PC MAGAZINE ®

THE INDEPENDENT GUIDE TO TECHNOLOGY

SUBSCRIBE ▶▶▶ NOW

| HOME | REVIEWS | DOWNLOADS | SOLUTIONS | NEWS | OPINIONS | SHOP | DISCUSSIONS |

❯ **Product Bulletin**   ❯ **Special Reports**   ❯ **Show Reports**

My Account | **Sign In**   Not a member? **Join now**   SEARCH

Home > **News and Analysis** > **Debugging for the Masses**

Debugging for the Masses
By Sebastian Rupley
May 14, 2004

Total posts: 2

Print   Email   Save   Discuss   Subscribe

Programming may be thought of as dry and complicated stuff, but more and more nonprogrammers are doing it—whether they stop and realize it or not. Tools from spreadsheets to Web authoring tools are chock-full of ways that average users can script seemingly simple programmatic instructions into the information they work with. Those same people may be using the tools to handle someone else's retirement planning or medical information. The ease with which the neophyte programmers make mistakes has given rise to new grants from the National Science Foundation (NSF) to develop smarter debugging tools for everyone. Some of the first results were shown at the CHI 2004 conference in Vienna, Austria.

"It's a shame that the tools used today for debugging are basically the same ones available in the 1940s," says Brad A. Myers, an associate research professor at Carnegie-Mellon University in Pittsburgh and a collaborator on the End Users Shaping Effective Software (EUSES) project. Myers' work is supported by a $1.2 million NSF award to provide better tools for debugging. The directors of EUSES estimate that the number of end-user programmers in the United States will reach 55 million by 2005, as compared to only 2.75 million professional programmers.

In a paper that EUSES collaborators Myers and Carnegie-Mellon graduate student Andrew Ko presented at the CHI 2004 conference, they described a novel environment that allows programmers to find bugs by asking "why did" or "why didn't" an event happen. They took an existing programming environment called Alice (www.alice.org), and put a "Why?" button in it, which can allow a user trying to create, say, a simple graphical animation to stop the animation at any point where an expected event went wrong. An example is shown in the screen shot , where the user is seeking the answer to why a Pac-Man character didn't make an expected left turn in an animation.

The button returns code location-specific information about the presence or absence of the right instructions for the event. In Myers' and Ko's research, the new debugging interface helped programmers find bugs

John C. Dvorak

Hoax Nation

How reliable are the images we see in the news?

Plus: The Looming Legal Threat to Wi-Fi

breaking news
12:00AM
Straightening Out the 'Longhorn' Time Frame

eight times faster and make 40 percent more progress in the programming task. Both Microsoft and IBM have taken an interest in their work.

"For end-user programmers, software engineering isn't their job and it shouldn't be," says EUSES project director Margaret M. Burnett, a professor of computer science at Oregon State University. In a separate CHI 2004 paper, Burnett and her Oregon State colleagues described their study of how best to tell spreadsheet programmers that they may have created buggy code. The team developed a fault localization routine, "What You See Is What You Test," for use within spreadsheets, where cells that might contain errors are indicated with darker colors corresponding to an increased likelihood of error. A picture of how it looks is available online.

The EUSES project is supported by a five-year, $2.6 million Information Technology Research award from the NSF. The general concepts that EUSES members are developing may also eventually find their way into almost any programming-like activity, such as programming a VCR or controlling a digital thermostat or home security system.