

基于语音卡的呼叫中心通用架构

来洪孝, 崔颖安, 崔杜武

(西安理工大学计算机科学与工程学院, 西安 710048)

摘要: 基于分层设计的思想, 设计了一种通用程序架构, 可通过灵活配置来构建中小规模的呼叫中心, 有效缩短了呼叫中心的建设周期, 提高了软件的开发效率, 为中小型呼叫中心的建设提供了一个参考模型, 分析实现了流程控制动态配置技术方案, 通过实践证明基于语音卡的呼叫中心通用架构设计思想的正确性和可行性。

关键词: 呼叫中心; 计算机电信集成; 文语转换; 分层设计; 流程控制

Universal Frame of Call Center Based on Voice Card

LAI Hong-xiao, CUI Ying-an, CUI Du-wu

(School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048)

【Abstract】 Based on the idea of multiple-layer design, this paper designs a universal programming frame, which can build a middle or small-scale call center with flexible configuration. The frame can shorten system development period and promote software efficiency, providing a reference model of experiences for establishing middle and small-sized call centers. It proceeds with analysis and implementation on technical plan of flow control's dynamic configuration. Practice of the projects proves the correctness and feasibility of the universal frame of the call center based on voice card.

【Key words】 call center; computer telecommunication integration(CTI); text-to-speech(TTS); multiple-layer design; flow control

呼叫中心(call center)又称客户服务中心(customer care center), 早期的呼叫中心是指以电话接入为主的呼叫响应中心, 为客户提供各种电话响应服务; 现今的呼叫中心是一种基于计算机电信集成(computer telecommunication integration, CTI)技术, 不断地将通信网、计算机网和信息领域的最新技术集成融合, 并与企业融为一体的综合信息服务系统, 充分利用各种先进的通信手段, 能有效地为客户提供高质量、全方位的服务^[1]。

实现呼叫中心有2种典型解决方案:(1)基于前置自动话务分配(auto calling distribution, ACD)的呼叫中心(简称ACD方案); (2)基于微机和语音板卡的呼叫中心(简称语音卡方案)。二者的组成结构和性能指标完全不同, 其根本分歧点在于CTI技术的实现。其中, ACD方案是基于CTI-Link标准、专业通信和计算机厂家利用各自优势分工合作实现的; 语音卡方案是由板卡生产厂商和应用软件开发商共同推动发展的, 是在微机平台上集成各种功能的语音处理卡, 完成通信接口、语音处理、传真处理、座席转接等功能, 再结合外部的计算机网络实现各种应用系统的需求。目前除了电信等大企业外, 社会上各行业对呼叫中心的需求多为中小规模, 适宜采用语音卡方案。

由于中小规模呼叫中心的功能需求都很相似, 而且各行业的呼叫中心应用系统多数是单独开发, 重复工作量大、软件复用率低、产品开发周期长、功能升级难、规模的稳定性不易保障^[2]。针对这种情况, 本文提出了基于语音卡的呼叫中心通用架构。

1 设计与实现

1.1 设计目标

该架构可通过快速配置实现系统的扩展和业务流程的更

动, 有效缩短构建呼叫中心的开发周期, 提高软件开发效率, 为广大中小规模的呼叫中心建设提供一个参考模型。

1.2 设计原则

架构应具有高可靠性、先进性、可扩展性、可移植性、可维护性、可重用性、灵活性、可配置性。

1.3 分层设计思想

本文从设计原则出发, 通过对呼叫中心的核心技术CTI的探索和研究, 采用分层设计的思想, 将呼叫中心模型分为4层: 接口层, 控制层, 业务层, 资源层(图1)。各层间相互独立, 上层的变化不会影响下层的稳定, 做到高内聚、低耦合。

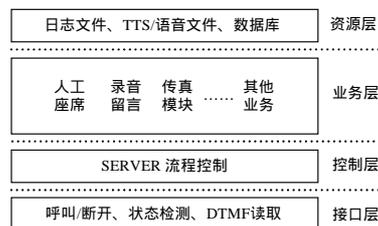


图1 呼叫中心分层实现模型

(1)接口层主要负责提供呼叫建立/断开、信号音检测、主叫号码提取、接收双音频码(dual tone multi-frequency, DTMF)、数字化录放音等。

(2)控制层主要负责处理呼叫中心的流程, 即根据用户的

基金项目: 陕西省教育厅产业化科研基金资助项目(03JC16)

