

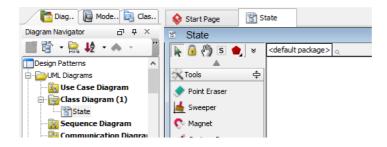
State Pattern Tutorial

Written Date : October 27, 2009

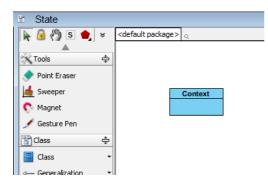
This tutorial is aimed to guide the definition and application of <u>Gang of Four (GoF)</u> state <u>design</u> <u>pattern</u>. By reading this tutorial, you will know how to develop a model for the state pattern, and how to apply it in practice.

Modeling Design Pattern with Class Diagram

- 1. Create a new project Design Patterns.
- 2. Create a class diagram *State*.



3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Context*.



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

Conte	xt			_	
		Add	•	Attribute	Alt+Shift+A
		Open Specification	Enter	Attribute with Getter and Setter	
		Stereotypes	+	Operation	Alt+Shift+O
		Model Element Properties	•	Constructor	
		Sub Diagrams	+	Template Parameter	
		Create Parent	•		

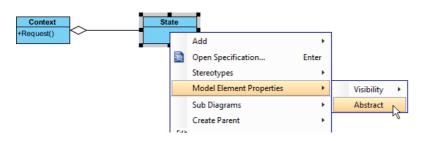
5. Name the operation *Request()*.



6. Move the mouse cursor over the *Context* class, and drag out **Aggregation** > **Class** to create an associated class *State*.



7. Right-click on *State*, and select **Model Element Properties > Abstract** to set it as abstract.



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

Context +Request()	State	State						
		Add	•	Attribute Alt+Shift+A				
	2	Open Specification	Enter	Attribute with Getter and Setter				
		Stereotypes	•	Operation Alt+Shift+O				
		Model Element Properties	•	Constructor				
		Sub Diagrams	+	Template Parameter				
		Create Parent						

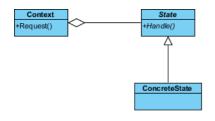
9. Name the operation *Handle()*.



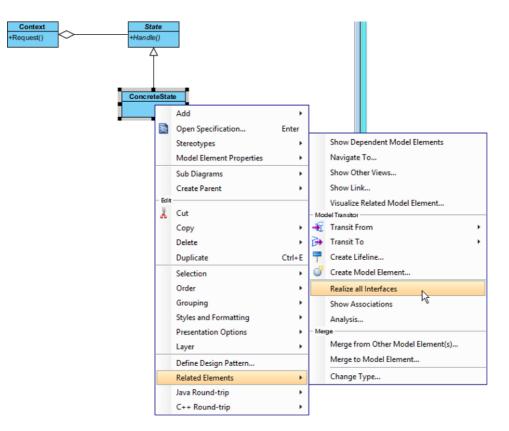
10. Right-click on *Handle*, and select **Model Element Properties > Abstract** to set it as abstract.

Context +Request() +Handk	ate e()			_	
		Open Specification	Enter		
		Stereotypes	+		
		Model Element Properties		Visibility	•
		New Attribute	Alt+Shift+A	Scope	+
		New Operation	Alt+Shift+O	Type Modifier	+
		Referenced Diagrams	•	Abstract	
	Ŵ	Delete		Query K	
		Duplicate	Ctrl+E		
		Selection	+		
		Formatting	•		
		Code Detail	•		
		Related Elements	•		

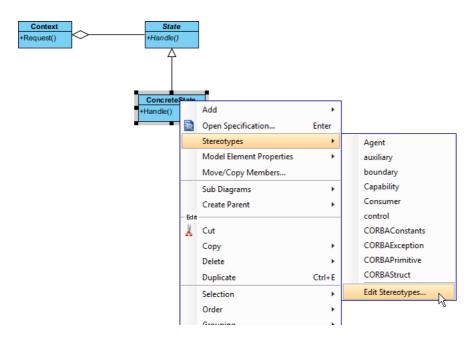
11. Move the mouse cursor over the *State* class, and drag out **Generalization** > **Class** to create subclasses *ConcreteState*.



12. We need to make the concrete states inherit operations from the state class. Right-click on *ConcreteState* and select **Related Elements** > **Realize all Interfaces** from the popup menu.



 In practice, there may be multiple concrete states. To represent this, stereotype the class *ConcreteState* as **PTN Cloneable**. Right-click on *ConcreteState* and select **Stereotypes** > **Stereotypes...** from the popup menu.

