

Application enabling in SAA

by D. E. Wolford

The Common Programming Interface (CPI), one of the four key elements of Systems Application Architecture, comprises a growing set of programming languages and services. The CPI indirectly offers end-user access through the Common User Access by providing the application developer with the necessary interfaces. The CPI addresses the application development requirement for portability of applications and programmer skills. As the CPI continues to expand, it addresses the requirements for access to host data through intelligent workstations and for transparent access to remote data and applications.

The announcement of Systems Application Architecture (SAA) on March 17, 1987 established the Common Programming Interface (CPI) as one of the four key elements of SAA. This announcement also set a clear direction for the management and availability of consistent application programming interfaces. Figure 1 illustrates the elements of the announced CPI. A number of implementations of these interfaces are available, and work continues to expand the interface implementations available in all SAA environments, while at the same time extending the CPI by enhancing the announced interfaces and adding new languages and services.

Objectives of the CPI

Four discrete objectives have been described for the CPI.

Consistency for the end user. This objective involves support of the Common User Access (CUA)¹ element of SAA through the CPI by providing for the application programmer a useful, complete, and productive interface that allows conformance with the CUA. It also means providing an interface that gives the application programmer choice and flexibility while

ensuring that he or she has consistent user interfaces and allowing the degree of control required by the application.

Programmer productivity. Consistent interfaces are provided that make application programmer skills and the programs portable among the SAA environments.

Applications that exploit the strengths of individual SAA environments. The CPI objective is to empower such applications by providing the needed function through the CPI, such as the *Communications interface*, or by providing *transparent enabling*, similar to the method by which access to distributed data is enabled through the SQL and the high-level language file I/O interface.

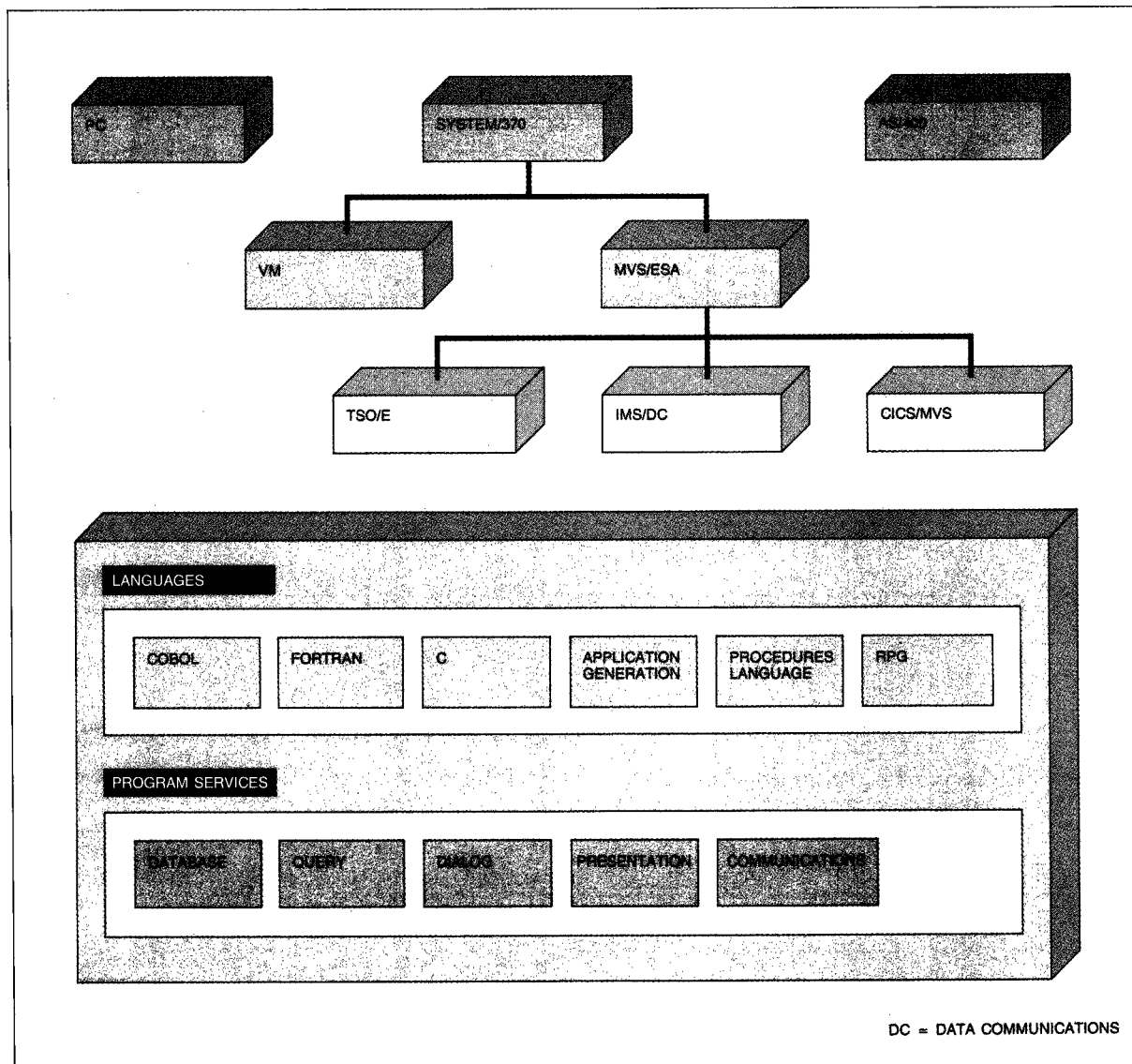
An enterprise-wide application development environment. The CPI must address the requirement to provide an application development model, tools, and services to meet a growing customer need.

