



(12) 发明专利

(10) 授权公告号 CN 101141742 B

(45) 授权公告日 2011.08.10

(21) 申请号 200710164043.3

第 7 页第 2 行至第 13 行, 图 5.

(22) 申请日 2007.10.12

CN 1859408 A, 2006.11.08, 全文.

CN 1954293 A, 2007.04.25, 全文.

CN 1859325 A, 2006.11.08, 全文.

(73) 专利权人 中兴通讯股份有限公司

地址 518057 广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦法律部

审查员 李祎

(72) 发明人 王泽民 杨新 鞠飞 古幼鹏

(74) 专利代理机构 北京安信方达知识产权代理有限公司 11262

代理人 龙洪 霍育栋

(51) Int. Cl.

H04W 4/14 (2009.01)

H04W 28/06 (2009.01)

H04W 60/00 (2009.01)

(56) 对比文件

CN 1811717 A, 2006.08.02, 全文.

CN 1713151 A, 2005.12.28, 说明书第 1 页第 19 行至第 21 行, 第 4 页第 4 行至第 16 行, 说明书

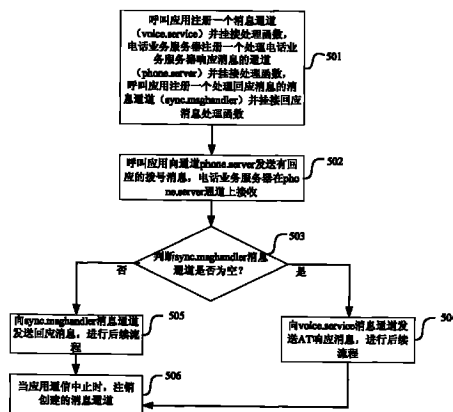
权利要求书 1 页 说明书 6 页 附图 4 页

(54) 发明名称

一种终端的应用通信方法

(57) 摘要

本发明公开了一种终端的应用通信方法, 包括如下步骤: (1) 需要进行应用通信的应用在所述终端操作系统中注册一消息通道, 并将所述应用的消息处理模块挂接在所述消息通道上; (2) 当另一应用向所述应用传输消息时, 将该消息发给该应用所注册的消息通道; (3) 所述消息通道收到所述消息后, 把该消息交给所述挂接的消息处理模块进行处理, 从而完成应用通信。本发明可以避免目前技术方案存在的缺点和不完善, 简单且高效地解决了现有技术应用通信方法在高、中、低端终端不通用的问题。



1. 一种终端的应用通信方法,其特征在于,包括如下步骤:

(1) 需要进行应用通信的应用在所述终端操作系统中注册一消息通道,并将所述应用的消息处理模块挂接在所述消息通道上;

(2) 当另一应用向所述应用传输消息时,将该消息发给该应用所注册的消息通道;

(3) 所述消息通道收到所述消息后,把该消息交给所述挂接的消息处理模块进行处理,从而完成应用通信;

其中,步骤(1)在注册所述消息通道时,对所述终端操作系统的应用进程间通信协议进行封装来创建一个应用通信消息处理对象,该对象中包括能够及时获取到达消息的信号和消息内容的所述消息通道以及对该通道的消息进行处理的所述消息处理模块;

其中,步骤(1)通过以下方式将所述消息处理模块挂接在所述消息通道中:在所述应用的主循环中挂接一个监控该应用通信消息处理对象的监控函数,同时将消息处理模块对应的回调函数指针作为该监控函数的输入参数保存起来。

2. 如权利要求1所述方法,其特征在于,所述方法还包括如下步骤:

(4) 当所述终端的应用通信中止时,注销所述应用注册的消息通道。

3. 如权利要求1所述方法,其特征在于,步骤(3)中,所述消息通道收到所述消息后,按照先进先出的原则把该消息交给所述挂接的消息处理模块进行处理。

4. 如权利要求1所述方法,其特征在于,步骤(1)中,所述消息通道通过消息通道名称字符串来标识。

5. 如权利要求1所述方法,其特征在于,步骤(1)中,将所述应用通信消息处理对象作为一个节点挂接在一个全局链表中,且该应用通信消息处理对象用消息通道名称来标识。

6. 如权利要求5所述方法,其特征在于,步骤(2)中,当另一应用向所述应用传输消息时,所述应用通过该应用注册的所述消息通道名称在所述全局链表中查找到对应的应用通信消息处理对象,然后将该消息直接写入到该应用通信消息处理对象的消息通道中,从而完成消息的发送。

7. 如权利要求6所述方法,其特征在于,当所述应用在主循环中检测到该应用通信消息处理对象的消息通道上有消息写入时,将会自动调用所述监控函数,而该监控函数会将在该通道上读取的消息交给它保存的输入参数对应的消息处理模块来进行处理。

