Visual ◆ Paradigm

# Observer Pattern Tutorial

Written Date : October 21, 2009

This tutorial is aimed to guide the definition and application of Gang of Four (GoF) observer design pattern. By reading this tutorial, you will know how to develop a model for the observer pattern, and how to apply it in practice.

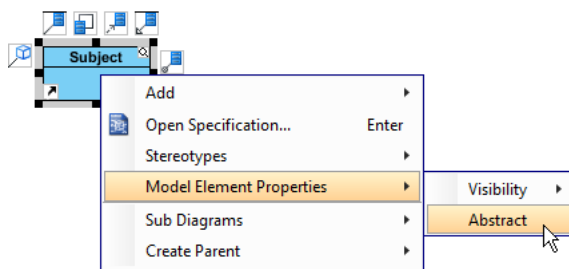## Modeling Design Pattern with Class Diagram

1. Create a new project *Design Patterns*.
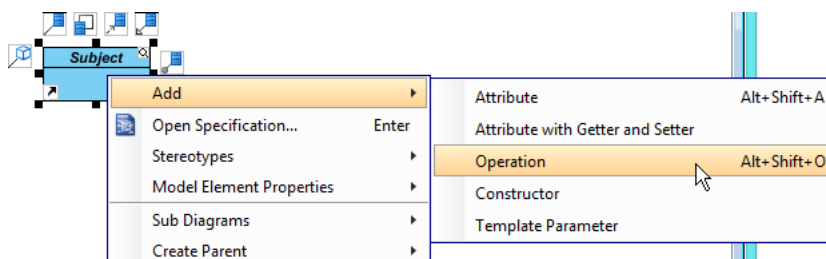
2. Create a class diagram *Observer*.
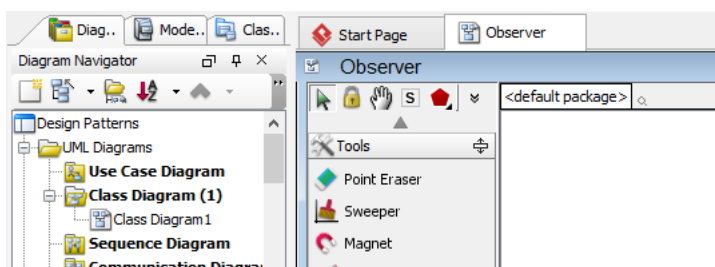
3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Subject*.
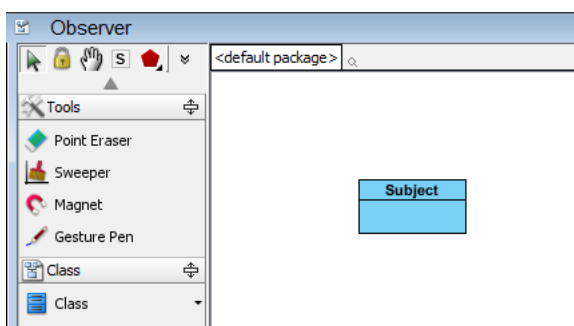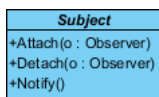
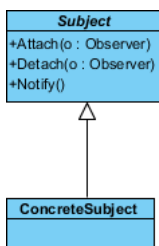4. Right-click on *Subject*, and select **Model Element Properties** > **Abstract** to set it as abstract.

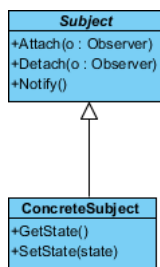5. Right-click on *Subject* class, and select **Add** > **Operation** from the popup menu.

6. Name the operation *Attach(o : Observer)*.

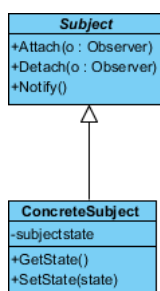7. Repeat step 5 and 6 to create the remaining two operations : *Detach (o : Observer)*, *Notify()*.

