

## CASE HISTORY

---

### THE QUEST FOR AN INTEGRATED TOOL SET

---

At a major research and development laboratory in Albuquerque, New Mexico, the quest for CASE began four years ago. Today, an eight-member pilot team is about six months away from completing its first project using Texas Instruments' Information Engineering Facility. *CASE Strategies* spoke with Joseph R. Schofield, Jr., computer specialist at the laboratory, whose responsibilities include the application of emerging technology to the software development

process, about the tool selection process and the status of the current project.

Three features of the R&D company's experience struck us as especially noteworthy. First, the organization's CASE requirements were particularly stringent, due to the naiveté of the people who developed them and the overblown claims of a fair number of vendors. Second, despite the reputed steep learning curve of the tool they selected, the pilot team will almost certainly achieve at least a

two-to-one productivity gain on its first project. Third, it appears that the company's training costs will be lower than conventional wisdom would predict.

The department in which Mr. Schofield works develops applications to support corporate administrative functions — core systems such as financial reporting, asset management, procurement, and human resources systems. In 1987 the laboratory began migrating to an IBM mainframe environment from another vendor's mainframe. At about the same time, Mr. Schofield, just back from his first CASE symposium, obtained management's authorization to continue monitoring the technology, with no particular target date for adopting it.

The development organization's strategy at the time was to take advantage of off-the-shelf IBM applications packages, and possibly a fourth generation language, in an effort to reduce the need for new development. However, before long it became apparent that some business systems customers were unwilling to change their business practices to accommodate constraints built into the packages. "It appeared as if there was a ripe opportunity to use a technology that would deliver customized systems," Mr. Schofield said. Early in 1989, management gave him permission to begin the process of acquiring CASE tools.

Mr. Schofield and four others drew up a list of tool requirements that included mainframe orientation, a single encyclopedia with which all tools could communicate directly, no need for interfaces or import-export facilities, a tool-enforced methodology that spanned the entire life cycle, a consistent human interface, and "the ability to generate 100% of the [COBOL] code, 100% of the time." Other desiderata included re-engineering capabilities, and interfaces to particular tools — mainly IBM's Repository Manager/MVS (a statement of intent would suffice) and the Bachman Information Systems tool set.

Based on informal conversations with a number of CASE vendors, Mr. Schofield said, "I thought there were at least a handful that could do what we wanted." He called those he thought could qualify, outlined what the company was seeking, and asked, "Do you have a product that fits that description?" Eight vendors said they did, and were invited to make presentations at the laboratory's facility. Those who came to Albuquerque were Andersen Consulting, CGI Systems, IBM, KnowledgeWare, On-Line Software, Pansophic, Sage Software, and Texas Instruments. Index Technology was omitted from the list because it had no plans for a mainframe tool.

Clearly, knowing the company's requirements, several companies on the list could have saved themselves the time and expense of a trip to Albuquerque; among those invited were a database management systems company (On-Line), two programming utilities vendors (Pansophic and Sage), and a back-end code generation company (CGI). (On-Line has since added CASE capability, CGI has added a front-end tool, and Sage has merged with Index, acquiring the front-end Excelerator CASE tool.)

"We suspected that four [of the vendors] could do [what we were looking for], but they were always fuzzy when we asked them questions," Mr. Schofield said. In writing the request for quotations (RFQ) the team of five took pains to define terms very precisely, hoping to counteract vendor obfuscation. For example, different vendors mean very different things when they talk about code generation. "Every vendor out there will tell you that they do code generation," Mr. Schofield said. But some in fact generate only COBOL copy libraries containing file descriptions and record layouts. Others build skeleton COBOL programs containing the data division and a rudimentary procedure division. The power of such code generators depends on the functionality of the associated encyclopedia, the diligence of developers in defining processes in the design tool, and the level of integration among tool, encyclopedia, and

generator. If a product generates "most of the code," Mr. Schofield noted, "then you spend the rest of your time figuring out what code it left out."

To Mr. Schofield, complete code generation means not only all of the application code, but also all of the data definition language, data manipulation language, job control language, and CICS as well. The RFQ defined code generation to that level of specificity.

Despite his best efforts, some replies to the RFQ specified what code components were generated and simply ignored the rest, leaving it to Mr. Schofield and his team to read between the lines. The omissions, Mr. Schofield said, seemed "intentional. There's no other way to [interpret] that." Vendors, he added, can be "pretty evasive. They'll respond with what they can do, and withhold what they can't do."

