CTI概要设计

作者:祁伟

时间:2012年12月13日

**目录**

[前言： 5](#_Toc437458418)

[专业术语： 6](#_Toc437458419)

[1 引言 7](#_Toc437458420)

[1.1 编写目的 7](#_Toc437458421)

[1.2 背景 7](#_Toc437458422)

[1.3 参考资料 7](#_Toc437458423)

[2 总体设计 7](#_Toc437458424)

[2.1 需求规定 7](#_Toc437458425)

[2.2 设计原则 7](#_Toc437458426)

[2.3 设计的基本概念 8](#_Toc437458427)

[2.3.1 状态机 8](#_Toc437458428)

[2.3.1.1 外线状态机 9](#_Toc437458429)

[2.3.1.2 分机状态机 9](#_Toc437458430)

[2.3.1.3 IVR分机状态机 9](#_Toc437458431)

[2.3.1.4 软电话服务端状态机 9](#_Toc437458432)

[2.3.1.5 IVR业务流程状态机 10](#_Toc437458433)

[2.3.1.6 软电话客户端状态机 10](#_Toc437458434)

[2.3.1.7 会议设备状态机 10](#_Toc437458435)

[2.3.1.8 ACD状态机 10](#_Toc437458436)

[2.4 结构 10](#_Toc437458437)

[2.4.1 CTIServer模块： 11](#_Toc437458438)

[2.4.2 语音处理模块： 11](#_Toc437458439)

[2.4.3 ACD模块： 11](#_Toc437458440)

[2.4.4 IVRFlow模块 12](#_Toc437458441)

[2.4.5 DeviceAdapter模块 12](#_Toc437458442)

[2.4.6 数据库模块 12](#_Toc437458443)

[2.4.7 Agent模块 13](#_Toc437458444)

[2.4.8 录音模块 13](#_Toc437458445)

[2.4.8.1 并线录音模式 13](#_Toc437458446)

[2.4.8.2 会议式录音 13](#_Toc437458447)

[2.4.9 日志模块 14](#_Toc437458448)

[2.5 系统配置方式 14](#_Toc437458449)

[2.5.1 分机配置方式 14](#_Toc437458450)

[2.5.2 IVR分机配置方式 14](#_Toc437458451)

[2.5.3 Agent配置方式 15](#_Toc437458452)

[2.5.4 组配置方式 15](#_Toc437458453)

[2.5.5 坐席组配置方式 15](#_Toc437458454)

[2.5.6 ACD配置方式 16](#_Toc437458455)

[2.5.7 示例 16](#_Toc437458456)

[2.6 尚未解决的问题 17](#_Toc437458457)

[2.6.1 集群式部署 17](#_Toc437458458)

[2.6.2 数据分析 17](#_Toc437458459)

[2.6.3 智能化管理系统 17](#_Toc437458460)

[2.6.4 实时监控 17](#_Toc437458461)

[3 接口设计 18](#_Toc437458462)

[3.1 用户接口 18](#_Toc437458463)

[3.1.1 软电话坐席人员接口 18](#_Toc437458464)

[3.1.2 系统管理员接口 18](#_Toc437458465)

[3.1.2.1 分机配置管理 18](#_Toc437458466)

[3.1.2.2 IVR流程编辑 18](#_Toc437458467)

[3.1.2.3 坐席工号管理 18](#_Toc437458468)

[3.1.2.4 技能组管理 18](#_Toc437458469)

[3.1.2.5 系统监控管理 18](#_Toc437458470)

[3.1.3 配置管理接口 18](#_Toc437458471)

[3.2 外部接口 18](#_Toc437458472)

[3.2.1 软电话接口 18](#_Toc437458473)

[3.2.2 数据库查询接口 18](#_Toc437458474)

[3.3 内部接口 19](#_Toc437458475)

[3.3.1 ACD 模块接口 19](#_Toc437458476)

[3.3.2 DeviceAdapter模块接口 19](#_Toc437458477)

[3.3.3 IVR模块接口 19](#_Toc437458478)

[3.3.4 Agent模块接口 19](#_Toc437458479)

[3.3.5 数据库模块接口 19](#_Toc437458480)

[4 运行设计 19](#_Toc437458481)

[4.1 电话呼入流程 20](#_Toc437458482)

[4.2 电话外呼流程： 21](#_Toc437458483)

[5 5、数据库结构 22](#_Toc437458484)

[5.1 数据库设计原则 22](#_Toc437458485)

[5.2 数据库表列表 22](#_Toc437458486)

[5.3 数据库各表字段 22](#_Toc437458487)

[5.3.1 呼入电话记录表（InboundCallDetail） 22](#_Toc437458488)

[5.3.2 呼出电话记录表（OutboundCallDetail） 23](#_Toc437458489)

[5.3.3 呼入IVR记录表（InboundIVRDetail） 24](#_Toc437458490)

[5.3.4 转技能组记录表（TransferSkillGroupDetail） 24](#_Toc437458491)

[5.3.5 转坐席记录表（TransferAgentDetail） 25](#_Toc437458492)

[5.3.6 软电话状态明细记录表（AgentStateDetail） 25](#_Toc437458493)

[5.3.7 录音明细记录表（RecordingDetail） 26](#_Toc437458494)

[5.3.8 会议明细记录表（ConferenceDetail） 26](#_Toc437458495)

[5.4 数据库结构图 27](#_Toc437458496)

# 前言：

本文档主要目的是从可行性上设计一个通用性呼叫中心系统，主要是从功能、技术理论上的设计。尽量做到框架简单，不使用复杂技术和高深设计，做到模块可添加，接口简单易用，稳定性好。

技术来源，朱彤、李鹤峰、郭唯、岳阳。

# 专业术语：

参考《CTI需求分析》专业术语部分。

CSTA: 计算机支持的电信应用协议

SIP：会话控制协议

IMS: 是IP多媒体子系统，是一种全新的多媒体业务形式，它能够满足现在的终端客户更新颖、更多样化多媒体业务的需求

CSCF: 呼叫会话控制功能，CSCF是IMS的核心部分，主要用于基于分组交换的SIP会话控制。在IMS中，CSCF负责对用户多媒体会话进行处理，可以看作IETF架构中的SIP服务器。根据各自不同的主要功能分为代理呼叫会话控制功能P-CSCF(Proxy CSCF)、问询呼叫会话控制功能I-CSCF(Interrogation CSCF)和服务呼叫会话控制功能S-CSCF(Serving CSCF)，三个功能在物理上可以分开，也可以独立。

HSS：本地用户服务器，HSS在IMS中作为用户信息存储的数据库，主要存放用户认证信息、签约用户的特定信息、签约用户的动态信息、网络策略规则和设备标识寄存器信息，用于移动性管理和用户业务数据管理。它是一个逻辑实体，物理上可以由多个物理数据库组成