8. Move the mouse cursor over the *Subject* class, and drag out **Generalization** > **Class** to create a subclass *ConcreteSubject*.
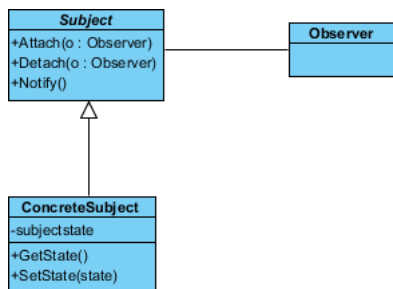
9. Repeat steps 5 and 6 to create the following operations in *ConcreteSubject: GetState()*, *SetState(state)*.

```
┌─────────────────────┐
│      Subject        │
├─────────────────────┤
│+Attach(o : Observer)│
│+Detach(o : Observer)│
│+Notify()            │
└─────────────────────┘
          △
          │
┌─────────────────────┐
│   ConcreteSubject   │
├─────────────────────┤
│+GetState()          │
│+SetState(state)     │
└─────────────────────┘
```

10. Right-click on the *ConcreteSubject* class, and select **Add** > **Attribute** from the popup menu. Name the attribute *subjectstate*.

```
┌─────────────────────┐
│      Subject        │
├─────────────────────┤
│+Attach(o : Observer)│
│+Detach(o : Observer)│
│+Notify()            │
└─────────────────────┘
          △
          │
┌─────────────────────┐
│   ConcreteSubject   │
├─────────────────────┤
│-subjectstate        │
├─────────────────────┤
│+GetState()          │
│+SetState(state)     │
└─────────────────────┘
```

11. Move the mouse cursor over the *Subject* class, and drag out **Association** > **Class** to create an associated class *Observer*.

```
┌─────────────────────┐        ┌──────────┐
│      Subject        │        │ Observer │
├─────────────────────┤        ├──────────┤
│+Attach(o : Observer)│────────│          │
│+Detach(o : Observer)│        └──────────┘
│+Notify()            │
└─────────────────────┘
          △
          │
┌─────────────────────┐
│   ConcreteSubject   │
├─────────────────────┤
│-subjectstate        │
├─────────────────────┤
│+GetState()          │
│+SetState(state)     │
└─────────────────────┘
```
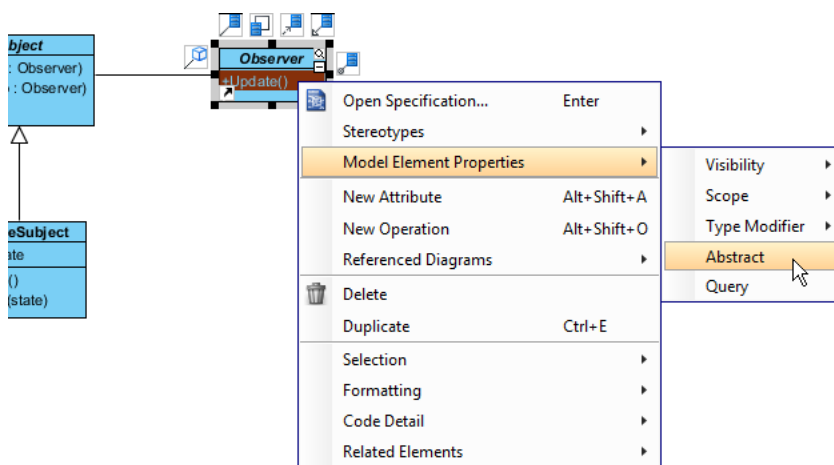
12. Right-click an *Observer*, and select **Model Element Properties** > **Abstract** to set it as abstract.

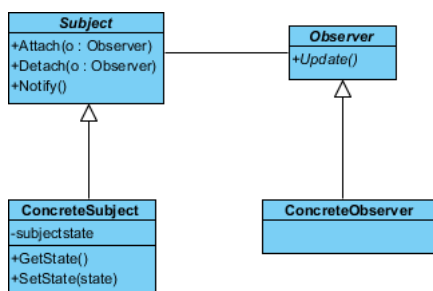13. Repeat steps 5 and 6 to create the following operation in *Observer: Update()*.
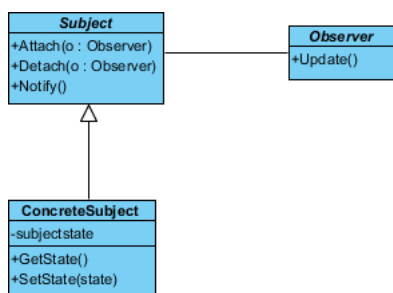

14. Right-click on *Update()*, and select **Model Element Properties** > **Abstract** to set it as abstract.


15. Move the mouse cursor over the *Observer* class, and drag out **Generalization** > **Class** to create a subclass *ConcreteObserver*.

