Case Study

In this case study, your task is to study the requirement of your final project and model the requirements into a UML diagram. By studying this requirement, you need to identify the classes needed to complete the implementation. Besides, you need to identify the association among classes. These association classes can be of aggregation and composition relation.

Your goal is to draw a UML class diagram for these classes and association relations you identify.

The Specs of Macro UML editor

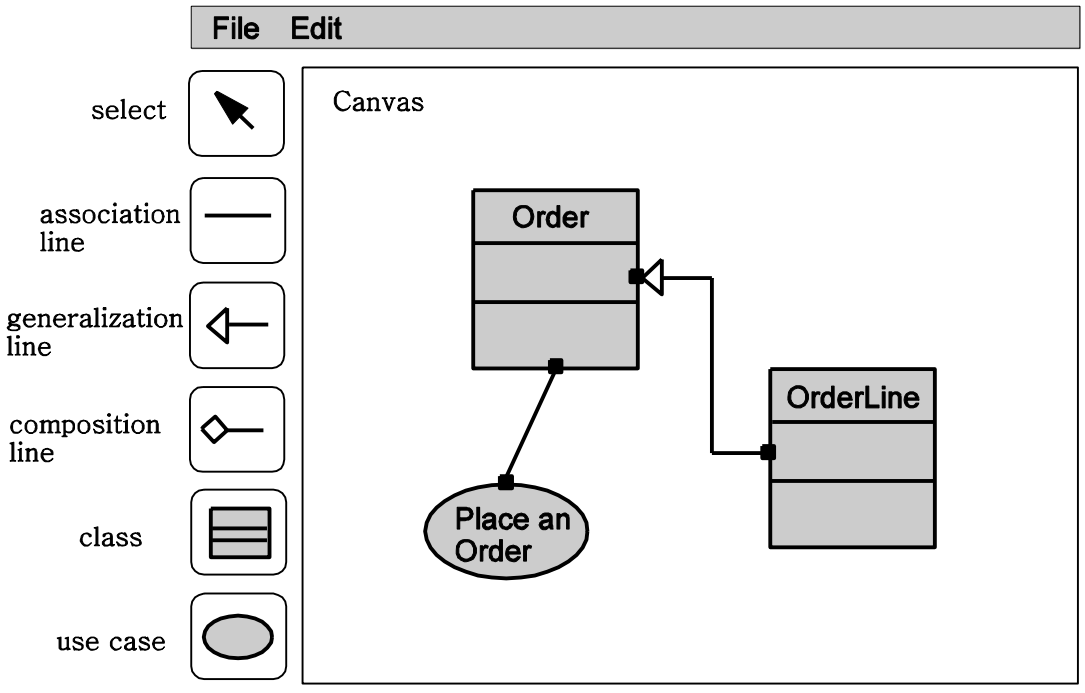
- **Summary:**

This document describes the requirements of UML editor which should be completed as your final term project.

- **Introduction:**

MacroSoft Ltd. is a major player in the market computer-aided software engineering tools. MacroSoft's goal is to ship a new product called Macro UML editor in year 2010. Drawing UML diagrams by pens or pencil is inconvenient for programmers. The goal is to produce a software tool that can draw UML diagrams easily and make changes easily.

The following is a prototype UI for the Macro UML editor.



.

This UML editor UI is divided into three regions. The first region allows tool users to place diagrams and make connections. This area is called "canvas area". On the left hand side is a region called "task bar" which contains a lot of buttons. In the first release, you only need to

implement 6 kinds of buttons. However, more buttons are expected to be extended and added, so your architecture must deal with such kind of evolution. The detailed specs of these buttons will be described in later sections of this document.

- **Detailed Specs:**

  In this section, we describe the detailed specs by use cases in UML .There are several terms need to be defined before we continue our detailed specs.

Terminology:

- **basic object (基本物件)：**
  The boxes that represent a class and the ellipses that represent the use cases are said to be “basic objects”. Basic objects can be created by buttons and then moved around in the editing regions. As in common UML semantics, associations can be constructed between two basic objects.
- **connection line (連結物件):**
  A connection line is an entity between basic objects. At current stage of the requirements, we have three types of connection line. They are “association”, “composition” and “generalization”.

- **composite object (複合物件)：**
  Just like many other object-oriented diagram editing tools. A tool user can select a set of objects and group them into a so-called “composite object”. A composite object can be composed from a set of basic objects and this composite object can be composed with other basic objects or composite objects to become a more complicated composite object. Composite object is a type of container which can be implemented by a tree hierarchy. Ungrouping a composite object is equal to tear down the top level of the tree.
- **The depth of an object:**
  As in many other object-oriented diagram editing tool, every object has an attribute called “depth (0-99)”. This attribute concerns the drawing order of the objects. A deeper object is drawn behind a object that has less depth.

