

Architectural Patterns/Styles

Audacity

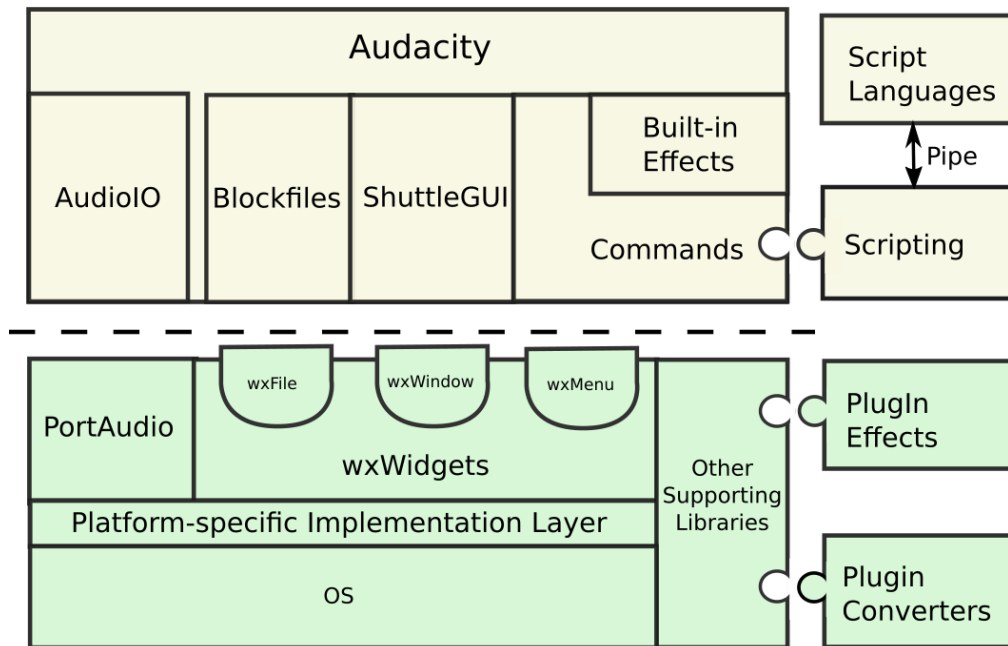
Purpose of the project :

Audacity เป็นโปรแกรมฟรีโอเพ่นซอร์สซอฟต์แวร์ฟรีที่ใช้สำหรับแก้ไขและบันทึกเสียงแบบหลายแทร็กที่ใช้งานง่ายสำหรับ Windows, macOS, GNU/Linux และระบบปฏิบัติการอื่นๆ โดยเป้าหมายหนึ่งของ Audacity คือการทำให้ User Interface ใช้งานง่าย ผู้ใช้งานสามารถใช้งานได้โดยไม่ต้องอ่านคู่มือ

Feature ของ Audacity มีดังนี้

- การบันทึกเสียง
- นำไฟล์เข้า-ออกได้หลายรูปแบบ
- คุณภาพเสียงรองรับ 16-bit, 24-bit และ 32-bit
- มี Plug-in จำนวนมาก
- สามารถแก้ไขได้หลายรูปแบบ
- ติดตั้ง effect ได้
- มีการเข้าถึงที่ง่าย มีคีย์ลัดมากมาย
- การวิเคราะห์
- โหมดดูสเปกตรัมสำหรับการแสดงภาพและเลือกความถี่ หน้าต่างพล็อตสเปกตรัมสำหรับการวิเคราะห์ความถี่โดยละเอียด

Architectural patterns/styles :



มีพื้นฐานจากการใช้ Library จำนวนมาก โดย Library ที่สำคัญคือ PortAudio ที่ให้ Low-level Audio Interface และ wxWidgets ที่ให้ GUI Components

Library อื่นๆที่ใช้ต่อยอดจาก wxWidgets และ PortAudio

- BlockFile ใช้ OS file system ผ่าน wxWidgets เพื่อให้วิธีการเก็บเสียงเป็นกลุ่มเล็กๆ ซึ่งทำให้สามารถตัดต่อปรับแต่งเสียงโดยไม่จำเป็นต้องปรับแต่งทั้งไฟล์
- ShuttleGUI ใช้ wxWidgets ในการจัดการ dialog, ปุ่ม และการควบคุมอื่นๆ เพื่อการเขียนโค้ดซ้ำ ผ่านการเก็บข้อมูลเป็นตัวแทน
- Command จัดการ Bind ปุ่มในคีย์บอร์ด ผ่าน wxWidgets
- AudioIO จัดการการเคลื่อนย้ายเสียงระหว่าง Sound card, memory, hard disk ผ่าน PortAudio

Design Architecture :

Service-Oriented Architecture จากโครงสร้างตามรูปด้านบนมีการใช้งาน API หลายชนิดที่มีความสัมพันธ์กัน เช่นการใช้งาน BlockFile ผ่าน wxWidget หรือการใช้ AudioIO ผ่าน PortAudio

