



รายงาน

Open Source Project Analysis

โดย

นาย ชญานิน บัวสละ รหัส 62015021

นาย ณัฐภูมิ ล้อมเขต รหัส 62015038

นาย ทรงวุฒิ นักร้อง รหัส 62015044

นาย ธนาธช อัศวรรตนากร รหัส 62015048

นาย เอกสรรค์ ทองอ่อน รหัส 62015123

นาย อรรถพล ตันติวัฒนาผล รหัส 62015131

อาจารย์ผู้สอน ดร. ปริญญา เอกปริญญา

รหัสวิชา 01076024 Software Architecture and Design

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคการศึกษาที่ 1 ปีการศึกษา 2564

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรหัสวิชา 01076024 Software Architecture and Design มีจุดประสงค์เพื่อ ศึกษาหาความรู้เกี่ยวกับ วิเคราะห์ Software Architecture, Quality Attribute, Weakness, Design patterns analysis ของตัวโปรแกรม Open Source ชื่อ PDFsam ว่าโครงสร้างภายในโปรแกรมมีโครงสร้างเป็นอย่างไร เพื่อให้เกิดความเข้าใจมากยิ่งขึ้นในวิชาที่กำลังศึกษา

ทั้งนี้เนื้อหา และขั้นตอนการวิเคราะห์ ได้มีการศึกษาจากในห้องเรียน และอินเทอร์เน็ต และสอบถามจากอาจารย์ผู้สอนโดยตรง ขอบพระคุณอาจารย์ผู้สอน เป็นอย่างสูง ที่ท่านให้คำแนะนำเพื่อแก้ไขให้ข้อเสนอแนะตลอดการทำงาน ผู้จัดทำรายงานหวังว่ารายงานฉบับนี้จะมีประโยชน์ต่อผู้ที่นำไปเกิดผลลัพธ์ตามความคาดหวัง

นายชุมานิน บัวسلح

นายณัฐภูมิ ล้อมเขต

นายทรงวุฒิ นักร้อง

นายธนชัย อัศวัตนากร

นายเสกสรรค์ ทองอ่อน

นายอรรถพล ตันติวัฒนาผล

ผู้จัดทำ

สารบัญ

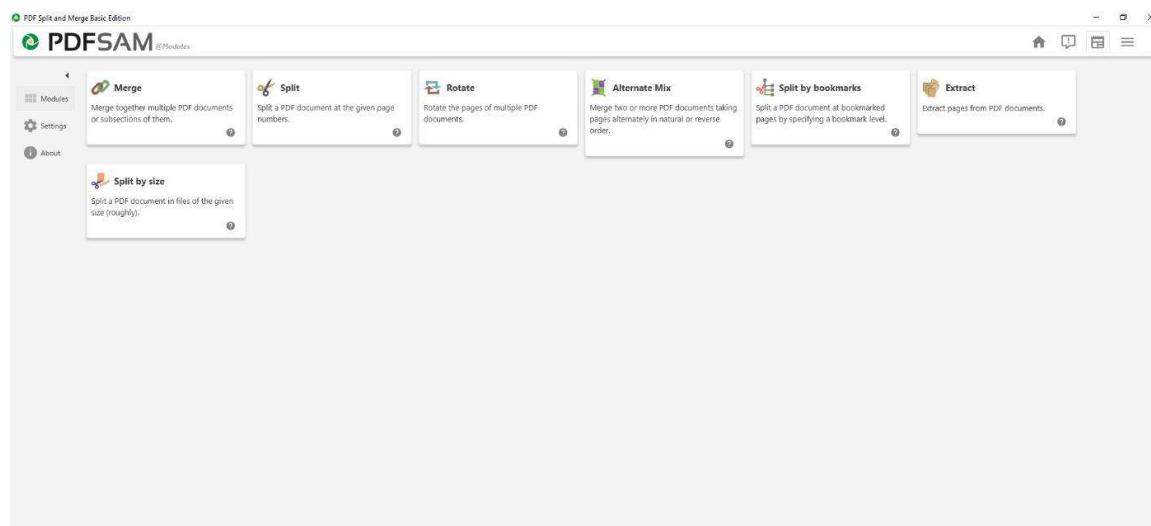
What is PDFSam	4
Software Architecture Analysis	5
1. N-Tier	5
1. Reliability (Availability)	7
2. Portability	8
3. Usability	9
4. Security.....	11
5. Modifiability.....	12
5.1 Modifiability.....	13
6. Reusability.....	14
Weakness	15
Design Pattern Analysis.....	16
1. Singleton	16
2. Builder.....	18
Source code ที่เกี่ยวข้องกับ Builder Design Pattern	19
3. Observer (Event Listeners)	21
Source code ที่เกี่ยวข้องกับ Observer Design Pattern	22
4. Dependency Injection.....	25
Source code ที่เกี่ยวข้องกับ Dependency Injection Design Pattern.....	26
5. Call back	28
Source code ที่เกี่ยวข้องกับ Call back Design Pattern.....	28

What is PDFSam

PDFSam เป็นโปรแกรมสำหรับ “PDF Split and Merge” ซึ่งเป็นโปรแกรมที่สำหรับการแก้ไขไฟล์ PDF ได้อย่างรวดเร็ว สามารถรวมหลายเอกสาร PDF ให้เป็นเอกสาร PDF เดียวได้, แยกไฟล์ PDF เป็นหลายไฟล์, แยกหน้าจากไฟล์ PDF, เรียงลำดับหน้าใหม่ในไฟล์ หรือ สร้างไฟล์ PDF ใหม่ที่มีหน้าเอกสารมาจากหลายไฟล์ได้

โดยการวิเคราะห์ Source code ชุดคำสั่งของ PDFSAM ใน Assignment นี้จะดำเนินการวิเคราะห์ และอ้างอิงจากชุดคำสั่งของ version 4.2.8 เป็นหลัก ซึ่งสามารถเข้าถึงได้ดังนี้

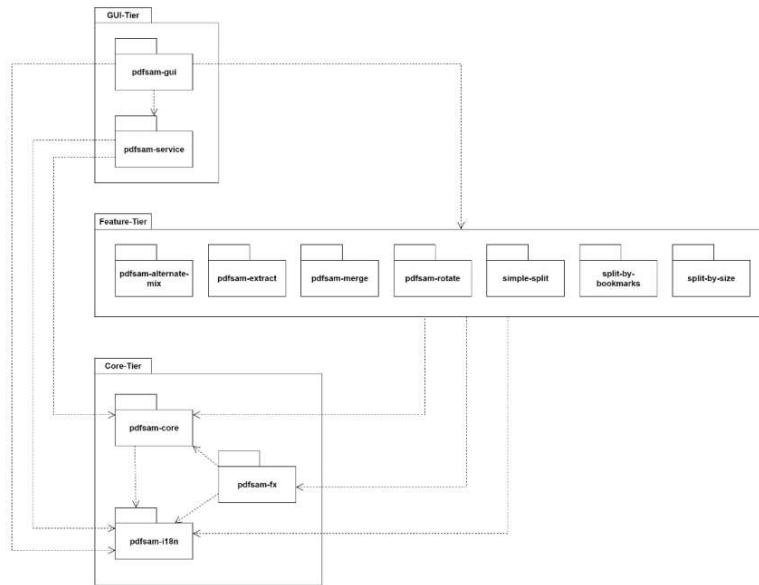
<https://github.com/torakiki/pdfsam/tree/v4.2.8>



