**Matplotlib**

matplotlib เป็นไลบรารีการวางแผนที่ใช้ Python พร้อมการสนับสนุนอย่างเต็มที่สําหรับ 2D และการสนับสนุนกราฟิก 3 มิติที่ จํากัด ซึ่งใช้กันอย่างแพร่หลายในชุมชนคอมพิวเตอร์วิทยาศาสตร์ Python ไลบรารีมีเป้าหมายเป็นกรณีการใช้งานที่หลากหลาย มันสามารถฝังกราฟิกในชุดเครื่องมือส่วนติดต่อผู้ใช้ที่คุณเลือกและปัจจุบันรองรับกราฟิกแบบโต้ตอบบนระบบปฏิบัติการเดสก์ท็อปที่สําคัญทั้งหมดโดยใช้ชุดเครื่องมือ GTK +, Qt, Tk, FLTK, wxWidgets และ Cocoa สามารถเรียกแบบโต้ตอบได้จากเชลล์ Python แบบโต้ตอบเพื่อสร้างกราฟิกด้วยคําสั่งขั้นตอนที่เรียบง่ายเช่นเดียวกับ Mathematica, IDL หรือ MATLAB matplotlib ยังสามารถฝังอยู่ในเว็บเซิร์ฟเวอร์แบบไม่มีหัวเพื่อให้เอกสารทั้งในรูปแบบแรสเตอร์เช่นกราฟิกเครือข่ายแบบพกพา (PNG) และรูปแบบเวกเตอร์เช่น PostScript รูปแบบเอกสารแบบพกพา (PDF) และกราฟิกเวกเตอร์ที่ปรับขนาดได้ (SVG) ซึ่งดูดีบนกระดาษ

วัตถุประสงค์

เพื่อให้สามารถสร้างภาพพล็อตด้วยความง่ายและสะดวกด้วยการใช้ภาษาpythonในการเขียนซึ่งไม่ยากต่อการเรียนรู้และการใช้งาน พร้อมทั้งสามารถพลิกแพลงเอาไปใช้งานได้อย่างหลากหลาย

**Architectural Patterns/styles**

รูปแบบสถาปัตยกรรมที่ Matplotlib ใช้นั้นเป็นรูปแบบ Layer architectural ซึ่งมีทั้งหมด 3 Layer

A screenshot of a computer

Description automatically generated with low confidence

ที่มา https://medium.com/dataseries/mastering-matplotlib-part-1-a480109171e3

**Backend layer**

เป็นเลเยอร์ชั้นหลังสุดจัดเตรียมการใช้งาน abstract interface classes โดยจะมีทั้งหมด 3 คลาส เป็นชั้นที่มีการทำงานซับซ้อนที่สุด

1.figure canva: คือส่วน matplotlib.backend\_bases.FigureCanvas object: มันจะครอบคลุมพื้นที่ของรูปวาดเป็นobject

2.Renderer: คือส่วน matplotlib.backend\_bases.Renderer object: มันรู้วิธีวาดภาพบนfigurecanva

3.Event: คือส่วน *matplotlib.backend\_bases.Event: มันจะคอยจัดการกับinputของผู้ใช้ เช่น การกดแป้นพิมพ์และการคลิกเมาส์*

**Artist layer**

ประกอบด้วยหนึ่งวัตถุหลัก — FigureArtist:-เป็นlayerที่ใช้ในการออกแบบ ทั้งการเขียนชื่อเรื่อง เส้น ความหนาความบาง รูปร่าง และ อีกมากมาย ที่ใช้ในการปับแต่งองค์ประกอบต่างๆ

1. **ดั้งเดิม:** เช่น เส้น,สีเหลี่ยม,วงกลมและข้อความ
2. **ประกอบขึ้นใหม่:**เช่น แกนx,แกนyและรูปร่างต่างๆ

**เรื่องน่ารู้**

-object matplotlib ระดับบนสุดที่มีและจัดการองค์ประกอบทั้งหมดในกราฟิกที่กำหนดคือ The FigureArtist

-งานartistที่สำคัญที่สุดคือ แกนหลัก เนื่องจากเป็นที่ที่วิธีการวางแผน matplotlib API ส่วนใหญ่ถูกกำหนดไว้ รวมถึงวิธีการสร้างหรือจัดการเส้นแกน ตาราง หรือพื้นหลังการลงจุด

-สิ่งสำคัญคือต้องสังเกตว่างานประกอบขึ้นใหม่แต่ละแบบอาจมีงานอื่น ๆ รวมถึงงานแบบดั้งเดิม ตัวอย่างเช่น งานรูปร่างอาจมี แกนและรูปสี่เหลี่ยมผืนผ้าหรือข้อความอยู่ในนั้น

