

策略可扩展的 ACD 模块的设计与实现

金 健, 王 力, 杜军朝, 张力勇

(西安电子科技大学 软件工程研究所, 陕西 西安 710071)

摘 要: ACD (自动呼叫分配) 作为呼叫中心的组成部分, 实现了坐席路由功能。传统的 ACD 均采用了固定路由策略的设计方式, 使得用户在使用时不能自主添加新的策略。为了增加整个呼叫中心系统的可用性, 基于动态链接技术以及程序模块化设计方法提出了一种有别于传统 ACD 的体系结构, 并以此为基础实现了一套 ACD 系统“KACD”。通过系统仿真试验, 证明了在该系统中, 用户能够在不丧失系统性能的前提下自主设计并添加坐席路由策略。

关键词: 自动呼叫分配; 呼叫中心; 路由策略; 用户设计; 用户添加

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)04-0104-03

Design and Implementation of Strategy Scalable ACD Module

JIN Jian, WANG Li, DU Jun-zhao, ZHANG Li-yong

(Software Engineering Institute, Xidian University, Xi'an 710071, China)

Abstract: As a component of call center system, ACD (automatic call distribution) implements the function of route. The traditional ACD all adopt the design method of fixed route strategy and made user can't add new route strategy while they are using. In order to increase availability of the call center system, based on the dynamic link technology and modular design programming, put forward an architecture which is different from the traditional ACD's, and takes this as foundation to implement an ACD system named "KACD". The simulation demonstrated that in this system, user can design and add route strategy by themselves without losing system performance.

Key words: ACD; call center; route strategy; user design; user add

0 引 言

呼叫中心又叫做客户服务中心^[1,2], 它是一种基于 CTI 技术、充分利用通信网和计算机网的多项功能集成, 并与企业连为一体的一个完整的综合信息服务系统, 利用现有的各种先进的通信手段, 有效地为客户提供高质量、高效率、全方位的服务。ACD^[3] 作为呼叫中心的一个组成部分, 承担着坐席话路分配的任务。目前的 ACD 又在此基础上添加了其他的话路功能, 例如: 来话排队等待, 客户身份认定等。但是, 在 ACD 的主要功能——话路分配上^[4,5], 目前的 ACD 仅仅能实现有限的几种坐席路由策略, 并且只能由呼叫中心的开发人员实现。在用户实际使用过程中, 并不能根据自己的需求来自主改变 ACD 的路由策略。这样就限制了呼叫中心系统在实际使用过程当中的有效性^[6,7]。考虑到用户对需求的了解程度高于呼叫中心的开发人员^[8], 应该设计一种开放式的 ACD, 使得用

户在使用过程当中, 能够根据实际的需求, 自主设计并实现自有的坐席路由策略, 并且添加到现有的 ACD 模块中。

1 用户可扩展模式的设计思想

基于模块化的设计思想^[9], 将 ACD 划分为多个模块, 各模块负责不同的任务, 模块间相互提供调用接口。其中用户模块可由用户按照约定的标准自行设计与实现, 并且在主系统发布以后的任意时刻添加到系统中来。

主系统向 ACD 的使用者提供了传统的调用接口, 但是, 在具体的路由策略上, 主系统并不实现, 而是交由用户扩展模块来完成。也就是说: 当 ACD 的使用者发出一个请求的时候, 主系统获得了该请求, 但并不处理该请求, 而是将其转发给用户扩展模块; 当用户扩展模块处理完并返回处理结果时, 主系统收到处理结果, 返回给 ACD 的使用者。

与传统 ACD 不同的是: 在这样一个过程当中, 文中设计的 ACD 系统 KACD 扮演的角色是一个消息的转发者, 而不是一个消息的处理者。

收稿日期: 2007-07-10

作者简介: 金 健 (1982-), 男, 陕西西安人, 硕士研究生, 研究方向为呼叫中心技术; 王 力, 教授, 硕士研究生导师, 研究方向为软件工程、虚拟现实与可视化技术。