Software Architecture Analysis

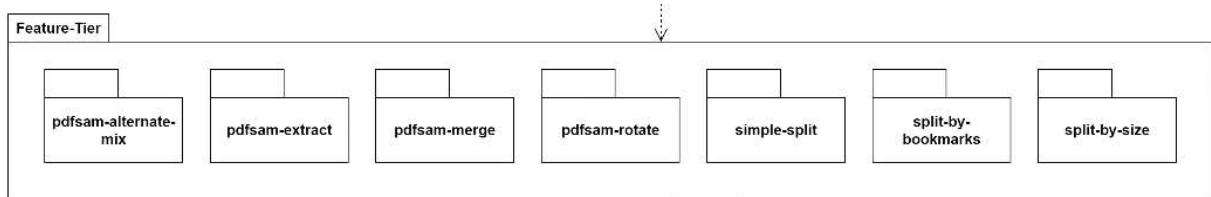
1. N-Tier

PDFSam มีการใช้ N-Tier Architecture มาใช้ จากรูป UML Package Diagram จะเห็นได้ว่ามีทั้งหมด 3 tier เริ่มจาก GUI Tier ที่มี package pdfsam-gui และ pdfsam-service โดยใน tier นี้จะจัดการในเรื่องของ graphic user interface ของโปรแกรม PDFSAM



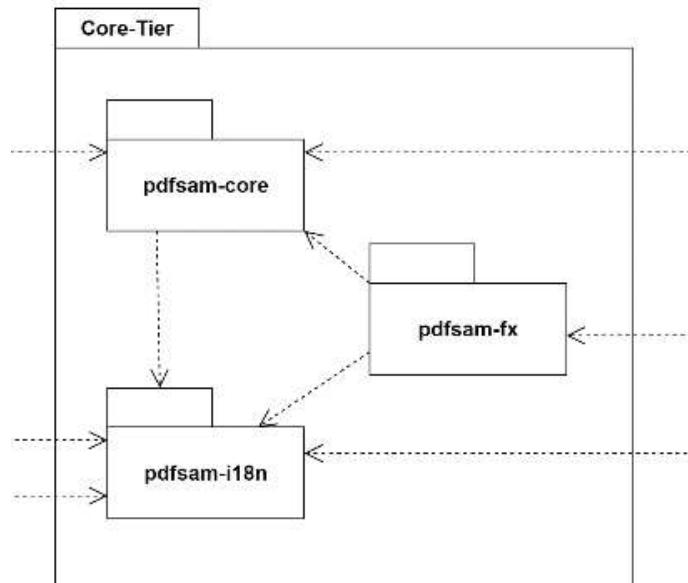
UML Package Diagram ของ N-Tier Architecture ของโปรแกรม PDFSAM

ต่อมาเป็น Feature-Tier โปรแกรม PDFSAM นั้นมี features หลักๆ ทั้งหมด 7 features โดยแต่ละ feature ก็ถูกแบ่งออกเป็น package ซึ่งใน tier นี้ ทุก feature จะมีการเรียกใช้ package อื่นที่เหมือนกัน คือจะเริ่มจากที่ pdfsam-gui จะมาเรียกใช้ feature ต่างๆ และ feature ต่างๆ ก็จะไปเรียกใช้ core-tier เพื่อสนับสนุน



UML Package Diagram ของ N-Tier Architecture ของโปรแกรม PDFSAM

สุดท้ายเป็น Core-tier ซึ่งใน tier นี้จะประกอบไปด้วย package หลักๆ ทั้งหมด 3 package คือ pdfsam-core เป็น package ที่ใช้จัดการการทำงานหลักๆ ของ pdfsam, pdfsam-fx เป็น package ที่ใช้จัดการพัฟฟ์ชั่นต่างๆ และ pdfsam-i18n เป็น package ที่ใช้การเรื่องของการใช้งานหลายภาษาภาษาที่ GUI-tier จะนำไปใช้ในการแสดงผล



UML Package Diagram ของ N-Tier Architecture ของโปรแกรม PDFSAM

Quality attribute Analysis

1. Reliability (Availability)

1. เหตุผลที่มี QUALITY ATTRIBUTE นี้

- PDFsam Basic เป็นซอฟต์แวร์โอเพ่นซอร์สฟรีสำหรับหลายแพลตฟอร์มที่ออกแบบมาเพื่อทำงานแบบ stand alone
 - PDF merge
 - PDF split
 - PDF mix
 - Rotate PDF
 - Extract pages

2. TACTICS ที่ใช้เพื่อให้ได้ QUALITY ATTRIBUTE นี้

- เพราะถ้าเห็นการขาดการเชื่อมต่ออินเทอร์เน็ต ตัวโปรแกรมก็สามารถทำงานตามฟังก์ชันปกติได้
- เพื่อรองรับการอัปเดตเวอร์ชันใหม่ๆ หรือแก้ไขข้อบกพร่อง

3. หลักฐานสนับสนุนเกี่ยวกับ QUALITY ATTRIBUTE นี้

System requirements?

Windows Vista 32 & 64, Windows 7 32 & 64, Windows 8 32 & 64, Windows 10 32 & 64. 512 MB (1 GB for Vista, 7, 8 & 10) or RAM. 200MB or more of disk space. Internet connection is required for version updates. Microsoft Office is required for plug-in use and conversion to Word

2. Portability

1. เหตุผลที่มี QUALITY ATTRIBUTE นี้
 - การที่โปรแกรมมี Portability สูงจะช่วยเพิ่มฐานผู้ใช้งานที่มากจากหลาย Platform
2. TACTICS ที่ใช้เพื่อให้ได้ QUALITY ATTRIBUTE นี้
 - มีการเลือกใช้การพัฒนาด้วยภาษา Java ที่ทำให้โปรแกรมสามารถใช้งาน cross platform ได้หลากหลาย
 - PDF SAM สามารถใช้งานได้หลาย platform เนื่องจากผู้พัฒนามีการพัฒนาบน java ซึ่ง platform ไหนที่รองรับ java ตัวโปรแกรมก็จะสามารถทำงานได้บน platform นั้น ซึ่งทั้ง windows, linux, mac นั้นก็รองรับ java
3. หลักฐานสนับสนุนเกี่ยวกับ QUALITY ATTRIBUTE นี้
 - ในเว็บ office PDF Sam มีให้โหลดสำหรับแต่ละ OS
 - ในเอกสาร FAQ ทางผู้พัฒนาได้ออกมาตอบคำถามเรื่องว่าสามารถลง PDF Sam ได้หลาย OS

The screenshot shows the PDFSAM website's download page. At the top, it says "Download PDFsam Basic v4.2.7". Below that are social sharing buttons for Email, Twitter, Facebook, WhatsApp, Pocket, and GitHub. A prominent button labeled "Windows downloader (.exe)" is shown, with a note: "Download and install PDFsam Basic on Windows systems. It offers to try PDFsam Enhanced with two free modules to view and create PDF files". Below this are sections for "Mac OS" (with links to "Apple disk images (.dmg)" and "Portable archive (.zip)"), "Linux" (with links to "Debian packages for Debian based Linux distributions (.deb)" and "Portable archive (.tar.gz)"), "Windows" (with links to "Windows installer (.msi)" and "Portable archive (.zip)"), "Command line" (with a link to "Command line interface (.jar)"), and "All the good stuff" (with a link to "Source code and packages for all the supported platforms (.ZIP)"). At the bottom left, there is a section titled "Multiplatform?" with the text "Yes, We provide installers for Windows, Linux and MacOS".

ที่มา: <https://pdfsam.org/download-pdfsam-basic/>

3. Usability

1. เหตุผลที่มี QUALITY ATTRIBUTE นี้

- การที่โปรแกรมมี **Usability** สูงทำให้เป็นมิตรต่อผู้ใช้งาน มี interface ที่ใช้งานง่าย
- End user ต้องการ merge pdf file หรือ split pdf file สามารถเลือกหน้าได้หน้าหนึ่งในไฟล์หรือเลือกบางหน้าที่ต้องการได้ จึงมีความ flexible สูง ทำให้ใช้งานได้ง่ายและ export ออกมาก็ได้รวดเร็ว

2. TACTICS ที่ใช้เพื่อให้ได้ QUALITY ATTRIBUTE นี้

- มีการใช้กลวิธี aggregate ในการออกแบบการใช้งาน คือ สามารถเลือกหน้าได้ก็ได้ภายใน pdf เช่น หน้า 10-17 และหน้า 20 ก็ได้ของแต่ละไฟล์ pdf ที่ต้องการ merge เข้าด้วยกัน สามารถเพิ่มหน้าว่างขั้นๆหน้าได้
- สามารถเพิ่มหน้าว่างขั้นๆหน้าได้
- สามารถแทรกหน้า footer ได้
- สามารถ normalize ให้ทุกหน้าปรับ page size เท่ากับหน้าแรก
- มีการจัดการตาม bookmark
- มีการจัดการในรูปแบบ form
- มีการแปลงให้เป็นรูปแบบตาราง
- มีการแนบ Hint User guide ไว้ในแต่ละ feature ของตัวโปรแกรม เพื่อให้ผู้ใช้สามารถใช้งานได้ง่าย

3. หลักฐานสนับสนุนเกี่ยวกับ QUALITY ATTRIBUTE นี้

การ Merge

Short guide on how to combine PDF files with PDFsam Basic

Select PDF files to merge

You can merge PDF files or part of them. Drag and drop the PDF files you want to combine or click **Add** to add files. Use the buttons **Move Up** and **Move Down** or right click to arrange the files order. Encrypted PDF files show a icon on their left, click on the lock icon and supply the unlock password.

If you only want to merge some pages, double click on the **Page ranges** cell and specify the pages or page ranges you want to include. Pages can be specified in the form of comma separated page ranges like 2-13,17 if you want pages from 2 to 13 and page 17. You can use the syntax 20- if you want pages from page 20 to the end of the file.

Add	Clear	Remove	Move Up	Move Down
Name	Size	Pages	Modified	Page ranges
Open-Advice.pdf	2 MB	310	09 June 2013 17:35:39	
document.pdf	8 KB	1	04 June 2015 09:38:45	
PDF32000_2008.pdf	8 MB	756	30 December 2014 18:...	20-152,300,700-

Merge settings

Merge settings

- Add a blank page if page number is odd ⊕
- Add a footer ⊕
- Normalise page size ⊕

Interactive forms (AcroForms): Merge fields ⊕

Bookmarks handling: Retain bookmarks ⊕

Table of contents: Don't generate ⊕

• *Add a blank page if page number is odd:* PDFsam Basic will add a blank page after every merged PDF whose page number is odd
 • *Add footer:* PDFsam Basic will add a page footer with the name of the file the page belonged to
 • *Normalise page size:* PDFsam Basic will resize all pages to have the same width of the first page
 • *Interactive forms:* What to do in case of some of the PDF files contain an AcroForm. Discard the forms if they are not needed, merge them together allowing fields with the same name or merge them renaming fields with the same name to avoid names clashes
 • *Bookmarks:* What to do in case of some of the PDF files contain bookmarks

- Retain:* Merge together bookmarks trees of the input PDF files
- Discard:* Bookmarks are discarded
- Create one entry for each merged document:* A new bookmarks tree containing one entry for each merged PDF file is created. Each entry pointing to the first page of the merged file
- Retain bookmarks as one entry for each document:* A new bookmarks tree containing one entry for each merged PDF files is created. Each entry will contain the whole bookmarks tree of the merged file

 • *Table of contents:* PDFsam Basic will add a table of contents to the beginning of the resulting PDF file

Reference: [Merge document - PDFsam](#)

การ Split

Short guide on how to divide PDF files with PDFsam Basic

Select a PDF file

Drag and drop the PDF file you want to split or click [Browse](#) and select the file. Encrypted PDF files show a  icon on their left, click on the lock icon and supply the unlock password.

/home/torakki/Scaricati/PDF32000_2008.pdf [Browse](#)

Pages: 756, PDF Version: 1.6

Split by page numbers

Using the split by page number you can tell PDFsam Basic where you want to divide the PDF file

Split settings

- Split after Every page
- Split after the following page numbers Page numbers to split at (n1,n2,n3...)
- Split by every "n" pages Number of pages

• *Split after:*

- *Every page:* Creates PDF files containing one page each
- *Even pages:* Creates PDF files containing two page each. The split will occur after page 2, 4, 6 etc
- *Odd pages:* Creates PDF files containing two page each. The split will occur after page 1, 3, 5 etc

• *Split after the following page numbers:* Set a comma separated list of page numbers after which you want to split

• *Split by every "n" page:* PDFsam Basic will split every *n* pages

เว็บไซต์หลัก: [Split PDF | Free and open source - PDFsam](#)

4. Security

1. เหตุผลที่มี QUALITY ATTRIBUTE นี้

- การที่โปรแกรมมี **Security** สูง เพื่อรักษาความเป็นส่วนตัวในขณะที่โปรแกรมทำงาน ไม่จำเป็นต้องเชื่อมต่ออินเทอร์เน็ตระหว่างการใช้งาน ทำให้ข้อมูลในไฟล์ PDF มีความเป็นส่วนตัว และทำให้ข้อมูลปลอดภัย

2. TACTICS ที่ใช้เพื่อให้ได้ QUALITY ATTRIBUTE นี้

- ในขณะที่ปิดการเชื่อมต่ออินเทอร์เน็ตก็สามารถนำไฟล์ PDF มาทำตามพังก์ชั่นที่มีในโปรแกรมได้
- ถึงระบบจะออนไลน์ หรือ ระบบจะออฟไลน์ เชื่อต่ออินเทอร์เน็ตหรือไม่ ก็ยังสามารถทำตามตัวเลือกเมนูต่างๆได้ขณะไม่มีอินเทอร์เน็ต

3. หลักฐานสนับสนุนเกี่ยวกับ Quality Attribute นี้

- ในขณะที่ใช้งานโปรแกรม ไม่มีการอัปโหลดไฟล์ขึ้นไปประมวลผลบน third party service เพราะทาง PDFsam รับรองว่าไม่ต้องอัปไฟล์ไปยัง third party service



- เว็บไซต์หลักอย่างเป็นทางการ : [Privacy Policy - PDFsam](#)

5. Modifiability

1. ให้เหตุผลสนับสนุนว่าทำไม่ผิดเงื่อนไขพัฒนาจึงเลือก Quality Attributes ดังกล่าว
 - การที่ software มี modifiability ที่สูงจะช่วยให้ developer แก้ไข feature ต่างๆ ทำได้容易 ลดความซับซ้อนในการจัดการโค้ดได้
2. ระบุและอธิบายว่า tactics ใดที่ใช้เพื่อให้ได้ Quality Attributes ที่ต้องการ
 - มีการใช้ tactics การ **split module** กับ feature ต่างๆ เช่น alternate-mix, extract, merge, rotate, split โดยการแยก feature ออกเป็น module ต่างๆ
3. หลักฐานสนับสนุนเกี่ยวกับ QUALITY ATTRIBUTE นี้

Merge module

File	Description	Date
MergeModule.java	repackaged eventstudio	2 years ago
MergeOptionsPane.java	Added Flatten to the AcroForm policies in Merge task	4 years ago
MergeParametersBuilder.java	upgraded Sejda to ver 4	2 years ago
MergeSelectionPane.java	Copyright notice, hopefully we got it right this time	4 years ago

Rotate module

File	Description	Date
RotateModule.java	repackaged eventstudio	2 years ago
RotateOptionsPane.java	Copyright notice, hopefully we got it right this time	4 years ago
RotateParametersBuilder.java	upgraded Sejda to ver 4	2 years ago
RotateSelectionPane.java	Copyright notice, hopefully we got it right this time	4 years ago