**The Scripting Layer:**

*เลเยอร์ Artist* ก่อนหน้านี้ค่อนข้างหนักในsyntaxและเหมาะสําหรับนักพัฒนามืออาชีพ ดังนั้น***scripting layer***จึงได้รับการพัฒนาสําหรับนักวิทยาศาสตร์ที่ไม่ใช่โปรแกรมเมอร์มืออาชีพ เห็นได้ชัดว่าผู้คนจํานวนมากชอบ***scripting layer*** มากกว่าartist layerที่มีsyntaxมากเพื่อการวิเคราะห์ข้อมูลบางอย่างอย่างรวดเร็ว

***Scripting layer Matplotlib เป็นอินเทอร์เฟซ matplotlib.pyplot***  ยอดนิยมทั้งหมดซึ่งทําให้กระบวนการกําหนด *FigureCanvas* เป็นไปโดยอัตโนมัติและกําหนดอินสแตนซ์ *FigureArtist* และเชื่อมต่อเข้าด้วยกัน

**Quality Attribute Scenarios**

-Modifiability

Source: Developer

Stimulus: Wishes to modify 3D function

Artifact: Code Environment: Development Time

Response: Modification is made with no side effects Response

measure: In Three hours

-Portability

Source: OS

Stimulus: Wishes to run on another OS

Artifact: Resource

Environment: Runtime

Response: Can run without error occurs Response

measure: In 30 minutes

-Testability

Source: Tester

Stimulus: Performs end to end test

Artifact: Complete application

Environment: At deployment time

Response: Perform a test sequence Response

measure: Path coverage of 85% is achieved within three hours

**แหล่งอ้างอิง**  
<https://www.aosabook.org/en/matplotlib.html>  
<https://medium.com/dataseries/mastering-matplotlib-part-1-a480109171e3>

**Audacity**

Audacity เป็นเครื่องบันทึกเสียงและโปรแกรมแก้ไขเสียงยอดนิยม มันเป็นโปรแกรมที่มีความสามารถในขณะที่ยังคงใช้งานง่าย ผู้ใช้ส่วนใหญ่อยู่ใน Windows แต่ซอร์สโค้ด Audacity เดียวกันคอมไพล์ให้ทํางานบน Linux และ Mac ด้วย

Dominic Mazzoni เขียน Audacity เวอร์ชันดั้งเดิมในปี 1999 ในขณะที่เขาเป็นนักศึกษาวิจัยที่มหาวิทยาลัยคาร์เนกีเมลลอน โดมินิกต้องการสร้างแพลตฟอร์มที่จะพัฒนาและดีบักอัลกอริทึมการประมวลผลเสียง ซอฟต์แวร์เติบโตขึ้นจนมีประโยชน์ในสิทธิของตนเองในรูปแบบอื่น ๆ อีกมากมาย เมื่อ Audacity ได้รับการปล่อยตัวเป็นซอฟต์แวร์โอเพนซอร์สแล้วมันก็ดึงดูดนักพัฒนารายอื่น ทีมผู้ที่ชื่นชอบขนาดเล็กที่ค่อยๆ เปลี่ยนแปลงได้ปรับเปลี่ยน บํารุงรักษา ทดสอบ อัปเดต เป็นลายลักษณ์อักษร สําหรับ ช่วยเหลือผู้ใช้ และแปลอินเทอร์เฟซของ Audacity เป็นภาษาอื่นๆ ในช่วงหลายปีที่ผ่านมา

เป้าหมายหนึ่งคืออินเทอร์เฟซผู้ใช้ควรสามารถค้นพบได้: ผู้คนควรสามารถนั่งลงได้โดยไม่ต้องใช้คู่มือและเริ่มใช้งานได้ทันทีค่อยๆค้นพบคุณสมบัติของมัน หลักการนี้มีความสําคัญอย่างยิ่งในการทําให้ Audacity มีความสอดคล้องกับอินเทอร์เฟซผู้ใช้มากกว่าที่อื่น สําหรับโครงการที่หลายคนมีมือหลักการรวมกันแบบนี้มีความสําคัญมากกว่าที่คิดในตอนแรก

**วัตถุประสงค์**

มีฟีเจอร์ให้ใช้งานที่หลากหลาย ตั้งแต่การอัดเสียง การปรับแต่งเสียง รวมไปถึงการมิกซ์เสียง และโปรแกรมนี้สามารถรองรับการใช้งานในหลายๆ แพลตฟอร์ม เช่น [Linux](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2098-linux-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)Windows Macs

