[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H04Q 7/32

[12] 发明专利说明书

[21] ZL 专利号 98812985.X

[45] 授权公告日 2004 年 8 月 4 日

[11] 授权公告号 CN 1160982C

[22] 申请日 1998.11.5 [21] 申请号 98812985.X
[30] 优先权
[32] 1997.11.7 [33] US [31] 08/965670
[86] 国际申请 PCT/EP1998/007207 1998.11.5
[87] 国际公布 WO1999/025141 英 1999.5.20
[85] 进入国家阶段日期 2000.7.7
[71] 专利权人 诺基亚流动电话有限公司
地址 芬兰埃斯波
[72] 发明人 M·韦约拉 A·阿拉尼米
A·图尔纳
审查员 张 慧

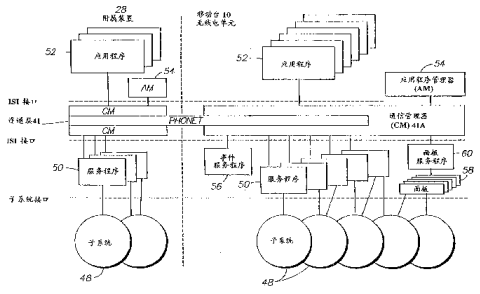
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 邹光新 张志醒

权利要求书 4 页 说明书 36 页 附图 15 页

[54] 发明名称 通信系统及其所用的方法

[57] 摘要

一种通信系统，有一个无线电单元，该单元包括多个应用程序、每一个都跟无线电单元的一个资源有关的多个服务程序以及位于这多个应用程序和多个服务程序之间有选择地将应用程序跟服务程序连接的一个连通层。该系统还包括至少一个附属装置，通过连通层跟无线电单元连接。这至少一个附属装置拥有至少一个应用程序和跟所述附属装置的资源有关的至少一个服务程序。该无线电单元装置的至少一个应用程序或者服务程序可以响应来自无线电单元的至少一个应用程序或者服务程序的资源请求消息，通过连通层跟附属装置的至少一个服务程序或者无线电单元的一个服务程序连接。这一资源请求消息是通过所述连通层传递的。



ISSN1008-4274

1.一种通信系统，包括一个无线电单元装置（10），该单元装置包括一个无线电收发信机（12、14、16）、多个对象（52）和多条链路（41、18A、18B、18C、42、43A、43B、43C），其中至少一个用于在所述对象之间传递消息，其特征在于，所述无线电单元装置还包括一个与所述对象连接的连通层（41），该连通层包含一个消息路由器（40）和一个连接层（42），所述路由器用于从对象经过连接层（42）接收注册请求，为发出请求的对象注册以及保存描述已注册对象的信息。

2.权利要求1的通信系统，其特征还在于其中的消息包括该消息的目标地逻辑地址，所述路由器响应收到的消息，将逻辑地址变换成至少一个物理地址，通过一条链路将所述消息传递给这至少一个物理地址。

3.权利要求1的通信系统，其特征还在于其中的消息包括所述消息的目的地的功能描述，所述路由器响应收到的消息，将这一功能描述变换成至少一个物理地址，并通过一条链路将这一消息传递给这至少一个物理地址。

4.权利要求1的通信系统，其特征还在于其中的路由器响应建立连接的一个请求，利用所述多条链路中的一条，在所述多个对象中的两个之间建立一条直接的通信管道。

5.权利要求1的通信系统，还包括至少一个外部装置（28、60、62、64、68），该外部装置包括通过所述多条链路中的至少一条跟所述无线电单元装置连接的至少一个外部应用程序（52），其特征还在于所述消息路由器能够从至少一个外部装置和至少一个外部应用程序接收注册请求消息。

6.权利要求5的通信系统，其特征还在于所述消息路由器还能记录注册请求指定的优选链路的至少一个身份，或者收到注册请求所经过的链路的身份。

7.权利要求1的通信系统，其特征还在于其中至少有一种注册请求包括注册应用程序的一个链路首选项。

8.权利要求5或7的通信系统，其特征还在于所述外部装置还包括一个消息子路由器（40'）。

9.权利要求5的通信系统，其特征还在于一个消息是资源请求消息，用于说明需要访问所述无线电单元装置或者所述外部装置的资源，其中

的另一个消息是响应消息，从负责资源的服务程序（5）发往资源请求消息的始发方。

10.权利要求 9 的通信系统，其特征还在于其中的一个消息是指示消息，从负责资源的服务程序发出，用于说明与所述资源有关的事件或者状态。

11.权利要求 9 的通信系统，其特征还在于其中的一个消息是一个查询请求消息，发自负责所述资源的服务程序，用于请求获得额外的信息，以便完成所述资源请求消息指定的任务。

12.权利要求 11 的通信系统，其特征还在于其中的一个消息是通知响应消息，响应收到的所述查询请求消息，发自所述资源请求消息的始发方。

13.权利要求 9 或它的任意一个附加权利要求的通信系统，其特征还在于其中的资源请求消息是由应用程序或者服务程序之一发出的。

14.权利要求 9 的通信系统，其特征还在于其中的一个消息是指示消息，发自负责所述资源的服务程序，说明资源内发生的事件或状态，其中另一个消息是查询请求消息，发自负责所述资源的服务程序，用于请求获得额外的信息，以便完成所述资源请求消息指定的任务，其中的指示消息和查询请求消息都有一个目的地地址字段，这些地址来自所述资源请求消息的一个地址字段。

15.权利要求 14 的通信系统，其特征还在于其中的指示消息和查询请求消息还包括一个唯一的事务处理标识字段，该字段来自所述资源请求消息的一个唯一事务处理标识字段。

16.权利要求 9 的通信系统，其特征还在于其中的一个消息是指示消息，发自负责所述资源的服务程序，用于说明在所述资源内发生的事件，该指示消息是发送给至少一个目的地的，在此之前这些目的地已经注册了一个请求，以便得知发生了这一事件。

