Researchers aim to make debugging simpler

By Mike Crissey
Pittsburgh
August 6, 2004

Computer bugs, or errors in software, can mess up just about anything. They've been blamed for missing homework, blackouts, prison breaks and even the loss of multimillion-dollar space probes.

They can be costly to the economy - almost $US60 billion ($A86 billion) a year, a 2002 federal study estimates.

But they're difficult pests to eliminate, because doing so requires programmers to perform "an elaborate detective investigation", said Brad Myers, a Carnegie Mellon University computer science professor.

"You are trying to make guesses about where the problem is and prove your hypothesis. A lot of time programmers guess wrong ... and add new bugs because they were trying to fix something that wasn't broken."

But help is on the way. Myers and a graduate student, Andrew Ko, have developed a debugging program that lets users ask questions about computer errors in plain English: Why didn't a program behave as expected?

Funded by $US1.2 million ($A1.72 million) from the National Science Foundation, Whyline - short for Workspace for Helping You Link Instructions to Numbers and Events - is designed for programmers of all kinds, from hard-core professionals to weekend web designers.

Their work is part of a larger, nationwide project - called End Users Shaping Effective Software, or EUSES - to make computers friendlier for everyday users by changing everything from how they look to how they act.

Ultimately, perhaps, error messages will be easier to understand than "general protection fault" or "fatal exceptions".

You may not know it, but chances are you, too, are a programmer.

If you've created a spreadsheet, made macros in Excel or Word or used a web application to fetch news about your hobby or favourite celebrity, you've programmed.

But while software companies have allowed users to do more than ever with their computers, they've offered few ways for people to solve their own problems.

Programmers "are completely isolated from the people they are trying to serve and, of course, they are coloured by what they know and what they like", said Margaret Burnett, a computer science professor at Oregon State University and director of EUSES.

"For them, finding out the hex address of a division overflow is exactly what they need, and it never occurs to them that it is not something that someone else will need."

Programming is a lot like translation. Writing code involves taking what you want to do and converting it into computer language; hunting down a bug requires the opposite, which sometimes isn't as easy.

Whyline aims to simplify debugging and troubleshooting.

While testing a program, if something appears to go awry, a user can hit the "Why" button, which stops the program.

Whyline then offers questions based on programmed events.

For example, if a program contains rules about Pac-Man shrinking when he hits a ghost, Whyline will let the programmer ask "Why didn't Pac-Man resize?"

Lines of programming code related to the question are highlighted in a window.

Another window shows what happened while the program was running, with a flow chart and timeline.

"This tool puts about 90 per cent of the questions people want to ask before them," Myers said.

In studies involving a handful of graduate students, from relative novices to experienced programmers, Ko and Myers found that Whyline could help users find bugs eight times faster and do 40 per cent more programming.

Other debugging tools include AskIgor, a web-based program from Andreas Zeller, a professor at Saarland University in Saarbrucken, Germany.

Programmers tell AskIgor when a program works and when it fails, and AskIgor tries to determine differences that cause the bugs.

"When you compare debugging to walking through a dark house with a torch, AskIgor walks through the house and tells you what it is all about. It gives you a diagnosis," Zeller said.

For now, though, AskIgor only works for specific Linux programs, and it moves onto another challenge if it can't find the problem in 15 minutes.

Whyline, meanwhile, has been used only to debug programs in Alice, an academic programming language with a limited vocabulary of commands to make interactive 3-D worlds, like video games.

And the more complex a program gets, the harder it is for Whyline to offer the right questions and answers.

It can't answer complicated questions such as, "Why did Pac-Man shrink after he ate the power pellet and was hit by a ghost?"

Adding Whyline to a different language, like Java, which is 10 times as complex, could limit how much Whyline can help.

So Whyline is a very long way from getting incorporated into the world's most widespread software, Microsoft Corp's Windows operating system.

(When asked about its own debugging efforts, Microsoft didn't comment.)

Despite Whyline's current limitations, "we have this new way to think about debugging tasks and I don't think there is any question we can make it better than it is today", Myers said.

"It really doesn't have to help 95 per cent of the time to be useful. If it helps half the time it is still dramatic."

AP