

FRANK HAYES ■ FRANKLY SPEAKING

# Paradigm Lost?

**I**S COBOL DEAD? Not if you believe the readers who wrote to object after I suggested in a recent column that “some IT practices and job titles will go the way of punched cards, Cobol and green screens.” One reader told me, “Cobol is alive and healthy. Oh, we use Web front ends, all of our data is on Oracle and our development staff uses a combination of desktop and server tools in their work. But the industrial-strength grunt work is Cobol on Unix servers. We haven’t found anything that can handle batch data in a more effective manner.”

“I suggest you check your facts,” wrote another reader. “Despite the barrels of ink claiming otherwise, the business world still runs on Cobol.” Still another said, “There continues to be this perception in the media that Cobol is dead. It’s very much alive and very much being evolved. Take note of the vendors (Fujitsu and Acucorp, for example) who have now ported this language to the Linux operating system. I don’t

think it’s going away anytime soon.” OK, let’s be clear on this: Cobol isn’t dead. It’s not at death’s door. It’s not even sick. It’s still an IT workhorse.

But let’s be clear on this, too: As recently as a decade ago, Cobol was the king of large-scale development. Now it’s not. And Cobol’s throne wasn’t usurped by PL/1 or any other competing language. Cobol lost its place to SAP and PeopleSoft and Siebel and Baan and other packaged enterprise applications.

IT shops stopped building those big projects and started buying. Our paradigm shifted, and Cobol lost its place at the core of corporate IT.

Dead? No. Demoted from king to hard-working commoner? Yes.

Now, here’s a more interesting question: Could King Cobol ever come back?

Answer: Maybe. Remember, just because a paradigm shifts doesn’t mean



**FRANK HAYES**, *Computerworld’s* senior news columnist, has covered IT for more than 20 years. Contact him at [frank\\_hayes@computerworld.com](mailto:frank_hayes@computerworld.com).

the original paradigm’s lost.

IT shifted to packaged apps because they were less expensive (we hoped) and more standardized than the aging custom Cobol code they replaced. After all, accounting is accounting. Inventory is inventory. Billing is billing. Why reinvent these routine business wheels over and over?

Why indeed? There was a reason companies built those custom systems in the first place. They want-

ed to gain a competitive advantage by fine-tuning their business processes in ways their competition couldn’t easily match.

The classic example is MCI’s original “Friends & Family” program from just a decade ago. It was essentially a specialized billing system for long-distance calls. AT&T’s oh-so-ordinary billing system couldn’t track “calling circles” the way MCI’s custom system could. So MCI got its foothold in long distance by customizing a routine accounting process.

You can’t get that kind of advantage with an enterprise package. Any competitor can buy the same software. Whatever you do to specially configure it, your competition can do the same.

If you want any chance at a unique business advantage, you’ll have to build it yourself. And that means

build-it-yourself enterprise applications just might make a comeback.

Yes, that would be another paradigm shift. CEOs and deep-thinking business gurus would have to decide that there’s a limit to the advantages of cost-cutting and that the new way to get competitive advantage from IT is pursuing unique business processes that only custom enterprise apps can deliver.

That won’t come this year — not in this economy. But watch for it. Paradigms keep shifting. And with modern design tools and development techniques, and without the albatross of decades-old legacy code that’s a nightmare to maintain, Cobol wouldn’t be a bad pick as the language of choice for the next wave of big custom development projects. After all, it’s mature. It’s familiar. It can do the job.

And who knows? Commoner Cobol just might have a shot at becoming the king of corporate IT all over again. ▀

**Net**  **COBOL**  
BY FUJITSU

Developer Tools Group  
3055 Orchard Drive  
San Jose, CA 95134  
Toll free: (800) 545-6774  
Phone: (408) 428-0300  
Fax: (408) 456-7272  
Email: [cobol@netcobol.com](mailto:cobol@netcobol.com)  
Web: <http://www.netcobol.com>