Split module

File	Description	Date
SplitAfterPredefinedSetOfPagesRadioButton.java	close #337: added a Clear all settings to single input modules	3 years ago
SplitAfterRadioButton.java	close #337: added a Clear all settings to single input modules	3 years ago
SplitByEveryRadioButton.java	close #337: added a Clear all settings to single input modules	3 years ago
SplitModule.java	repackaged eventstudio	2 years ago
SplitOptionsPane.java	close #337: added a Clear all settings to single input modules	3 years ago
SplitParametersBuilderCreator.java	Copyright notice, hopefully we got it right this time	4 years ago

5.1 Modifiability

1. เหตุผลที่มี QUALITY ATTRIBUTE นี้
 - การที่ software มี modifiability สูงในเรื่องของโมดูลภาษา จะทำให้โปรแกรมรองรับหลายภาษาและเพิ่มฐานผู้ใช้งาน
2. TACTICS ที่ใช้เพื่อให้ได้ QUALITY ATTRIBUTE นี้
 - มีการเปิดชุดภาษาให้บุคคลทั่วไปแปลเพื่อให้โปรแกรมรองรับหลายภาษาและนำเข้าไปในโปรแกรมผ่านชุดโมดูล
3. หลักฐานสนับสนุนเกี่ยวกับ QUALITY ATTRIBUTE นี้

The screenshot shows the 'Translations' tab of the Launchpad.net interface for the 'PDF Split and Merge' project. It includes sections for Permissions, Administration, Sharing Information, and Automatic synchronization. The 'Translation status by language' section features a chart for the Thai language, indicating high completion. A table below shows file counts for untranslated and needs review categories.

Category	Count
Untranslated	137
Needs review	0

[Series pdfsam-v3 : Translations : PDF Split and Merge \(launchpad.net\)](#)

```
/**
 * Default implementation of the {@link I18nContext}.
 *
 * @author Andrea Vacondio
 */
public final class DefaultI18nContext implements I18nContext {

    private static final Logger LOG = LoggerFactory.getLogger(DefaultI18nContext.class);

    private final Set<Locale> supportedLocales = Set.of(new Locale("af"), new Locale("eu"), new Locale("bs"),
        new Locale("pt", "BR"), Locale.SIMPLIFIED_CHINESE, Locale.TRADITIONAL_CHINESE, new Locale("co"),
        new Locale("hr"), new Locale("cs"), new Locale("da"), new Locale("nl"), Locale.UK, Locale.FRENCH,
        Locale.GERMAN, new Locale("he"), new Locale("hu"), new Locale("el"), Locale.JAPANESE, Locale.ITALIAN,
        new Locale("pl"), new Locale("pt"), new Locale("ro"), new Locale("ru"), new Locale("sk"), new Locale("sl"),
        new Locale("sr"), new Locale("sv"), new Locale("es"), new Locale("tr"), new Locale("uk"), new Locale("fi"),
        new Locale("ko"));

    private I18n i18n;

    DefaultI18nContext() {
        eventStudio().addAnnotatedListeners(this);
    }
}
```

6. Reusability

1. เหตุผลที่มี QUALITY ATTRIBUTE นี้

- การที่ software มี Reusability สูงในส่วนของการนำ Method จาก class อื่นมาใช้ช้าในการทำงานของโปรแกรม ทำให้การออกแบบและพัฒนาชุดคำสั่งเป็นระบบและง่ายขึ้น

2. TACTICS ที่ใช้เพื่อให้ได้ QUALITY ATTRIBUTE นี้

- ภายใน class ของโปรแกรมมีการเรียก Method ไว้มากมายซึ่งในแต่ละ Method นั้นมีการเรียกใช้ Method จาก class อื่นที่ extend มา

3. หลักฐานสนับสนุนเกี่ยวกับ QUALITY ATTRIBUTE นี้

- เช่น method getSelectionModel ถูกออกแบบและถูกเรียกใช้จากต่าง class สอดคล้องกับ กลวิธี Reusability

```

@Override
public void resetView() {
    super.resetView();
    version.resetView();
    compress.setSelected(false);
    compress.setSelected(userContext.isCompressionEnabled());

    @Override
    public void resetView() {
        this.rotationType.getSelectionModel().selectFirst();
        this.rotation.getSelectionModel().selectFirst();
    }

    @Override
    public void resetView() {
        splitAfterPredefined.resetView();
        splitAfter.resetView();
        splitByEvery.resetView();
    }
}

```

Weakness

PDFsam(version 4) ทำงานในระบบ 32-bits ไม่ได้แล้ว หากต้องการใช้ PDFsam ในระบบ 32-bits ต้องใช้ version 3

The screenshot shows the official PDFSAM website. At the top, there are navigation links: 'Download', 'PDFsam Basic', 'Features', 'PDFsam Enhanced', and 'PDFsam Visual'. Below these, a section titled 'Does it run on 32-bit systems?' contains the text: 'Version 4 requires a 64-bit system and it doesn't run on 32-bit while version 3 can run on 32-bit systems.' There is also a link 'For business'.

ใน Linux จะแยก package OpenJDK และ JavaFX ออกจากกัน ซึ่งถ้าขาด package ใด package หนึ่งก็จะไม่สามารถรัน PDFsam version 3 ได้ (version 4 ขึ้นไปจะไม่มีปัญหา เพราะไม่ต้องติดตั้ง JDK)

Open JDK on Linux?

As of version 4, a simplified version of OpenJDK 11 is bundled with the application and used to run PDFsam Basic. To run version 3 using OpenJDK check [this](#) out

java.lang.NoClassDefFoundError: javafx/application/Application

Some Linux distributions have separate packages for OpenJDK and JavaFX so, if you install the OpenJDK package and try to run PDFsam 3.x, you get an error because of the missing JavaFX :

```
Exception in thread "main" java.lang.NoClassDefFoundError: javafx/application/Application
  at java.lang.ClassLoader.defineClass1(Native Method)
  at java.lang.ClassLoader.defineClass(ClassLoader.java:760)
  at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)
  at java.net.URLClassLoader.defineClass(URLClassLoader.java:467)
  at java.net.URLClassLoader.access$100(URLClassLoader.java:73)
  at java.net.URLClassLoader$1.run(URLClassLoader.java:368)
  at java.net.URLClassLoader$1.run(URLClassLoader.java:362)
  at java.security.AccessController.doPrivileged(Native Method)
  at java.net.URLClassLoader.findClass(URLClassLoader.java:361)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
  at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:331)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
  at org.pdfsam.community.App.main(App.java:34)
Caused by: java.lang.ClassNotFoundException: javafx.application.Application
  at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
  at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:331)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
  ... 13 more
```

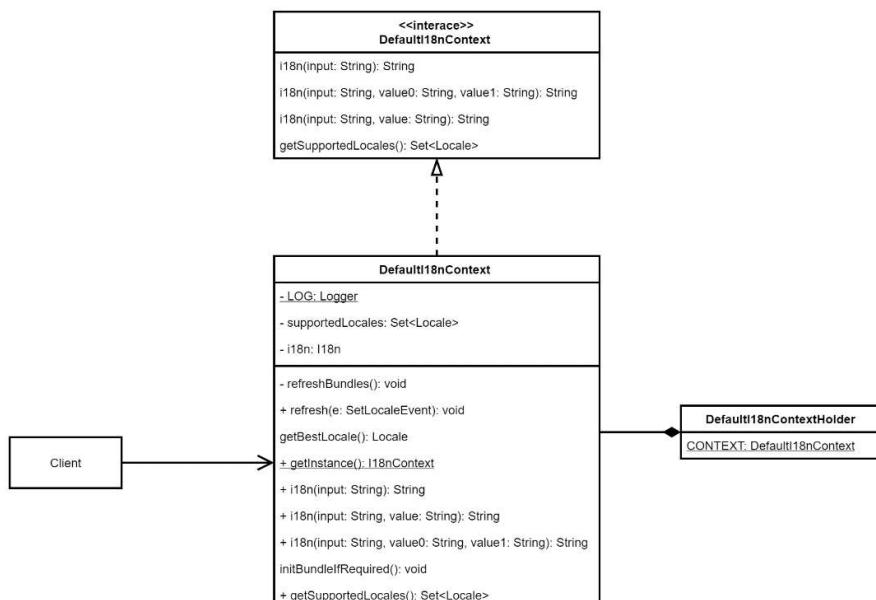
ซึ่งถ้าหากใช้ระบบปฏิบัติการ Linux 32-bits ถ้าไม่รู้ตระนี้จะไม่สามารถใช้งานได้ จึงส่งผลต่อ Compatibility ที่ลดลง

