



# Open Source Project Analysis

## BlueJ

### สมาชิกกลุ่ม

61010910 ฤทธิกร เชาว์ชนพันธ์

61011215 อรุณ ด่านศักดิ์ชัย

62010358 ธนภณ เวชสุทธานนท์

62010758 ร่มธรรม ตั้งสุนันท์ธรรม

62010893 สุภณัฐ วันดี

(01076024) SOFTWARE ARCHITECTURE AND DESIGN

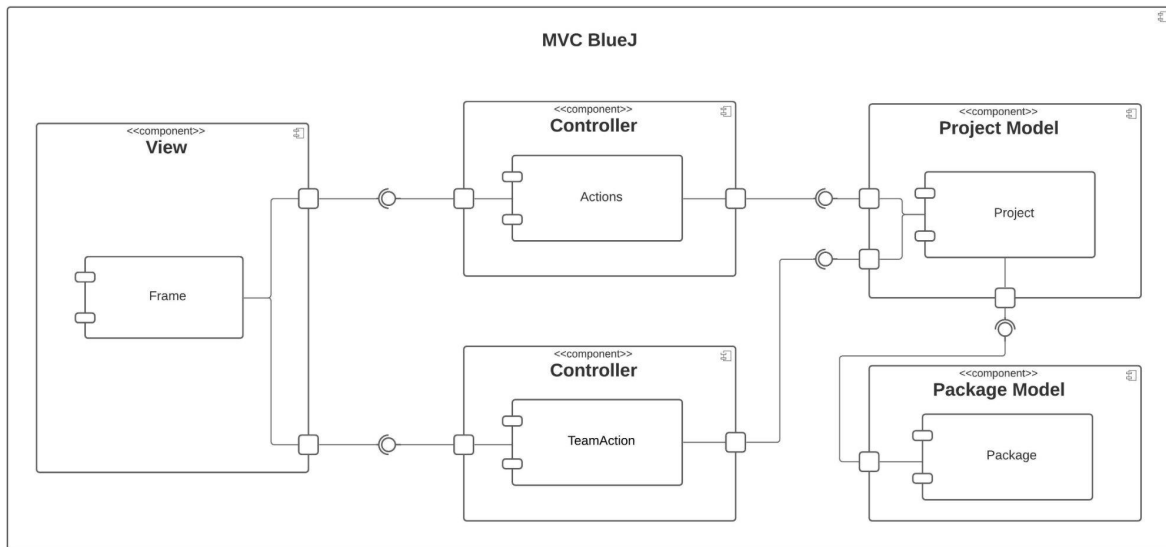
อาจารย์ผู้สอน ดร.ปริญญา เอกปริญญา

ภาคเรียนที่ 1 ปีการศึกษา 2564

## Software/System Architecture Analysis : MVC

Source : /doc/BlueJ-architecture-and-design.txt

Component Diagram ของ MVC ใน BlueJ



BlueJ มีการใช้ Architectural Style ในรูปแบบของ View-Model-Controller เรียกย่อ ๆ ว่า MVC ซึ่งอย่างที่ทราบกันดีว่า MVC จะมีการแบ่งโครงสร้างออกเป็น 3 ส่วนหลัก ๆ

1. Model เป็นส่วนของการเก็บข้อมูล
2. View เป็นการแสดงผลกับผู้ใช้ ซึ่งก็คือ User Interface หรือ UI
3. Controller เป็นส่วนของการประมวลผล โดยจะทำงานร่วมกับส่วนของ Model และส่วนของ View

โดย BlueJ จะมี Components หลักๆ อยู่ 3 ส่วน คือ Component ส่วนของ View, Component ส่วนของ Model ,และ Component ส่วนของ Controller

1. View Components จะมีส่วน Artifact ที่ชื่อ Frame ซึ่งในส่วนนี้จะมีฟังก์ชันต่างๆของส่วน UI อยู่ในนั้น โดยจะมีฟังก์ชันการทำงานสำคัญๆ เช่น ฟังก์ชันในการ Create Frame ฟังก์ชันในการ Remove Frame และฟังก์ชันอื่นๆอีกหลายอย่าง ที่เกี่ยวกับ UI

