

System Specification

David Zwerdling
zwerlds@gmail.com

Version 1
September 20, 2023

Contents

1	Version History	2
2	Introduction	3
3	System Overview	4
3.1	Scope	4
3.2	Context	4
3.3	Audience	4
3.4	Statement of Purpose	4
4	Glossary	6
5	Stakeholders	7
6	Requirements	8
6.1	Functionality	8
6.2	Differentiation	8
6.3	Use Cases	8
7	UI Mockups	9
8	System Qualities	10
8.1	Performance	10
9	Architecture	11
9.1	Drivers	11
9.2	Styles and Patterns	12
9.3	Rationales	12
9.4	Alternative Architectures	12
9.5	Challenges and Limitations	13
9.5.1	Performance	13
10	Architectural Views	14
10.1	Components & Connectors	14
10.2	Modules	14

1 Version History

Version	Date	Author	Comments
1	September 20, 2023	Zwerdling	Initial version

2 Introduction

GlyphMosaic is a graphic design program. The application facilitates a specialized graphic design workflow, in which a user supplies a source image, text, and other parameters. The application then produces a reproduction of the source image using a mosaic of textual elements from a user-supplied source.

3 System Overview

3.1 Scope

GlyphMosaic is a specialized graphic design application. It comprises the following functionality:

- Enable the user to finely specify the parameters with which the output image is generated.
- Produce the output image.

3.2 Context

The application is installed on a host system. Users interact with the system's GUI to create, load, and modify graphical mosaics.

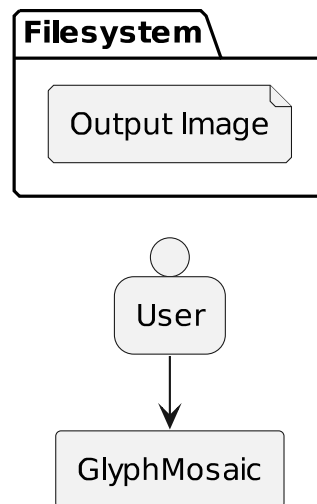


Figure 1: A hypothetical local install of GlyphMosaic and related components.

3.3 Audience

This document is intended for developers, open-source contributors, project managers, and consumers involved in the continuous development, maintenance, and use of a GlyphMosaic installation. It will provide the audience with a detailed understanding of the underlying architecture of the GlyphMosaic software application to gain better insight into how all components interact to deliver functional and robust productivity. By reading this document developers and open-source contributors can make informed decisions about how to design and implement various functionalities that will deliver an optimal user experience. This document aims to provide the audience knowledge to support decisions about identifying potential issues or areas of improvement in the system architecture. The purpose is to be a comprehensive overview of the GlyphMosaic software architecture to support all audience members involved in the development, maintenance, or usage of a GlyphMosaic application.

3.4 Statement of Purpose

GlyphMosaic is thoroughly described in this architectural document. Diagrams are also supplied to make it easier to comprehend the internal workings of the system. Various aspects of the GlyphMosaic architecture, including its core elements, modules, and code structure, are described. Functionality and quality attributes are described. The investigation has shown that the GlyphMosaic architecture offers several advantages, including a modular design, flexibility, and usability that have helped it become so popular. Nevertheless, the investigation has also pointed out other flaws that

may be strengthened to improve the application's overall performance, such as its heavy reliance on plugins and the possible security risks they provide. This document should act as a reference for any person interested in better understanding the GlyphMosaic architecture.

This report attempts to accomplish the following goals:

- Describe the complete architecture of the system.
- Describe available process alternative methods and their trade-offs.
- Provide a broad statement of use for the system itself.

4 Glossary

- **Bitmap**

A 2-dimensional matrix of pixels, representing an image. Each pixel may contain an arbitrary number of channels. In practice, bitmaps typically contain 3 channels (in the case of source images and output images) or a single channel (in the case of region masks and path dilation masks).

- **Document**

A file which contains all the necessary elements to recreate the output image. A user will create and modify documents for each output file they wish to create. Documents are stored and managed outside this system, by the host.

- **Glyph**

A single textual element. Generally corresponds to a single letter, but that definition is incomplete due to complexities introduced by Unicode, which the system utilizes in the process of rendering.

- **Output Image**

A system-rendered bitmap. Users may export output images

- **Path Dilation Masks**

The system generates a series of dilations around the given region line kernel. This involves taking the prior iteration, or kernel in the case of the first iteration, and accreting (dilating) pixels within a given radius of existing pixels, generating the next iteration. In this method, a series of concentric lines can be generated, enabling a text path to be calculated.

- **Render**

- **Region**

- **Region Line Kernel**

- **Region Mask**

- **Source Image**

A bitmap which serves as an oracle for individual pixels. Source image pixels are sampled to determine the size and color of each glyph before stamping.