16. *ConcreteObserver* will inherits the operations from *Observer*. Right-click on *ConcreteObserver* and select **Related Elements** > **Realize all Interfaces** from the popup menu.
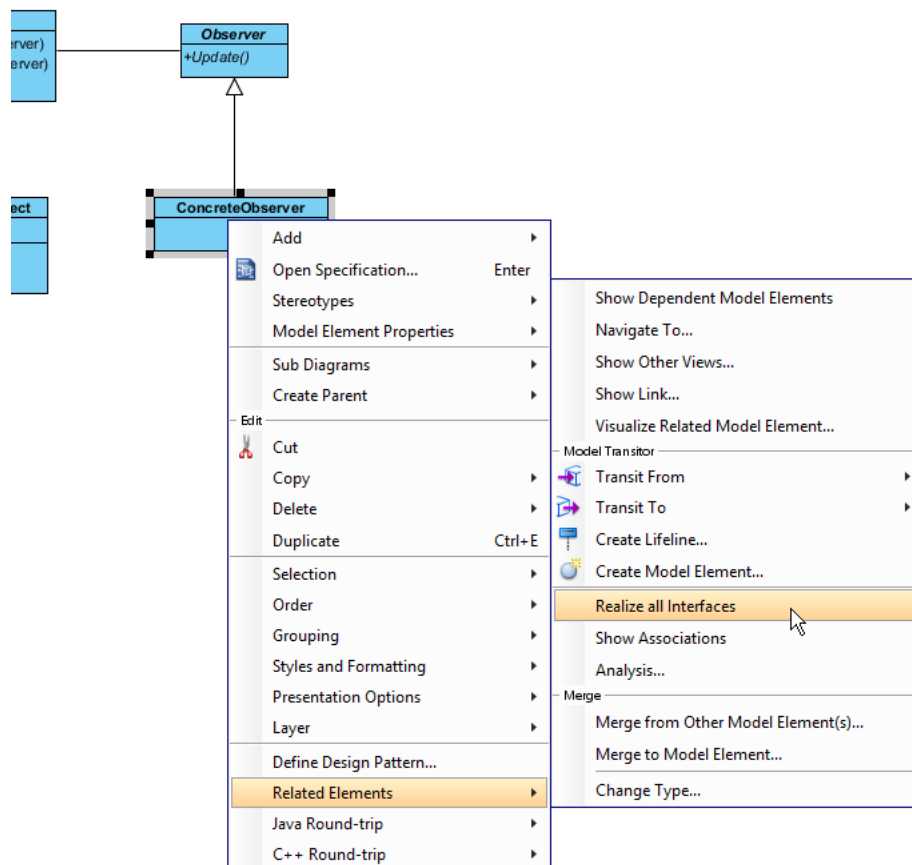

17. Right-click on the *ConcreteObserver* class, and select **Add** > **Attribute** from the popup menu. Name the attribute *observerstate*. Up to now, the diagram should look like:
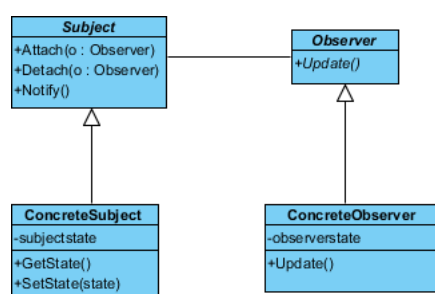
18. In practice, there may be multiple concrete subjects and observers. To represent this, stereotype the class *ConcreteSubject* and *ConcreteObserver* as **PTN Cloneable**. Right-click on *ConcreteSubject* and select **Stereotypes** > **Stereotypes...** from the popup menu.
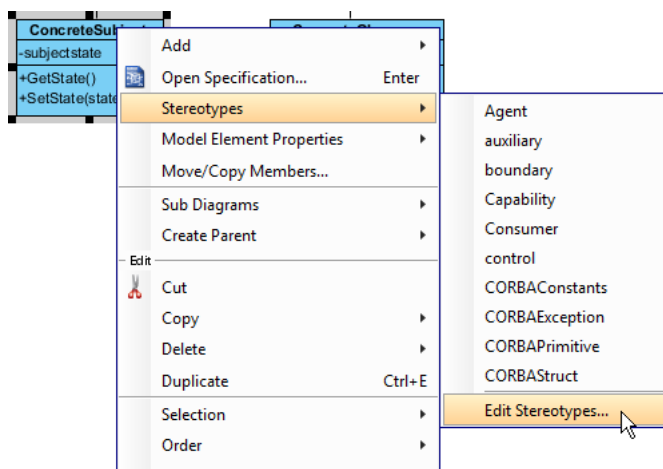

19. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to *ConcreteSubject* class. Click **OK** to confirm.
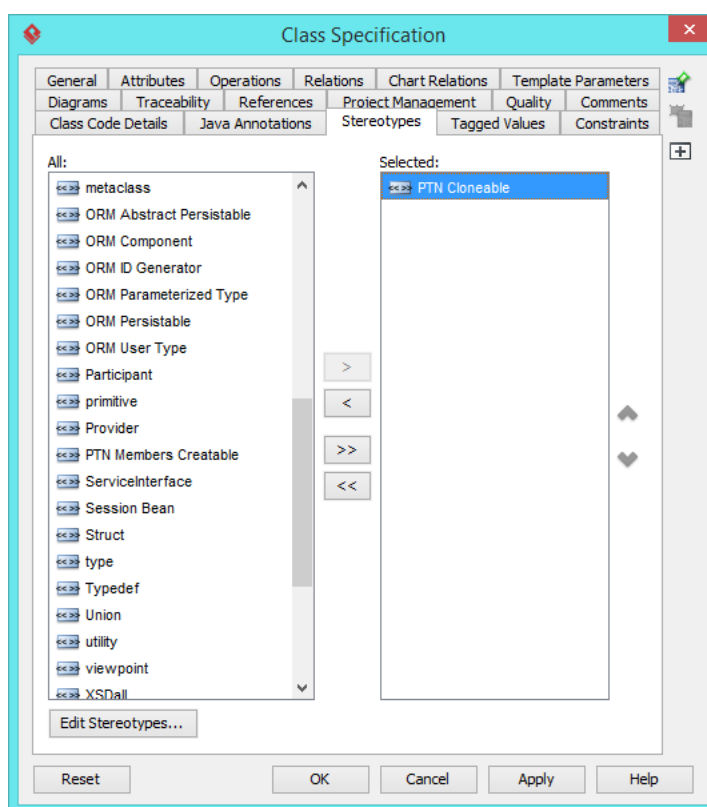
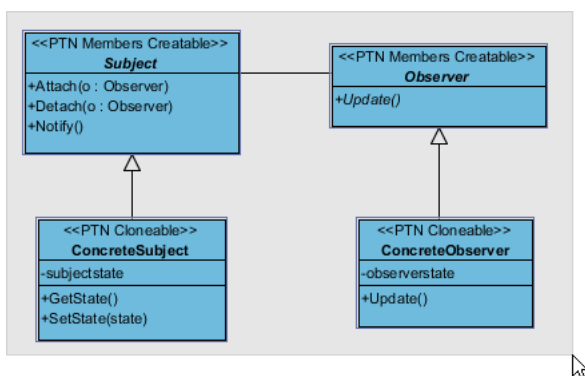20. Repeat steps 18 and 19 on *ConcreteObserver*.
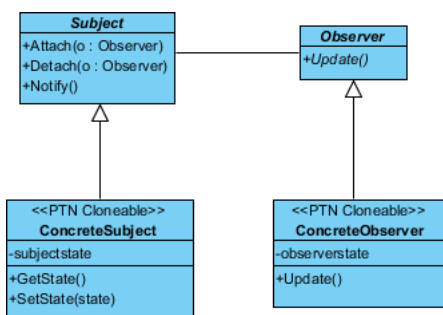
21. In practice, there are domain specific operations