Design Pattern Analysis

1. Singleton

PDFSam มีการใช้ Singleton design pattern ในคลาส DefaultI18nContext สาเหตุที่มีการใช้ singleton เพราะ มีการใช้ library ที่ชื่อว่า i18next เป็น library สำหรับการทำแอปพลิเคชันหลายภาษา ดังนั้นมีผู้ใช้งานเลือกภาษาของแอปพลิเคชันภาษาใดภาษาหนึ่งไปแล้ว ทั้งแอปพลิเคชันก็ควรแสดงผลภาษาที่ผู้ใช้งานเลือกแค่เพียงภาษาเดียว

จาก diagram จะเห็นได้ว่า class DefaultI18nContext มี method ที่ชื่อว่า getInstance() ซึ่งเป็น method ที่จะ return instance ที่ชื่อว่า CONTEXT ใน inner class ของ DefaultI18nContext ซึ่งตรงตามหลักการของ singleton design pattern



UML Class Diagram Singleton ของ DefaultI18nContext

```

92     */
93     public static I18nContext getInstance() {
94         return DefaultI18nContextHolder.CONTEXT;
95     }
  
```

ไฟล์ pdfsam/pdfsam-i18n/src/main/java/org/pdfsam/i18n/DefaultI18nContext.java บรรทัดที่ 93 ถึง 95
<https://github.com/torakiki/pdfsam/blob/master/pdfsam-i18n/src/main/java/org/pdfsam/i18n/DefaultI18nContext.java>

```

127     /**
128      * Lazy initialization holder class idiom (Joshua Bloch, Effective Java second edition, item 71).
129      *
130      * @author Andrea Vacondio
131      *
132      */
133     private static final class DefaultI18nContextHolder {
134
135         private DefaultI18nContextHolder() {
136             // hide constructor
137         }
138
139         static final DefaultI18nContext CONTEXT = new DefaultI18nContext();
140     }
141 }
```

ไฟล์ pdfsam/pdfsam-i18n/src/main/java/org/pdfsam/i18n/DefaultI18nContext.java บรรทัดที่ 133 ถึง 140

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-i18n/src/main/java/org/pdfsam/i18n/DefaultI18nContext.java>

```

113     @EventListener
114     public void onTaskFailed(TaskExecutionFailedEvent event) {
115         openButton.setVisible(false);
116         failed.setVisible(true);
117         statusLabel.setText(DefaultI18nContext.getInstance().i18n("Failed"));
118     }
119
120     @EventListener
121     public void onProgress(PercentageOfWorkDoneChangedEvent event) {
122         statusLabel.setText(DefaultI18nContext.getInstance().i18n("Running"));
123         if (event.isUndetermined()) {
```

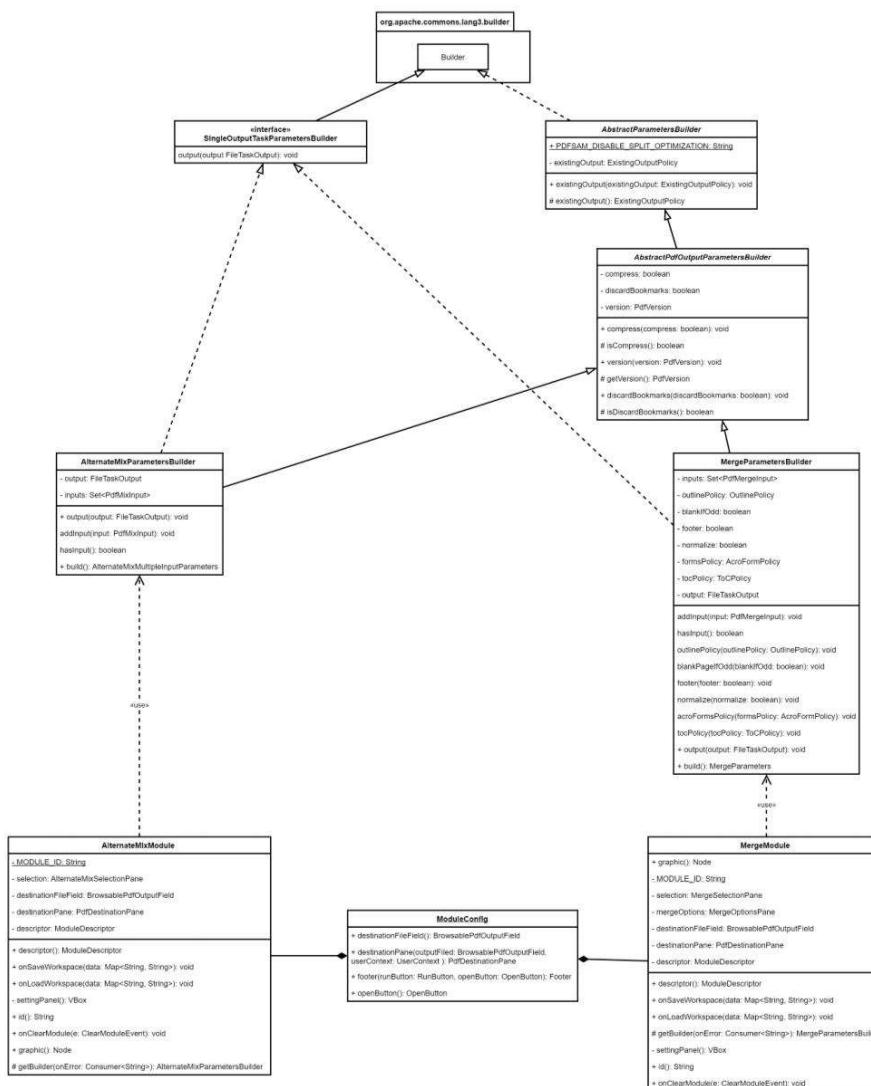
การใช้ Method getInstance() ใน Class Footer ในไฟล์ Footer.java บรรทัดที่ 117 และ 122

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/module/Footer.java>

2. Builder

PDFSam มีการใช้ Builder design pattern ใน module การจัดการไฟล์ pdf ต่างๆ เช่น merge, alternate-mix, rotate เป็นต้น เพื่อใช้ในการสร้าง Parameters ของ module ต่างๆ เพราะแต่ละ module ก็มีวิธีในการสร้าง Parameters ที่แตกต่างกัน แต่ก็ยังมีบางอย่างที่เหมือนกันอยู่ จึงใช้ Builder design pattern มาใช้งาน

จาก UML Class Diagram จะเห็นได้ว่ามีการสร้าง AbstractPdfOutputParametersBuilder มา implement ให้ Builder ของ module ต่างๆ มา extends ไป implement เพื่อเพิ่ม method ที่ต้องใช้สำหรับการสร้าง Parameters ให้แต่ละ module



ภาพต้นฉบับ : <https://raw.githubusercontent.com/thanatath/open-source-project-analysis/main/Design%20Pattern%20image/builder.png>

Source code ที่เกี่ยวข้องกับ Builder Design Pattern

```

25  * Abstract builder for {@link AbstractPdfOutputParameters}.
26  *
27  * @author Andrea Vacondio
28  * @param <P>
29  *      type of the parameters the builder builds
30  */
31 public abstract class AbstractPdfOutputParametersBuilder<P extends AbstractPdfOutputParameters>
32     extends AbstractParametersBuilder<P> {
33     private boolean compress;
34     private boolean discardBookmarks = false;
35     private PdfVersion version;
36

