# Introduction

[Intro]

# Architecture Overview

[Overview/Summary]

[High Level Diagram]

## Module Decomposition Table

[Description]

[Module Decomposition Chart]

## Module Decomposition Diagram

[Description]

[Module Decomposition Diagram]

## Module Data Flows Table

[Description]

[Module Data Flows Table]

## Producer Consumer Table

[Description]

[Producer Consumer Table]

## Module Functional Descriptions

### Use Interface Layer Modules

[Module Functional Descriptions]

### Preprocessing Layer Modules

[Module Functional Descriptions]

### Processing Layer Modules

[Module Functional Descriptions]

### Post Processing Layer Modules

[Module Functional Descriptions]

### Physical Layer Modules

[Module Functional Descriptions]

### Printer Feedback Layer Modules

[Module Functional Descriptions]

### Communication Layer Modules

[Module Functional Descriptions]

# User Interface Layer

The Interface Layer’s purpose is to collect user information and serve it to the processing layers. There are three subsystems providing data to lower levels. The GUI subsystem provides the user the ability to interact with the system and provides the ability to import files, create material profiles, set general configuration, and print specific configuration. The database subsystem provides a persistence framework for the other subsystems in this layer for storage and retrieval of configuration and material information. The controller subsystem decouples the user interface from the program logic. The controller provides the GUI with the information to present and takes in the user actions allowing the initiation of a print run, then packing the collected configuration, object files, and material data in to a print request object for the preprocessing layer.

## GUI Subsystem Modules

The GUI Subsystem is responsible for providing an interface to the user so that he or she can import model files and manage print input configuration information related to materials and printing hardware. The GUI Subsystem is also responsible for providing an interface from which the user can set configuration options for a specific print and initiate a print. During an ongoing print, GUI Subsystem is responsible for displaying the printer and current print status to the user.

### Import GUI Module

#### Prologue

Import GUI Module is a button or menu option that begins the event to import an object file

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Material GUI Module

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Configuration GUI Module

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Print GUI Module

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Status GUI Module

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

## Controller Subsystem Modules

The Controller Subsystem is responsible for collecting all the material and printer configuration information required for the print, bundling that information with the object file information received from the Import Subsystem, and sending that bundle to the Pre Processing Layer to begin the printing process. The Controller Subsystem is also responsible for telling the Printer State Controller to pause, resume, or stop a print job based on user input.

### Import Controller

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Material Controller

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Configuration Controller

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Print Controller

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Status Controller

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

## Database Subsystem Modules

The Database Interface subsystem is responsible for providing an abstract interface between the database of the system and any subsystems that need to store or retrieve information from the database. As such, the Database Interface Subsystem is responsible for exposing all the methods necessary for the other subsystems to communicate with the database in an abstract manner.

### Persistence Framework

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

### Command Structure

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS]

#### Process

[Pseudo-code]

# Preprocessing Layer

The Preprocessing Layer provides an abstract interface between the User Interface and the Processing Layer. This layer‘s purpose is to translate and repackage the print request object in to the format that the processing layer needs. The Preprocessing Layer starts by receiving all of the necessary configuration and object data from the User Interface’s Print Subsystem then converts it into a unified format that the Processing Layer understands. In the current iteration of this project, the Preprocessing Layer only has one subsystem, the Normalization Subsystem and will be converting STL files it receives from the object sent to it into an AMF file. The Preprocessing Layer then packages the configuration, object definition, and material data into the correct format for the Processing Layer.

## Normalization Subsystem Modules

The Normalization Subsystem is responsible for receiving the bundle containing the object data and printer configurations from the User Interface’s Print Subsystem and converting the object data into a format the Processing Layer can understand. After the object file(s) is converted, this subsystem sends a modified bundle of the object file information and the printer configuration to the Processing Layer so it can calculate a printing path.

### File Translation Module

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

# processing Layer

The Processing Layer takes the formatted package and translates it into G-codes. The slicing engine is the only subsystem in this layer. The slicing engine needs to be able to be replaced based on the growing needs of future development. The Processing Layer is designed to be expandable. The initial implementation will use Slic3r as its engine, but the normalized data from the Preprocessing layer will need to change if the slicing implementation changes. This allows the slicing engine to be easily replaced with a different implementation and provides the future possibility of allowing the user to select which slicing engine they wish to use in any given print. One the Processing Layer processes the normalized data from the Preprocessing Layer, the Processing Layer passes a packet of G-Codes and printer configuration information to the Post-Processing Layer.

