

GPSD

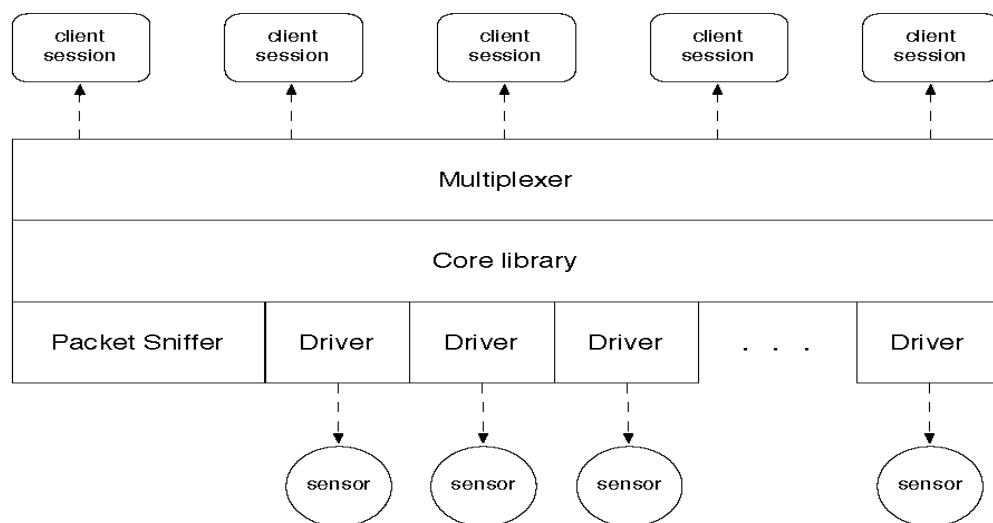
Purpose of the project

GPSD คือ ชุดเครื่องมือสำหรับการจัดการชุดของอุปกรณ์GPS และ Sensorอื่นๆที่เกี่ยวข้องกับระบบนำทาง และ ระบบจับเวลาที่ รวมถึง ระบบวิทยุAIS และ เซ็มทิศดิจิตอล

GPSD ถูกพัฒนาขึ้นมาเพราะแอปพลิเคชันที่มาพร้อมกับGPSและเซ็นเซอร์ที่เกี่ยวข้องกับการนำทางอื่นๆ ที่ได้รับการออกแบบมาแย่ง, เอกสารคู่มือการใช้งานไม่ดี , ประเภทของอุปกรณ์sensorและรุ่นของมันก็มีหลากหลายมาก และโปรโตคอลต่างๆที่ผู้ขายพากันสร้างขึ้นมาก็มีมากมายเกินไปจนเกิดความยุ่งเหยิง ถ้าหากแอปพลิเคชันต้องเป็นคนที่รับมือกับความซับซ้อนพวกนี้เอง ผลลัพธ์ก็จะทำให้มีโค้ดการทำงานที่ซ้ำซ้อน ทำให้เพิ่มโอกาสที่ผู้ใช้งานจะสังเกตเห็นBugได้ และต้องคอยรับมือกับการเปลี่ยนแปลงของHardwareในอนาคตอยู่เสมอๆด้วย

Architectural patterns/styles

ใช้แบบ Layers โดยแบ่งออกเป็น 4 ส่วน คือ the drivers, the packet sniffer, the core library and the multiplexer



The drivers นั้นเป็นไดรเวอร์อุปกรณ์พื้นที่ผู้ใช้สำหรับชิปเซ็ต sensor แต่ละประเภทที่สนับสนุน /*จุดเริ่มต้นที่สำคัญคือวิธีการแยกแยะเกิดข้อมูลเป็นข้อมูลตำแหน่งเวลา-ความเร็วหรือสถานะ เปลี่ยนโหมดหรือ baud rate โพรบสำหรับประเภทย่อย