Consistency for the end user

One objective of the SAA is especially clear—consistency of the end user's access to IBM's SAA environments. Support of the CUA specification is also the CPI's first objective. The Dialog and Presentation interfaces of the CPI have accepted the requirement for the support of the CUA. These interfaces will enable the CUA and, in time, offer enforcement of

© Copyright 1988 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

Figure 1 CPI elements



CUA conformance to those who desire it. The paper in this issue by Uhler² addresses enabling the CUA and the direction to be taken by enabling in the future.

Productivity enhancement for the application programmer

A predominant theme in the designation of the current elements of the CPI is application productiv-

ity. The CPI addresses productivity in two ways: *portability of applications* and *portability of skills*.

Portable applications. Applications developed using the CPI can be moved from one SAA environment to another with a minimum of rewriting. While there is no guarantee of complete application portability among the SAA environments, the CPI is being designed, managed, and developed in the SAA environments to minimize the amount of rewriting required

to enable portability. This portability of applications is expected to be especially beneficial for vendors of software that runs on IBM systems. These vendors can significantly expand the market for their products with less additional development effort by using the CPI.

Portable skills. The application programmer who learns the necessary elements of the CPI to code an application in one SAA environment can, without relearning or learning new CPI elements, use the same skills to write programs in any other SAA environment. The CPI objectives embrace another equally potent productivity enhancement for application developers: Consistency of the language and service implementation guarantees more portable skills. The CPI language and service implementations in the System/370 (VM/CMS, TSO/E, CICS/MVS, and IMS/VS Data Communications), IBM OS/400™, and OS/2™ Extended Edition are virtually syntactically and semantically consistent.

The paper by Haynes³ on the Application Generator (AG) element of the CPI traces the history and future of several implementations of the AG, namely the Cross System Product implementations. This fourth-generation language continues to meet application programmers' requirements for portability of applications and application development skills. Because

of these attributes, the Application Generator is a central component of the CPI.

Enabling applications that exploit the strengths of the SAA environments

Another objective of the CPI is to make possible applications that require data or programs residing in other interconnected computers. The Common Communications Support element of Systems Application Architecture provides the necessary interconnections; the CPI provides the programming languages and services.