- **Source Text**

A corpus of text is specified as a source for glyphs, which are then applied to the calculated glyph data elements. These glyphs are scaled, translated, and rotated accordingly, and then stamped on a bitmap, forming the final image render.

5 Stakeholders

Individuals who have an investment in the system include:

- **Output Image Consumers**
Individuals who view or request graphic design produced by the system.
- **Developers**
Individuals responsible for producing the system as described in this document.
- **Graphic Designers (Users)**
Individuals who produce graphic designs using the system.

6 Requirements

Requirements for GlyphMosaic aim to create a responsive and productive environment for its users. The GlyphMosaic System provides a specialized method to build unique graphic mosaic designs.

6.1 Functionality

GlyphMosaic’s main built-in functionality is focused entirely on the workflow of creating mosaics. In the most common workflow, a user will load source image and source text into the program, adjust various settings and bitmaps to their needs, render a high-resolution version of the output mosaic, and then save that file for use outside the application.

6.2 Differentiation

Existing systems perform a similar, but limited subset, of functionality of the system. Textaizer[?], for example, has the ability to specify various line patterns, such as spirals. Many other examples exist that focus solely on LTR horizontal mosaics.

GlyphMosaic does not have these limitations. The creator is able to specify the glyph line pattern by specifying a region line kernel for each region. The system then determines a text path which encompasses the entire region, and develops a

6.3 Use Cases

- **System Installation**

- ◆ Users should be able to download, install, and run the system on their own devices.

- **Content Creation**

- ◆ Users should be able to create and modify GM documents.

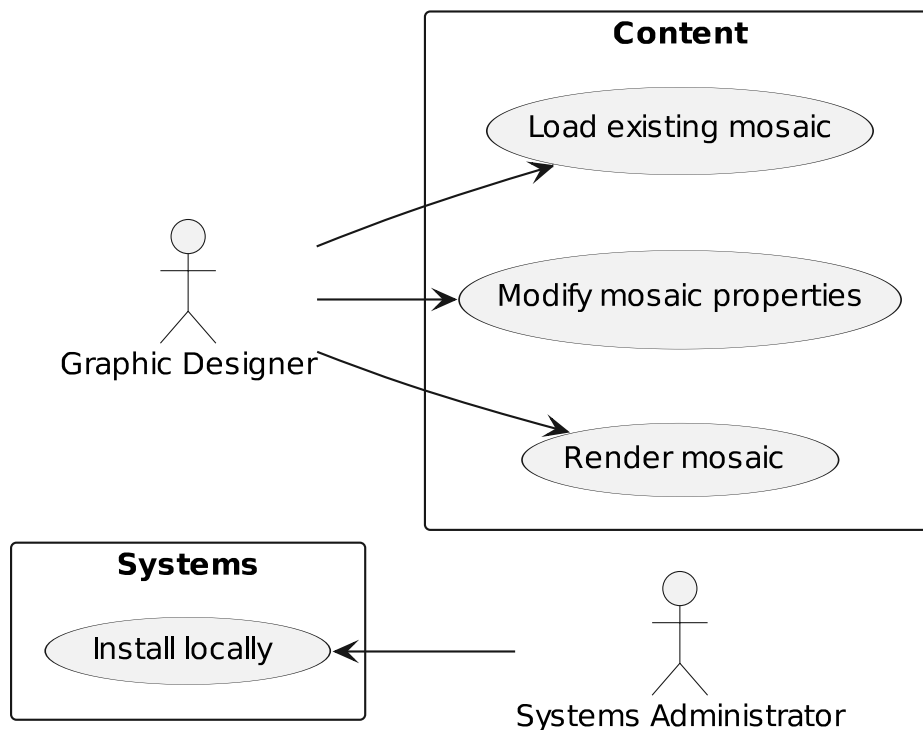


Figure 2: GlyphMosaic models a specialized graphic design workflow.

7 UI Mockups

8 System Qualities

8.1 Performance

Users should be able to rapidly receive feedback representing the current state of the project. This includes a preview of the document they wish to create.

- **Tactics**

- ◆ **Cacheing**

Throughout the image creation pipeline, certain modules may not necessarily produce different output, given a user's input. Recomputing these results is wasteful, and avoided by implementing a cacheing layer available to other modules in the system. This method involves trading future computational load for memory space.

- ◆ **Profiling**

Architects of the system utilize system profiling on typified workloads to determine bottlenecks in performance. These bottlenecks are then examined and the system is revised to eliminate it.

- **Scenarios**

- ◆ **Region Drawing**

When a user draws a region mask on the preview pane, the system should display a preview of the mask within one tenth of a second.

