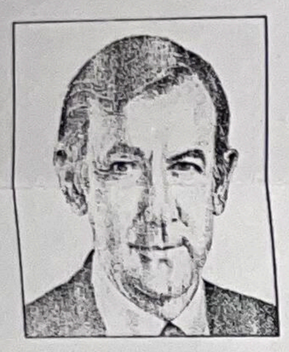
APPLIED INTELLIGENCE

Integrated CASE Tools a Must for High-Speed Development



JAMES MARTIN

This is the fifth in a series on rapid application development (RAD), a development methodology designed to yield much faster results than traditional methods.

Rapid application development, as I've said before, is a new life-cycle process that harnesses the major increase in productivity that can be achieved by small teams of highly motivated developers using integrated computer-aided software engineering (CASE) tools. Although the success of RAD is primarily due to innovative management techniques (to be discussed in future columns), advances in applications-development technology also play a major role.

An important component of the RAD life-cycle process is the use of integrated CASE (I-CASE) tools to develop entire applications from design specifications. Small teams of analysts are intensively trained in the use of I-CASE tools and in a methodology optimized to achieve high-speed development.

As shown in the figure, I-CASE products incorporate separate workbench components for each aspect of the applications-development process, including strategic planning, analysis, design and code generation. Additional components support documentation generation, database generation and project management.

The front-end planning, analysis and design components of these products are closely integrated with the back-end code, database and documentation-generation facilities through the use of a common repository of design specifications.

Specifications developed by individual designers are stored in a personal repository on the PC. Specifications from multiple project analysts on a LAN are consolidated in a project-level repository accessed via a file server on the LAN. Design specifications that are common across the organization are stored in a corporate-level repository on a central computer.

In an increasing number of corporations, all the components of I-CASE products, including the code generator, are being run on desktop workstations. These tools provide each workstation user with customization, powerful graphics, project-level coordination and intelligent operation.

The most advanced CASE tools used for RAD incorporate features such as graphical design aids, automated design analyzers, expert systems and integrated, desktop code generators. These tools are capable of generating highly efficient code that can be used without performance penalty in a heavily loaded transaction-processing environment.

Some information-systems organizations that have been using third-generation languages such as COBOL are choosing to leapfrog current fourth-generation language technology and move directly to the next generation of tools—integrated CASE workstations closely coupled to efficient, desktop code generators.

CASE tools must have certain characteristics to be effective in RAD life cycles, but many of the CASE tool sets on the market lack these characteristics.

At first glance, the tools appear spectacular with their flashy graphics, but they fail to incorporate some of the most important features required for RAD. Among these are support for the entire life-cycle process, an integrated code generator capable of generating 100 percent of the code for an application and support for a life-cycle process that is optimized for high-speed development. The manager of systems development must select tools that can support RAD life cycles; otherwise, development efficiency will be low.

A principle of CASE is that, whenever possible, diagrams are used as an aid to clear thinking. The systems analyst uses the front-end workstation components of an integrated CASE tool to define design specifications by means of diagrams. These diagrams, generated by interacting with the CASE tool, represent planning information, an overview of systems, data models and data flows, detailed designs and program structures.

The meaning represented by diagrams and their detail windows is stored in a repository—the heart of an I-CASE system. The repository steadily accumulates information relating to the planning, analysis, design, construction and, later, maintenance of the systems.

An integral part of repository-based systems is a design analyzer, which ensures consistency among the different pieces of knowledge that reside in the repository.

When a person using a workstation enters new information into that workstation, the design analyzer checks whether it obeys the rules and is consistent with what is already in the repository.

Normally, an individual or design team is not familiar with the entire set of designs in the project-level repository or the corporate repository. When an individual starts to create a design, he or she will extract the information in the project-level repository that relates to that design.

The individual may, for example, extract a portion of a data model. He may be designing the detail of a process that is already shown in a higher-level representation. He then works on his design, largely independently of the project-level repository.

When the design is ready for review, it can be consolidated with the knowledge already in the project-level repository. When the design is coordinated and approved, it will reside in the project-level repository, where it is available to other designers in the team.

The simplest CASE tools are little more than diagramming aids—word processors for diagrams. They simplify the drawing of diagrams and enable them to be modified quickly and kept up to date. Most CASE tools incorporate a design analyzer, which detects logical errors and inconsistencies at the design stage.

More sophisticated CASE tools are required to support RAD. The tools should provide integrated support for each of the four stages of development: planning, analysis, design and construction.

An I-CASE environment provides an integrated set of tools for all phases of the life-cycle process. The term "I-CASE" should be used only to describe products with this level of integration.

In an I-CASE tool, the front-end workbenches and the code generator use a common repository to generate program code, database code and documentation for the entire application. The code that is generated should never be touched by maintenance programmers; all maintenance is performed at the specification level.

The integrated CASE tool should do the following: facilitate the rapid building and modification of prototypes; generate system documentation; generate test data and provide testing tools; and generate database code and job control code so that the program can be quickly executed when design changes are made.