Program-to-program linkage. Access to the computing and data storage capacity of IBM's transaction management subsystems—IMS/VS Data Communications and CICS/MVS—from all of an enterprise's computers is becoming steadily more interesting to IBM users and is a requirement for some. This requirement has been accepted in the CPI, and a first step has been taken toward enabling the linkage through the Communications interface in each of the SAA environments. Through this LU 6.2 (Advanced Program-to-Program Communication) implementation, application segments in two environments can be linked, providing access to IMS/VS Data Communications and CICS/MVS data and applications from intelligent workstations (IWS) running

Figure 2 Intelligent Workstation (IWS)

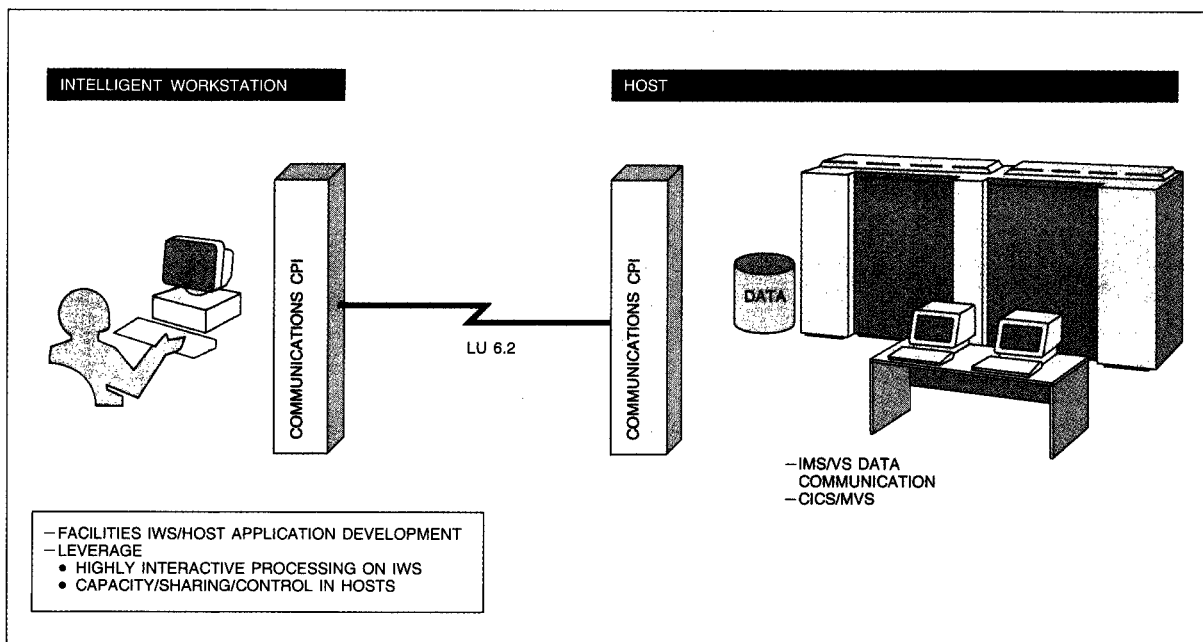
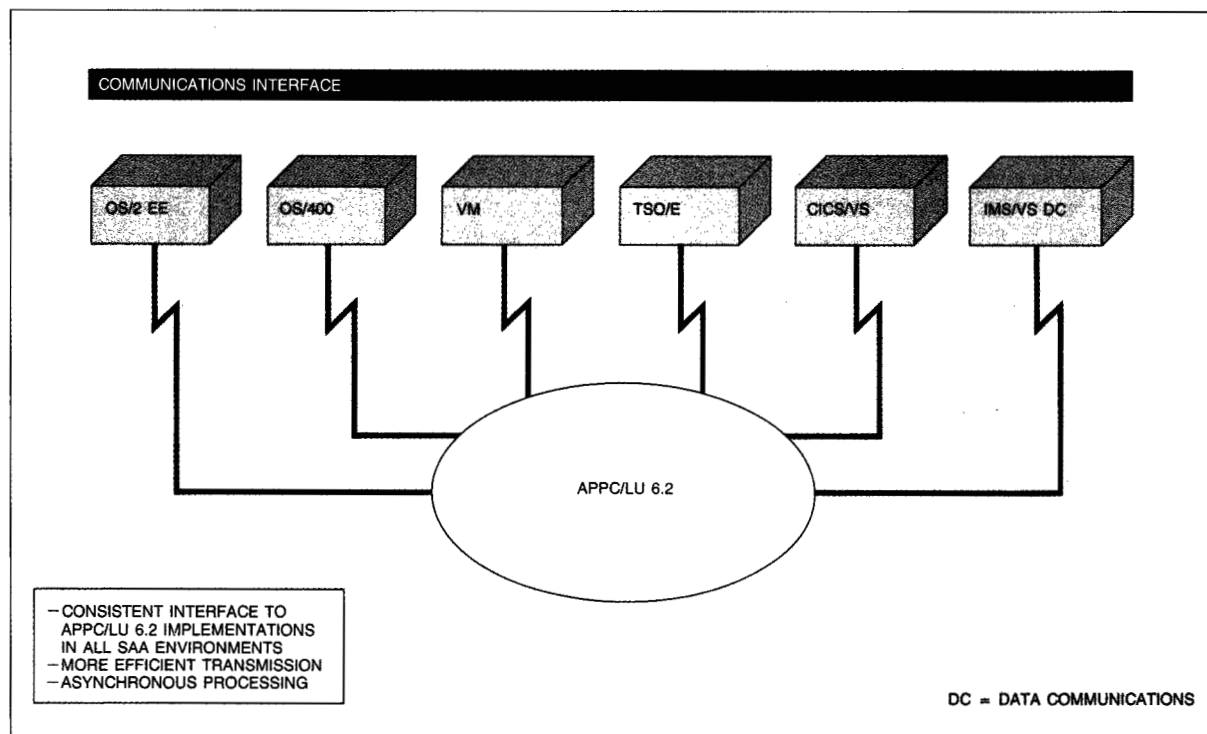


