**CAB和SCSF编程**

1. 松散耦合

分割小块，CAB的Service机制缝合，插件式开发，易于开发，扩展和维护

1. CAB提供的预制的框架用于支持松散耦合
2. 运行期模块装载的中央控制，主要有模块遍历和装载服务
3. 一些服务架构
4. WorkItem机制和依赖注入
5. Workspace(容器Frame)和SmartParts(控制容器)
6. UI扩展点
7. 时间发布和订阅服务
8. CAB的一些术语
9. Shell Application：主WinForm应用程序，CAB程序的各个部分的外部容器。它管理CAB的启动过程
10. Shell Form：Shell Application的主窗口。一般来说就是通常所说的Shell。它通常包含Workspaces和用户界面元素，如菜单和工具条
11. Workspace：SmartPart的容器窗口，而SmartPart一般是由WorkItem拥有的。Workspace可以控制SmartPart是显示还是隐藏。CAB提供了一些标准的Workspace类，也可以自定义
12. WorkItem：对象和服务的运行期容器，一般作为CAB应用程序的组成部分存在。可以把它作为逻辑上的子程序或子进程。它是CAB的基本单元。用户所写的业务逻辑一般都在一个或多个WorkItem中。
13. SmartPart：WorkItem所拥有的书的可是展示或视图。它为WorkItem所拥有并显示在Workspace中。SmartPart一般就是WinForm控件的实现，一般包含其他的WinForm控件。除显示数据之外，也允许用户修改数据。
14. Service：为松散耦合所服务的一些支撑类，通常包含一些与其他WorkItem独立的工具方法。
15. Module：一个.NET程序集，为WorkItem，Service，和其他支持类提供的一个物理容器。
16. CAB的组成

CAB主要有三个工程组成：CompositeUI、CompositeUI.WinForms和ObjectBuilder。这三个工程创建三个动态链接库：CompositeUI.dll，CompositeUI.WinForm.dll，ObjectBuilder.dll。

三、简单CAB程序：Walkthrough

1. 简单的CAB程序结构

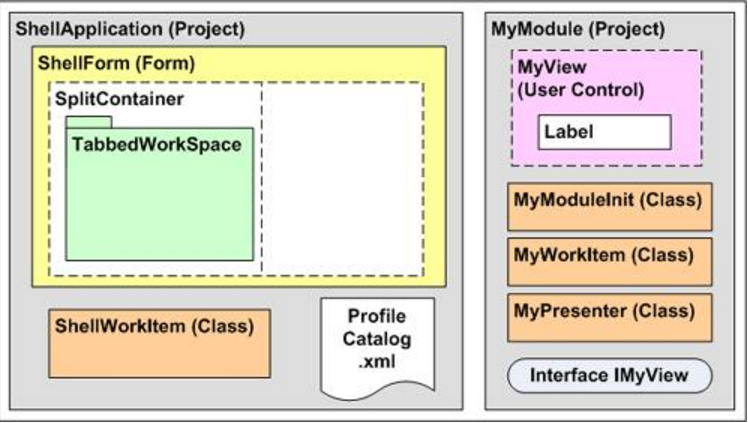
1) ShellApplication.cs文件，Main方法。类ShellApplication衍生自FormShellApplication。在其参数化类型列表中，传递的是Shell的根WorkItem和ShellForm。

2) 对于根WorkItem，可以创建一个类继承自WorkItem。在这个简单程序中该类是空的，什么也不做，仅仅是作为根WorkItem而存在。

3) ShellForm是Shell程序创建的主窗口。主要在其上放置Workspace和UI元素。

4) 程序执行顺序：

CAB框架创建根WorkItem🡪创建ShellForm的一个实例🡪装载ProfileCatalog.xml中定义的Module



2. ProfileCatalog.xml文件结构

每个<ModuleInfo>元素指定一个module

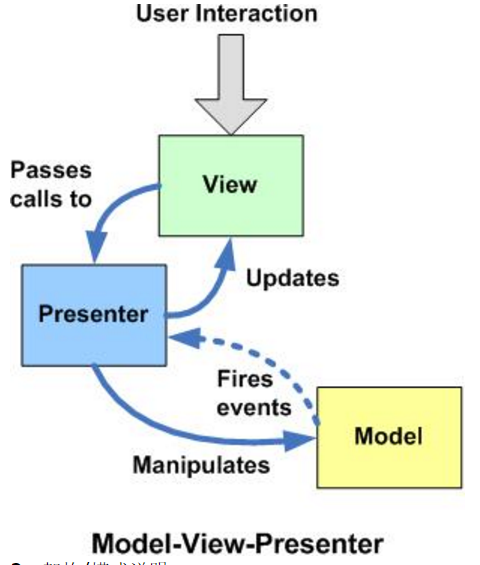
当一个Module被装载进程序的地址空间时，装载器会查找衍生自基类ModuleInite的类。在这个类中可以放一些初始化代码。

在该类中一般有ParentWorkItem的属性，类型为WorkItem。该属性只有写方法，可以从外部写入，但不能读取。它被标为[ServiceDependency]属性，从而导致依赖注入(这是CAB的一个用于松散耦合的一个重要特性)。CAB的模块装载器可以根据该属性知道该模块需要一个特定的类型，也就是说是WorkItem。这里的Service是指通过松散耦合机制访问的任何类。

一旦模块被装载进程序的地址空间，框架就会调用其Load方法。我们在实现Module时一般需要重载该方法。在实现该方法时，一般流程如下：(a) 调用基类的Load方法；(b) 创建一个子WorkItem，并把它放进父WorkItem(该父WorkItem是通过注入得来的)的WorkItem集合中。

四、MVP模式(Model-View-Presenter设计模式)

1. 基本架构图



2. 架构/模式说明

MVP是在客户程序中实现三层架构的基本思想。与CAB相对应，WorkItem代表图中的Model，它包含程序的数据和状态。业务逻辑主要由Presenter所表示。View是控件显示。

3. View与Presenter的交互

定义一个接口负责Presenter和View之间的通信。一般Presenter不需要知道也不想知道该接口在View中是如何实现的。

一般都需要该接口作为Presenter的分离层，如果没有，则只能让Model直接与View交互，从而导致紧耦合。

4. View实现

使用.NET本身的UserControl等等，常规的实现。

第三方工具 CAB Visualizer