

APPLIED INTELLIGENCE

Previewing the Application Development Revolution



This is the last article in a series 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 1990s.

JAMES MARTIN

The focus of earlier articles in this series on SAA was the Common Programming Interface (CPI). The CPI provides programming services that are common across all supported SAA environments. These services include common programming languages, common user interfaces, common communication interfaces and common database-access facilities. Positioned on top of the CPI is the Application Development Environment (ADE). As the name implies, it's the environment in which developers will build SAA applications.

IBM hasn't announced the details of the ADE yet, but there have been fairly strong indications of what it will look like. The ADE will be an SAA cooperative application, with the programmable workstation providing the user interface and much of the power. As shown in the graph, the workstation will work in a client-server mode with the host-based repository, which will contain all of the definitions required to build an application.

A Consistent Environment

The workstation user interface will use the most advanced Common User Access (CUA) specifications. Developers will thus have a consistent environment in which to work.

The ADE will make it possible to generate applications for all SAA environments from the programmable workstation. For example, a programmer building a cooperative application with portions on the workstation and portions on an AS/400 would write and develop all of the code on the workstation. The workstation-based ADE is connected cooperatively to the AS/400, allowing the programmer to develop application components that can be sent to the AS/400 for execution.

In the case of third-generation languages, such as C, the developer might ship the source code to the AS/400 for compilation, or the ADE might provide cross-compilers that allow compilation on the programmable workstation. In either case, the programmer's window on the development process will be the workstation.

The application generator technology in the ADE will parallel that of the current Cross System Product (CSP), an IBM fourth-generation language on which the application generator is based. Currently, the source code for CSP is developed only in the S/370 mainframe environment, and from there, an executable version can be gen-

erated for other environments, including the AS/400. Since the SAA Application Development Environment will be the programmable workstation, CSP will have to be reimplemented to support the development and generation of code in the workstation environment.

There are no plans to put the user interface of the ADE on either the S/370 or the AS/400.

An important component of the ADE will be IBM's central repository. It will most likely be host-based and act in a server mode for multiple development workstations, each of which will have a local repository of design information. The central repository will hold all the

central repository and CASE tools.

The design specifications of a CSP application can now be stated in External Source Format (ESF), which is a tag language for defining the application. The ESF maps directly onto the CSP source files, which are used to generate executable versions of CSP applications. Now that the ESF is formalized and published, it's possible to build CASE tools that output ESF. At least two CASE vendors, Index Technology Corp. and KnowledgeWare Inc., have already taken steps in that direction.

The ESF, while currently in database files, can clearly be incorporated in the repository when available. The ESF can

The CASE tool shields the programmer from having to know the details of how to build CUA-compliant screens. The programmer only specifies logically what is desired from the user—that is, field specification on a data-input form, or the selection of an item from a list. An interface that conforms with CUA standards is automatically built from the logical definition by the Dialog Manager, one of the tools in the ADE.

Similarly, the complexities of database access can be hidden in the tools of the ADE. High-level specifications of database access can be made in a CASE tool. These specifications are translated into the primitive forms in the repository. When the executable form of the application is generated, the Structured Query Language needed to access the database is automatically built from the specifications in the repository.

The repository itself might not be confined to just a single host environment; it might be stored in a distributed relational database. The developer can use the local repository on the programmable workstation to store those portions of application definition that are being worked on. The distributed-database mechanisms of SAA will keep the databases synchronized.

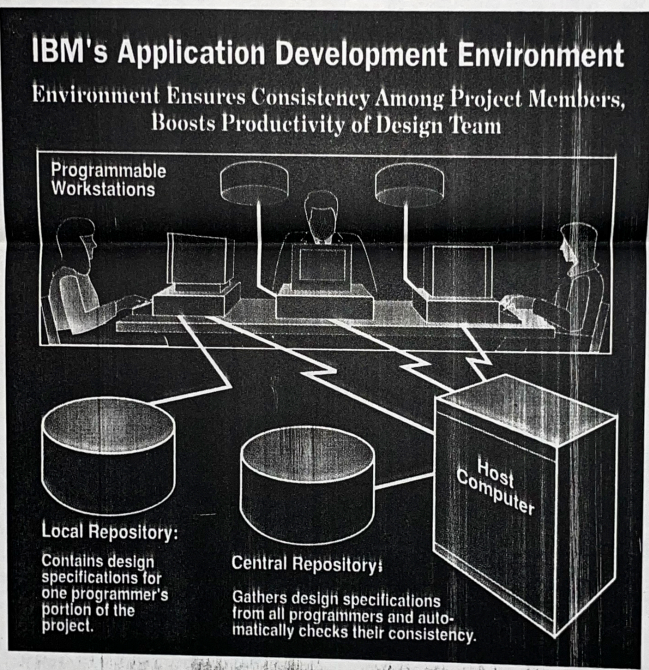
Better Productivity

Any portion of application development that is repeatable can be abstracted to higher levels and built into CASE tools. These higher levels of abstraction then generate the specifics for a particular application from relatively few parameters. The result is higher productivity for the developer.

Once design specifications for an application have been stored in the repository, they can be used to generate executable versions of the applications. Using the facilities of the ADE, code can be generated in any of the SAA languages, such as C or COBOL.

The application-development environment of the future will be much different from the one developers are used to now. The cooperative environment between a host-based repository and graphical user interface on a programmable workstation is a powerful combination. The generation of conventional third-generation languages from the application-development environment will allow existing optimizing compilers to make the applications as efficient as possible. The result will be a revolution in application-development technology.

Next week, I'll begin a series of articles on object-oriented programming, a technology that is changing the way programmers and users interact with computers. ■



John Avakian

The Application Development Environment will be an SAA cooperative application, with the programmable workstation providing the user interface and much of the power.

definitions of an application, such as data elements, screen definitions and source code. Teams of developers will share the repository.

Computer-aided software engineering (CASE) technology will also play a role in the ADE. Front-end CASE tools, supporting planning, analysis and design functions, will generate application definitions in the repository. Other tools will use the information in the repository to generate executable code, databases, documentation and project-management information.

Hints of these directions can be seen in the current CSP product, especially in the relationship between CSP, the cen-

then be used directly with the CSP development tools, CASE development tools and the code-generation portions of CSP. In fact, the developer can move from one of these tools to the next.

ADE's complexity is made manageable by restricting the types of applications it generates. The more restricted a class of application is, the easier it is to build tools that generate it. The CASE tools and application generator will be designed for the applications that deal with most normal business applications. These applications generally perform transactions against corporate databases. User input is usually in the form of menus or forms.

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.