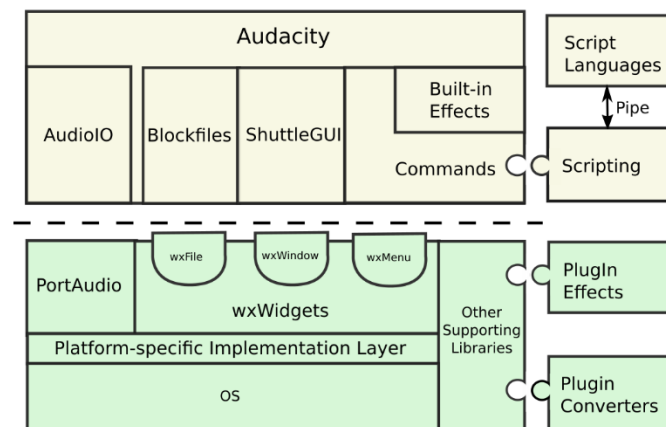


Architectural Pattern Styles

Software แบบที่ 1

1. Audacity

- **Purpose** เป็นโปรแกรม OpenSource ที่ใช้ตัดต่อและบันทึกเสียงด้วยการใช้ Effect ต่างๆ ซึ่งมีอยู่มากมายและครบครัน เช่น เพิ่มเสียง ลดเสียง อัดเสียง ตัดต่อเสียง และอื่นๆ เป็นโปรแกรมที่มีเครื่องมือให้ใช้หลากหลาย แต่ก็ยังคงเป็นโปรแกรมที่ใช้งานได้ง่าย และรองรับในระบบปฏิบัติการในทุกแพลตฟอร์ม ไม่ว่าจะเป็น Window, Mac, Linux เป็นต้น
- **Architectural patterns/styles** Audacity มี Architectural patterns เป็นแบบ Monolithic แล้วจะจัดอยู่ในรูปแบบที่เรียกว่า Plugin ซึ่งการทำงานของ Audacity มีลักษณะเป็นชั้นๆ อยู่บน Libraries ซึ่งการทำงานส่วนใหญ่จะขึ้นอยู่กับ Libraries ที่เป็น Function build in และสามารถรองรับ LADSPA ซึ่งช่วยในการ Download audio Effect มาใช้ได้



- **Quality attribute**

Scenarios 1 : Integrability

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : ต้องการเพิ่ม Plugin
- Environment : Runtime
- Artifact : Componet Metadata
- Response : New configuration/Function
- Response Measure : มี Plugin ใช้งานได้มากขึ้น

Scenarios 2 : Usability

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : เรียนรู้การใช้งาน
- Environment : Runtime
- Artifact : Voice Editor
- Response : ช่วยในการใช้งาน
- Response Measure : ความพึงพอใจของผู้ใช้

Scenarios 3 : Modifiability

- Source of Stimulus : Developer
- Stimulus : ปรับแต่งการทำงาน
- Environment : Development time
- Artifact : code
- Response : modify
- Response Measure : complexity, efforts are greatly reduced

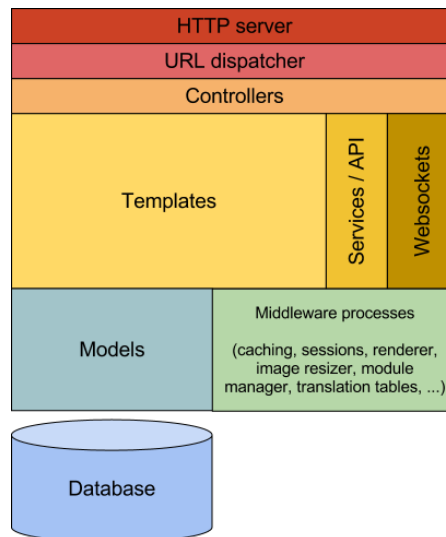
<https://www.mindphp.com/บทความ/web-app-graphic/8040-audacity-program.html>

<https://wiki.audacityteam.org/wiki/ArchitecturalDesign>

<https://www.audacityteam.org/about/voluntary-product-accessibility-template/>

2. Zotonic

- **Purpose** เป็น Open source software ที่พัฒนาเว็บไซต์แบบ full-stack หรือทำทั้ง frontend - backend ซึ่งตัวโปรแกรมเองพัฒนามาจากระบบ CMS โดยจุดเด่นคือใช้ภาษา Erlang ในการพัฒนาทำให้ตัว framework ทนทานต่อข้อผิดพลาดและมี performance ที่ดี จุดประสงค์หลักๆที่ framework นี้จะช่วยแก้ไขในเรื่อง scalability ที่ดีกว่าเดิม
- **Architectural patterns/styles** Zotonic มี Architectural patterns เป็นแบบ client server ซึ่ง zotonic มี web server แบบ build-in ชื่อว่า mochiweb มี controller ไว้สำหรับ handle request ในรูปแบบของ RESTful api และใช้ webMachine ในการคุม URL routing system ส่วนในเรื่องของ database จะเก็บเนื้อหาไว้ใน postgresSQL ส่วนในเรื่องของหน้าเว็บจะใช้ Erlydtl ซึ่งเป็น web template language ซึ่งมีลักษณะคล้ายๆ asp และ based มาจาก Django framework ของ python