MRF：多媒体资源功能，MRF主要完成多方呼叫与多媒体会议功能。MRF由多媒体资源功能控制器MRFC(Multimedia Resource Function Controller)和多媒体资源功能处理器MRFP(Multimedia Resource Function Processor)构成，分别完成媒体流的控制和承载功能。MRFC解释从S．CSCF收到的SIP信令，并且使用媒体网关控制协议指令来控制MRFP完成相应的媒体流编解码、转换、混合和播放功能

MGCF：媒体网关控制功能

**SGW:**信令网关

**软电话**：坐席电话控制程序。

**物理电话:** 模拟电话或者VOIP电话。

# 引言

## 编写目的

开发出适合外呼、呼入等功能的通用型呼叫中心系统，主要应用方向适合呼叫中心、电话客服中心、IVR系统、电话销售中心、录音系统、公司办公系统。可以使用电话语音盒应用于4线以下的场合，也可以使用工控机+语音卡模式应用于50线以下场合，也可以使用交换机应用于大规模呼叫中心，也可以按照IMS架构使用集群模式用于超大型呼叫管理系统。

## 背景

## 参考资料

1、FreeSwitch 1.0.6 cookbook

2、数据库系统原理 经济科学出版社 主编/丁保康

3、实用软件工程 第二版 清华大学出版社 郑人杰 殷人昆 陶永磊著

4、《CTI需求分析》

5、基于有限状态机理论和工作流理论的IVR系统的设计与实现 屈金泉

6、《参考基于CSTA模型的呼叫实现方法.doc》

7、《ECMA-269》

# 总体设计

## 需求规定

此文档是以《CTI需求分析》为基础，主要实现上述文档中的功能需求和其他的需求。《CTI需求分析》中要求的功能主要可分为语音通话、ACD、IVR、坐席管理、数据记录、录音、性能需求、兼容性需求等。

## 设计原则

1、系统要有可靠稳定的性能，事件的处理及时、正确，数据的记录准确。

2、系统的操作接口应简单、易用、友好。

3、业务的实现要以脚本来实现，不依赖程序的代码部分，程序作为是一个脚本解析器，处理好资源调度、线程同步、临界资源等问题，功能都在脚本中实现，修改业务只需要修改脚本和配置。

4、配置文件要以xml格式来实现，要做到分层和继承，每个程序的配置文件首先加载“程序名.XML”文件，各个实体对象的配置以面向对象方式实现，子对象的配置可以继承父对象配置，如果子对象有自己的配置将覆盖父对象的配置。

5、系统应对运行中出现的错误有充分的考虑，出现错误后要有良好的错误提示，不影响使用。做到多个通话之间无关性，一个通话的异常不会影响其他的通话。

6、考录系统的扩展性，当系统需要扩展时，只需要叠加需要的模块，其他模块可保持不变。

## 设计的基本概念

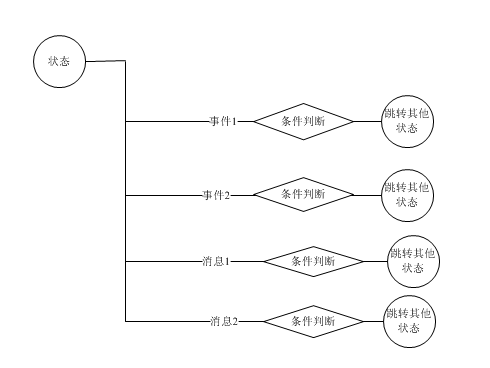
根据CSTA模型，对呼叫的过程可分Call、Device两层进行管理，对呼叫的处理主要在Call层，Device层和通道的状态紧密结合。Call层和Device层间通讯可以通过互发消息来实现。Device层可以接受外界发来的消息，如软交换报上来的消息、IVR发来的消息等，并对它们进行处理。Call层也可以接受外界的消息如RouteServer 发来的消息，但主要是接受Device层发来的消息实现对呼叫的处理控制。根据CSTA模型，暂不支持同层内（即Device层内部、Call层内部）的消息通讯。

在CTI环境中所包含的设备、业务流程都是有状态的，坐席人员的操作、设备的控制、状态变化都是在各个状态之间变化。业务流程、硬件设备在某一时刻都是处于一个状态，业务进行时一直在多个状态之间变化，此部分比较适合用状态机实现。业务逻辑的变换和修改也比较适合用状态机来描述，所以设计的基本理念是各种类型状态机之间的协同工作，包括硬件设备状态机、业务状态机两类。

将系统中的所有资源以分机描述，分机、组、技能组、坐席、IVR都虚拟成分机，所有的资源支持分机的一些基本功能，例如：转入、转出、忙、空闲、超时等。将一个电话转入IVR 可以直接将此分机转入IVR分机号，系统会将这两个资源进行连接。

### 状态机

系统中的设计以状态机为核心，各个模块和功能围绕状态机的状态改变或操作来实现。用状态机来描述设备的状态或业务逻辑，每个状态机的状态是有限的，但是每个状态机都是闭环，能从不同的状态之间转移，所以能实现无限个状态。状态机中包括状态、事件、动作、次态（下一状态）。状态结构图：



由于状态的树形结构，结合xml语言的特性，状态机使用xml语言来描述，以下是一个状态的示例。

<state id="idle">

<onentry>

<log level = "info" expr="enter idle state."/>

<send target="extension" type="cmd" idlocation="extension.number;" from="extension.number;" event="reloadConfig"></send>