2. Controller Component จะทำงานร่วมกับ Component ส่วนของ View และ Model โดยจะมี Component Controller อยู่ 2 ส่วน คือ

2.1. Action จะเป็นส่วนที่ควบคุมเกี่ยวกับตัว Editor เช่น การ Create Project หรือ Package, การ Save Project หรือแม้กระทั่ง การทำงานกับส่วนของ Java Debugger เป็นต้น

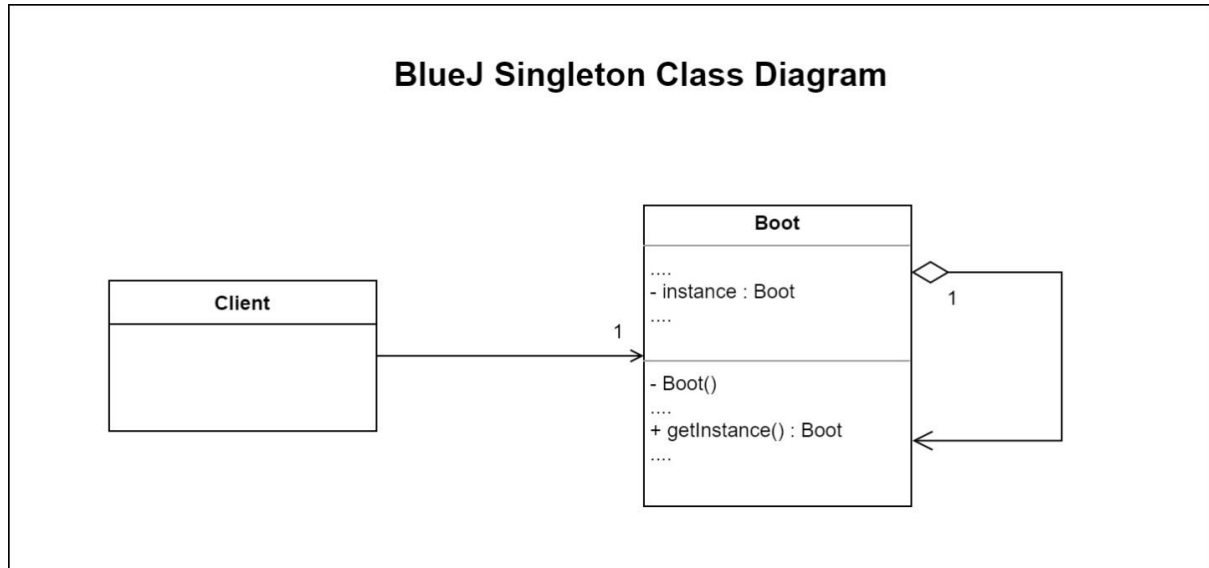
2.2. Team Action จะเป็นส่วนที่ควบคุมการทำงานกับ Source Code Hosting ต่างๆ อย่างเช่น Github โดยใน Team Action มีฟังก์ชันที่หลากหลาย เช่น Commit หรือ Push เป็นต้น

3. Model Component จะมี Artifact 2 ตัวทำงานร่วมกัน คือ ส่วนของ Project กับส่วนของ Package โดยที่ส่วนของ Project จะเป็นส่วนที่เกี่ยวกับตัว Project File ของ BlueJ โดยจะมีฟังก์ชันการทำงานที่สำคัญๆ เช่น ฟังก์ชัน CreateNewProject ฟังก์ชัน getFilesInProject เป็นต้น และส่วนของ Package จะเกี่ยวกับการสร้าง Package ขึ้นมา จากตัว Project File ที่มีอยู่แล้ว

โดยส่วนของ Project จะเป็นส่วนที่ทำงานร่วมกับ Controller โดยจะมี Package คอยให้ข้อมูลกับส่วนของ Project ดังที่แสดงใน Diagram

