PIE: A Tool for Visualizing the Lifecycle of Design Patterns in Open Source Software Projects

SEER★LAB | Ontario Tech UNIVERSITY

Kashif Hussain ● Christopher Collins ● Jeremy S. Bradbury
Software Engineering & Education Research Lab ● Faculty of Science ● Ontario Tech University ● Oshawa, ON, Canada
kashif.hussain1@ontariotechu.net, christopher.collins@ontariotechu.ca, jeremy.bradbury@ontariotechu.ca

1. Motivation

- A pattern *"...describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice"* [1].
- Design patterns are employed in source code to solve commonly occurring programming tasks using understood best practices.
- One challenge is that over the life of a project, these patterns can undergo unplanned changes as a side effect of maintenance tasks.
  - Can result in increased brittleness of the code
  - May only be detected when the code brittleness leads to a software bug.

2. Design Goals

- PIE was designed to allow for exploration of the following questions with respect to a specific project:
  - What design patterns have existed over the lifetime of the project?
  - When have design patterns been created, broken, and removed?
  - What code changes resulted in a design pattern breaking?
  - How have design patterns evolved, both in scope or into other patterns?

3. Architecture
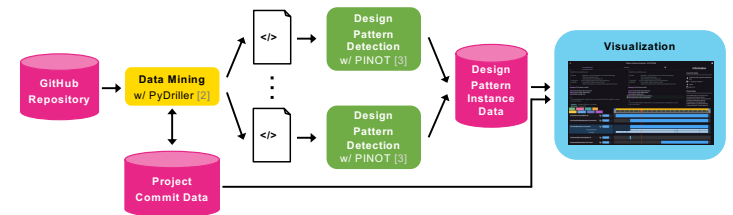
4. User Interface & Visualization

Timeline View (1)
- Shows a timeline for each design pattern instance which is colour-coded based on design pattern type
- Shows the commits where each pattern instance was detected in a time-line
- Expands to show file-level timelines which where vertical lines denote commits where a file has been modified.

Timeline Controller
- Commit numbers
- Red marker (●)
  - Used to select a specific commit to display in Code Viewer and Information Panel.
- Yellow slider with blue handles (▮▬▮)
  - Blue handles adjust the scale of the timeline, yellow slider moves the timeline position along the commit history

Pattern Break Annotations (∧)
- Indicates a pattern break is more likely to have occurred

Code Viewer (2)
- Displays relevant commit-level code changes of the currently selected pattern instance.
- Presents commit-level code changes as added (●) and deleted (●) code.
- Supports both an inline and split view to help users analyze the code differences.

Information Panel (3)
- Displays information relevant to the currently selected pattern instance
  - PINOT analysis data
  - Commit data (e.g., commit message)

Watch Our Demo Video!

[1] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, A Pattern Language. *Oxford University Press*, 1977.
[2] D. Spadini, M. Aniche, and A. Bacchelli, "PyDriller: Python framework for mining software repositories," in *Proc. of ESEC/FSE 2018*, pp. 908–911.
[3] N. Shi and R. A. Olsson, "Reverse engineering of design patterns from Java source code," in *Proc. of ASE'06*, pp. 123–134.