

source code diverges from the graphical specifications stored in the CASE tool, the investment in using the CASE tool to produce design specifications is lost. This may cause the CASE tool to contribute negative productivity to the development and maintenance process.

Bridge to External Generator

In order to provide a greater level of automatic code generation capability, many CASE tools incorporate a bridge to an external code generator (see Figure 2). For example, Exceleator has a bridge to Telon, which is used to generate procedural logic. Similarly, Teamwork has a bridge to INGRES, used to generate database schemas.

The primary difficulty of the bridge approach is that it creates two repositories of design information — the design information contained in the repository of the CASE tool and the procedural code or database schemas generated by the external tool.

Rather than being stored in a single repository, the design specifications for

the application are stored in two potentially incompatible repositories.

Generally, there is no automated technique to ensure the logical consistency of the two repositories of design information. It is the responsibility of project managers to maintain the consistency of the two repositories through manual techniques.

Components of Integrated Tools

To overcome the limitations of non-integrated CASE tools, many organizations have turned to integrated tools. As illustrated in Figure 3 (page 20), the components of integrated CASE tools include the following:

- Planning workbench
- Analysis workbench
- Design workbench
- Tightly integrated Construction workbench
- PC-level repository
- Project-level repository
- Corporate-level repository

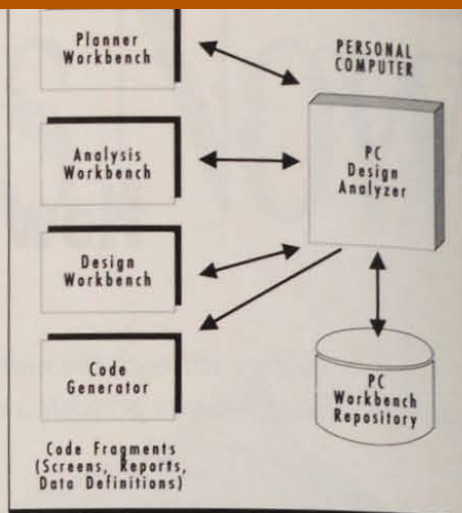


Figure 1. Components of Non-Integrated Tools

The planning workbench is used to support strategic planning, enterprise modeling and high-level data modeling. The analysis workbench is used to support business area modeling and detailed data modeling. The development of prototype applications is supported by the design workbench. Prototype development includes the specification of screens,

CASE Weight Test

A hands on approach to comparing and selecting CASE tools at Twentieth Century Services.

R.G. Eaton and Karl W. Hayes are spearheading the implementation of Information Engineering (IE) at Twentieth Century Services, Inc. in Kansas City, MO. Twentieth Century had already decided on the need for automating an IE approach, so their task was straight forward: form a research group with the goal of comparing CASE vendors and recommending one toolset.

According to Eaton, "Application developers with varying experience and expertise composed the evaluation team. For many of them it was their initial exposure to CASE technology. We needed to provide the team with a common understanding of IE and the role of CASE. We also wanted to ensure objec-

tivity and consistency between team members."

Researching the trade press and vendor supplied information, they soon realized that comparing two CASE tools



R.G. Eaton (left) and Karl W. Hayes are spearheading the implementation of Information Engineering at Twentieth Century Services, Inc. (Kansas City, MO)

would be "like comparing a cattle cutting horse to a Clydesdale" says Hayes, as "each was a breed for a specific job." Before evaluating the tools, they determined their specific requirements for the tool. Once these requirements were established, they compared the tools with respect to meeting their specific requirements. So, instead of comparing the tools directly against each other, they compared them against their version of an ideal CASE tool, which was identified by the task force.

The requirements were grouped into logical criteria categories, and then summarized in matrix format (see Figure 2). Each criteria was carefully reviewed and refined. Criteria sub-categories sprang from the refinement process. Criteria categories and sub-categories were then calibrated by defining the grading qualifications. Evaluation criteria covered a variety of categories — including each

reports, dialogues, procedural logic and database access. Prototype applications are demonstrated to the end user in order to verify that the application meets the business needs of the end user.

Automatic generation of consistent source code, physical databases and documentation is supported by the construction workbench. In an I-CASE tool, the construction workbench incorporates a tightly integrated code generator capable of generating complete applications in the target environment of the CASE tool. The target environment for most existing integrated CASE tools are business systems running on IBM mainframes or LANs.

