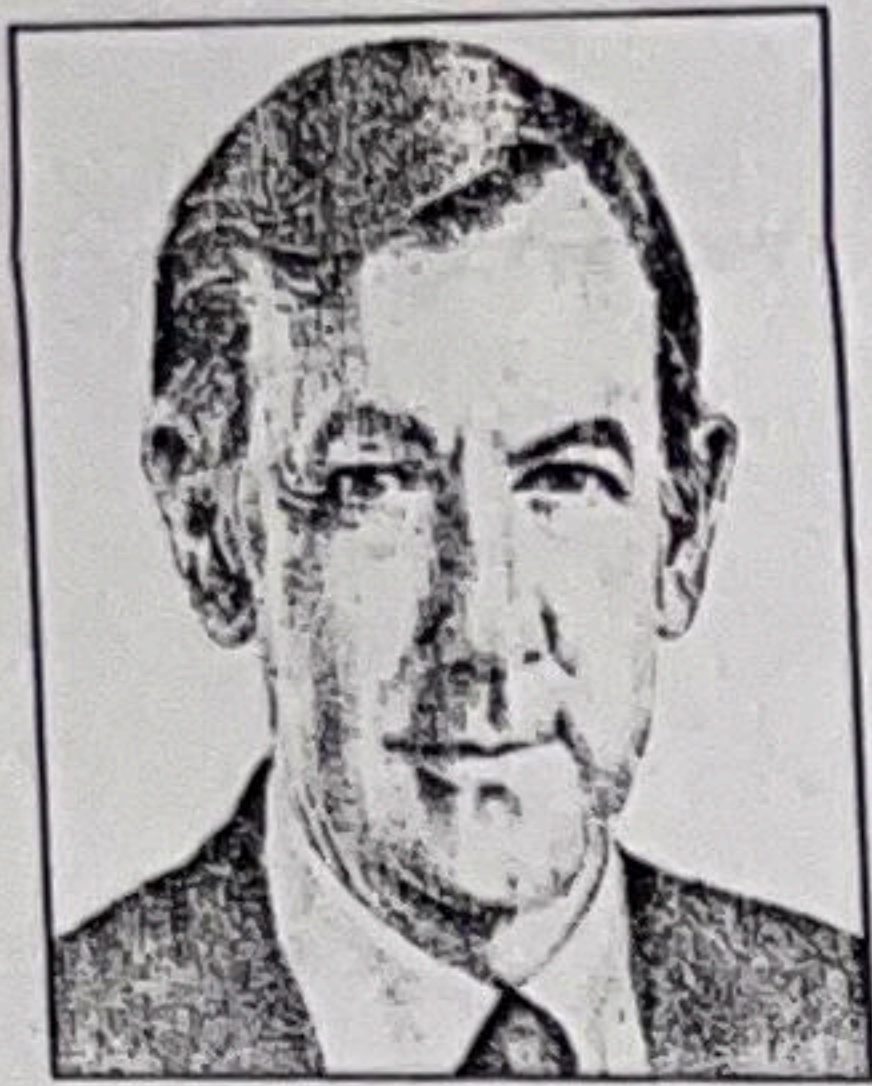


APPLIED INTELLIGENCE

SAA Has Strengths, Limits in Integrated Environments



JAMES MARTIN

This is part 4 of a series of articles on IBM's Systems Application Architecture (SAA). The introduction of integrated computing environments such as SAA will have a major impact on the software technologies of the '90s.

A major strategic objective for organizations in the '90s will be to replace centralized computer facilities with distributed computing systems made up of multiple cooperating processors.

In a distributed-processing environment, each element of the computing network is interconnected and optimized to do what it does best. User interfaces and local processing are supported on desktop workstations. Mainframes and midrange computers are used to store central data files and share processing functions.

Integrated computing environments such as IBM's SAA are designed to support the distributed computer architectures needed for the '90s. As I said in earlier articles in this series, it's vital for managers to weigh the strategic strength of these architectures and lay the groundwork for their introduction.

I'll discuss here some of SAA's major strengths and limitations to illustrate the issues that need to be addressed in evaluating integrated environments.

The primary strength of SAA is related to its goal of delivering cooperative applications. Cooperative processing will grow more important in any organization that is using more than a single PC. What is now an extremely difficult programming task will become much more straightforward as SAA matures.

Applications will be easier to port to different platforms in SAA-compliant environments. A pricing module can run on the System/370, AS/400 or PS/2 and can be used as part of a distributed application on any of those platforms.

