EPL Filament Unit Multiplexing Engine (FUME)

CS EPL FUME
Portland State University
Computer Science Capstone Program
Winter - Spring 2024

Table of Contents:

1. Metadata

1.1 General Metadata:

- **Title:** Filament Unit Multiplexing Engine (FUME)
- **Date:** Winter-Spring 2024 (WS24)
- **University:** Portland State University
- **College:** Maseeh College of Engineering and Computer Science
- **Department:** Computer Science Department
- **Program:** Computer Science Capstone Program
- **Supervisor:** Bruce Irvin
    - rbi@pdx.edu, 503-725-6109
- **Sponsor:** Ed Ivory
    - edivory@pdx.edu
- **Other Techincal Contact in EPL:** Benjamin Crall
    - bcrall@pdx.edu

1.2 Team-specific Metadata:

Team members:

- **Polina Kuleshova**: Capstone team lead
    - **Jared Foos**: Parsing + Database group lead
        - **Christian Torralba**: Parsing + Database group member
    - **Samuel Dainard**: Queue + Terminal group lead
        - **Victor Castro:** Queue + Terminal group member
    - **Tazwell Borquist**: OctoPrint + ME group lead
        - **Hannah Martens:** OctoPrint + ME group member
        - **Matthew Newman:** OctoPrint + ME group member

1.3 Project Metadata:

Repository:

- The code for this project is on the EPL GitHub
    - https://github.com/psu-epl/FUME

Documentation::

- Dev and project documentation pdf in the same repository mentioned above
- REAME.md in the repository for steps to run

2. High-level Overview

2.1 Problem Statement:

The FUME project is a part of the larger goal to automate the 3d print farm environment automated system for accepting, managing, and routing 3D printing jobs within the Electronics Prototyping Lab's (EPL) print farm. This system will optimize printer utilization, reduce manual intervention, and provide real-time monitoring and reporting to enhance overall efficiency and productivity.

2.2 Product Description

Our product accepts print jobs submitted by the EPL PARQE system, determines the most suitable printer based on specific criteria, and starts the print when a compatible printer is available. If a filament change is needed, it displays a “Needs human intervention” error.

2.3 Clients and Usage:

EPL staff will use our product directly through a graphical interface to manage printer status, the job queue, and filaments and printers in the print farm. EPL users will indirectly use the product, as our software is part of the pipeline to manage their print jobs in the background whenever they submit them.

3. User Storries

3.1 EPL Staff User Stories

Title: Graphical Representation of Print Farm
Description: As an EPL staff member, I can see all the printers and the print farm and their current status.

Title: Modify Printers/Filaments
Description: As an EPL staff member, I can modify the filament and printer info via the FUME webpage.

Title: View Print Queue
Description: As an EPL staff member I can view the queue of upcoming 3D Print jobs.

Title: Filament Run Out
Description: As an EPL staff member, I receive an error message on the main desktop when the filament runs out during a print.

Title: Error During Print
Description: As an EPL staff member, I receive an error message on the main desktop when OctoPrint detects an error during a print.

Title: Emergency Stop
Description: As an EPL staff member, if I want or need to stop the print for any reason, I can use the emergency stop button on the FUME webpage.

3.2 PARQE User Stories

Title: One Print Job
Description: As the PARQE system, I can send a job to the FUME system, and that job will be printed when a printer is available.

Title: Multiple Print Jobs
Description: As the PARQE system, I can send a dozen jobs to the FUME system, and those jobs will be printed in the order they were sent when a printer is available.

Title: Print Job with Multiple Quantities
Description: As per the PARQE system, I can send a job with a quantity of two, and the print job will be printed twice.

4. Functionality

4.1 Features Implemented

- POST requests were received and parsed from the CS PARQE capstone team.
    - JSON packets and Gcode were contained within the requests.
- Print jobs with a quantity greater than one were handled.
- A database that included current printers and filaments available was maintained.
    - Additionally, a way to update these was provided.
- Appropriate printers for a print job were determined.
- All print jobs were stored in a database, which allowed for queue functionality.
- A graphical interface was created to allow EPL staff members to see the queue of upcoming print jobs.
- Communication with Octoprint to allow jobs to print
- Have the product working independently of the ME FUME group:
    - If the next job in the queue needed a filament change, an error was thrown on the webpage.
- Also, the capability for ME integration was provided:
    - A script for easy future integration with the ME FUME system, which uses serial communication, was created.