8. 如权利要求1所述方法,其特征在于,步骤(2)中,所述另一应用需要结合所述终端的应用通信的需求提供所述发送消息的无回应发送和有回应发送,对所述消息的有回应发送,发送所述消息的所述另一应用采用建一个消息通道来专门处理回应消息。

9. 如权利要求1~8任一所述方法,其特征在于,所述需要进行应用通信的应用为电话业务服务器,所述另一应用为呼叫应用、或短消息应用、或电话本应用。

一种终端的应用通信方法

技术领域

[0001] 本发明涉及终端软件领域,尤其涉及一种终端的应用通信方法。

背景技术

[0002] 手机终端设备是一种嵌入式软件设备,同时,区别于通用的嵌入式设备,它具有良好的人机交互性,满足了人对终端的功能、样式、使用习惯等个性化的要求;而且随着手机终端的迅猛发展和使用的日益广泛,需要一个手机终端能够完成尽可能多的任务,以方便人们的工作和生活。因此,手机终端设备在内部的嵌入式软件实现中不可避免的存在多个任务,多个进程,而这些多任务多进程之间的应用通信的效率在很大程度上决定了手机终端的性能高低,所以手机终端中的应用通信的简洁、高效是手机终端软件开发中的一个关键问题。

[0003] 应用通信为不同应用程序间通信提供了基本通信功能,不管这些通信的应用是在同一线程(或进程)内部还是在不同的线程(或进程)之间。最基本的应用通信形式是操作系统提供的 IPC(InterProcess Communications protocol,进程间通信协议),但是操作系统提供的 IPC 形式过于原始,功能过于简单,一般只在编写比较低层的软件系统使用比较适合。

[0004] 因此,在手机开发中,不可避免的需要解决由于操作系统本身的 IPC 不易用且无法直接使用的缺陷,大多数手机开发者都选择了在操作系统 IPC 之上实现自己的应用通信,除开那些移植困难,在其他手机上复用需要耗费大量移植工作的应用通讯机制和方法,目前,在手机开发中比较有影响的应用通信方法有 Trolltech 公司 QTOPIA 软件产品中的 Qt Cop(QCOP) 通信协议和开源的消息总线系统 D-Bus 通信机制。

[0005] QCop 通信机制采用了 publish/subscribe 模式,使通信具有很好的扩展性;它用于不同的客户之间在同一地址空间内部或者不同的进程之间的通信。消息的发起者只需要创建一个 QCopEnvelope 对象,其中包含通道名称、消息名称、参数等;消息的接收者要想接收消息,必须首先创建一个 QCopChannel 对象,想接收消息的通道标明了同消息发送者之间的对应关系;同时,依托于其内部特有的函数回调机制——信号-槽机制, QCop 通信机制很好的解决了接收消息和对接收消息的处理关系。因此对于 Qcop 通信机制,通信参与者只要知道了 Qcop 的消息通道名称,任意发送者可以发送任意消息给任意的接收者,使通信具有很好的灵活性和扩展性。但是,也应该看到, QCop 通信机制其设计的出发点是针对高端智能手机,因而在结构的复杂性、运行机制上面并不是很适合中低端手机;同时,其面向对象的设计思想移植到面向过程的设计存在一定的困难,而内部引入的信号-槽机制使其内部耦合度高;对手机应用通信而言, QCop 通信机制最大的缺陷是不支持消息的同步发送,为开发者带来了额外的开发难度。

[0006] D-Bus 应用通信是个消息总线系统,它为应用程序间的通信提供了一个简单的途径,该消息总线是建立在普通的一对一消息传输的框架之上。

[0007] 虽然,进程通讯最简单的方式是一对一的进程通信,但是由于 D-BUS 本身是基于

总线的。所以，D-BUS 却提供了一个后台服务程序 (demon, 也称为消息总线后台程序)，它在一个特定的总线上来路由消息，从而形成了一个总线拓扑结构，它使得进程可以在同一时间来与一个或多个应用程序通信，应用程序可以在总线上发送或侦听各种事件。可见，对比 QCop 通信机制，D-BUS 虽然支持消息的同步发送，但它的实现机制是异常复杂的，且需要在系统中专门启动一个后台消息总线服务器，因此比较适合于高端智能手机框架的应用间通信。

[0008] 综上所述，QCop 通信协议和 D-Bus 应用通信对中低端手机来说，都显得结构庞大，需要耗费大量的手机资源；对于包括了高端智能手机和中低端手机的手机开发来说，就显示出了不能通用的问题。在和上述手机终端功能类似的其他高、中、低端终端上也存在不能通用的问题。