</onentry>

<event event="E\_CHG\_ChState" cond="extension.eventData=='S\_CALL\_RINGING';" >

<transition target="ringing"/>

</event>

<event event="E\_CHG\_ChState" cond="extension.eventData=='S\_CALL\_STANDBY';">

<transition target="idle"/>

</event>

<onexit>

<log level = "info" expr="enter idle state."/>

</onexit>

</state>

state：状态元素

onentry:进入状态执行元素

onexit:离开状态执行元素

event:事件元素

log：日志元素

transition：跳转元素

以上6个元素可以实现状态机的主要功能，如需要其他的功能可以扩展其他的元素。

根据系统中的实体对象，计划分为八种类型状态机。

#### 外线状态机

每个外线根据呼叫号码和呼叫ID 分配一个状态机，当呼叫创建时生成，挂机时销毁。

#### 分机状态机

每个分机根据呼叫号码和呼叫ID分配一个状态机，当呼叫创建时生成，挂机时销毁。

#### IVR分机状态机

相当于只为IVR功能服务的外线分机，如果外线分机上可以满足IVR的媒体操作，此状态机可以和外线分机合并；每个配置的IVR分机都启用一个状态机。

#### 软电话服务端状态机

此状态机是由软件实现的状态机，并没有物理化的分机，是描述软电话在服务器端的状态，分机配置后此状态机生成。

#### IVR业务流程状态机

此状态机是一个逻辑上的设备，并没有直接相对应的硬件，是IVR的业务流程，即IVR流程脚本，根据配置生成相应的状态机。

#### 软电话客户端状态机

软电话客户端的状态，显示软电话的业务状态，控制软电话的界面显示等，此状态机在启动软电话程序后生成，关闭软电话后销毁。

#### 会议设备状态机

会议设备的状态，根据会议号和会议ID分配的状态机，当会议创建时建立，会议结束后状态机销毁。

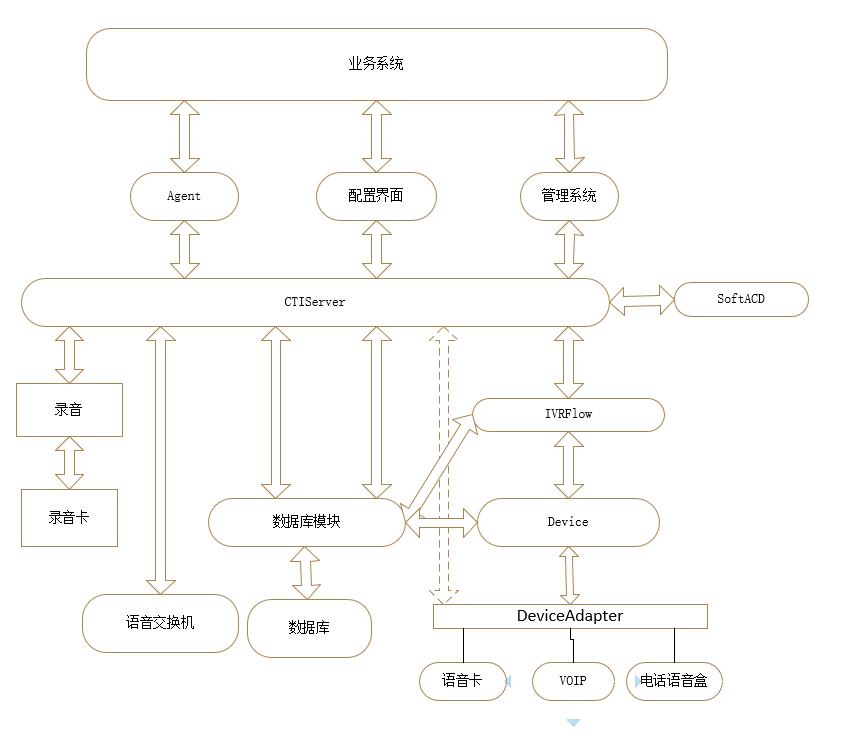
#### ACD状态机

ACD模块使用状态机来实现，ACD的业务使用状态机脚本编写，当有一个话路产生时，会新建一个ACD状态机，以后的ACD业务会在此状态机中完成，当话路销毁时通知此状态机，然后此状态机销毁。

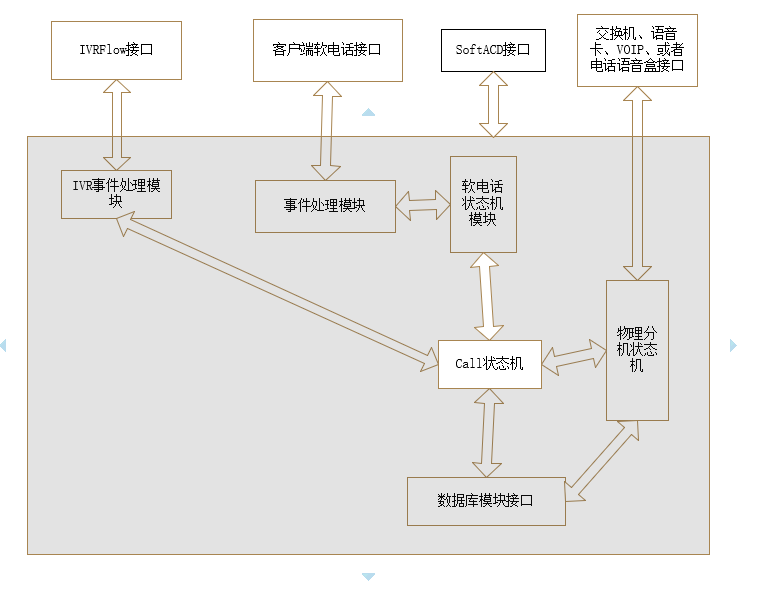
## 结构

系统中按功能的不同划分为多个模块，各个模块围绕状态机来实现，各个模块之间的通信通过传递json消息来实现，基于json的可扩展性，对各个模块之间的功能扩展会比较简便。各个模块将收到的消息放入消息队列中，按照顺序进行处理，处理模块根据消息来改变状态机的状态，执行状态机的动作对硬件设备进行控制。

以状态机概念为基础，结合本人这几年对CTI的理解，将系统划分为以下几个主要模块：



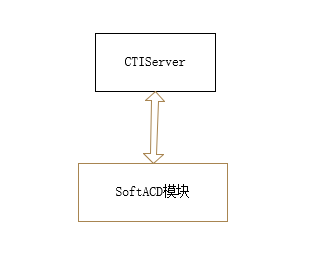
### CTIServer模块：

负责软电话的登录、管理软电话状态迁移、物理电话状态和软电话之间的逻辑同步、和软电话的交互、 和媒体模块之间消息传递（语音交换机、语音卡、VOIP服务器、语音盒等）、资源管理（分机资源、软电话资源）、和SoftACD进行资源同步，路由请求 。  


### 语音处理模块：

包括语音交换机、语音卡、VOIP系统、话机等，此系统中不包括此部分，实现过程中系统会调用语音处理模块的接口来控制语音和其他媒体。

### ACD模块：

此模块功能是系统中的路由调度和排队模块核心。支持话务桥接、话务路由、队列功能。此模块中所有的功能都可配置，可无限制扩充配置。SoftACD根据业务需求将需要监控的资源状态维护在自己的进程中，可以通过CTIServer接口主动查询和CTIServer根据配置将某一部分资源的状态推送给指定的SoftACD。某一个SoftACD实例中所需要访问的资源必需在本实例中，此模块不支持集群部署，但可以支持热备模式，某一实例出现故障后，新启动的实例可以通过接口查询出旧实例所维护的资源状态，可以在不中断业务的情况下继续执行。  


