

Matplotlib

matplotlib สามารถใช้ในการตั้งค่าต่างๆ ผู้ใช้ส่วนใหญ่คุ้นเคยกับบรรทัดคำสั่งสำหรับการสร้างพล็อตและรูปภาพแบบโต้ตอบ อินเทอร์เฟซนี้มีหน้าต่างป๊อปอัพอย่างง่ายสำหรับการแสดงและจัดการข้อมูล อย่างไรก็ตาม พลังที่แท้จริงของ matplotlib คือไลบรารีการลงจุดพื้นฐาน ซึ่งเป็นระบบปฏิบัติการที่ไม่ขึ้นกับระบบปฏิบัติการและส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) สามารถใช้โดยไม่มี GUI เป็นส่วนหนึ่งของเว็บเซิร์ฟเวอร์เพื่อสร้างพล็อตและรูปภาพในเอด์พุตสำเนาที่หลากหลาย หรือฝังลงในแอปพลิเคชันขนาดใหญ่ได้โดยใช้ GUI ตัวใดตัวหนึ่ง (เช่น GTK, Tk หรือ Wxwindows) ที่ทำงานบนหนึ่งในระบบปฏิบัติการหลายตัว (เช่น Windows, OS X, Solaris และ Linux)

เป้าหมายเริ่มต้นของ matplotlib คือ:

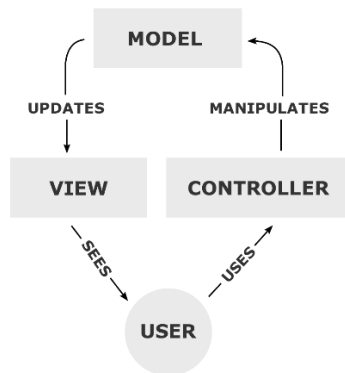
- โครงเรื่องควรมีคุณภาพในการตีพิมพ์; โดยเฉพาะข้อความ (antialiased, rotated เป็นต้น)
- เอด์พุต PostScript เพื่อรวมเข้ากับเอกสาร TEX
- ฝังได้ในส่วนติดต่อผู้ใช้แบบกราฟิกสำหรับการพัฒนาแอปพลิเคชัน
- code ควรจะเข้าใจได้ง่าย
- การสร้าง plot ควรจะทำได้ง่ายๆ
- ซอฟต์แวร์นี้เป็นโอเพ่นซอร์ส จึงสามารถดาวน์โหลด ใช้ และแจกจ่ายได้อย่างอิสระ

Software Architecture

matplotlib แบ่งออกเป็นสามส่วนตามแนวคิด:

- อินเทอร์เฟซของ MATLAB คือชุดของฟังก์ชันที่อนุญาตให้ผู้ใช้สร้างพล็อตจากบรรทัดคำสั่ง
- frontend หรือ matplotlib API คือชุดของคลาสที่ทำงานหนักๆ โดยการสร้างและจัดการตัวเลข ข้อความ เส้น โครงเรื่อง ฯลฯ นี้คือส่วนต่อประสานนามธรรมที่ไม่รู้อะไรเลยเกี่ยวกับผลลัพธ์
- แบ็กเอนด์เป็นอุปกรณ์วาดภาพหรือเรนเดอร์ที่ขึ้นอยู่กับอุปกรณ์ที่เปลี่ยนการแสดงผลส่วนหน้าเป็นเอกสาร (JPEG, PNG, PDF, PS, SVG, Paint, GD) หรืออุปกรณ์แสดงผล (Agg, GTK/GTKAgg, TkAgg, WX/WXAgg). โค้ดการเรนเดอร์ที่สำคัญส่วนใหญ่เขียนด้วยภาษา C/C++ จึงให้ประสิทธิภาพที่ดีมาก

Design Architecture



จะเป็นรูปแบบ design ของ **Model-View-Controller (MVC)**

โดยจาก software architecture นั้น จะเห็นได้ว่าการเรียกใช้ไลบรารี matplotlib โดยตัว user นั้นจะใช้ interface ในการสั่งการจึงเปรียบ interface ได้กับ cougntroller จากนั้นจะส่งคำสั่งไปยัง front-end ซึ่งก็คือ model จากนั้นจึง จะ update ไปยัง view หรือก็คือ back-end ซึ่งจะทำหน้าที่ในการแสดงผล/renderผลลัพธ์ส่งไปยัง user

Quality Attribute Scenarios

1. Usability

โดย matplotlib นั้นสามารถเรียนรู้ได้รวดเร็ว, ใช้ได้อย่างมีประสิทธิภาพ, สามารถปรับเปลี่ยนตามความต้องการของผู้ใช้ได้, สามารถทำงานได้อย่างง่ายเพราะ code ที่ใช้งานง่าย

2. Modifiability

โดย matplotlib นั้นสามารถแก้ไขดัดแปลงได้เนื่องจากเป็น software ที่เป็นในรูปแบบ open-source