主系统与用户扩展模块间的相互通信基于动态链接技术^[10]传递函数指针^[11]的方法。

2 ACD 体系结构设计

ACD 体系结构如图 1 所示。

(1) 向上接口提供层: 向调用者提供传统 ACD 的各种接口。

(2) 信息处理维护层: 维护 ACD 运转所需的数据; 接收上层的消息, 加工并向下层传递; 根据下层的执行结果, 产生事件返回给上层。

① 数据维护更新模块: 对每次的分配结果进行数据整理, 并更新相应的数据。

② 等待队列维护模块: 维护等待队列, 根据是否采用 VIP 用户模式进行排队。

③ 坐席信息维护模块: 维护、更新坐席的相关信息。

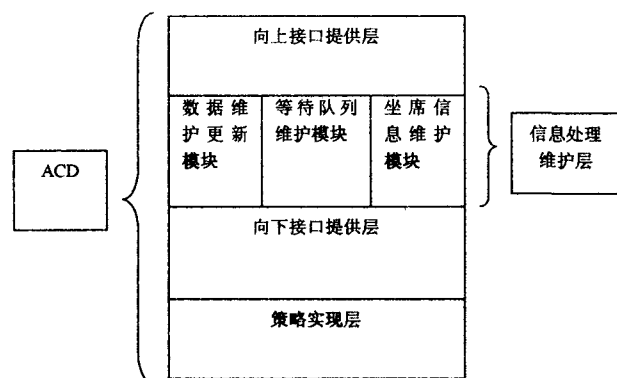


图 1 ACD 模块体系结构

(3) 向下接口提供层: 向策略实现层提供必要的调用接口。

(4) 策略实现层: 实现具体的 ACD 策略(该层由用户自行设计与实现)。

3 用户可扩展模式的实现方法

3.1 非用户模块的实现方法

3.1.1 结构设计

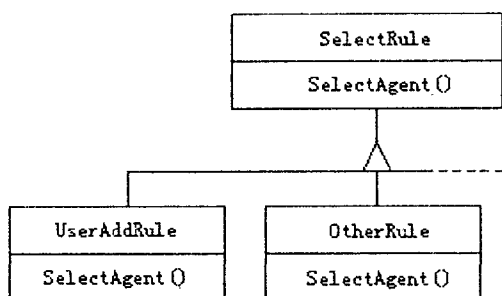


图 2 State 模式在文中的应用

使用 State 模式(如图 2 所示), 确保 ACD 在运行

时可以切换分配策略, 并且使 ACD 拥有的策略在数量上不受限制, 从而使得用户可以添加任意多的策略模块。

3.1.2 实现方法

UserAddRule 派生类, 与用户扩展策略模块交互。

① 定义了用于指向用户模块以及向下接口提供层提供的接口的函数指针。

② 定义了 UserAddRule 类。其构造函数通过①定义的函数指针获得了向下接口提供层提供的接口。

③ 定义了 InitClass 函数, 该函数加载并初始化相应的用户模块。

```
InitClass()
{
    /* 1. 加载用户模块 */
    /* 2. 获得用户模块提供的接口 */
    /* 3. 初始化用户模块 */
}
```

④ 定义了 SelectAgent 函数, 该函数响应 ACD 调用者分配坐席的请求, 然后直接调用用户扩展模块提供的相应接口。

3.2 用户模块的实现方法

InitModule: 保存上层模块提供的接口, 并根据不同的路由策略进行相关初始化工作。

SelectAgent: 实现用户自定义的坐席分配策略, 并在分配结束后, 调用向下接口提供层提供的接口, 从而使信息处理维护层能够进行必要的的数据更新, 最后返回分配结果, 使向上接口提供层通知 ACD 调用者。

```
SelectAgent ()
{
    /* 1. 用户自定义坐席分配策略的实现细节 */
    /* 2. 调用向下接口提供层提供的接口 */
    /* 3. 返回分配结果 */
}
```

