Exposing unforeseen consequences of software change

David Notkin
Computer Science & Engineering
University of Washington
Seattle Washington 98195-2350 USA
notkin@cs.washington.edu

Abstract: Changing source code can have unintended effects on a program's behavior. Seemingly trivial changes have incurred significant cost, distress, and catastrophe: that is, the concern about the consequences of a change is not merely theoretical. At the same time, many — perhaps most — software changes do not cause problems in practice, instead improving the program's behavior in intended ways. The changes that improve program behavior clearly collectively outweigh those that harm it.

Nevertheless, nobody would argue that programmers make changes with certainty about the future behavior of a program. This is in part due to Dijkstra's observation from decades ago about the conceptual gap between the static program and the dynamic execution process. This observation led to many aspects of structured programming, most notably the aggressive use of one-in/one-out control structures such as if-then-else statements and while loops. However, for a number of reasons, the source-behavior relationship has become more opaque rather than less so, leaving programmers with relatively little help in distinguishing between changes leading to intended versus unintended behaviors.

To aid programmers facing these difficulties, we have developed an approach that takes a pair of program versions, models each version's source code as static call graphs and each version's behavior as dynamic call graphs, partitions these four dependence graphs based on their set intersections, and identifies partitions that tend to expose changes that are likely to have unforeseen consequences on the program's source-behavior relationship.

This is joint work with Reid Holmes.

Biography: David Notkin received his bachelor's in computer science, cum laude with honors, from Brown University in 1977, and his PhD in computer science from Carnegie Mellon University in 1984. He has been on the Computer Science & Engineering faculty at the University of Washington since 1984, serving as department chair (2001-06) and now holding the endowed Bradley Chair. Among his honors and awards are a 1988 National Science Foundation Presidential Young Investigator Award, the 2000 University of Washington Distinguished Graduate Mentor Award, and election as an ACM Fellow (1998) and IEEE Fellow (2007). He is editor-in-chief of ACM Transactions on Software Engineering and Methodology (2007-2013). He is a member of the Computing Research Association board of directors (2005-), and was co-chair of the Academic Alliance of the National Center for Women in Information Technology (2004-2008). He was the program chair for the 1st ACM SIGSOFT Symposium on the Foundations of Software Engineering and program co-chair for the 17th International Conference on Software Engineering. He served as the chair of ACM SIGSOFT, the special interest group on software engineering (1997-2001). He has advised 19 PhD students and several dozen master's students. He was a visiting faculty member at both Tokyo Institute of Technology and Osaka University in 1990-91. In 1997-98, he spent four months as a visiting researcher at the IBM Haifa Research Laboratory, and in 2006-07 he was a visiting researcher at Lund University in Sweden.