3.1 Object Adapter Pattern

The adapter pattern can be adapted by an adapter object.

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 can access methods of the class Target. The implementation of these functions is via the adapter. The role is realized in this representation. The adapter calls methods of the class GenericAdaptee.

This class can play the role Adaptee and this is the representative of the already existing classes from which algorithms are used. The algorithms are shown through in the model as the method specificOperation. GenericAdaptee classes are also subclasses of Adaptees. There are several Adaptee roles which are the instantiation of the design pattern. The Adapter has different Adaptees and their subclasses can use it. The third class is the Compartment Type Object Adapter pattern. It contains methods about which the client functions the components.

Need to add pic2!!!!!!!

Figure 3.1.2 Adapter Pattern Role Model

There is a role Adapter. It will be in her interface that clients can access. This contains beside independent methods including those that call functions of the adaptee classes. The independent Adaptee methods are represented in the model by newOperation. The adaptee role is created by the useAdaptee relation, that an adapter can call the methods of an adapter. 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. It has every adapter always has at least one adaptee. But it is also possible that an adapter several Adaptees.But, not every Adaptee has to be one at any one time can be used by Adapter.

Evaluation

The following features are explained in **Gang of Four** .

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

2. The design pattern should be for functions that do not use every object of a class needs to be applied.

3. The design pattern helps if a static inheritance as a feature extension a class is not practical. This feature will also be textual in the part Consequences detected.

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 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.

???? Need to check last conclusion part.