Autom Software Eng (2007) 14:1–2
DOI 10.1007/s10515-007-0006-4

Introduction

Kurt Stirewalt · Virginie Wiels

Published online: 26 January 2007

This issue of the Journal of Automated Software Engineering comprises a selected set of papers that originally were among the nominated papers for the best paper award of the 19th International Conference on Automated Software Engineering (ASE 2004), which took place in September 2004 in Linz, Austria. Authors of the selected papers were asked to submit an expanded version for publication in the journal. The extended papers were then reviewed thoroughly and a subset of these papers was selected for publication in this issue.

The paper by Apiwattanapong, Orso, and Harrold describes a new technique for comparing two versions of an object-oriented program, identifying both differences and correspondences. The technique is implemented in a tool, called JDiff, which is described in the paper and evaluated by means of four empirical studies.

The paper by Heimdahl and George presents another data point in the debate regarding the effect of test-suite reduction on the quality of the reduced suite when this suite is automatically generated from a formal model. The authors report the results of an experiment that probes reduced suites for their effectiveness at finding faults. The results indicate that the size of the suites can be dramatically reduced but that the quality of the reduced test-suites is adversely affected. This decrease in quality is particularly significant in the domain of specification-based testing.

The paper by Marceau et al. describes a scripting language that can be used to codify and reuse the many tedious and repetitive operations that one must perform when interacting with a debugger. The language offers powerful primitives that can precisely

K. Stirewalt (✉)
Department of Computer Science and Engineering, Michigan State University,
East Lansing, Michigan 48824, USA
e-mail: stire@cse.msu.edu

V. Wiels
ONERA-CERT/DTIM, 2 Avenue Edouard Belin,
BP 4025, Toulouse cedex 4, France
e-mail: virginie.wiels@cert.fr

and concisely capture many important debugging and comprehension metaphors. In addition to describing the language, the paper also describes a pair of debuggers, one for Java and the other for Scheme, built in accordance with these principles.

The paper by Taghdiri and Jackson presents a method for statically analyzing a program to uncover counter-examples to user-provided assertions. The method requires no program annotations. Rather, it automatically infers a context-dependent specification for each procedure call so that only as much information about a procedure is used as is needed to analyze its caller. Specifications are inferred iteratively and refined in response to spurious counterexamples. When the analysis terminates, any remaining counterexample is guaranteed to be valid. However, the absence of a counterexample does not prove the validity of the assertion.

We greatly appreciate the work that the authors invested in extending their papers and in responding to requested revisions. We would also like to thank the anonymous referees, whose hard work and attention to detail made this special issue possible. Finally, we would like to thank Paul Grünbacher, the general chair of ASE 2004, for a wonderful conference venue in Linz.