作者简介: 来洪孝(1980-), 男, 硕士研究生, 主研方向: 计算机电话集成技术, 网络通信; 崔颖安, 讲师; 崔杜武, 教授、博士生导师

收稿日期: 2006-12-30 **E-mail:** lai hx0425@126.com

呼叫请求自动路由到业务层中相应的业务功能模块。

(3)业务层主要负责根据业务需要,在接口层和控制层的基础上,加载相应的业务功能模块与业务管理模块,实现CCS(call center system)的基本功能和系统业务的扩展。

(4)资源层主要负责信息数据的存储与提取。根据业务层的要求,资源层将用户数据及相关业务数据库中的数据关联到一起后,及时把信息反馈给业务层使用。该层主要解决了信息数据的复杂性。

在分层设计思想的基础上,通用程序架构可以对流程控制进行动态配置,即只须通过增加、删除、修改业务层上的业务模块,就可以实现不同行业的呼叫中心或同一呼叫中心扩展业务的动态变化。

1.4 呼叫中心通用架构的设计与实现

呼叫中心的通用架构包括以下 5 部分:(1)封装的语音板卡函数库;(2)流程控制配置表;(3)人工座席、录音留言、传真收发这 3 个常用的功能模块和其他业务模块;(4)架构所需 TTS 文本、语音、日志文件和数据库;(5)主程序(如图 2 所示,其中,1 表示使用关系;2 表示查询关系;3 表示增、删、改和查询关系;4 表示调用关系)。

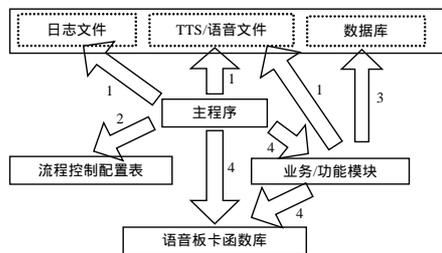


图 2 呼叫中心的通用架构

这里的通用指:(1)基于该架构可以实现不同业务的中小型呼叫中心;(2)增加业务模块或更动业务流程后无须修改程序流程和相关源代码,只须编辑流程控制动态配置表并加载相应业务模块即可;(3)该架构可以作为中小型呼叫中心二次开发的参考模型;(4)该架构适用于不同的操作系统平台和数据库。

1.4.1 语音板卡函数库

封装的语音板卡函数库属于接口层,实现呼叫建立/断开及信号音检测等。封装的函数主要包括获取主叫号码、判断外线摘挂机状态、内线的馈电或振铃、信号音检测、接收用户的按键、数字化录放音等。目前厂商提供的语音卡包括语音处理和电话信令处理功能,并对语音板卡提供的相关底层函数进行适当的封装,以函数库的形式提供给业务/功能模块和主程序进行调用,且该层屏蔽了语音板卡硬件的细节以便程序进行二次开发。

1.4.2 流程控制配置表

流程控制配置表属于控制层,主要字段如下:

- (1)AJXL//按键序列:一个按键序列对应一个当前节点;
- (2)QDAJ//驱动按键:用户每次输入的按键;
- (3)BDXL//比对序列:前一个按键序列;
- (4)DTLJK//动态链接库;
- (5)GNMC//功能名称:dll 内封装的函数名称;
- (6)YWHZJD//有无孩子节点:0 表示无,1 表示有;
- (7)FYNR//放音内容:文中用的是 TTS 文本;
- (8)AJGS//按键个数:驱动按键的个数;
- (9)JDCSBS//节点层数标识:标识当前节点所处的层数。

按键结束用#结尾,如:第 1 次驱动按键是 1,第 2 次驱

动按键是 22,则按键序列为 1#22#。

D-Config(动态配置)算法过程如下:

- (1)初始化状态:JDCSBS 为 F0, AJXL, BDXL 均为空。转步骤(2)。
- (2)播放语音提示:请进行按键选择。转步骤(3)。
- (3)接收 DTMF(双音频码)驱动按键。转步骤(4)。
- (4)查询配置表,若(BDXL+QDAJ=AJXL),转步骤(5);否则,限拨次数+1,转步骤(7)。
- (5)若(YWHZJD=1),转入步骤(6);若(YWHZJD<>1)即(YWHZJD=0),转步骤(8)。
- (6)若(GNMC 为空),限拨次数+1,转步骤(7);否则转步骤(8)。
- (7)若(限拨次数<4),转步骤(2);否则转步骤(16)。
- (8)播放语音提示:请输入用户名和密码。转步骤(9)。
- (9)接收 DTMF 按键,转步骤(10)。
- (10)根据输入的条件进行匹配,查询数据库,查询条件正确则返回查询结果(字符串),转入步骤(11);否则转步骤(13)。
- (11)存储查询结果,转步骤(12)。
- (12)实时播放查询结果;转步骤(17)。
- (13)播放语音提示:请重新输入;转步骤(14)。
- (14)限拨次数+1;转步骤(15)。
- (15)若(限拨次数<4),转步骤(9);否则转步骤(16)。
- (16)播放语音提示:你的限拨次数已到,请挂机;转步骤(17)。
- (17)退出该系统。