4.2 Rejected Features

We rejected one feature: determining how much filament remains (potentially with the QR code on a spool). This was rejected because the sensors will let us know when they're empty so it's not necessary. Additionally, estimating the remaining filament would lack accuracy and could lead to wasted material. Although this feature has potential for future capstone projects, it is not currently required.

4.3 Technologies:

Technologies and languages used:

- Docker
- Flask
- Gcode
- Jinja
- Postgres
- Python
- OctoPrint
- OctoRest
- SQLAlchemy

4.4 Constraints:

The sponsor did not impose any constraints. Given that the OctoPrint backend is predominantly written in Python, we have chosen to go with Python for the backend of our project. Additionally, we decided to work within the existing PARQE framework as much as possible to ensure integration; this means having our OctoPrint side resemble PARQE's OctoPrint API.

4.5 Product Delivery:

We delivered a GitHub repository, which will reside on the EPL GitHub. The repository contains all our code and documentation.

4.6 References:

Given that we are interfacing with OctoPrint, we needed to ensure that our code works with the existing interface. Additionally, the Gcode flavor we are using is Marlin. Reference to those is provided below:

- https://github.com/dougbrion/OctoRest
- https://github.com/MarlinFirmware

Since we are collaborating with the PARQE and ME FUME teams, we prioritized future integration when writing our software. The PARQE and ME FUME teams do not directly interact, nor do they need to know more than the vague idea of what each of these teams does. Our capstone serves as the intermediary between both teams. A link to the latest PARQE repository is provided below. Additional questions regarding the PARQE or ME FUME team can be directed to our team sponsor, who also sponsors their teams.

- https://github.com/Monarch-Capstone-Parqe/CapstoneParqe

5. Architecture

5.1 Components:

- Database
    - This component manages the storage and retrieval of data, which includes print jobs, printers, and filaments. It provides a structured and efficient way to store, organize, and access information used by the software.
- Web UI Frontend
    - This component handles a basic graphical interface that builds off the PARQE interface and allows EPL staff to see the queue, error messages, a print status page, and a way to update filaments and printers available.
- DB and Communication Backend
    - This component receives and parsies print jobs received fromt PARQE qand manages printer and filament settings
- Printer Interface
    - This component provides high-level functions for interacting with printers and manages FUME ME communications and functions