ACD 队列排队算法

1、线性加权优先级排队算法。

2、预测等待时间算法。

3、基于号码识别的路由选择。

4、基于坐席技能级别的路由算法

5、最长等待时间算法

6、坐席空闲时间最长算法

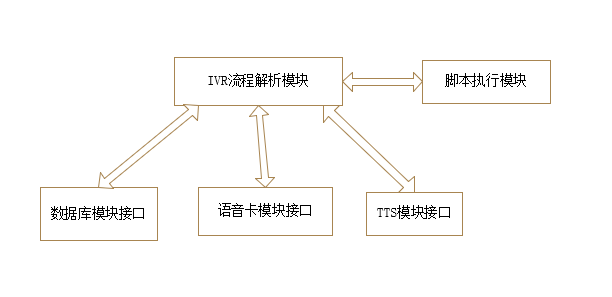
7、坐席服务次数最少算法

8、VIP、黑名单

9、第三方实现接口

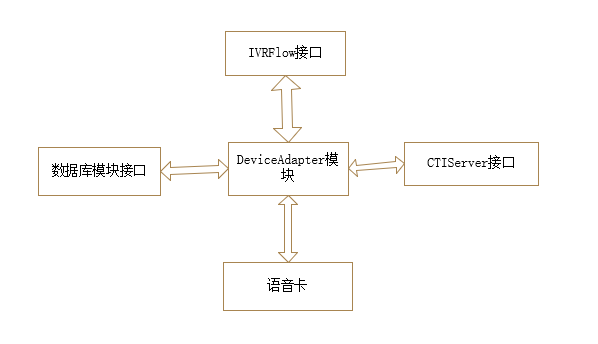
### IVRFlow模块

此模块主要解析IVR流程，根据通话事件决定流程走向，IVR功能需要调用语音卡接口、TTS接口、脚本执行模块、数据库模块，通道放音等操作。



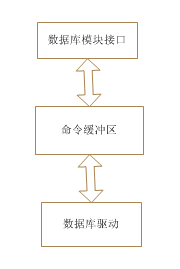
### DeviceAdapter模块

负责对媒体设备的控制，接收来自Call、IVRFlow模块的通话的命令，完成放音、收号、呼叫、来电、会议等功能，还可以保留数据库接口，完成对数据的记录。



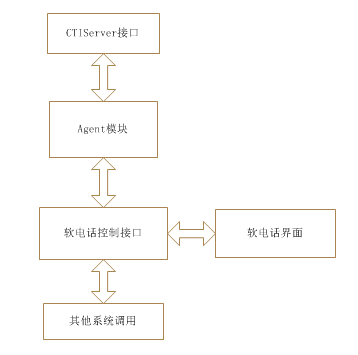
### 数据库模块

负责接收来自其他模块的数据记录命令，将数据记录进数据库中。



### Agent模块

软电话客户端模块，负责CTIServer、操作人员之间的连接，和对应的物理电话状态同步，返回所有的电话状态事件，以软件形势控制物理电话。提供和其他系统耦合的接口和坐席人员操作接口。



### 录音模块

受CTI控制，启动录音设备进行录音，将录音文件存储在指定的设备中，并将录音结果返回给CTI。录音方式可以使用直接录音、并线录音、会议录音、备份录音四种方式。

#### 直接录音模式

物理话机本身支持录音功能，可以直接调用物理话机录音接口进行录音。

#### 并线录音模式

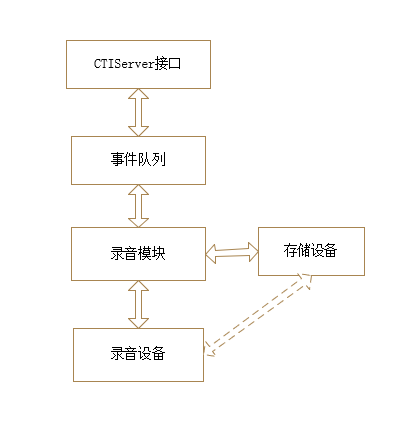
采用高阻录音设备和分机进行并联，每个话机链接一个录音设备，当需要录音时启动录音设备。

#### 会议式录音

将分机和录音设备都接入交换机或语音卡中，当需要录音时创建一个会议，将分机、录音设备都加入，启动录音设备进行录音，采用此方式录音，并不需要每个分机配置一个录音设备，只需要在录音时选用其中一个录音设备。

#### 备份录音

为防止录音丢失，采用和系统无关的抓包方式进行录音，对录音做到双保险



### 日志模块

日志模块并不是一个单独运行的模块，在代码的各个部分都和日志模块都有耦合，在代码中需要记录日志的地方加入日志输出代码，日志模块可以收集此部分输出到一定地方，可以输出到屏幕、输出到文件、输出到端口。

## 系统配置方式

系统支持从数据库中读取配置和从文件中加载配置两种方式，每个进程启动时会加载一个xml格式的配置文件，配置文件为“程序名.xml”，配置文件定义可以分为Call、 EventReport（消息接口）、FreeSwitch（软交换）、IVR（IVR分机）、MySql（数据库）、坐席（Agent）。

### 分机配置方式

<FreeSwitch id = "freeswitch001" host="47.88.171.173" port ="18021" password="qiwei">

<MySql>mysql</MySql>

<Devices>

<Device DeviceID="1000" StateMachine="./conf/FreeSwitchDevice.xml"></Device>

<Device DeviceID=".\*" StateMachine="./conf/FreeSwitchDevice.xml"></Device>

</Devices>

</FreeSwitch>

Device: 此元素是设备元素，表明是一个设备的配置信息。

DeviceID:此设备的电话号码

StateMachine：状态机脚本文件。

### IVR分机配置方式

<IVR>

<IVR DeviceID="5000" StateMachine="./config/IVR.xml"/ >

</IVR>

IVR: 此元素是设备，表明是一个设备的配置信息。

DeviceID:设备的电话号码，符合此号码规则的呼叫加载此项配置。

StateMachine：状态机脚本文件。

### Agent配置方式

<Agent>

<Agent Id=”2000” Name=”” Password=”” StateMachine=”../config/Agent.xml” ></Agent>

</Agent>

Agent：表明此元素是一个坐席配置。

Id：坐席 id，坐席的唯一标识。

Name：坐席名称

Password:坐席密码

StateMachine：状态机脚本文件。

### Call配置方式

<Call>

<Call DeviceID ="5000" StateMachine=”../config/Call.xml”/>

</Call>

Call:表明此元素是Call的配置。

DeviceID：设备的电话号码，符合此号码规则的呼叫加载此项配置。

StateMachine：状态机脚本文件

子元素：组下的子元素是组所包含的分机号码。如<Extension>5001</Extension>表示此组中包括5001分机。

### Mysql配置方式

<MySql id="mysql" Host="192.168.2.232:3306" UserID="root" Password="unitcc" DataBase="cc"/>

Host：数据库地址。

UserID：数据库用户名。

Password:数据库密码

DataBase:数据库名称

### 示例

以下是一个配置文件样例：

<?xml version="1.0" encoding="UTF-8"?>

<Config>

<Call id="call001" StateMachine="./conf/Call.xml"/>

<EventReport id="eventreport" TCPPort="9985" WSPort="9986">

<Call>call001</Call>

</EventReport>

<FreeSwitch id = "freeswitch001" host="127.0.0.1" port ="8021" password="1234">

<MySql>mysql</MySql>

<Devices>

<Device DeviceID="1000" StateMachine="./conf/FreeSwitchDevice.xml"></Device>

<Device DeviceID=".\*" StateMachine="./conf/FreeSwitchDevice.xml"></Device>

</Devices>

</FreeSwitch>

<MySql id="mysql" Host="192.168.1.232:3306" UserID="root" Password="1234 " DataBase="cti "/>

</Config>

## 集群式部署

可以同时部署多个功能相同的服务器，服务器之间负载均衡，当其中一个出现问题时，业务自动转移到其他服务器上。

### SIP服务器

SIP服务器参考IMS系统架构，使用kamailio + FreeSwitch 部署，kamailio 作为代理服务器、位置服务器、重定向服务器、软交换服务器；FreeSwitch 作为媒体服务器。

