

利用反射技术实现一种新型软件系统维护模型

詹泽东¹, 谢爽², 藏文亚¹, 陆华¹, 李正民¹

(1. 成都理工大学, 四川 成都 610059;

2. 中国石油化工集团 西北分公司, 新疆 阿克苏 842017)

摘要: 在深入研究.NET 反射技术基础上, 结合.NET 框架结构的特性, 从资源整合的角度, 探索性地提出了一种新型的软件系统维护模式, 保证了复杂的大型系统维护的高效性和经济性, 具有很大的实际意义。

关键词: .NET 框架; 反射技术; 资源整合; 系统维护

中图分类号: TP311.53

文献标识码: A

文章编号: 1674-7720(2012)01-0070-03

Using reflection technology to implement a new kind of software system maintenance

Zhan Zedong¹, Xie Shuang², Zang Wenya¹, Lu Hua¹, Li Zhengmin¹

(1. Chengdu University of Technology, Chengdu 610059, China;

2. Northwest Branch of China Petroleum & Chemical Group, Aksu 842017, China)

Abstract: This paper based on the in-depth research of .NET reflection technique, combined the technology highlight of .NET framework, but from the perspective of resource integration and innovation to put forward a new software system maintenance mode, ensuring large, complex systems efficient system maintenance and economy, has great practical significance.

Key words: .NET framework; reflection technology; resource integration; system maintenance

随着计算机技术不断更新和发展, 大型复杂系统的开发层出不穷, 但系统的复杂性为后期系统的维护带来了极大不便。软件工程的提出和软件测试的规范化为后期维护提供了大量的文档资料, 从宏观上解决了后期系统维护工作任务。但是软件系统开发作为一项复杂的脑力活动, 不可避免地会出现一些问题, 尤其是复杂的大型系统。于是运行时出错的定位成为了一大难题, 往往需要消耗大量的人力、物力和财力, 甚至影响到企业的正常运作, 造成巨大损失。本文对于大型系统运行时出错的定位问题, 针对.NET 框架平台(目前系统开发的最大平台之一, 也是大型系统的构架平台)提供的特性技术和反射技术, 从系统的安全性和经济可行性出发, 从资源整合的角度出发, 研究并实现了一种新型的软件维护模型, 从而弥补了软件工程思想对于系统维护在微观上的定位的不足。

1 相关技术简介

新型软件系统的维护模型的技术实现主要涉及

.NET Framework 提供的特性技术、反射技术以及资源整合思想中邮件服务。

1.1 特性技术

公共语言运行时允许添加类似关键字的描述声明, 叫做特性^[1](Attributes), 它对程序中的元素进行标注, 如类型、字段、方法和属性等。Attributes 和 Microsoft .NET Framework 文件的元数据保存在一起, 可以用来向运行时描述代码, 或者在程序运行的时候影响应用程序的行为; 特性分为定制特性(.NET 平台已经封装实现)和自定义特性, 新型软件系统维护模型使用自定义特性, 用于系统运行时异常定位。

1.2 反射技术

在.NET 平台中, 软件是以程序集的方式存在的, 它包含模块, 而模块包含类, 类又包含成员。

.NET 中的程序集是自描述的, 反射则提供了封装程序集、模块和类的对象。使用反射隐藏一些不想公开的方法, 并在软件设计中进行了类设计及部分的核心编码^[2],

技术与方法 Technique and Method

动态地创建类的实例,将类绑定到现有对象,或从现有对象中获取类,然后调用类的方法或访问其字段和属性。反射通常具有以下用途^[3]:

(1)使用 Assembly 定义和加载程序集,加载在程序集清单中列出的模块,并从此程序集中找类型并创建该类型的实例。

(2)使用 Module 了解模块的程序集以及模块中的类等信息。

(3)使用 ConstructorInfo 了解构造函数的名称、参数、访问修饰符(如 public 或 private)和实现(如 abstract 或 virtual)等信息。

(4)使用 MethodInfo 了解方法的名称、返回类型、参数、访问修饰符(如 public 或 private)和实现(如 abstract 或 virtual)等信息。

(5)一个 FieldInfo 可了解字段的名称、访问修饰符(如 public 或 private)和实现(如 static)等信息,并获取或设置字段值。

(6)使用 EventInfo 了解事件的名称、事件处理程序数据类型、自定义属性、生命类型和反射类型等信息,并获取或设置属性值。

(7)使用 PropertyInfo 了解属性的名称、数据类型、声明类型、反射类型和只读或可写状态等信息,并获取或设置属性值。

(8)使用 ParameterInfo 了解参数的名称、数据类型、是输入参数还是输出参数。以及参数在方法签名中的位置等信息。

1.3 资源整合

与企业优化资源配置不同,随着 Internet 的推广使用,网络服务资源日益增多,资源共享和合理利用免费资源也成为企业的一个重要的战略目标。大型系统的维护借助于免费资源的使用可以大大降低成本,而邮件服务是一个重要的免费资源。本文在研究新型软件系统维护模型时,将系统软件运行时出错(或者异常)信息以邮件的形式将追踪到的系统运行出错定位信息发送到指定维护人员的邮箱,维护人员借助此信息可以及时、高效地完成维护工作,提高了维护的灵活性和及时性。

2 新型软件系统维护模型的提出

由于大型系统的复杂性或多或少会导致系统运行时出错或者发生异常,利用 .NET Framework 框架提供的特性与反射技术相互结合,可以准确地定位异常发生的具体位置,同时反射出出错信息处的代码相关信息,保证了系统的维护的快捷性和方便性,如图 1 所示。

3 新型软件系统维护模型的实现

3.1 特性类的实现

在系统运行时出错定位,必须使用自定义特性,所有自定义特性都继承于 System.Attribute 这个类^[1],为了在系统运行发生异常时获取异常信息(这里只获取系统

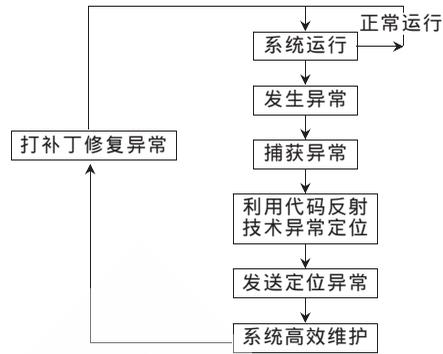


图1 新型软件系统维护模型

代码所在行的定位信息,其他信息可以根据实际需要获取),并将异常信息通过邮件的形式发送出去,需要为自定义特性定义如下数据成员,如表 1 所示。

表 1 自定义特性字段描述

字段名	含义	语义
sendEmail	申请的免费邮箱地址	作为发送方如系统
acceptEmail	开发或者维护人员的邮箱地址	接收异常信息
serverAdress	免费邮件服务器地址	邮件服务器
userName	邮件的用户名	邮件的用户名
passWord	邮件的密码	邮件的密码
modelName	代码所在模块名	作为异常定位

其中在表 1 中的邮件用户名和邮件密码指的是发送邮件方的用户名和密码,自定义特性类的成员函数主要包括构造函数,各个数据成员的属性方法。具体使用时在每个模块的定义部分使用自定义特性类构造对象,给相关的模块代码添加附加信息进行关联,本文主要是代码所在的模块名。使用 BS 结构的母版页作为实例,具体形式如下:

```
[myClassAttribute (string sendEmail, string acceptEmail,
string serverAdress, string userName, string passWord, string
modelName)] //自定义特性实例化
```

```
public partial class MasterPage: System.Web.UI. MasterPage
//母版页类
```

其中 myClassAttribute 为自定义特性的类名,实际实例化时使用实际参数,这样就建立起了一个信息关联。

3.2 反射技术的实现

反射技术的实现在本实例中主要分为三个步骤(可以根据实际情况做相关的改变):

(1)使用 C# 的 typeof 运算符、所有类从 System.Object 类继承的 GetType 方法或者 System.Type 类的静态 GetType 方法获取 System.Type 类的引用。

(2)创建反射实例,根据实际需要可以选择 ConstructorInfo、EventInfo、FieldInfo、InterfaceInfo、MemberInfo、MethodInfo、PropertyInfo,本文实例中选择构造 MemberInfo 实例。

(3)使用 Attribute 类的 GetCustomAttribute 静态方法

技术与方法 Technique and Method

构造自定义特性对象,其中所包含的本模块的关联的信息就可以获取出来。

3.3 捕获异常

C# 专门提供了处理异常机制,可以使用.NET 提供的预定义异常类,也可以使用自定义异常类,本文主要是使用 try{……}catch{……}finally{……}结构捕获异常。

3.4 资源整合的实现

为了降低开发和维护成本,本文中使用了免费邮件服务器接受系统运行时出错或者异常信息,将该信息以邮件^[4]的形式转发给系统开发者或维护人员,这一过程可以借助于.NET 提供的命名空间 web.mail 来实现。它提供了 SmtpMail、MailMessage 和 MailAttachment 三个类,分别用于提供属性和方法来创建一个邮件消息对象、提供属性和方法来创建一个邮件附件对象、提供属性和方法通过使用 windows2000 CDOSYS 的消息组件的联合数据对象来发送邮件消息。同时使用免费发送邮件服务,在设置好各个类的属性后还需加上这三句(省略号表示根据实际参数填写参数):

```
objMailMessage.Fields.Add(……); //基本权限
objMailMessage.Fields.Add(……); //用户名
objMailMessage.Fields.Add(……); //密码
```

4 新型软件系统维护模型的测试

本文在采用 BS 结构的模板页中故意设置一个异常点,依次按上述 3.1~3.4 的步骤实现,将系统进行运行得到运行结果,如图 2 所示。

利用.NET 平台下的特性技术和反射技术相结合



图2 新型软件系统维护模型测试图

进行异常信息的定位,为系统的维护探索性提出了新型模型,可以有效地提高维护高效性、灵活性,同时融合资源整合的思想,降低了开发和维护的成本,为系统的维护和开发起了指导的作用。随着远程控制的普及,该模型与远程控制相结合将取得更好的效果。

参考文献

- [1] NAGEL C, EVJEN B, GLYNN J, 等.C# 高级编程(第四版)[M].北京:清华大学出版社,2006.
- [2] 李天平,亮剑.NET: .NET 深入体验与实战精要 [M].北京:电子工业出版社,2009.
- [3] 叶树华,高志红.网络编程实用教材[M].北京:人民邮电出版社,2006.
- [4] 梅元昭.基于反射的 WEB SERVICE 的设计与实现[J].软件导刊,2010,9(1):3-5.

(收稿日期:2011-06-08)

作者简介:

詹泽东,男,1986年生,硕士,主要研究方向:地震勘探、测井解释及软件工程。