Four vendors replied to the RFQ, but the real choice came down to Texas Instruments' Information Engineering Facility (IEF) and KnowledgeWare's Information Engineering Workbench (IEW). Both claimed complete integration among tools, encyclopedia, and code generator, as well as methodology support for the full systems development life cycle.

The competitive procurement process led to the selection in May 1990 of Texas Instruments' IEF. There were two dominant reasons for the choice, Mr. Schofield said. First was the perception that the IEW's connection with Gamma, KnowledgeWare's code generator, created after the two tools had been developed by two different companies, implied the risk that metadata might be lost in transition between the two products' encyclopedias. Second, TI had already delivered on its 1987 promise of a consistent user interface on the workstation and mainframe components of the IEF, which made the rest of its statement of direction all the more credible. Mr. Schofield contrasted the two companies' approaches this way: The message from TI, he said, was "This is our vision. We aren't there yet, but this is how all of

our products will plug into our encyclopedia." KnowledgeWare's approach, according to Mr. Schofield, was, " 'OK, now that we've got the products let's try to get them to work together.' One," he said, "was worth waiting for, the other may never really materialize."

In addition, while both products support the information engineering methodology, KnowledgeWare's support consists of a strategic relationship with the consulting firm Ernst and Young, who supplies the methodology in book form. The IEW does not require users to follow the methodology from start to finish, while the IEF does impose that discipline. The R&D organization had been using a paper-based methodology for about ten years, "but in a CASE environment, we wanted rules that are embedded and enforced in the tool itself," Mr. Schofield said.

To say that eight members of the company's 100-person administrative systems development organization were chosen for the initial IEF pilot overstates the situation. Mr. Schofield said that the pilot team refers to itself as "the few, the proud, the available. While we're grateful that we have the caliber of people that we have, we also realize that they were the people who were available. There were other good reasons for having them on the team, but had they not been available they would not be on the team," he said. A better approach, he added, would be to choose team members and plan to make them available at the proper time.

Consistent allocation of human resources has turned out to be critical to the introduction of CASE. Although reluctant to do so, management frequently finds it necessary to assign a CASE team member temporarily to some other project. Mr. Schofield said the impact not only disrupts the project schedule, but is also "devastating to the team." People engaged in learning and applying the new skills that CASE technology demands should not be diverted, he said. Also, team members have assumed key roles — one took charge of the project encyclopedia, another is in charge of

# CASE STRATEGIES

## CASE HISTORY

the mainframe portion of the IEF, and still another is project leader. Everyone's efficiency suffers when one of these people is removed from the project.

The pilot team's assignment is to replace a just-in-time ordering application that required about 22 work-years to develop. Mr. Schofield thinks the original project may have taken longer than it should have because it was staffed with new employees, people unfamiliar with the application, and contractors who were unfamiliar with the corporation and the project as well. By contrast, one of the pilot team's members is familiar with the just-in-time application; none is newly hired and there are no outside contractors involved.

By October, when the project is slated for delivery to the customer, Mr. Schofield expects the team will have invested seven or eight work years — less than half of the original development effort.

Training in information engineering and the IEF tool will have played a role in the speed with which the team has achieved productivity. To become "even reasonably good at building models," Mr. Schofield said, "I think the average developer in the average organization needs something like two solid, intensive weeks of training in the methodology, followed by several months of supervised practice, including periodic follow-up consulting."

The industry rule of thumb for CASE tool adoption is that companies should be prepared to spend

\$2 to \$3 on training for every dollar spent on the tool itself. Mr. Schofield said IEF training at the company may some day reach the lower of the two figures, but so far, the company's direct outlay has been less than \$5,000 per person, not even one-quarter of the tool's per-seat cost.

Paralleling the CASE implementation effort is a function point measurement process that has been underway for the past two years. Citing the book, *Software Metrics*, by Robert Grady and Deborah Caswell (Prentice Hall, 1987), Mr. Schofield said that it takes between three and five years to assemble enough data to implement meaningful changes in the development process. "One of the things that I do is try to discourage management from using those metrics too soon," he said.

Eventually, Mr. Schofield said, he hopes to use tools to re-engineer existing applications back to the specification level, populate the development organization's central encyclopedia with processes and data, and then regenerate a customized version of the software.

[Editor's note: Mr. Schofield asked us to point out that these opinions are his own, and not those of the laboratory for which he works. Mail may be sent to him at the College of Santa Fe, where he is a member of the MIS faculty, 2201 San Pedro Drive N.E., Building 4, Albuquerque NM 87110, USA. Telephone messages may be left at (505) 292-7220.]