# Module Interface Specification for ImgBeamer

Joachim de Fourestier

March 19, 2023

# 1 Revision History

| Date | Version | Notes |
|---|---|---|
| 2023/03/18 | 0.1.0 | Creation |
| | 0.1.1 | Update module hierarchy |
| 2023/03/19 | 0.1.2 | Add in module specifications |

## 2 Symbols, Abbreviations and Acronyms

See SRS [2] and the MG [1] Documentation.

# Contents

# 3   Introduction

The following document details the Module Interface Specifications for ImgBeamer (SEM image formation demo tool).

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/joedf/CAS741_w23.

# 4   Notation

[You should describe your notation. You can use what is below as a starting point. —SS]

The structure of the MIS for modules comes from Hoffman and Strooper [4], with the addition that template modules have been adapted from [3]. The mathematical notation comes from Chapter 3 of Hoffman and Strooper [4]. For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by ImgBeamer.

| Data Type | Notation | Description |
|---|---|---|
| character | char | a single symbol or digit |
| integer | $\mathbb{Z}$ | a number without a fractional component in $(-\infty, \infty)$ |
| natural number | $\mathbb{N}$ | a number without a fractional component in $[1, \infty)$ |
| real | $\mathbb{R}$ | any number in $(-\infty, \infty)$ |

The specification of ImgBeamer uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, ImgBeamer uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

# 5 Module Decomposition

The following table is taken directly from the Module Guide [1] document for this project.

| Level 1 | Level 2 | Level 3 |
|---|---|---|
| Hardware-Hiding Module | | |
| Behaviour-Hiding Module | Application Control | |
| | Input | Ground Truth Image Input |
| | | Imaging Parameters Input |
| | | Spot Profile Input |
| | Output | Information and Metrics Display |
| | | Image Export |
| | Visualization Display | Ground Truth |
| | | Subregion |
| | | Spot Profile |
| | | Spot Content |
| | | Spot Signal |
| | | Spot Layout |
| | | Sampled Subregion |
| | | Resulting Subregion |
| | | Resulting Image |
| Software Decision Module | Display Control | |
| | Graphical User Interface | |
| | Image Manipulation | Drawing Stage / Canvas Module |
| | | Rendering |
| | | Metrics Calculation |

Table 1: Module Hierarchy

# 6  MIS of Application Control

## 6.1  Module

main

## 6.2  Uses

Metrics Calculation Module (M**??**), GUI Module (M**??**)

## 6.3  Syntax

### 6.3.1  Exported Constants

### 6.3.2  Exported Access Programs

| Name | In | Out | Exceptions |
|------|-----|-----|------------|
| main | - | - | - |

## 6.4  Semantics

### 6.4.1  State Variables

None

### 6.4.2  Environment Variables

None

### 6.4.3  Assumptions

- The application is run in an HTML5 compliant web browser.

- The GUI is running and displayed without issue.

### 6.4.4  Access Routine Semantics

main():

- transition: Modifies the state and environment variables of the GUI Module (M**??**).

### 6.4.5  Local Functions

UpdateBaseImage(): Updates the GUI and propagates a change in the input ground truth throughout the application. updateImageMetricsInfo():

# 7 MIS of [Module Name —SS]

[Use labels for cross-referencing —SS]
   [You can reference SRS labels, such as R1. —SS]
   [It is also possible to use LaTeX for hypperlinks to external documents. —SS]

## 7.1 Module

[Short name for the module —SS]

## 7.2 Uses

## 7.3 Syntax

### 7.3.1 Exported Constants

### 7.3.2 Exported Access Programs

| Name | In | Out | Exceptions |
|------|-----|-----|-----------|
| [accessProg —SS] | - | - | - |

## 7.4 Semantics

### 7.4.1 State Variables

[Not all modules will have state variables. State variables give the module a memory. —SS]

### 7.4.2 Environment Variables

[This section is not necessary for all modules. Its purpose is to capture when the module has external interaction with the environment, such as for a device driver, screen interface, keyboard, file, etc. —SS]

### 7.4.3 Assumptions

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

### 7.4.4 Access Routine Semantics

[accessProg —SS]():

- transition: [if appropriate —SS]

- output: [if appropriate —SS]

4

- exception: [if appropriate —SS]

[A module without environment variables or state variables is unlikely to have a state transition. In this case a state transition can only occur if the module is changing the state of another module. —SS]

[Modules rarely have both a transition and an output. In most cases you will have one or the other. —SS]

### 7.4.5 Local Functions

[As appropriate —SS] [These functions are for the purpose of specification. They are not necessarily something that is going to be implemented explicitly. Even if they are implemented, they are not exported; they only have local scope. —SS]

# References

[1] J. de Fourestier. Module guide for imgbeamer, 2023. URL https://github.com/joedf/CAS741_w23/blob/main/docs/Design/SoftArchitecture/MG.pdf.

[2] J. de Fourestier. Software requirements specification for imgbeamer: Scanning electron microscope image formation, 2023. URL https://github.com/joedf/CAS741_w23/blob/main/docs/SRS/SRS.pdf.

[3] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering.* Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

[4] Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach.* International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

# 8 Appendix

[Extra information if required —SS]