Figure 3 Any-to-any connectivity in SAA



OS/2 Extended Edition. Figure 2 illustrates this configuration. In time, implementations of the Communications interface will make it possible for an application running in any interconnected SAA system to obtain access to data and applications in any other SAA system, as illustrated in Figure 3. Further, this Communications interface is the foundation for the next advance in improved interconnection across the enterprise: the ability to obtain access to a program or a programming service in another computer without knowing where it resides.

Access to remote relational data and flat files

Access to certain kinds of data without knowing the computer in which they reside is supported through the CPI. Relational data are available through the database interface's Structured Query Language (SQL). Files that are of sequential, keyed, or direct organization are available through the high-level language file input and output language statements. These CPI implementations are further addressed by Reinsch⁴ and Demers⁵ in this issue.

Providing an enterprise-wide application development environment

Beyond the languages and services that constitute the CPI, application developers need an integrated environment for designing, modeling, developing, integrating, testing, and maintaining their applications systems throughout their life cycle. This environment must provide consistent access to the application developer; it must offer a development process that allows easy monitoring; and finally, the environment must offer a full set of application development tools.

OS/400 and OS/2 are trademarks of International Business Machines Corporation.

Cited references

1. R. E. Berry, "Common User Access—A consistent and usable human-computer interface for the SAA environments," *IBM Systems Journal* 27, No. 3, 281-300 (1988, this issue).
2. S. Uhler, "Enabling the user interface," *IBM Systems Journal* 27, No. 3, 306-314 (1988, this issue).

3. W. K. Haynes, M. E. Dewell, and P. J. Herman, "The Cross System Product application generator: An evolution," *IBM Systems Journal* 27, No. 3, 384-390 (1988, this issue).
4. R. Reinsch, "Distributed database for SAA," *IBM Systems Journal* 27, No. 3, 362-369 (1988, this issue).
5. R. A. Demers, "Distributed files for SAA," *IBM Systems Journal* 27, No. 3, 348-361 (1988, this issue).

Dean E. Wolford *IBM General Products Division, 555 Bailey Avenue, San Jose, California 95141.* Mr. Wolford is currently a senior programmer at GPD's Santa Teresa Laboratory. He joined the System Development Division's Programming Laboratory in 1969, and held positions from 1969 to 1975 as an information developer, planning and writing user documentation for such programming products as COBOL, PL/I, VSAM, and other data management components of System/360 and System/370 operating systems. From 1975 to 1980, Mr. Wolford managed information development for IMS/VS, and, from 1980 to 1984, information development for DB2, QMF, DXT, and Data Dictionary. His most recent management assignment was again in information development, providing guidance to customers implementing the GPD storage strategy. Mr. Wolford is currently assigned to the STL technical staff with responsibility for SAA introductions.

Reprint Order No. G321-5326.