3 quality attribute scenarios :

1. Useability จากจุดมุ่งหมายที่ต้องการให้แอปพลิเคชันใช้งานง่าย
2. Modifiability การเป็น Open-Source, การพยายามเขียนโค้ดในลักษณะเดียวกัน
3. Performance จากการใช้ BlockFile ให้ไม่จำเป็นต้องแก้ไขไฟล์ทั้งไฟล์

Report references :

- <https://wiki.audacityteam.org/wiki/ArchitecturalDesign>
- <http://www.aosabook.org/en/audacity.html>

Matplotlib

Purpose of the project :

Matplotlib เป็นไลบรารีที่ครอบคลุมสำหรับการสร้างการแสดงผลแบบ static ภาพเคลื่อนไหว และแบบ interactive ใน Python โดยเริ่มแรกได้รับการออกแบบด้วยหลักที่ว่า “คุณควรจะ plot graph ง่ายๆ ได้ด้วยคำสั่งเพียงไม่กี่คำสั่ง” โดยปัจจุบันสิ่งที่ Matplotlib ทำได้คือ

- สร้างการ plot แบบคุณภาพ
- มี interactive ภาพที่แสดงผล สามารถขยาย เลื่อน อัปเดต ได้
- ปรับแต่งสไตล์และเลย์เอาต์ของภาพ
- ส่งออกไฟล์ได้หลายรูปแบบ
- Embed ใน JupyterLab และ Graphical User Interfaces
- ใช้อาร์เรย์ที่หลากหลายของแพ็คเกจบุคคลที่สามที่สร้างบน Matplotlib

Architectural patterns/styles :

matplotlib แบ่งออกเป็น 3 ส่วนตามแนวคิด

- Interface ของ MATLAB คือชุดของฟังก์ชันที่อนุญาตให้ผู้ใช้สร้างพล็อตจากบรรทัดคำสั่ง
- frontend หรือ matplotlib API คือชุดของคลาสที่ทำงานหนักๆ โดยการสร้างและจัดการตัวเลข ข้อความ เส้น โครงเรื่อง ฯลฯ นี่เป็นส่วนต่อประสานนามธรรมที่ไม่รู้อะไรเลยเกี่ยวกับผลลัพธ์
- backend เป็นอุปกรณ์วาดภาพหรือเรนเดอร์ที่ขึ้นอยู่กับอุปกรณ์ที่เปลี่ยนการแสดงผลส่วนหน้าเป็นเอกสาร (JPEG, PNG, PDF, PS, SVG, Paint, GD) หรืออุปกรณ์แสดงผล (Agg, GTK/GTKAgg, TkAgg, WX/WXAgg) โค้ดการเรนเดอร์ที่สำคัญส่วนใหญ่เขียนด้วยภาษา C/C++ จึงให้ประสิทธิภาพที่ดีมาก

Design Architecture :

Model-View-Controller (MVC) โดยจาก software architecture นั้น จะเห็นได้ว่าเมื่อมีการเรียกใช้ไลบรารี matplotlib โดยตัว user นั้นจะใช้ interface ในการสั่งการจึงเปรียบ interface ได้กับ controller จากนั้นจะส่งคำสั่งไปยัง front-end ซึ่งก็คือ model จากนั้นจึงจะ update ไปยัง view หรือก็คือ back-end ซึ่งจะทำหน้าที่ในการแสดงผล/renderผลลัพธ์ส่งไปยัง user

3 quality attribute scenarios :

1. Usability โดย matplotlib นั้นสามารถเรียนรู้ได้รวดเร็ว, ใช้ได้อย่างมีประสิทธิภาพ, สามารถปรับเปลี่ยนตามความต้องการของผู้ใช้ได้, สามารถทำงานได้อย่างง่ายเพราะ code ที่ใช้งานง่าย
2. Modifiability โดย matplotlib นั้นสามารถแก้ไขดัดแปลงได้เนื่องจากเป็นsoftwareที่เป็นในรูปแบบ open-source
3. Integrability โดย matplotlib นั้นใช้งานร่วมกับไฟล์ชนิด TEX, การเซฟรูป figure เป็นไฟล์รูปชนิดมาตรฐาน

Report references :

- <https://medium.com/@codingpilot25/architecture-of-matplotlib-1a2d44370f5a>
- [https://www.researchgate.net/publication/234238535_matplotlib _--
_A_Portable_Python_Plotting_Package](https://www.researchgate.net/publication/234238535_matplotlib_-_A_Portable_Python_Plotting_Package)