## Design Pattern Analysis

### 1. Singleton



Singleton เป็น Design Pattern สำหรับการควบคุมจำนวน Instance() ของ class ซึ่งจะเจอได้ในไฟล์ `main.java` และ `boot.java`

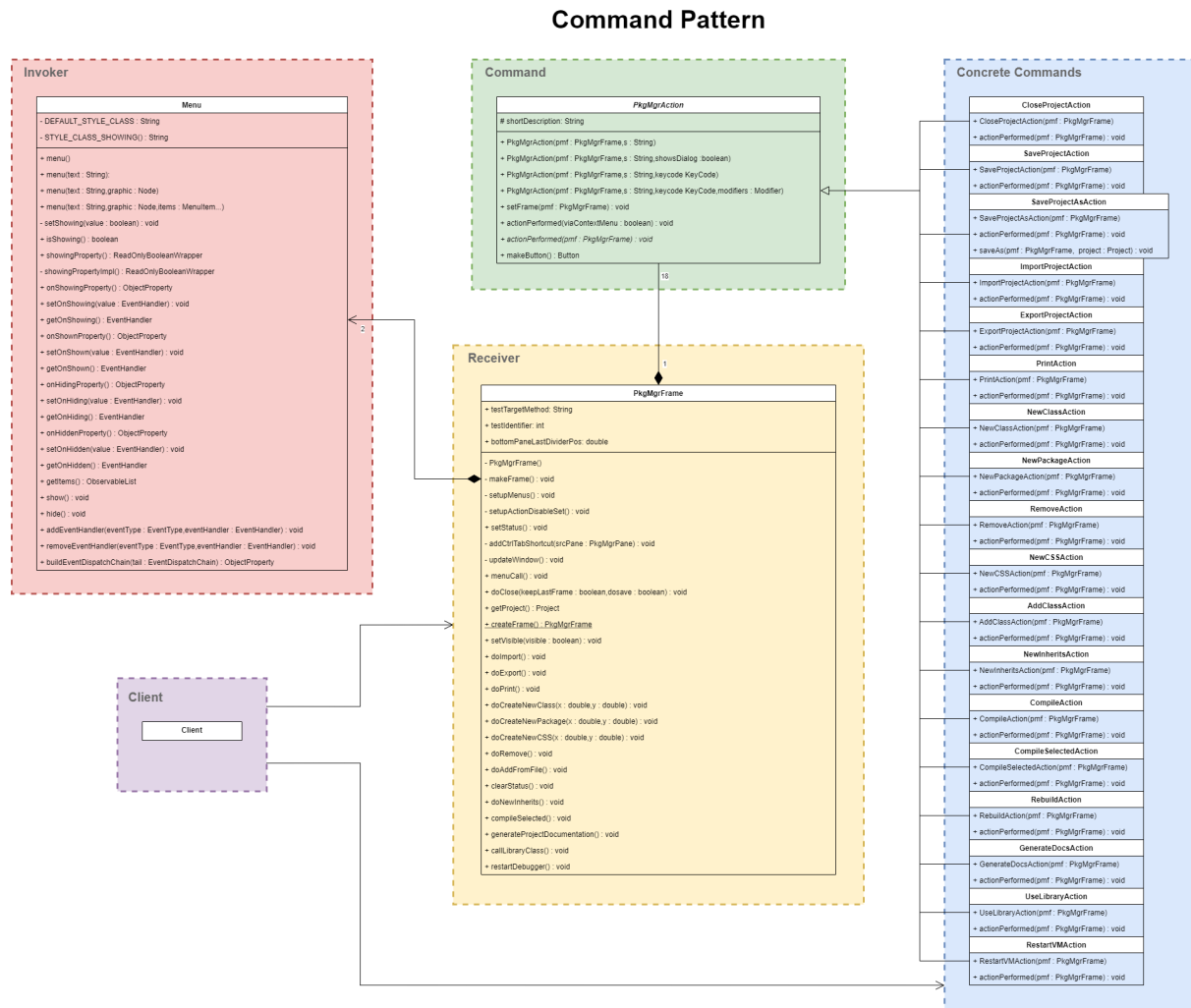
โดย `main.java` จะเจอได้ใน Constructor `main` ซึ่งจะเป็นการสร้าง Object `Boot` ซึ่งจะทำการเรียก method `getInstance()` ซึ่งอยู่ในไฟล์ `Boot.java`