发明内容

[0009] 本发明提供一种终端的应用通信方法，以解决现有技术方案中存在的通信方法在高、中、低端终端不通用的问题。

[0010] 为了解决上述技术问题，本发明提供一种终端的应用通信方法，包括如下步骤：

[0011] (1) 需要进行应用通信的应用在所述终端操作系统中注册一消息通道，并将所述应用的消息处理模块挂接在所述消息通道上；

[0012] (2) 当另一应用向所述应用传输消息时，将该消息发给该应用所注册的消息通道；

[0013] (3) 所述消息通道收到所述消息后，把该消息交给所述挂接的消息处理模块进行处理，从而完成应用通信。

[0014] 本发明所述方法，其中，所述方法还包括如下步骤：

[0015] (4) 当所述终端的应用通信中止时，注销所述应用注册的消息通道。

[0016] 本发明所述方法，其中，步骤 (3) 中，所述消息通道收到所述消息后，按照先进先出的原则把该消息交给所述挂接的消息处理模块进行处理。

[0017] 本发明所述方法，其中，步骤 (1) 中，所述消息通道通过消息通道名称字符串来标识。

[0018] 本发明所述方法，其中，步骤 (1) 在注册所述消息通道时，对所述终端操作系统的应用进程间通信协议进行封装来创建一个应用通信消息处理对象，该对象中包括能够及时获取到达消息的信号和消息内容的所述消息通道以及对该通道的消息进行处理的所述消息处理模块，将所述应用通信消息处理对象作为一个节点挂接在一个全局链表中，且该应用通信消息处理对象用消息通道名称来标识。

[0019] 进一步地，步骤 (1) 通过以下方式将所述消息处理模块挂接在所述消息通道中：在所述应用的主循环中挂接一个监控该应用通信消息处理对象的监控函数，同时将消息处理模块对应的回调函数指针作为该监控函数的输入参数保存起来。

[0020] 进一步地，步骤 (2) 中，当另一应用向所述应用传输消息时，所述应用通过该应用注册的所述消息通道名称在所述全局链表中查找到对应的应用通信消息处理对象，然后将该消息直接写入到该应用通信消息处理对象的消息通道中，从而完成消息的发送。

[0021] 进一步地，当所述应用在主循环中检测到该应用通信消息处理对象的消息通道上

有消息写入时,将会自动调用所述监控函数,而该监控函数会将在该通道上读取的消息交给它保存的输入参数对应的消息处理模块来进行处理。

[0022] 本发明所述方法,其中,步骤(2)中,所述另一应用需要结合所述终端的应用通信的需求提供所述发送消息的无回应发送和有回应发送,对所述消息的有回应发送,发送所述消息的所述另一应用采用建一个消息通道来专门处理回应消息。

[0023] 本发明所述方法,其中,所述需要进行应用通信的应用为电话业务服务器,所述另一应用为呼叫应用、或短消息应用、或电话本应用。

[0024] 与现有技术相比,本发明应用通信采用基于通道的方式进行,发送者把消息发送到通道上,接收者在通道上获取发送者发送到通道上的消息,消息通道通过通道名称字符串来标识,同时在手机终端的嵌入式软件中的,在对各个应用的应用通信处理的服务器守护进程中实现了消息派发功能,避免了类似于D-BUS等常用通讯机制专用的后台通讯服务器;从而可以避免目前技术方案存在的缺点和不完善,简单且高效地实现了高、中、低端终端的应用通信。

附图说明

[0025] 图1为本发明实施例消息通道注册流程图;

[0026] 图2为本发明实施例消息发送处理流程图;

[0027] 图3为本发明实施例接收消息处理流程图;

[0028] 图4为无线应用接口模块消息层结构示意图;

[0029] 图5为本发明应用实例呼叫应用与电话业务服务器的消息流程图;

[0030] 图6为本发明应用实例电话业务服务器消息处理示意图。

具体实施方式

[0031] 以下结合附图和具体实施方式对本发明所述技术方案进行详细描述。

[0032] 本发明所述一种终端的应用通信方法中,所述方法的典型应用终端为手机终端。

[0033] 本发明实施例一种手机终端的应用通信方法,包括如下步骤:

[0034] 步骤1,需要进行应用通信的应用在所述手机终端操作系统中注册一消息通道,并将该应用的消息处理模块挂接在所述消息通道上,以便消息通道接收到消息后,触发消息处理模块对消息进行处理;

[0035] 步骤2,当另一应用向所述应用发送消息时,将该消息发给步骤1中该应用所注册的消息通道;

[0036] 步骤3,消息通道收到消息后,把该消息交给步骤1中挂接的消息处理模块进行处理,从而完成两个应用的应用通信;

[0037] 步骤4,当应用通信中止时,释放该应用通信的上下文结构,注销应用所注册的消息通道。

[0038] 此处释放资源,别的应用不会再发送消息,即使发送也不会被处理,如果别的应用还要发送消息,就不会执行步骤4,所以,如果这时发送消息会告警,提示发送失败。一般来说,步骤1是手机开机时进行,步骤4是手机关机时进行。

