

上面的类图说明了对象适配器实例。它说明了在Shape类中声明的BoundingBox请求如何被转换成在TextView类中定义的GetExtent请求。由于TextShape将TextView的接口与Shape的接口进行了匹配，因此绘图编辑器就可以复用原先并不兼容的TextView类。

Adapter时常还要负责提供那些被匹配的类所没有提供的功能，上面的类图中说明了适配器如何实现这些职责。由于绘图编辑器允许用户交互的将每一个Shape对象“拖动”到一个新的位置，而TextView设计中没有这种功能。我们可以实现TextShape类的CreateManipulator操作，从而增加这个缺少的功能，这个操作返回相应的Manipulator子类的一个实例。

Manipulator是一个抽象类，它所描述的对象知道如何驱动Shape类响应相应的用户输入，例如将图形拖动到一个新的位置。对应于不同形状的图形，Manipulator有不同的子类；例如子类TextManipulator对应于TextShape。TextShape通过返回一个TextManipulator实例，增加了TextView中缺少而Shape需要的功能。

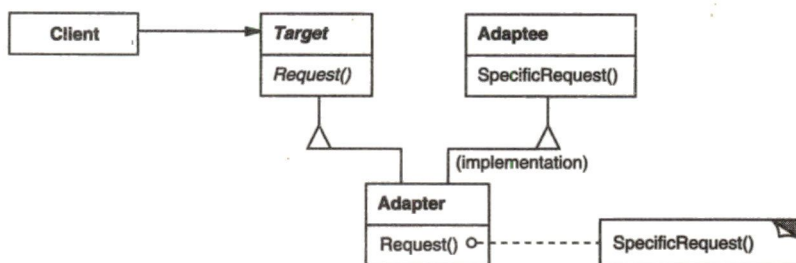
#### 4. 适用性

以下情况使用Adapter模式

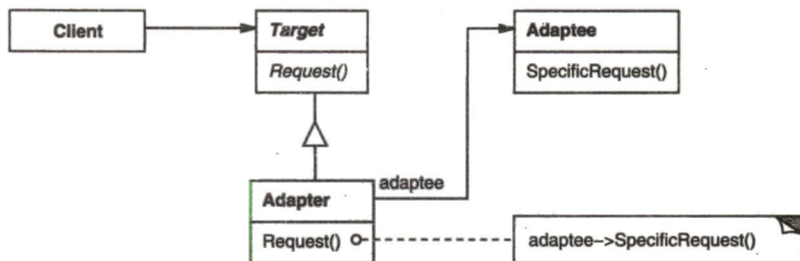
- 你想使用一个已经存在的类，而它的接口不符合你的需求。
- 你想创建一个可以复用的类，该类可以与其他不相关的类或不可预见的类（即那些接口可能不一定兼容的类）协同工作。
- （仅适用于对象Adapter）你想使用一些已经存在的子类，但是不可能对每一个都进行子类化以匹配它们的接口。对象适配器可以适配它的父类接口。

#### 5. 结构

类适配器使用多重继承对一个接口与另一个接口进行匹配，如下图所示。



对象适配器依赖于对象组合，如下图所示。



#### 6. 参与者

- Target (Shape)
  - 定义Client使用的与特定领域相关的接口。
- Client (DrawingEditor)