Workshop on Program COmprehension through Dynamic Analysis (PCODA'06)

Andy Zaidman
Delft Univ. of Technology,
The Netherlands &
University of Antwerp, Belgium
Andy.Zaidman@ua.ac.be

Orla Greevy
Software Composition
Group, Univ. of Bern,
Switzerland
greevy@iam.unibe.ch

Abdelwahab Hamou-Lhadj
School of Information
Technology and Engineering
University of Ottawa, Canada
ahamou@site.uottawa.ca

Abstract

Software maintenance and evolution can be made easier with program comprehension techniques. The aim of this workshop is to gather together researchers working in the area of program comprehension with an emphasis on dynamic analysis. We are interested in investigating how dynamic analysis techniques are or can be used to enable better comprehension of a software system. The objective is to find common case studies, compare existing techniques, and find possible symbioses for existing solutions. Building upon the previous edition of the workshop, PCODA 2005, we aim to set up a forum for exchanging experiences, discussing solutions, and exploring new ideas.

1 Introduction

Without consistent or adequately complete documentation, maintainers are faced with the inevitable problem of understanding how the system is implemented prior to undertaking any maintenance task. Research into the discipline of program comprehension aims to reduce the impact of this problem. Studies [1] have shown that software engineers tend to spend up to 50% of their time trying to comprehend the structure of a software system. This has led to the development of several program comprehension models and techniques such as those presented by Storey et al. [2].

PCODA focuses on program comprehension techniques that rely specifically on dynamic analysis. As such, this workshop builds on previous experiences from existing workshops and conferences such as ICPC (International Conference on Program Comprehension), WODA (Workshop on Dynamic Analysis), and PCODA.

2 Topics

The main theme is program comprehension through dynamic analysis. Topics of interest include:

- **Program comprehension models:**
  - Theories and models for software comprehension based on dynamic analysis
  - Program comprehension processes and strategies involving dynamic analysis techniques
  - Research methodologies
- **Techniques and tools:**
  - Applications of dynamic analysis techniques to program comprehension
  - Strengths and limitations of existing techniques
  - Trace analysis and exploration techniques
  - Techniques for reducing the large size of runtime information
  - Hybrid analyses that involve both static and dynamic analysis
  - Dynamic analysis tools with an emphasis on program comprehension
- **Evaluation Techniques:**
  - Criteria for evaluating techniques
  - Experiments and case studies with a focus on program comprehension using dynamic analysis
  - Empirical effectiveness studies

3 Goals

We have a number of goals for each of the participants:

- Bring forward innovative techniques in the field
- Introduce comprehension strategies based on dynamic analysis that have proven themselves in the field
- Share results of case studies and experiments – find common case studies
- Compare various techniques
- Discuss empirical studies to validate results
- Inventorize dynamic analysis techniques (with an emphasis on program comprehension) used in research

References

[1] T. A. Corbi. Program understanding: Challenge for the 1990s. *IBM Systems Journal*, 28(2):294–306, 1989.

[2] M.-A. D. Storey, K. Wong, and H. A. Müller. How do program understanding tools affect how programmers understand programs? In *WCRE*, pages 183–207. IEEE, 1997.