17.权利要求 1 的通信系统，其特征还在于所述无线电单元装置还包括多个应用程序（52）、跟为每个资源提供接口的所述无线电单元装置的资源有关的多个服务程序（50）、以及这多个应用程序和多个服务程序之间用于有选择地将应用程序跟服务程序连接的连通层（41），该系统还包括至少一个附属装置（28），通过所述连通层跟所述无线电单元装置连接，这至少一个附属装置包括至少一个应用程序（52）和跟所述附属装

置的资源（48）有关的至少一个服务程序（50），其中所述无线电单元装置的至少一个应用程序或服务程序可以响应来自所述无线电单元装置的至少一个应用程序或服务程序的资源请求，通过所述连通层跟所述附属装置的至少一个服务程序或者所述无线电单元装置的一个服务程序连接，所述资源请求消息是通过所述连通层传递的。

18.权利要求17的通信系统，其特征还在于其中多个服务程序中至少有一个是一个面板服务程序（60），跟至少一个面板连接，为所述无线电单元装置提供用户接口，所述面板服务程序用于在所述至少一个应用程序和与该至少一个应用程序有关的一个或多个面板之间提供通信。

19.权利要求17的通信系统，其特征还在于该系统包括一个事件服务程序（56），用于提供事件指示给已经注册以接收这些指示的所述应用程序中预定的那些。

20.权利要求17的通信系统，其特征还在于其中多个服务程序中的至少一个是一个系统模式控制器（50A），用于为所述无线电单元装置提供跟多个不同的无线电通信系统中的每一个进行双向连接的能力。

21.权利要求17的通信系统，其特征还在于其中的另一个消息是响应消息，通过连通层从负责所述资源的所述服务程序发往最初发出所述资源请求消息的应用程序或者服务程序。

22.权利要求17的通信系统，其特征还在于其中的另一个消息是指示消息，发自负责所述资源的服务程序，用于说明所述资源的一个事件或者一种状态。

23.权利要求17的通信系统，其特征还在于其中的另一个消息是查询请求消息，发自负责所述资源的服务程序，用于请求获得额外的信息，以便完成所述资源请求消息指定的任务。

24.权利要求23的通信系统，其特征还在于其中的另一个消息是通知响应消息，是在收到所述查询请求消息以后从所述资源请求消息的发出方发出的。

25.权利要求1的通信系统，还包括一个附属装置（28），用于通过无线接口（18C、43A）或者有线接口（18A、43C、18B、43B）中的至少一个跟所述无线电单元装置建立双向连接，该无线电单元装置包括多个资源（48），该附属装置包括至少一个应用程序（52），通过消息连通层（41）跟所述无线或者有线接口连接，用于发送资源请求消息或者资源

请求响应消息中的至少一个给所述无线电单元装置，并通过所述无线电单元装置的所述消息路由器（40），发送给跟某一无线电单元装置资源（48）有关的资源服务程序（50），这里所述附属装置的至少一个应用程序能够使用多个无线电单元装置资源中的至少一个。

26.一种方法，用于包括一个无线电单元的通信系统，该无线电单元能够工作于至少一种无线通信网，该无线电单元包括多个应用程序和多个通信媒介，该方法的特征在于包括以下步骤：

为该无线电单元提供一个主路由器层和所述多个应用程序跟所述通信媒介之间的一个连接层；

通过一个从属路由器层将一个外部应用程序跟所述无线电单元连接；

从所述多个应用程序中的一个接收消息；和

通过所述连接层、至少一个通信媒介和从属路由器层，将从所述主路由器层收到的信息传递给所述外部应用程序。

27.权利要求26的方法，其特征还在于所述主路由器层在收到注册请求消息时，对其作出响应而动态地分配一个地址给发出注册请求的实体。

通信系统及其所用的方法

技术领域

本发明涉及无线电电话，更具体地说，涉及无线电电话或者移动台，例如那些能在蜂窝网内工作并且能跟一个或者多个外围设备连接的无线电电话或者移动台。

背景技术

电话通过智能附件跟个人计算机（PC）和其它装置连接，它的角色已经扩展到了用于一般性的信息交换，因此只将电话用于通话的概念已经变得十分陈旧了。显然，这种发展要求将它实际的支持装置的复杂性从用户眼前掩藏起来，只让用户看见并使用所需要的业务。

部分地由于蜂窝电话和相关市场的爆炸性增长，需要用有线方式或者无线方式，例如红外（IR）连接或者象低电平射频（RF）信号之类的某些其它通信媒介，将无线电话或者无线电电话，也叫做移动台，跟一个或者多个智能外设实际连接起来。WO 97/32439 和 WO 94/24775 是现有技术中连接蜂窝电话这样的通信装置的例子。

尽管目前在这一领域的功能有限，但是本发明人了解的现有方法都不能令人满意地将用户跟支持硬件/软件的复杂性隔离开来。

在为移动台开发硬件、软件和接口以提供新功能或者增强功能方面，也存在问题。众所周知，由于许许多多的原因，要跟新的外部装置连接的时候，没有必要全部甚至没有必要部分地重新设计移动台和/或附属软件的一些部分。

同样，从点到点配置，例如跟一个外部装置连接的一个移动台，到移动台同时跟多个外部装置连接的多点配置，依然没有必要进行修改。

发明内容

因此，本发明的第一个目的和优点是提供象蜂窝电话或者个人通信器这样的移动台，以及一种或者多种外部装置之间的一种改进接口。

本发明的另一个目的和优点是提供一种消息传递路由器和连接功能，通过所选类型的移动台接口将移动台跟一个或者多个外部装置无缝地集成起来。

本发明还有一个目的和优点，那就是提供一种模块化的软件协议，从而能够用不同的装置和应用程序，以及物理传输媒介，而不需要对移

动台的主要协议软件进行大的改动。

本发明的再一个目的和优点是为移动台提供一种多点连接能力，更具体地说就是提供一组配置功能，从而使软件协议更加灵活。

利用本发明的方法和装置，可以解决这些问题和其它问题，达到本发明的这些目的。

本发明公开了一种通信系统，包括一个无线电单元装置，该单元装置包括一个无线电收发信机、多个对象和多条链路，其中至少一个用于在所述对象之间传递消息，其特征在于，所述无线电单元装置还包括一个与所述对象连接的连通层，该连通层包含一个消息路由器和一个连接层，所述路由器用于从对象经过连接层接收注册请求，为发出请求的对象注册以及保存描述已注册对象的信息。