ของอุปกรณ์ ฯลฯ วิธีการเสริมอาจสนับสนุนการดำเนินการควบคุมไดรเวอร์ เช่น การเปลี่ยนความเร็วรอบของ อุปกรณ์. อินเทอร์เฟซทั้งหมดไปยังไดรเวอร์เป็นโครงสร้างภาษา C ที่เต็มไปด้วยข้อมูลและพอยน์เตอร์เมธอด ซึ่งจำลองขึ้นโดยเจตนาบนโครงสร้างไดรเวอร์อุปกรณ์ Unix*/

the packet sniffer มีหน้าที่ในการขุดแพ็กเก็ตข้อมูลออกจากสตรีมอินพุตแบบอนุกรมโดยพื้นฐานแล้วมันเป็นเครื่องของรัฐที่คอยเฝ้าดูทุกสิ่งที่ดูเหมือนหนึ่งใน 20 ประเภทแพ็กเก็ตที่เรารู้จักกัน เนื่องจากอุปกรณ์สามารถเสียบปลั๊กหรือเปลี่ยนโหมดได้ ประเภทของแพ็กเก็ตที่จะดึงมาจากพอร์ตอนุกรมหรือพอร์ต USB ไม่จำเป็นต้องได้รับการแก้ไขอย่างถาวรโดยตัวแรกที่รู้จัก

the core library จัดการ session ด้วยอุปกรณ์ sensor หลักๆคือ

- เริ่มต้น session โดยเปิดอุปกรณ์และอ่านข้อมูลจากอุปกรณ์ ไล่ตาม baud rate และ parity/stopbit ผสมกัน จนกว่า packet sniffer จะล็อกการซิงโครไนซ์กับประเภทแพ็กเก็ตที่รู้จัก
- การสำรวจอุปกรณ์สำหรับแพ็กเก็ต
- ปิดอุปกรณ์และปิด session

คุณลักษณะหลักของ core library คือมีหน้าที่ในการเปลี่ยนการเชื่อมต่อ GPS แต่ละครั้งเพื่อให้ไดรเวอร์อุปกรณ์ที่ถูกต้อง โดยขึ้นอยู่กับประเภทแพ็กเก็ตที่the sniffer ส่งกลับ ค่านี้ไม่ได้กำหนดค่าไว้ล่วงหน้าและอาจเปลี่ยนแปลงได้เมื่อเวลาผ่านไป โดยเฉพาะอย่างยิ่งหากอุปกรณ์สลับไปมาระหว่างการรายงานของโปรโตคอลที่แตกต่างกัน

the multiplexer มัลติเพล็กซ์เซอร์เป็นส่วนหนึ่งของ daemon ที่จัดการ session ไคลเอ็นต์และการกำหนดอุปกรณ์ มีหน้าที่ส่งรายงานไปยังไคลเอ็นต์ ยอมรับคำสั่งไคลเอ็นต์และตอบสนองต่อการแจ้งเตือน hotplug โดยพื้นฐานแล้วจะมีอยู่ในไฟล์ต้นฉบับหนึ่งไฟล์ gpsd.c และไม่เคยพูดคุยกับไดรเวอร์อุปกรณ์โดยตรง

Testability scenario

Source of Stimulus	User, Developer
Stimulus	คำสั่ง gpsfake

Artifacts	Gpsd
Environment	the complete implement of whole system
Response	sensor log shipped with a bug report Regression-test result
Response measure	สามารถหา Bug ได้ และ spot จุดที่พังได้ สามารถหาส่วนที่เปลี่ยนแปลงแล้วไป config กับส่วนอื่นๆได้

Reliability scenario

Source of Stimulus	Unit tester
Stimulus	Change random code to break our device driver
Artifacts	Code
Environment	Unit test Time
Response	program deactivate
Response measure	Time for restore code or rollback version

Usability scenario

Source of Stimulus	End user
--------------------	----------

Stimulus	end users plug a USB GPS into their laptop
Artifacts	Gpsd
Environment	Normal Operation
Response	location Reports
Response measure	ส่ง location Reports ทันทีที่เสียบ USB ไม่มี muss ไม่ยุ่งยาก

<https://aosabook.org/en/gpsd.html>

Matplotlib

Purpose of the project

Matplotlib คือ เฟรมเวิร์กในการพลอตกราฟข้อมูล 2D และ แสดงภาพ ที่เน้นในการทำ Visualization ข้อมูลทาง วิทยาศาสตร์ วิศวกรรม และการเงิน สามารถใช้งานผ่าน Python shell, ผ่านการเรียกใช้งานด้วยการเขียนPython Script หรือ ใช้งานผ่านApplication GUIก็ได้(GTK, Wk, Tk, Windows) สามารถExportกราฟได้หลายรูปแบบเช่น JPEG, PNG, PostScript และSVG

Matplotlib ถูกดีไซน์ขึ้นมาด้วยจุดประสงค์ที่ว่าผู้ใช้งานจะต้องสามารถสร้างกราฟง่ายๆ โดยใช้คำสั่งเพียงไม่กี่คำสั่ง หรือ เพียงแค่1คำสั่งเท่านั้น

