



PIE: Pattern Instance Explorer

**Kashif Hussain, Christopher Collins,
Jeremy S. Bradbury**

Software Engineering & Education Research Lab
Ontario Tech University, Oshawa, Canada

<http://www.seerlab.ca>

Motivation

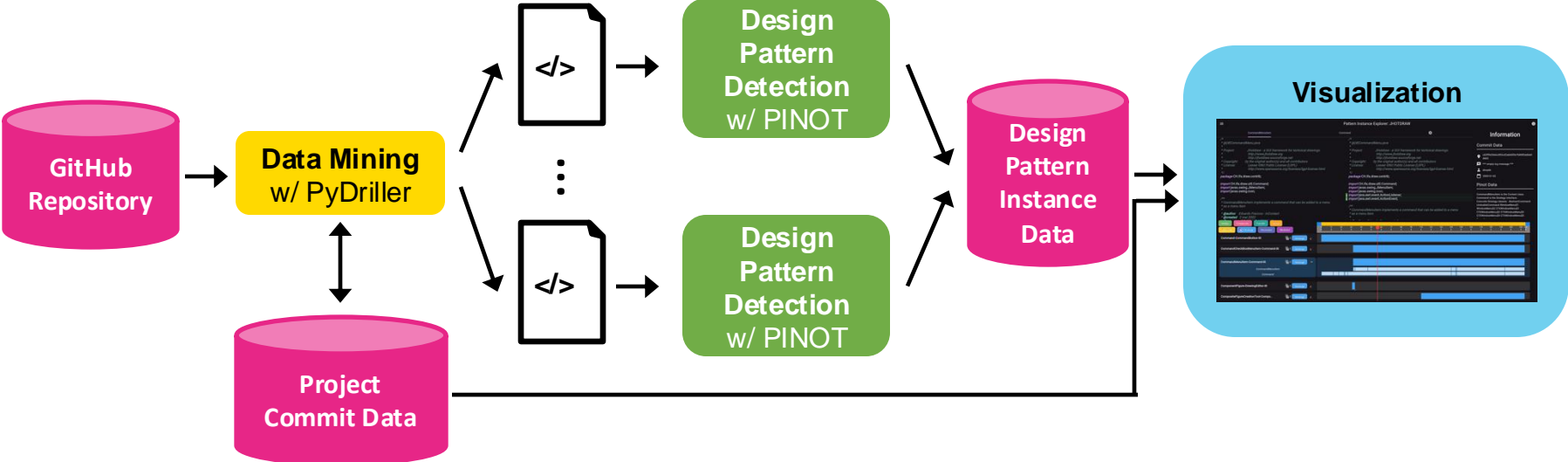
- A pattern *“...describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”* [1].
- Design patterns are employed in source code to solve commonly occurring programming tasks using understood best practices.
- 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 brittleness leads to a bug.

[1] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, A Pattern Language. *Oxford University Press*, 1977.

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?

Architecture



Pattern Instance Explorer: JHOTDRAW

CommandMenuItem
Command
⚙️

```

/*
 * @(#)CommandMenu.java
 *
 * Project:      JHotdraw - a GUI framework for technical drawings
 *              http://www.jhotdraw.org
 *              http://jhotdraw.sourceforge.net
 * Copyright:   by the original author(s) and all contributors
 * License:     Lesser GNU Public License (LGPL)
 *              http://www.opensource.org/licenses/lgpl-license.html
 */
package CH.ifa.draw.contrib;

import CH.ifa.draw.util.Command;
import javax.swing.JMenuitem;
import javax.swing.Icon;

/**
 * CommandMenuItem implements a command that can be added to a menu
 * as a menu item
 *
 * @author Eduardo Francos - InContext
 * @created 2 mai 2002
 */

```

```

/*
 * @(#)CommandMenu.java
 *
 * Project:      JHotdraw - a GUI framework for technical drawings
 *              http://www.jhotdraw.org
 *              http://jhotdraw.sourceforge.net
 * Copyright:   by the original author(s) and all contributors
 * License:     Lesser GNU Public License (LGPL)
 *              http://www.opensource.org/licenses/lgpl-license.html
 */
package CH.ifa.draw.contrib;

import CH.ifa.draw.util.Command;
import javax.swing.JMenuitem;
import javax.swing.Icon;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

/**
 * CommandMenuItem implements a command that can be added to a menu
 * as a menu item
 *
 */

```

Information

Commit Data

- 📍 c63fffe20e6c003cd2ab66f0e7b8083a8de0 deb3
- 💬 *** empty log message ***
- 👤 dnoyeb
- 📅 2003-01-02

Pinot Data

CommandMenuItem is the Context class.
 Command is the Strategy interface.
 Concrete Strategy classes: AbstractCommand
 UndoableCommand WindowMenu\$1
 WindowMenu\$2 CTXWindowMenu\$1
 CTXWindowMenu\$2 CTXWindowMenu\$3
 CTXWindowMenu\$4 CTXWindowMenu\$5

Bridge Composite Facade Proxy

Flyweight Strategy Decorator Mediator

Command-CommandButton-St Strategy <

CommandCheckBoxMenuItem-Command-St Strategy <

CommandMenuItem-Command-St Strategy ▾

CommandMenuItem
Command

ComponentFigure-DrawingEditor-St Strategy <

CompositeFigureCreationTool-Compo... Strategy <



PIE: Pattern Instance Explorer

**Kashif Hussain, Christopher Collins,
Jeremy S. Bradbury**

Software Engineering & Education Research Lab
Ontario Tech University, Oshawa, Canada

<http://www.seerlab.ca>