UML Editor - Iteration 2 by FiVe

1 Problem statement

We are attempting to develop software that allows the end user to generate UML diagrams that appropriately and accurately model software and programs.

2 System Personnel

2.1 Description of Users

End User is Dr. Hutchens. He is a professor of computer science.

2.2 Description of System Developers

System developers are Eric Dougherty, Kelsey Fulton, Ryan Peterson, Matthew McAnulty, and Timothy Kettering.

3 Operational Setting

3.1 Target Platforms

Target platforms are Windows, Mac OS X, and Linux.

3.2 Required Software Environment

The required software environment is the Java Runtime Environment 8.

4 Functional Requirements

4.1 Functional Description

4.1.1 Overview

The UML editor allows the user to create class boxes, remove class boxes, move class boxes, and create relations. Relations can be labled and also deleted. The editor has a horizontal file menu, vertical context menu, and a workspace.

4.1.2 Feature List

The UML editor, once opened, will allow the user to create a UML diagram. The window will display a horizontal file menu, a vertical context menu, and a large work space. The class button is the only button initially available to the user. Clicking the button will create a class box. Once the box is created, the user can fill out the areas of the class box: class name, attributes, operations, and there is a fourth box for any other miscellaneous information. A user can then click, and drag the class box to the desired area on a display grid. The display grid becomes visible only when the user clicks to move the box. Once the user is done moving the box, the box will snap to the nearest grid location, and the grid will disappear. All objects on the UML work space are selectable. Once a UML object is selected the toolbar buttons will change to reflect the actions available for the selected option. By selecting the class box the user can select two more options from the toolbar. They can delete the class box, and they can create relations. To delete a class box, the user must select the box and choose the delete option from the context menu. Once the user has two class boxes, they can create a relation between the class boxes using a button from the context menu.

4.2 User Interface

4.2.1 Overview

The user interface includes a window when the application is opened. The window includes a file menu bar, a context menu bar, and a workspace. Options will appear on the context menu bar once an object is selected. A class box once selected will expand to give the end user access to four editable fields; : class name, attributes, operations, and there is a fourth box for any other miscellaneous information. A relation once selected will display a field for labeling the relation by clicking on the field.

4.2.2 Menus

The menus included are a file menu bar and a context menu bar. The file menu bar contains file, edit, preferences, and help selections. The context menu bar contains buttons for add box, add relation, and a delete box selections.

4.2.3 Windows

The UML editor has one window that appears when you open the application. The window contains a file menu bar, a context menu bar, and a workspace.

4.3 Use Cases

Preconditions: UML Editor is open and visible on the desktop.

Scenario / Software Reaction:

4.3.1. Operator clicks class button. / Class Box Appears.
4.3.2. Operator selects class box and clicks ‘add class name’. / Class Box field becomes editable.
4.3.3. Operator selects class box and clicks ‘add attribute’. / Class Box field becomes editable.
4.3.4. Operator selects class box and clicks ‘add operation’. / Class Box field becomes editable.
4.3.5. Operator selects class box and clicks ‘add miscellaneous’. / Class Box field becomes editable.
4.3.6. Operator clicks and holds on a class box. / Grid appears and Class Box becomes movable.
4.3.7. Operator clicks/selects a class box. / Relation and Delete buttons appear in toolbar.
4.3.8. Operator clicks delete button. / Selected object is deleted.
4.3.9. Operator selects relation and clicks ‘add text here’. / Editable text box appears.
4.3.10. Operator selects object and click delete. / Selected object disappears from work space.

5 Non-Functional Requirements

5.1 Reliability

While important to the individual user, the data is not of great importance overall. Therefore, we chose not to emphasize reliability; however, JavaFX has proven to be very reliable as the program has not crashed during regular use.

5.2 Performance

We do not want any visible performance issues, so we took care to keep code clean and refactored it when necessary – sometimes even rewriting certain sections entirely. The program is responsive on all tested devices. The program compiles cleanly and without errors.

5.3 Usability

Emphasis was placed on usability so that even in the early stage the user could easily navigate the program, and understand what each button did. In an effort to add to the usability the program also hides features contextually so that they are only visible when they can be used.

5.4 Portability

We placed a high emphasis on portability so that the program was transferrable across Windows, Mac OS X, and Linux through Java.

6 Future Enhancements

Future enhancements include the addition of different types of relations as well as adding a dashed line for the dependency relation. Each different type of relation will have an arrowhead and line that is changeable via a toolbar button when the relation is clicked/selected in the workspace.

The user will also have the ability to select and move multiple relations by clicking and dragging them to a new location in the work area.

Another feature that the team would like to add is the ability for the work area to become scrollable if the diagram becomes too large to view in the visible work area.

The team is also working to add print (as a pdf), save and load functionality to the menu bar.