```

Class AbstractPdfOutputParametersBuilder ในไฟล์ AbstractPdfOutputParametersBuilder.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-core/src/main/java/org/pdfsam/support/params/AbstractPdfOutputParametersBuilder.java>

```

33 /**
34  * Builder for {@link MergeParameters}
35  *
36  * @author Andrea Vacondio
37  *
38 */
39 class MergeParametersBuilder extends AbstractPdfOutputParametersBuilder<MergeParameters>
40     implements SingleOutputTaskParametersBuilder<MergeParameters> {
41
42     private Set<PdfMergeInput> inputs = new NullSafeSet<>();
43     private OutlinePolicy outlinePolicy = OutlinePolicy.RETAIN;
44     private boolean blankIfOdd;
45     private boolean footer;
46     private boolean normalize;
47     private AcroFormPolicy formsPolicy = AcroFormPolicy.MERGE;
48     private ToCPolicy tocPolicy = ToCPolicy.NONE;
49     private FileTaskOutput output;

```

Class MergeParametersBuilder ในไฟล์ MergeParametersBuilder.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-merge/src/main/java/org/pdfsam/merge/MergeParametersBuilder.java>

```

30 /**
31  * Builder for the {@link AlternateMixMultipleInputParameters}
32  *
33  * @author Andrea Vacondio
34  *
35 */
36 class AlternateMixParametersBuilder extends AbstractPdfOutputParametersBuilder<AlternateMixMultipleInputParameters>
37     implements SingleOutputTaskParametersBuilder<AlternateMixMultipleInputParameters> {
38
39     private FileTaskOutput output;
40     private Set<PdfMixInput> inputs = new NullSafeSet<>();
41

```

Class AlternateMixParametersBuilder ในไฟล์ AlternateMixParametersBuilder.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-alternate-mix/src/main/java/org/pdfsam/alternatemix/AlternateMixParametersBuilder.java>

```

75
76     @Inject
77     public MergeModule(@Named(MODULE_ID + "field") BrowsablePdfOutputField destinationFileField,
78                         @Named(MODULE_ID + "pane") PdfDestinationPane destinationPane, @Named(MODULE_ID + "footer") Footer footer) {
79         super(footer);
80         this.destinationFileField = destinationFileField;
81         this.destinationPane = destinationPane;
82         initModuleSettingsPanel(settingPanel());
83     }
84

```

การสั่ง initModuleSettingPanel ของ MergeModule ในไฟล์ MergeModule.java บรรทัดที่ 82

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-merge/src/main/java/org/pdfsam/merge/MergeModule.java>

```

58     protected final void initModuleSettingsPanel(VBox panel) {
59         panel.getStyleClass().addAll(Style.DEFAULT_CONTAINER.css());
60         panel.getStyleClass().addAll(Style.MODULE_CONTAINER.css());
61         panel.getChildren().add(footer);
62
63         footer.runButton().setOnAction(event -> {
64             ErrorTracker errorTracker = new ErrorTracker();
65             Builder< AbstractParameters> builder = getBuilder(errorTracker
66                     .andThen(s -> eventStudio().broadcast(new AddNotificationRequestEvent(NotificationType.ERROR, s,
67                         DefaultI18nContext.getInstance().i18n("Invalid parameters"))));
68             if (!errorTracker.errorOnBuild()) {
69                 eventStudio().broadcast(new TaskExecutionRequestEvent(id(), builder.build()));
70             }
71         });
72         modulePanel.setCenter(panel);
73         eventStudio().addAnnotatedListeners(this);
74     }

```

การสั่ง getBuilder ของ BaseTaskExecutionModule ในไฟล์ BaseTaskExecutionModule.java บรรทัดที่ 65 ถึง 67

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-xf/src/main/java/org/pdfsam/ui/module/BaseTaskExecutionModule.java>

AlternateMixModule.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-alternate-mix/src/main/java/org/pdfsam/alternatemix/AlternateMixModule.java>

SingleOutputTaskParametersBuilder.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-core/src/main/java/org/pdfsam/support/params/SingleOutputTaskParametersBuilder.java>

AbstractParametersBuilder.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-core/src/main/java/org/pdfsam/support/params/AbstractParametersBuilder.java>

3. Observer (Event Listeners)

Why?

There are many different libraries with slightly different approaches implementing the pub/sub pattern but none of them was a perfect match for the problem I was trying to solve.

The problem

Working on a rich client software that can have multiple modules/plugins and:

- I wanted to be able to send events to the whole application (every listener subscribed on a given event)
- I wanted to be able to send events to a single module (every listener belonging to a given module and subscribed on a given event)
- I wanted the event to be queued in case no listener was listening and delivered as soon as a listener registers.
- I wanted listeners to be able to define at runtime what module they belong to. The idea was to create components (e.g. a SpecializedComboBox) listening for events (e.g. DisableEvent) and be able to reuse those components in different modules. I couldn't use the classical annotation approach with some topic/filter static value since every new instance of the component needed to be registered as listener for the module creating it.

Enter the Station

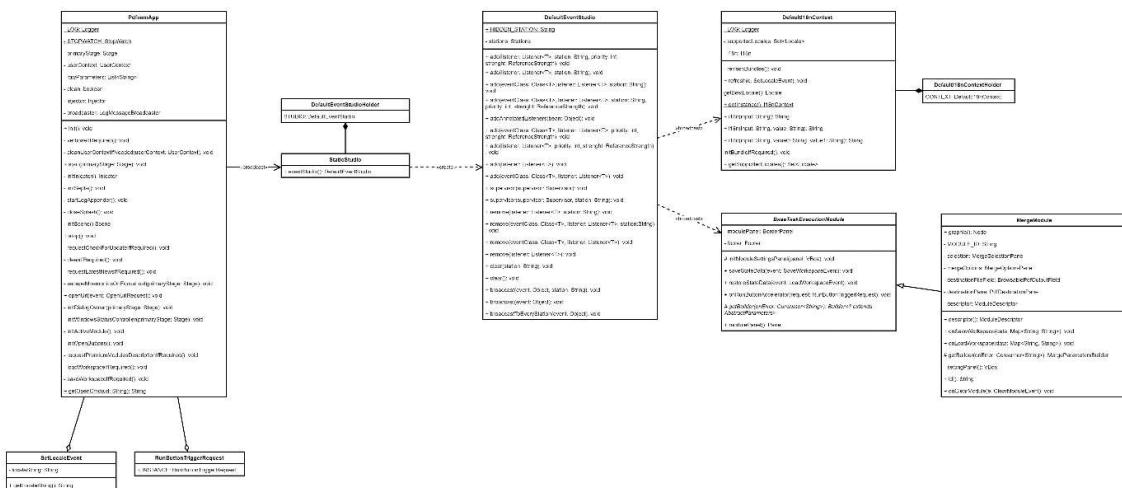
The solution I found was to mimic a network of radio stations. For those familiar with RabbitMQ, the architecture is loosely resembling a DirectExchange where the event class is the binding key.

```

graph LR
    P((P)) -- "broadcast SomeEvent" --> S1[Station1]
    P -- "broadcast AnotherEvent" --> S2[Station2]
    S1 -- "SomeEvent -->" --> L1((L1))
    S1 -- "SomeEvent -->" --> L2((L2))
    S2 -- "AnotherEvent -->" --> L3((L3))
    S2 -- "AnotherEvent -->" --> L4((L4))
    ...
    SN[StationN] -- "AnotherEvent -->" --> L5((L5))
  