3. Integrability

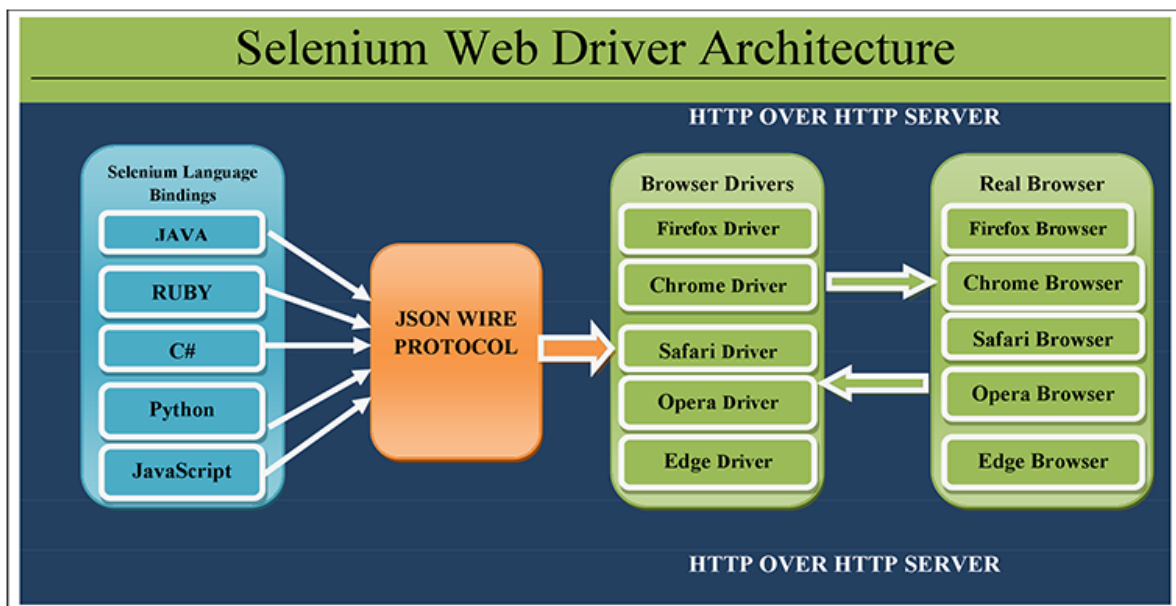
โดย matplotlib นั้นใช้งานร่วมกับไฟล์ชนิด TEX, การเซฟรูป figure เป็นไฟล์รูปชนิดมาตรฐาน

อ้างอิง: [\(PDF\) matplotlib -- A Portable Python Plotting Package \(researchgate.net\)](#)

Selenium WebDriver

เรียกอีกอย่างว่า selenium 2.0 Selenium webdriver สื่อสารโดยตรงกับเบราว์เซอร์ ดังนั้น selenium webdriver จึงเร็วกว่า selenium RC Selenium webdriver รองรับเว็บเบราว์เซอร์หลายตัวและยังรองรับแอปพลิเคชัน Ajax เป้าหมายหลักของซีลีเนียม webdriver คือการปรับปรุงการสนับสนุนสำหรับปัญหาการทดสอบเว็บแอปพลิเคชันที่ทันสมัย Selenium webdriver รองรับหลายภาษาในการเขียนสคริปต์ทดสอบ API ของ Selenium webdriver นั้นง่ายกว่า selenium RC's . อย่างไรก็ตาม แม้จะมีข้อดีทั้งหมดของไดรเวอร์เว็บซีลีเนียม แต่ก็ยังมีข้อจำกัดบางประการในการทดสอบเว็บแอปพลิเคชัน Selenium webdriver ไม่มีฟังก์ชันการทำงานในการสร้างภาพหน้าจอสำหรับกรณีทดสอบความล้มเหลว Selenium webdriver ไม่มีความสามารถในการสร้างผลการทดสอบ ขึ้นอยู่กับเครื่องมือของบุคคลที่สามในการสร้างรายงานการทดสอบ

Design Architecture



จะเป็นรูปแบบ design ของ **Plug-in (Microkernel)**

โดยจะเห็นได้ว่ามีสอง ตัวประกอบหลักๆโดยจะมีชุดการทำงานหลักและตัวแปรเฉพาะที่เพิ่มฟังก์ชันการทำงานให้กับแกน

Quality Attribute Scenarios

1.Testability

สามารถนำเสนอ/สาธิตได้ว่ามีข้อผิดพลาด / ตรวจจับข้อผิดพลาดได้ในการวัด จะวัดประสิทธิภาพ คือ หาข้อผิดพลาดได้ทุกครั้ง, วัดระยะเวลานานแค่ไหนในการทดสอบ, การทดสอบจะครอบคลุมหรือไม่

2.Modifiability

โดย Selenium webdriver รองรับหลายภาษาในการเขียนสคริปต์ทดสอบ API ของ Selenium webdriver และมีการปรับปรุงเรื่อยๆ

