

2004

# Using Source-Code Analysis to Help End-user Programmers Create Dependable Software

Gregg Rothermel

*University of Nebraska-Lincoln*, [grothermel2@unl.edu](mailto:grothermel2@unl.edu)

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



Part of the [Computer Sciences Commons](#)

---

Rothermel, Gregg, "Using Source-Code Analysis to Help End-user Programmers Create Dependable Software" (2004). *CSE Conference and Workshop Papers*. 136.

<http://digitalcommons.unl.edu/cseconfwork/136>

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.

## Using Source-Code Analysis to Help End-user Programmers Create Dependable Software

Gregg Rothermel

*Department of Computer Science and Engineering  
University of Nebraska – Lincoln*

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. Despite such evidence, until recently, relatively little research had been done to help end-users create more dependable software.

We have been working to address this problem by finding ways to provide at least some of the benefits of formal software engineering techniques to end-user programmers. In this talk, focusing on the spreadsheet application paradigm, I present several of our approaches, focusing on methodologies that utilize source-code-analysis techniques to help end-users build more dependable spreadsheets. Behind the scenes, our 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 in a manner that does not require an understanding of the theory behind the analyses, yet supports the interactive, incremental process by which spreadsheets are created. Finally, I present empirical results gathered in the use of our methodologies that highlight several costs and benefits tradeoffs, and many opportunities for future work.