

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers

Computer Science and Engineering, Department
of

2006

Helping End-User Programmers “Engineer” Dependable Software

Gregg Rothermel

University of Nebraska-Lincoln, gerother@ncsu.edu

Follow this and additional works at: <https://digitalcommons.unl.edu/cseconfwork>



Part of the [Computer Sciences Commons](#)

Rothermel, Gregg, "Helping End-User Programmers “Engineer” Dependable Software" (2006). *CSE Conference and Workshop Papers*. 134.

<https://digitalcommons.unl.edu/cseconfwork/134>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Helping End-User Programmers “Engineer” Dependable Software

Gregg Rothermel
Department of Computer Science and Engineering
University of Nebraska — Lincoln

Abstract

Not long ago, most software was written by professional programmers, who could be presumed to have an interest in software engineering methodologies and in tools and techniques for improving software dependability. Today, however, a great deal of software is written not by professionals but by end-users, who create applications such as multimedia simulations, dynamic web pages, and spreadsheets. Applications such as these are often used to guide important decisions or aid in important tasks, and it is important that they be sufficiently dependable, but evidence shows that they frequently are not. For example, studies have shown that a large percentage of the spreadsheets created by end-users contain faults, and stories abound of spreadsheet faults that have led to multi-million dollar losses. Despite such evidence, until recently, relatively little research had been done to help end-users create more dependable software.

In this talk, I'll show that it's possible to address this problem by adapting formal software engineering techniques to the realm of end-user programming. Focusing on work done in the spreadsheet application paradigm, I present several of approaches to using program analysis techniques to help end-users build more dependable spreadsheets. Behind the scenes, these methodologies use static analyses such as dataflow analysis and slicing, together with dynamic analyses such as execution monitoring, to support user tasks such as validation and fault localization. I show how, to accommodate the user base of spreadsheet languages, an interface to these methodologies can be provided that does not require users to understand the theory behind the analyses, yet supports the interactive, incremental process by which spreadsheets are created. I present empirical results gathered in the use of these methodologies, that highlight several costs and benefits tradeoffs.

Biography

Gregg Rothermel's research interests include software engineering and program analysis, with emphases on the

application of program analysis techniques to problems in software maintenance and testing, end-user software engineering, and empirical studies.

Dr. Rothermel is a Program Co-Chair for the 2007 International Conference on Software Engineering, and a member of the Editorial Boards of the Empirical Software Engineering Journal and Software Quality Journal. Previous positions include Associate Editor in Chief for *IEEE Transactions on Software Engineering*, Program Chair for the 2004 ACM International Symposium on Software Testing and Analysis, and Chair of the Steering Committee for the International Conference on Software Maintenance. He has served on the program committees for numerous software engineering conferences including the IEEE International Conference on Software Engineering, the ACM International Symposium on Foundations of Software Engineering, and the ACM International Symposium on Software Testing and Analysis.

Dr. Rothermel received a B. A. in Philosophy from Reed College, an M. S. in Computer Science from SUNY Albany, and a Ph. D. in Computer Science from Clemson University. Prior to returning to academia, Dr. Rothermel was employed as a software engineer, and as Vice President, Quality Assurance and Quality Control for Palette Systems, Inc., a manufacturer of CAD/CAM software. Since 2004, he has been a Professor and Jensen Chair of Software Engineering in the Computer Science and Engineering Department at University of Nebraska — Lincoln, where he is a founding member of the Laboratory for Empirically-based Software Quality Research and Development (ESQuaReD).