Until fairly recently, integrated CASE tools such as IEW and ADW from KnowledgeWare, IEF from Texas Instruments, PACBASE from CGI Systems, APS from Intersolv and TELON from Pansophic Systems required a mainframe to convert design specifications automatically into source code, database and documentation. The mainframe code

and database generation modules for these products typically cost between \$200,000 and \$300,000, severely limiting their applicability.

Within the past two years, integrated CASE tools have evolved very rapidly, particularly in the area of desktop code generation. As shown in Figure 3, major integrated CASE tools have moved the code generation components from the mainframe to the PC.

Integrated CASE tools are now available that generate code for entire applications on the desktop.

CASE Tools and LANs

Another major change in integrated CASE tools is consolidation of design specifications on a local area network. Previously, it was necessary to consolidate specifications from multiple ana-

lysts on a mainframe. The major integrated CASE vendors are moving rapidly to support consolidation of project-level specifications within a LAN using a high-speed file server. Specifications from multiple projects are consolidated in a corporate-level repository.

One of the biggest challenges for CASE vendors in the 1990s will be the automated generation of cooperative pro-

(continued on the following page)

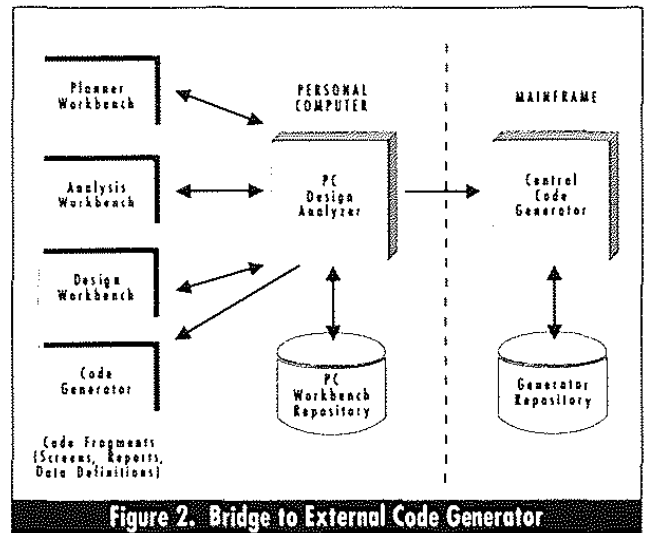


Figure 2. Bridge to External Code Generator

METHODS W/F 5		
TOOLS W/F 3		
ENCYCLOPEDIA W/F 3		
GENERAL W/F 2		
COST W/F 2	REFERENCES W/F 3	VENDOR STABILITY W/F 1
<input type="checkbox"/> 350,000 or less <input type="checkbox"/> 350,001 to 500,000 <input type="checkbox"/> Greater than 500,000	<input type="checkbox"/> Exceeds value expectations <input type="checkbox"/> Meets value expectations <input type="checkbox"/> Below value expectations	<input type="checkbox"/> 10 years in software ind. <input type="checkbox"/> 5 to 10 years in software industry <input type="checkbox"/> < 5 years in software industry

Figure 1. Rating Forms Containing Evaluation Criteria

CATEGORY	WEIGHTING FACTOR	RATING VALUE	NET RATING
GENERAL			
Cost	2	xxx	xxx
References	3	xxx	xxx
Vendor Stability	1	xxx	xxx
GENERAL RATING	2		xxx

Figure 2. An Evaluation Matrix

evaluator's gut feel. The matrix then defined the ideal and least ideal CASE tool.

Evaluation

The evaluation included interviewing users of each tool, an in-house demonstration and company research. The demonstration involved each vendor generating a very small application from their specifications. While evaluating the two products, each team member recorded their ratings on a uniform rating form (see Figure 1). Their ratings were loaded

into the matrix. At the conclusion of the evaluation, the matrix provided an objective and consistent comparison. According to Eaton, "Most importantly we had a quantified valuation to support their recommendation. Using this approach we could provide management with a numerical advantage of one product over the other, e.g., tool A was 17% better (in satisfying our requirements) than tool B."

Says Eaton, "Using this approach can provide objective recommendations that

can then become routine. All evaluation criteria needs to be weighted to your specifications. Most importantly, it allows you to weight the gut feel appropriately - just remember to stay true to your model (once the weighting factors are established). If you find yourself facing the CASE issue head-on, consider the benefits of a quantified evaluation." *