Empir Software Eng
DOI 10.1007/s10664-008-9097-1

EDITORIAL

Special issue on information retrieval for program comprehension

Letha Etzkorn · Tim Menzies

Editor: Lionel Briand

Welcome to the special issue on information retrieval for program comprehension (IR4PC). IR4PC employs various interdisciplinary information search techniques to examine the properties of both existing (legacy) and newly created software. IR4PC is important for software reuse, software maintenance and evolution, and reverse engineering, just to mention a few areas.

Back in the 1980s and early 1990s, much program comprehension involved representing program code as control and data flow graphs. Recognizing program constructs was performed by comparing flow graphs to a plan library of known constructs (e.g. Rich and Waters 1989). However, formal non-heuristic approaches to program comprehension have been shown to be NP-hard and their success was often illustrated only in toy domains (Woods and Yang 1996). For this reason, heuristic approaches acquired new importance.

In the 21st century, much program comprehension research has focused on applying various information retrieval techniques (e.g. text mining, LSI, knowledge-based NL understanding) to software. These new IR4PC semantic measures examine informal information in the tokens within the software itself (e.g. identifier names, function names and variable names, code comments) as well as the natural language content in external documentation such as software requirements documents or software design documents.

In the past, IR4PC techniques have been successfully applied to (among other areas) static concept location (using information derived from informal tokens together with structural information such as call graphs to locate code sections that are related to given concepts), to determining whether a particular software component is reusable, to dynamic search or software reconnaissance (examining informal tokens along execution traces of program executed with and without a particular feature), to developer identification (determining which developer is the best one to perform a particular task), to bug location

L. Etzkorn
University of Alabama in Huntsville, Huntsville, AL, USA

T. Menzies (✉)
West Virginia University, Morgantown, WV, USA
e-mail: tim@menzies.us

(determining which parts of the source code should be changed to fix a particular bug), to traceability recovery, to detecting code clones (different code fragments that are very similar to each other such that any modifications must be made to all similar fragments), for impact analysis (to determine which parts of source code are affected by a particular change), and to provide a new kind of software metric independent of simple syntactic code variations.

The goal of this special issue was to survey the state of the art. After several rounds of extensive reviewing, we selected the following papers:

1. *Using Information Retrieval based Coupling Measures for Impact Analysis* (Marcus, Poshyvanyk, Ferenc and Gyimóthy):

   IR4PC research area examined: This paper investigates, for the purpose of impact analysis, new software metrics that are independent of syntactic or structural code variations. Therefore it is representative of two important IR4PC research areas.
   Summary of paper: In this paper, the authors present a new set of coupling measures for Object-Oriented (OO) software systems measuring conceptual coupling of classes; i.e. the degree to which the identifiers and comments from different classes relate to each other. The paper reports the findings of a case study in the source code of the Mozilla web browser, where the conceptual coupling metrics were compared to nine existing structural coupling metrics. One of the new coupling measures proved to be a better predictor than existing metrics for determining classes impacted by changes (impact analysis)

2. *An Information Retrieval Process to Aid in the Analysis of Code Clones* (Tairas, Gray):

   IR4PC research area examined: code clone detection. As the reviewers commented, the problem of comprehending large sets of cloning data is very important and certainly is of practical value. This paper provides a taxonomy or classification of code clones, which is an essential and long awaited technology for applying code clone detection to large scale source code
   Summary of paper: In this paper, Latent Semantic Indexing (LSI) is used to cluster clone classes that have been identified initially by a clone detection tool. Using a case study with the Microsoft Windows NT kernel source code, LSI is used to detect trends and associations among the clustered clone classes and determine if they provide further comprehension to assist in the maintenance of clones.

3. Assessing IR-based Traceability Recovery Tools through Controlled Experiments (Oliveto, Lucia, Tortora):

   IR4PC research area examined: traceability recovery. As several of the reviewers commented, there has been a need for many years to study the degree to which human decision making improves when using traceability recovery tools. This paper addresses this issue.
   Summary of paper: This paper reports a controlled experiment to assess the usefulness of an IR-based traceability recovery tool. The use of a traceability recovery tool significantly reduces the time spent by the software engineer with respect to manual tracing. The paper comments extensively on the retrieval accuracy achieved by the software engineers with and without the tool support and with different levels of experience and ability.

4. An Empirical Analysis of Information Retrieval based Concept Location techniques in Software Comprehension (Cleary, Exton, Buckley, English).

   IR4PC research area examined: concept location. In addition to examining a potential new approach to concept location, this paper also did a large empirical study of several concept location techniques. It also provided a fairly extensive review of existing concept location techniques.
   Summary of paper: This paper studies a new concept location approach that combines IR theory with cognitive theory. The approach is novel in that it leverages implicit information available in system documentation. Existing approaches such as VSM, LSI, and KLD were compared to each other, and to the new approach. Surprisingly, empirical evaluation of the new approach showed little performance benefit overall compared to KLD (in other environments this kind of approach had shown as superior to KLD), although both the new approach and KLD outperform the other approaches. Several possible explanations are forwarded for this finding.

The papers in this special issue thus address five of the most important information retrieval in program comprehension research areas.

This issue would not have been possible without the dedicated work of many people. We offer our heart-felt thanks to all our reviewers, all our authors, and the excellent production staff at Springer-Verlag.

References

Rich C, Waters RC 1989 Intelligent assistance for program recognition, design, and debugging, AI Memo 1100, MIT Artificial Intelligence Laboratory

Woods S, Yang Q 1996 The Program Understanding Problem: Analysis and A Heuristic Approach, Proceedings of the 18th International Conference on Software Engineerig, ICSE-18, pages 6–15

Letha H. Etzkorn received the bachelor's and master's degree in Electrical Engineering from the Georgia Institute of Technology and the PhD degree in Computer Science from the University of Alabama in Huntsville. She is an associate professor in the Computer Science Department of the University of Alabama in Huntsville. Her primary research areas are in software engineering, primarily software metrics and

program comprehension (including information retrieval for program comprehension), and mobile and intelligent agents.

Dr. Tim Menzies (tim@menzies.us) has been working on advanced modeling and AI since 1986. He received his PhD from the University of New South Wales, Sydney, Australia and is the author of over 170 refereed papers. A former research chair for NASA, Dr. Menzies is now a associate professor at the West Virginia University's Lane Department of Computer Science and Electrical Engineering. For more information, visit his web page at http://menzies.us.