4 用户扩展策略的使用流程

ACD 分配坐席流程如图 3 所示。

① ACD 调用者请求 ACD 分配一个坐席。

② ACD 查询是否存在请求使用的坐席路由策略 R。

③ 如果不存在, 则直接跳转到步骤⑥。

④ 调用提供该策略(R)的 UserAddRule 类对象 O 的方法 SelectAgent。

⑤ 对象 O 的方法 SelectAgent 调用用户模块提供的接口, 获得路由结果。

⑥ 向 ACD 调用者返回结果。

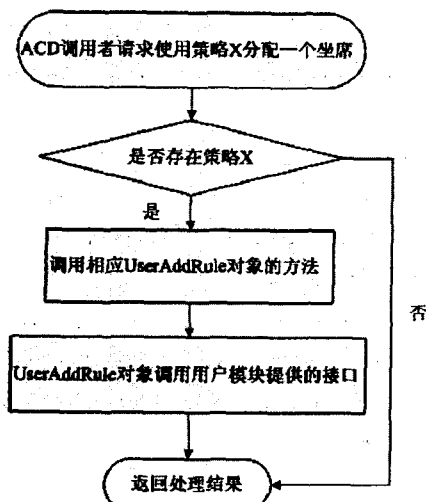


图 3 ACD 分配坐席流程

5 仿真数据对比

5.1 仿真方法

使用文中提出模式的 ACD 与没有使用这种模式的 ACD 在平均分配时长、系统内存消耗量上进行对比。使用多种分配策略,不同的坐席数量,不同的呼叫量分别进行仿真。

5.2 仿真结果的图表表示

图 4 为 10 坐席平均等待时间对比,图 5 为 10 坐席平均内存消耗量对比,图 6 为 20 坐席平均等待时间对比,图 7 为 20 坐席平均内存消耗量对比,图 8 为 50 坐席平均等待时间对比,图 9 为 50 坐席平均内存消耗量对比。