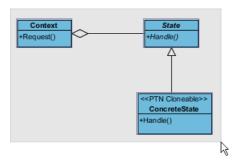


14. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to *ConcreteState* class. Click **OK** to confirm.

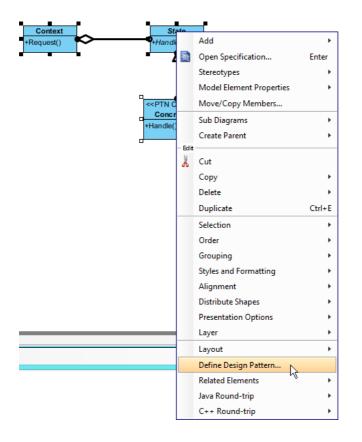
\$		(lass Spe	cificatio	n			×
General Diagrams Class Code	Attributes Traceabil e Details	Operations ity Referen Java Annotati		iect Manao eotypes	Taggeo	Templat Quality Values	te Parameters Comments Constraints	* *
Mes: Mes: Mes: ORM	s smmentationCla face sage Driven i sageType class Abstract Pe Component ID Generato Parameteriz Persistable User Type cipant tive ider Members Cru icelnterface sion Bean	Bean rrsistable rr red Type	▲	Selected	: N Cloneat		*	
Reset			ОК	Cano	el	Apply	Help	

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 *State*. Keep the file name as is. Click **OK** to proceed.

\$	Define Design Pattern
Name:	State
File name:	State.pat
Location	
Save	to workspace:
⊖ Save	to directory:
Director	/: C:\Users\John\Applications\Visual Paradigm 11.1\bin\vpworkspace\vp_design_pattern_repo v
Destination	n: C:\Users\John\Applications\Visual Paradigm 11.1\bin\vpworkspace\vp_design_pattern_repo\State.pat
	OK Cancel

Applying Design Pattern on Class Diagram

In this section, we are going to apply the state pattern in modeling a life.

- 1. Create a new project Life.
- 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			
(în	Paste View			
	Paste Model Element			
	Handi-Selection	+		
	Diagram Content	•		
	Connectors	•		
	Presentation Options	•		
6	Layers			
	Zoom	•		
	Layout	•		
	Select in Tree			
	Show Link			
	Utilities	•		Apply Design Pattern
٢	Print			Visual Diff
	Export	•		Create Matrix Diagram
				Synchronize Classes Documentation to ERD
				Repair Connector Ends
				Repair Model Views

&	Design Pattern	
Patterns: State	Context +Request() < <ptn cloneable<br="">ConcreteState +Handle()</ptn>	
	Diagram Element <all></all>	
	Diagram Element <all></all>	
		~
	Auto Rename	~
	Auto Rename	~
	Auto Rename GoncreteState ConcreteState Gontext Context Gequest Request	~
	Auto Rename GoncreteState ConcreteState Handle Gontext Context	~

4.

5. At the bottom pane, rename Context, State and ConcreteState to Life, LifeState and IdleState.

Context +Request() +Handle() ConcreteState +Handle()								
Diagram Element <a< td=""><td>ll></td><td>~</td></a<>	ll>	~						
Auto Rename								
ConcreteState	IdleState	+						
😝 Handle	Handle	~						
Context	Left	~						
😂 Request	Request	~						
📑 State	LiftState	~						
😝 Handle	Handle	~						

6. We need 2 more concrete states for up and down. Click on the + button at the ConcreteState row and select **Clone...** from the popup menu.

reteState	IdleState	¥	+
ndle	Handle		Clone N
ext	Left		~
juest	Request		¥
	l iffState		

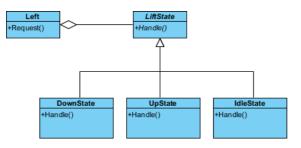
7. Enter 2 to be the number of classes to clone. Click **OK** to confirm.



8. Rename ConcreteState2 and ConcreteState3 to UpState and DownState.

ConcreteState	IdleState	~	÷
😝 Handle	Handle		¥
ConcreteState2	UpState		¥
😂 Handle	Handle		~
ConcreteState3	DownState		¥
😝 Handle	Handle		¥
Context	Left		¥
😝 Request	Request		¥
📑 State	LiftState		~
😝 Handle	Handle		¥

- 9. Click **OK** to apply the pattern to diagram.
- 10. Tidy up the diagram. Here is result:



Resources

1. <u>Design Patterns.vpp</u>

2. <u>State.pat</u>

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/)