in subject and observer. To represent this, stereotype the class *Subject* and *Observer* as **PTN Members Creatable**. Repeat steps 18 and 19 to stereotype *Subject* and *Observer* as **PTN Members Creatable**.
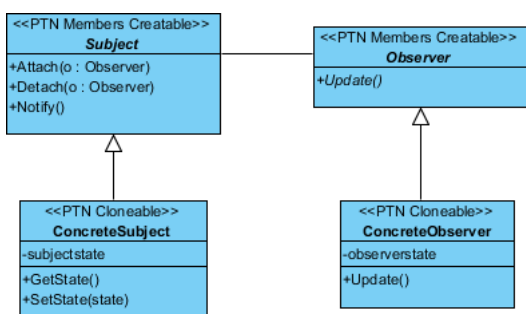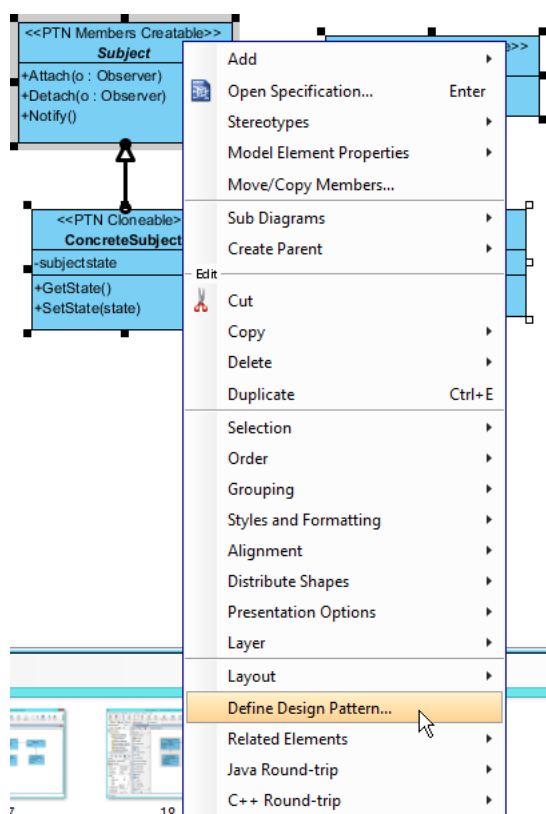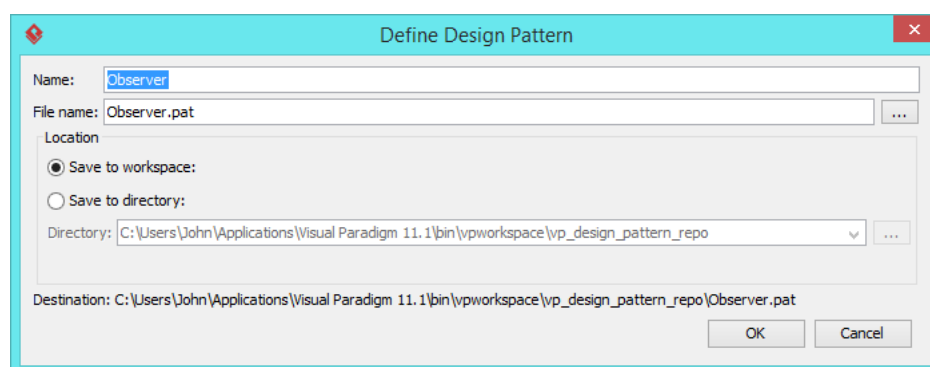