一种消息是资源请求消息，用来说明需要访问无线电单元装置或者外部装置的资源，另一种消息是响应消息，是负责管理资源的服务程序发送给资源请求消息的发出方的。资源请求消息可以由一个应用程序或者一个服务程序发出。这另一种消息是负责资源的服务程序发给资源请求消息的始发方的一个指示消息。该资源请求消息可以由应用程序或者服务程序发送。另一个消息是指示消息，发自负责所述资源的服务程序，用于说明该资源的一个事件或者状态。这些消息中还有一种是负责资源管理的服务程序发出的查询请求消息，用来请求获得额外的信息，以便完成资源请求消息指定的任务。再一种消息就是资源请求消息发出方在收到了查询请求消息以后发出的通知响应消息。

本发明还包括一种通信系统，该系统包括能够在至少一种无线通信网中工作的一个无线电单元，该无线电单元包括多个应用程序和多通信媒介，该方法的特征在于：为所述无线电单元提供一个主路由器层和位于所述多个应用程序和所述通信媒介之间的一个连接层；通过一个从属路由器层将一个外部应用程序跟所述无线电单元连接；从所述多个应用程序中的一个接收消息；通过所述连接层、至少一个通信媒介和所述从属路由器层，将收到的消息和所述主路由器层传递给所述外部应用程序。

本发明还涉及包括多个对象和多条链路，在所述对象之间传递消息的一个无线电单元装置，这一无线电单元装置还包括一个消息路由器，用于从这些对象接受注册请求。

本发明还涉及一种附属装置，用于通过至少一个无线接口或者有线

接口跟无线电单元装置建立双向连接，该附属装置包括至少一个应用程序，通过消息路由器层跟所述无线或者有线接口连接，将至少一个资源请求消息或者资源请求响应消息传递给所述无线电单元装置。

本发明还提供了一种方法，用于包括一个无线电单元的通信系统，该无线电单元能够工作于至少一种无线通信网，该无线电单元包括多个应用程序和多个通信媒介，该方法的特征在于包括以下步骤：为该无线电单元提供一个主路由器层和所述多个应用程序跟所述通信媒介之间的一个连接层；通过一个从属路由器层将一个外部应用程序跟所述无线电单元连接；从所述多个应用程序中的一个接收消息；和通过所述连接层、至少一个通信媒介和从属路由器层，将从所述主路由器层收到的信息传递给所述外部应用程序。

附图说明

下面只是通过例子并参考附图来介绍本发明，在这些附图中：

图 1 是按照本发明构成和工作的移动台的框图；

图 2 是图 1 所示移动台的一个立视图，图 2 还进一步说明了移动台通过无线 RF 链路双向连接的一个蜂窝通信系统；

图 3 是按照本发明构成电话和网络、外围主机-对象网络（PhoNet）系统一部分的一个路由器层和一个连接层；

图 4 说明 PhoNet 系统所用逻辑地址的一种格式；

图 5 和图 6 有助于理解 PhoNet 消息管道的工作原理；

图 7 描述的是 PhoNet 主路由器和一条媒体驱动消息链路的一个互连例子，其中的 OS 消息业务被看作一个媒介；

图 8 描述的是装置和主路由器之间用于“呼叫创建”消息的一个消息流例子；

图 9 描述的是从一个对象（应用程序）通过路由器层和连接层，跨过 MCU/DSP 接口（IF），到达请求获得的物理媒介（例如 IR 硬件（HW））的一个消息流例子；

图 10 是一个高层次示意图，它描述的是有一个附属装置跟一个无线电单元连接的 PhoNet 系统的一个软件体系结构；

图 11 是一个高层次示意图，它描述的是有一个无线电单元和个人计算机、外部无线电单元和蜂窝网应用程序这样的多个不同外部装置的 PhoNet 系统的一个物理体系结构；

图 12 是描述 PhoNet 系统任务视图的一个高层次示意图；

图 13 是应用程序和服务程序正在执行同一任务的情况下，一个示例

性的消息序列图；

图 14 是应用程序和服务程序处于同一处理器内，但正在执行不同任务这种情况下的一个示例性消息序列图；

图 15 是应用程序和服务程序位于不同处理器内这种情况下的一个示例性消息序列图；

图 16 是应用程序向电话号码（PND）服务程序发出资源请求这种情况下的一个示例性事务处理图；

图 17 是应用程序向呼叫服务程序发出资源请求的一个示例性事务处理图；

图 18 是应用程序向 PND 服务程序发出资源请求的示例性事务处理图，其中的 PND 认定它必须从要执行的应用程序获得许可；

图 19 是呼叫服务程序发出 Call_Coming_IND 给一个应用程序或者一些应用程序的情况下的示例性事件指示事务处理图，在此之前这一应用程序或者这些应用程序注了册以便从呼叫处理器接收事件指示；

图 20 是一个示例性事件指示事务处理图，它包括一个继后操作，例如发出绑在事件指示上的资源请求的应用程序；

图 21 是一个示例性查询请求事务处理图，其中的服务程序发送一个查询请求给应用程序，而且这一个查询请求不是正在进行的资源请求事务处理的一部分；和

图 22A 和 22B 说明系统模式控制（SMC）服务程序的两种实施方案，其中集成了系统选择规则，以及它跟应用程序和系统服务程序的连接。

具体实施方式

阅读后面对本发明的描述时，以下略语和缩写将非常有用，特别是在描述电话和网络、外围主机-对象网络（PhoNet）时。

AMPS	高级移动电话业务
CS	蜂窝系统
连接层	PhoNet 的较低层，它包括一组物理传输媒介模块，例如红外（IR）、RF 或者总线模块
装置	支持应用程序的物理装置（也就是附件、MS）
DSP	数字信号处理器/处理
ECI	外部控制接口，ISI 和 VAS 的一个通称。包括业务、装置和通常整个连接概念

