State-Machine Configurator

Analysis and Design Document

Student: Sas Cosmin Andrei

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18.03.2020 | 1.0 | <details> | Sas Cosmin Andrei |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# The project represents a platform (a development tool) that will generate C code based on a State Machine. The state machine will be created by the user using the built-in drag&drop interface.

# Elaboration – Iteration 1.1

# Domain Model

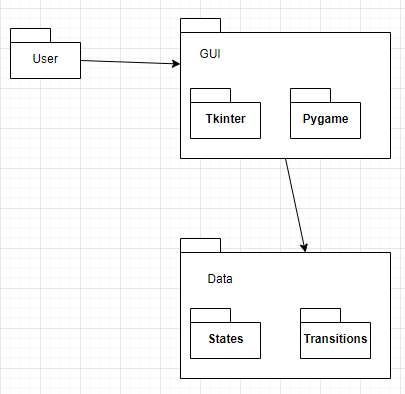
* The user will interact with the tkinter part and the pygame part
* As the data is modified by the tkinter and pygame parts, the new structure will be calculated in the background

# Architectural Design

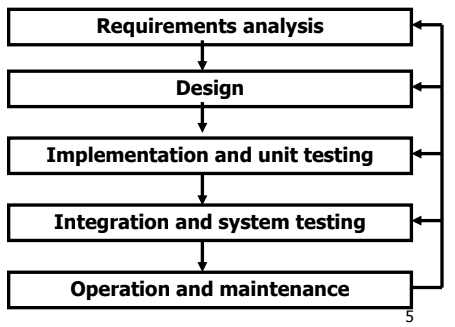
## Conceptual Architecture

* I will use a component-based architectural style
* The pattern will be Model-View-Controller

## Package Design



## Component and Deployment Diagrams



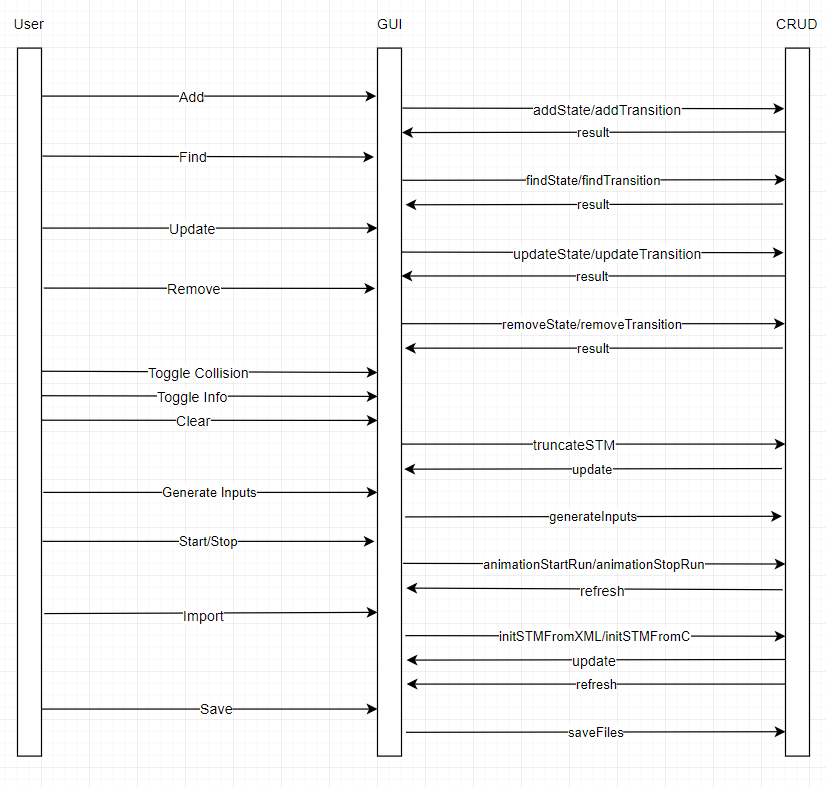
# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