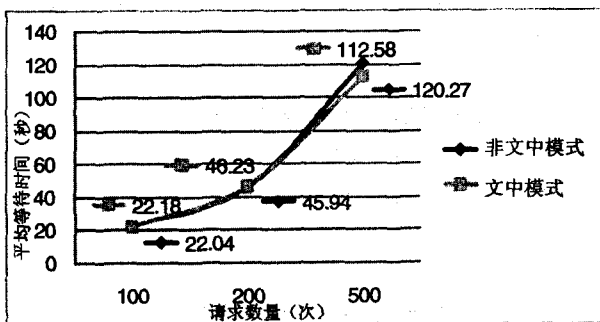


图 4 10 坐席平均等待时间对比

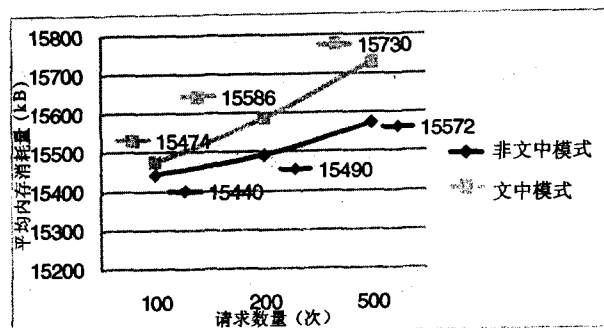


图 5 10 坐席平均内存消耗量对比

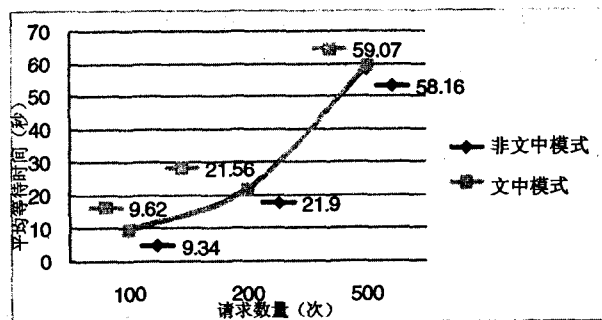


图 6 20 坐席平均等待时间对比

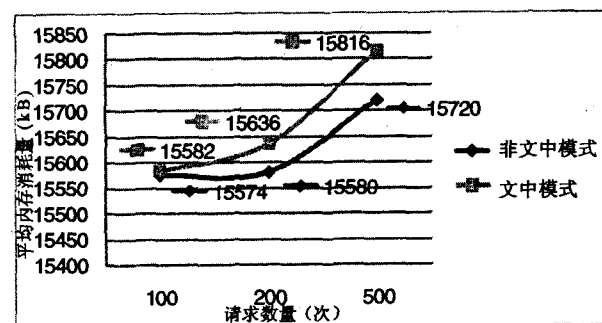


图 7 20 坐席平均内存消耗量对比

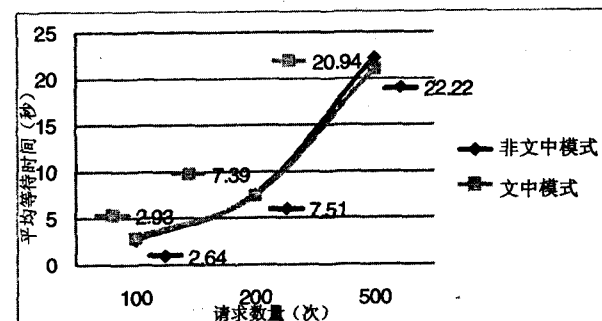


图 8 50 坐席平均等待时间对比

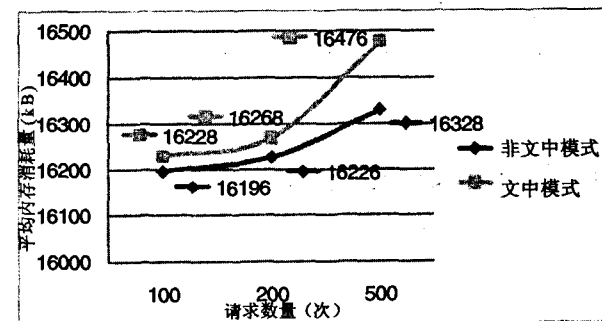


图 9 50 坐席平均内存消耗量对比

6 结语

通过仿真数据以及图表可以看出,使用文中提出模式的 ACD 在平均等待时间与系统内存消耗量上与传统模式的 ACD 只存在细微的差别,在实际的系统运行中可以忽略不计。而通过文中提出的模式,可以更

(下转第 110 页)

明性事务,针对服务类中有的方法涉及多次数据操作配置的同一事务中,以确保事务的一致性与完整性。业务逻辑层包括的业务逻辑具体有以下几个方面:

(a)模型操作服务:实现 UML 流程模型的各种操作并为上层提供访问接口;调用数据访问层的接口将模型持久化;为流程引擎提供模型查找服务。

(b)用户组织服务:通过对数据访问层接口 API 的操作,实现对组织机构、用户信息、角色信息以及用户的角色的维护操作,并为上层提供访问接口;为流程引擎提供用户组织查找服务。

(c)工作流引擎服务:通过对数据访问层接口、模型操作服务接口、用户组织服务接口 API 的调用,实现工作流引擎所提供的功能(即流程模型的解释与执行、工作项列表的管理、消息服务、过程的监控服务),并为上层提供调用接口。它是系统的核心部分,控制着流程实例的运转、对各服务的监控、各服务状态的管理。

2.3 Web 客户端

较其它客户端形式而言,浏览器 Web 客户端不需安装客户端软件,可以实现客户端零维护,从而降低了维护费用,并方便系统的升级。本系统中,流程建模、系统管理人员操作客户端、普通用户操作客户端都采用 Struts 框架实现。客户端具体的处理逻辑在 Struts 提供的 Action 对象的子对象中实现。Action 子对象访问 Spring 容器,调用其管理的业务逻辑模块对数据予以处理,并将处理结果输出到 JSP 表单呈现给客户。

