# Spinning Into Control

*Bell Labs' verification tool: coming to a compiler near you?*

"Telephone switches are among the most complex human-made objects," says Dr. Gerard Holzmann, director of research at Bell Labs. With 20 to 30 million lines of code, only control software for nuclear power plants and space flight software come close in complexity. "It's a very large, distributed system with a lot of concurrency and asynchronicity. On a modern switch there are 40 to 100 features that the user can select, and the switch cannot be down for more than three minutes per year. Try doing that with your PC."

"A few years ago, SPIN was used to verify the control software on a storm surge barrier—a movable dam—in Rotterdam. As a Dutchman, that pleased me tremendously. I'm also intrigued by the applications that NASA is doing on space flight software; we applied it on the Cassini-Huygens mission, which is now on its way to Saturn."
—Dr. Gerard Holzmann

Since 1980, Holzmann has focused on verifying high-reliability telephony systems, his work culminating in a widely used software package called SPIN—which was recognized this year with the ACM Software System Award. Holzmann, who feels that tools like his are increasingly a viable part of the development cycle, explains SPIN and its applications.

**Have verification tools lagged?**

Yes. The computational complexity of doing these checks is astoundingly large, so you have to be very clever. There are useful static analysis tools that scan the source text for common problems, but we deliver a result with the value of a mathematical proof. We've never seen a handwritten program—despite the fact that people say "This is robust code that we completely trust"—that didn't have many errors found by a mechanical tool like SPIN.

**But would they ever manifest themselves?**

That's tough to say. Researchers at NASA Ames used SPIN to study the control software for Deep Space 1 and found a couple of errors. In one case, the programmers said, "We'll fix it, but it's so unlikely to happen that we're really wasting our time." Then the mission was launched and it hit a bug in a component that hadn't been looked at. It was the same type of bug as the one they had said wasn't worth fixing. It deadlocked the software while it was in space.

**How long does SPIN take?**

Until recently, the only way to apply SPIN was to look at the source code, understand the algorithms, talk with the programmers and designers, and build a model for SPIN to analyze. The model has all the algorithmic content of the application, but at a higher level of abstraction. After you build the model, the actual checking is very fast. For Deep Space 1, it took three months to get to the point where one could check for errors. Now we can automatically generate models from source code. We did that on Lucent's Pathstar switch, developed in 1998; we proved the core processing software correct with mechanical model extraction techniques.

**Can we learn from the errors SPIN finds?**

The classic concurrency errors, like system deadlocks, happen rarely these days. We eventually want to merge checking tools into the compiler so that it can warn the programmer about subtle stuff like race conditions, complicated deadlocks and design requirements violations. With Pathstar, we worked with Lucent's best programmers. Some asked if it was worth doing verification, given that these people are so good. But the ratio of errors in software is fairly independent of experience. The novice will make simple mistakes. A veteran will make the same number while trying to do smarter things—it's like the saying: The better your four-wheel drive, the farther out you get stuck. —*A. Weber Morales*

## Sweet, Sweet Surveillance

*Orwell would have loved these DARPA projects.*

Networked sensors to detect the movement of hostile forces and materials—and longer-term approaches for changing the environment in which terrorism breeds—are the latest focus at Sandia National Laboratories in Albuquerque, New Mexico. The intent is to identify and track people in urban environments via hundreds of gadgets that may package a global positioning locator, sensor, RF communicator and computer.

But sensory observations are superficial. Researchers at the Lockheed Martin–owned lab suggest integrating sociology, group theory, biology and biosciences, as well as gang theory and the effects of racism, to understand terrorists and the setting they come from. The work would complement the Defense Advanced Research Projects Agency's collaboration with neurosciences to develop models of learning, and the Defense Threat Reduction Agency's exploration with Hollywood and the artificial intelligence community of creative computerized scenarios to address similar goals.

—*A. Weber Morales*