method `getInstance()` จะเป็นการ return ตัวแปร `instance` ซึ่งเป็น Object ของ Class `Boot`

path: `/boot/src/bluej/Boot.java`

Line: 338 - 341

## 2. Command



แนวคิดของ Command Pattern คือให้เปลี่ยน Action ของผู้กระทำมาเป็น Object และถ้าต้องมีข้อมูลอะไรมาเกี่ยวข้อง ก็ให้หัดเข้ามาใน Object ตัวนั้นด้วยเลย ซึ่ง Object พวกนั้นเราจะเรียกมันว่า Command

ตัว Command object นี้จะเป็นเหมือนตัวเชื่อมระหว่าง UI layer กับ Business logic layer

BlueJ จะใช้ Command Pattern ในส่วนของปุ่ม (พวก menubar และ Toolbar)

โดย Invoker ของ BlueJ คือ class menu.java โดย class นี้จะเป็นตัวเก็บ Object ของ Concrete Commands ต่าง ๆ ไว้(เมื่อ User กดปุ่ม จะ Invoke ในส่วน onAction เป็นส่วนที่สืบทอดมาจาก class MenuItem.java) เมื่อ Client กดปุ่มใน UI มันจะไป Invoke ในส่วน onAction ของ menu ให้คำสั่งที่ Client กดนั้นมันทำงาน และตัว Concrete Commands จะส่งงานที่ Client ต้องการให้ Receiver หรือก็คือ class PkgMgrFrame ทำงานต่อไป

## Overview

- 1.Invoker(menu.java) - เป็นตัวที่เก็บ Command ต่าง ๆ ที่มีไว้ให้ Client เรียกเมื่อใช้งาน
- 2.Command(PkgMgrAction.java) - เป็นคลาสแม่ที่มี Abstract method เป็นแม่แบบให้คลาสลูกไป Override ( actionPerformed() )
- 3.Concrete Commands - เป็นตัวเก็บข้อมูลของงานที่จะทำ และรู้ว่า Receiver ตัวไหนจะมารับงานต่อ
- 4.Receiver(PkfMgrFrame.java) - เป็นตัวที่ทำงานจริง ๆ (Business Logic จะอยู่ที่นี้)
- 5.Client - ผู้ใช้ของระบบ

## Invoker

path : Javafx default library(Menu.java)

Line: 429 - 431

## Command

path: /boot/src/bluej/pkgmgr/actions

Method actionPerformed()

