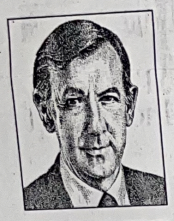
APPLIED INTELLIGENCE

I-CASE Tools Strongly Affect the Development Cycle



JAMES MARTIN

In this, the fifth of six columns on I-CASE technology, we look at the functional characteristics you should look for in these tools.

Perhaps the best way to discuss the life cycle of an I-CASE tool is to resort to an illustration. As you see, we've chosen a pyramid.

The life-cycle process, as the pyramid shows, consists of four phases: strategy planning, analysis, design and construction. You'll notice that the figure depicts a fully integrated development methodology, supporting all phases of the life-cycle process.

At the top is strategic planning, which encompasses the business strategic-planning functions that are the foundation for subsequent phases of the development process. The next level is analysis, wherein a model is built of the fundamental data and the processes needed to run the particular enterprise. From this analysis the system's need is determined. The third level is design with CASE tools. Designs are expressed in rigorous, complete detail. The bottom level is construction, where systems are actually built.

On the left side of the pyramid is data; on the right are activities (or processes). Both data and activities are managed from a high-level, management-oriented view at the top through to implementation at the bottom.

At the strategic level on the data side, an overview of the information needed to run the enterprise is created with an entity-relationship diagram. At the data-analysis level, this is extended into a normalized data model. At the strategic level on the right-hand (process) side, an overview model of the enterprise is created, and planning tools are used to relate its goals, problems, critical success factors and so on to the functions in the enterprise. At the analysis level, a model of the processes is created and linked to the data model.

At the design level, specifications for procedures are created and filled out in sufficient detail to drive a code generator.

Tools required at this level include a screen painter, report generator, data-structure diagrammer and tool for showing program structures. At the construction level, program code, database code and job-control code should be generated.

Categories of CASE Tools

It's best to have CASE tools for each of the four stages of information-systems (IS) development: planning, analysis, design and construction.

Some vendors sell separate workbenches for each of these sets of activities. Such workbenches should be fully integrated and employ a common encyclopedia.

Specification of applications should evolve from the planning phase to construction, with the knowledge acquired in one phase being used in the next phase. There also should be a seamless interface between the phases.

A wide variety of CASE tools are available. Keep in mind the following:

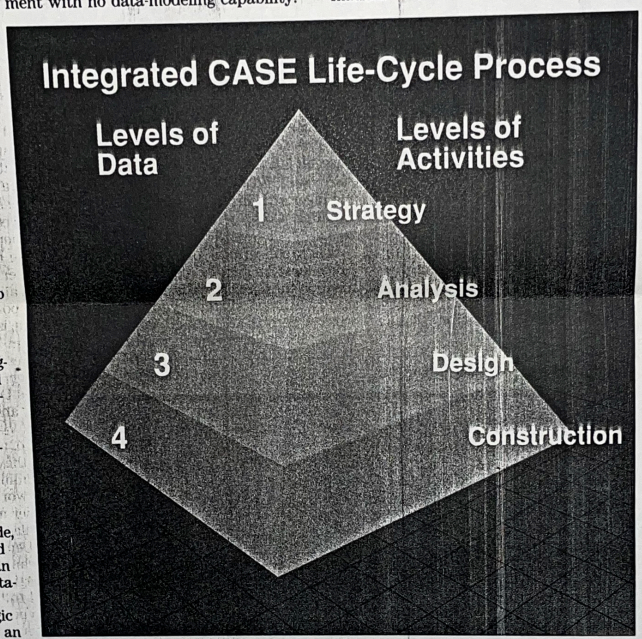
- Some CASE tools support only analysis and design components and do not contain business-planning or strategic-planning components.
- Some are code generators with separate planning, analysis or design tools.
- Some analysis and design tool kits have a process-oriented view of development with no data-modeling capability.
- Some provide data-modeling tools without process analysis or design.

Most CASE tools do not incorporate a tightly integrated code-generator component. Some vendors have specialized in building code generators without front-end analysis and design tools. Other vendors have built front-end planning, analysis and design tools without an integrated back-end code generator. Many attempts have been made to couple front-end analysis and design tools to back-end code generators.

Building a loosely coupled bridge between the front and back end of the tool is a solution that's not fully satisfactory, because much manual work is still required to complete the code-generator function. What is needed is full integration between the CASE front-end tools and the generator, so that code is automatically generated from the front-end tools.

Integration Is Key

The term I-CASE should be used only to describe products that are fully integrated and that support all parts of the pyramid. This implies that the front-end tools and the back-end code generator use the same encyclopedia. A common encyclopedia should be used to generate program code, database code, documentation and project-management information.



David Hamum

CASE tools are used for the four stages of IS development. Specification of applications should evolve with the knowledge moving from planning through to construction.

The desirable characteristics of CASE tools and I-CASE tools are listed below. Remember: A tool should not be referred to as CASE if it does not perform the following basic functions:

- Enable the user to draw diagrams for planning, analysis or design phases on a workstation screen.
- Solicit information about the objects in the diagram and relationships among the objects so that a complete set of information is collected.
- Store the meaning of the diagram, rather than the diagram itself, in a repository.
- Check the diagram for accuracy, integrity and completeness. The diagram types used should be chosen to facilitate this.
- Enable the user to employ multiple types of diagrams representing different facets of an analysis or design.
- Enable the user to draw program procedures with diagrams, showing conditions, loops, case structures and other constructs of structured programming.
- Enforce structured modeling and design of a type that enables accuracy and consistency checks to be as complete as possible.
- Coordinate the information on multiple diagrams, checking that they are consistent, have accuracy and integrity and are complete.
- Store the information built up at workstations in a central repository shared by all analysts and designers.
- Coordinate the information in the central repository, ensuring consistency among the work of all analysts and designers.

Characteristics of I-CASE Tools

CASE tools provide a much higher level of component integration than basic CASE products. The activities of planning, analysis, design and construction are supported by separate and distinct workbench tools. These workbenches are fully integrated so that one workbench directly employs the information from another.

A code generator is fully integrated with the design workbench (as opposed to having a bridge to a separate code generator with its own separate syntax). The code generator utilizes the same encyclopedia as the front-end planning, analysis and design components.

The code generator employs the facilities of requisite operating systems and database-management systems, including the data dictionary. It automatically generates the required database statements and job-control language. The output of the generator may be fed into an optimizer, which adjusts the code and database accesses to give optimal machine performance. Thorough documentation is generated automatically.

The components of an I-CASE tool support all phases of the project life cycle in an integrated manner. In addition, an I-CASE tool kit provides enterprisewide planning, data modeling and process modeling to create a framework that fits many different types of project life cycles. These powerful facilities permit I-CASE tool kits to be utilized for corporatewide information-engineering applications, as opposed to projectwide software-engineering applications.

Next week, we'll discuss the improvement in productivity that can be realized with the use of CASE tools. ■

The James Martin Productivity Series, an information service updated quarterly, is available through High Productivity Software Inc., of Marblehead, Mass. 1-(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.