Aiding Java Developers with Interactive Fault Localization in Eclipse IDE

Gergő Balogh, Ferenc Horszth, Árpád Beszédes
Department of Software Engineering, University of Szeged, Hungary

Problem
Spectrum-Based Fault Localization methods are popular due to their relative simplicity to implement.

Studies highlighted some barriers to the adoption of SBFL in practical settings.

Goal
Increase the practical usefulness of SBFL tools.

Solution
The developer has additional information about the system of which the SBFL engine is not aware.

Involve interactivity between the user and the FL algorithm.

Challenges
Do not generate unnecessary overhead by disturbing the typical workflow of the developers.

Tools should be extensible to make it possible to integrate various already existing SBFL algorithms and future iFL algorithm variants.

Researchers have to define a set of meaningful options to select from.

Appropriate actions for various kinds of feedback and relevant code entity context should be defined.

Achievements
The knowledge of the user is exploited in the ranked list, with which larger code entities can be repositioned in their suspiciousness.

The process starts by calculating an initial rank based on some traditional SBFL.

The elements are then shown to the user, and the SBFL engine is waiting for user feedback.

The user investigates the recommended item and gives a feedback.

Experimental Results
Goal to have a preliminary view of expected improvement in fault localization effectiveness of iFL.

<table>
<thead>
<tr>
<th>Program</th>
<th>Threats</th>
<th>iFL</th>
<th>Diff</th>
<th>Imp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>commons-lang</td>
<td>3.81 (0.19%)</td>
<td>2.00 (0.09%)</td>
<td>-1.81 (-0.10%)</td>
<td>47.96%</td>
</tr>
<tr>
<td>commons-math</td>
<td>7.96 (0.17%)</td>
<td>7.21 (0.15%)</td>
<td>-0.75 (-0.02%)</td>
<td>8.56%</td>
</tr>
<tr>
<td>joda-time</td>
<td>17.56 (0.43%)</td>
<td>4.70 (0.12%)</td>
<td>-12.86 (-0.31%)</td>
<td>73.23%</td>
</tr>
<tr>
<td>Average</td>
<td>9.75 (0.20%)</td>
<td>5.64 (0.12%)</td>
<td>-4.11 (-0.14%)</td>
<td>63.98%</td>
</tr>
</tbody>
</table>

Simulation of User Actions
We implemented the approach to handle Java systems using simulated users instead of real programmer feedback.

Item is not faulty, neither its context.
Everything but context gets 0 score.

Item is not faulty, fault is in context.
Whole context gets 0 score.

Item is not faulty, neither its context?

iFL for Eclipse
It is an Eclipse plug-in for supporting iFL for Java projects. It reads the tree of project elements and lists them in a table.

User sends feedback to the FL engine about the next element in the table.

The main UI is an Eclipse part, a graphical panel, serving as the front end. It is connected to the back end components, whose purpose is the update of scores and the recalculation of the rank list based on user input.

Plugin Architecture

Fast-forward to bugs