3. Availability

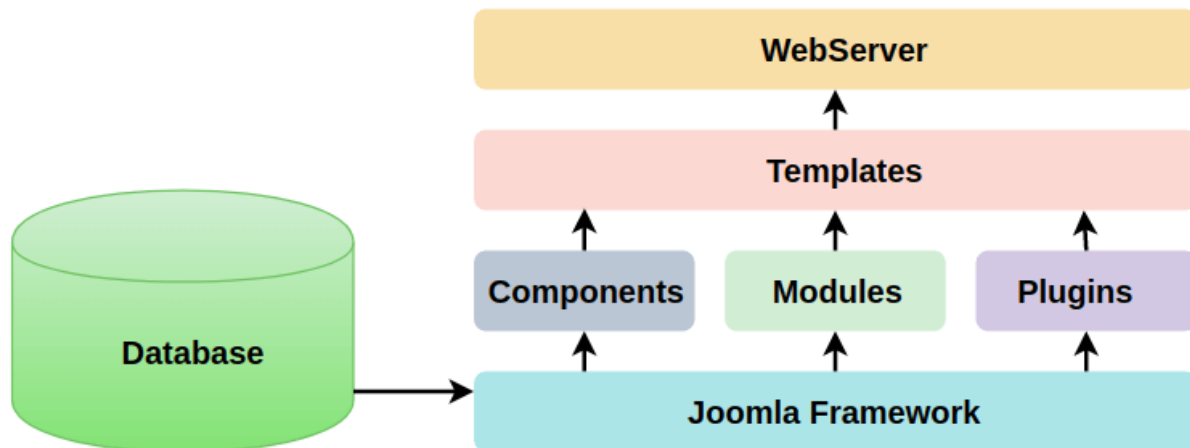
เกี่ยวกับความสนใจที่ระบบไม่พร้อมใช้งาน หรือ ระบบล้ม (Failure) และผลที่เกิดจากระบบล้ม และอาจมีSystem failure เกิดจากการที่ระบบไม่สามารถให้บริการได้ในระยะเวลาหนึ่ง

อ้างอิง: [Analysis and Design of Selenium WebDriver Automation Testing Framework - ScienceDirect](#)

Joomla

Joomla เป็นระบบจัดการเนื้อหาโอเพ่นซอร์ส (CMS) ที่ได้รับความนิยมมากที่สุด ซึ่งทำให้ผู้ใช้สามารถสร้างเว็บไซต์และแอปพลิเคชันออนไลน์ที่ทรงพลังที่สามารถใช้งานได้โดยอิสระ Joomla เขียนด้วย PHP (ภาษาสคริปต์) โดยใช้เทคนิคการเขียนโปรแกรมเชิงวัตถุ (OOP) และฐานข้อมูล MySQL สำหรับส่วนแบ็กเอนด์ Joomla เป็นมากกว่าระบบจัดการเนื้อหาที่ทรงพลัง ด้วยรายการคุณสมบัติ การจัดการผู้ใช้ การจัดการแบนเนอร์ ตัวจัดการสื่อ เทมเพลต ตัวจัดการภาษา ติดต่อ โพล การค้นหาเว็บไซต์ เนื้อหา เมนู การจัดการบริการเว็บและการขยายที่มีประสิทธิภาพ Joomla เก็บเนื้อหาไว้ในฐานข้อมูลเพื่อให้มีการจัดรูปแบบและการแสดงผลแบบไดนามิกสำหรับผู้ใช้ Joomla ถูกใช้ทั่วโลกเพื่อสร้างเว็บไซต์ที่มีประสิทธิภาพในรูปทรงและขนาดทั้งหมด

Design Architecture



Joomla Architecture

จะเป็นรูปแบบ design ของ Layer

โดยจาก software architecture นั้น จะเห็นได้ว่าการแบ่งเป็น layer อย่างชัดเจน โดยแต่ละ layer คือการจัดกลุ่มของ module ที่นำเสนอชุดบริการที่สอดคล้องกัน และความสัมพันธ์ของแต่ละ layer นั้นต้องไปในทิศทางเดียวกัน

Quality Attribute Scenarios

1. Usability

ใช้เวลาน้อยลงในการเขียนโค้ดและลดงานที่น่าเบื่อที่เกี่ยวข้องกับการสร้างอินเทอร์เน็ตเพจใน Joomla 3.9 Joomla เวอร์ชันล่าสุด

มีคุณสมบัติที่มี CSS และ jQuery น้อยกว่า ดังนั้น คุณจึงเขียนโค้ดได้น้อยลง

2. Modifiability

Joomla เป็นการจัดการเนื้อหาที่เป็นมิตรต่อผู้ใช้และใช้กันอย่างแพร่หลายและระบบนั้นที่มีคุณสมบัติที่ปรับแต่งได้ง่าย ในกรณีของการอัปเดต ผู้ดูแลระบบ/นักพัฒนาสามารถอัปเดตเวอร์ชัน joomla ได้อย่างง่ายดายด้วยปุ่มอัปเดตเพียงคลิกเดียว

3. Performance

Joomla เป็นระบบการจัดการเนื้อหาที่เป็นหัวใจสำคัญ และมีคุณสมบัติที่เชื่อมโยงบางอย่างที่ทำให้การจัดระเบียบ และจัดการเนื้อหาของผู้ใช้เป็นเรื่องง่าย