Besides, when a set of objects are overlapped. The object on top of the objects always intercepts the mouse and keyboard events. For example, when a mouse clicks on two overlapped objects, only the object on the top receives the mouse/keyboard events.

UseCase A.1 Creating an UML object

precondition:

This use case only applies to the buttons of “class” and “use case”

1. Click the button and the button is changed to black color so that users know the current mode is controlled by this button.
2. Move the cursor to editing area。
3. Click at (x,y) with left mouse button，an object (“class” or “use case”) is created at (x,y) as its top-left corner.
4. Users can repeat the steps 2-3 to create same kind of objects until the mode is changed.

Alternatives 1.a

When other buttons in task bar is clicked, this mode is terminated and other mode control the behaviors of mouse.

UseCase B.1 Creating an UML connection line

precondition:

This use case applies to the buttons of “association line”, “generalization line”, and “composition line” in the task bar.

1. In editing area, a user presses at (x,y) by mouse left button and (x,y) is inside some basic object A (“class box” or “use case ellipse”). The user keeps the mouse button pressed.
2. The user drags the mouse
3. The user releases (mouse released) the mouse button at (x’,y’) and (x’,y’) is inside some basic object B.
4. A connection line is created between A and B. The arrow type of the connection line is displayed at the port of A.

Alternative 1.a

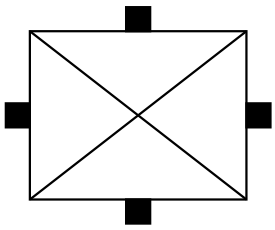
(x,y) is not inside any basic objects. The following actions from mouse pressed -> mouse drag -> mouse released make no effects.

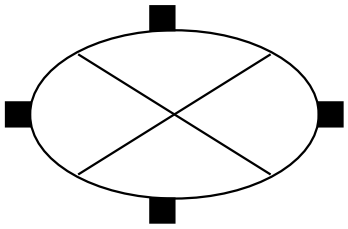
Alternative 3.a

(x',y') is not inside any basic objects. No connection line is created.

Definition:

Please assume each basic object has 4 ports as in the following figure. A connection line is established between two ports of basic objects. A simple rule is used to compute which port should be chosen when mouse is clicked. For example, when a (x,y) is pressed, the distances to the four ports can be computed. The port that has the shortest distance to (x,y) should be chosen as the connection port for the new connection line.





UseCase C.1 Select/Unselect a single objects

precondition:

This use case only apply to the cases where "select" button is the current mode.

definition

We says an object is "selected" if its four connection ports are displayed as the figure shown above. On the other hand, when an object is not selected, the four connection ports should be hidden.

1. A user clicks at (x,y) which is inside an object X.

2. If any objects in the editing area are “selected”, un-select them.

3. Make X as an “selected” object by showing its four connection ports.

Alternative 1.a

if (x,y) is not inside any objects, do nothing.

un-select any objects that are previously selected.

UseCase C.2 Select/Unselect a group of objects

precondition

This use case only applies to the cases where “select” button control the current mode.

1. A user press mouse button at (x1,y1) without releasing the button and (x1,y1) is not inside any objects.

2. un-select any objects that are previously selected.

3. The user drags the mouse ◦

4. The user releases the mouse button at (x2,y2)

5. (x1,y1,x2,y2) for a rectangular area. Any objects that are completely inside the area are selected.

Alternative 4.a (x1,y1,x2,y2) form a rectangular region and no objects are included. This case is equal to un-select all the objects.

UseCase D.1 Group objects

precondition:

This use case only applies to the case that at least two objects are selected under “select” mode.

1. The user goes to Edit Menu and click “group”
2. The selected objects are grouped into a composite object.

UseCase D.2 UnGroup objects

precondition:

This use cases only applies to the case when only one composite object has been selected.

1. The user goes to Edit Menu and click “ungroup”.
2. The composite object is torn down one level.

UseCase E.1 Move objects

precondition:

This use case only applies to “select” mode.

1. The user presses mouse button at (x,y) without releasing the button and (x,y) is inside some object X (including composite objects).
2. The user drags the mouse ◦
3. The user releases button at (x’,y’).
4. Object X is moved to the new position (x’,y’) ◦
5. All the connection lines that connect to object X are redrawn.

Note: (x,y) could fall inside more than one objects (they are overlapped). In this case, the object on top (with less depth) should be moved.

UseCase F.1 Change Object Name

precondition:

This use case only applies to “select” mode and only one basic object is selected.

1. The user goes to Edit Menu and click “change object name”.
2. A window pops up. The window has one text area, OK button and Cancel button.
3. The user enters string at text area and press OK. The string entered should be displayed on the basic objects.

Alternative 3.a Cancel

1. User click Cancel button, the popped window disappears and do nothing.