[0039] 其中,步骤1中,在手机终端操作系统中注册一消息通道,需要基于手机系统中的

底层机制来实现,对操作系统本身的 IPC,比如管道 (pipe)、套接字 (socket)、消息队列等进行封装来创建一个对象,称为应用通信消息处理对象 :GMsgHandler,该对象中包括能够及时获取到达消息的信号和消息内容的消息通道 (步骤 1 中注册的) 以及对该通道的消息进行处理的所述消息处理模块,将所述应用通信消息处理对象作为一个节点挂接在一个全局链表中 (只所以需要保存在全局链表中,主要是因为具体应用 (注册时) 和应用主循环都需要操作该链表;当然实现方法很多,全局变量某种意义上是开辟一段共享内存供具体应用和应用主循环来操作),且该应用通信消息处理对象用消息通道名称来标识。步骤 (1) 通过以下方式将所述消息处理模块挂接在所述消息通道中:在应用的主循环中挂接一个监控该对象的监控函数,以用于该 GMsgHandler 对象在接收到消息到达信号后进行调用,同时将消息处理模块对应的回调函数 (又叫钩子函数) 指针作为该监控函数的输入参数保存起来。这样,当应用在主循环中检测到该对象有数据输入时 (该对象得到一个消息到达的信号,即得知通道上收到消息,有数据输入),将会自动调用该监控函数,而监控函数会将在该通道上读取的消息交给它保存的输入参数对应的消息处理模块来进行处理。

[0040] 如图 1 所示,本发明实施例消息通道注册流程,包括如下步骤:

[0041] 步骤 101,消息通道注册开始;

[0042] 步骤 102,需要进行应用通信的应用判断该通道是否已在 GMsgHandler 全局链表中注册:如是,则转步骤 104;否则,继续;

[0043] 步骤 103,需要进行应用通信的手机终端应用在 GMsgHandler 链表中注册该消息通道以及相应的处理函数 (这个处理函数是对收到的消息进行处理的函数,即消息处理模块运行的函数);

[0044] 步骤 104,消息通道注册结束。

[0045] 其中,步骤 2 中,消息通道通过通道名称字符串来标识,消息通道名称在整个系统范围内具有唯一性,因此,消息发送通过该应用注册的消息通道名在全局链表中查找到对应的 GMsgHandler 对象,然后将该消息直接写入到该 GMsgHandler 对象的通道中,从而完成消息的发送。其中,消息通道的实现依赖于底层的 IPC,比如对应于消息队列机制来说,消息通道就是一个消息队列,消息通道名称就是消息队列的名称,唯一标识该消息队列,把消息写入到 GMsgHandler 对象的通道中就是把消息拷贝到其对应的消息队列中。

[0046] 如图 2 所示,本发明实施例消息发送处理流程,包括如下步骤:

[0047] 步骤 201,消息发送开始;

[0048] 步骤 202,发送消息的应用判断对应的消息通道名是否为空:如是,则转步骤 205;如否,则继续;

[0049] 步骤 203,发送消息的应用判断消息通道在 GMsgHandler 链表中是否注册:如否,则转步骤 205;如是,则继续;

[0050] 步骤 204,通过底层 IPC 在对应的消息通道上发送消息,流程结束;

[0051] 步骤 205,消息发送失败。

[0052] 如图 3 所示,本发明实施例接收消息处理流程,包括如下步骤:

[0053] 步骤 301,底层 IPC 接收到消息;

[0054] 步骤 302,将消息放入相应的消息通道中;

[0055] 步骤 303,应用在主循环中检测到消息;

[0056] 步骤 304, 在 GMsgHandler 链表中查找该通道挂接的处理函数;

[0057] 步骤 305, 消息处理模块对该消息执行查找到的处理函数。

[0058] 其中, 步骤 2 中, 需要结合手机的应用通信的需求提供消息的无回应发送和有回应发送 (如语音呼叫向电话服务器发送一个拨号请求后, 电话服务器会话会向语音呼叫应用返回一个呼叫标识), 无回应发送类似于消息的异步发送, 将消息发送到消息通道后就可继续执行其它步骤, 因而很容易实现; 而有回应发送类似于消息的同步发送, 是将消息发送到消息通道后一直等待该应用的消息处理函数被执行为止才可继续执行其它步骤。考虑到在中低端手机的单任务框架中, 如果采用单纯的阻塞方式会造成系统的死循环, 在应用通信的发送时一般采用新建一个通道来专门处理回应消息 (重复一个步骤 1、步骤 2、步骤 3 的过程), 从而不会影响系统的运行效率。

[0059] 其中, 步骤 3 中, 当该应用在主循环中检测到该 GMsgHandler 对象上有消息写入时, 调用所述挂接的监控函数对所述消息进行处理。

[0060] 以下为一个本发明实际应用的具体实例, 考虑到手机无线应用模块同上层应用交互, 需要处理大量的应用通信, 故下面结合手机无线应用接口模块的应用通信进行说明。