- **Quality attribute**

Scenarios 1 : Usability

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : เข้าไปใช้งาน
- Environment : Runtime
- Artifact : UI

- Response : สามารถเข้าไปใช้งานได้
- Response Measure : ความพึงพอใจของผู้ใช้

Scenarios 2 : Performance

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : เข้าไปใช้งาน
- Environment : Runtime
- Artifact : เว็บไซต์
- Response : การตอบโต้ของตัวเว็บ
- Response Measure : เวลาที่ใช้ในการตอบโต้กับUser

Scenarios 3 : Availability

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : ต้องการให้รองรับ User ได้มากขึ้น
- Environment : Development time
- Artifact : Code, Server
- Response : apache test user
- Response Measure : latency

<https://zotonic.com/>

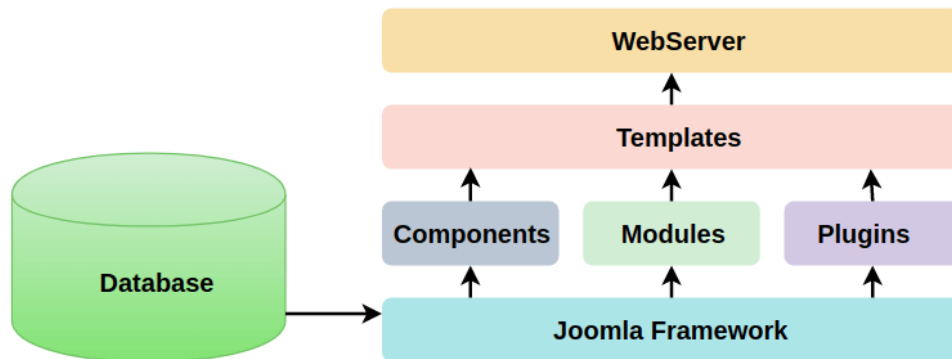
<https://www.aosabook.org/en/posa/zotonic.html>

<https://groups.google.com/g/zotonic-users/c/EeLJ54OZLqo/m/llcljlafAUoJ?pli=1>

Software แบบที่ 2

1. Joomla

- **Purpose** Joomla เป็นระบบบริหารจัดการเว็บไซต์ หรือ Web CMS (Web Content Management System) แบบ Open Source อีกตัวหนึ่งที่มีความนิยม และใช้งานกันแพร่หลายในปัจจุบัน พัฒนาโดยใช้PHP และใช้ฐานข้อมูล MySQL ซึ่งCMS นั้นหมายความว่า สิ่งที่เราจะต้องดูแลก็คือเนื้อหาเว็บไซต์ ไม่จำเป็นต้องมานั่งเขียนโค้ด HTML JS PHP SQL เพียงแต่เรียนรู้วิธีการติดตั้ง การใช้งาน และการปรับแต่ง CMS เท่านั้นเอง
- **Architectural patterns/styles** Joomla ใช้สถาปัตยกรรมการออกแบบ Model-View-Controller(MVC) โดยเมื่อ Joomla ประมวลผลคำขอ อันดับแรกจะวิเคราะห์ URL เพื่อประเมินว่าองค์ประกอบใดจะประมวลผลคำขอ โมเดลประกอบด้วยข้อมูลที่ส่วนประกอบใช้ นอกจากนี้ยังเป็นความรับผิดชอบของ Model ในการอัปเดตฐานข้อมูลเมื่อจำเป็น



Joomla Architecture

- **Quality attribute**

Scenarios 1 : Usability

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : เรียนรู้การใช้งาน
- Environment : Runtime
- Artifact : เกิดขั้นตอนการ Generate Graph
- Response : UI Graph
- Measure : ความพึงพอใจของผู้ใช้

Scenarios 2 : Performance

- Source of Stimulus : ผู้ใช้งาน
- Stimulus : คำสั่งจากผู้ใช้งาน
- Environment : Normal Status
- Artifact : เกิดขั้นตอนการ Generate Graph / Timer
- Response : เวลาในการสร้าง Graph เปรียบเทียบกับ Matlab
- Measure : Latency

Scenarios 3 : Security

- Source of Stimulus : แฮกเกอร์
- Stimulus : พยายามเข้าถึงโดยที่ไม่ได้รับอนุญาต
- Environment : ออนไลน์
- Artifact : Certain service
- Response : การตอบสนองการเข้าถึงที่เข้าโดยไม่ได้รับอนุญาต
- Measure : ไม่มีข้อมูลสูญหายหรือมีการเข้าถึง

https://www2.bsru.ac.th/downloads/manual_joomla2558.pdf

<https://www.mindphp.com/คู่มือ/73-คืออะไร/2560-joomla-cms.html>