

基于 AOP 的动态日志系统的实现

牛永洁

(延安大学 计算中心, 陕西 延安 716000)

摘要:为了避免在软件系统中的代码散射(scattering)和代码缠结(tangling),提高软件系统的可维护性与可扩展性,需要将系统必要的但与系统的主要业务无关的一些功能分离出来,单独作为一个模块来实现。然后使用静态织入或者动态织入的方法将这些功能与主要业务联接在一起,组合成一个完整的软件系统。采用面向方面(Aspect-Oriented Programming, AOP)的编程方式将日志功能与主要业务功能组合在一起,并使用反射及动态编译的方法实现了二者结合点的可定制化,进一步提高了系统的灵活性与可扩展性。通过在仪器设备管理系统中的应用,这种方法取得了良好的效果,具有一定的推广价值。

关键词:面向方面;横切关注点;日志系统;反射;编译

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2011)03-0102-03

Implementation of Dynamic Log System Based on AOP

NIU Yong-jie

(Computing Center, Yan'an University, Yan'an 716000, China)

Abstract: In order to avoid code scattering and code tangles then improve maintainability and scalability of system, some functions which were necessary to system but has nothing with the major business, they were separated and implemented as a separate module. Then they were weaved into key business through static or dynamic weaving method, combined into a complete system. In this paper, logging together with key business functions were combined together by AOP, reflection and dynamic compilation were used to be customized. Through application in the equipment management system, this method achieves good results, has a potential value.

Key words: aspect-oriented programming; cross-cutting concerns; log system; reflection; compile

1 面向方面编程

1.1 概述

日志管理在现代软件系统中有着非常重要的作用,可以记录系统所产生的一些关键行为。根据记录的信息可以为系统排错、性能优化提供帮助,或者根据这些信息对用户的一些行为进行跟踪等。但是对日志系统的实现却存在以下两点不足^[1]:

(1)代码交织。在软件系统中,常常将软件的业务逻辑作为软件设计的主线,即核心关注点。同时在系统中也常常包含许多系统级的关注点,如日志记录、系统安全、异常处理等,这类关注点又被称为横切关注点。核心关注点可能需要同时与多种横切关注点交互,最终的结果会导致代码混乱。

(2)代码分散。由于非核心关注点贯穿于整个核心模块,使得相关的实现也遍布于其中。

AOP(Aspect-Oriented Programming)是一种能有效管理、分离横切关注点的程序设计方法学,是面向对象编程思想的一种有效补充^[2]。面向方面编程是施乐公司帕洛阿尔托研究中心(Xerox PARC)在20世纪90年代开发的一种编程范式^[3-5],使开发者更好地将那些横切关注点分离开,从而为程序提供更好的封装性和互操作性。AOP的核心思想是把一个复杂的系统看作是由多个关注点来组合实现。需求分析后的关注点分为两部分:横切关注点和核心业务关注点^[6]。核心业务关注点指的是系统将要满足的业务逻辑需求,表现为各个业务子系统,如设备管理系统等;而横切关注点则是各个业务子系统都可能涉及的一些公共需求,如上文所说的日志系统等。

1.2 AOP 相关概念

AOP 相关概念如下:

(1)联结点(Joint Point):一个联结点是在程序执行过程中的一个特定点,如调用一个方法、类初始化、对象初始化等。

(2)通知(Advice):在某一个特定的联结点处运行的代码称为通知。一般可以分为5种类型,分别是

收稿日期:2010-08-03;修回日期:2010-11-08

基金项目:延安市科学技术研究发展计划项目(2009KG-10)

作者简介:牛永洁(1977-),男,河南许昌人,硕士,讲师,CCF 会员,研究方向为软件工程、数据挖掘。

包围在联结点执行的前、后执行的环绕通知 (Around Advise); 在某个联结点执行之前执行的前置通知 (Before Advise); 在某个联结点执行之后执行的后置通知 (After returning Advise), 它能获得操作的返回值; 能够捕获联结点异常的异常通知 (throws advice) 和特殊的拦截通知引入通知 (introduction advice)。

(3) 切入点 (Point Cut): 用来定义某一个通知该何时执行的一组联结点。

(4) 方面 (Aspect): 通知和切入点的组合, 定义了一段程序中应该包括的逻辑, 以及何时应该执行该逻辑。

(5) 织入 (Weaving): 将方面真正加入程序代码的过程。

(6) 引入 (Introduction): 通过引入, 可以在一个对象中加入新的方法或属性, 以改变它的结构。这样, 即使该对象的类没有实现某一个接口, 也可以修改它, 使之实现该接口。

目前, AOP 实现的方法有多, 在 Java 平台下比较著名的有: AspectJ、Spring 等, 在 .NET 平台下比较著名的有 AspectC#、Spring.NET 等。各种 AOP 技术的一个主要区别, 在于其将程序和方面结合方法的不同, 实现的方法主要分为静态织入与动态织入^[7]。

2 基于 AOP 的动态日志的实现

2.1 实现原理

动态日志将应用于仪器设备管理系统中, 在仪器设备管理系统中存在很多关键点, 对关键操作的调用就是联结点, 这些关键操作都被封装在系统的业务逻辑类中, 仪器设备管理系统中的关键操作如图 1 所示。

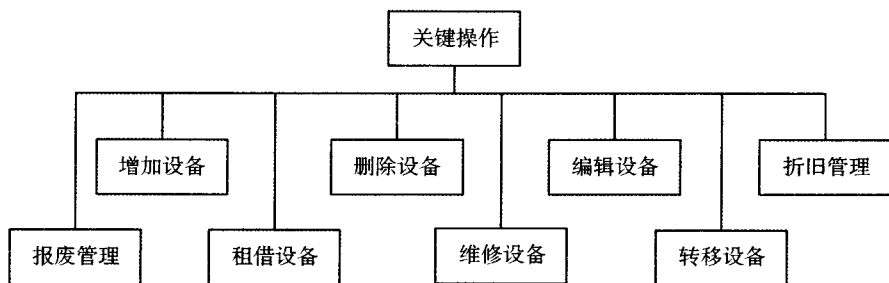


图 1 关键操作图

在关键操作发生时, 日志将记录操作的调用者、调用时间、用户地址、操作类型、操作对象以及返回值等信息。日志的记录操作就是 AOP 中通知发生的动作, 将系统中需要被日志记录的操作作为一个接口来实现, 凡定义在该接口中的操作都将被拦截并被日志记录, 日志系统将记录发生的动作分别放在前置与后置通知中。

为了实现横切关注点与核心关注点之间的分离, 在设备管理系统中调用关键操作时, 首先得到业务逻辑类的代理工厂 (ProxyFactory), 将前置通知和后置通知动态地添加到代理工厂中, 以后通过代理工厂调用的关键操作将自动被拦截并记录。

但是, 被拦截的操作仅仅局限于接口中定义的操作, 而这些操作往往被硬编码实现, 不能动态地增加或删除接口中定义的关键操作列表, 不利于系统的扩展。为了能够动态地控制接口中关键操作的定义列表^[8], 使用反射与动态编译机制实现向一个空接口中添加依据配置信息的关键操作。前提条件是配置文件中的操作必须是业务逻辑类中定义的关键操作的子集。

2.2 代码设计与实现

动态日志系统在 .NET 环境下实现, AOP 的实现采用 Spring.NET^[9], 日志的操作采用 Log4Net^[10], 生成动态接口使用 Reflection 与 Compiler 类^[11,12]。

配置文件中对动态生成接口的配置信息:

```

<interface >
  <method name=" AddDevice" >
    <return>bool</return>
    <paramType>Device</paramType>
    <paramType>User</paramType>
  </method>
  <! --其他方法列表-->
</interface>
  
```

使用类 DynamicClass 向空接口 ILogService 中添加方法列表, 然后在仪器设备管理系统中的业务逻辑类 BusinessDevice 实现该接口, 为了拦截 ILogService 接口中定义的操作, 实现了前置通知 BeforeAdvise、后置通知 AfterReturningAdvise, 通过代理工厂将前置与后置通知绑定在 BusinessDevice 上, 为了系统使用的方便,

对绑定操作进行了进一步的封装, 封装为类 BusiDevWithLog。在系统中使用 BusiDevWithLog 进行业务操作时, 被动态添加进 ILogService 的操作会被自动拦截, 并执行前置与后置通知中定义的日志操作。相关的类图如图 2 所示。

示。

表 1 列出了一些方法及接口的相关说明。

2.3 动态日志系统的运行流程

系统开始运行时, 首先从配置文件中读取关于关键操作列表的信息, 然后根据配置信息动态生成接口 ILogService。再生成 BusinessDevice 类的 ProxyFactory 工厂, 其中 ILogService 中关键操作的列表应是 BusinessDevice 类中关键操作的子集。将通知 AfterReturn-

ingAdvise、BeforeAdvise 绑定到 ProxyFactory 工厂,使用 ProxyFactory 工厂的 GetProxy 方法得到 ILogService 类型的对象 service。在仪器设备管理系统中,需要拦截的关键操作必须通过 service 进行,这些关键操作都会被通知 AfterReturningAdvise、BeforeAdvise 拦截,进而执行通知中的日志记录操作。

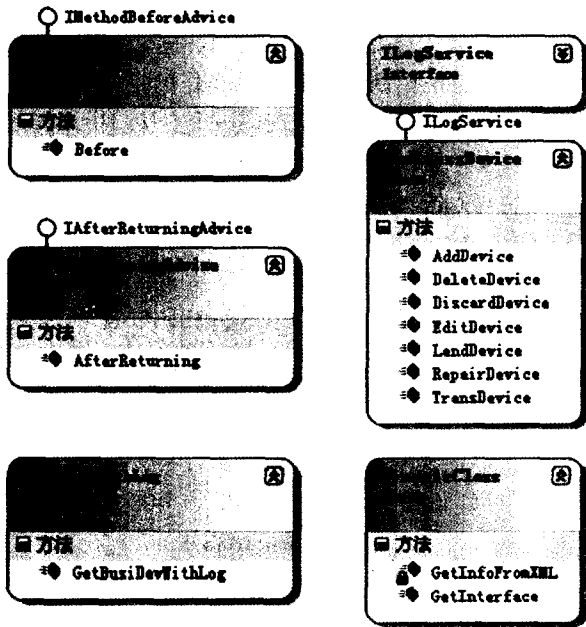


图 2 动态日志系统实现的相关类
系统运行流程如图 3 所示。

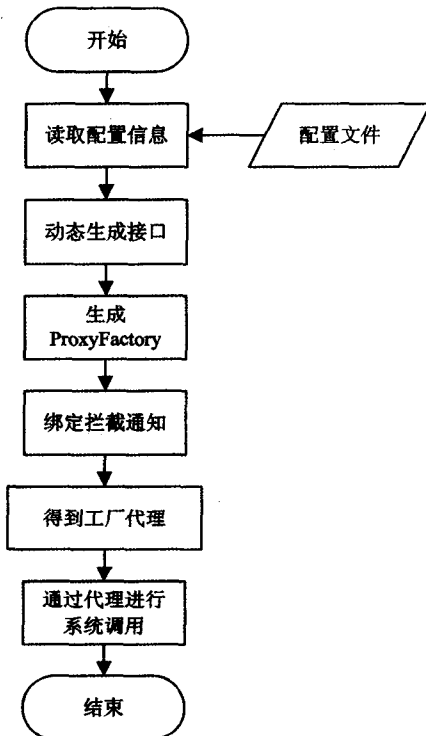


图 3 系统运行流程图

其中 BusiDevWithLog 类实现了通知与代理工厂的

绑定以及返回系统中调用的代理,源代码如图 4 所示。

表 1 一些方法及接口的说明

名称	所属类	功能说明
Before	BeforeAdvise	前置通知方法,在关键操作调用之前执行,进行信息记录
AfterReturning	AfterReturningAdvise	后置通知方法,在关键操作调用之后执行,进行信息记录
ILogService	空的接口	需要根据配置信息动态向其中添加关键操作列表
GetInfoFromXML	DynamicClass	从配置文件中得到关键操作的配置信息
GetInterface	DynamicClass	根据 GetInfoFromXML 返回的信息,添加关键操作,返回 ILogService 接口
GetBusiDevWithLog	BusiDevWithLog	对原有业务类进行二次封装,返回的业务类已具有日志记录功能
AddDevice	BusinessDevice	原业务逻辑类中的添加设备操作

```

class BusiDevWithLog
{
    public static ILogService GetBusiDevWithLog()
    {
        ProxyFactory factory = new ProxyFactory( new BusinessDevice
        ());
        factory.AddAdvice(new AfterReturningAdvise());
        factory.AddAdvice(new BeforeAdvise());
        ILogService service = (ILogService)factory.GetProxy();
        return service;
    }
}
  
```

图 4 BusiDevWithLog 类源代码

3 结束语

为了将分布在软件系统中的横切关注点与核心关注点相分离,更好地实现系统的可维护性与可扩展性,使用 Spring.NET 的 AOP 功能,将 Log4Net 实现的日志管理动态织入仪器设备关系系统的核心业务逻辑中,通过使用反射与动态编译的方法,根据配置文件中的信息能够动态定义系统中被拦截的关键操作。经过在仪器设备管理系统的使用,达到了良好的效果。

参考文献:

[1] 葛峰,张云华,赵国平.基于 AOP 技术的 Web 系统日志管理[J].计算机系统应用,2007,10:117-118.
 [2] 尹涛,李翔,林祥,等.基于 AOP 的角色访问控制模型设计与实现[J].计算机技术与发展,2008,18(10):136-139.
 [3] Kiczales G, Lamping J, Mendhekar A, et al. Aspect-oriented Programming[C]//Proceeding of 11th European Conference of Object-Oriented Programming, LNCS 1241. [s. l.]: [s. n.], 1997:220-242.

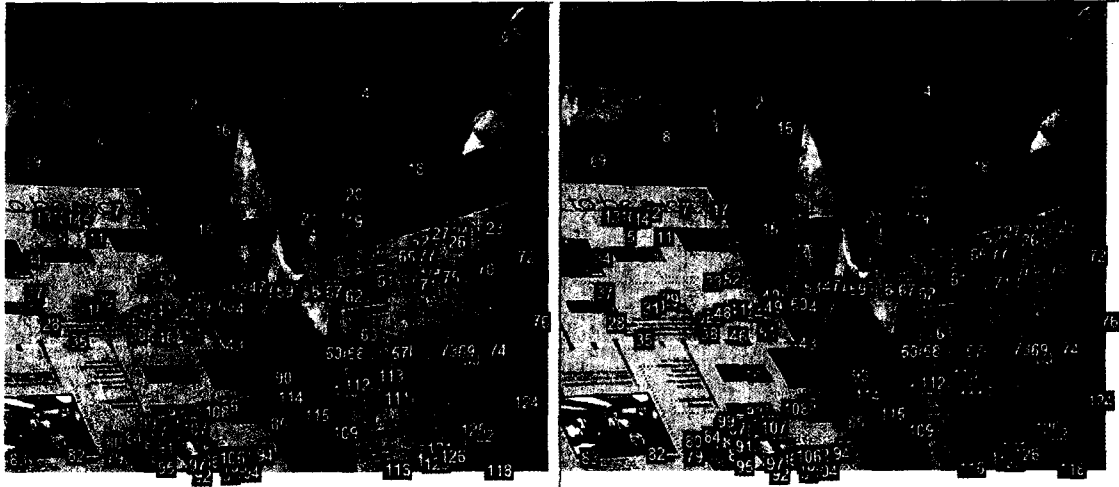


图 4 匹配点距离相关性约束进行引导匹配结果图

5 结束语

文中给出了一个新的立体匹配约束条件:匹配点距离相关性约束,据此提出了一种基于 LMedS 与匹配点距离相关性约束的立体匹配算法,在使用 LMedS 获得部分正确匹配点集的基础上,再利用匹配点距离相关性约束进行引导匹配。实验表明,文中提出的立体匹配算法能够很好地实现特征点的正确匹配。

参考文献:

- [1] Lhuillier M, Quan L. Robust dense matching using local and global geometric constraints[C]//In: Proceedings of the 16th International Conference on Pattern Recognition. Barcelona, Spain:[s. n.], 2000:968-972.
- [2] 谭磊,张桦,薛彦斌.一种基于特征点的图像匹配算法[J].天津理工大学学报,2006,22(6):65-69.
- [3] 戚世贵,戚素娟.一种基于图像特征点的图像匹配算法[J].理论与方法,2008,27(1):3-4.
- [4] 张少辉,沈晓蓉,范耀祖.一种基于特点特征点提取及匹配的方法[J].北京航空航天大学学报,2008,34(5):516-519.

- [5] 郭龙源,夏永泉,杨静宇.基于梯度的快速区域匹配方法[J].计算机科学,2007,34(4):239-240.
- [6] 云捷,顾磊,吴慧石.基于 Zernike 矩的区域匹配方法[J].中国图象图形学报,2008,13(8):1517-1524.
- [7] 姚益平,张颖星,陈欣,等.基于索引排序的快速动态区域匹配算法[J].计算机学报,2009,32(7):1375-1381.
- [8] 毛雁明,兰美辉,王运琼,等.一种改进的基于 Harris 的角点检测方法[J].计算机技术与发展,2009,19(5):130-133.
- [9] 王燕婷,张玮.图像匹配技术在土地利用动态监测中的应用方法研究[J].中国农学通报,2007(6):607-610.
- [10] Chojnacki W, Brooks M J, Galwey D, et al. A new approach to constrained parameter estimation applicable to some computer vision problems[C]//Statistical Methods in Video Processing Workshop Held in Conjunction with ECCV'02. Copenhagen, Denmark:[s. n.], 2002:43-48.
- [11] 邹定海.三维视觉检测研究及其 ADC 应用[D].天津:天津大学,1992.
- [12] Armangué X, Salvi J. Overall view regarding fundamental matrix estimation[J]. Image and Vision Computing, 2003, 21: 205-220.

(上接第 104 页)

- [4] 曹东刚,梅宏.面向 Aspect 的程序设计——一种新的编程泛型[J].计算机科学,2003,30(9):5-10.
- [5] 周改霞,高尚,杨镇江.基于 AOP 的 Web 系统事务管理的应用研究[J].计算机技术与发展,2010,20(4):246-249.
- [6] 陈月娟,李慧,刘光远.基于 AOP 的信息管理系统的研究与实现[J].计算机应用与软件,2010,27(2):130-133.
- [7] 胡冰,程家兴,郭建业.一种 .NET 平台下 AOP 框架的实现方法[J].计算机与现代化,2009,11:85-89.
- [8] 陈敏智.一种基于 AOP 的 BPEL 过程动态发现和绑定服务的方法[J].浙江工业大学学报,2009,37(5):511-514.

- [9] Colyer A, Clement A, Harley G, et al. Eclipse AspectJ: Aspect-Oriented Programming with AspectJ and the Eclipse AspectJ Development Tools[M]. [s. l.]: Addison Wesley Professional, 2004.
- [10] 蔡宇,赵正文.基于 NHibernate 和 Log4Net 实现用户登录合法性验证[J].信息技术,2008,11:32-35.
- [11] Hoffman K, Kruger L. Microsoft Visual C#. NET 2003 Unleashed/C#. NET 技术内幕[M].董梁,高猛,译.北京:清华大学出版社,2006:544-556.
- [12] Robison S, Nagel C. C#高级编程[M].北京:清华大学出版社,2004.