



Researchers To Help Exterminate Bugs In Spreadsheets, Web Applications

An error in a spreadsheet or Web page calculation sounds harmless enough, unless you're the person whose retirement funds, credit history or medical treatment rely on decisions based on that calculation. A six-campus team of computer scientists led by Margaret Burnett at Oregon State University is working to help exterminate the bugs that infest the spreadsheets and other "programs" created by millions of computer users.

You may not think of yourself as a programmer, but that's just what you are if you've ever created a simple Web application that grabs data, such as current weather conditions, from another site, entered a formula in a spreadsheet or automated a repetitive task in your e-mail client or word processor. Experts estimate the number of these so-called "end-user programmers" to reach 55 million by 2005, and the same experts suggest that nearly half of the programs created by these end users have nontrivial bugs.

"For end-user programmers, software engineering isn't their job and it shouldn't be," said Burnett, a professor of computer science and director of the End Users Shaping Effective Software (EUSES) project. "Programming is a means to an end. They consider themselves successful when they've won new business through their Web site or completed a budget analysis. The problem, in part, is motivating them and focusing their attention on their programming errors amid other matters."

In addition to Oregon State, EUSES involves researchers from Carnegie Mellon University, Drexel University, Pennsylvania State University, the University of Nebraska, Lincoln, and Cambridge University. The project is supported by a five-year, \$2.6 million Information Technology Research award from the National Science Foundation.

Some of the first results from EUSES were recently presented at CHI 2004 in Vienna, Austria. In their CHI 2004 paper, Burnett and colleagues at Oregon State described a study of how best to tell spreadsheet programmers they may have created some buggy code. The Oregon State team compared immediate interruptions (such as pop-up error messages that demand immediate attention) with "negotiated" interruptions (similar to a word processor underlining a misspelled word).

Other Sponsors of the Forum

BASF



DAIMLERCHRYSLER



BEHR



accenture

Deutsche Bank



Continental



Changing the rules of business™



HEIDELBERG

watt



Drägermedical
A Dräger and Siemens Company

LipoNova



Dresdner Bank
Die Beraterbank

G.A.G BioScience



HTP HIGH TECH PLASTICS AG

Roland Berger
Strategy Consultants

EnBW
Die Energie AG.





BBK

Booz | Allen | Hamilton



Heraeus

IVECO



Content Partners of the Forum

By every measure, the negotiated interruptions were superior. The spreadsheet users learned more about the tools they were using, caught and corrected more errors and had a better idea of how well they had done. "We learned: Stay out of their way, give them hints to explore and they'll get more done," Burnett said.

In a separate paper, EUSES collaborators Andrew Ko and Brad Myers of Carnegie Mellon University described a novel environment that allows programmers to zero in on bugs by asking questions about "why did" or "why didn't" something happen. The new debugging interface helped programmers find bugs eight times faster and make 40 percent more progress in the programming task.

"It's an incredible scandal that the tools used today for debugging are the same ones available in the 1940s," said Myers. "It's time that new tools made their way into all programming environments so end-user and professional programmers won't have to suffer the way we still suffer now." Myers's work is also supported by a separate \$1.2 million NSF award to provide better tools for developing and debugging programs.

The two papers are part of the EUSES effort to develop new techniques and tools for debugging and testing, as well as to apply more behind-the-scenes intelligence for coaching programmers along the way. The work to create more effective tools also includes projects led by Martin Erwig and Gregg Rothermel of Oregon State, Sebastian Elbaum at Nebraska and Mary Shaw of Carnegie Mellon. These projects reason behind the scenes about the program itself, the data being processed and the edits the user makes to the program to keep potential errors out and to help the user track down the sources of errors that may already be present.

Another aspect of EUSES involves efforts to get inside end-user programmers' heads and understand the best ways to help them. Mary Beth Rosson of Penn State, Susan Wiedenbeck of Drexel University, Curtis Cook of Oregon State and Alan Blackwell of Cambridge University are leading efforts to observe end-user programmers in their natural habitats, watching as they debug simple web applications, surveying their attitudes and concerns and learning why people choose to use (or not use) the tools available to them.

"Our first principle in EUSES is 'Do no harm'-no matter what fancy features we add, we don't want to get in the way," Burnett said. "For experienced users, though, we want to encourage good testing habits and support quality control. The system will know about software engineering techniques and help the user monitor the dependability of what they've programmed. We want to provide the user with a willing collaborator in the system."

A third track in EUSES, led by Maggie Niess and Ellen Ford

of Oregon State, focuses on education. With participation from K-12 teachers, Niess and Ford are developing summer workshops for both teachers and students with follow-up in the classroom. This work is being done in partnership with Saturday Academy, directed by Ford. Saturday Academy is a non-profit cooperative that provides intensive, hands-on, after-school study opportunities in science and technology for middle- and high-school youth in both urban and rural Oregon communities. Besides considering education as a testbed for EUSES concepts, Niess and Ford are developing changes to how technology is taught, so that quality control is part of the learning process, not an afterthought.

The general concepts behind EUSES may also eventually find their way to almost any programming-like activity, such as programming a VCR, digital thermostat or home security system. Rather than risk making an error, many people skip that sort of programming altogether. According to Burnett, EUSES may uncover reasons for this avoidance and lead to systems that encourage people to safely take that leap.

More information:

www.nsf.gov/od/lpa

eecs.oregonstate.edu/EUSES/

www.nsf.gov