5.2 Diagrams:

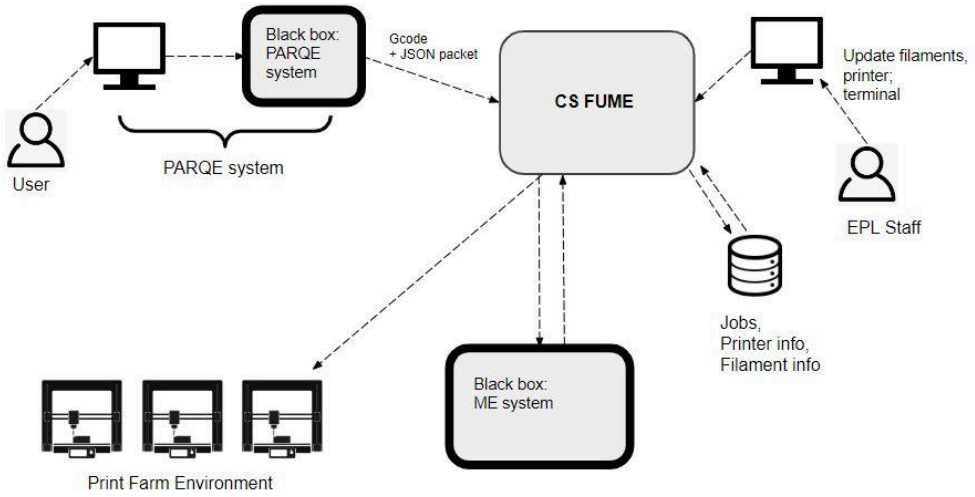

Figure 4.0: High-level Overview of Architecture

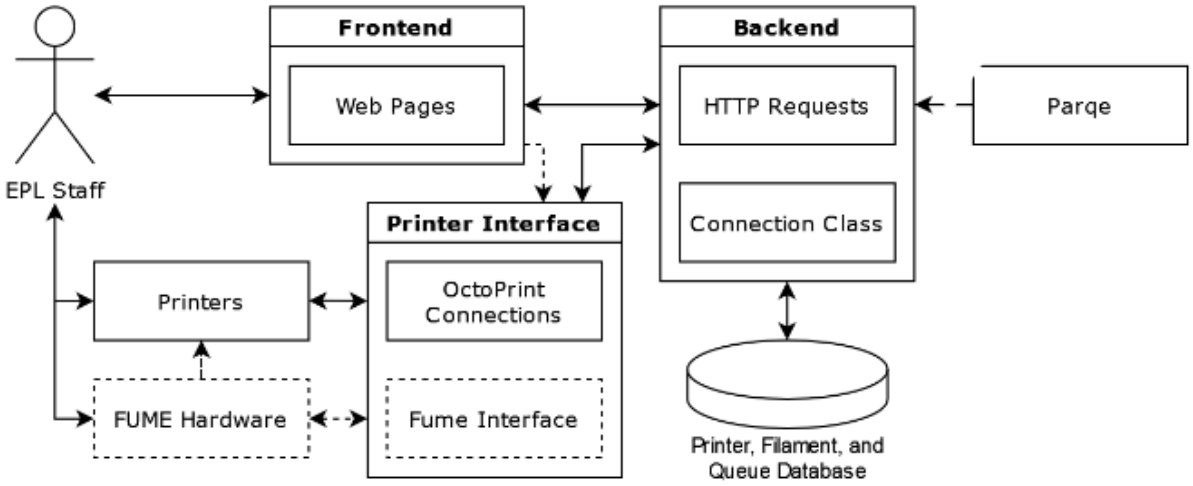

Figure 4.1: Architecture Diagram with Component Labels

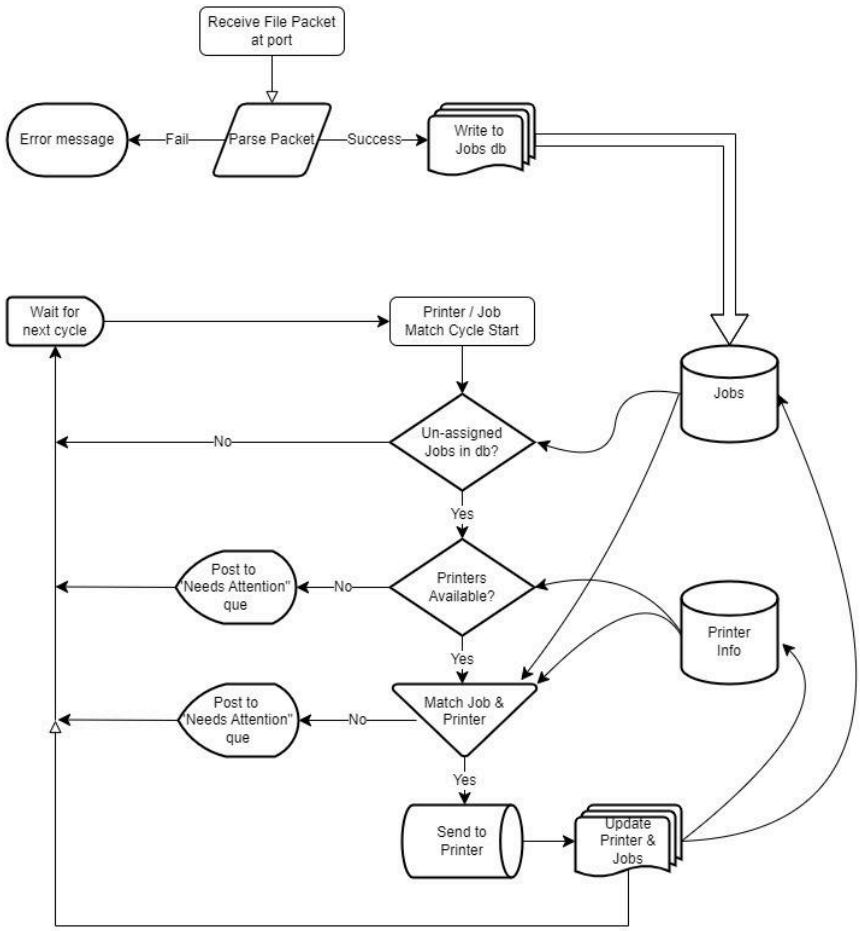

Figure 4.2: Diagram of Component Interactions

6. Process

6.1 Development:

The software development process our team followed was a simplified version of the Agile process. The simplified Agile was implemented by following two-week sprint cycles, where we had sprint planning meetings at the beginning and discussed, re-evaluated, and made adjustments as necessary for the next sprint.

6.2 Tools Used:

To facilitate project workflow and support our sprint methodology, we utilized Jira for backlog management, tracking tasks allocated for the current sprint, and identifying roadblocks encountered during sprint execution.

6.3 Team Roles / Work Allocation:

Since we started from the ground up on this project, the team lead divided the project into three groups, each working on a different section. Group assignments were chosen with early research, prior expertise, and personal interest in mind. Group leads were responsible for further breaking down how their work was allocated. The Capstone team lead was responsible for ensuring the workload for each group was achievable and logical for the time available and the number of group members. Throughout the project, if additional assistance was needed, team members from other groups stepped in to assist.

- The PARQE/DB Group
    - Consists of Jared Foos as the group leader and Christian Torralba as a group member
    - This group's work revolves around processing the POST requests received from the CS PARQE capstone team, determining which printers work for a print job, and creating the database and views.
- UI Group
    - Consists of Samuel Dainard as the group leader and Victor Castro as a group member

    - This group is responsible for creating a graphical interface that allows EPL staff to see printer status, the queue of upcoming jobs, and edit printers/filaments available.

- Octo/ME Group
    - Consists of Tazwell Borquist as the group leader and Hannah Martens and Matthew Newman as group members.
    - This group's work revolves around having our system integrate with the ME FUME systems and communicating with the ME group and Octoprint to ensure print job completion.