外部对象	可以用 PhoNet 和某些物理媒介访问的一种对象
FBUS	跟 DSP 连接的快速双线异步消息总线
GSM	全球移动通信系统
主机	包括 PhoNet 主路由器的装置
ISI	智能业务接口
内部对象	同一处理器内可以用 OS 访问的一种对象，例如红外对象
链路	等价于物理传输媒介
MBUS	媒介模块和 PhoNet 的一种物理媒介
MDI	MCU/DSP 接口
媒介模块	PhoNet 的较低级别（lower levels）。媒介模块是为了通过特定的媒介，例如 IR、RF 或者任意总线，传递消息而完成所需操作的模块。媒介模块在 PhoNet 连接层内。
MS	移动台，例如蜂窝电话、PCS 电话、无线电话、个人通信器或者 PC 模块或 PCMCIA 卡那样的卡
对象	外部装置里的应用程序，或者同一处理器内的模块，例如 ISI、CS 或者 UI
OS	操作系统
RPC	远程过程调用
RF	射频
路由器、路由器层	PhoNet 的核心元素
SIM	用户识别模块
从属装置	包括 PhoNet 子路由器的装置
传输媒介	传递 PhoNet 消息的媒介（例如 IR、RF、导线）
UI	用户接口
VAS	增值业务

下面参考图 1 和图 2 来说明无线用户终端或者移动台 10，例如，但不局限于，适合于实践本发明的蜂窝无线电电话或者个人通信器。移动台 10 有一付天线 12，用于发射信号给基站 30 以及从基站接收信号。基站 30 可以是蜂窝网的一部分，该蜂窝网有包括一个移动交换中心

（MSC）34 的一个基站/移动交换中心/互通功能（BMI）32。当移动台 10 正在通话时，MSC 34 用于跟陆线干线连接。

该移动台包括一个调制器（MOD）14A、一个发射机 14、一个接收机 16、一个解调器（DEMOD）16A 和一个提供信号给发射机 14 并从接收机 16 接收信号的控制器 18。这些信号包括符合可用蜂窝系统或其它系统空中接口标准的信令信息，还包括用户语音和/或用户数据。空中接口标准规范对于理解和实现本发明来说无关紧要，因此不做进一步介绍。

显然，控制器 18 还可以包括实现移动台音频和逻辑功能的电路。举例来说，控制器 18 可以包括主控制单元（MCU）21，一般而言它是一个微处理器装置，还可以包括一个数字信号处理器（DSP）23，通过一个 MCU/DSP 接口（IF）跟另一个数字信号处理器通信。控制器 18 还可以包括各种模数转换器、数模转换器和其它的支持电路。移动台 10 的控制和信号处理功能按照它们的能力分布在这些装置里。

为了跟外部装置例如图 2 所示的附属装置 28 接口，还提供了多条总线，例如跟 MCU 21 连接的异步通信总线（MBUS）18A、跟 DSP 23 连接的快速异步通信总线（FBUS）18B，以及一条无线总线 18C，例如红外（IR）或者低功率 RF 链路。附属装置 28 的例子有，但不限于，个人计算机、个人通信器、外部无线电单元和录音机、寻呼机这样的装置、外部手机和车用附件。可以有一个以上的附属装置 28 跟移动台 10 连接，如下文所示。

用户接口包括普通耳机或者扬声器 17、普通麦克风 19、显示器 20 和用户输入装置，通常是小键盘 22，它们都跟控制器 18 连接。小键盘 22 包括普通数字键（0~9）和相关键（#、*）22a 以及其它键 22b，用于操作移动台 10。这些其它键 22b 可以包括，例如，发送键、各种菜单滚动键和软键以及电源键。移动台 10 还包括一个电池 26，为控制移动台所需要的各种电路供电。

移动台 10 还包括各种存储器，一起用存储器 24 表示，其中储存了移动台工作过程中控制器 18 使用的多个常数和变量。例如，存储器 24 储存了各种蜂窝系统参数的值和号码分配模块（NAM）。控制控制器 18 工作过程的操作程序也储存在存储器 24（一般是 ROM 装置）里。存储器 24 还可能存有数据，包括显示给用户看之前从 BMI 32 收到的用户消

息。还可以提供抽取式 SIM 27 用于储存特定移动台 10 的信息。

显然，移动台 10 可以是车载式的、手持式的，或者是任何其它合适的形式，例如插入个人计算机或其它类型计算机插槽（例如 PCMCIA 插槽）的卡。

还应当明白，移动台 10 能够使用一个或者多个空中接口标准、调制类型和接入类型，例如 TDMA、CDMA 和模拟（FM）。

存储器 24 里的操作程序包括实现 PhoNet 系统的软件，下面将详细介绍。

参考图 3，本发明的 PhoNet 包括两个独立的功能层，叫做路由器层 40 和连接层 42，它也可以叫做连通层 41。连接层 42 包括一个控制模块 42A 和链路专用媒介模块，具体而言就是无线媒介模块 42B、FBUS 媒介模块 42C 和 MBUS 媒介模块 42D。路由器 40 跟控制和媒介模块 42A~42D 可以放在 MCU 21 内，而帧过滤器层 42E 则可以放在 DSP 23 内，它也是连接层的一部分。无线（例如 IR 或者 RF）、FBUS 和 MBUS 硬件（HW）模块 43A~43C 则跟它们的对应总线连接，用于为外部装置提供必要的硬件和物理接口。

图 3 所示的层结构用于保证硬件的独立性，并实现 PhoNet 所需要的模块化。

路由器 40 包括一个路由器模块，它的主要功能是为消息选择路由，提供所连接装置和对象的信息，以及支持更改复杂的配置。路由器层 40 包括用最佳方式将消息传递给不同物理传输媒介模块 43 的逻辑。

连接层 42 的控制模块 42A 被用作从 MDI 到 DSP SW 以发送控制消息的一个门。控制模块 42A 提供一组公共配置功能，它们不是链路专用的，而是所有链路公用的。它还支持用分开的控制消息进行管道传输，如下所述。此外，可以由控制模块 42A 来决定不同装置可能使用的协议过滤器。

