3.1 Object Adapter Pattern

The adapter pattern can be adapted by an adapter object. Adapter works between the incompatible interfaces of the classes which work together.

Documentation

Need to add pic1!!!!!!!

Figure 3.1.1 Object Adapter Pattern Class Model

The Figure 3.1.1 describes the class model of the Object Adapter Pattern. It contains the class GenericComponent. The clients access methods of the Target class. The implementation of these functions is done by the adapter. The representation is the role. The adapter calls methods of the class GenericAdaptee.

The Adaptee class uses the algorithms from the existing classes.The algorithms are shown as the method specificOperation.GenericAdaptee is the subclasses of Adaptee. There are several Adaptee roles which are the instantiation of the pattern.The Adapter has different Adaptees and their subclasses can use it. The last type is the Compartment Type Object Adapter pattern. It contains methods by which the client uses the components.

Need to add pic2!!!!!!!

Figure 3.1.2 Adapter Pattern Role Model

There is a role Adapter. The client can access it by the interface. This contains independent methods and function calls of the Adaptee classes. The method newOperation is the example of independent Adaptee method.The adaptee role is created by the useAdaptee relation.The Adaptee class plays for the eponymous role.

The roles adapter and adaptee are instantiated atleast once. From this, the two occurrence constraints can be derived. The relation useAdaptee represents an adapter calling methods from its adaptee. Thus every adapter always has at least one adaptee. But it is also possible that an adapter can have several Adaptees.But, not every Adaptee has to be one at any one time can be used by an Adapter.

Evaluation

1. The design pattern allows the transparent extension of the functions of an object at runtime.

2. The pattern should be for functions, such that every object of the class shoul not apply.

3. The pattern helps in static inheritance extension of a class, which is not practical.

4. The design pattern allows the use of classes in unforeseen cases, in which the interfaces of clients and adaptees do not have to be compatible. This feature is described in the applicability component.

5. The design pattern allows subclasses of the adaptees to be adapted.

6. The design pattern contains the following classes: client, target, adapter, and adaptee.

7. The Adapter class inherits from the Target class.

8. The Adapter class has a pointer to the Adaptee class.

9. The Adapter class delegates method calls to the Adaptee class.

10. The class Client acts with the class Target.

11. The Target class defines the interface that clients can access.

12. The class Adaptee provides certain algorithms and the interface to be adapted to disposal.

13. The adapter class adapts the interface from adapter to target.

14. Multiple interfaces can be adapted with the design pattern.

15. The design pattern makes it difficult to override the behavior of adaptees.

16. An adapter may involve translating method names and parameter lists restrict or even implement their own algorithms.

17. Classes that have a built-in interface adaptation can be reused.

18. There are several ways a built-in interface for adaptation of a class. This feature is only explained in the Implementation section.
a) The class in which the interface adaptation is to be installed is a superclass. The subclasses of these implement the abstract methods of the superclass and use methods of the adaptees in this implementation.
b) The class into which the interface adaptation is to be installed delegates method calls to corresponding other objects, which then use methods of the adaptees.

19. It may be necessary for two classes to be mutually adapted to each other. Both classes are then once adapters and once Adaptee.

20. The following roles participate in the design pattern: client, adapter, and adaptee.

21. The composition constraints from the Role Relationship Matrix [Rie09, p. 28] apply.

The Feature Table Explains all the properties as shown below.

	Features	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18a	18b	19	20	21
GoF	Texual	X	X	X	X	X				X	X	X	X	X	X	X	X	X	X	X	X		
	Code																						
	Graphically	X					X	X	X	X	X												
Riehle	Texual	X								X	X			X									
	Graphically									X	X											X	X
CROM	Graphically	X		X		X	X	X	X	X	X	X	X	X	X							X	X

Figure 3.1.3 Object Adapter Pattern features in 3 representations

Role adapters and GenericAdaptee classes explicitly indicate the characteristics 1,3, 12 and 13. Adaptee and its subclasses fulfill characteristics 5 and 14. The Adapter role plays the functionalities of class Target by GenericComponent.The inheritance leads to the same interfaces of the Target and Adapter classes.

