Basic Heroes

Qantas app displays SGML aircraft data

By sending aircraft-maintenance manuals to commercial airlines in SGML format, aircraft manufacturers such as Boeing can provide one electronic document that describes features common to all its aircraft and branches into separate sections to describe features of individual aircraft. But Qantas Airlines ran into some turbulence when dealing with these manuals. Australia's largest commercial airline found it difficult to manage, update, and provide access to the SGML data it receives every month. Instructors who assemble training materials had to parse the data, map the different branching combinations according to the different training courses, and store them as separate documents.

"They were overwhelmed" by managing the data this way, says Paul W. Mott, senior development engineer at Computervision Corp. Qantas wanted an SGML document management system to give its trainers and engineers fast access to current information about its aircraft. So Computervision built an SGML-parser module that integrates with Optegra Locator, a VB3 front end to the company's Optegra database system.

The parser module "bursts" the data into a hierarchical tree display when the user requests information on a particular aircraft, and a "recombination" module puts the data back together. The developers at the Bedford, Massachusetts-based Computervision used Bennet-Tec Information Systems' TList control to present the

parsed SGML data. For other parts of the app they used Sheridan's Data Widgets and Designer Widgets, Desaware's VersionStamper, and WhippleWare's VB Compress Pro.

It was easy to add the new SGML components to Optegra Locator, according to Mott, because each of the app's components is an individual EXE that carries out a single command. The EXEs use DDE to communicate with the main program that communicates directly with the database. The multiple-EXE approach "turned out to be so successful, that now we're writing all our programs this way," says Mott, who wrote the original app.

Instructors and engineers at the Qantas training facility in Sydney have been using their customized version of Optegra Locator since September, and the app is certainly taking flight. Qantas plans to distribute it to remote users at its maintenance facilities around the world, and to publish static copies of the application and data on CD-ROMs for engineers who don't have online access to the database. —*Nina Goldschlager*

*You can reach Paul W. Mott at pwm@connectnet.com.*
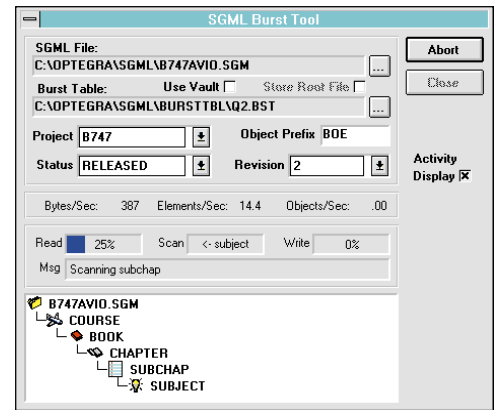
Warning: emerging mainframe problems

Tipped off by a customer in Switzerland, a company named Systar, whose parent company is based in Paris, France, learned of an undocumented interface released in the new Unisys 2200 operating system, SB5R3. Known as INSPECT\$, this interface allowed the retrieval of raw statistics of each run in the system, such as current I/O count, memory in use, files in use, and disk space in use. If this data were to be sampled, say every 15 seconds, the differences could be computed and consumption rates could be displayed. For the first time in the Unisys mainframe world, system operators could monitor workload in real time and dynamically track programs causing system slowdowns.

Tom Brazil, Unisys Mainframe product line manager, and Bob Blair, software engineer, who both work at a U.S. Systar branch, knew the enormous potential for a product that took advantage of this technology. "We also knew that the interface was going to be documented soon, and we had to beat the pack in getting a product to market that utilized this feature," says Brazil.

Using VB3 Pro, the two developed SystemScan, a mainframe performance monitor that enables system operators to detect emerging incidents before they escalate into damaging system problems. SystemScan displays detailed graphs that show which runs are using the most resources. If one run starts to act up, the
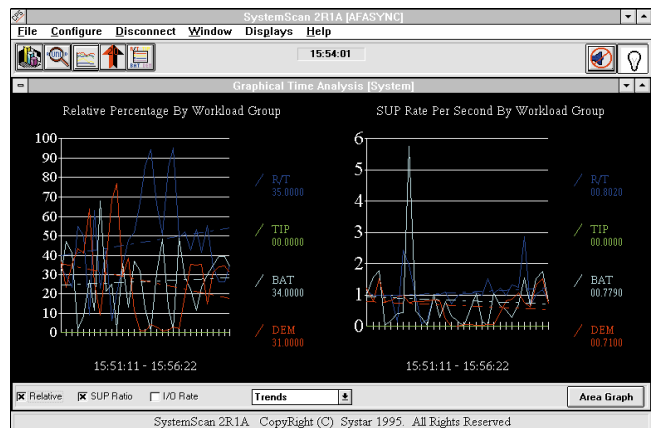
operator can immediately suspend it, before it affects response times. Before, operators had to solve these problems after the fact.

The programmers' main challenges were figuring out how to build a UI to connect to and access the INSPECT\$ interface, as well as developing the application as quickly as possible. "We chose Visual Basic, and sure enough, six weeks later we had a full-scale product that you could connect to a mainframe three different ways," says Brazil. The app has approximately 13,000 lines of code and a 120K EXE size. DevSoft's IP Port allowed them to quickly create a WinSock connection.

SystemScan sells for about $15,000 for a single copy. "And, they *are* selling," notes Brazil. —*Amy Little*

You can reach Tom Brazil by e-mail at braz@usa.systar-corp.com.