1. 代理收到注册或者INVITE消息后发送到位置服务器。
2. 位置服务器收到消息，如果是注册消息，从数据库中查询是否已经注册到某个软交换上，已经注册响应重定向到软交换信息，未注册响应重定向一组可用的软交换。如果是INVITE消息，从数据库中查询被叫是否在本地数据库中，如果是本地用户，响应重定向消息到本地用户注册的软交换；不是本地用户根据规则响应一组重定向软交换。
3. 软交换收到注册消息，将用户在本服务器注册并更新数据库。
4. 软交换收到INVITE消息，如果INVITE是从FreeSwitch发出，直接呼叫被叫；否则，发送INVITE到FreeSwitch。FreeSwitch 发送事件到CTI。

### CTI

CTI 通过esl和freeswitch连接，发送命令给FreeSwitch，esl连接不接收呼叫事件；呼叫事件在FreeSwitch 的拨号计划中通过socket通知到cm。

### SoftACD

CTI根据配置将资源状态变化发送到指定的SoftACD上，SoftACD也可以根据配置向CTI获取指定的资源状态，当CTI是集群部署时，必需向多个CTI发送请求获取资源。

CTI有路由需要时，发送请求到SoftACD，SoftACD根据业务规则返回给路由的目标地址给CTI。

## 尚未解决的问题

### 数据分析

将CTI 中记录的基本数据进行统计，反映出系统的硬件使用率和管理方面的概况。

### 智能化管理系统

可以帮助系统管理人员智能的进行管理和提示。可以对系统中的坐席排班，技能组划分，未接电话提示等进行辅助性提示。

### 实时监控

可以提供一个模块收集各个状态机的状态，并提供主动通知和查询功能，实现《CTI需求分析》3.11 实时监控。

# 接口设计

## 用户接口

### 软电话坐席人员接口

参考《CTI需求分析》3.5 软电话

### 系统管理员接口

#### 分机配置管理

分机的配置信息保存在数据库中，SIP注册服务器通过读取数据库进行分机信息验证。

#### IVR流程编辑

IVR流程主要以xml文件构成，签入脚本文件，此脚本编辑人工直接修改比较麻烦，不直观，需要提供可视化编辑器，实现方式为用Web版本，打开编辑器后可以加载系统中存在的IVR流程，编辑界面模仿VS界面，左侧是结构区，以树形结构显示出IVR的功能节点，右侧是功能编辑区，选中一个功能后，可以在右侧编辑他的属性。在结构区一个功能节点上点击右键，可以添加其他的功能节点。编辑完成后可以检查、测试所编辑的流程。测试完成后应用新流程。

#### 坐席工号管理

坐席的工号配置保存在数据库中，多个CTI实例读取同一份数据库的账号配置，同一个组或者同一个企业的坐席可以登录到不同的CTI上。

#### 技能组管理

和坐席工号管理的修改相同。

#### 系统监控管理

可以配置一些错误发生后执行的动作，可以配置发送邮件、拨打维护人员电话。

### 配置管理接口

系统的配置功能都是以xml+数据库实现，xml中的配置包含一些程序启动所需要基本的信息。包括监听端口号，数据库地址，线程数量等；数据库中的配置是一些业务配置，包含账号管理，流程脚本等。系统启动后加载xml配置，修改xml的配置需要重启程序，修改数据库配置不需要重启程序。

## 外部接口

### 软电话接口

提供3．1．1 软电话坐席人员接口的所有功能，以API方式提供，方便和其他程序的耦合。

### 数据库查询接口

系统中的数据记录完成后需要供业务人员查询、分析。因此需要定义出数据库的查询接口，对数据库中的表不允许程序直接访问，都以存储过程方式间接访问，对数据库中的访问用户设置多种权限，每个用户下可以允许相应的存储过程。

## 内部接口

内部接口是系统中各个模块之间互相通信使用的接口，其他开发人员也可以根据此接口进行扩展功能的开发。消息体使用json格式实现，消息类型分为请求、响应、事件三种类型。

