Title : Crowd Debugging
Authors : Fuxiang Chen and Sunghun Kim
Review written by : Doti SandhyaRani

Motivation:

Research shows that, many developers are facing common issues in their source code and there are lots of recurring questions in stack overflow, in which developers need to check them each time and should be well known with the content to obtain the crowd's knowledge. But due to the huge amount of QA data, it is hard for developers to catch up. Previous research has shown that users ask recurring questions about code idioms in the developers QA site, Stack Overflow. Although mechanisms have already put into place to avoid recurring questions, still there exits a huge number of recurring questions. This can be improved by mining QA sites to help developers debug their code.

Proposed Solution:

The proposed solution is mining the QA sites, to help developers debug their code using crowd knowledge and to evaluate the number of repeated questions in Stack Overflow. The crowd knowledge provides the solutions and explanations to the source code issues which are absent from existing static analysis tools (FindBugs, PMD, and JLint) when compared. This mechanism identifies the defective code fragments by first detecting code clones. And then coupled with crowds suggested solution and reports to developers for the feedback.

Evaluation:

The approach of mining QA sites is evaluated based on generating reasonable number of (189) warnings in eight high- quality and well maintained projects, and produce a high percentage 90.5% (171) of bugs which are confirmed by the developers. Comparing the results with existing static analysis tools (FindBugs, JLint and PMD) , out of 171 bugs identified, six of the confirmed bugs are detected by FindBugs and these static analyses tools missed 165 (96.6%) bugs when detecting the six of the same bugs.

Analysis:

Good Point: In this paper, the tool not only identifies the bugs, but also provides comprehensive explanation using the SO posts to the developers. This will further enhance their understanding on the detected warnings.

Bad Point: The approach focused only on the code terms and the SO title. But it is not sufficient to determine the effectiveness of mining QA sites to help developers debug their code. Because developers may present some code terms in natural text language in SO question and SO answer which we may have missed them.

Potential Project: As developers may write some code terms in natural text language which we are not able to recognize them. The project can be extended to investigate pattern matching techniques to identify the natural text in SO question and SO answer.

Questions:

1) What is the need to compare the confirmed bugs with the static analysis tools in the proposed solution?

2) What approach you will implement to recognize the natural text in SO site to match the pattern with the crowd knowledge ?