[0061] 如图 4 所示, 无线应用接口模块消息层结构示意图, 无线应用接口模块对应于图中的接口层、电话业务服务器、通讯处理层, 电话业务服务器通过通讯处理模块 (比如 AT 命令 (AT 命令是调制解调器 Modem 可以识别并执行的命令)) 同无线模块进行交互。无线应用接口模块位于手机操作系统与应用程序之间, 起到无线应用接口的作用。无线应用接口模块主要提供接收应用发来的无线通讯请求, 接收从无线模块发来的命令响应, 接收从无线模块发来的通知消息, 保证收发命令的实时性, 命令的优先级控制, 命令的并行处理, 命令的异常处理, 无线模块数据缓冲和建立数据呼叫链路连接等。无线应用接口模块需要同电话本应用、短消息应用、呼叫应用等进行应用通信。

[0062] 如图 5 所示, 为本发明应用实例呼叫应用与电话业务服务器的消息流程, 包括如下步骤:

[0063] 步骤 501, 需要进行应用通信的应用在操作系统中注册一个消息通道, 并挂接消息处理模块到该消息通道上;

[0064] 呼叫应用要同电话业务服务器进行应用通信, 就需要在操作系统注册一个消息通道, 比如叫做 voice.service (使用字符串进行描述) 通道, 接收电话业务服务器发送来的消息, 挂接处理接收到拨号成功 / 失败的消息的处理函数 (比如拨号成功时进行通话, 拨号失败时在手机界面向用户显示拨号失败);

[0065] 电话业务服务器要接收来自呼叫应用的消息, 也需要注册一个消息通道 (如图 6 所示), 比如叫做 phone.server 通道, 挂接对拨号消息的处理函数 (把拨号命令转换为响应的 AT 命令发送到 Modem, 向呼叫应用回应该次呼叫的呼叫标识, 在接收到 AT 命令的响应后向呼叫应用回应拨号成功 / 失败消息等)。

[0066] 同时因为呼叫应用需要电话业务服务器响应一个呼叫标识, 所以, 需要使用有响应的消息发送, 因此特地注册一个专门处理响应消息的消息通道 sync.msghandler, 并且挂接响应消息的处理函数, 此处对响应消息的处理是在内部数据结构中保存呼叫标识以及对应的呼叫上下文数据。

[0067] 在手机终端操作系统中注册消息通道, 需要在应用通信消息处理对象

(GMsgHandler) 的链表结构中保存该消息通道名称和相应的处理函数。至于接收消息的触发可以采取诸如轮询或是底层驱动来触发（这里指的是触发底层的接收，底层接收后才能主循环中检测到，主循环是通过底层的事件机制（比如有消息抵达）来驱动的）。这样一旦有消息抵达，会触发 IPC 对消息进行接收，消息的派发（通过上文的监控函数实现）则根据消息通道名称在 GMsgHandler 的链表结构中查找对应的处理函数，并且调用该处理函数进行运行。

[0068] 其中，需要说明的是对有回应消息的发送和接收。由于对中低端手机的单任务框架来说，不可能采用阻塞回应的方式，以避免整个系统运行低效。可以采用一个专门的处理回应消息的消息通道的方法来解决这个问题。这就需要额外注册一个消息通道 `sync.msghandler`，有回应的消息发送时，在发送消息体中记录 `sync.msghandler` 通道的名称，一旦消息接收方解析消息发现 `sync.msghandler` 通道的名称不为空，表明该消息发送方需要回应，则回应消息通过 `sync.msghandler` 消息通道发送；正常的响应消息（如本实例中的 AT 命令响应等）还是通过和无回应消息一样的 `voice.service` 消息通道进行发送。

[0069] 步骤 502，消息的发送是通过在 GMsgHandler 对象链表中查找对应的消息通道，然后通过操作系统的 IPC 进行发送的：呼叫应用向通道 `phone.server` 发送有回应的拨号消息，电话业务服务器在 `phone.server` 通道上接收该拨号消息，电话业务服务器在 `phone.server` 通道上接收到拨号消息后，会去调用注册的对呼叫的处理函数处理（如图 6 所示）；

[0070] 步骤 503，电话业务服务器判断 `sync.msghandler` 消息通道是否为空：如否，则转步骤 505；如是，则继续；

[0071] 步骤 504，电话业务服务器向 `voice.service` 消息通道发送 AT 响应消息，进行后续流程，然后转至步骤 506；

[0072] 步骤 505，电话业务服务器向 `sync.msghandler` 消息通道发送回应消息，进行后续流程；

[0073] 步骤 506，当此呼叫应用通信中止时，注销步骤 501 中创建的消息通道以及相应的内存资源。

[0074] 上述步骤 504 中电话业务服务器发送 AT 响应消息，与步骤 502 中电话业务服务器在 `phone.server` 通道上接收到拨号消息后，去调用注册的对呼叫的处理函数处理，这两步没有固定的时间先后顺序。

