

DOTNET 平台组件配置工具的设计与实现

杨扬

(河南省电力通信自动化公司, 河南 郑州 450002)

摘要: 在研究了 DOTNET 平台 Castle 框架的基础上设计出了一个反转控制(IOC)和面向切面编程(AOP)配置文件自动生成的可视化工具,该工具为程序员快速完成应用系统中 IOC 和 AOP 配置文件提供了一个很好的平台,该工具后来还被作成了插件版本。

关键词: RBAC; IOC; Castle; 配置工具

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2010)12-0014-04

Design and implementation of component configuration tools based on DOTNET

YANG Yang

(Henan Electric Power Communication & Automation Company, Zhengzhou 450002, China)

Abstract: In the article, on the basis of in-depth study, a visual config tool for IOC and AOP was designed. In order to simplify programmer's workload when they design application system based on IOC and AOP mechanisms of Castle, which provide a good platform for the generation of config files based on IOC and AOP, an add-in version has been made.

Key words: RBAC IOC; Castle; config tool

反转控制(IOC)和面向切面编程(AOP)技术作为当前比较流行的技术,其优势已受到广泛关注,但是这两项新技术在实际项目上的应用研究却很落后,而且在.NET平台下实现这两项技术没有形成可以广泛套用的框架,使得开发人员在.NET平台下应用IOC和AOP技术的难度和成本大幅度提高。Castle提供了.NET平台下开发IOC和AOP技术的框架,一定程度上解决了开发人员的难题,但是如何高效地实现IOC和AOP技术的开发,至今无人提出较好的建议。

本文主要研究 Castle 框架中的 IOC 和 AOP 机制,目标是设计出一个可视化工具,实现 IOC 和 AOP 机制中配置文件的可视化配置。该工具经过修改和完善也适用于其他框架中的 IOC 和 AOP 配置文件的生成。

1 Castle 框架和相关技术介绍

1.1 Castle 框架介绍

Castle 是 .NET 平台下的一个系统级框架,它是通过将一些成熟开源应用进行无缝整合而形成的一套完善的系统级应用框架,这其中包括了 ActiveRecord、Aspect-Sharp、MonoRail 及 MicroKernel/Windsor。其中 MicroKernel 是 IOC 框架的核心;AspectSharp 是 AOP 的框架。

1.2 IOC 技术介绍

反转控制 IOC(Inversion of Control)^[1],又称依赖注入 DI(Dependency Injection),是面向对象领域新兴的编程思想。简单地说,IOC 就是指由容器来控制程序之间的关系,而非传统实现中的由程序代码直接操控。这也就是所谓“反转控制”的概念所在:控制权由应用代码转到外部容器,控制权的转移,也就是所谓的反转。

1.3 AOP 技术介绍

AOP^[2](Aspect Oriented Programming)是 OOP 的延续,意思是面向切面编程。可以通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种技术。它可以解决 OOP 和过程化方法不能够很好解决的横切(crosscut)问题,如:事务、安全、日志等横切关注。当未来系统变得越来越复杂,横切关注点就成为一个大问题的时候,AOP 就可以很轻松地解决横切关注点的问题。

2 可视化 IOC 和 AOP 配置工具设计思路

可视化 IOC 和 AOP 配置工具是一个在 .NET 平台下开发的 WinForm 程序,这个工具的主要作用就是在 IOC 组件或者 AOP 组件都开发好的条件下,利用这个工具生

成 IOC 或者 AOP 的配置文件,方法是读取现存的 IOC 组件或者 AOP 组件。在工具中进行相应的配置,最后直接生成开发人员所需的配置文件。

该工具的主要功能是通过读入现存的 IOC 组件或 AOP 组件,提取并暂存组件中与配置有关的重要信息,以可视化的方式显示在工具界面中,然后由开发人员操作该工具完成组件对应的配置文件的生成。这个过程可以划分为 5 个阶段(如图 1):读取类文件>分析类文件>显示配置信息>开发人员对参数进行配置>生成配置文件。其中前三层是该工具主要实现的功能。因此,该可视化工具的代码部分分为 3 个层次来编写。分别是:读类文件、分析提取信息、显示信息。

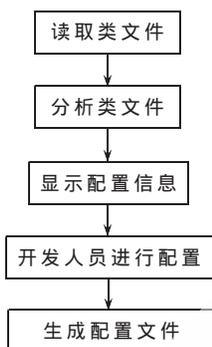


图 1 可视化配置工具的操作流程

第一层主要是完成类文件的读取。在 .NET 平台下使用到了 System.IO 命名空间中文件操作相关的 File 类和 StreamReader 类,最终将整个读取的文件内容以 string 类型返回,以供第二层分析文件使用。

第二层的功能是在第一层的读取文件内容的基础上分析并提取关键字,即取出配置文件需要的重要信息,例如 IOC 反转控制模式主要有设值注入和构造方法注入,设值注入就是对类中的属性进行注入,而构造方法注入是对构造函数中的参数值进行注入。因此类中的属性和构造函数的参数就属于 IOC 配置文件的重要信息,第二层主要完成对类似的重要信息的识别和分类。在第二层中,完成的识别功能主要有:

(1) 识别类文件的命名空间、类名、接口名

命名空间、类名、接口名都是组成 IOC 和 AOP 配置文件的必备元素。

在 IOC 配置文件中,命名空间、类名、接口名常出现在 component 标签的 service 和 type 属性值中,而在 AOP 的配置文件中命名空间出现在 Imports 关键字之后。所以对命名空间、类名、接口名的提取是生成配置文件的必要条件。

(2) 识别类文件中的属性

IOC 反转控制模式中的设值注入就是对类中的属性进行注入,如采用该注入模式设计系统,在 IOC 配置文件中,需要把类中的属性作为配置文件的标签。

(3) 识别类的构造方法中的参数名

IOC 反转控制模式中的构造注入是对构造函数中的参数值进行注入,如果采用该模式设计系统,在 IOC 的配置文件中,需要把构造方法的参数名作为配置文件中的标签。

以上介绍的是第二层所识别的关键字的类别。在程序中关键字的获取主要是通过编写获取方法来实现。获取命名空间使用 GetNamespace 方法;获取类名使用 GetClass 方法;获取接口名使用 GetInterface 方法;获取属性名使用 GetProperty 方法。实现的思路是在遵循一定规约的前提下通过属性在类中的位置特征将其提取出来并且依次保存到一个 ArrayList 对象中。

为了保证该可视化工具识别关键字的准确性,程序员在编写 IOC 组件时要遵循以下规约:

- (1) 要求 IOC 组件类的修饰符为“Public”
- (2) 要求类中的全局变量都声明为“Private”
- (3) 要求属性的修饰符为“Public”
- (4) 要求构造方法的修饰符为“Public”

第三层是页面层。主要负责页面可视化设计。页面层用到了 TreeView、Datagrid、TabControl、ContextMenuStrip 等控件。TreeView 控件以树形结构展示了当前读入的组件;Datagrid 控件用来展示选中组件的构造方法参数或属性,并可以设置其参数值或属性值;ContextMenuStrip 控件用于控制添加和删除组件操作;TabControl 控件则是其他控件的容器。

以下,对该层使用到的主要控件及其触发事件按照其重要程度依次做介绍:

(1) TreeView 控件

该控件的内容占第三层代码的三分之一,主要涉及的触发事件有 NodeMouseClicked(单击)、NodeMouseDoubleClick(双击)、AfterCheck(复选框被选中)。设计的功能主要有:单击 TreeView 控件结点可查看该结点所代表组件的命名空间和类名;双击组件可以显示出该组件的构造方法参数或属性信息;选中复选框后可以将该组件标签添加到 IOC 配置文件中。

(2) TabControl 控件

该控件主要用于分栏。在该系统中一共用到了 2 个 TabControl 控件,第一个用于左栏即控制栏,该栏一共包含了 2 个 TabPage。这两个 TabPage 分别作为 IOC 组件的控制栏和 AOP 组件的控制栏。第二个 TabControl 控件作为显示栏,共包含了 3 个 TabPage:第一个用来显示 IOC 的配置信息;第二个用来显示 AOP 的配置信息;第三个用来实时地显示当前正在配置的 XML 文件。

(3) Datagrid 控件

Datagrid 主要用于分行显示读入的构造方法的参数名或者类的属性名。

(4) ContextMenuStrip 控件

该控件作为快捷菜单出现在单击 TreeView 控件某个结点后,它通常包含 2 个选项:“添加组件”和“删除所有组件”。主要使用到的触发事件就是该控件的单击事件。

(5) OpenFileDialog 控件

打开文件对话框控件,在本系统中,用于读入要配置的组件文件。

(6) SaveFileDialog 控件

保存文件对话框控件,在本系统中,用于保存配置好的 XML 文件到指定的位置。

(7) WebBrowser 控件

Web 浏览器控件,在本系统中,主要用来实时的显示当前正在配置的 XML 文件。

除了上述设计内容外,还添加了两个类文件作为页面层的辅助类。主要负责一些常用的初始化代码及其辅助性方法代码,比如配置文件的初始化方法、DataTable 的初始化方法、还有 DataTable 与 Arraylist 转换函数等。

以上主要介绍了可视化配置工具代码的三层结构。这三层结构是紧密联系的,层与层之间的代码是调用与被调用的关系。它们共同完成了可视化工具所需的功能。完成的主要功能有:

(1) 支持 IOC 中的构造方法注入和设值注入两种注入方式。

(2) 可以对 IOC 的构造方法参数值或者属性值进行设置或者修改;支持数组类型的构造参数。

(3) 可以设置或修改组件 ID。

(4) 可以保存上次关闭时的组件及其状态信息。

(5) 支持 AOP 组件配置;可以在参数设置栏设置拦截器键值、自动识别切面的类型等。

(6) 可以同时配置 AOP 和 IOC 组件。

3 可视化配置工具的实现和应用

通过三层结构,可视化工具就实现了预计的功能;在 IOC 和 AOP 组件存在的前提下,使用该工具可以帮助开发人员完成配置文件的生成。

3.1 IOC 可视化配置工具的实现

可视化工具的 IOC 部分的操作方法和操作步骤如下:

(1) 在左栏 Component 节点上单击鼠标右键,选择添加组件,如图 2,即可把组件信息载入进来;如果选择“删除所有组件”,就可以把树型结构中的所有加载的组件删除,并把 IOC 配置文件中的该组件标签的全部信息相应删除。

(2) 添加完组件后单击该组件可以查看该组件的命名空间和类名;双击组件可以在 IOC 配置栏中显示出该组件的构造方法参数或属性信息。该工具支持两种 IOC 配置模式,即设值注入(属性注入)和构造方法注入。在遵循一定规则的前提下可以自动区分属性注入或构造



图 2 添加组件

方法注入。图 3、图 4 展示的是一个设值注入(属性注入)的实例。



图 3 设置注入中的参数配置



图 4 IOC 参数配置查看

3.2 AOP 可视化配置工具的实现

与 IOC 不同,AOP 的配置也需要读入相应的类文件,根据 AOP 的配置特点,要分别读入拦截器 (Interceptor)、混淆器 (Mixin) 和切面 (Aspect 对象),其中 Mixin 是可选项。

AOP 配置界面中“参数配置”栏可以对 Interceptor、Mixin 设置别名,这就如同 HashTable 中键和值的一一对应关系,在使用到某个 Interceptor 或 Mixin 时可以用它的别名代替。

软件天地 Software Technology

“Aspect 配置”栏显示了当前读入的拦截器 (Interceptor)、混淆器 (Mixin) 和切面信息。根据 Castle 中 AOP 的配置原则进行相应配置后点击“添加该方面”按钮就可以生成对应的 AOP 配置文件,同样可以查看配置结果。

图 5、图 6 展示了 AOP 配置的过程。



图 5 AOP 配置过程

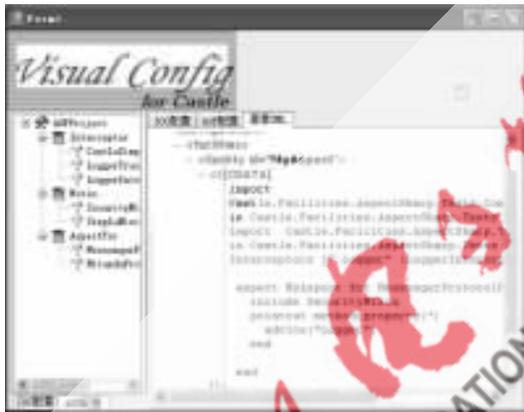


图 6 AOP 配置结果查看

4 可视化工具的插件版本

可视化配置工具的插件版本是把先前在 .NET 平台下做好的 Winform 程序做成 Sharpdevelop 的插件。这是为了满足更多开发者的需要,Sharpdevelop 的灵活和轻巧博得了很多开发人员的喜爱。

可视化工具插件版本主要由两个文件构成,一个是

Dll 文件,负责整个插件的功能逻辑,这部分代码与 .NET 下编写好的代码基本一致;另一个是 Addin 文件,负责该插件在 Sharpdevelop 开发工具中出现的位置和名称等。该插件位置设置为工具菜单下的最后一个选项。名为“组件装配”。

图 7 展示了在 SharpDevelop 平台下打开“组件装配”插件后的效果。

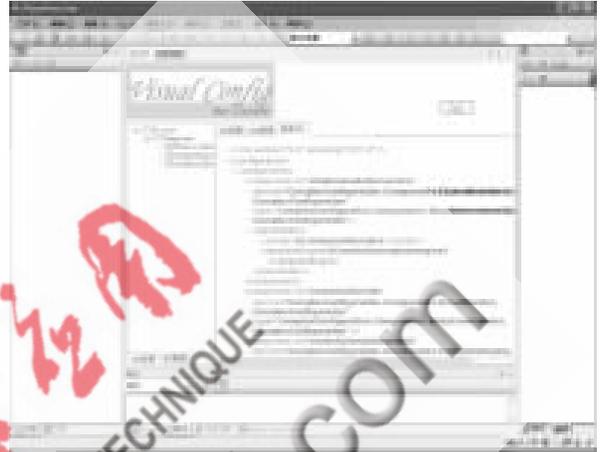


图 7 工具插件版本

参考文献

- [1] 罗时灼.精通 Spring[M].北京:电子工业出版社,2005:14-50.
- [2] SHEN Y, CHEN H. Extending SOFL features for AOP modeling. Engineering of Complex Computer Systems, 2005,5(2):14-15.
- [3] 李志纯.面向 Aspect 编程的应用研究[J].计算机技术与发展,2006,16(5):217-219.
- [4] 刘晓华.精通.NET核心技术原理与构架[M].北京:电子工业出版社,2002,6.
- [5] 张立编著.C# 2.0 宝典[M].北京:电子工业出版社,2006.

(收稿日期:2010-01-26)

作者简介:

杨扬,女,1982年生,硕士,助理工程师,主要研究方向:软件开发。