If an object of class GenericComponent plays the role Adapter, it takes over automatically the interface of the played role. The client is not separated in this model. The features 8th and 9th are defined by the relation useAdaptee. The client role through the methods of Compartment Types are shown for feature 20.

????

3.2 Class Adapter Pattern

The Class Adapter Pattern is similar to the Object Adapter pattern. In this variant, only one class can be adapted by an adapter object,

Documentation

Need to add pic1!!!!!!!

Figure 3.2.1 Class Adapter Pattern Class Model

Figure 3.2.1 shows the class model of Class Adapter variant. It has the class GenericAdaptee as natural type. The ClassAdapterCompartment is a compartment which plays the roles of Adapter. Here there is no separate class GenericAdaptee as Natural type. Clients access the adaptee to use its algorithms. Parameters are also adapted by the methods of ClassAdapterCompartment. GenericAdaptee can indicate which adaptee should be used.

Need to add pic2!!!!!!!

Figure 3.2.2 Class Adapter Pattern Role Model

Figure 3.2.2 shows the role diagram. It consists solely of the role adapter, which also has in this variant of the adapter pattern of the adapter and dependent independent methods. When an adaptee class fulfills this role, it inherits the interface of the role. This fulfills the same function as the inheritance of Adaptee's Class Adapters, which is described by the Gang of Four. This allows a client to access the GenericAdaptee class without knowing the interface of that class. Instead, the methods of role adapters are used. So the adaptation takes place

Evaluation

The following features were pure by the Gang of Four, unless otherwise stated textually described.

1. The design pattern converts one interface to another. This feature is documented in both intention and applicability.

2. This pattern allows for 2 incompatible interfaces to work together which is described in intention part.

3. The adapter class can implement the additional functionalities which the adaptee does not own. This is a component motivation characteristics.

4. The Pattern allows the compatibility in the cases in which the interfaces of clients and adaptees are not compatible.

5. The pattern has client, target, adapter, and adaptee classes.

6. The Adapter class inherits from the Target class.

7. The adapter class inherits from class Adaptee.

8. The Adapter class delegates method calls to the Adaptee class.

9. The class Client acts with the class Target.

10. The Target class defines the interface that clients can access.

11. The Adaptee class provides certain algorithms and the interface to be adapted.

12. The adapter class adapts the interface from adapter to target.

13. The pattern can use an adapter which the adaptee can adapt.

14. The design pattern overrides adaptee behavior.

15. The adapter pattern can directly use the behavior of the adaptees. There is no need for an additional pointer.

16. An adapter may involve in translating method names and parameter lists. It can restrict or even implement their own algorithms.

17. The classes those have a built-in interface adaptation can be reused easily..

18. There are several ways to built-in interface adaptation for a class.
a) The superclass has the interface adaptation.
 The implementation is done by the abstract methods of superclass and adaptees use them.
b) The class having the interface adaptation delegates method calls to corresponding the other objects and use the methods of the adaptees.

19. Different classes have multiple inheritance by multiple inheritance adaptation.

20. The role participants in the pattern are client, adapter, and adaptee.

21. The composition constraints from the Role Relationship Matrix [Rie09, p. 28] apply.

The Feature Table Explains all the properties as shown below.

	Features	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18a	18b	19	20	21
GoF	Texual	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X		
	Code	X		X			X		X								X						
	Graphically			X		X	X	X	X	X													
Riehle	Texual	X							X	X			X										
	Graphically								X	X												X	X
CROM	Graphically	X		X		X	X	X	X	X	X	X	X	X	X	X							

Figure 3.2.3 Class Adapter Pattern features in 3 representations

Features 1, 3, 11 and 12 can be represented by the CROM. This can be explained by adapters specifying by the role adapters and generic class adaptee. The characteristics 6th, 7th, 10th and 13th to 15th are defined by the adaptee-Class that can play the adapter role. The 8th characteristic is defined by the relation useAdaptee.

3.3 Object & Class Adapter Pattern Comparision