Defining Pattern

1. Select all classes on the class diagram.

2.   Right-click on the selection and select **Define Design Pattern...** from the popup menu.

3.   In the **Define Design Pattern** dialog box, specify the pattern name *Observer*. Keep the file name as is. Click **OK** to proceed.
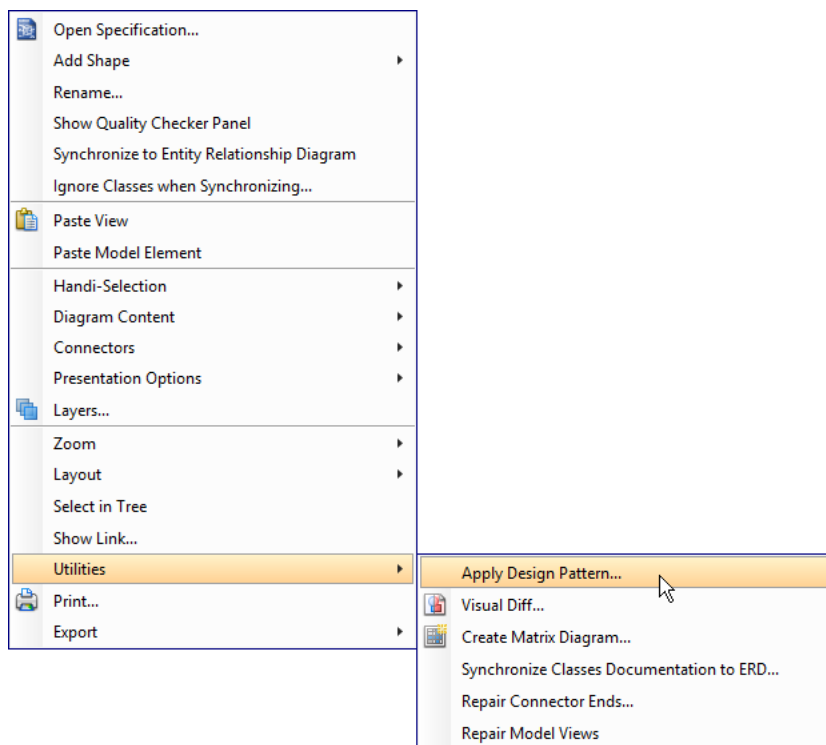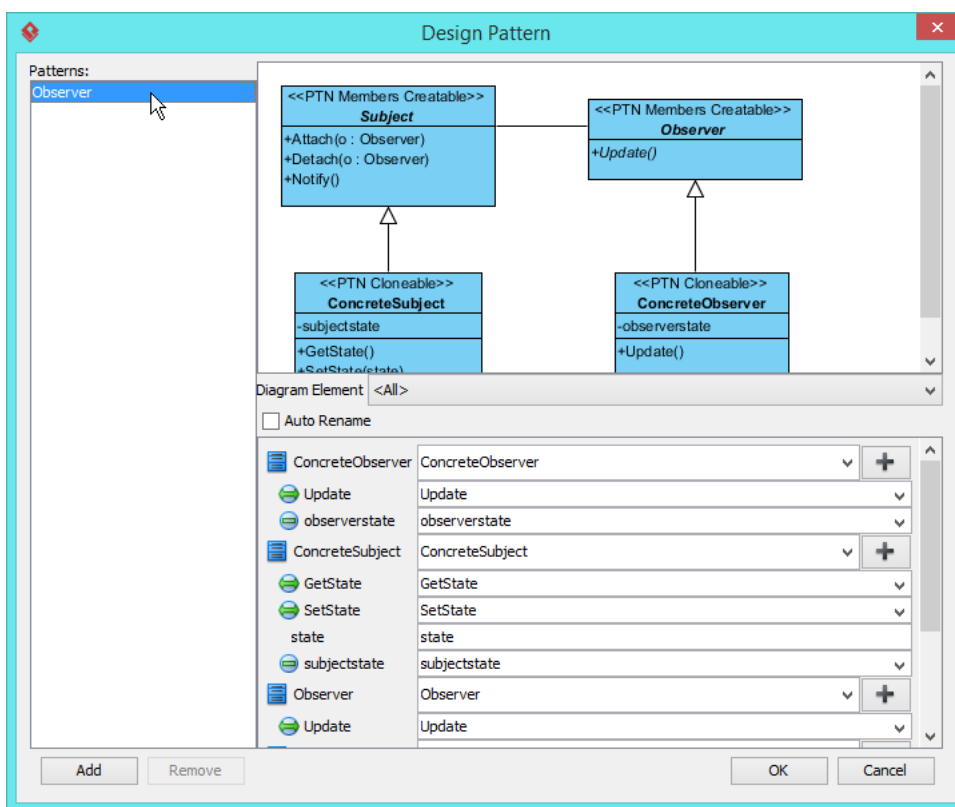

Applying Design Pattern on Class Diagram

In this section, we are going to apply the observer pattern to model a diagram editor for observing changes of model, and calling various panes like the property and overview panes to update their content.

1.   Create a new project *Diagram Editor*.

2.   Create a class diagram *Domain Model*.

3. Right-click on the class diagram and select **Utilities** > **Apply Design Pattern...** from the popup menu.

Open Specification...	
Add Shape	▶
Rename...	
Show Quality Checker Panel	
Synchronize to Entity Relationship Diagram	
Ignore Classes when Synchronizing...	
Paste View	
Paste Model Element	
Handi-Selection	▶
Diagram Content	▶
Connectors	▶
Presentation Options	▶
Layers...	
Zoom	▶
Layout	▶
Select in Tree	
Show Link...	
Utilities	▶
Print...	
Export	▶

| |
|---|
| Apply Design Pattern... |
| Visual Diff... |
| Create Matrix Diagram... |
| Synchronize Classes Documentation to ERD... |
| Repair Connector Ends... |
| Repair Model Views |