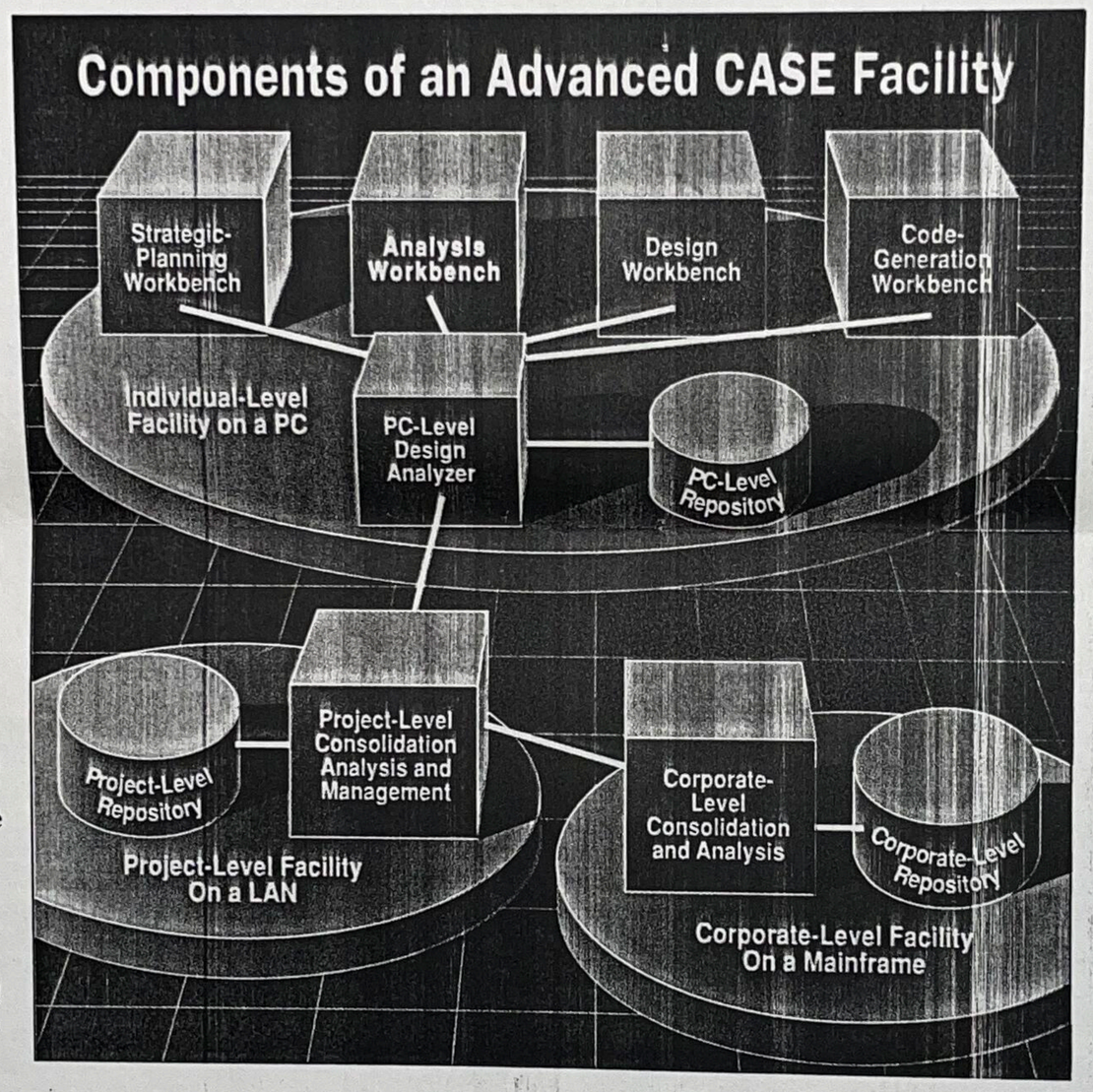
Prototyping Is Essential

Prototyping is an important part of the RAD life cycle. It is essential to show the users what they're going to get, allowing them to react to it. The RAD tool set must include the prototyping capability to create screens and dialogues quickly and to modify them while interacting with users.

The ability to generate code, test it, modify it quickly and regenerate it makes RAD prototyping different from conventional prototyping. With RAD, the evolving prototype is not discarded but is part of the final system.

This is an essential element of RAD. The prototype should become the final system. It should deliver the machine performance, reliability and other features required by the final system.

Next week's column discusses the need for a new development methodology and shows why RAD techniques provide far more productivity than the conventional development process. ■



John Avakian

The most advanced CASE tools used for RAD can generate efficient code that can be used without performance penalty in a heavily loaded transaction-processing environment.

The concepts embodied in RAD are described in a new volume in the James Martin Report Series. For more information on this volume, call (800) 242-1240. 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.