Architectural patterns/styles

Matplotlib ใช้ Architecture แบบ Layers ซึ่งแบ่งออกเป็น 3 Layersหลัก

Backend Layer

จัดการงานหนักทั้งหมดผ่านการสื่อสารไปยังชุดเครื่องมือ เช่น wxPython หรือ drawing language เช่น PostScript ใน เครื่องของคุณ เป็นเลเยอร์ที่ซับซ้อนที่สุดของไลบรารี Matplotlib

มีคลาสอินเตอร์เฟซนามธรรมในตัวสามคลาสหลัก:

FigureCanvas — matplotlib.backend_bases.FigureCanvasBase Canvasgเป็นที่Figureถูกสร้างขึ้น

Renderer — matplotlib.backend_bases.RendererBase Abstract classสำหรับการจัดการการวาด/การเรนเดอร์ รับผิดชอบการ วาดภาพใน FigureCanvas

เหตุการณ์ — matplotlib.backend_bases.Event จัดการอินพุตของผู้ใช้ เช่น การกดแป้นพิมพ์และเมาส์

Artist Layer

ช่วยให้สามารถควบคุมและปรับแต่งองค์ประกอบต่างๆ ของภาพ เช่นเดียวกับที่ศิลปินวาดภาพบนผืนผ้าใบ

Layerนี้ประกอบด้วยออบเจกต์หลักหนึ่งรายการ ใช้ `Renderer` เพื่อวาดภาพบนCanvas Layerนี้ช่วยให้คุณปรับแต่งได้มากกว่าเมื่อเปรียบเทียบกับเลเยอร์ `Scripting` และสะดวกกว่าสำหรับการพล็อตขั้นสูง

โดยเฉพาะอย่างยิ่งเมื่อต้องการจัดการกับตัวเลขเยอะๆ หรือ แกนหลายๆแกน คุณจะไม่ต้องสับสนว่าตัวใดกำลังทำงานอยู่ เนื่องจากทุก `Subplot`ถูกAssignให้กับArtist Object

นั่นเป็นสาเหตุที่บางครั้งLayerนี้บางครั้งก็ถูกเรียกว่า `object-based plotting`

มีการใช้บ่อยขึ้นเมื่อเขียนเว็บแอปพลิเคชันหรือแอปพลิเคชัน UI หรือเขียนScriptเพื่อแชร์กับDevคนอื่น

ทุกอย่างที่มองเห็นได้บนฟิสิกเกอร์ Matplotlib เช่น title เส้น เครื่องหมายขีด รูปภาพ และอื่นๆ เป็น Artist instance ซึ่งสอดคล้องกับ Artist objectแต่ละอัน

Scripting Layer

เป็นLayerบนสุดที่ออกแบบมาเพื่อให้ Matplotlib ทำงานเหมือน MATLAB script เป็นชุดของฟังก์ชันรูปแบบคำสั่งและถือเป็นเลเยอร์ที่ง่ายที่สุดในการใช้งาน

Artist layerเขียนโค้ดเยอะเนื่องจากมีไว้สำหรับนักพัฒนา และเหมาะกับคนที่ต้องการที่จะทำการวิเคราะห์ข้อมูลบางส่วนแบบเร็วๆ

นี่คือเหตุผลที่Matplotlib tutorial จำนวนมาเริ่มสอนจากเลเยอร์นี้ เพราะเป็นส่วนที่ง่ายที่สุดในการเริ่มต้นและใช้งาน

Scripting layer plottingบางครั้งก็ถูกเรียกว่า `Procedural plotting`

Scripting Layer

matplotlib.pyplot

Artist Layer

matplotlib.artist

Backend Layer

matplotlib.backend_bases

<https://medium.datadriveninvestor.com/data-visualization-with-python-matplotlib-architecture-6b05af533569>

<https://aosabook.org/en/matplotlib.html>

Performance Scenario

Source of Stimulus	User, Developer
Stimulus	คำสั่งplotกราฟ
Artifacts	Matplotlib script
Environment	Normal Operation
Response	กราฟ
Response measure	ระยะเวลาตั้งแต่สั่งrunจนได้ผลลัพธ์เป็นรูปกราฟ

Useability Scenario

Source of Stimulus	End user
Stimulus	เลื่อนปรับการแสดงผลของกราฟ(ย่อขยาย)
Artifacts	Matplotlib Gui
Environment	runtime
Response	กราฟที่ย่อขยาย
Response measure	เวลาที่ใช้ในการปรับ

