

'Combinatorial' approach squashes software bugs faster, cheaper

A team of computer scientists and mathematicians from the National Institute of Standards and Technology and the University of Texas, Arlington is developing an open-source tool that catches programming errors by using an emerging approach called “combinatorial testing.” The NIST-Texas tool, described at a recent conference,* could save software developers significant time and money when it is released next year.

Studying software crashes in a variety of applications from medical devices to Web browsers, NIST researchers obtained hard evidence to support long-held conventional wisdom: most software failures result from simple events rather than complex ones.** Even for Web browsers containing hundreds of different variables, most failures were caused by interactions between just two variables. Nonetheless, in the applications that the researchers studied, additional failures could result from interactions of up to six variables.

Based on that insight, the NIST-Texas team went beyond the popular practice of “pairwise testing,” or exploring interactions between only two variables at a time, and designed a method for efficiently testing different combinations of settings in up to at least six interacting variables at a time. Their technique resembles combinatorial chemistry in which scientists screen multiple chemical compounds simultaneously rather than one at a time.

For example, imagine a word-processing program that features 10 different text formats. Certain combinations of settings (such as turning on superscript, subscript and italics at the same time) could cause the software to crash. Trying all possible combinations of the 10 effects together would require 1,024 tests. However, testing all possible combinations of any three effects requires just 13 different tests, thanks in part to the fact that if the tests are selected judiciously the 10 different variables allow you to explore 120 combinations of “triples” simultaneously.

The new tool generates tests for exploring interactions among the settings of multiple variables in a given program. Compared to most earlier combinatorial testing software, which has typically focused on testing interactions between just two variables, the tool excels at quickly generating efficient tests for 6-way interactions or more.

The researchers plan to release the tool early next year as open-source code. They currently are inviting developers to participate in beta testing of the tool before release. This new approach for finding bugs to squash may be particularly useful for increasing the reliability of e-commerce Web sites, which often contain many interacting variables, as well as industrial process controls, such as for robotic assembly lines of high-definition televisions, which contain many interacting software-controlled elements that regularly turn on and off.

References:

* Y. Lei, R. Kacker, D. R. Kuhn, V. Okun and J. Lawrence, IPOG: A general strategy for t-way software testing. IEEE International Conference on Engineering of Computer-Based Systems March 26-29, 2007, pp 549-556, Tucson AZ, USA.

** D.R. Kuhn, D.R. Wallace and A.J. Gallo, Jr. Software fault interactions and implications for software testing. IEEE Trans. on Software Engineering, June 2004 (Vol. 30, No. 6) pp. 418-421.

Source: National Institute of Standards and Technology

This document is subject to copyright. Apart from any fair dealing for the purpose of private study, research, no part may be reproduced without the written permission. The content is provided for information purposes only.