## Slicing Subsystem Modules

The responsibility of the Slicing Engine is to read the object file and divide the geometry of the object into appropriate layers. Then for each material object of each layer, the subsystem will draw out a printing path for the head to follow. Once the object has been drawn out, the printing instructions will be converted to G-Codes and passed to the Post-Processing Layer for final changes to the instructions.

### Slicing Engine Wrapper

#### Prologue

The Slicing Engine Wrapper is responsible for wrapping an abstract API around the slicing engine of the system.

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

### Slicing Engine

#### Prologue

The Slicing Engine is responsible for slicing 3-D objects into printable layers based on provided configuration data.

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

# Post processing Layer

The Post Processing Layer receives the G-Codes from the Processing Layer and modifies the instructions to match special considerations the printer may need. G-Codes are somewhat standard, but some printers have instructions that are specific to that printer only. It is the job of the Post Processing Layer to correct the G-Code received from the Processing Layer to accommodate for those special instructions. Once finished, the Post Processing Layer will output G-Codes and the configuration to the physical layer. Similar to the Preprocessing and Processing Layers, the Post-Processing layer is designed to be easily expanded in future iterations. Initially, this layer will only support the printer supplied by the Mechanical Engineering Team.

## G-Code Preparation Subsystem Modules

The responsibility of the G-Code Preparation Subsystem is to modify the G-Code produced by the Processing Layer to be 100% compatible with the printer being used. This subsystem therefore must accommodate for any special instructions that are unique to that printer.

### Parser Module

#### Prologue

The Parser Module is responsible for parsing G-Code data it receives and modifying G-Codes to ensure that they are compliant with the G-Code standard required by the printing hardware.

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

### Unification Module

#### Prologue

The Unification Module is responsible for combining separated G-Codes into a single string of G-Codes that can be streamed to the printer. When parts of an object are sliced separately, the Unification Module must be used to combine the separated G-Codes together.

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

# Physical Layer

The physical layer receives inbound G-Codes and printer configurations from the post processing layer. The physical layer will then assemble and serialize the data received in preparation to be sent to the communications layer. The G-Codes may be modified contingent on data received from the printer feedback and user interface layers. If either the printer feedback or user interface layers indicate that the print must be stopped then the physical layer must insert halt commands into the G-Code command stream such that the machine terminates the print in a timely manner.

## [Subsystem] Modules

[Description]

### [Module Name]

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

# Printer Feedback Layer

The printer feedback layer is responsible for receiving streamed input from the printer hardware, interpreting the data, then formatting and dispatching the information to the physical and user interface layers. The information received from the printer will mainly consist of printer state such as extruder temperature, arm position, and other operating parameters. The printer feedback layer will transform the data received from the printer into useful and readable data that can be passed to the user interface and physical layers.

## [Subsystem] Modules

[Description]

### [Module Name]

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

# Communication Layer

The purpose of the communication layer is to serve as an interface between the client software (The 3-D Printer Fabrication System) and the printer firmware itself. The communication layer shall be able to implement most RS-232 compliant serial communications medium such that data can be transferred from the client software to the printer firmware reliably. The communications layer serves both the printer feedback and physical layers. The physical layer represents outbound communication (from client software to printer) while the printer feedback layer represents inbound communication (from printer to client software).

## [Subsystem] Modules

[Description]

### [Module Name]

[Description]

#### Prologue

[Name/Description/Function]

#### Interfaces

|  |  |  |
| --- | --- | --- |
| Interface | Information Required | Information Returned |
|  |  |  |
|  |  |  |
|  |  |  |

#### External Data Dependencies

|  |  |
| --- | --- |
| Data | Source |
|  |  |
|  |  |

#### Internal Data Descriptors

[NOT EXACTLY SURE WHAT THIS IS

#### Process

[Pseudo-code]

# Quality Assurance

[Description of test plans and procedures]

## Unit Testing

### [Layer Name]

#### [Subsystem Name]

##### [Module Name]

[Description of tests on the module]

## Component Testing

[NOT SURE WHAT EXACTLY THIS IS YET]

## Integration Testing

[Description of integration testing]

## System Verification Testing

[Description of system verification testing (possibly take from SRS)]

## Test Cases

[Description]

|  |  |
| --- | --- |
| Test Case | Expected Result |
|  |  |
|  |  |

# Requirements Traceability Matrix

[Description]

[Requirements traceability matrix (probably want to break this into layers)]

# Acceptance Plan

[Description/Overview]

## Package and Installation

[Probably use requirements from SRS]

## Acceptance Testing

[Description]

## Acceptance Criteria

[Reiterate acceptance criteria for requirements]

# Appendices