Programmer productivity will be dramatically enhanced by SAA. The same skills will be used to implement cooperative applications on all of the platforms in the environment. In addition, the SAA Application Development Environment will make it easier to develop and maintain large suites of applications.

SAA provides for data distribution. Data can reside on any machine in the network. Corporate data can be stored on a centralized mainframe, departmental data can be stored on minicomputers and personal data can be stored on programmable workstations.

Distributed data is only partially implemented today and requires the programmer to know the data's physical address. Successive releases of SAA will enable data to be located anywhere in the network and accessed automatically.

The graphical presentation standards incorporated in the SAA Common User Access (CUA) interface will enable us-

ers to move easily from one application to another since they will all use the same interface conventions. The user interfaces will be easy to use and will take advantage of the programmable workstations' graphical features.

New user-interface technology will soon be incorporated into SAA with the next release of CUA. The use of more graphic interfaces on the programmable workstations will be encouraged. Using the highest-level Dialog Interface of SAA, advanced user-interface features can be implemented automatically.

For example, the programmer might specify that the user select one item from a list. The Dialog Manager compo-

sitions purchased from third-party vendors will follow the same communication, database and user-interface standards of in-house applications.

SAA will push IBM into providing superior systems-software products. The old piecemeal products will give way to a well-integrated set of application-enabling and system-support tools.

One of the major limitations of integrated environments is their proprietary nature. SAA is a proprietary product from IBM, not the result of a consortium. Managers need to assess how open SAA really is.

SAA is open in that it specifies a common set of software interfaces across

However, SAA locks in IBM as a vendor. If SAA becomes the standard for an organization, then IBM or IBM-compatible hardware and software will be preferred in that environment.

Competing vendors are forced either to accept the SAA standard or develop their own. Customers are forced either to accept SAA as the standard or forgo its benefits.

IBM is responsive to the demands of the marketplace for an open architecture, as can be seen in the case of the Open Systems Interconnect (OSI) communications standards. SAA had originally used IBM's proprietary Systems Network Architecture (SNA) for communications, but many customers asked for support of the International OSI standard. That support has been announced and partially implemented.

In addition, much of the marketplace prefers Unix as the environment for programmable workstations. IBM is responding to this demand with its version of Unix, AIX, which is evolving to maintain compliance with the communication and programming interfaces of SAA. This will allow IBM customers to use either Unix or OS/2.

Difficulties in Choosing

Still, difficulties remain, because the programmer must choose between using the SNA or OSI communications interface. The SAA Common Communication Services Interface will resolve this issue by providing common communication programming services that allow users to network both SAA and non-SAA machines.

The same problems will exist with the programming interfaces, database services and user interface. IBM is sensitive to language standards and will most likely continue to support international language standards. The database standards will be driven by its own products, as will the user-interface standards.

An important current limitation is the lack of SAA compliant components. Many of SAA's goals are not realized with current products. For example, the Cross System Product development environment has been identified by IBM as a strategic application enabling tool, but it doesn't enforce CUA interface standards, and it exists only on the System/370.

Related to product availability is the limitation of scope. SAA can be viewed as clusters of technology. Some clusters have been announced, but only partially delivered for SAA.

Next week, I'll describe how cooperative processing applications are developed using SAA functions. ■

The James Martin Productivity Series, an information service updated quarterly, is available through High Productivity Software Inc., of Marblehead, Mass. (800) 242-1240. For information on seminars, please contact (in the United States and Canada) Technology Transfer Institute, 741 10th St., Santa Monica, Calif. 90402 (213) 394-8305. In Europe, contact Savant, 2 New St., Carnforth, Lancs., LA5 9BX United Kingdom (0524) 734 505.



SAA plays a critical role in IBM's strategy to increase its dominance in hardware and software systems. But this strategy doesn't necessarily conflict with IBM customers' needs.

ment of CUA decides how to present the list to the user. Today it presents the user with a menu. In the future it may present iconic radio buttons instead. In either case, the program does not need to change.

The SAA Common Communication Services interface will allow for easy connectivity between the machines in an enterprise, including non-SAA machines. As the Common Communication Services interface matures, this connectivity will become more and more transparent.

Vendors of packaged applications will provide products that can be integrated more easily into an organization. Appli-

multiple hardware platforms. It supports both proprietary services and protocols such as Systems Network Architecture and LU 6.2 as well as international standards, including adherence to the Open Systems Interconnect reference model for communications.

However, SAA plays a critical role in IBM's strategy to increase its dominance in both hardware and software systems in the '90s. This strategy doesn't necessarily conflict with IBM customers' needs, however. IBM customers are currently expressing a growing need for greater connectivity and distributed-processing capability. SAA is designed to meet that need.