## Concrete Commands

path : /boot/src/bluej/pkgmgr/actions/PkgMgrAction.java

Line: 106

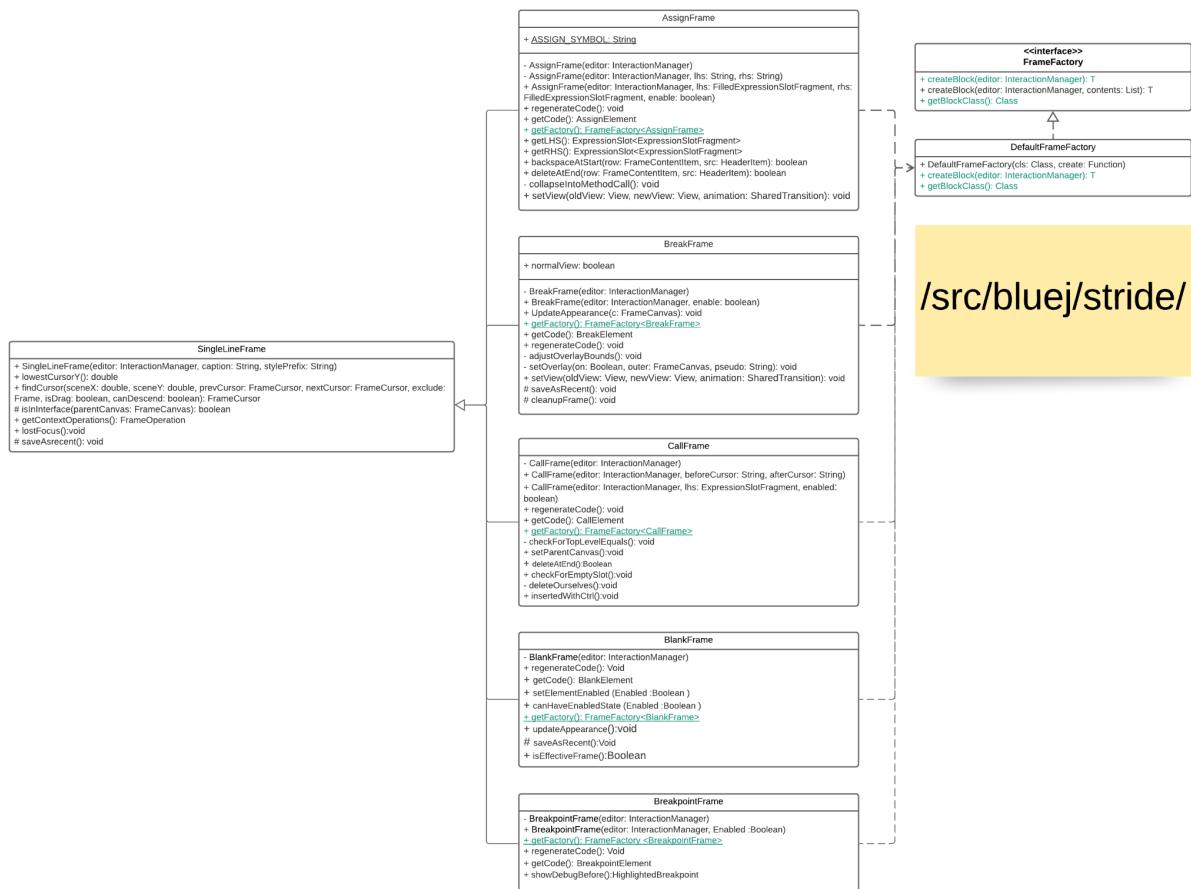
## Receiver

path : /boot/src/bluej/pkgmgr/PkgMgrFrame.java

Line: 552-3355

### 3. Abstract Factory

BlueJ Abstract Factory Class Diagram



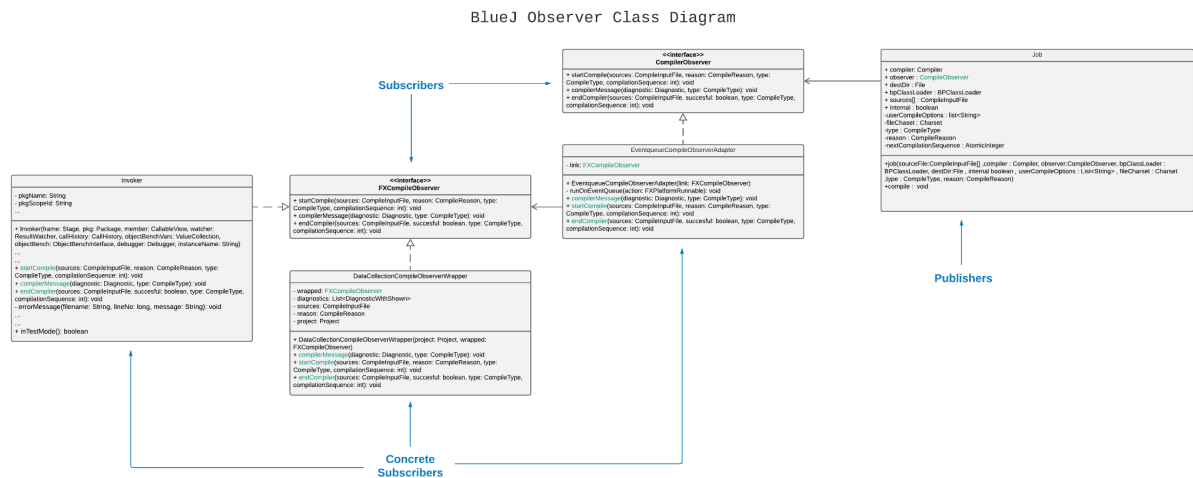
เป้าหมายหลักของ Abstract Factory คือการสร้างกลุ่มของ Object โดยไม่ต้องกำหนด concrete class ซึ่งในที่นี้ตัวโรงงาน คือ FrameFactory เป็น interface ถูกนำไป implement ต่อด้วย class DefaultFrameFactory สิ่งที่น่าสนใจก็คือ ในบรรทัดที่ 32 นั้นผู้เขียน BlueJ ได้อธิบายสั้น ๆ ว่า เมื่อไหร่ที่มีการเรียกใช้