```

PDFSam มีการใช้งาน Observer หรือ Event Listeners เพื่อใช้ในการส่ง event ไปยัง class ต่างๆ หลาย class โดย Event Listeners ของ PDFSam จะมีการแยก package ที่ชื่อว่า pdfsam.eventstudio ออกมา ซึ่งจะประกอบด้วย Publisher, Station และ Listener ซึ่งตรงตามหลักการของ Observer ที่ต้องมี Publisher, subscriber และ Event Manager โดย Event Studio มีการใช้ Annotation ในการกำหนดว่า method ไหนจะเป็น Listeners โดยไม่จำเป็นต้องไป subscribe ตามรูปแบบของ observer ทั่วๆไป โดยใช้ การกำหนด @EventListener ไว้ก่อน method

ยกตัวอย่าง class ที่ชื่อว่า BaseTaskExecutionModule ที่จะมีการกำหนด @EventListener ไว้ที่ method onRunButtonAccelerator() ที่มี parameter request ที่มี type เป็น RunButtonTriggerRequest ดังนั้นแปลว่า method นี้จะรับการ broadcast จาก class ใดก็ตามที่ทำการ broadcast event ที่เป็น RunButtonTriggerRequest มาให้ และใน class PdfsamApp ก็จะมีการ broadcast RunButtonTriggerRequest.INSTANCE ซึ่งก็คือการส่ง ไปให้ BaseTaskExecutionModule นั่นเอง สามารถดูได้จาก document ของ eventstudio ที่มีการเขียนบอกไว้ชัดเจน



ภาพต้นฉบับ : <https://raw.githubusercontent.com/thanatath/open-source-project-analysis/main/Design%20Pattern%20Image/observer.png>

Source code ที่เกี่ยวข้องกับ Observer Design Pattern

```

43     import org.pdfsam.i18n.DefaultI18nContext;
44     import org.pdfsam.i18n.SetLocaleEvent;
45     import org.pdfsam.injector.Injector;
46     import org.pdfsam.module.Module;
47     import org.pdfsam.event.StaticStudioEvent;

```

การ import StaticStudio.eventStudio ในไฟล์ PdfsamApp.java บรรทัดที่ 24

```

122         cleanUserContextIfNeeded(userContext);
123         String localeString = userContext.getLocale();
124         if (isNotBlank(localeString)) {
125             eventStudio().broadcast(new SetLocaleEvent(localeString));
126         }
127         String defaultWorkingPath = userContext.getDefaultWorkingPath();
128         if (isNotBlank(defaultWorkingPath)) {

```

การ broadcast event SetLocaleEvent ในไฟล์ PdfsamApp.java บรรทัดที่ 125

```

226             mainScene.getAccelerators().put(new KeyCodeCombination(KeyCode.X, KeyCodeCombination.CONTROL_DOWN),
227                 () -> eventStudio().broadcast(RunButtonTriggerRequest.INSTANCE));

```

การ broadcast event RunButtonTriggerRequest ในไฟล์ PdfsamApp.java บรรทัดที่ 226

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-ui/src/main/java/org/pdfsam/PdfsamApp.java>

```

26  public final class StaticStudio {
27
28      private StaticStudio() {
29          // hide
30      }
31
32      public static DefaultEventStudio eventStudio() {
33          return DefaultEventStudioHolder.STUDIO;
34      }
35
36      /**
37       * Lazy initialization
38       *
39       * @author Andrea Vacondio
40       *
41       */
42      private static final class DefaultEventStudioHolder {
43
44          private DefaultEventStudioHolder() {
45              // hide constructor
46          }
47
48          static final DefaultEventStudio STUDIO = new DefaultEventStudio();
49      }
50  }

```

การสร้าง DefaultEventStudio ใน Inner class DefaultEventStudioHolder

ในไฟล์ StaticStudio.java บรรทัดที่ 48

<https://github.com/torakiki/event-studio/blob/master/src/main/java/org/pdfsam/eventstudio/StaticStudio.java>

```

55  */
56  public class DefaultEventStudio implements EventStudio {
57      /**
58      * A reserved station name that is used to hide the station abstraction. Using the provided helper methods the station concept remains totally hidden and {@link EventStudio#broadcast(Object, String) broadcast} can be used as a more traditional Event Bus with pub/sub pattern.
59      */
60      public static final String HIDDEN_STATION = "hidden.station";
61
62      private Stations stations = new Stations();
63
64
197      public void broadcast(Object event, String station) {
198          stations.getStation(station).broadcast(event);
199      }
200
201      /**
202       * Broadcast the event to the hidden station
203       *
204       * @see EventStudio#broadcast(Object, String)
205       * @see DefaultEventStudio#HIDDEN_STATION
206       */
207      public void broadcast(Object event) {
208          stations.getStation(HIDDEN_STATION).broadcast(event);
209      }
210

```

Method broadcast ใน class DefaultEventStudio ในไฟล์ DefaultEventStudio.java บรรทัดที่ 197 ถึง 209

<https://github.com/torakiki/event-studio/blob/master/src/main/java/org/pdfsam/eventstudio/DefaultEventStudio.java>

```

65      @EventListener
66      public void refresh(SetLocaleEvent e) {
67          String localeString = e.getLocaleString();
68          if (StringUtil.isNotBlank(localeString)) {
69              LOG.trace("Setting default locale to {}", localeString);
70              Optional.ofNullable(Locale.forLanguageTag(localeString))
71                  .ifPresent(Locale::setDefault);
72              refreshBundles();
73          }
74      }
75

```

Method refresh ใน class Defaul18nContext ที่มีการใช้ @EventListener

ในไฟล์ Defaul18nContext.java บรรทัดที่ 65 ถึง 74

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-i18n/src/main/java/org/pdfsam/i18n/DefaultI18nContext.java>

```
85  
86     @EventListener  
87     public void onRunButtonAccelerator(RunButtonTriggerRequest request) {  
88         footer.runButton().fire();  
89     }  
90
```

Method onRunButtonAccelerator ใน class BaseTaskExecutionModule
ที่มีการใช้ @EventListener ในไฟล์ BaseTaskExecutionModule.java บรรทัดที่ 86 ถึง 89
<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/module/BaseTaskExecutionModule.java>

MergeModule.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-merge/src/main/java/org/pdfsam/merge/MergeModule.java>

RunButtonTriggerRequest.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/module/RunButtonTriggerRequest.java>

SetLocaleEvent.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-i18n/src/main/java/org/pdfsam/i18n/SetLocaleEvent.java>

