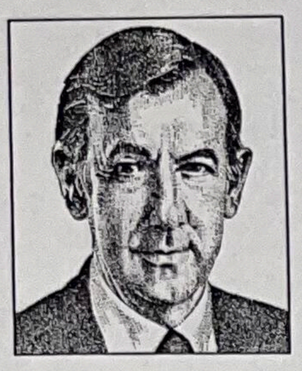
APPLIED INTELLIGENCE

Software's Future Lies in Cooperative-Processing Systems



JAMES MARTIN

This is the second in a series of articles on key industry trends affecting computer hardware, software, database environments and communications environments.

Trends in computer software are changing rapidly, posing challenges for many organizations. Some of the trends are the growth of end-user computing; the transition to graphical user interfaces; the introduction of powerful development tools, such as integrated computer-aided software engineering (I-CASE) tools; and the growing commitment to integrated computing environments, such as IBM's Systems Application Architecture (SAA) and Digital Equipment Corp.'s Network Application Support (NAS).

These architectures and tools will increasingly be used to build enterprisewide cooperative-processing systems that distribute data and processes across multiple hardware platforms (see chart). At the same time, the widespread adoption of software standards will provide consistent operation of graphical user interfaces, programming interfaces, communication interfaces and repository standards.

Software Strategies

By the end of the 1980s, many large organizations had adopted a strategy for software use within the firm. Off-the-shelf PC applications were commonly used by office workers and other end users, query languages and decision-support systems were used by business analysts, fourth-generation languages were used for the creation of departmental applications and some corporate systems, and COBOL was still pervasive as the de facto language of information systems. But the language infrastructure will crumble in the 1990s.

While end users will continue to require the functionality of off-the-shelf products, they will demand that the functionality of several products be integrated and user interfaces be standardized. Much of this integration will be achieved through the use of multitasking operating systems such as OS/2 and graphical user interfaces such as Windows 3.0 and Presentation Manager.

While both Windows 3.0 and Presentation Manager offer many of the same interface and multitasking features, the underlying operating systems differ radically.

The Windows 3.0 interface operates within the DOS environment. As such, applications are limited to single tasks and restricted to a base memory of 640K bytes. A restricted form of multitasking is available by switching banks of 640K bytes from extended memory to base memory.

However, this solution cannot be used as a platform for the development of large, complex applications. Already, the basic kernel software of I-CASE tools can barely operate within the base memory of 640K bytes. If the high-speed loop of these products exceeds 640K bytes, the bank-switching functions provided by DOS extenders and products such as Windows 3.0 will not provide a solution. Because of the fundamental limitations of DOS, high-end CASE tools are likely to migrate rapidly to OS/2.

As the hardware computing resources increase in functionality, DOS and Windows 3.0 will not be able to fully exploit the capabilities of the hardware; OS/2 and Presentation Manager, which support both multitasking and direct addressing of large memory applications, will.

The use of query languages and decision-support languages will enable users to more easily access and analyze large databases. Except for sophisticated mathematical analyses, much of the data access and manipulation will be performed using graphical and forms-based user interfaces.

It's not like the early 1980s, when a user had to understand both the physical data structure and a precise language syntax for performing a query; newer facilities allow the user to concentrate on the semantic meaning of the request. In addition, the use of artificial intelligence and natural language techniques will assist the user. These more powerful—yet easier to use—query and decision-support systems will become the basis for commonplace executive information systems.

Fourth-generation languages will continue to evolve in several significant ways. First, the languages will become more portable—moving to smaller computing platforms. They will also improve in power, performance and integrated capabilities.

COBOL will continue to retain its place in the IS organization, but will take on a new image. Rather than users coding applications by hand, COBOL applications will be generated automatically via CASE tools. Current CASE products let the user enter specifications using graphical metaphors and symbols. The specifications are checked for accuracy and completeness, then used as input to source-code generators. While these techniques are in their infancy today, they will become the standard means by which applications are developed and maintained in the future.

More important than the elimination of hand coding and the development of COBOL applications, CASE products will also assist in the maintenance of existing COBOL systems. Reverse-engineering and re-engineering of existing applications through the use of CASE products will continue to preserve software investments.

Perhaps the most significant trend in computer software will be the ongoing development in the field of CASE. CASE products have evolved from simple graphical editors with data dictionaries to robust front-end tools that enfranchise systems-development methodologies.

Several CASE products are now capable of supporting the entire systems-development life cycle, from business planning through testing and enhancement of the application. Users will be assisted with expert-system rules that ensure methodologies are being followed. Once the application has been defined and specified, the user will select an appropriate language version, and the application will be automatically generated.

CASE products that provide much of this functionality are already in the market. The products will continue to improve by adding facilities that mathematically or logically validate the systems design, incorporate AI and expert systems techniques, and generate multiple language versions from a singular set of specifications.