算法流程如图 3 所示。

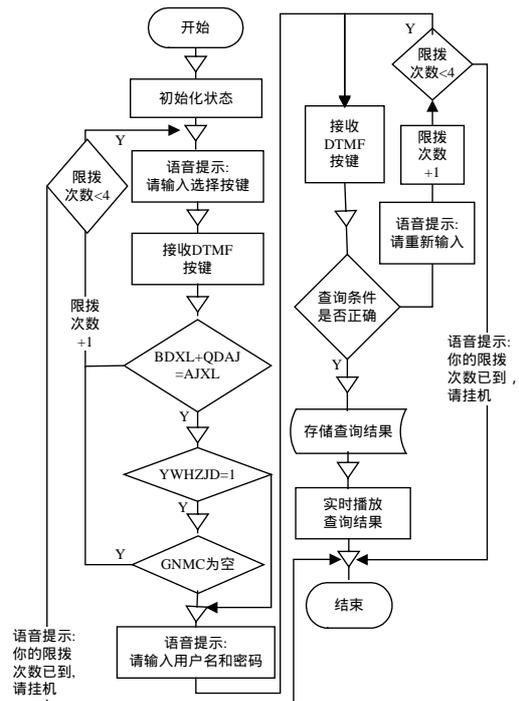


图 3 流程控制的动态配置流程

算法的关键实现代码如下:

```

DJTTS3_CheckPlayTextEnd(Chnl); //检查通道 TTS 放音是否结束,并维持放音持续
If DtmfHit(Chnl) then
//指定的通道有 DTMF 按键
DJTTS3_StopPlayText(Chnl);
//停止通道 TTS 放音
Code:=GetDtmfCode(Chnl);
//取该通道收到的 DTMF 编码
Channel[Chnl].Dtmf:=Channel[Chnl].Dtmf+Convert(Code);
//把 Code 转为字符存储
    
```

1.4.3 业务/功能模块

业务/功能模块属于业务层,将呼叫中心常用的功能模块用动态链接库(dynamic link libraries, DLL)封装实现,便于软件复用,可提高系统开发效率,缩短开发周期。本文把常用的3个功能(人工座席、录音留言、传真收发)和具体业务功能封装成DLL形式,使得系统中同时拨入呼叫中心的多个语音通道的用户访问相同业务时可以共享同一模块,使用同一个DLL。其优点如下:(1)DLL只须装入内存一次,节约了系统内存资源,节省了装入时间,能显著提高系统效率;(2)使用新版本DLL代替旧版本时不会出现重大问题,便于版本升级;(3)DLL独立于任何编程语言,这为软件开发提供了方便。

该模块的关键实现代码如下:

```
Dll_Name:=PChar(Name); //DLL名称
Func_Name:=PChar(FuncName); //函数名称
Dll_Param:=PChar(Param); //调用DLL时传递的参数
AHandle:=LoadLibrary(Dll_Name); //动态加载DLL,返回其句柄
If AHandle<>0 then
  @Call_Func:=GetProcAddress(AHandle,Func_Name); //获得函数
//的入口地址
If Assigned(@Call_Func) then //找到函数
  Result:=Call_Func(Dll_Param); //把函数的指针赋给函数类型
//变量
FreeLibrary(AHandle); //释放句柄
```

当业务变动频繁而需要修改该业务模块的时候,无须对系统架构的源代码作任何修改,只需要重新编辑流程控制配置表并对业务模块稍作调整即可。

1.4.4 日志文件、文语转换/语音文件、数据库

日志文件、文语转换(text-to-speech, TTS)/语音文件、数据库属于资源层,日志文件和TTS/语音文件直接存放在CTI服务器上,而与具体业务相关的数据库则保存在数据库服务器上,这样有利于呼叫中心平台的稳定,系统运行和数据库的运作相对独立,避免因数据库出错而导致系统瘫痪进而丢失所有呼叫记录的情况。该层以数据库为中心,易于系统进行扩展,实现与其他业务系统的无缝集成。