[0075] 可见，上述实例简洁方便的实现了呼叫应用同电话业务服务器之间的应用通信，有效避免了目前现有技术方法中存在的通信方法在高、中、低端手机终端不通用的缺陷，保证了在各种手机中快速高效的移植应用通信，从而从一个方面缩短了手机软件的开发周期和开发成本。

[0076] 本发明所述方案，并不仅仅限于说明书和实施方式中所列运用。对本发明技术所属领域的普通技术人员来说，可根据本发明作出各种相应的改变和变形，而所有这些相应的改变和变形都属于本发明权利要求的保护范围。

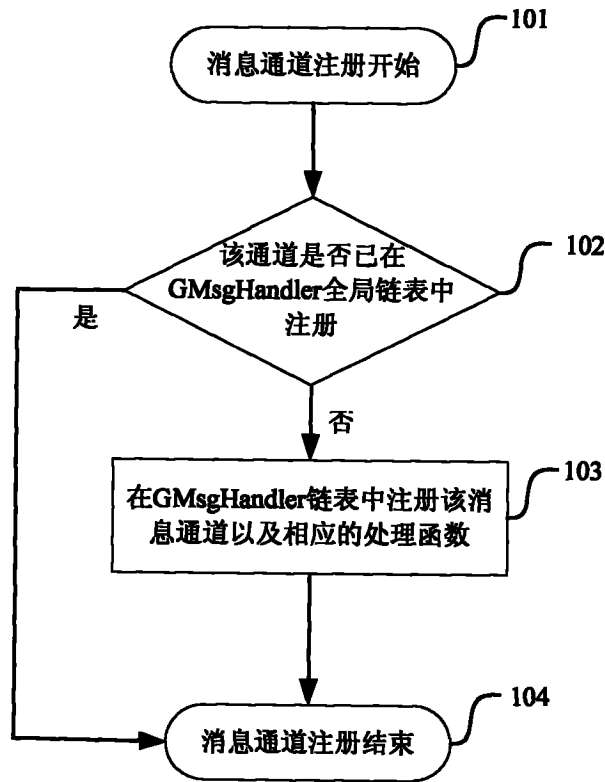


图 1

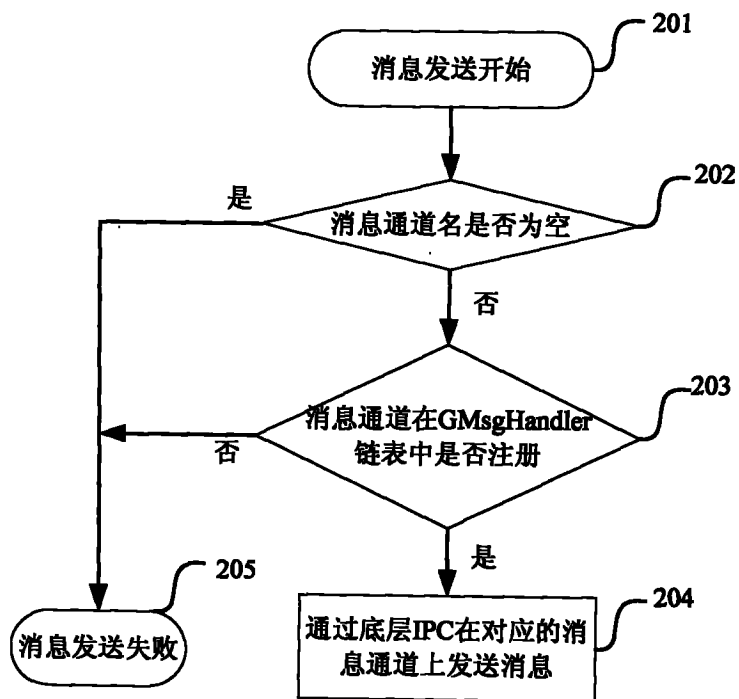


图 2

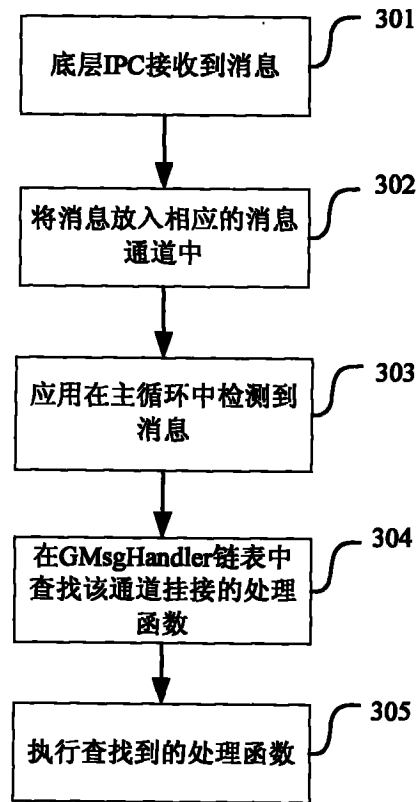