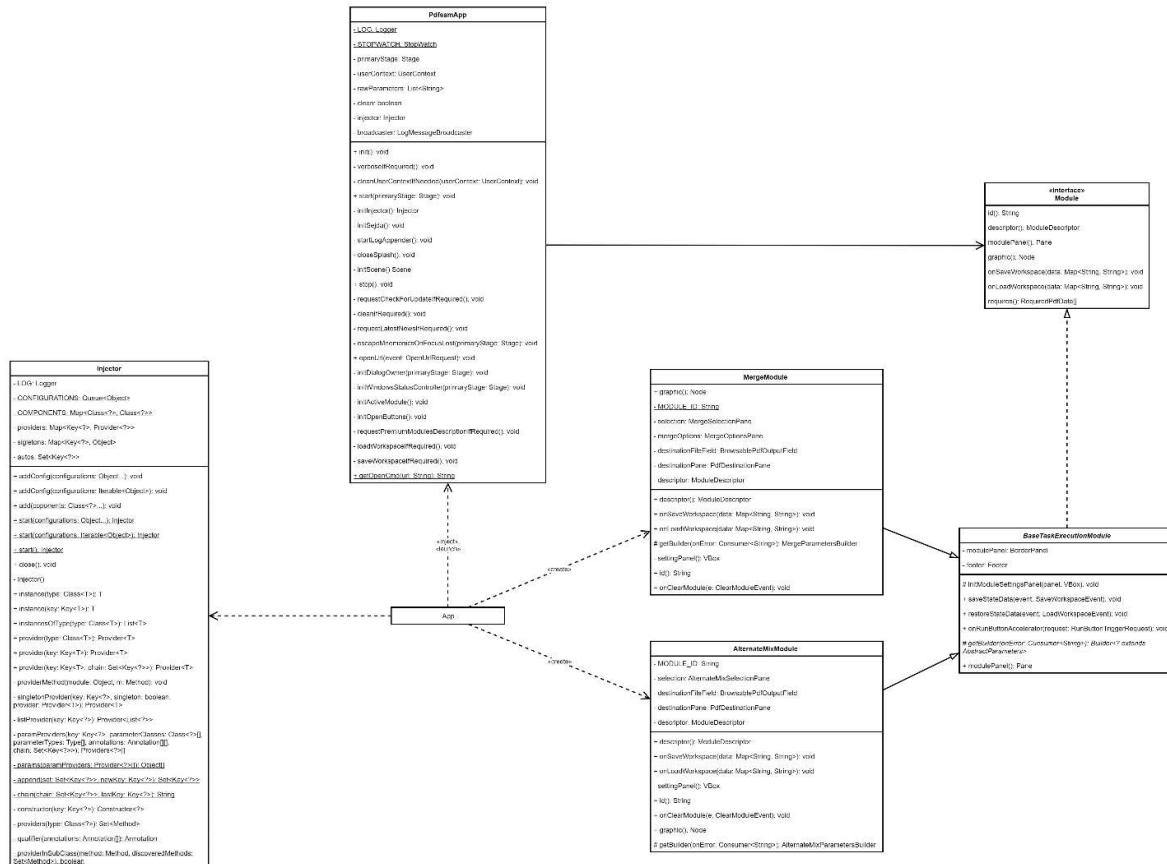
4. Dependency Injection

PDFSam มีการใช้ Dependency Injection Design Pattern เพื่อป้อน config ของ module ที่ใช้ jackson API PDF ต่างๆ ให้แก่ class PdfsamApp เพื่อรับการ config ของแต่ละ module ที่แตกต่างกัน

จาก UML Class Diagram จะเห็นได้ว่า class App เป็น class เริ่มต้นการทำงานโดยมันจะไปเรียกใช้ injector จากนั้นก็ส่ง config ของ module ต่างๆ เข้าไปใน method addConfig() และให้ตัว application launch PdfsamApp

จากนั้นเมื่อเราดูใน class PdfsamApp จะเห็นได้ว่าการเรียกใช้ module ต่างๆ จะมีการเรียกใช้ injector กับ class Module ซึ่งเป็น interface ของ module ใน การจัดการ pdf ต่างๆ ซึ่งตรงตามหลักการของ dependency injection ที่จะมี Consumer, Service และ Injector

PdfsamApp คือ consumer, App เป็นตัวเรียกใช้ injector และ Module ในการจัดการ pdf เป็น service



ภาพทั้งฉบับ : <https://github.com/thanatath/open-source-project-analysis/blob/main/Design%20Pattern%20image/dependency-injection.png?raw=true>

Source code ที่เกี่ยวข้องกับ Dependency Injection Design Pattern

```

27  * PDFsam Basic Edition App
28  *
29  * @author Andrea Vacondio
30  *
31  */
32 public class App {
33     public static void main(String[] args) {
34         Injector.addConfig(new PdfsamBasicConfig(), new org.pdfsam.alternatemix.AlternateMixModule.ModuleConfig(),
35             new org.pdfsam.extract.ExtractModule.ModuleConfig(), new org.pdfsam.merge.MergeModule.ModuleConfig(),
36             new org.pdfsam.rotate.RotateModule.ModuleConfig(), new org.pdfsam.split.SplitModule.ModuleConfig(),
37             new org.pdfsam.splitbybookmarks.SplitByBookmarksModule.ModuleConfig(),
38             new org.pdfsam.splitysize.SplitBySizeModule.ModuleConfig());
39         Application.launch(PdfsamApp.class, args);
40     }
41 }
42 
```

การใช้ Injector injects config ของ module ต่างๆ ใน class main

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-basic/src/main/java/org/pdfsam/basic/App.java>

```

99  */
100 public class PdfsamApp extends Application {
101     private static final Logger LOG = LoggerFactory.getLogger(PdfsamApp.class);
102
103     private static final Stopwatch STOPWATCH = new Stopwatch();
104     private Stage primaryStage;
105     private UserContext userContext = new DefaultUserContext();
106     private List<String> rawParameters;
107     private boolean clean;
108     private Injector injector;
109     private LogMessageBroadcaster broadcaster;
110
111     private void initOpenButtons() {
112         List<Module> modules = injector.instancesOfType(Module.class);
113         List<OpenButton> openButtons = injector.instancesOfType(OpenButton.class);
114         for (OpenButton button : openButtons) {
115             button.initModules(modules);
116         }
117     }
118 }
119 
```

การนำ injector ไปใช้งานใน class PdfsamApp ไฟล์ PdfsamApp.java

บรรทัดที่ 108 และบรรทัดที่ 319 ถึง 325

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-gui/src/main/java/org/pdfsam/PdfsamApp.java>

```

156     * @return an instance of type
157     */
158     @SuppressWarnings("unchecked")
159     public <T> List<T> instancesOfType(Class<T> type) {
160         return (List<T>) listProvider(Key.of(type)).get();
161     }
162 
```

Method instancesOfType() ในคลาส Injector

ไฟล์ Injector.java บรรทัดที่ 156 ถึง 161

<https://github.com/torakiki/pdfsam-injector/blob/master/src/main/java/org/pdfsam/injector/Injector.java>

MergeModule.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-merge/src/main/java/org/pdfsam/merge/MergeModule.java>

AlternateMixModule.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-alternate-mix/src/main/java/org/pdfsam/alternatemix/AlternateMixModule.java>

BaseTaskExecutionModule.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/module/BaseTaskExecutionModule.java>

Module.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-core/src/main/java/org/pdfsam/module/Module.java>

5. Call back

นอกจาก Design pattern 4 รูปแบบจากที่เจอใน PDFSam แล้ว นอกจากนั้นยังทางผู้จัดทำ ยังพบรูปแบบนี้อีกตัวอย่างหนึ่งที่ใช้ Call back ที่เป็น Design Pattern อีกรูปแบบนอกเหนือจากบทเรียนดังนี้

```

51     default Callback<TableColumn<SelectionTableRowData, T>, TableCell<SelectionTableRowData, T>> cellFactory() {
52         return new Callback<>() {
53             @Override
54             public TableCell<SelectionTableRowData, T> call(TableColumn<SelectionTableRowData, T> param) {
55                 return new TableCell<>() {
56                     @Override
57                     public void updateItem(final T item, boolean empty) {
58                         super.updateItem(item, empty);
59                         if (empty || item == null) {
60                             setText("");
61                         } else {
62                             setText(getTextValue(item));
63                         }
64                     }
65                 };
66             }
67         };
68     };

```

แสดงการนำ callback มาใช้สร้าง TableCell

หมายเหตุ ในหลายๆ ตำแหน่งและความคิดเห็นอื่นๆ อาจจะไม่เห็นด้วยกับการที่ Call back ถูกใช้เป็น Design pattern แต่ในอีกในหลายๆ ความเห็น ก็ยังคงมองว่า Call back เป็นหนึ่งใน Design pattern

อ้างอิง

<https://java-design-patterns.com/patterns/callback/>

<https://subscription.packtpub.com/book/web-development/9781783287314/1/ch01lvl1sec10/the-callback-pattern>

Source code ที่เกี่ยวข้องกับ Call back Design Pattern

LoadingColumn.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/selection/multiple>LoadingColumn.java>

IndexColumn.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/selection/multiple>

Selection TableColumn.java

<https://github.com/torakiki/pdfsam/blob/master/pdfsam-fx/src/main/java/org/pdfsam/ui/selection/multiple>