**Architectural Patterns/styles**

รูปนี้เป็นภาพประกอบเวอร์ชันที่แต่งใหม่จาก[สถาปัตยกรรมของโอเพนซอร์ส](http://www.aosabook.org/en/audacity.html)

Diagram

Description automatically generated

Audacityใช้libraryหลายแห่ง libraryหลักสําหรับอินเทอร์เฟซผู้ใช้คือ wxWidgets ซึ่งเป็นแพลตฟอร์มข้าม (Windows, Mac, Linux) แผนภาพด้านบนแสดงให้เห็นว่า Audacity วางเลเยอร์สิ่งที่เป็นabstractเพิ่มเติมไปยังlibraryฐานอย่างไร

1. BlockFile ใช้ระบบไฟล์ระบบปฏิบัติการผ่าน wxWidgets wxFile เพื่อมอบวิธีการจัดเก็บเสียงเป็นชิ้นเล็ก ๆ จํานวนมาก ชิ้นเล็ก ๆ ทําให้ตัดวางและจัดเรียงเสียงใหม่ได้อย่างรวดเร็วโดยไม่ต้องคัดลอกและแก้ไขไฟล์เสียงทั้งหมดสําหรับการเปลี่ยนแปลงเล็กน้อยแต่ละครั้ง
2. ShuttleGui ใช้กล่องโต้ตอบ wxWidgets ปุ่ม และตัวควบคุมอื่นๆ มันจัดระเบียบพวกเขาด้วยโครงสร้างเพิ่มเติมที่ช่วยลดรหัสซ้ํา ๆ ซึ่งส่วนใหญ่ใช้เพื่อ 'รถรับส่ง' ค่าจากที่หนึ่งไปยังอีกที่หนึ่ง สิ่งสําคัญที่สุดคือการตั้งค่าของผู้ใช้เช่นคุณภาพเสียงจะถูกเก็บไว้ในไฟล์ ข้อมูลจะต้องถูกส่งจากไฟล์ไปยังตัวแปรจากตัวแปรไปยังวิดเจ็ตที่แสดงค่าและในทิศทางย้อนกลับด้วย
3. การจัดการคําสั่งใน Audacity ผูกการกดแป้นพิมพ์และรายการเมนูกับคําสั่งภายในภายใน Audacity มันใช้ wxMenu จาก wxWWidgets

แม้ว่า PortAudio จะไม่ได้เป็นส่วนหนึ่งของ wxWidgets แต่ก็มี 'การสะท้อน' ในAudacityเช่นกัน

1. AudioIO จัดการกระบวนการย้ายเสียงระหว่างการ์ดเสียงหน่วยความจําและฮาร์ดดิสก์โดยให้สิ่งที่เป็นนามธรรมเพิ่มเติมเกี่ยวกับการจัดการการ์ดเสียงข้ามแพลตฟอร์มใน PortAudio

**Quality Attribute Scenarios**

**- Modifiability**Source : Developer

Stimulus : Wishes to modify recording function

Artifact : Code

Environment : Development Time

Response : Modification is made with no side effects

Response measure : In Three hours

**- Portability**

Source : OS

Stimulus : Wishes to run on another OS

Artifact : Resource

Environment : Runtime

Response : Can run without error occurs

Response measure : In 30 minutes

**- Testability**

Source : Tester

Stimulus : Performs end to end test

Artifact : Complete application

Environment : At deployment time

Response : Perform a test sequence

Response measure : Path coverage of 85% is achieved within four hours

**แหล่งอ้างอิง**

<https://www.aosabook.org/en/audacity.html>

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

**Jitsi**

Jitsiเป็นแอปพลิเคชั่นที่ช่วยให้ผู้คนสามารถโทรผ่านวิดีโอและเสียงแชร์เดสก์ท็อปของพวกเขาและแลกเปลี่ยนไฟล์และข้อความ ที่สําคัญกว่านั้นคือช่วยให้ผู้คนสามารถทําสิ่งนี้ผ่านโปรโตคอลต่างๆ ได้หลายโปรโตคอล ตั้งแต่ XMPP ที่ได้มาตรฐาน (การส่งข้อความที่ขยายได้และโปรโตคอลการแสดงตน) และ SIP (โปรโตคอลการเริ่มต้นเซสชัน) ไปจนถึงโปรโตคอลที่เป็นกรรมสิทธิ์เช่น Yahoo! และ Windows Live Messenger (MSN) มันทํางานบนไมโครซอฟท์วินโดวส์, แอปเปิ้ล Mac OS X, ลินุกซ์, และ FreeBSD. ส่วนใหญ่เขียนด้วยภาษา Java แต่ยังมีส่วนที่เขียนด้วยnative code