1.4.5 主程序

通用程序架构的主程序是一个通过时间片轮询实现多线程的主调程序,程序流程位为:(1)初始化相关资源卡设备驱动程序,初始化电话卡,加载系统配置文件,初始化主界面,显示呼叫中心系统的提示信息;(2)等待电话呼叫接入,记录该次呼叫的主叫号码、开始时间等信息以便进行后台的数据挖掘,根据语音板卡各通道的忙闲情况合理分配一个通道;(3)根据获得的用户按键,与流程控制配置表比对,路由到相应模块,执行该模块实现其功能;(4)实时检测是否挂机,若已挂机,则自动释放本次呼叫所占语音卡的通道资源。

1.5 关键技术

1.5.1 CTI技术

CTI涵盖了数据通信网络及传统语音通信网络的内容,由传统的计算机电话集成技术(computer telephony integration)演变而来,是随着电信技术和计算机技术的发展而发展起来的。二者的逐步融合,在计算机领域中引入了通信技术,在信息设备中也增加了计算机技术的应用,由此诞生了CTI这个横跨电信和计算机2大领域的新技术。

1.5.2 交互式语音应答技术

交互式语音应答(interactive voice response, IVR)^[3]是指利用计算机语音合成技术,向用户播放不同的语音菜单提示

内容,根据用户的选择情况播放相关的语音信息,为用户提供各种服务。

1.5.3 语音总线技术

语音总线是使各种功能单一的语音板卡连接成一个功能复杂的系统的核心,也是微机CTI系统实现交换的基础。

1.5.4 机间扩展总线技术

限于微机的处理能力,一台计算机只能完成一部分工作或实现某一项功能,要将独立的语音工作站互连成一个大系统,就需要机间总线技术把多台计算机连接起来。

1.5.5 TTS技术

TTS是CTI的一项核心技术,可以将文本即时转换成语音,满足海量信息和动态信息实时发布的需求,其系统框架如图4所示^[4]。

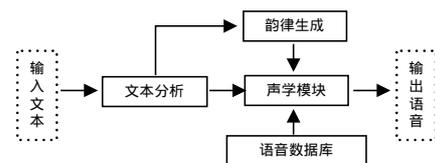


图4 TTS系统

(1)文本分析的主要功能是使计算机知道发什么音、怎么发音,并将发音的方式告诉计算机。对于汉语来说,还要让计算机知道文本中的词边界、短语边界、句子边界,以便发音时设置不同长度的停顿。

(2)韵律生成的主要功能是为合成语音规划出音段特征,如音高、音长和音强等,使合成语音能更准确、自然地表达语意。

(3)声学模块的主要功能是从语音数据库中选取适当的语音基元,拼接成句,再经过韵律修饰,输出自然连续的语声流。

(4)语音数据库的主要功能是存储语音基元。

2 结束语

笔者已在Windows 2000 Professional平台下基于呼叫中心的通用程序架构实现了西安理工大学互动平台中的语音交互平台系统和某电力客户服务中心系统,2个程序架构完全相同,只是根据不同的具体业务配置了相应的流程控制配置表,另外,西安理工大学的语音交互平台系统无需人工座席模块。整个系统必须有东进公司的D161A模拟语音卡、D081A 2F/4F传真资源卡、中继模块、用户模块等硬件支持,驱动程序开发工具包是DJDBDK V3.3.1,TTS是东进TTS3.0,服务器采用的数据库是Oracle 9i,开发工具为Visual C++ 6.0和Delphi7。系统投入使用后,运行状态良好,充分证明了基于语音卡的呼叫中心通用架构设计思想的正确性和可行性。

参考文献

- 1 胡延平,廖蕾,刘启明,等.基于CTI的呼叫中心系统设计与实现[J].计算机工程与设计,2003,24(3).
- 2 Singh H, Lee M H, Lu G, et al. MorphoSys: An Integrated Reconfigurable System for Data-parallel and Computation-intensive Applications[J]. IEEE Transactions on Computers, 2000, 49(5): 465.
- 3 周宽久,曾琳,李瑶.一个语音信息门户的设计与实现[J].计算机工程,2006,32(9): 101.
- 4 叶振兴,蔡莲红.一种基于决策树模型的音库构建和基元选取方法[J].计算机工程,2006,32(10): 189.