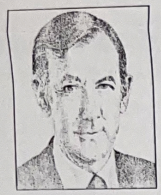
APPLIED INTELLIGENCE

SAA's Application Generator Will Boost Productivity



JAMES MARTIN

This is Part 7 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.

An important trend in the 1990s will be the rapid movement of application-development functions from the mainframe environment to PCs. The low cost of mips (million instructions per second), faster response time, ease of use and substantial processing power of the programmable workstation makes it an attractive environment for application development.

Integrated computing environments such as SAA will accelerate this trend by allowing programmers to develop enterprisewide applications on the PC that run on any supported SAA platform. Applications will incorporate seamless interfaces that enable information and processes to be distributed among all machines on the network.

In the future, SAA-compliant applications will be developed using a highly integrated application-development environment consisting of front-end CASE tools, a common repository of design information and a back-end SAA application generator. The integration of these facilities into a common development environment represents a major challenge for IBM and other vendors.

The industry's current application-development procedures are far from the ideal described above. Most applications are still developed by hand, using conventional third-generation languages such as COBOL or FORTRAN. Handcrafted applications require a great deal of time and effort, are very expensive and often don't meet users' needs. A major objective of SAA is to improve programmer productivity through the provision of automated application-generation tools. The primary component of SAA that will be used for the automatic generation of applications is the SAA application generator. The function of the application generator is to create code that is compliant with the user, programmer and communication interfaces defined by SAA.

Cross System Product (CSP), an IBM fourth-generation language, has been designated as the SAA application generator. CSP was originally designed to allow for consistent applications across different IBM platforms. Initial offerings of CSP were relatively weak compared with other vendors' products. It has since matured into a competitive application generator.

In CSP's current form, applications are developed on a System/370, but executable versions can be generated for the PC and the 8100 machine, as well as MVS, VM and DOS. An executable version of CSP is also being developed for the AS/400.

Although CSP is a good mainframe-based application generator, it is not yet SAA-compliant. Applications can be generated and run on the AS/400 and PS/2, but they must be maintained on the System/370. CSP's built-in screen painter is System/370-based, and is not even Common User Access (CUA) 3270 at that. Finally, communications are missing entirely. It is only by using awkward external products that some semblance of connectivity can be simulated.

When the application-generation facilities are put on the programmable workstation, the programmer will be able to develop applications that run on any supported SAA environments. Applications developed with the application generator will access relational data through the Distributed Data Manager (DDM), which provides transparent access to data anywhere in the network. Modules of the application generator will be distributed as well. The subsystem running the application will handle calls to modules and automatically locate the module in the network.

As shown in the figure, application-generator applications will be built in a sophisticated development environment. A mainframe-based repository will store all the information related to an application, including screens, code and data definitions. Computer-aided software engineering (CASE) tools will be integrated with the repository, the SAA application generator and the facilities of the Common Programming Interface-Communications (CPI-C).

Unfortunately, the evolution of CSP to support these high-level application-generation functions will take at least a year. However, the productivity gains of application generators are not to be ignored; the integration of CASE and application generators is a major trend of application development.

The dilemma for organizations that want to use SAA today is that there are two technology paths to pursue. One is compliance with CUA and the development of cooperative-processing applications. The other is utilization of application-generator technology with CSP. Due to the current limitations of SAA, it is difficult to pursue both.

Cooperative-processing applications may be built today with SAA-compliant facilities using lower-level languages such as C or COBOL, rather than CSP. These languages currently provide more direct access to SAA user interfaces and communication tools.

An alternative strategy more in line with the future of SAA is to start now with application-generation technology based on CSP. This approach will grow to include other aspects of SAA.

Although IBM has a lot of work to do to develop CSP into an SAA application generator, there are numerous advantages to using CSP today.

CSP offers a screen painter for designing panels that interact with the user. This is a powerful productivity enhancer. The screen painter does not enforce CUA standards today; however, developers can use it to design screens that have some level of CUA conformance.

The full, programmable-workstation CUA interface is still just a direction for CSP. The most reasonable CUA steps to take with CSP are to follow CUA guidelines as closely as possible within the CSP screen painter.

The database facilities of CSP are closer to the SAA ideal than the capabilities of the user interface. CSP provides excellent support for both DB2 and SQL/DS. It automatically generates Structured Query Language (SQL) from high-level CSP statements. The programmer can either accept the generated SQL or, if more sophisticated data access is required, modify it. Because DB2 and SQL/DS both now provide some degree of homogeneous data distribution, CSP also has access to distributed data.

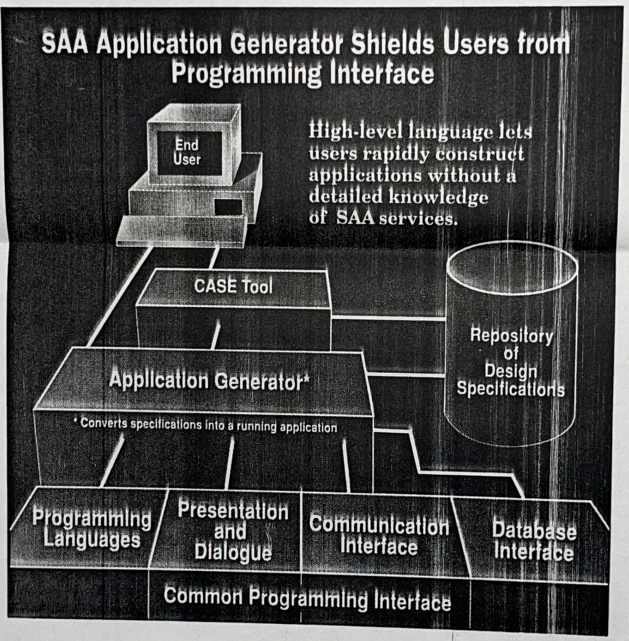
CSP and the Data Manager

On the programmable workstation, the relationship between CSP and the Data Manager is not as clean. It requires the user to access an external COBOL module, which performs the database access. This will probably not be addressed until the full CSP application-development environment is ported to the programmable workstation.

Finally, the tools for implementing cooperative processing will not be available in the near future. It still is not clear whether CSP will provide communications verbs, or if the subsystem running CSP on a given machine will be responsible for accessing different modules. CSP developers are hoping that the latter proves to be the case.

It is possible to do some cooperative processing today with CSP by using software that is available under both DOS and OS/2. You must, however, leave CSP to accomplish this. The tool to use is the Enhanced Connectivity Facility (ECF). It provides the ability to build a shell around an application that can then be made up of components on different machines. ECF provides the communication. This type of facility will continue to evolve under OS/2 Extended Edition.

Next week, the final article on SAA will describe the facilities in the SAA environment. ■



John Avakian

IBM has a lot to do to develop Cross System Product (CSP) into an application generator for its Systems Application Architecture, but there are advantages to using CSP today.

To learn more about the subject of these articles, please call The James Martin Report, an information service updated quarterly, at (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.