Testability Scenario

Source of Stimulus	End user
Stimulus	ทำ Regression Testing
Artifacts	Matplotlib Gui
Environment	testing framework
Response	generate a report
Response measure	สามารถเช็คได้ว่า tests ชุดไหน failed บ้าง

Joomla

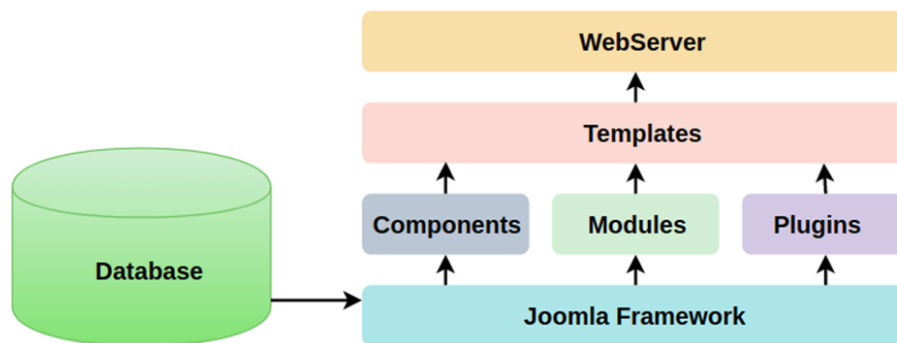
Purpose of the project

Joomla เป็นหนึ่งในระบบจัดการเนื้อหาโอเพนซอร์ซที่ดีที่สุด (CMS) ที่สามารถใช้สร้างเว็บไซต์ที่ทรงพลังและแอปพลิเคชันออนไลน์ได้ เป็นเทมเพลตฟรี ขยายได้ และแยกออกเป็นเทมเพลตส่วนหน้าและส่วนหลัง (เรียกใช้จากฝั่งผู้ดูแลระบบ) มันถูกสร้างขึ้นบนเฟรมเวิร์ก model-view-controller ซึ่งสามารถใช้งานได้โดยไม่ขึ้นกับ Content Management System (CMS)

Joomla เป็นแพลตฟอร์มที่เกิดจากการโต้เถียงกันระหว่างสมาชิกของคณะกรรมการประสานงานของมูลนิธิแมมโบ้ ด้วยเหตุนี้ การเกิดของ 'Joomla' ที่ได้สร้างเอนทิตีใหม่ที่เรียกว่า Open Source Matter

Architectural patterns/styles

Joomla เขียนด้วย PHP และขึ้นอยู่กับรูปแบบการออกแบบ MVC (Model-View-Controller) โดยใช้ MySQL เพื่อจัดเก็บข้อมูล มีคุณสมบัติต่าง ๆ ซึ่งทำให้ Joomla เป็นที่นิยมสำหรับเป็นระบบการจัดการเนื้อหา หรือ CMS (Content Management System)



Joomla Architecture

สถาปัตยกรรมของ Joomla ประกอบด้วยเลเยอร์ต่อไปนี้:

- o ฐานข้อมูล (Database)
- o Joomla Framework
- o Components
- o Modules
- o Plugin
- o Templates
- o Web Server

ฐานข้อมูล (Database)

ฐานข้อมูลประกอบด้วยข้อมูลโดยเฉพาะไฟล์ภาพและเอกสารที่สามารถจัดเก็บจัดการและจัดระเบียบในลักษณะเฉพาะ มันรวมถึงข้อมูลผู้ใช้เนื้อหาและข้อมูลที่จำเป็นอื่น ๆ ของไซต์ นอกจากนี้ยังมีข้อมูลการดูแลระบบเพื่อให้ผู้ดูแลระบบสามารถเข้าถึงไซต์และจัดการได้อย่างปลอดภัย เลเยอร์ฐานข้อมูลของ Joomla เป็นหนึ่งในปัจจัยที่สำคัญที่สุดที่รับประกันความยืดหยุ่นและความเข้ากันได้สูงสุดสำหรับในส่วขยาย

Joomla Framework

Joomla Framework ประกอบด้วยคอลเลกชันของไลบรารีซอฟต์แวร์ โอเพนซอร์สหรือแพ็คเกจ ซึ่งระบบการจัดการเนื้อหา Joomla สร้างขึ้น ไม่จำเป็นต้องติดตั้ง Joomla Framework เพื่อใช้ CMS หรือในทางกลับกัน Joomla Framework มีกลุ่มไฟล์ที่มีประโยชน์ในการสร้างทั้งเว็บและแอปพลิเคชัน command-line มันแบ่งเฟรมเวิร์คออกเป็นแพ็คเกจโมดูลาร์เดี่ยวและยิ่งไปกว่านั้นยังช่วยให้แต่ละแพ็คเกจพัฒนาได้ง่ายขึ้น