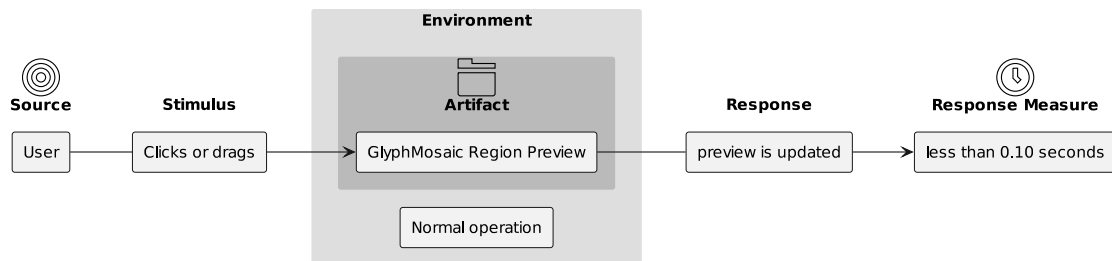


Figure 3: GlyphMosaic gives near-instantaneous feedback for region specification.

- ◆ **Mosaic Preview “Cold” Response**

When a user opens the preview pane, it should display a preview of the mosaic within one second.

- ◆ **Mosaic Preview “Warm” Response**

When a user changes a parameter in the document while the preview pane is already open and viewing a result of that change, the user should see the result of that change within one quarter of one second.

- ◆ **Mosaic Full Render Response**

When the user triggers a full render of the document, the system should write the result to the host filesystem within 20 seconds.

9 Architecture

9.1 Drivers

- **Design Purposes**

- ♦ **Modeling a Specialized Workflow**

The system is designed to facilitate a specific graphic design workflow. This workflow follows an interactive approach, like most graphic design workflows, in which a document would be created and modified in successive phases until the designer is content with the result. At this point, the designer indicates to the system that the image should be rendered at high-resolution for output and consumption by other systems.

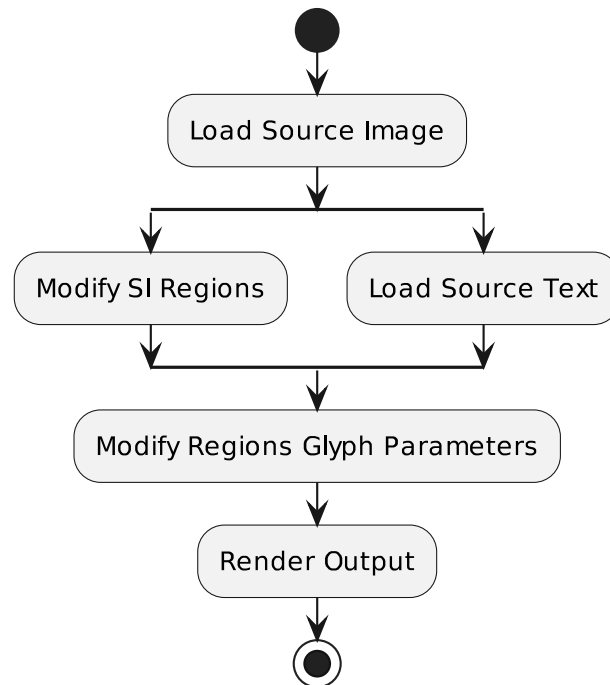


Figure 4: The system workflow models specialized graphic design phases.

- **Constraints**

- ♦ **Resource Use Limitations**

Devices serving GlyphMosaic may be constrained by their ability to perform the large number of computations necessary for the production of the end results. In particular, resource use may be constrained in the following aspects of the system's functions:

- ★ **Path Calculation**

The system must calculate the path on which glyphs are then vectorized. This involves using the region kernel specified by the user and applying a dilation function onto it. Over successive operations, this eventually will cover the entire region. The accretion operation can be implemented using various methods. However, the highest-quality version involves creating an image kernel sized corresponding to the user-defined line distance and applying the transform over a large area.

The system implements this calculation as detailed in Section ??.

- ★ **Glyph Data**

The position and direction of each individual glyph must then be calculated based on the calculated path. This involves walking the path and generating locations at user-specified

intervals. Additionally, each glyph must calculate its scale and color by sampling the source image at the calculated location. This operation is computationally intensive due to the large number of glyphs, and must be optimized to this document's performance standards.

The system implements this calculation as detailed in Section ??.

★ **Glyph Rendering**

Each glyph is stamped on a bitmap as part of the rendering process. This is constrained by the host system's ability to process potentially thousands of glyphs. Glyphs can be stamped on arbitrary bitmaps which can then be merged to form the final rendered bitmap. This pattern allows the system to trade memory (additional bitmaps) and processing power (additional glyph-data-consuming threads) for wall-clock time.

The system implements this calculation as detailed in Section ??.