4. In the **Design Pattern** dialog box, select *Observer* from the list of patterns.

5.   At the bottom pane, rename classes *Subject*, *Observer*, *ConcreteSubject* and *ConcreteObserver*
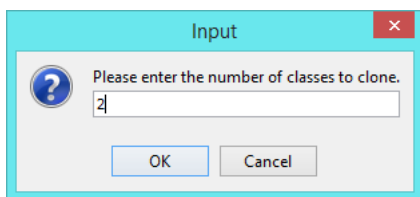   to *Shape*, *Pane*, *Rectangle* and *PropertyPane* respectively.
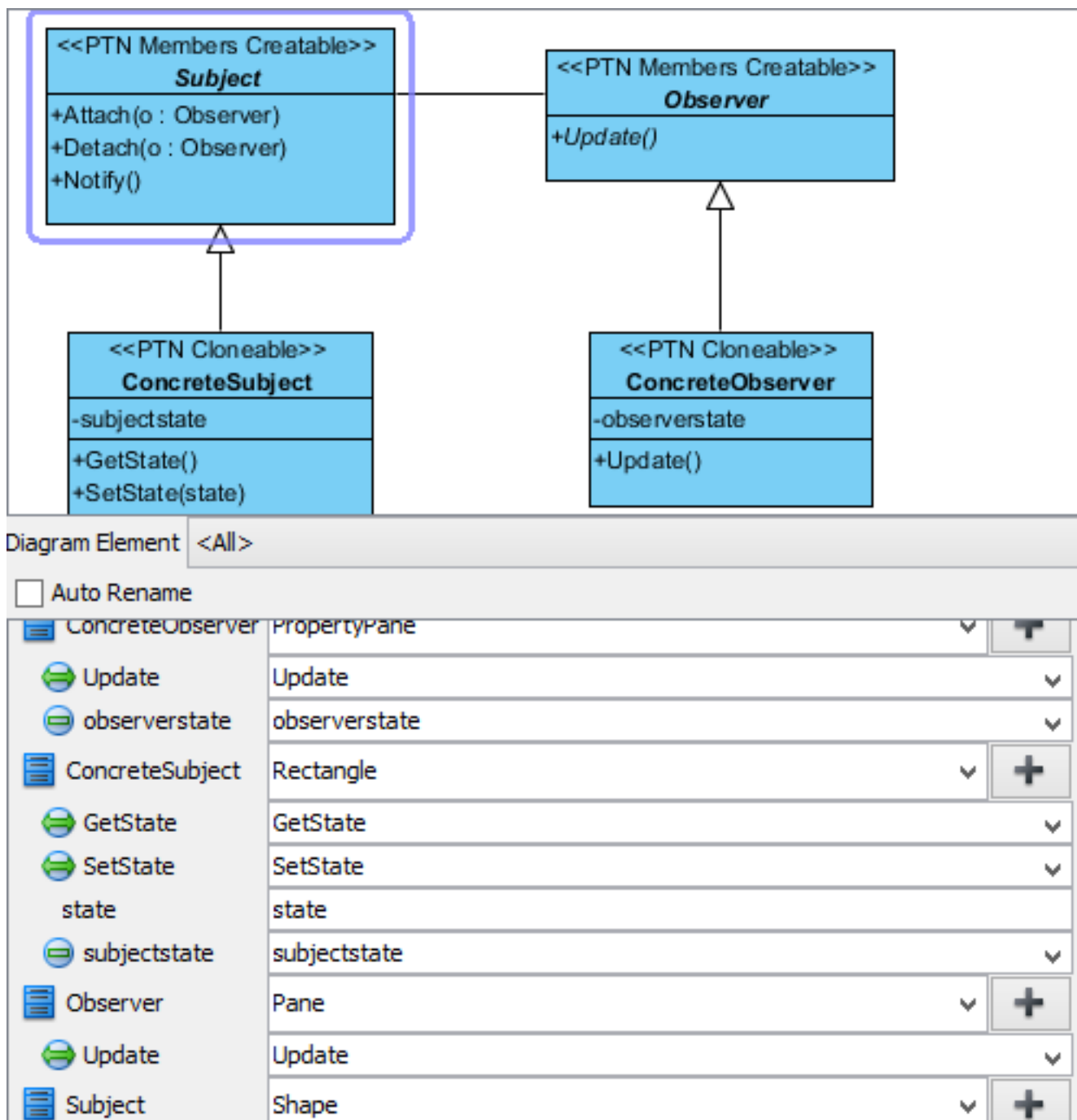
6. Besides rectangle, there are more types of shape like circle and triangle. Select *ConcreteSubject* from the overview pane.
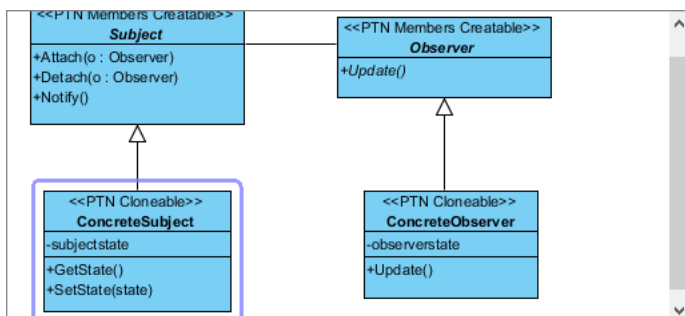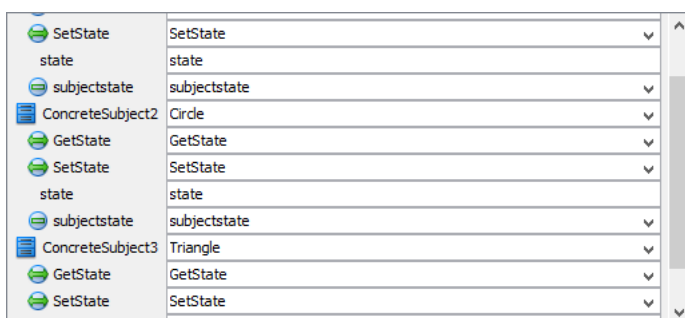

