



Software systems have a complex lifecycle, and Gail Kaiser likes to work on all aspects of it. Her research ranges from creating systems that make recommendations to finding flaws in “non-testable” programs. “I like to find solutions in one domain and then generalize them,” she said.

For example, she is working on three “recommender” systems. One system monitors how biologists use tools for genomic analysis, and then gives novices recommendations based on the workflow of more experienced users. A second mines past experience to help programmers convert software to parallel code for multicore processors. The third helps computer science students solve certain errors in the code they write.

“We built all three systems independently, then noticed that they all used essentially the same architecture,” Kaiser said. “This let us derive a general reference architecture that might be useful in building future recommender systems.”

Kaiser is also interested in testing so-called “non-testable” programs. These include machine learning, simulation, data mining, optimization, and scientific computing systems. “I come from a software engineering background, but work at the borders of my discipline and operating systems, databases, and security,” said Kaiser. “I’m concerned with how to build systems—not just coding software, but how to design and test systems over their full lifecycle.

“Ordinarily, you can look at a program’s input and see if the output is correct. But what if you can’t tell? After all, non-testable programs are written to answer questions whose solutions are unknown. What if the answers are wrong in some cases, but not others? These programs could have all sorts of arbitrary errors, but how would we know?”

Kaiser has developed a number of approaches to test machine learning programs. One is a technique many math students will remember. She provides a problem, then changes the order of inputs to see if the program still generates the same answer.

“We found a lot of bugs in certain packages widely used in the machine learning community by using these approaches,” Kaiser said.

She has also developed methods to test for errors on computers deployed in the field. Field tests look at the widest possible range of software operating conditions. With so many variations, she can find subtle errors that elude even the most comprehensive lab testing programs.

“You can never get all the bugs out of them,” she said, “but the more bugs you remove, the better.”

B.S., Massachusetts Institute of Technology, 1979; M.S., MIT, 1980; Ph.D., Carnegie Mellon University, 1985

Testing What Cannot Be Tested

GAIL E. KAISER

Professor of Computer Science