

Decorator

- 將一些新的行為動態加入到某個已存在的物件中
- 讓未來的使用者可以擴充某個特定的物件行為
- 在沒有AOP的年代，可以用Decorator來幫忙達成AOP
- 通常稱之為「包裝者」(Wrapper)

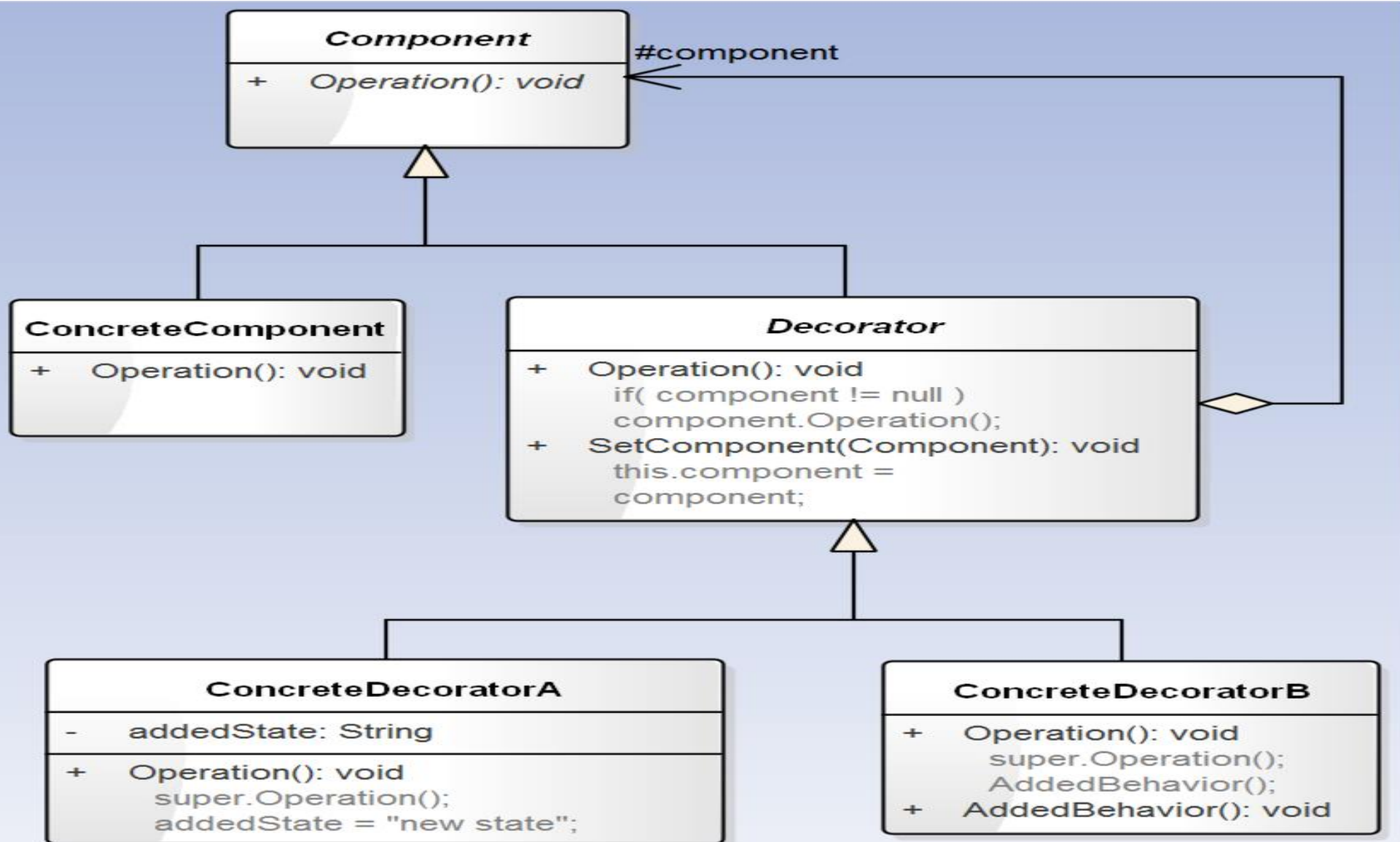
事前與事後

- 在已存在的系統(事前)中，要如何讓舊有的功能可以加入新的特徵(事後)？
- 在進行Framework設計(事前)時，如何讓系統可以容納未來的改變(事後)？

解決方案

舊瓶裝新酒，使用者還是透過舊瓶的介面，
但可以得到新酒的服務
把需要修改的行為，抽象化成Component，
然後新增一個Decorator的Abstract Class，
把該需要修改的Component做成
Decorator的Child(只能有一個)

結構



Decorator可以應用在以下幾個時機：

- 針對已存在的系統，進行新功能的Append
- 在設計Framework時，預留未來的Developer針對系統修正的空間

案例描述

某系統要開發一套可擴充的Authentication & Authority的系統，該系統可以透過使用者的ID來確認使用者是否屬於該系統，使用者ID則記錄在Database中

使用者基本可以用的功能，則可以透過設定檔的方式直接取得

但該系統希望能夠擴充Authentication的方式，可以讓Client決定是否採用LDAP或是Database的Table來進行認證

除此之外，有些額外的功能，也希望能夠直接透過擴充的方式，取得使用者的額外功能

要如何設計才能讓這個系統具備上述的特性呢？

Extensible A&A