媒介模块 42B~42D 将一般的路由器 40 消息转换给所需要的物理传输媒介。给定媒介模块是到有关硬件模块 43 的接口。虽然图 3 中有三种不同的媒介模块，但不一定是三种，根据需要，还可以规定新媒介模块 42 和有关的硬件模块 43。媒介模块 42A~42C 最好还能重发消息并能够确认。每一个还包括任意需要的 I/O 缓冲器，用于储存经过这一链路的消息和数据。

如果需要，帧过滤器模块 42E 将 PhoNet 帧结构格式变换成另一种帧结构格式。这种转换最好由较高速的 DSP 23 用软件来完成。

路由器 40 将提供消息路由选择作为主要业务，可以将它看作 PhoNet 的核心。路由器 40 还提供其它的服务，例如更改配置和查询，并提供所连接装置和对象的信息。

PhoNet 有一个与主机有关的主路由器 40。通常，主路由器 40 位于移动台 10 里。只有主路由器 40 提供完整的消息路由选择业务。路由器 40 对物理传输媒介的使用进行优化。路由器 40 监视物理链路的状态和负载程度，并能选择一种最好（快速）的方式传递给定消息。还可以有几个子路由器或者从属路由器跟系统连接。子路由器一般都有主路由器 40 的一个功能子集。

可以配置和重定规模的路由器 40 的功能包括：

（A）路由器 40 将逻辑地址转换成物理地址。

（B）路由器 40 可以将一个逻辑地址变换成几个物理地址（例如移动台的用户接口可以将一个消息发送给多个应用程序）。

（C）主路由器 40 可以提供路由选择功能。

（D）路由器 40 可以进行默认地址变化/动态配置，以及配置查询。

（E）路由器 40 可以使物理地址（媒介）有不同的特权。

（F）路由器 40 可以在路由器层和连接层 42（例如媒介可用性和状态）之间提供信令机制。

（G）路由器 40 可以在路由器 40 跟装置和对象之间、路由器 40 跟连接层 42 之间以及连接层 42 跟物理媒介 43 之间提供清晰的接口（clear interfaces）。

（H）路由器 40 可以为路由器配置的改变和查询提供一个清晰的接口。

（I）路由器 40 能够储存被连接装置和对象的信息，还支持远程过程调用（RPC），而没有参数个数的限制。

（J）还能够配置路由器 40，为所需接口提供断开信息。

对于主机（例如移动台 10）内的主路由器 40，消息的路由选择基本上包括按照功能要求将一则消息传送给某一逻辑地址，并将逻辑地址转换成包括要使用什么物理链路的信息的物理地址。这些消息可以发送给某些功能而不必知道逻辑地址，或者直接发送给已知的逻辑地址。主路

由器 40 更新不同功能的信息：例如（a）系统可用的链路；（b）被连接的装置和它们的链路；（c）注册进入了的装置的信息和它们的链路首选项；（d）象 ISI、UI 或者 CS 这样的对象任务的路由选择信息；以及（e）活动管道的信息。

更详细地考虑这些功能信息，以及对于系统（a）能获得的链路，这些链路可以有三种状态之一：（1）活动的(active)；（2）繁忙的(busy)（无法使用），和不能使用（not available）。表 1 给出一个示例性的链路状态条件。

表 1

链路	状态
MBUS	活动的
FBUS	活动的
IR	繁忙的
RF	不能使用

考虑到连接的装置和它们的链路（b），主路由器 40 将所有可用装置的信息和当前可以使用的装置的信息储存起来。它还储存了链路负载程度信息，从而为消息传递选择最佳方案，这样，如果加载了一条需要的链路，消息就会传递到另一条可用的链路上去。表 2 说明的是装置状态信息和它们使用的链路信息，以及指向对象表的指针信息。“指针”栏包括指向表 1 的指针，用于获得状态信息。“媒介”栏说明最优媒介（见表 3），或者如果没有给出最优媒介，就说明装置最先注册的媒介。

表 2

装置	媒介	状态	指针				对象指针
01	IR	活动的	MBUS	FBUS		RF	指针 1……
02	IR	活动的	FBUS				指针 2……
03	RF	不活动的	IR	RF			空
04	FBUS	活动的	IR				空

表3给出了跟注册进入对象装置和它们的链路首选（c）有关的信息例子。

表3

对象	链路首选
01	MBUS
02	MBUS
03	FBUS
04	IR
05	没有首选
06	RF
07	MBUS
08	FBUS

至于 ISI、UI 或者 CS 这样的内部对象任务的路由选择信息，表 4 给出了一个配置例子。

表4

功能	地址 1（逻辑地址）	地址 2……（物理地址）
ISI	0001	0203
LOC	0002	
CS		
AUD	708	07

利用说明，可以将给定 PhoNet 的消息分发给移动台 10 的“内部”对象，例如蜂窝系统（CS）模块、用户接口（UI）模块、音频软件模块或者 ISI 模块。既然这些对象不是任何装置所必需的，而且它们不使用任何通信链路，就通过将模块的逻辑地址转换成物理地址，提供了特殊内部对象表（表 4）用于跟这些模块通信。因此表 4 是消息功能指的是内部对象的情况下由路由器 40 使用的。

最后，考虑到活动管道的信息，路由器 40 可以在不同装置或者对象之间建立一条快速消息链路或者管道。管道可以用无线媒介、FBUS 或

者 MBUS 建立。一旦管道成为活动的，两个对象就可以互相通信而不需要路由器 40 参与。也就是说，这两个对象可以传输“纯”数据，而不需要采用路由器能够识别的帧结构。总之，路由器 40 储存了活动管道的信息，控制着活动管道，并为有需要的装置建立一条消息管道（如果可以获得管道）。

表 5 给出了所有 PhoNet 管道当前状态的一张“快照”例子，也就是说，建立管道所需要的装置和链路的地址。

表 5

始发方	对方	链路
0102	0205	FBUS
0104	0204	IR
0201	0703	RF
0701	0803	MBUS

总的来说，有两种不同类型的管道。第一种类型的管道支持从一个附件到另一个附件之间直接又快速的数据流。第二种较慢类型的管道通过链路主机（不一定用 PhoNet 路由器 MCU 21）从一个装置向另一个装置传递数据。