DefaultFrameFactory จะสังเกตจาก method getFactory() ซึ่งผู้เรียนได้ค้นพบว่าไม่ต่ำกว่า 10 class ที่มีการเรียกใช้ ซึ่งถือว่าเป็นจำนวนมาก ไม่เพียงเท่านั้น แต่ทุก class ที่เรียกใช้ ต่างก็ถูก extends ด้วย class เดียวกัน นั่นคือ SingleLineFrame จากที่กล่าวมาสามารถอนุมานได้ว่า เรา กำลังรับมือกับ Abstract Factory อยู่นั่นเอง

path: /src/bluej/stride/generic/DefaultFrameFactory.java

Line:31-38

## 4. Observer



มีจุดมุ่งหมาย ในการสร้างตัวกลางในการรับข้อมูลอัปเดต

โดยใน BlueJ เราให้เหตุผลว่า job เป็น Publisher ส่วน Subscribers ได้แก่ CompilerObserver และ FXCompilerObserver ที่เป็น interface ที่จะเป็นตัวกลาง รับข่าวสารการอัปเดต จาก job ไปให้ Concrete Subscribers คือ EventqueueCompileObserverAdapter, DataCollectionCompileObserverWrapper และ invoker ที่มี method compilerMessenger(), startCompile() และ EndCompile() เป็นจุดร่วม

path: /src/bluej/debugmgr/Invoker.java

Line:113-128



## Quality Attribute

### 1. Extensibility

ตัวโปรแกรม BlueJ สามารถติดตั้งส่วนขยายหลายอย่าง que เพิ่มฟังก์ชันพื้นฐานของตัวโปรแกรม โดยที่ใช้วิธีการเพิ่ม Extension หรือ third parties extensions ต่างๆ ในโปรแกรม ผ่านการ install jar file

### 2. Portability

โปรแกรม BlueJ สามารถทำงานบน Windows, Mac OS X, Linux และแพลตฟอร์มอื่นๆ ที่ใช้ Java ได้ครับ ซึ่ง BlueJ ถูกเขียนโดย java programming และใช้การรันบน Java Virtual machine ทำให้ BlueJ สามารถรันได้ในทุกๆ java platform

### 3. Usability

ตัว Editor ของ BlueJ จะมี scope highlight ในแต่ละส่วนต่างกัน โดยที่พื้นหลังของบล็อกโค้ดแต่ละอัน จะถูก highlight เพื่อให้สามารถค้นหาโค้ดได้รวดเร็วยิ่งขึ้น อีกทั้ง BlueJ สามารถเรียกใช้โค้ด Java ได้โดยตรงโดยไม่ต้องคอมไพล์ เพียงแค่พิมพ์โค้ด ผลลัพธ์จะถูกแสดงผลออกมา โดยจะเป็นการ Support การทำงานของผู้ใช้โดยตรง

## Weakness : Reusability

ตัว Project ของ BlueJ ไม่สามารถไปเปิดใน IDE ที่รองรับภาษา Java อื่นๆ ได้ เช่น Eclipse Netbeans IntelliJ เป็นต้น