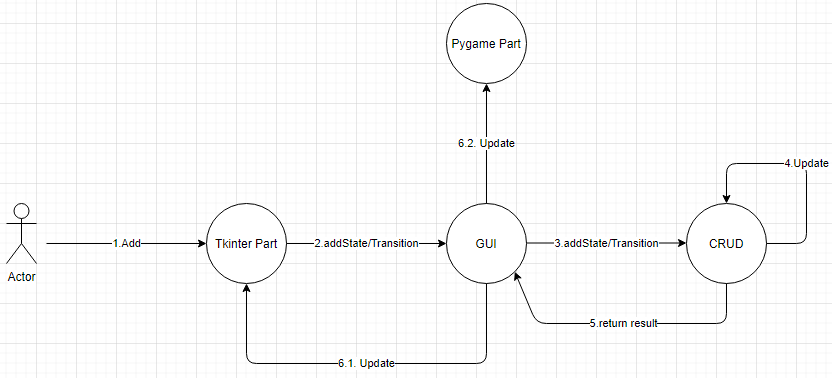
*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

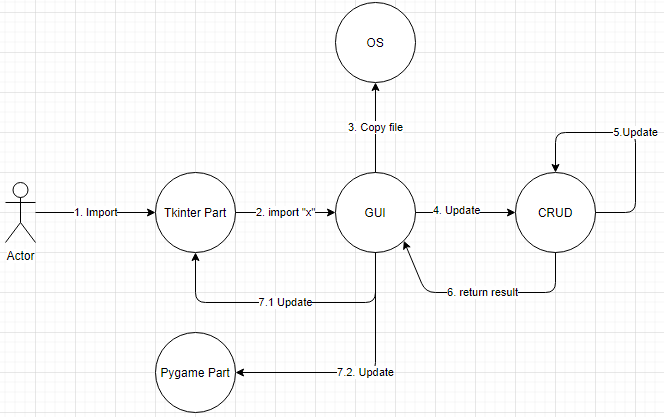
*Sequence Diagram (for all scenarios)*



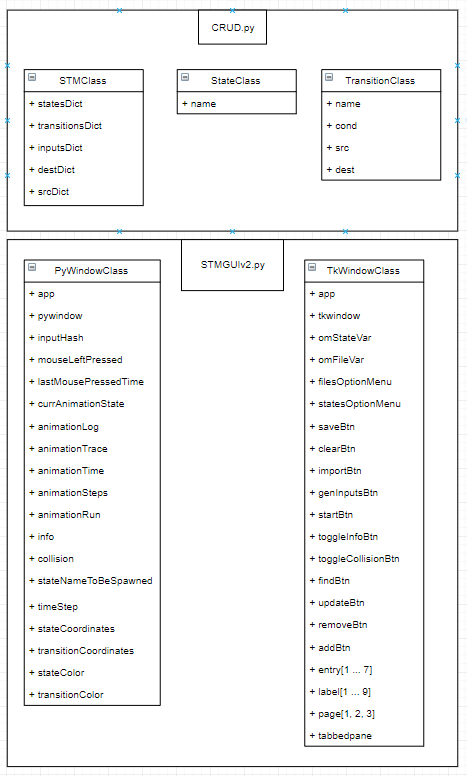
*Communication Diagram*

* *Adding an item to the STM*

**

* *Importing a STM  
  *

## Class Design



# Data Model

*[Create the data model for the system.]*

* *This project will not use a database for storing data. Instead it will use a set of files*
* *The C file and the XML file contain the same thing (the functionality of the state machine along with all the states and transitions) but in different formats.*
* *The inputs file contains the name of each input, the times of its changes and its values*
* *The coord file contains the coordinates of each state on the pygame interface*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

* For the internal state-machine tree I can test every function separately

# Future improvements

* I think the greatest improvement would be making it work on the GPU.

# Bibliography