6.4 Licensing Considerations:

The license used was the MIT open-source license, which allows us to freely use, modify, and distribute software, with minimal restrictions and liability.

6.5 Risks:

One risk associated with this project is its interdisciplinary nature, especially with the absence of an ECE capstone. The absence of an ECE capstone will affect the ME FUME group's progress and, consequently, our project.

Another risk is that the ME FUME team may not have hardware developed with enough time for us to integrate and test with. To mitigate this, we're developing adaptable systems and leveraging Agile methodologies for flexible collaboration with the ME FUME team. This includes creating a script tailored to their specified integration requirements.

6.5 Testing:

Our software allows us to simulate POST requests from PARQE to allow us to test independently. More details on that are in the Dev documentation in our repo. The script we created to communicate with the ME FUME team has not yet been tested, however, our plan for that is included in the backlog.

7. Schedule

Dates	Weeks of Term	Activity	Sprint #
1/8/24 - 1/19/24	Weeks 1-2	Team Formation	-
1/20/24 - 1/30/24	Weeks 3 - 4	Prospective Project Research	-
1/31/24	**Week 4**	**Capstone Preference Presentation**	-
2/2/24	Week 4	Projects Assigned	-
2/5/24 - 2/12/24	Weeks 5 - 6	Meet the Team, Sponsor Kick-off meeting	-
2/13/24 - 2/27/24	weeks 6 - 8	Sponsor Meetings, 3D printer training, research tasks in Jira, meeting with other capstone teams	1
2/27/24 - 3/12/24	weeks 8 - 10	Meeting with other capstone teams, understanding the big picture, splitting into groups, brainstorming action items, prepping for mid-project presentations	2
3/18/24 6:20 PM	**FINALS WEEK**	**Mid-Project Presentations**	**-**
4/124 - 4/14/24	Weeks 1-2	DB - researching Flask and UI expectations, exploring docker image creation UI - developing basic web pages, coordination with DB and Octo/ME team Octo/ME - researching pySerial and OctoRest, and exploring filament runout detection	**3**
4/15/24 - 4/28/24	Weeks 3-4	DB - db creation, docker image development UI - developing printer setting page, queue Octo/ME - development on pyserial script, filament runout detection continued	**4**