7. Click on the + button next to the class name and select **Clone...** from the popup menu.
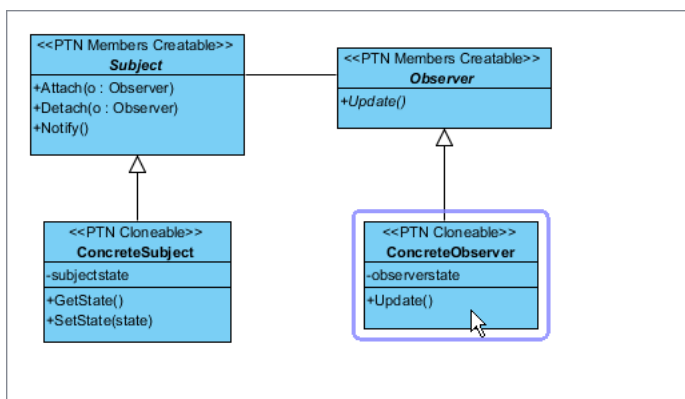

8. Enter 2, which is the number of classes to clone, and click **OK** to confirm.

9. At the bottom pane, rename *ConcreteSubject2* and *ConcreteSubject3* to *Circle* and *Triangle*.

10. For observers, there are also panes like overview pane. Select *ConcreteObserver* from the overview pane.

11. Click on the + button next to the class name and select **Clone...** from the popup menu.

12. Enter 1, which is the number of classes to clone, and click **OK** to confirm.

13. At the bottom pane, rename *ConcreteObserver2* to *OverviewPane*.
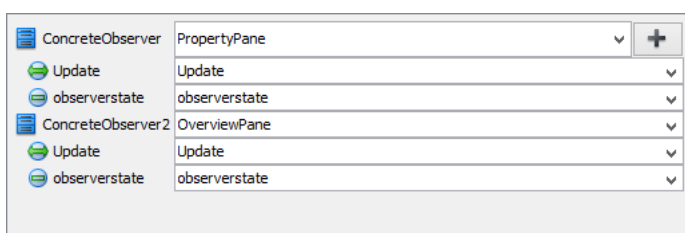
14. Click **OK** to confirm. Here is the diagram formed:
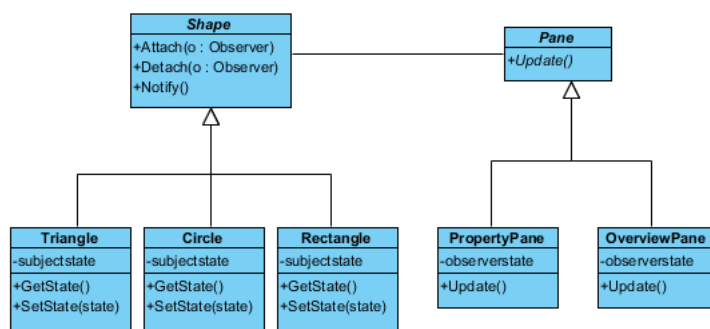
Resources
1. [Design Patterns.vpp]
2. [Observer.pat]

Related Links

• [Full set of UML tools and UML diagrams](#)

Visual Paradigm home page
(https://www.visual-paradigm.com/)

Visual Paradigm tutorials
(https://www.visual-paradigm.com/tutorials/)