图 3

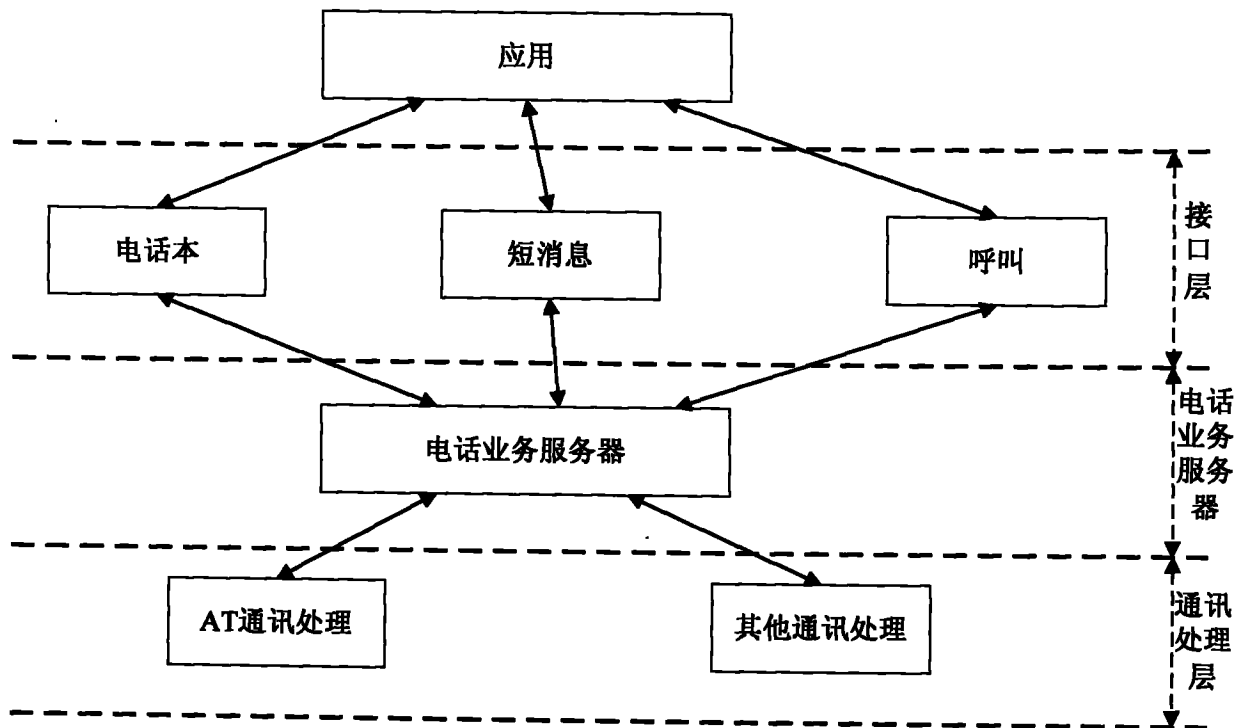


图 4

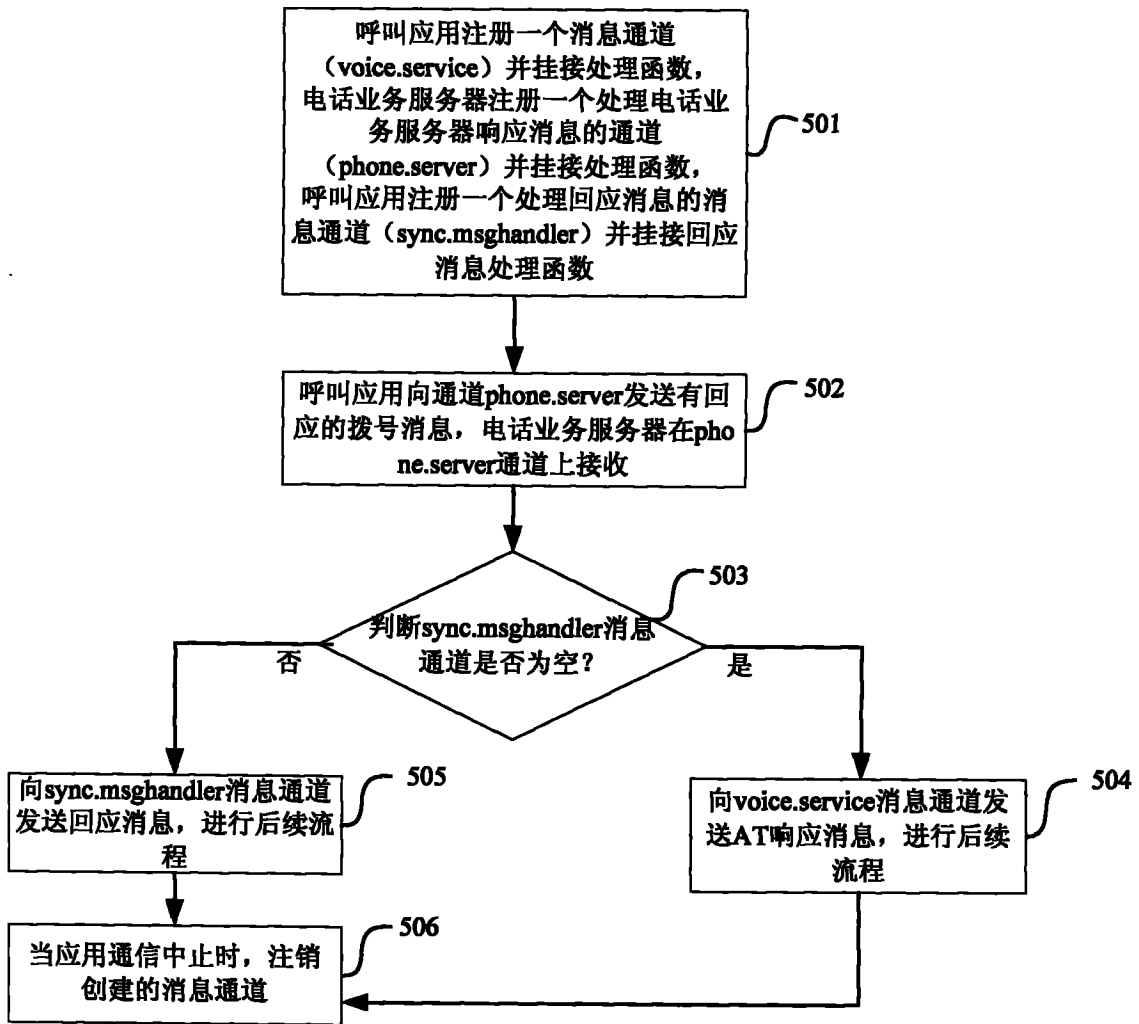


图 5

GMsgHandler对象链表

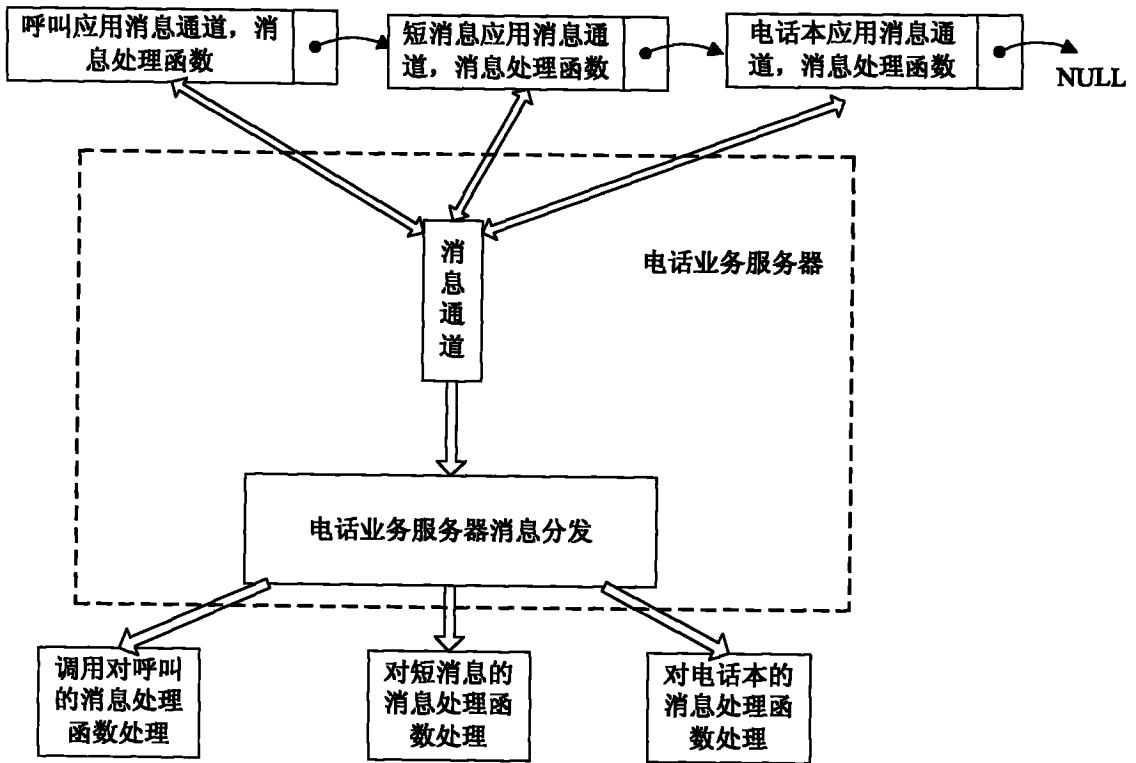


图 6