第一种类型的管道能够在例如两个 DSP（例如数字音频）之间快速传输数据。经过管道的通信对于主机来说是不可见的。取而代之，主机在单独一个表里（见图 5 和 6）保存了活动管道的信息。

第二种类型的管道可以用于变换来自接收方没有的一种媒介的数据。例如，只有一条 IR 链路的一个附件可以将数据传输给只有一条 RF 链路或者某种其它总线的附件。可以让消息经过用例如 DSP 23 软件实现的某种协议变换器，从而在传输过程中实现协议变换。

不再使用一条管道时，必须告诉主机该管道已经释放。主机还可以控制和中断给定管道。在任意给定时刻，都有可能有一条以上的活动管道，还有，如图 5 所示，通过同样的媒介拥有一条以上的活动管道。

图 6 说明总的管道体系结构，其中的控制由 MCU 21（路由器 40）进行，实际的管道经过了两个装置（例如装置 3 和 4）之间的 DSP 23。

某一子路由器（位于从属装置内）更新以下信息：它自己的连接状态（注册、进入/退出）；它自己的链路状态（活动/不活动，加载程度等等）；主机的可用/活动链路；以及消息路由选择。跟这些任务有关的基本功能类似于主路由器40的功能，虽然消息的路由选择比较简单，因为它只需要将逻辑地址变换成物理地址。

考虑到路由器40的配置，两个主路由器功能都有默认的和动态路由选择方案：路由选择和逻辑地址到物理地址的变换。默认的路由选择是在软件的编译阶段定下来的，而动态路由选择则可以在运行时刻动态地改变。内部对象的个数最好是固定的，并且在编译时确定下来。

可以用配置查询来获得跟当前默认的或者动态路由选择有关的信息。为了查询配置和更改配置，提供了一个清晰的接口功能。这一接口基本上是一种消息接口（只分配了某些功能用于配置）。可以提供一组业务功能用于构造消息。

可以配置路由器40的某些特征，从而使路由器能够根据系统需要改变规模。例如，跟功能完整的数字蜂窝电话，例如GSM或者IS-136电话相比，单模式AMPS蜂窝电话的路由器功能非常有限。

至于编址方法，本发明的优选实施方案采用了两级地址。逻辑级地址包括一个装置号（例如电话、PC等等）和一个内部对象号（例如UI或者CS）。物理级地址在物理地址上增加了所用媒介的信息。

外部装置的地址空间分成两组。一组是固定装置的地址（也就是有预定地址的装置），而另一组地址则用这里公开的动态注册方法为装置预约。

如果某一装置使用固定地址，那么它就不必在PhoNet主机那里进行实际的注册，因为发送一则消息就足以告诉主机该装置是有连接的，除非该装置希望让主机知道它的首选项之类。使用固定地址就可以使用简单的外部装置，而不需要很高程度的智能。例如，可以将数字音频发送给有一个硬编码固定地址，例如所谓的“免提”单元，的简单的低成本外部从属装置。

在这种情况下的注册信息可以只包括接收方部分里的一个固定主机地址和发送方部分里的一个固定从属装置地址，于是注册响应在接收方部分里包括固定的从属地址并在发送方部分里包括固定的主机地址。

现在考虑采用动态方式注册的装置，动态寻址指的是一些从属装置

没有固定的装置号。装置号是在注册阶段由 PhoNet 主机动态地分配的。主机总是用一个固定的装置号。共用或者广播类型的消息也有一个固定地址，所有相连的装置都接收发往这一固定广播地址的所有消息。

图 4 说明了一种逻辑地址的格式。按照这一编址方案，网络可以连接例如 255 个装置，每一个相连的装置可以有最多 256 个内部对象。每一装置的内部对象号最好在软件的编译阶段就固定。根据本发明的一个方面，每一个装置，以及一个装置里的对象，都可以单独寻址，可以参与通信事务处理。

装置地址可以有三种不同的状态：自由的、活动的或者挂起来的。自由地址可以分配给任意新装置，活动地址是已经分配给某一连接装置的地址，挂起来的地址是等待确认有关装置已经退出的地址。当跟某一装置的通信失败时，分配的地址就被路由器 40 挂起来。如果有关装置“苏醒过来”挂起来的地址就可以变成活动的，或者在运行某种恢复软件以后，或者在预定的超时以后，成为自由的。

路由器 40 和物理媒介（例如 IR 链路）的最佳消息格式用表 6 表示，其中的“b”表示“比特”：

表 6

媒介	8b
接收方装置	8b
发送方装置	8b
功能	8b
长度	16b
数据（0）接收方对象	8b（可选）
数据（1）发送方对象	8b（可选）
.	
.	
.	
数据（n）	8b

其中：

媒介=例如 IR、FBUS、MBUS

接收方=装置地址

发送方=装置地址

功能=消息类型标识符

长度=数据字节数

对象=低级地址（low level address），例如 PC 应用程序地址数据=要传输的数据

装置可以在系统启动的时候注册，无论该装置是什么时候跟系统连接的，无论主路由器 40 是什么时候需要它。注册是用该装置选择的物理传输媒介进行的。如果因为某种原因所选媒介无法使用，就可以尝试使用其它的物理媒介。注册技术可以是完全可配置的，这样就可以选择最佳媒介用于注册，如果最佳媒介无法使用，就可以选择一个优先级以使用其它媒介。

在注册消息里，正在注册的装置采用临时地址，直到主路由器 40 分配了一个真实的装置地址。注册消息包含这一特定装置的可用物理传输媒介的信息。注册消息还包括一个固定部分，例如设备类型，和某种随机生成的部分。随机生成的部分用于在装置试图识别它们自己的注册确认消息时避免冲突。装置还可以给出最佳装置号的建议。如果被请求的装置号是空闲的，主机（主路由器 40）就会分配这一装置号。

更详细地说，从一个装置发送给主机的注册消息包括:

作为接收方的固定主机的地址;

临时装置发送方地址;

装置类型标识符;

随机号码;

如果有就分配的最佳装置号; 和

可用链路的信息。

在注册确认消息里，主路由器 40 给正在注册的装置分配装置号，并告诉这一装置它可以用什么物理媒介。

更详细地说，从主机发往从属（装置）的注册消息包括:

作为接收方的临时地址;

作为发送方的固定主机的地址;

装置类型标识符（来自注册消息）;

随机号码（来自注册消息）;

主机分配的装置号；和

主机内可用链路的信息。

这些装置的外部对象也可以注册进入该系统。如果外部对象想改变例如它的路由，它就象那些装置一样向主机表明身份。该对象还通知主机它想用哪种物理链路。对象注册消息主要只包括最佳链路信息。为了在两个外部对象之间建立管道，它们必须都在主机这里注册。

对装置连接的监视是在 PhoNet 连接层 42 里进行的。当连接层 42 检测到某一装置已经断开时，它就通知 PhoNet 路由器 40。然后，路由器 40 将这一装置和该装置拥有的所有外部对象标为已经断开。首先将该装置的地址改成挂起状态，说明不能将这一地址分配给任何其它装置，直到路由器 40 确信该装置已经真的退出，或者跟 PhoNet 系统断开。当路由器 40 认定该装置已经退出时，就将该地址改成自由状态，可以重新分配给另一个装置。

路由器 40 提供一种业务，以获得任意装置的连接信息。任何对象都可以请求路由器 40 自动地通知对象一个装置或者一些装置是否已断开。对象还可能采用路由器 40 的业务功能来首先确定需要的对象/装置已经断开还是正在连接中。

所有路由器（主路由器和子路由器）都支持将逻辑地址转换成物理地址。一个逻辑地址可以对应几个物理地址。同时提供默认配置和动态配置。

主路由器 40 能够按照功能为消息寻找路由，也就是说，接收方地址可以是未知的。呼叫创建消息就是一个例子，因为呼叫创建消息可以在不知道是在什么地方创建呼叫或者特定时刻哪一个 CS（在多 CS 电话中）是活动的情况下创建。但是，主路由器 40 将这一消息传送给合适的逻辑地址。这一路由选择是完全可以配置的，例如，如果 CS 业务切换到另一个系统，路由就跟着改变。

PhoNet 系统有单独的安全检查模块。如果需要，可以为每一个逻辑地址和选择路由功能提供安全保障。例如，可以用命令锁定一个功能或者一个逻辑地址来保证它的安全，只有用特定的安全操作才能打开这把锁。

连接层 42 为实际的物理链路，例如 Fbus、IR 或者 RF，提供路径。每一种物理媒介都有它自己的低级驱动器，用于跟特定的媒介模块接

口。OS 也被看成一种传输媒介，虽然它跟物理媒介不相似。

连接层 42 包括上述协议变换器模块（也就是帧过滤器 42E），通过它将消息变换成需要的帧结构。这一处理是在将消息传递给媒介模块 43 之前进行的。连接层 42 还提供一种优化消息传递方式，这样，几个 PhoNet 消息就可以打包成一个物理层帧。

总之，连接层 42 有以下特征和优点：

（a）通过物理媒介提供可靠的消息传输；

（b）负责向路由器 40 报告链路丢失/发现信息；

（c）需要的时候将帧映射成较低级的协议；

（d）为路由器 40 提供一种接口，这种接口是固定的，对于所有的较低级媒介模块 43 来说都平等；

（e）使得较低级媒介能够很容易地添加到 PhoNet 系统里去；

（f）使媒介模块 43 能够更新链路功能的诊断（例如，重新发射，误码率，缓冲器状态，等等）；和

（g）提供上述帧过滤器 42E，将 PhoNet 帧结构变换成其它的帧结构，或者需要的话反过来。

图 7 给出了 PhoNet 主路由器 40 跟多个媒介驱动消息路径的一个互连例子，其中的 OS 42F 消息业务被看作一个媒介。

图 8 给出了一个装置和主路有器之间“呼叫创建”消息的消息流例子。在这种情况下，呼叫创建消息通过到主路由器 40（装置 00）的 IR 链路通过了从属装置 03 中的一个子路由器 40'，将呼叫创建消息通过 OS 42F 传递给 CS。在这一例子中，逻辑地址 00.00 有两种物理媒介，IR 是比 FBUS 更好的选择。

图 9 描述的是从一个对象（应用程序）通过路由器层 40 和连接层 42，通过 MCU/DSP 接口（IF），并到达请求获得的物理媒介（例如 IR 硬件 43A），的一个消息流例子。更具体地说，采取了以下步骤：

1. 一个对象，例如 ISI，发送一则要选择路由的消息。

2. PhoNet 路由器 40 定义 -什么地方：装置 + 对象
-什么方式：物理媒介

3. 路由器 40 发送这一消息给完成以下操作从而实现必须的协议的物理媒介模块（例如 IR HW 43A）：

- 缓冲

- 消息编号
- 确认
- 重新传输
- 向路由器 40 发送失败报告
- 以及其它所有的较低级任务

4.媒介模块功能可以在 MCU 21 和 DSP 23 之间分开。总的来说，所有需要的控制都由 MCU 21 完成，DSP 23 可以添加一些额外的信息给帧，但主要功能是“驱动”物理媒介。通常，MCU 21 太慢，难以驱动快速总线。给定媒介模块可以“驱动”一个或者几个不同的物理媒介（例如无线媒介模块 42B 可以通过它们相应的硬件接口驱动 IR 或 RF 链路）。

5.消息是用被请求获得的物理媒介（IR）传输的。

对本发明有了一个总的了解以后，下面将更详细地介绍本发明。

移动台 10 有许多基本能力。包括跟蜂窝（可能还有无绳）系统有关的能力（例如能够发出、接受和管理呼叫），非系统能力（例如能够储存和取出号码簿条目）。移动台 10 包括所有这些基本能力，它们一起用一种相干的方式集成起来，实现电话功能。