9.2 Styles and Patterns

Monolithic GlyphMosaic communicates with the host operating system using standard methods. Within the host operating system, the process exists within a singular executable. This simplifies potential complexity of interacting with the host operating system by reducing the abstract footprint of the application. In summary: the system need only manage a single executable element, eliminating the need to implement IPC-, or networking-related requirements. This decision does impact potential performance, as discussed in Section 9.4.

Layered Within the monolithic system, subsystems compose into macroscopic functionality as layers. This method of design is an attempt to mitigate complexity of the system by reducing the possible interaction between sub-systems. This approach is utilized in the following locations:

- **Test**

Description.

Pipe/Filter In many cases, systems are composed in series to build larger functionality. This approach is utilized in the following locations:

- **Test**

Description.

Model-View-Controller This approach is utilized in the following locations:

- **Test**

Description.

Event Bus

This approach is utilized in the following locations:

- **Test**

Description.

9.3 Rationales

A few of the most enabled qualities include:

- **Performance**

- **Maintainability**

9.4 Alternative Architectures

- **Distributed Computing**

Breaking up components of GlyphMosaic into services could enable more computational resources to be added to the system.

9.5 Challenges and Limitations

9.5.1 Performance

10 Architectural Views

10.1 Components & Connectors

The system interacts with the following aspects of the host system:

- **Filesystem**

The system retrieves source images and text from the local filesystem.

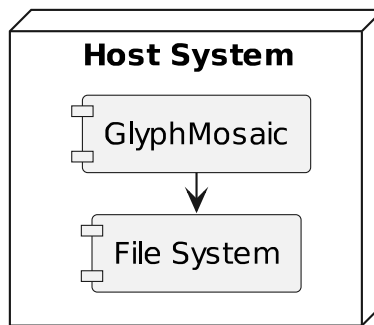


Figure 5: The GlyphMosaic system is fairly limited in responsibility from an application/systems administrator perspective.

10.2 Modules

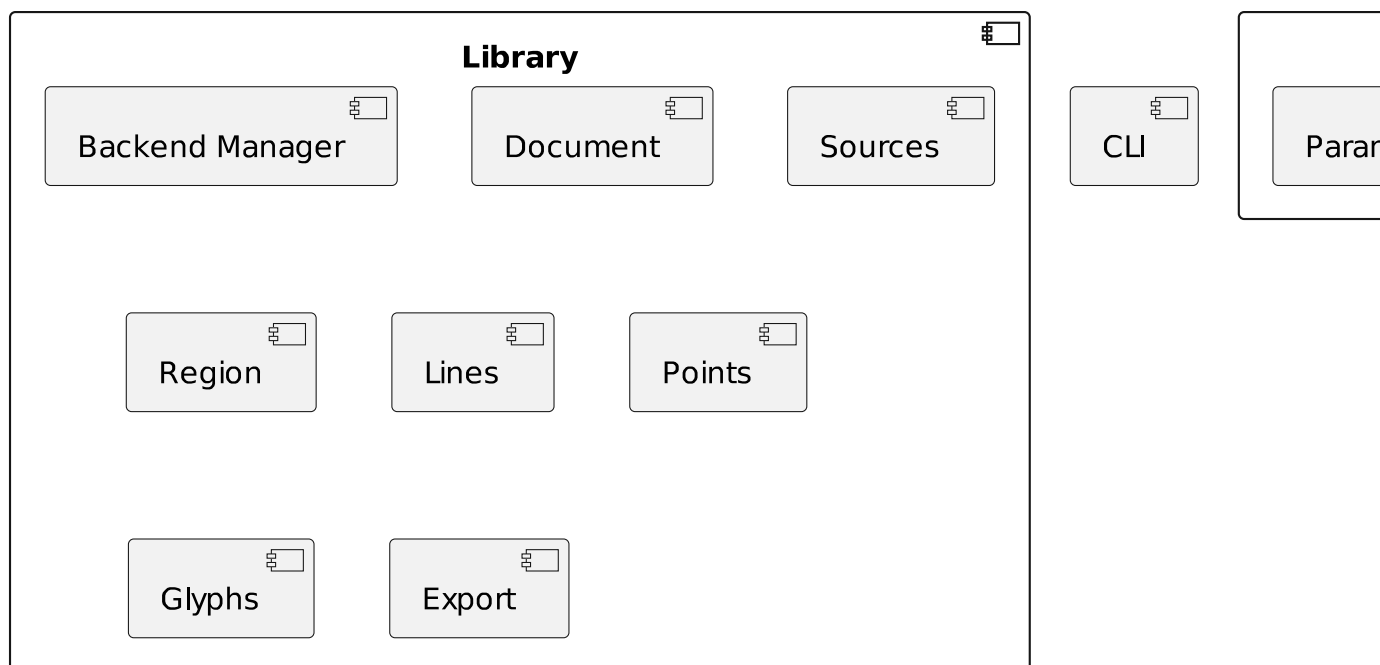


Figure 6: The top-level organization of the system.