Parichayana: An Eclipse Plugin for Detecting Exception Handling Anti-Patterns and Code Smells in Java Programs

Ashish Sureka (ABB Corporate Research Center, India) ashish.sureka@in.abb.com

**Exception handling**

- Signs in the source code which are not defects
- Does not prevent the program from functioning - Does not cause compile errors
- Indicators of deeper and bigger problems

**Novel Contributions**

- Eclipse plug-in (called as Parichayana) for detecting exception handling anti-patterns and code smells in Java programs
- Parichayana is the only tool which detects all the 14 exception handling anti-patterns in comparison to other tools like CHECKSTYLE, FINDBUGS, JEXCRIULE, and PMD
- Extend the Eclipse IDE and create new menu entries and associated action via the Parichayana plug-in (free and open-source hosted on GitHub)
- Used Parichayana on several large open-source Java based projects and detected presence of exception handling anti-patterns

**Plug-in Architecture**

- We use the Eclipse Plug-in Development Environment (PDE) which consists of PDE UI (editors, wizards and launchers), API tools and build facilities
- We use the Eclipse JDT (Java Development Tools) and the Eclipse AST (Abstract Syntax Tree) libraries to access and read the elements of a Java program.
- The AST is a detailed tree representation of the Java source code (comparable to the DOM tree model of an XML file).
- We use the classes in org.eclipse.jdt.core.dom package such as CatchClause for extracting the body of a catch clause.

**Features**

- Browse all detected code smells and immediately go to the location of the individual code smells in the Java source code (which is colored) to start refactoring in the Eclipse Java code editor
- Markers are the Eclipse mechanism for resource annotations. We use markers to show the user what code smells were detected by the plug-in and where in the source code they can be found.
- Parichayana output report consists of the problematic code and the anti-pattern name and description.
- Preference page can be used to set preferences for the plug-in, like enabling or disabling detection of a certain type of code smells.
- Provides a per-project basis preference setting. A user can set plug-in preferences on a per-project basis, as opposed to a situation in which settings are made system-wide. Using regular expressions, one can set class and package-level code smell detection.

**References**