3 结束语

文中在 WfMC 提出的工作流参考模型的基础上,结合 Spring IOC 及 Hibernate 持久化技术,设计并实现了一种多层体系结构的轻量级工作流管理系统。项目

开发的实践证明,此工作流管理系统具有低耦合、可扩展性强、构件的可重用性高等特点。而且,在本系统基础上进行信息系统的开发,能节省开发时间,提高开发效率和效益。

参考文献:

- [1] Workflow Management Coalition. The Workflow Reference Model[M]. TC00-1003 Issue 1.1. Hampshire, UK:[s. n.], 1995.
 - [2] 牟军,吕立.使用轻量级框架进行 J2EE 应用开发[J]. 小型微型计算机系统,2006,27(6):1149-1152.
 - [3] Walls C, Breidenbach R. Spring in Action[M]. Greenwich, CT:Manning Publications Co,2005:6-8.
 - [4] OMG. Unified modeling language specification version1.5 [EB/OL]. Object Management Group, 2003. <http://www.omg.org>.
 - [5] 汪涛,吴耿锋,黄力芹,等.工作流管理的过去、现在和未来[J].上海大学学报:自然科学版,1999,5(s):1-7.
 - [6] 谭支鹏.基于角色的工作流模型及其应用[J].小型微型计算机系统,2003,24(6):1064-1066.
 - [7] 范延平,曾建鹰.基于 J2EE 架构的工作流引擎的分析与设计[J].武汉大学学报:信息科学版,2005,30(12):1107-1110.
 - [8] 范玉顺.工作流管理技术基础[M].北京:清华大学出版社,2002:58-60.
 - [9] Nock C. 数据访问模式:面向对象应用中的数据库交互[M]. 鄢爱兰,王安鹏译.北京:中国电力出版社,2004:120-124.
 - [10] 王卫平,王松涛,王名茗.一种基于 J2EE、Spring 和 Hibernate 的轻量级 EAI 构架[J].计算机系统应用,2005(11):38-41.
 - [11] 贾昆,甘切初,高慧颖.数据访问对象模式在企业应用集成中的应用[J].计算机工程与设计,2006,27(3):373-375.
-
- (上接第 106 页)
- 加灵活地调整 ACD 的坐席路由策略,提高系统在实际使用过程中的可用性。
- 参考文献:
- [1] 朱翼勇.呼叫中心排队模型的研究[J].镇江高专学报,2006,19(4):1-2.
 - [2] 李洪荣.建立市政系统呼叫中心[J].中国科技信息,2007(1):1-2.
 - [3] 朱仁祥,朱翼勇,方基奎.重试反馈 M/M/s/k 排队的呼叫中心性能分析[J].系统工程学报,2006,21(6):2-3.
 - [4] 段云峰.呼叫中心系统的构建[R/OL].2000-09. <http://www.ctiforum.com/forum/2000/09/f0910.htm>.
 - [5] 那塔利·卡尔弗特. Gower handbook of call and contact centre management[M].北京:电子工业出版社,2006.
 - [6] 李跃.呼叫中心分布模式[M].北京:人民邮电出版社,2007.
 - [7] 袁道唯,田淑红.呼叫中心的视野与格局[M].北京:清华大学出版社,2005.
 - [8] 段云峰.自动呼叫分配 ACD 系统帮助路由[R/OL].2003-06-19. <http://www.ctiforum.com/callcenter.html>.
 - [9] Shalloway A, Trott J R. Design patterns explained: a new perspective on object-oriented design[M]. Beijing: China Machine Press,2006.
 - [10] Klein M. Windows programmer's guide to DLLs and memory management[M].北京:清华大学出版社,1995.
 - [11] Lippman S B, Lajoie J. C++ primer[M]. Beijing: Posts & Telecom Press, 2005.