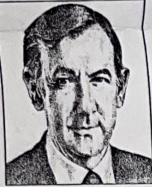


## APPLIED INTELLIGENCE

## The Company and the CASE Tool: Making the Match



**JAMES  
MARTIN**

In evaluating computer-aided software engineering (CASE) tools, it's important to divide them into categories that match the functional needs of an organization. Useful categories include the following:

- integrated tools capable of

generating 100 percent of the code for an application from design specifications;

- non-integrated tools that generate fragments of an application but require some hand-generated code outside of the CASE tool;
- tools that support an information engineering methodology;
- tools that support a structured engineering methodology;
- tools that support C code generation for engineering workstations in a Unix environment; and
- tools that support real-time applications.

The figure shows a series of questions that companies can ask to help them determine how to categorize the tools. The top-level question is: Does the tool generate code for 100 percent of an application?

Integrated CASE tools fitting that category are listed on the left side of the diagram. Non-integrated tools that generate fragments of an application are listed on the right.

There are only a small number of integrated tools that provide tight integration between the front end and the back end. Tightly integrated tools are recommended to support rapid

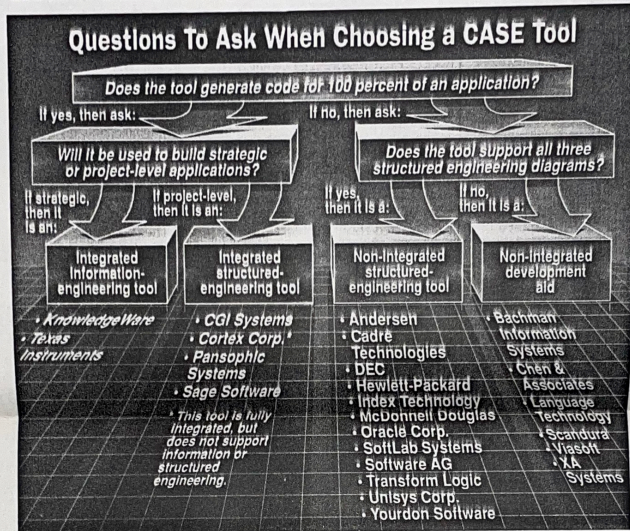
prototyping methodologies.

Integrated CASE products on the left side of the diagram are further subdivided into tools that support strategic applications and those that are oriented primarily toward project-level applications. Strategic applications that may be used across divisions of an organization

prise modeling and business-area analyses.

The choice between a methodology based on information engineering or structured engineering may be dictated by the corporate culture for applications development.

Structured engineering is a relatively



John Avakian

are best supported by an information engineering methodology that works top-down from enterprise models and business-area analyses to the design of individual projects.

Structured engineering is widely used for the development of project-level applications. It is generally not concerned with higher-level issues such as enter-

simple methodology that's widely used by many organizations. Information engineering is a more complex methodology that isn't as widely used. However, information engineering is the recommended methodology if strategic applications are to be built that will progressively computerize all functions of the organization.

In spite of the high return on investment provided by integrated CASE tools, there are many reasons why organizations should consider non-integrated tools.

Current integrated tools do not support C code generation, the Unix environment, real-time applications, traceability and other important functions. Several of the non-integrated tools do support these functions.

Tools on the right side of the diagram may be further subdivided by asking the question: Does the tool support all three diagram types of structured engineering, including data-flow diagrams, decomposition diagrams and entity-relationship diagrams?

Non-integrated tools that do not support the three basic diagram types are typically used as niche products for a portion of the development life cycle—for example, data modeling, source-code scanning and source-code restructuring.

The figure is oriented toward the selection of CASE products to support business applications in an IBM target environment. Requirements to support other environments, such as Unix workstations, generation of C code and support of real-time applications, would substantially change the categorization of CASE products.

Next week, I will discuss the future evolution of CASE products. ■

The concepts embodied in this article are described in the CASE 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.

## Migration \ More Firms Traveling the OS/2 Development Path

Continued from Page 57

real-time Unix operating system for his assembly-line automation system, he vetoed it in favor of OS/2 to avoid being tied to one vendor. "We wanted to use a general-purpose operating system like OS/2," he said.

Another enticement to travel down the OS/2 development path, according to respondents, is that operating system's added horsepower—particularly its multitasking capabilities and built-in communications and database support.

For Paul Morales, these bonuses translated into a powerful environment in which to create a front end to a mainframe-based insurance policy system.

"We needed to do several types of host sessions and have them up and running at the same time," said Morales, a senior project engineer at Equitable Financial Cos., a life-insurance company in New York.

While OS/2 furnishes communications capabilities for building cooperative-processing applications, its real strength lies in networking, according to some of the respondents.

"Where OS/2 really shines is in LAN support, because we can easily connect OS/2 machines to our Token-Ring network," said Greg Barnes, a programmer/analyst for Farmland Inc., a farm cooperative in Kansas City, Mo. This makes it easier to distribute commodities and market information to 400 PCs on a local area network, he said.

OS/2's communications capabilities also provide developers with an entry into the mainframe world.

"We have an AS/400 [minicomputer] and IBM 3000 mainframes, and providing connectivity to those platforms is the leading reason to go to OS/2," said Gregory Brunet, a systems analyst for Waste Management Inc., a waste treatment center in Oak Brook, Ill.

While many respondents found OS/2 development more time-consuming than expected, they said the operating system's palette of programming tools and built-in debugging facilities are far superior to those of DOS.

"OS/2 helped us to make a better-quality program," said Andrew Fischer, a programmer for Vector Software Inc., a software developer in New York. "We found bugs under OS/2 that we would never have found under DOS."

OS/2 is clearly meeting the needs of some developers, yet a significant number are still hesitant about moving over. The biggest hurdles, respondents noted, are the initial investment in hardware upgrades and incompatibilities with the still-dominant DOS world.

"We were developing in OS/2 but put it aside because of incompatibility with what we'd developed before," said Alon Miller, a partner in Alon Miller Associates, a consulting firm in Staten Island, N.Y.

While users expect these issues to be resolved in the next few years, several are eyeing Microsoft Corp.'s Windows as a short-term alternative because of its graphical user interface and limited multitasking abilities.

"We're going to Windows whole-hog in the short term," said Steve Silver, a vice president at Chase Manhattan Bank in New York. "There's less resistance to putting a GUI over DOS."

Other respondents said they prefer to wait for OS/2 version 2.0, the full 32-bit implementation of the operating system that's due later this year.

"I look to OS/2 2.0 to be the first OS/2 that users will want, because they can get to it easily," said Rick Stuckey, a partner in Andersen Consulting, the Chicago-based consulting arm of accounting firm Arthur Andersen & Co.

"Hardware won't be as expensive," he said, "and faster processors will bring down the cost of OS/2 machines." ■