如：{"type":"event","event":"notify","status":0,"param":""}

各个内部模块保持一个消息队列，向此模块交互时需要将消息发送到此队列，此模块的执行线程会获取此消息执行。

### SoftACD 模块接口

### DeviceAdapter模块接口

### IVR模块接口

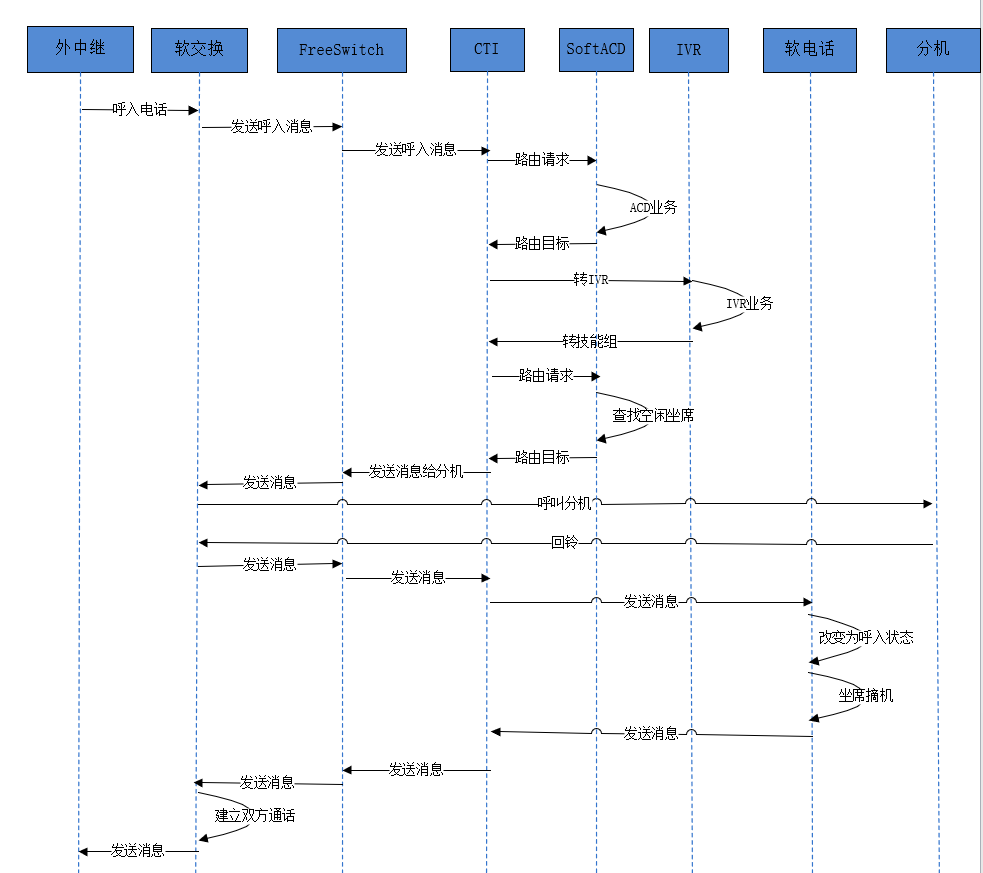
### Agent模块接口

### 数据库模块接口

# 运行设计

运行设计是针对程序在运行过程中的数据流向和对数据的处理方法，是程序的实时动态，包括统中的业务和各个模块、状态机之间的配合。

## 电话呼入流程



步骤：

1、中继上有电话呼入。

2、软交换上收到呼叫，转给FreeSwitch

3、FreeSwitch传送给CTI。

4、CTI根据配置发送路由请求到需要处理的SoftACD。

5、SoftACD根据主被叫号码等信息，进入ACD业务。

6、如果是转IVR的电话，转IVR。

7、电话进去IVR流程，如果客户操作了转人工技能组，发送消息给CTI。

8、CTI发送路由请求到SoftACD，SoftACD查找可以转接的坐席，返回给CTI。

9、CTI呼叫坐席的分机，将消息发送到FreeSwitch。

10、FreeSwitch 发起呼叫到软交换。

11、软交换根据SIP位置服务器返回的信息，继续呼叫到SIP注册服务器。

12、SIP注册服务器呼叫分机。

13、分机回铃->SIP注册服务器->软交换->FreeSwitch。

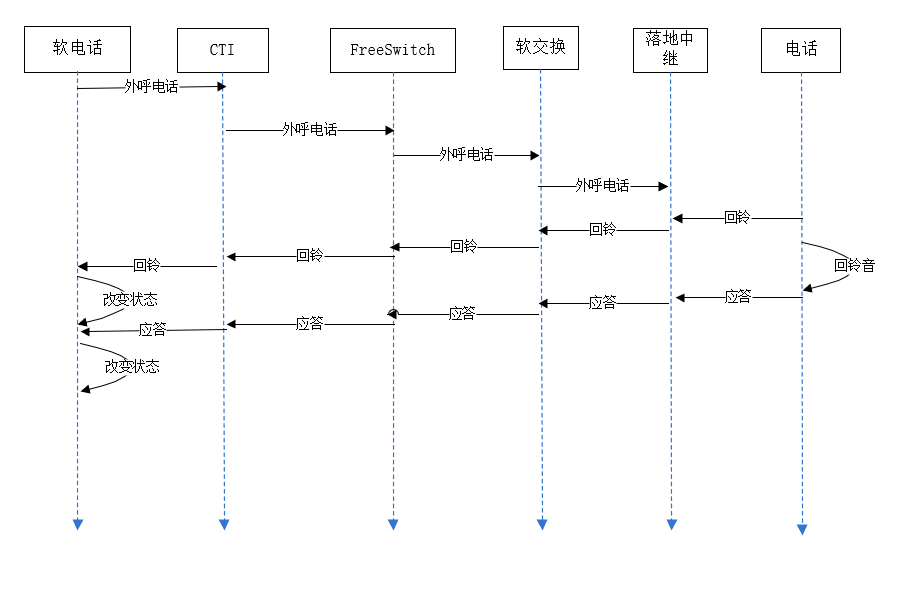
14、FreeSwitch将分机回铃消息发送给CTI。

15、CTI收到分机回铃消息后，改变软电话状态机，并将消息发送给软电话客户端。

16、坐席点击“应答”按钮摘机，分机应答->SIP注册服务器->软交换->FreeSwitch->CTI。

17、CTI收到消息后，改变软电话状态机，和分机状态机。

## 电话外呼流程：



步骤：

1、坐席操作软电话，输入电话号码后进行外呼，软电话将消息发送到CTI。

2、CTI将外呼消息发送给FreeSwitch。

3、FreeSwitch收到消息后，通过软交换外呼。

4、软交换外呼落地中继，经过PSTN到达被叫端。

5、被叫设备回铃并开始振铃。

6、中继将回铃信息传送到软交换->FreeSwitch->CTI。

7、CTI收到回铃消息，改变分机状态机、软电话状态机，并将消息发送给软电话客户端。

8、软电话客户端根据回铃消息改变软电话状态。

9、被叫应答，将应答消息发送给软交换->FreeSwitch->CTI。

10、CTI收到被叫应答消息，改变软电话状态机，发送消息给软电话客户端。

11、软电话客户端收到被叫应答消息，改变软电话客户端状态。

## CSTA流程实现

图例说明：

### Answer Call







D2是主叫，D1是被叫。被叫通道D1的控制者向D1发CmdAnswerCall，D1再向C1发AnswerCall。此时存在两种情况：

D1、D2还未连接，C1收到AnswerCall后使D1、D2连接，连接成功后分别向D1、D2发Established，并向 D2发AnswerRequest。（如图2所视）

D1、D2已经连接，C1收到AnswerCall后分别向D2发AnswerRequest。（如图3所视）

1.2 ClearCall Service



 当D1的控制者向D1发出ClearCall，或D1因为本身的原因要使呼叫清除时，D1向C1发ClearCall。C1向它所连的Device都报告一个CallCleared事件（BroadcastCallClearedEvent）。使各Device对应的通道释放，如果通道是ClearChannel则将通道状态置为空闲，而不用释放通道。

1.3ClearConnection



 在会议等多方连接的情况下，D1为主控设备。若D1的控制者要清除跟D3的连接，它要向D1发ClearConnection消息，其中包含着要被清除的Connection：C1D3的信息。D1再向C1发ClearConnection。C1将D3和它的连接清除，并分别向D1和 D3发ConnectionCleared事件，D1收到ConnectionCleared后也向其控制者发ConnectionCleared事件，表示呼叫清除成功。

1.4MakeCall





D1的控制设备向D1发MakeCallRequest，D1向C1发MakeCallRequest， C1在选择通道后向D2发Delivered消息。D1根据实际情况呼出或通知其控制设备。C1收到D2的地址全信号AddressCompleteRequest后向D1转发AddressCompleteRequest。D1收到AddressComplete后根据需要作相应的处理（如SS7通道要向交换机发PPLEventRequest，使交换机向对端发ACM），处理完成毕后向C1发IndicateAddressCompleteTransfered以通知C1。余下的过程跟AnswerCall的过程一样。

1.5 HoldCall





D1—C1—D2原来处于连接状态，D1的控制者D1发HoldCall消息。D1马上回响应RespHoldCall，并向C1发HoldCall。C1收到HoldCall后向交换机发ParkChannel消息，使D1、D2处于Parked状态。C1在收到ParkChannelAck后分别向D1、D2发Held事件。D1在再把Held事件转发给控制者，表示HoldCall成功。

1.6 RetrieveCall





HoldCall后D1、C1、D2的状态如图（1）的左图所视。D1的控制者向D1发RetrieveCall消息。D1将RetrieveCall转发到C1。C1向交换机发Connect命令。使D1、D2重新连接。C1收到ConnectAck后分别向D1、D2发Established事件。D1收到Established后向控制者发Retrieved事件，表示RetrieveCall成功。

