Utility and accuracy of smell-driven performance analysis for end-user programmers

Christopher Chambers, Christopher Scaffidi*
{ chamberc, scaffidc} @onid.orst.edu

School of Electrical Engineering and Computer Science
Oregon State University
1148 Kelley Engineering Center
Corvallis, OR 97331

An earlier version of this paper appeared as the following: Chambers, C., and Scaffidi, C. (2013) Smell-driven performance analysis for end-user programmers, IEEE Symposium on Visual Languages and Human-Centric Computing.

The current paper expands on the earlier paper by providing (1) an enhanced detection method called smell-driven profiling that also incorporates runtime analysis in addition to the static analysis presented in the earlier work, and (2) a new study assessing how accurately the original and enhanced methods can identify non-trivial performance problems.

Refer to the JVLC website for the paper in its entirety
Abstract

This paper proposes a technique, called Smell-driven performance analysis (SDPA), which automatically provides situated explanations within a visual dataflow language IDE to help end-user programmers to overcome performance problems without leaving the visual dataflow paradigm. An experiment showed SDPA increased end-user programmers’ success rates at finding performance problems and decreased the time required for finding solutions. Another study, based on using SDPA to analyze a corpus of example end-user programs, revealed that it is usually accurate at identifying performance problems. Based on these results, we conclude that SDPA provides a reliable basis for helping end-user programmers to troubleshoot performance problems, as well as a potential foundation for future work aimed at training users and at aiding code reuse.

Keywords: end-user programming; performance; visual language

Highlights

- Smell-Driven Performance Analysis (SDPA) finds dataflow performance problems.
- SDPA provides situated explanations within the visual dataflow language.
- We present an extended form of the technique that incorporates runtime profiling.
- In a user study, participants could more easily diagnose performance problems.
- A second study confirmed that profiling improves accuracy.