**วัถตุประสงค์**

เพื่อทำให้สามารถใช้เป็นตัวกลางในการประชุมทางไกลหรือที่เรียกว่า video conferencing serviceได้ และสามารถรองรับการสนทนาได้หลายคนและสามารถบันทึกภาพและเสียงในการสนทนาได้ พร้อมทั้งยังสามารถแชร์หน้าจอตัวเองออกมาได้อีกด้วยแล้วผู้ใช้งานไม่จำเป็นต้องสร้างบัญชีเพื่อการใช้งานอีกด้วย

**Architectural Patterns/styles**

Diagram

Description automatically generated

jitsiมีมานานพอๆ กับ WebRTC โดย Jitsi Videobridge เปิดตัวครั้งแรกในปี 2013 ตลอดหลายปีที่ผ่านมามันมีขนาดและคุณสมบัติเพิ่มขึ้น แผนภาพด้านล่างแสดงสถาปัตยกรรมทางเทคนิคของการปรับใช้ Jitsi ทั่วไป

ส่วนประกอบหลักของjitsiคือ:

1. [Jitsi Videobridge (JVB)](https://jitsi.org/jitsi-videobridge/) ซึ่งเป็นหัวใจสําคัญของบริการJitsiถูกจําลองเป็น [SFU](https://bloggeek.me/webrtcglossary/sfu/) ซึ่งทําหน้าที่เป็นเซิร์ฟเวอร์การกําหนดเส้นทางอัจฉริยะที่สามารถโฮสต์แฮงเอาท์วิดีโอแบบกลุ่มได้ในวงกว้าง
2. [Prosody](https://prosody.im/) เซิร์ฟเวอร์ส่งสัญญาณที่ใช้โดย Jitsi ซึ่งใช้ [XMPP](https://bloggeek.me/webrtcglossary/xmpp/) เป็นโปรโตคอลการส่งสัญญาณ สิ่งนี้ถูกใช้เป็นตัวกลางที่ช่วยให้ผู้ใช้สามารถเข้าร่วมห้องและสื่อสารกันในจิ้ดสิ
3. [Jitsi Meet](https://jitsi.org/jitsi-meet) การใช้งานเว็บอินเตอร์เฟสส่วนหน้า สิ่งนี้ค่อนข้างไม่เหมือนใครสําหรับจิ้ดสิเนื่องจากมอบประสบการณ์การประชุมที่ทํางานได้อย่างสมบูรณ์และพร้อมที่จะปรับใช้ มันถูกสร้างขึ้นโดยใช้ปฏิกิริยาและตอบสนองพื้นเมือง.
4. [Jitsi Conference Focus (jicofo)](https://github.com/jitsi/jicofo) ซึ่งเป็นโหลดบาลานเซอร์ที่จัดการและจัดการการปรับขนาดแนวนอนสําหรับจิ้ดสิ
5. [Jitsi Broadcasting Interface (jibri)](https://github.com/jitsi/jibri) ซึ่งเป็นบริการบันทึก/สตรีมมิงสําหรับ JVB โดยพื้นฐานแล้วสิ่งนี้สร้างขึ้นจาก Chrome และ ffmpeg แบบไม่มีส่วนหน้าเพื่อนําเสนอสตรีมวิดีโอเดียวจากการประชุมทางวิดีโอของ Jitsi ที่กําลังดําเนินอยู่
6. [Jitsi Gateway to SIP (jigasi)](https://github.com/jitsi/jigasi) เกตเวย์ที่Jitsiนําเสนอสําหรับผู้ที่ต้องการเชื่อมต่อบริการโทรศัพท์เข้ากับการประชุมที่Jitsi

**Quality Attribute Scenarios**

- Usability

Source: Users

Stimulus: Use system efficiently

Artifact: System

Environment: Runtime

Response: Wishes to record audio and video Response

measure: Video and audio recording takes less than a second

- Modifiability

Source: Developer

Stimulus: Wishes to add screen sharing function

Artifact: Code

Environment: Development time

Response: Modifications were made without side effects Response

measure: In Six hours

– Portability

Source: iOS

Stimulus: Wishes to run on iOS

Artifact: Resource

Environment: Runtime

Response: Can run on iOS without error occurs Response

measure: In 30 minutes

แหล่งอ้างอิง

<https://www.aosabook.org/en/jitsi.html>  
<https://www.infoq.com/articles/mobile-video-conferencing-jitsi>