1.7 ConsultationCall





D1—C1—D2原来已经处于连接状态，App向D1发出ConsultationCall消息。D1收到ConsultationCall消息后，向C1发HoldCall使D1、D2处于Parked状态。D1再进行MakeCall产生一个新的Call：C2。D1收到MakeCall 的响应RespMakeCall后，向App发RespConsultationCall消息和Held事件（若RespMakeCall为失败的响应时，则D1只向App发表示失败的RespConsultationCall）。C2使D1、D2处于l连接状态（MakeCall的具体过程参见1.9 MakeCall）。当D1收到C2发来的AnswerRequest后，D1向App报告Answered事件，表示ConsultationCall成功。

在SC的流程中ConsultationCall有两种形式：（1）本端ConsultationCall：App为D1的控制者。（2）远端ConsultationCall：App为D2的控制者。这两种形式在内部流程上都是由D1发起ConsultationCall的，区别是外部发起者的不同，（1）的发起者为D1的控制设备；（2）的发起者是D3的控制设备。D1向外界收发消息的对象也随之不同。

1.8AlternateCall

状态迁移如下图所视:



ConsultationCall后D1、D3处于连接状态，D1、D2处于保持状态，D1收到外界发来的AlternateCall命令后向C2发HoldCall命令，C3使D1、D3处于Parked状态并向D1发Held事件，D1收到Held事件后再向C1发RetrieveCall命令，使D1、D2重连，连接成功后C1向D1 发Established。

若AlternateCall失败则请求者会收到失败的RespAlternateCall，如果成功，请求者会收到成功的RespAlternateCall及Established。



1.9 DivertCall



 D2为主叫端发起呼叫，D1为被叫端。D1已经向C1发了AddressCompleteRequest但还没发AnswerCall。此时D1收到App发来的DivertCall消息，则D1向C1发DivertCall。C1重新选择通道，向D3发DeliverCall。当C2收到D3的AddressCompleteRequest后再向D2发RespDivertCall。D2将RespDiverCall转发给控制设备App，表示DivertCall成功。之后，D2释放回到空闲状态，C2则按一般的呼叫流程使D1、D3连接。

1.10 InterventCall

D3发起强拆请求到C1，连接标识是C1D1，C1根据连接标识向D1发送ConnectionCleared事件，向D3发送Delivered事件，D3应答后，建立连接。

1.10 ReconnectCall



 ConsultationCall后状态如图（1）左图所视。若要恢复ConsultationCall前的状态，D1的控制者向D1发ReconnectCall消息。D1再向C2发ClearCall。使D1、D3的连接断开，再向C1发RetrieveCall。C1向交换机发Connect消息使D1、D2重连。RetrieveCall成功后，D1向其控制者发Retrieved事件，表示ReconnectCall成功。

1.11 Single Step Transfer Call



 D1、C1、D2原来处于正常的连接状态。D1的控制者向D1发SingleStepTransfer消息。D1收到后发起SingleStepTransfer，创建一个新Call：C2。C2选择通道后向D3发Delivered，在收到D3的AddressCompleteRequest后向D1发SingleStepTransfer的回应消息RespSingleStepTransfer，并向D2发Transferred事件以通知D2。D1将RespSingleStepTransfer消息转发给控制者，表示单步转移成功。D2、C2、D3余下的呼叫过程跟AnswerCall一致。

1.12 Transfer Call





ConsultationCall模型处于图（1）左图的状态。D1的控制者向D1发TransferCall消息。D1收到后发起TransferCall，创建一个新Call：C3。C3向交换机发Connect消息，使D2、D3连接。C3收到ConnectAck后分别向D1、D2、D3发Transferred事件。D1收到Transferred后向C1、C2发DeleteCall使C1、C2释放回到空闲状态，并向其控制设备发CallCleared事件。

1.13ConferenceCall



 D1—C1—D2本来处于一般的连接状态，D1的控制者向D1发出ConferenceCall，D1向C1发HoldCall，使D1、D2处于Parked状态。由D1发起ConferenceCall，创建一个新的Call ：C2，再由C2创建会议资源，并把D1、D2连到会议资源上。由于D1是会议的控制者，每一个Device连到会议上都向它报告，即C2收到ConnectToConferenceAck后向D1报告Conferenced事件。会议建立成功后D1向C1发DeleteCall消息，使C1释放。

 要把D3加入会议，D1的控制者先向D1发ConsultationCall消息。D1向C2发HoldCall，使D1暂时和会议资源断开。D1再MakeCall产生一个新的Call：C3，使D1、D3连接，形成ConsultationCall的形式。 D1的控制者再向D1发ConferenceCall消息。D1先向C3发HoldCall使D1、D3处于Parked状态，再向C2发RetrieveCall使D1重新连到会议资源上，向C2发AddToConference使D3连到会议资源上。会议建立成功后，D1会向C3发DeleteCall使C3释放。如果把其他的通道加入会议，其过程和将D3加入会议的过程一样。

1.14 ListenCall

D1、D2为被监听的通话方，D3为监听方。

D3发起请求，创建监听呼叫，监听呼叫申请会议资源，把D1、D2连接到会议上，D3以只听方式连接到会议上就可以实现监听。



# 5、数据库结构

## 数据库设计原则

此数据库按照关系型数据库进行设计，要求对系统中的所有实体的状态变化都有记录，各个实体之间的关联关系明确，数据记录完整，不重复，不冲突，除以上要求外还遵循以下数据库设计规则：

1. 第一范式（1NF）,如果关系模式R的每个关系r的属性值都是不可分的原子值，那么称R是第一范式（First Normal Form,简记为1NF）的模式。
2. 第二范式（2NF）,如果关系模式R是1NF,且每个非主属性完全函数依赖于候选键，那么称R是第二范式（2NF）的模式。如果数据库模式中每个关系模式都是2NF,则称数据库模式为2NF的数据库模式。
3. 第三范式（3NF）,如果关系模式R是1NF，且每个非主属性都不传递依赖于R的候选键，那么称R是第三范式（3NF）的模式。如果数据库模式中每个关系模式都是3NF，则称其为3NF的数据库模式。

## 数据库表列表

数据表是记录系统中数据的实体，表的名称和功能如下

|  |  |  |
| --- | --- | --- |
| 序号 | 表名 | 说明 |
| 1 | InboundCallDetail | 呼入电话记录，记录系统中的呼入电话。 |
| 2 | OutboundCallDetail | 呼出电话记录，记录系统中呼出的电话。 |
| 3 | InboundIVRDetail | 呼入IVR记录。 |
| 4 | TransferSkillGroupDetail | 转技能组记录。 |
| 5 | TransferAgentDetail | 转坐席记录 |
| 6 | AgentStateDetail | 软电话状态明细 |
| 7 | RecordingDetail | 录音明细 |
| 8 | ConferenceDetail | 会议明细 |
| 9 | AgentLogin | 坐席登录表 |