4/24/24	**Week 4**	**End of week 3:Sponsor will see some part of the software demoed. Software demo-ed from each group:**
		- DB:
		  - docker to start db
		  - Showing db backend
		- UI:
		  - Showed existing web pages
		- Octo/ME:
			4

Dates	Weeks of Term	Activity	Sprint #
		- Pyserial script for ME side to use - OctoRest script on outside	
4/29/24 - 5/12/24	Weeks 5-6	DB - db views, docker testing, developing startup script UI - continued development of web pages, integration with DB and Octo/ME teams Octo/ME - implementing API and component functions	5
5/15/24	**Week 7**	**Week 7: sponsor first tries using (an incomplete version of) the software themselves** - Staring software with startup script - Navigating UI	**6**
5/15/24 - 6/7/24	Weeks 7-8	All teams - integrating sponsor feedback into software, resolving issues with integration of components and individual components	6-7
6/7/24	**WEEK 10**	**Final Presentation Video DUE** - 15-25 minute screencast video	**-**
6/9/24 - 6/14/24	**FINALS WEEK**	**Delivery to Sponsor**	**-**

8. Backlog

The items in this backlog have not been implemented in this project. They represent logical next steps that we identified during our work. However, due to our limited time, we were unable to incorporate them.

- Combing PARQE and FUME database
    - Currently, each group has its own, and certain data is duplicated
- Test the ME FUME script we wrote with a Raspberry Pi Pico
- Once ME FUME capstone has mechanical components to move filaments, integrate and test with FUME software
- Make the UI responsive and accessible
- Process the prints sent to the larger printer in the EPL
    - There are some hiccups with/ Octoprint working with the MK4 printer. Differences with the build envelope, how hot film can get.
- Develop a way to prioritize print jobs
    - i.e. a regular queue and one for urgent jobs
- Video Monitoring
    - Having cameras - as a way to monitor print bed

9. Appendix: Glossary of Terms

- **Agile Methodology**
    - An iterative and flexible approach to software development that emphasizes collaboration, adaptability, and delivering incremental value to customers.
- **DB:**
    - A shorthand for database. Our database consists of 3 tables containing filaments, printers, and jobs. The database is written in Postgres.
- **DB/PARQE Group:**
    - One of the subteams in our project. The team responsible for creating the database and handling post requests from PARQE. See more in section 6.3.
- **ECE**
    - Electrical and Computer Engineering
- **EPL (Electronics Prototyping Lab):**
    - A lab at PSU for rapidly prototyping electronics projects. Houses and an array of different machinery, among those, a collection of 3D printers.
- **Filament**
    - A thermoplastic thread that is fed into the 3D printer and used for printing.
- **ME FUME**
    - A Mechanical Engineering FUME capstone team (fall-spring 2023-2024) is responsible for creating the mechanical component to be able to automate filament changing in the EPL print farm environment via serial communication
- **Octoprint:**
    - An open-source software platform that enables remote monitoring and control of 3D printers via a web interface.
- **Octo/ME Group:**
    - One of the subteams in our project. The group responsible for the OctoPrint/Octorest component and working with the mechanical engineering capstone group. See more in section 6.3

- **PARQE**
    - Shorthand which refers to the software for payment processing and order submission from the PrusaSlicer Automated Routing and Quoting Engine capstone team from fall-winter 2023.
- **Print Farm Environment:**
    - A setup where multiple 3D printers are organized and managed in one space. The current EPL print farm has 6 printers that we were working with.
- **Print Job**
    - 2 files, one that contains the gcode and a json file with printer information like filament type, nozzle size, etc.
- **UI:**
    - User Interface, the website used for interacting with the printers.
- **UI Group:**
    - One of the subteams in our project. Group responsible for creating the user interface. See more in section 6.3.
- **3D Printer**
    - A machine used to print 3D objects by laying many thin layers of thermoplastic thread.