[**.NET中的设计模式---由吃龙虾想到的**](http://www.cnblogs.com/niyw/archive/2010/07/18/1780253.html)

　　今天吃小龙虾的时候忽然想到了以前一个湖北朋友讲的虾的故事.这位朋友是湖北人,据他说在他小时候他们那里很多虾,特别是夏天雨后,满地爬的都是.因为传说那是美国对付中国的秘密武器,居然没有人敢吃.后来偶然有人提了半桶换卖了5块钱回来,慢慢的大家认识到虾的价值,随后就有了今天吃得五香小龙虾.然而近几年出现的"洗虾粉"又让人心有余悸.回想起昨天看到亚力山大<[召集讨论设计模式是语言表达能力低下的产物](http://www.cnblogs.com/Alexander-Lee/archive/2010/07/17/1779905.html)>一贴.突然发现**设计模式**和**虾**居然有相似的命运.

　　设计模式从出现到发展,到普及再到今天,人们对它的认识也同样经过了"陌生"->"认识使用"->"批评".设计模式在程序员心目的地位中也经历了"新奇"->"圣经"->"怀疑".然而,不管人们如何认识龙虾,从始至终虾还是虾,没有变过.同样,设计模式自出现现到今天,一样没有变,变得只是次第更新的软件设计开发技术,变得只是一代又一代的程序员.

　　设计模式是20世纪60-70年代的软件危机之后大规模软件的发展的结果.这里有个问题,什么是设计模式.相比能够真正回答正确的人不多,真正理解的人也不多,因此经历的辉煌之后随之而来的怀疑就不可避免了,不过也正是这些批判怀疑促进了设计模式更进一的发展.

　　什么是设计模式,我也无法给出权威的答案,不过我敢肯定,设计模式不是<gof23种设计模式>中的UML图,也不是一行行的示例代码,它有的只是一种思想,一种思考问题的方式,就像<加勒比海盗>中的<海盗法典>,它只是指导,而并非律法.

　　既然只是一种指导,那么在使用的时候就应当根据实际情况使用它或不使用它,用一种方法实现它或用另外一种方法实现它.**指导告诉我们一个解决问题的方向,而例子告诉我们到达目的地众多路径中的一条,它可能是最优,也可能是最差.**<gof23种设计模式>一书不仅详细的表达了什么是设计模式,而且有很多生动的例子.这些例子用来具体说明每种设计模式.**然而在今天看来,那些Smalltalk或C++的例子却成了一幅无形的枷锁,让很多人去套用而不知道因地制宜的发散扩展.**比如当不断的用if ...else if选择工厂或策略的时候,就不会想到在.NET中可以使用反射来去掉让人生厌的if...else.

　　今天讨论的是.NET中的设计模式.大家都知道设计模式出现时还没有.NET,Gof不可能预见会有C#的出现,写论文的时候也不可能知道有**反射,委托,事件等**技术的出现.因而亚历山大同志单凭**"比如C#内置了事件机制，那么Observer还有意思"**就**认为设计模式已过时就显得过于武断**了.

**.NET中有很多特性使得设计模式的实现更加简单,然而并不是说一定要用这些特性来实现设计模式,,就好比有了乘法,有时候我们用加法实现1+1=2可能更好,你不能因此而否定了乘法.**同样,既然"事件"能够实现"Observer"的效果,那么一个事件不正是一个Observer吗?Observer不是加入到.NET中了吗!它只不过换了个名字,本质没有变.

　　所以说我们在用设计模式的时候首先应该深入挖掘语言的潜力,而后结合设计模式的思想,创造独有的应用设计模式.或许在C++中实现一个Observer有很多行代码,在.NET中一行即可.书中的例子只是学习的一种途径,没有创新,生搬硬套只能是鹦鹉学舌,邯郸学步,最终也只是贻笑大方了.