## 数据库各表字段

### 坐席登陆表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | char | 64 | 0 | Not Null |  | uuid |
| 2 | AgentID | char | 10 | 0 | Not Null |  | 坐席ID |
| 3 | Extension | char | 10 | 0 | Not Null |  | 坐席此次登录的分机号 |
| 4 | LoginTime | DateTime | 8 | 0 | Not Null |  | 坐席登陆的时间 |
| 5 | LogoutTime | DateTime | 8 | 0 | NULL | NULL | 坐席登出的时间 |

### 坐席状态表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | char | 64 | 0 | Not Null |  | uuid |
| 2 | AgentID | char | 10 | 0 | Not Null |  | 坐席ID |
| 3 | State | int | 4 |  | NULL |  | 坐席状态类型：0-就绪，1-离席，2-事后处理 |
| 4 | Begintime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 5 | EndTime | DateTime | 8 | 0 | NULL | NULL | 结束时间 |
| 6 | AUX\_Code | int | 4 | 0 | NULL | -1 | 离席原因码 |
| 7 | AUX\_Name | char | 50 |  | NULL |  | 离席原因文字描述 |
| 8 | Remark | char | 255 |  | NULL |  | 备注 |

### 呼入电话记录表（InboundCallDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | char | 64 | 0 | Not Null |  | uuid |
| 2 | CallerNumber | Char | 20 | 0 | Null | NULL | 主叫号码 |
| 3 | CalledNumber | Char | 20 | 0 | NULL | NULL | 被叫号码 |
| 4 | StartTime | DateTime | 8 | 0 | Not Null |  | 呼入开始时间 |
| 5 | EndTime | DateTime | 8 | 0 | Not Null |  | 呼叫结束时间 |
| 6 | HangUpReason | Int | 4 | 0 | Not Null |  | 挂机原因 |
| 7 | ChannelId | Int | 4 | 0 | Null | NULL | 通道号 |
| 8 | TrunkId | Int | 4 | 0 | Null | NULL | 中继号 |
| 9 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |
| 10 | RecordingSessionID | GUID | 64 | 0 | Null | Null | 录音ID |

主键：GlobalSessionID

### 呼出电话记录表（OutboundCallDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | CallID | Char | 64 | 0 | Not Null |  | Call ID |
| 2 | ConnectID | Char | 64 | 0 | Not Null |  | 连接ID |
| 3 | Direction | Int | 1 | 0 | Not Null | 0 | 0:外呼，1：呼入 |
| 3 | Caller | Char | 20 | 0 | NULL | NULL | 主叫号码 |
| 3 | AgentID | Char | 10 | 0 | NULL | NULL | 坐席工号 |
| 4 | Called | Char | 20 | 0 | NULL | NULL | 被叫号码 |
| 5 | StartTime | DateTime | 8 | 0 | Not Null |  | 呼叫开始时间 |
| 6 | RingingTime | DateTime | 8 | 0 | NULL | NULL | 外呼振铃时间 |
| 7 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 8 | EndTime | DateTime | 8 | 0 | Not Null |  | 呼叫结束时间 |
| 9 | HangUpReason | Int | 4 | 0 | Not Null |  | 挂机原因 |
| 10 | ChannelId | Int | 4 | 0 | Null | NULL | 通道号 |
| 11 | TrunkId | Int | 4 | 0 | Null | NULL | 中继号 |
| 13 | RecordSessionID | GUID | 74 | 0 | Null |  | 录音ID |

### 呼入IVR记录表（InboundIVRDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 3 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 4 | EndTime | DateTime | 8 | 0 | Not Null |  | 转出时间 |
| 5 | ChannelId | Int | 4 | 0 | Null | NULL | 通道号 |
| 6 | TrunkId | Int | 4 | 0 | Null | NULL | 中继号 |
| 7 | IVRSessionID | GUID | 64 | 0 | Not Null |  | IVR唯一标识 |
| 8 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 转技能组记录表（TransferSkillGroupDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | SkillGroupID | Char | 10 | 0 | Not Null |  | 技能组号 |
| 3 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 4 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 5 | AgentID | Char | 10 | 0 | NULL | NULL | 应答坐席工号 |
| 6 | ExtensionID | Char | 10 | 0 | NULL | NULL | 应答分机号 |
| 7 | EndTime | DateTime | 8 | 0 | Not Null |  | 转出时间 |
| 8 | NoAnswerReasion | Int | 4 | 0 | Null | NULL | 无应答原因 |
| 9 | SkillSessionID | GUID | 64 | 0 | Null | NULL | 转技能组唯一标识 |
| 10 | IVRSessionID | GUID | 64 | 0 | Not Null |  | IVR唯一标识 |
| 11 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 转坐席记录表（TransferAgentDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 3 | RingTime | DateTime | 8 | 0 | NULL | NULL | 振铃时间 |
| 4 | AgentID | Char | 10 | 0 | NULL | NULL | 坐席工号 |
| 5 | ExtensionID | Char | 10 | 0 | NULL | NULL | 分机号 |
| 6 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 7 | EndTime | DateTime | 8 | 0 | Not Null |  | 挂机时间 |
| 8 | NoAnswerReasion | Int | 4 | 0 | Null | NULL | 无应答原因 |
| 9 | SkillSessionID | GUID | 64 | 0 | Null | NULL | 转技能组唯一标识 |
| 10 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 软电话状态明细记录表（AgentStateDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 3 | AgentID | Char | 10 | 0 | Not Null |  | 坐席工号 |
| 4 | ExtensionID | Char | 10 | 0 | NULL | NULL | 分机号 |
| 5 | EndTime | DateTime | 8 | 0 | Not Null |  | 结束时间 |
| 6 | AgentSessionID | GUID | 64 | 0 | Not Null |  | 全局标识 |
| 7 | RecordingSessionID | GUID | 64 | 0 | Null |  | 录音ID |

### 录音明细记录表（RecordingDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | CallerNumber | Char | 20 | 0 | Null | Null | 主叫号码 |
| 3 | CalledNumber | Char | 20 | 0 | Null | Null | 被叫号码 |
| 4 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 5 | AgentID | Char | 10 | 0 | NULL | NULL | 坐席工号 |
| 6 | ExtensionID | Char | 10 | 0 | NULL | NULL | 分机号 |
| 7 | EndTime | DateTime | 8 | 0 | Not Null |  | 挂机时间 |
| 8 | ChannelID | Int | 4 | 0 | Null | Null | 录音通道号 |
| 9 | TrunkID | Int | 4 | 0 | Null | Null | 录音设备号 |
| 10 | RecordSessionID | GUID | 64 | 0 | Not Null |  | 录音全局标识 |
| 11 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 会议明细记录表（ConferenceDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | ConferenceID | Char | 10 | 0 | Not Null | Null | 会议号码 |
| 3 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 4 | ConferenceSessionID | GUID | 64 | 0 | Not Null |  | 会议唯一标识 |
| 5 | EndTime | DateTime | 8 | 0 | Not Null |  | 挂机时间 |
| 6 | RecordingSessionID | GUID | 64 | 0 | NuLL |  | 录音ID |

## 数据库结构图

数据库中的各表主要关系如下：

