Computer Science
&ENGINEERING

State-Based Plot Coloring

Implementation and Integration

Haoxuan Lin, Myeongwan Beom, Zachary Young

Team #36

Instructors: Dr. Sergiu Dascalu,
Prof. Devrin Lee

External Advisor: Mr. Eric Gilchrist

May 07th, 2018

Table of Contents

1. Project Summary

1.1 A list of use cases fully developed

Use Case ID	Use Case Name	Use Case Description
UC01	ViewBodePlot	The user selects the Bode Plot to view which results in the plot being displayed on screen.
UC02	ViewCascadePlot	The user selects the Cascade Plot to view which results in the plot being displayed on screen.
UC03	ViewTrendPlot	The user selects the Trend Plot to view which results in the plot being displayed on screen.
UC04	ChooseStateValues	The user enters their own values for each state that is either a default state or a state defined by the user. These values are then used when plotting the data.
UC05	ChooseStateColors	The user chooses the colors to represent each default or user defined state which are then used when plotting the data.
UC06	CreateCustomStates	The user can select to use the default states for the machine or define their own custom states that will be saved for future use.
UC07	ChooseMachine	The user selects a machine in one of the three plots to select a specific machine to view rather than viewing all machines.
UC08	AdjustWindowSize	The user adjusts the size of the window to make the window larger or smaller to fit their needs.

UC09	AdjustElevationAnd Azimuth	The user adjusts the elevation and azimuth of the cascade plot using two sliders.
UC11	ExitProgram	The user can exit the program and all plots that are not saved will be lost giving the user an option to save before exiting.

1.2 A list of functional requirements fully developed

Requirement ID	Requirement Priority	Requirement Description
FR01	1	The system shall read in data from sample machine data.
FR02	1	The system shall plot the data using a Bode Plot
FR03	1	The system shall plot the data using a Cascade Plot
FR04	1	The system shall plot the data using a Trend Plot
FR05	1	The system shall color the different plots based on the state of the machine.
FR06	1	The system shall allow the value ranges for each the states to be defined by the default values for the machine.
FR07	1	The system shall allow the value ranges for each the states to be defined by the user.
FR08	1	The system shall handle edge cases where the machine is in two or more states at the same time.
FR09	1	The system shall allow the user to select a machine to view in the plot when multiple machines are present.

FR10	1	The system shall allow the user to switch between the different plots.
FR11	2	The system should allow the user to display multiple plots at once.
FR12	2	The system should allow the user to define their own states that they want to monitor.
FR13	2	The system should allow the user to choose their own colors for each state.

1.3 A list of non-functional requirements fully developed

Requirement ID	Requirement Description
NFR02	The system shall be implemented using C#.
NFR03	The system shall run on Windows7/8/10 operating systems.
NFR04	The user interface shall be easy to understand and use.
NFR05	The user shall be able to select their colors using a traditional color wheel.
NFR06	The system shall be usable on a web based platform.
NFR07	The plots shall be plotted in such a way that is easy to understand and distinguish between the states.

1.4 Functionality not implemented

Requirement ID	Requirement Priority	Requirement Description	Reasons why it is not completed
FR14	2	The system should allow the user to save a plot of data including relevant user entered states, values, and colors.	The requirement is not so helpful since the user can load the data to the program and display it within a few seconds when the sample size is not too large, so we decide not to include it in our program.
FR15	1	The system might read in real machine data.	We did not receive real machine data. The data that is used for our program is generated by the data generator that is given by our sponsor.

1.5 Contributions of team members

1.5.1 Contributions of team members on project assignment 5

Haoxuan Lin spent 2 hours contributing to the following:

- Cover Page
- Use Cases
- A list of use cases fully developed
- A list of functional requirements fully developed Cascade Plot
- A list of non-functional requirements fully developed
- Functionality not implemented

Myeongwan Beom spent 2 hours contributing to the following:

- Table of Contents
- Contribution Matrix
- Contributions of team members on the implementation, integration, and testing part
- Software Overveiw

Zachary Young spent 2 hours contributing to the following:

- Software Overveiw

1.5.2 Contribution Matrix

Team Member	Number of Templates Created	Lines of Code	Hours Worked
Haoxuan Lin	8	1800	150 hours
Myeongwan Beom	0	548	100 hours
Zachary Young	0	605	100 hours

1.5.3 Contributions of team members on the implementation, integration, and testing part

Functionalities	Who implemented the functionalities	Who integrated the functionalities	Who tested the functionalities
User Interface for Cascade Plot	Haoxuan Lin	Haoxuan Lin	Myeongwan Beom
User Interface for Bode Plot	Myeongwan Beom	Myeongwan Beom	Zachary Young
User Interface for Trend Plot	Zachary Young	Zachary Young	Haoxuan Lin
Text File Reader	Haoxuan Lin	Haoxuan Lin	Myeongwan Beom
Displaying Data in Cascade Plot	Haoxuan Lin	Haoxuan Lin	Myeongwan Beom
Displaying Data in Bode Plot	Myeongwan Beom	Myeongwan Beom	Zachary Young
Displaying Data in Trend Plot	Zachary Young	Zachary Young	Haoxuan Lin
Displaying Overlapping States in Cascade Plot	Haoxuan Lin	Haoxuan Lin	Myeongwan Beom
Displaying Overlapping States in Bode Plot	Myeongwan Beom	Myeongwan Beom	Zachary Young
Displaying Overlapping States in Trend Plot	Zachary Young	Zachary Young	Haoxuan Lin
Readable Overlapping Machines in Cascade Plot	Haoxuan Lin	Haoxuan Lin	Myeongwan Beom
Readable Overlapping Machines in Trend Plot	Zachary Young	Zachary Young	Myeongwan Beom
Communication between WPF windows	Haoxuan Lin	Haoxuan Lin	Zachary Young

2. Software Overview

The software is split into 14 main classes and xaml files associated with some of the classes.

TextFileReader
This class reads in all the files the user selects and stores the information into an array to be used later.

Machine
This class initializes the xaml for the window and handles interactions with the window such as selecting data, bode data, and state file.

User
This class initializes the xaml for the window and handles the interactions for selecting the files and the user customizing their states.

AllInfo
This class shares information between the Trend Plot, Cascade Plot, and Bode Plot and handles coloring of the plots.

BodePlot_Chart
This class handles formatting of the chart for the Bode Plot.

ChartStyle2D
This class handles formatting of the chart for the Trend Plot.

DataSeries2D
This class stores a list of polylines with their associated configurations.

DataCollection2D
This class stores a list of DataSeries2D. For each data series, the AddLines function draws 2D polylines on a 2D screen.

ChartStyle3D
This class handles formatting of the chart for the Cascade Plot.

DataSeries3D
This class is used to store a list of 3D points.

DataCollection3D
This class stores a list of DataSeries3D. For each data series, the AddLines function draws 3D polylines on a 2D screen.

Utility3D
This class is used to define the Azimuth and Elevation view matrix to a 3D parallel projection on a 2D screen.

TrendPlot
This class is used to interact with the xaml to display the Trend Plot and initialize all components of the trend plot. This class also utilizes ChartStyle2D, DataSeries2D, and DataCollection2D to display the data.

BodePlot
This class is used to interact with the xaml to display the Bode Plot and initialize all components of the Bode Plot. The class also iterates through the data to display the data into the Plot.

CascadePlot
This class is used to interact with the xaml to display the Cascade Plot and initialize all components of the Cascade Plot. This class also utilizes ChartStyle3D, DataSeries3D, and DataCollection3D to display the data.