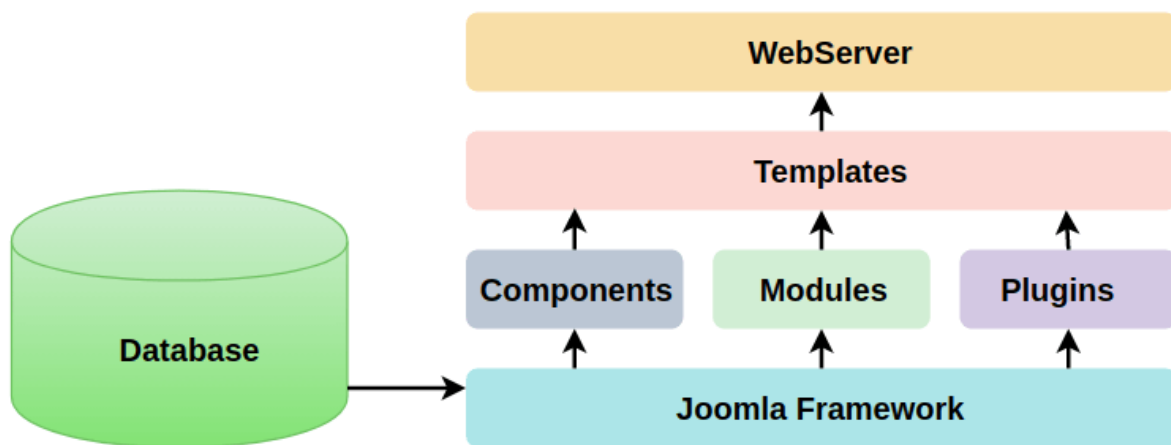
Joomla

Purpose of the project :

Joomla คือระบบจัดการเนื้อหาแบบโอเพ่นซอร์ส (CMS) สำหรับการเผยแพร่เนื้อหาบนเว็บ ทำให้ผู้ใช้สามารถสร้างเว็บไซต์และแอปพลิเคชันออนไลน์ที่ทรงพลังที่สามารถใช้งานได้โดยอิสระ โดยที่ CMS หรือ content management system เป็นแอปพลิเคชัน (บนเว็บ) ที่ให้ความสามารถสำหรับผู้ใช้หลายคนที่มีระดับ permission ที่แตกต่างกันในการจัดการเนื้อหา ข้อมูล หรือข้อมูลของ website project หรือแอปพลิเคชันอินเทอร์เน็ต

Managing content (การจัดการเนื้อหา) หมายถึง การสร้าง แก้ไข เก็บถาวร เผยแพร่ ทำงานร่วมกัน การรายงาน กระจายเนื้อหาเว็บไซต์และข้อมูล

Architectural patterns/styles :



Joomla Architecture

Joomla เขียนด้วย PHP และอิงตามรูปแบบการออกแบบ MVC (Model-View-Controller) ใช้ MySQL (MS SQL เวอร์ชัน 2.5 ขึ้นไป และ PostgreSQL เวอร์ชัน 3.0 ขึ้นไป) เพื่อจัดเก็บข้อมูล มีคุณสมบัติหลากหลาย (เช่น การแคชหน้า บล็อก โพล การสนับสนุนภาษาสากล และฟีด RSS เป็นต้น) ซึ่งทำให้ Joomla เป็นตัวเลือกที่ยอดเยี่ยมสำหรับ CMS (ระบบการจัดการเนื้อหา)

Design Architecture :

Layer โดยจาก software architecture นั้น จะเห็นได้ว่าการแบ่งเป็น layer อย่างชัดเจนโดยแต่ละ layer คือการจัดกลุ่มของ module ที่นำเสนอชุดบริการที่สอดคล้องกัน และความสัมพันธ์ของแต่ละ layer นั้น ต้องไปในทิศทางเดียวกัน

โดย Layer จะมีดังนี้

- Database
- Joomla Framework
- Components
- Modules
- Plugin
- Templates
- Web Server

3 quality attribute scenarios :

1. Usability ใช้เวลาน้อยลงในการเขียนโค้ดและลดงานที่น่าเบื่อที่เกี่ยวข้องกับการสร้างอินเทอร์เน็ตเพจใน Joomla 3.9 Joomla เวอร์ชันล่าสุด มีคุณสมบัติที่มี CSS และ jQuery น้อยกว่า ดังนั้น คุณจึงเขียนโค้ดได้น้อยลง
2. Modifiability การจัดการเนื้อหาที่เป็นมิตรต่อผู้ใช้และใช้กันอย่างแพร่หลายและระบบนั้นมีคุณสมบัติที่ปรับแต่งได้ง่าย ในกรณีของการอัปเดต ผู้ดูแลระบบ/นักพัฒนาสามารถอัปเดตเวอร์ชัน joomla ได้อย่างง่ายดายด้วยปุ่มอัปเดตเพียงคลิกเดียว
3. Performance ระบบการจัดการเนื้อหาที่เป็นหัวใจสำคัญ และมีคุณสมบัติที่ยืดหยุ่นบางอย่างที่ทำให้การจัดระเบียบและจัดการเนื้อหาของผู้ใช้เป็นเรื่องง่าย

Report references :

- <https://www.javatpoint.com/architecture-of-joomla>
- https://www.tutorialspoint.com/joomla/joomla_architecture.htm