Components

ส่วนประกอบต่าง ๆ เรียกว่ามินิแอปพลิเคชันซึ่งประกอบด้วยสองส่วน คือ

- o Administrator
- o Site

เมื่อใดก็ตามที่โหลดหน้าเว็บส่วนประกอบจะถูกเรียกเพื่อแสดงเนื้อหาของหน้าหลัก ส่วนผู้ดูแลระบบจะจัดการแง่มุมต่างๆ ของส่วนประกอบ และส่วนไซต์ ช่วยในการแสดงเพจเมื่อผู้เยี่ยมชมไซต์ทำการร้องขอ ส่วนประกอบเป็นที่รู้จักกันว่าเป็นหน่วยการทำงานที่สำคัญของ Joomla

Modules

โมดูลสามารถกำหนดเป็นส่วนขยายที่ใช้ในการแสดงหน้าใน Joomla ใช้เพื่อแสดงข้อมูลใหม่จากส่วนประกอบ พวกเขาสามารถยืนได้ด้วยตัวเองและได้รับการจัดการโดย 'ตัวจัดการโมดูล' ซึ่งเป็นส่วนประกอบที่เป็นส่วน ๆ เช่นโมดูลเข้าสู่ระบบ นอกจากนี้ยังช่วยในการแสดงเนื้อหาและรูปภาพใหม่เมื่อโมดูลเชื่อมโยงกับส่วนประกอบ Joomla

Plugin

ปลั๊กอินสามารถอธิบายได้ว่าเป็นส่วนขยาย Joomla ที่ยืดหยุ่นและทรงพลังมาก ซึ่งใช้เพื่อขยายเฟรมเวิร์ค ปลั๊กอินเป็นรหัสไม่กึ่งรหัสที่ดำเนินการในบางโอกาสของทริกเกอร์เหตุการณ์ที่เฉพาะเจาะจง โดยทั่วไปจะใช้เพื่อจัดรูปแบบผลลัพธ์ของส่วนประกอบหรือโมดูลเมื่อมีการพัฒนาเพจ ฟังก์ชันปลั๊กอินที่เกี่ยวข้องกับเหตุการณ์มักจะดำเนินการตามลำดับเมื่อใดก็ตามที่มีเหตุการณ์ใดเหตุการณ์หนึ่งเกิดขึ้น

Templates

เทมเพลตใช้เพื่อจัดการรูปลักษณ์ของเว็บไซต์ Joomla โดยทั่วไปมีเทมเพลตสองประเภทให้เลือก Front-end และ Back-end เทมเพลต Front-end เป็นวิธีจัดการรูปลักษณ์ของเว็บไซต์ซึ่งผู้ใช้เห็นได้ เทมเพลต Back-end ใช้เพื่อจัดการหรือควบคุมฟังก์ชันโดยผู้ดูแลระบบ เทมเพลตเป็นวิธีที่ง่ายที่สุดในการสร้างหรือกำหนดไซต์เอง ใช้เพื่อเพิ่มความยืดหยุ่นสูงสุดเพื่อให้ไซต์ของคุณดูน่าสนใจ

Web Server

เป็นเซิร์ฟเวอร์ที่ใช้เชื่อมต่อผู้ใช้กับไซต์ มีหน้าเว็บให้กับลูกค้า HTTP (HyperText Transfer Protocol) ถูกใช้เพื่อสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์

Useability Scenario

Source of Stimulus	Website admin, developer
Stimulus	ต้องการที่จะเรียนรู้วิธีการใช้ Joomla
Artifacts	Joomla
Environment	Normal Operation
Response	ทำงานได้ตามที่userต้องการ
Response measure	เวลาที่ใช้ในการเรียนรู้

Security Scenario

Source of Stimulus	User, hacker
Stimulus	Unauthorized request for access
Artifacts	Joomla database
Environment	Normal operation
Response	ปฏิเสธการเข้าถึง
Response measure	จำนวนครั้งที่ป้องกัน Unauthorized access ได้

Availability scenario

Source of Stimulus	user
Stimulus	Request จำนวนมากๆ
Artifacts	Joomla WebServer
Environment	Production
Response	ตอบrequest ได้ทัน(inform operator)
Response measure	Availability Percentage