Back To The Drawing Board!

Contents

1 Introduction **3**

2 What do we have so far? [24 July 2014, draw-icons branch] **3**
2.1 Flow 3
2.1.1 Flowchart 3
2.1.2 Components 3
2.1.3 Editors 4
2.2 GUI 4

3 Where do we want to go? **5**

4 So what now? **6**

1 Introduction

Alright Think-a-torians, we're going back to the drawing board!
And just as Dr Gramme Edwards said: "It's not the plan that is important, it's the planning"
So before the semester gets really hectic let's secure the planning of our project.
I'm still as eager to work on the project and I hope that you are too!
So, let's take a look at where were are so far...

2 What do we have so far? [24 July 2014, draw-icons branch]

Currently we have the following working packages and files:

2.1 Flow

- Main
  This is where the method main() is. It is responsible for instatiating the MainDisplay object (GUI) and showing the splash.

- Sheet
  This class currently exists and it extends JComponent, I am not yet sure of its purpose.

2.1.1 Flowchart

//Flowchart is within the Flow package

- Component
  This class is an abstract class, it extends JPanel, it is the class from which the components are extended.

- ComponentEditor
  This class in an interface class, it provides the methods that should be implemented by the concrete Editors.

- Flowchart
  This class extends Sheet, I am not yet sure of its purpose.

2.1.2 Components

//Components is within the Flowchart package
The classes below extend the Component class.

- CommentComponent
- DecisionComponent
- EndComponent
- FlowLineComponent
- InputComponent
- JumpComponent
- ModuleComponent
- Output Component

- ProcessComponent
- ReturnComponent
- StartComponent

2.1.3 Editors

//Editors is within the Flowchart package
The classes below implement the Component class.

- CommentComponentEditor
- DecisionComponentEditor
- Editors
- InputComponentEditor
- ModuleComponentEditor
- OutputComponentEditor
- ProcessComponentEditor
- ReturnComponentEditor
- StandardEditor

2.2 GUI

- MainDisplay
  This is where the displays are handled.
- Splash
- FlowCanvas
  This class extends JPanel, I am not sure of its purpose

3 Where do we want to go?

In the meeting [24 July] we discussed using the Model-View-Control Architecture.

Here is a recap on how Model-View-Control works:

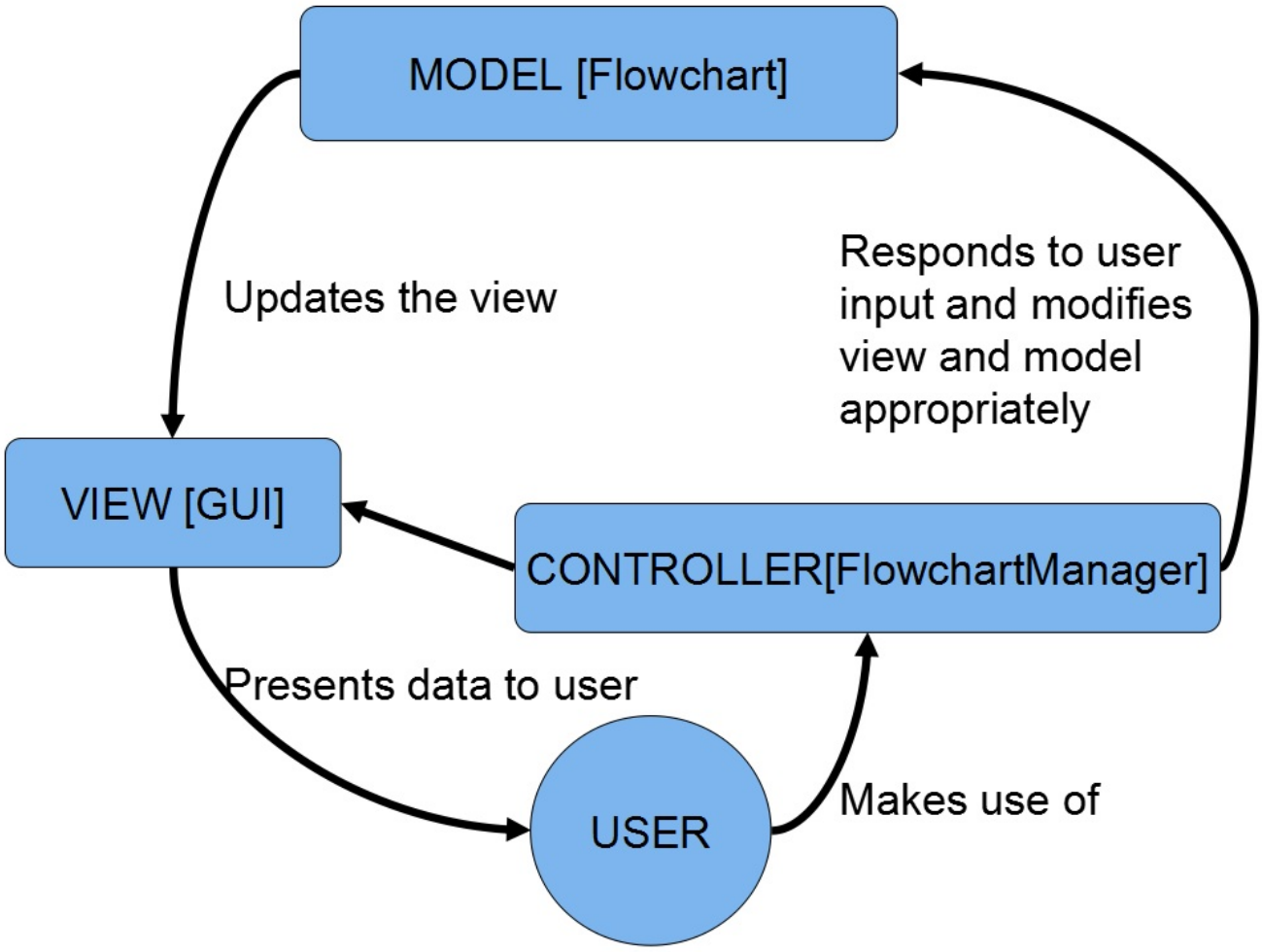
The Model-View-Controller allows for dividing the application into three components (this makes the problem domain independent from the user interface).

The Model-View-Controller also allows for separation of concerns and it facilitates code reusability.

A model stores the data that matters to the user.

A view represents the output to the user.

A controller responds to the user's inputs and updates the model (e.g., adding a component to the flowchart) and the view (e.g., maximizing the flowchart window) when needs be.



4 So what now?

I hope this has made your mind juices flow and has given you perspective of how the project should be structured.
So if you have any ideas of how the UML diagram should turn out - please share.
Also note the actual relationship between the classes - composition and the like when you do MVC.
Also include the script stuff in the UML.
Also if I have made mistakes please point them out.
I will also try to create a UML in MVC for the project maybe by Sunday.
NB! Also please put comments on the files like: Sheet and Flowchart and the like in the meantime so that it is clear what each purpose of a class is.
I don't really understand where these classes fit in so clarity will be greatly appreciated.
Enjoy your weekends, see you at 08:30 on Monday! [or 07:30 on Saturday :)]

We should have restructured our code by the end of our meeting on Monday.
And definitely be working with the functionality thereafter.

Also, they'd like a burndown chart, if someone can download the app to do it that would be nice (I will also try to download it).