Smart Client Software Factory(SCSF)

**Module 模块**

在组合的界面里独立运行的项目

独立的相互不依赖的.net项目，没有强引用

**Shell 公用窗体**

组合界面的主人

public class Program : FormShellApplication<WorkItem, Form1>

**CAB**

1. 实例化显示shell
2. 寻找profilecatalog.xml，读取程序集
3. 搜寻他们中的moduleinit类，并调用其中的load方法

**Workitem 容器组件**

逻辑上放在一个组的class集合

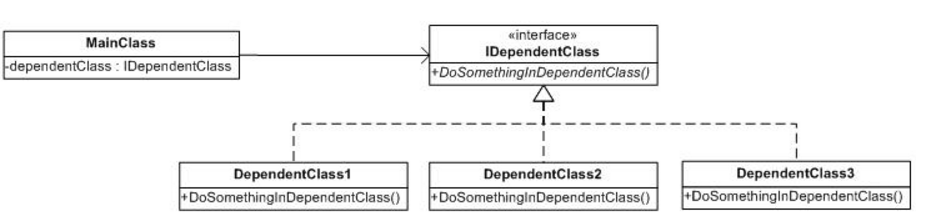
树状结构，RootWorkItem 含其他workitem他们又含其他workitem

这个树状结构连接了modules，即使不同的modules没有相互引用，我们可以使用这个workitem的树状结构去使用其他modules里的组件

**依赖注入**

允许一个主class可以在不直接引用第二个class的情况下调用它

主class耦合与接口



构造器注入，接口注入，setter注入

**反转控制**

好莱坞原则 不要call我们，我们会call你

Sort的时候传人的比较函数，被别人调用

三个项目可以相互通信却不相互引用，workitems就是为了这个目的而被设计出来的，我们可以让每个project里都放置一个workitem，把代码放在他们的collections里面，共享workitem从而共享code

**[ServiceDependency]**

告诉cab框架，当它在装载模块的时候，它需要寻找一个合适的workitem去注入

三种方式

ComponentDependency(string id)

ServiceDependency

CreateNew 泛型方法

**Service**

CAB定义为一种为其他组件提供功能的松耦合的支持类

在CAB中，每一种service被暴露在一个类或一个相关的接口，从而成为我们提供功能的一个大门

Services Item item collection

三种方法加入service

Add AddNew

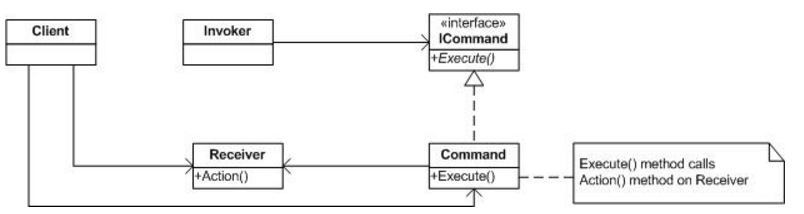
XML configuration

Service属性 当一个module被装载时，CAB会使用反射去寻找所有程序集里的public class，看看谁含有Service属性，所有这些类都得到初始化，并且加到CAB application里的Services collection

两种方式获得Service

Get

Command 将请求封装在一个对象里，根据不同的请求来确定客户



职责分离

Command helloCommand = RootWorkItem.Commands["HelloCommand"];

ToolStripItem toolStripItem = this.Shell.mainToolStrip.Items["toolStripButton1"];

helloCommand.AddInvoker(toolStripItem, "Click");

[CommandHandler("HelloCommand")]

public void HelloHandler(object sender, EventArgs e)

{

MessageBox.Show("Hello world using CAB commands");

}

[CommandHandler("EnableDisableHelloCommand")]

public void EnableDisableHelloCommandHandler(object sender, EventArgs e)

{

Command helloCommand = \_workItem.Commands["HelloCommand"];

if (helloCommand.Status == CommandStatus.Enabled)

helloCommand.Status = CommandStatus.Disabled;

// Change this to the line below if you want to hide the cabCommandToolStripButton

//helloCommand.Status = CommandStatus.Unavailable;

else

helloCommand.Status = CommandStatus.Enabled;

}

Event

[EventSubscription("MyEvent")]

public void MyEventHandler(object sender, EventArgs e)

{

MessageBox.Show("Hello from the CAB event handler"); }

workItem.EventTopics["MyEvent"].Fire(this, EventArgs.Empty, null, PublicationScope.Global);

将click事件挂在Command object上