map to the HLM CMIP/LLC-profile. The real issue here is who defines the CMIP mapping mechanism: IBM or the OSI Network Management Forum?

Entering the 1990s, it is fittingly symbolic for IBM to promote this work. This move is another evidence of IBM's plans to repaint SNA with an OSI paint brush, as I have suggested in previous columns. Incorporation of OSI CMIP management in millions of simple systems (desktop PCs) bodes well for ultimate support of OSI management in all systems—bridges, routers, hubs, communications controllers, cluster controllers, gateways, midrange systems, and mainframes.

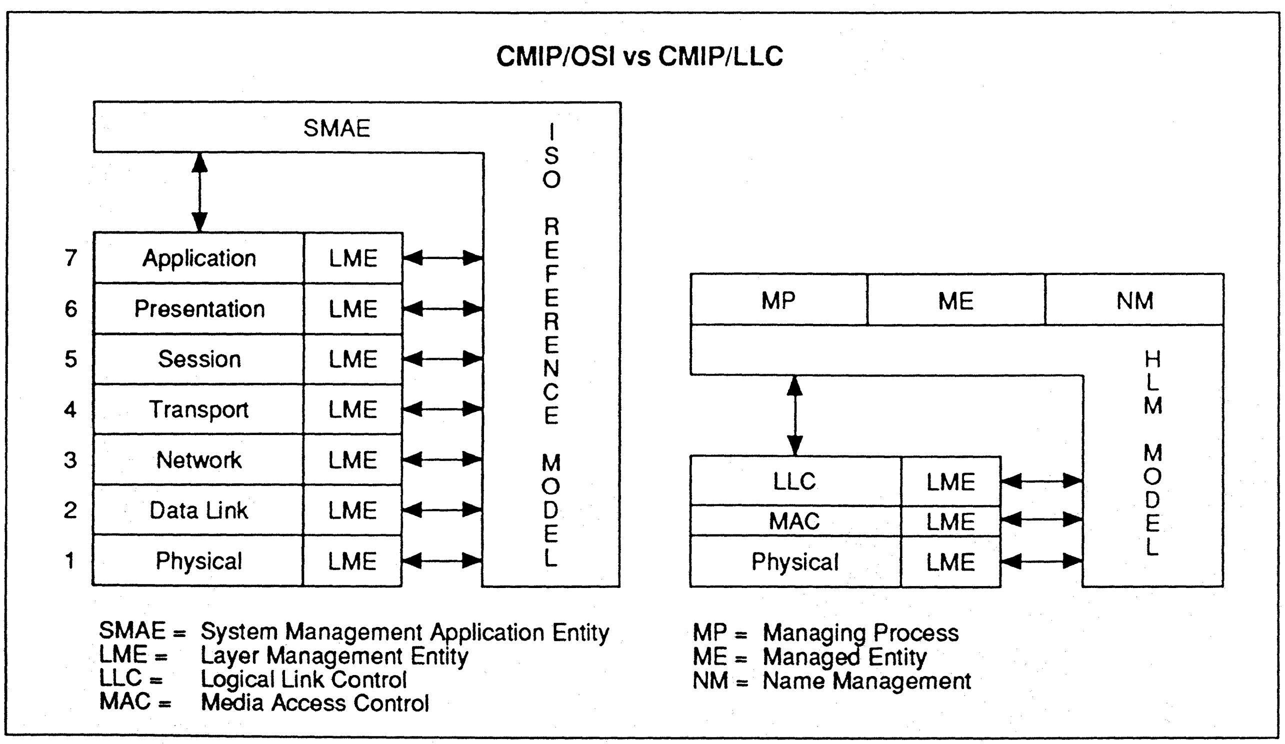

Figure 11

IBM Announcements

OfficeVision Delayed—Again

IBM appears to have missed its goal of getting OfficeVision Release 2 out the door in 1990. At press time, IBM acknowledged there would be delays but had not reported a revised availability schedule, though word was expected from IBM by year's end. Feedback from several sources indicates that mid-1991 will be the new target. The delays appear to relate to problems in development of the workstation software. This is the second scheduling delay for OfficeVision. The first delay was announced in February 1990, a month before the initially scheduled shipment date. This new delay will push it two years past its announcement date.

SNA Perspective believes that the application's development has been hampered by the weight of incorporating SAA compliance in an already complex product. These further delays are certain to harm OfficeVision's potential market success. However, it is not expected to harm SAA's eventual acceptance, which has always been a long-range strategy. However, OfficeVision was intended to be SAA's flagship product, and these delays tarnish SAA's glow somewhat.

IBM, 3Com Unveil CMOL Specs

IBM/3Com HLM Specifications Available

3Com and IBM recently announced their jointly-developed network management specifications are available for industry-wide review. Publication of the draft HLM specifications follows preliminary evaluation by technical teams of four leading computer software companies—Banyan Systems, Microsoft Corporation, Novell Inc., and The Santa Cruz Operation.

HLM is a set of architectural documents that attempt to define a management platform to satisfy the management needs of mixed Ethernet and Token-Ring environments regardless of the operating system supporting each LAN node. (See Architect's Corner in this issue.) *SNA Perspective* feels that today's network management is more complex because most major LAN installations have multiple media types and operating system environments.

Draft HLM specifications are available without charge from IBM and 3Com. Interested parties may obtain the specifications by writing to IBM Corporation, P.O. Box 12195, Department C13, Building 002, Research Triangle Park, NC 27709 or to 3Com Corporation, Attn: Jim Healey, 5400 Bayfront Plaza, Santa Clara, CA 95052.

IEEE 802 Accepts HLM Approach for Network Management

In November 1990, the IEEE 802.1b committee reviewed the specification and elected to adopt portions of the HLM architecture as the new 802.1b LAN/WAN management standard proposal. It is anticipated that the IEEE 802.1b will become an accepted standard by the end of 1991. Other standards bodies will be approached with the intent of gaining acceptance of all or part of the HLM specification in other standards.

IEEE 802 defines standards for LAN media, metropolitan area networks, hubs, and bridges. IEEE 802.1b defines the network management architecture used by all IEEE 802 groups.

Prior to the new technical approach, IEEE 802 had defined an almost-CMIP management model and a set of 802-specific PDU formats for the various CMIP functions. However, the model was missing key functions such as CMIS create/delete (useful for counters and portable gauges—portable gauges are a recently proposed 802 concept) and scope (at least one for tables) and filter (at least simple expressions for compare-and-set functionality). Also, the IEEE 802 management flows did not utilize the ROSE remote procedure call function.

Like the HLM approach, the 802 standard required only the services of LLC Type 1 (connectionless services).

IBM to Move Communications Systems to UK

The Communications Systems division headquarters staff, including general manager Ellen Hancock, will be relocated from New York to Staines, England, just outside of London. The SNA development staff will remain in Raleigh, North Carolina.

IBM noted that this move will put CS headquarters closer to the La Gaude, France communications processor development, the CICS lab in Hursley, England, and the OSI and NetView development in Rome. *SNA Perspective* sees this move as more of a public relations move than substantive organizational change. It is intended to appeal to European companies by increasing IBM's European presence, especially with OSI, in anticipation of 1992.

Quieter IBM Reorganizations

IBM has been quietly moving more of its OSI and NetView development function to Rome. This will affect U.S. IBM staff in several locations, including Palo Alto, California.

IBM has also quietly established the Systems Structure and Management group in Somers, New York. This group reports to Earl Wheeler in the Programming Systems group, which champions SAA among other functions. Systems Structure and Management will coordinate integration between SAA and AIX (see *SNA Perspective* April 1990).

Another new group also in Somers, the Programming unit, reports into the Personal Systems group. This new unit is charged with coordinating common subsystems for OS/2 and AIX, and will be responsible for all personal computer and workstation operating system development.

SNA Announcements

AT&T Makes Bid for NCR

AT&T has been attempting to purchase NCR, a move NCR characterizes as unfriendly. The offer is for $6.1 billion, or $90 per share. AT&T indicates that, after the merger, the NCR subsidiary would run the computer operations for both companies.

Some synergies between the two firms exist, including a commitment to client/server computing, Unix, and open systems. Also, their industry dominance does not overlap—AT&T is strong in government and telecommunications, while NCR's strength is in finance and retail. Most analysts believe that the AT&T 3B2 and the NCR Tower would be discontinued if the merger were to happen. However, their demise seems likely at some point in any event, given the trend to Intel and RISC based architectures.

SNA Perspective believes that the acquisition of the NCR Comten subsidiary could prove a beneficial part of such a merger. NCR Comten, an active player in the IBM compatible communications processor market, has been focusing on processors that can handle OSI and TCP/IP traffic as well as SNA.

OSI/NMF Registry

The OSI/Network Management Forum is searching for a vendor-independent organization to develop a registry of the objects and attributes used to manage networks. A similar registry is run by the University of Southern California for TCP/IP, supporting network management information from more than 130 vendors. One function of the registry is to provide access for users and vendors to network management information on products (devices and software—called objects) and their variables—called attributes. This registry would be a short-run solution. In the long run, the OSI implementors workshops may set up an International Management Information Library.

In addition, the OSI/NMF is considering converting simple network management protocol (SNMP) variables to the OSI format, so they could be stored in an OSI directory. This move is in acknowledgement of the popularity of SNMP and the probability that SNMP, like TCP/IP, will coexist with CMIP and OSI for the long run. *SNA Perspective* believes that IBM will work to have its network management products evolve to support the OSI format in the long run.

Network Systems Enhances Supercomputer Links

Network Systems Corp. of Minneapolis, Minnesota, introduced four new switches for connecting supercomputers and their servers. These switches, the PS8-8 and three members of the PS32 line, can send and receive data with an aggregate speed of 25.6 gigabits per second. They all comply with the ANSI draft High-Performance Parallel Interface (HIPPI) standard, which relates to high-speed access over distances up to 25 meters. The PS32 line was jointly developed with supercomputer vendor Cray Research, and will not be available until the second half of 1991.