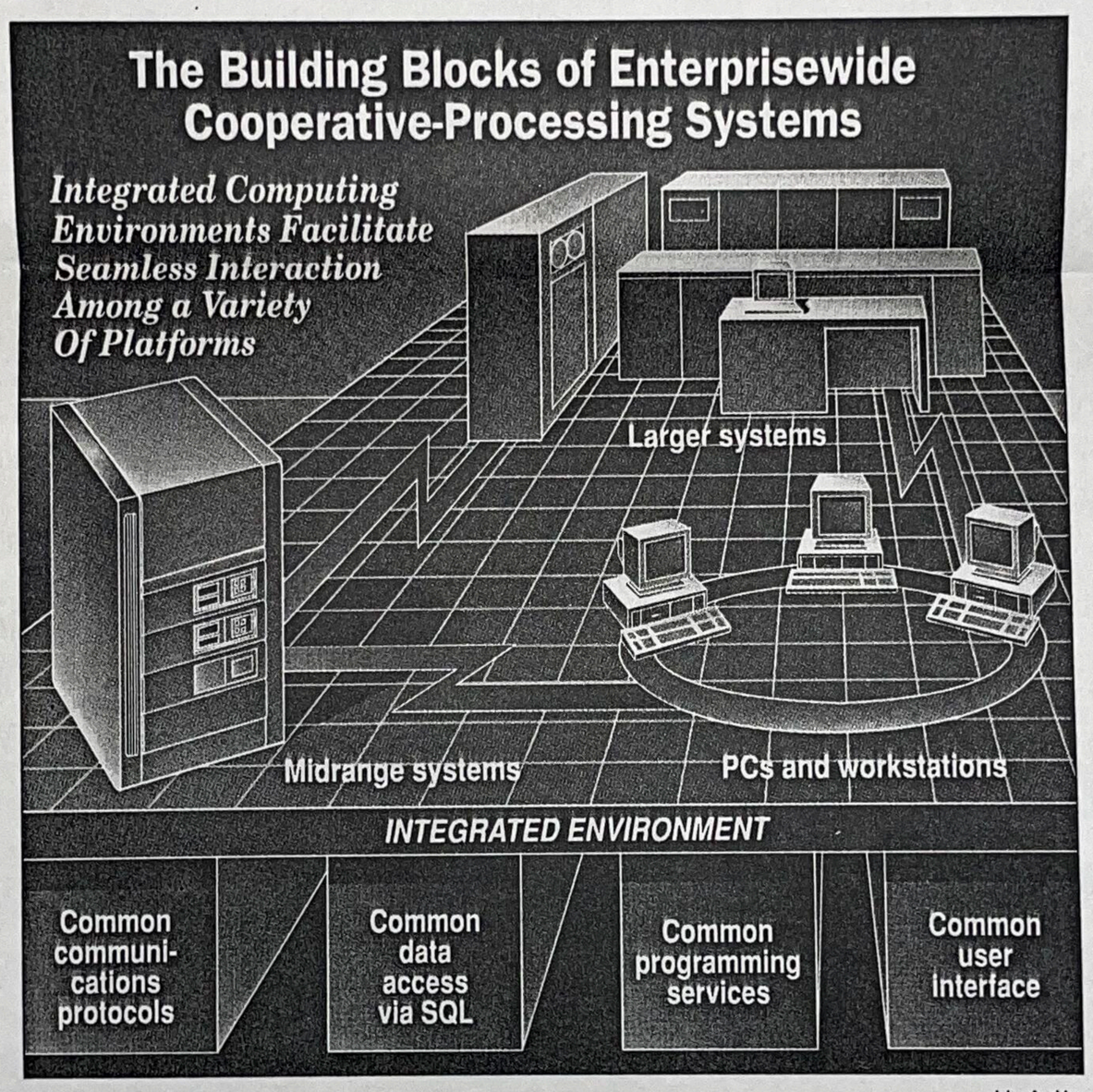
Client/Server Applications

CASE software products will also evolve to support the development of client/server-based applications. In a client/server application, the application, data and computing resources may be distributed throughout a network. Developing applications that are optimally tuned for network distribution is today a complex task, inhibiting the growth of the client/server architecture.

A current obstacle in the CASE market is the lack of standardization among vendors, particularly in the repository. It is not currently feasible to mix and match facilities from several vendors into an integrated CASE environment. The repository becomes the focal point for linking diagramming facilities, design analyzers, code generators, documentation generators and project-management aids.

Without a common repository for storing specifications, users are unable to select a preferred tool to support a specific function or phase of the systems-development life cycle. As a result, single-vendor I-CASE products currently offer the only solution for complete life-cycle coverage.

Next week's focus will be the major trends now occurring in database-access software. ■



John Avakian

Users will need the functionality of several off-the-shelf products to be integrated. Much of this integration will be achieved through multitasking operating systems.

The concepts embodied in this article are described in the High-Productivity Technology volume in The James Martin Report Series. For more information on this volume, call (617) 639-1958. For information on seminars, 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.