

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

C#面向对象设计模式纵横谈

20. Chain of Responsibility 职责链（行为型模式）

李建忠

jianzhong.lee@gmail.com

设计模式论坛:

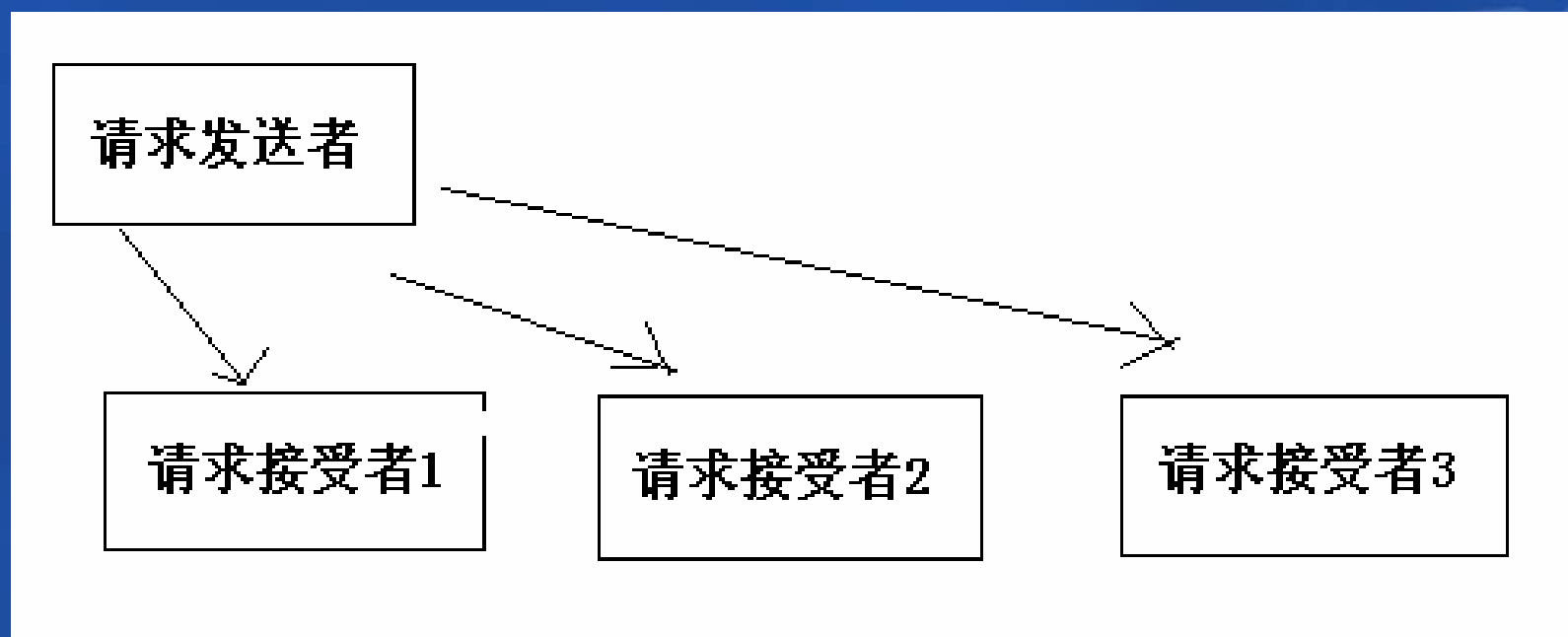
forum.softcompass.com

上海祝成科技 高级培训讲师

www.softcompass.com

请求的发送者与接受者

某些对象请求的接受者可能多种多样，变化无常.....



动机 (Motivation)

在软件构建过程中，一个请求可能被多个对象处理，但是每个请求在运行时只能有一个接受者，如果显式指定，将必不可少地带来请求发送者与接受者的紧耦合。

如何使请求的发送者不需要指定具体的接受者？让请求的接受者自己在运行时决定来处理请求，从而使两者解耦。

意图 (Intent)

使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链传递请求，直到有一个对象处理它为止。

——《设计模式》GoF

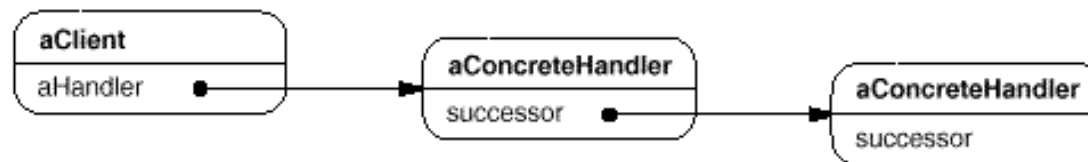
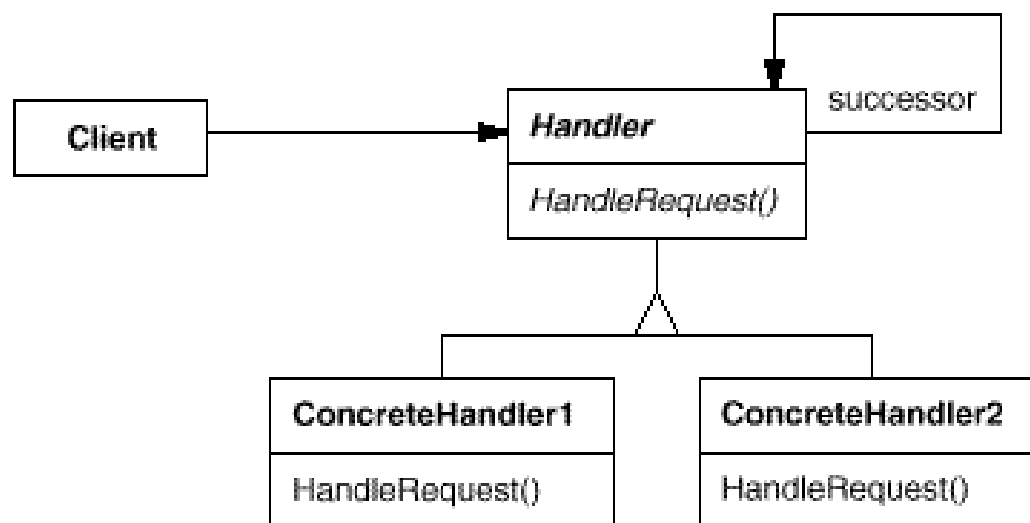
您的潜力，我们的动力

例说Chain of Responsibility应用

Microsoft
微软(中国)有限公司

Codes in VS.NET

结构 (Structure)



Chain of Responsibility 模式的几个要点

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- Chain of Responsibility 模式的应用场合在于“一个请求可能有多个接受者，但是最后真正的接受者只有一个”，只有这时候请求发送者与接受者的耦合才有可能出现“变化脆弱”的症状，职责链的目的就是将二者解耦，从而更好地应对变化。
- 应用了Chain of Responsibility 模式后，对象的职责分派将更具灵活性。我们可以在运行时动态添加/修改请求的处理职责。
- 如果请求传递到职责链的末尾仍得不到处理，应该有一个合理的缺省机制。这也是每一个接受对象的责任，而不是发出请求的对象的责任。

推荐资源


- 《设计模式：可复用面向对象软件的基础》 GoF
- 《面向对象分析与设计》 Grady Booch
- 《敏捷软件开发：原则、模式与实践》 Robert C. Martin
- 《重构：改善既有代码的设计》 Martin Fowler
- 《Refactoring to Patterns》 Joshua Kerievsky

获取更多MSDN资源

- **MSDN中文网站**
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.mspx>



Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问

提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts