

# 基于 SOA 反射工厂的软件体系架构

祝连鹏<sup>1</sup>, 王 虎<sup>2</sup>, 赵学臣<sup>3</sup>

(1. 山东师范大学 信息科学与工程学院, 山东 济南 250014;

2. 普联软件(中国)有限公司, 山东 济南 250101;

3. 山东省分布式计算机软件新技术重点实验室, 山东 济南 250014)

**摘 要:**在现代企业级应用开发领域中, SOA 以其能提供粗粒度服务接口、组件服务具有“松散耦合”性、编码灵活、更高的可用性等优点, 越来越多地被应用到企业级开发中, 反射工厂模式也以其能提供和用户的高交互性而被广泛应用。为了解决软件平台中间件之间的实时交互问题, 在分析现有体系架构基础上, 将中间件技术、反射模式中的工厂模式以及 SOA 服务性架构进行整合, 并且应用了改进的反射协议栈技术, 提出基于 SOA 和反射工厂的组合软件体系架构, 并通过 Panschema 二次开发平台系统验证, 软件开发实验证明, 本架构在交互实时性上优于其他的架构体系。文中的运行环境为 .Net 环境。

**关键词:**SOA; 反射工厂; 改进反射协议栈

**中图分类号:**TP311.5

**文献标识码:**A

**文章编号:**1673-629X(2011)07-0125-04

## Reflection Factory Software Architecture Based on SOA

ZHU Lian-peng<sup>1</sup>, WANG Hu<sup>2</sup>, ZHAO Xue-chen<sup>3</sup>

(1. School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China;

2. Pansoft Company, Jinan 250101, China;

3. Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250014, China)

**Abstract:** In the modern enterprise applications development field, SOA can be applied to the enterprise development increasingly with advantages of coarse granularity service interfaces, component services have "loose coupling", coding flexible, higher usability etc, the reflection factory model is widely used can provide the user's high interactivity. In order to solve real-time interactive question between software platform middleware, in the analysis of existing framework based on middleware technology, integrate middleware technology, reflectance mode of factory model and SOA service framework, and using reflection protocol stack improved technology, propose the combination of software architecture based on SOA and reflective factory, and through Panschema second development platform system verification, software development experiments show that this framework in interactive real-time is better than other framework system. The operating environment is the .Net environment.

**Key words:** SOA; reflection factory; improved reflection protocol stack

## 0 引 言

SOA 服务型架构(service oriented architecture)不是一种产品或者体系结构, 而是一种一体化应用软件的概念。SOA 早期的概念是要求所有的应用软件的设计界面都要具备一定意义上的商业用途, 例如可以用来处理购货订单报表或者进行库存表单的实物清算。只要确定开始了服务就能根据服务自动完成整套的商业流程。

SOA 要成立必须满足 6 个前提<sup>[1]</sup>, 该前提是假设状态: 首先系统必须是松散耦合的; 第二界面是非物质的交换; 第三程序要具有 RPC(remote procedure call, 远程功能呼叫)的功能; 第四界面是基于消息的界面; 第五消息要符合 XML 数据格式; 第六界面要支持两种数据传输, 该传输是同步的或者非同步的。

当其中的一个系统进行工作时, 不应该对另一个系统运行产生较大程度影响, 与此同时服务的实施是在幕后进行的, 这时系统就被判断为是松散耦合的。当然 SOA 服务也有可能是同步进行的, 即发送完毕请求然后就转为等待即时消息回应<sup>[2]</sup>。

以上这些比较简单的 SOA 模式是很难在系统中实现的。其中要依据为系统提供的服务来确定符合要

收稿日期: 2010-12-17; 修回日期: 2011-03-20

基金项目: 山东省科技型中小企业技术创新基金(06C26213701395)

作者简介: 祝连鹏(1986-), 男, 山东临沂人, 硕士研究生, 主要研究方向为 SOA、MDA、软件设计模式。

求的程度和类型。服务要是精细型的,也就是要执行像改变其中一个数据要素等;服务也可以是粗放型的,此过程中要求可以处理即时的、重要和复杂的商务过程。

工厂模式<sup>[3]</sup>是最常用的模式了,比如 Jive 论坛,就大量使用了工厂模式,工厂模式在 C# 程序系统可以说是随处可见。因为工厂模式就相当于创建实例对象的 new,经常要根据类 Class 生成实例对象,这便于给系统带来更大的可扩展性和尽量少的修改量。

为了获得更高效率的优化系统,将 SOA 模型和反射工厂模式的优势结合起来是一种可行可操作的新的体系架构。

## 1 相关研究

### 1.1 流行的 SOA 架构模型

SOA 是一种架构模型,它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。服务层是 SOA 的基础,可以直接被应用调用,从而有效控制系统中与软件代理交互的人为依赖性。SOA 是一种在计算环境中设计、开发、部署和管理离散逻辑单元(服务)的模型<sup>[4]</sup>,这一定义阐明了 SOA 的职能范围。

### 1.2 工厂模式中的反射模式

工厂模式是一种重要的设计模式,它能为创建对象提供过渡接口<sup>[5]</sup>,以便将创建对象的具体过程屏蔽隔离起来,达到提高灵活性的目的。文中用到的是工厂模式中的反射工厂模式,它通过获取用户的输入,实例化一个对应的表和实现类。

### 1.3 中间件技术

一般的软件设计模式都是遵循黑箱(black-box)抽象的原则<sup>[6]</sup>,即要实现的模式应该只外露其功能而隐藏其中的实现过程。虽然黑箱抽象原则有上述优点,但是如果一个模块仅仅是暴露了其功能性界面,往往不能符合不同客户群的需求,当前的流行的中间件<sup>[7]</sup>,像 CORBA,DCOM 还有 Java RMI,它们都是使用了流行的黑箱原则,但是很缺乏灵活性,然而中间件要处理的是比较复杂的分布式的不同应用程序难题,所以不得不时时面对来自变化的环境和用户需求的改变的挑战。

基于以上三种模型思想,文中提出了一种结合 SOA 和反射工厂的结合模式,为了实现“外部改动实

时更新内部,内部改动实时更新外部组件”的思想,使用了中间件技术,所有的需要改动和动态传递的参数都放在 config XML 配置文件中。

## 2 基于 SOA 的反射工厂模式思想

### 2.1 索引结构

为了方便客户端与服务器以及中间件之间的通讯,文中引入了索引机制,即把需要传递的参数存放在 config 配置文件中,配置文件中应该包含本构件的所有参数、子构件的 config 配置文件的名称和对应的参数。此外,为每个 config 配置文件构造一个类,用来传递或者接收参数。

### 2.2 组件集成规范

#### 2.2.1 模式结构图描述

如图 1 所示,在控制端(即前端)节点有一个 config 配置文件,里面包含所有子节点的参数信息。每个子节点中也包含一个子配置文件,里面包含的是自己的参数信息和父节点(即前端节点)的参数信息。

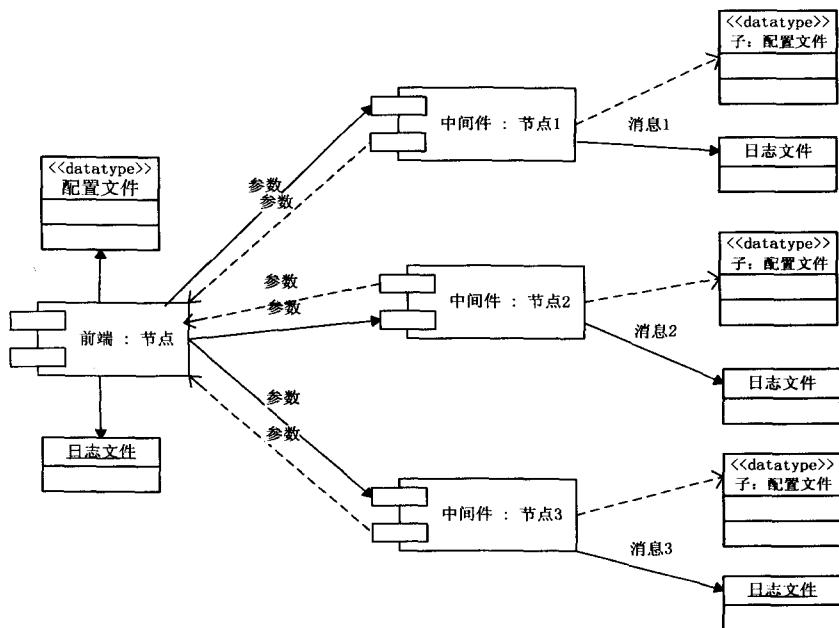


图 1 基于 SOA 的反射工厂模式结构图

当前端节点需要更新一个参数信息时,如果需要更新的是子节点 1 的参数,查找 config 文件,找到子节点 1 的地址,把参数传过去,然后更新状态为已发送;子节点 1 收到传来的参数以后,更新本地参数,然后发送一个回执给前端节点;前端节点收到回执以后,更新状态为“已更新”。

当子节点根据实际情况,需要更新一个参数时,把需要更新的参数传到前端节点,更新前端节点的 config 文件中参数的值。

对参数的每一步操作都会记录在日志文件中,提高了安全性,便于发生异常时回滚。

通过这种模式,可以实现前端节点和中间件的参数实时更新,比较以往的更新参数就要重新写中间件的模式,在代码复用和实时控制上更具有优势。

2.2.2 改进的反射协议栈

本系统在反射模式中应用了一种改进的反射协议栈机制。

如图2所示,在 RECOM 系统中<sup>[8]</sup>,这是一个简单的客户桩对象,该对象用来把特定要调用的类型转换为比较通用的模式,同时向最上层的决策者发送来自调用的请求。对于其他常用的系统,RECOM 的优势在于它的客户桩结构是比较简单的,因为它并不是用本地的语言来表示同字节流(用于网络传输)之间的转换。鉴于这种比较简单的原因,客户桩类一般是在程序运行中自动生成(这是对调用界面,然后利用 Java 的结构反射功能进行分析后完成的<sup>[9]</sup>),生成以后与程序中的动态链接进行转换。

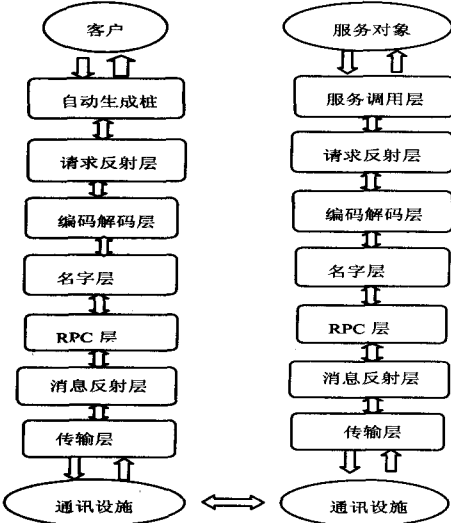


图2 RECOM 的反射协议栈

文中基于 .Net 环境描述,其中用到的改进反射协议栈如图3所示,将 RECOM 中的四个中间层合并,用

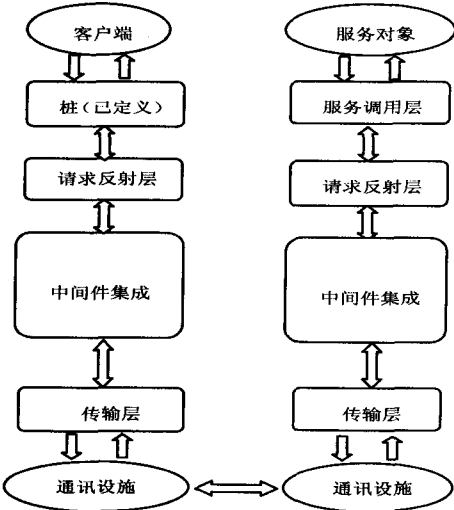


图3 改进的 .Net 协议栈

一个中间件来实现,这样既减少了中间流程,又可以实现代码的复用。

2.3 模式表达式描述

在模式的构造和参数的传递过程中,用一个数学模型来解释上述模式。

$$F \sim (M, S, Fn, N) \quad n = 1, 2, 3, \dots \tag{1}$$

公式(1)中,  $F$  表示中间件配置文件;  $M$  代表更新方向,是单项或者双向;  $S$  是关联映射规则,表明父节点和子节点之间的对应关系;  $Fn$  表示下一级子节点,  $N$  代表依赖于本节点的日志文件。

对于其中的  $S$ ,要根据实际情况来设置。

2.4 服务质量 (QoS) 管理

由于客户端和中间件以及最终的服务器不是放在同一个地点,为了保证参数等信息在网络中的顺利传输,基于 SOA<sup>[10]</sup>的反射工厂模式符合 QoS 制定的管理规范。

QoS 管理主要的内容有两点<sup>[11]</sup>:

- (1)在期望服务质量属性和服务质量(由环境设定)的基础上来进行各种不同组件和资源的初始化配置;
- (2)根据在试验中获取的服务质量和现实环境中提供的实际服务质量来进行判断并且来调整和检测各种条件下的组件图。

文中注重的是服务质量管理<sup>[12]</sup>的动态方面。因此在组件图结构中鉴于实际情况引入了管理组件。在操作中为了区分管理组件和被管理组件之间的关联又引入了事件通知机制。表1是管理组件的内容形式列表<sup>[13]</sup>。

表1 管理组件的内容形式列表

检测到的事件	
事件收集者	监视一些比较重要的组件的行为,自动生成服务质量的事件
检测者	收集来自各方面的服务质量事件,上报异常行为
控制决策	
策略选择者	根据上面检测着反馈的选择来进行调整策略
策略激发者	对组件图进行操作,决定选择的策略

服务质量管理策略如公式(2)所示:

$$S = (T, T0 \rightarrow, I) \tag{2}$$

- $S$  代表有限状态集;
  - $T0 \in S$  初始化定义为初始状态;
  - $\rightarrow$  表示各种状态之间的转换;
  - $I$  代表确定状态后的进行变量赋值的函数。
- 其中各种转换关系之间的公式如下:
- $\rightarrow = (l, g, a, r, l')$  (3)

公式(3)中, $l, l'$  为自动机的节点; $a$  为动作; $g$  为守护; $r$  为复位(重新赋值函数)。

3 系统验证

Panschema 平台是为中石油量身定制的二次开发平台,用基于 SOA 的反射工厂模式开发,运行于 .Net 环境下。该平台实现了满足客户所有需求的功能,能根据不同用户的不同选择,生成不同的报表,只要具有简单数据库知识的工作人员就可以随意制定出符合报表需求的各种功能和样式的报表系统。

通过前端获取的用户输入,填写卡片数据,然后编辑卡片窗口,如图 4 所示。完成后,反射自动生成数据库表和 XML 文件,在此过程应用了基于 SOA 的反射工厂模式。此步骤用来定义报表的显示内容和格式。

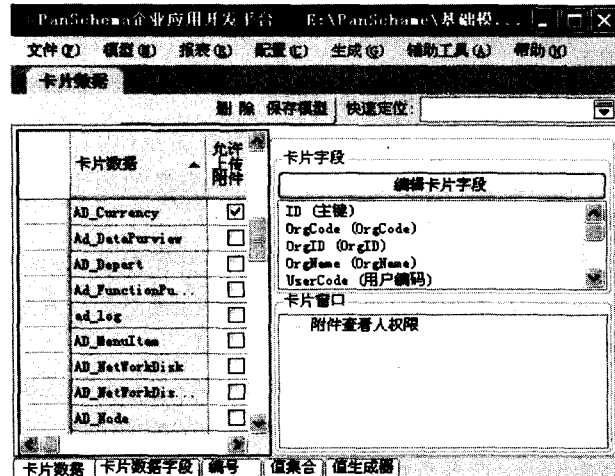


图 4 PanSchema 系统卡片数据界面

填写表单数据,然后生成表单窗口,如图 5 所示。此步骤用来定义多个报表的显示模式。

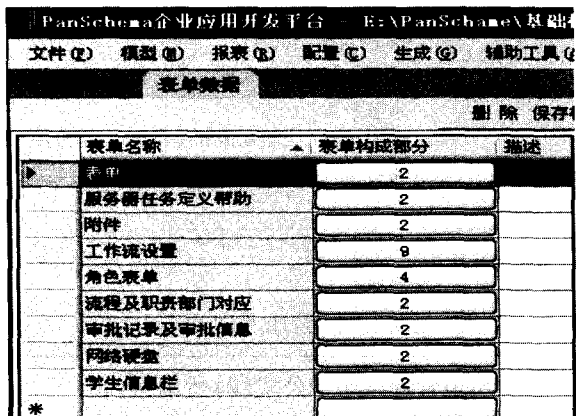


图 5 PanSchema 系统表单数据界面

填写菜单项以后,每一个菜单项选择一个处理功能。

最后点击生成,生成各种文件,并上传到服务器,就可以在内控系统中登录并看到自己选择设置的界面和功能。

4 结束语

文中在分析了现在流行的 SOA、反射工厂模式和中间件技术三种软件设计方法之后,提出了组合三种方法的基于 SOA 的反射工厂模式,并通过 Panschema 系统得到验证,基于 SOA 的反射工厂模式能实现面向服务和用户的高交互性,和其他模式相比,该组合模式在代码复用、交互性、耦合度、系统优化等方面都具有明显的优势。

参考文献:

- [1] Andrei M, Ang J. Patterns: Service-Oriented Architecture and Web Services [M]. [s. l.]: IBM Redbooks, 2004.
- [2] 毛新生. SOA 原理、方法和实践 [M]. 北京: 电子工业出版社, 2007.
- [3] 甄 镭. .Net 与设计模式 [M]. 北京: 电子工业出版社, 2005.
- [4] Erl T. SOA 概念、技术与设计 [M]. 王满红, 陈荣华, 译. 北京: 机械工业出版社, 2007.
- [5] Capra L, Emmerich W, Mascolo C. CARISMA: Contextaware reflective middleware system for mobile applications [J]. IEEE Trans on Software Engineering, 2003(10): 4-5.
- [6] 黄 昱, 王千祥, 梅 宏, 等. 基于软件体系结构的反射式中间件研究 [J]. 软件学报, 2003, 14(11): 1819-1826.
- [7] 李 冰, 许林英. 中间件平台的反射技术 [J]. 计算机应用研究, 2003(8): 47-50.
- [8] 杨思中, 刘锦得, 骆志刚. 反射中间件的研究 [J]. 小型微型计算机系统, 2002, 23(5): 618-622.
- [9] Maes P. Concepts and experiments in computational reflection [C]//OOPSLA'8. [s. l.]: [s. n.], 1987.
- [10] 谷玉奎, 曹宝香, 袁玉珠. 基于 SOA 的分布式构件库管理模式 [J]. 计算机技术与发展, 2008, 18(4): 101-103.
- [11] 林 闯, 单志广, 任丰原. 计算机网络的服务质量 (QoS) [M]. 北京: 清华大学出版社, 2004.
- [12] 李莹峰, 邓晓衡. DHT 网络中基于测量的 QoS 监控系统 [J]. 计算机技术与发展, 2009, 19(5): 101-103.
- [13] 姚世军, 陈楚湘, 党力明, 等. 动态选择 Web 服务的 QoS 本体设计 [J]. 计算机工程与设计, 2008(6): 1548-1550.

(上接第 124 页)

- [10] 蔡佳佳, 李名世, 郑 峰. 多核微机基于 OpenMP 的并行计算 [J]. 计算机技术与发展, 2007, 17(10): 87-91.
- [11] 康立山, 谢 云, 罗祖华. 非数值并行算法: (第一册) 模拟

退火算法 [M]. 北京: 科学出版社, 2000.

- [12] 宋 伟, 宋 玉. 基于 SMP 集群系统的并行编程模式研究与分析 [J]. 计算机技术与发展, 2007, 17(2): 164-167.