智能软件体系结构（ISA）将移动台的基本能力分成逻辑组或者资源。在 ISA 里，移动台的资源由服务程序控制。每一种资源都有一个服务程序。服务程序提供一种接口，使得任意数目的应用程序能够使用这一资源。服务程序的目的是用一个服务程序接口封装它控制的资源。设计和实现服务程序和服务程序接口时没有直接考虑要使用它的应用程序。

应用程序链路将移动台 10 的资源跟实现移动台功能的逻辑组合起来。应用程序通过服务程序可以使用任意数量的资源，实现它们自己的功能。应用程序通过连通层 41（见图 3）发出“资源请求”，对于这一目的，可以认为连通层 41 包括路由器 40 和连接层 42 的功能。连通层 41 利用路由器 40 将“资源请求”消息发送给控制所需资源的服务程序。服务程序执行被请求的操作，然后通过连通层 41 发送适当的响应消息给发出请求的应用程序。连通层 41 提供一种直接的消息传送机制，使得应用程序能够使用移动电话终端的资源，而不必知道这些资源或者控制这些资源的服务程序在哪里以及是如何实现的。应用程序只需知道资源和能

够用它的接口的特性。连通层 41 支持资源的动态配置。这意味着移动台 10 的资源和能力能够动态地改变。如果需要，可以让应用程序知道这些改变。

连通层 41 包括 PhoNet 的协议软件和可以在控制模块 42A 里实现的一个通信管理器。连通层 41 按照 PhoNet 消息格式处理智能业务接口（ISI）消息。这些消息和 PhoNet 并不是受限于某种总线硬件。通信管理器负责将事件分配给内部应用程序、服务程序和面板，并在应用程序之间进行通信。应用程序管理器负责应用程序的注册。面板用于按照所需要的用户接口类型将应用程序产生的信息显示给用户。跟以前的移动台软件体系结构的一个很大的差别是将用户接口软件分成了应用程序和服务程序。可以利用连通层 41 和 ISI 接口从移动台 10 的外部调用服务程序。利用同样的服务程序的应用程序可以利用连通层 41 和 ISI 接口放在无线电模块或者移动台 10 外面。使用相同服务程序的应用程序可以安装在于无线电模块里面或者外面，例如，在 PC 里或者附属装置里。在一种以前的软件体系结构里，MCU SW 被分成子系统，每一个子系统都是有给其它子系统的指定业务的 MCU 软件的自主部分。在 ISA 里，以前的子系统很可能仍然存在，但在应用程序（UI）和子系统之间引入了服务程序。

ISA 的目的包括：将基本资源的控制跟功能控制和逻辑分开来；分开需要和提供业务的实体（应用程序&服务程序）；产生支持服务程序为应用程序提供服务的接口；产生支持改变服务程序而不影响应用程序的接口（去耦）；产生一个接口，保证服务程序不需要知道使用这些业务的应用程序的任何情况，也就是允许引入新的应用程序而不会影响服务程序；以及隐藏应用程序和服务程序的位置，甚至在几个处理器之间（分布）。

在随后的讨论中给出了这一体系结构的几个视图。展开视图（development view）（图 10）主要是介绍软件的结构、它的划分和层。物理视图（physical view）（图 11）介绍的是系统配置。任务视图（图 12）描述的是实体的 OS 任务分配。接下来提供了消息序列，用来说明应用程序、连通层 41 和服务程序之间的通信。

参阅图 10，展开视图描述的是软件的结构，从中可以看出，移动台以及附属装置 28 的资源或者子系统 48，是由服务程序 50 控制的，它提

供一个 ISI 接口，并使得应用程序 52 能够使用移动台的子系统 48。应用程序 52 用 ISI 消息向连通层 41 发出资源请求。如上所述，连通层 41 包括 PhoNet 和通信管理器 41A。应用程序管理器 54，它也可以是通信管理器 41A 的一部分，负责给应用程序 52 注册。事件服务程序 56，它同样也可以是通信管理器 41A 的一部分，处理从服务程序 50 到应用程序 52 的事件分配。面板 58 为应用程序 52 提供 IO 功能，它还有一个有关的面板服务程序 60。同一处理器里的这些各种子系统 48 互相之间还可以直接通信。

图 11 的物理视图说明的是利用系统总线（FBUS、MBUS、IR）将移动台 10（也叫做无线电单元）跟例如附属装置 28、个人计算机（PC）60、跟第二个 PC 60 连接的 PCMCIA 卡 62 这样的 PC 卡、DCS 电话的 AMPS 包这样的外部无线电单元 64、蜂窝网 66 连接，以及通过蜂窝网 66 跟网络应用程序连接，例如跟电话号码查询、内部链路等等这样的无线电（OTA）功能连接。这一例子中的移动台 10 包括增值业务（VAS）应用程序 70 和便携式用户接口应用程序 72。

图 12 的任务视图（它跟图 10 的展开视图相似）说明的是 MCU 21 软件的 OS 任务 74 分配。每一个 OS 任务 74 都用一个虚线框来表示。用户接口任务包括应用程序 52、服务程序 50、应用程序管理器 54（这一功能可以集成到通信管理器 41A 中去）和面板 58。在其它的 MCU 21 OS 任务中还有其它的应用程序。服务程序 50 可以位于跟子系统一样的任务中，或者不同的任务中，而多数复杂的子系统（例如 CS）都包括几个 OS 任务。PhoNet 有它自己的 OS 任务。或者每一个任务包括一个应用程序，或者服务程序有它自己的通信管理器 41A。

描述完总的体系结构以后，下面将介绍实体。实体指的是应用程序 52、服务程序 50、连通层 41 或者子系统 48。ISA 将软件分成应用程序 52 和服务程序 50。有必要区分服务程序 50 和应用程序 52，以保证体系结构的正确划分。这一信息有助于理解某一实体在干什么以及添加新功能时如何划分系统。

服务程序 50 的特性包括：服务程序控制资源、服务程序 50 提供接口时其它实体能够访问它控制的资源（假设该接口符合 ISI 规范）；只能利用有效的资源地址通过连通层 41 发出资源请求来访问服务程序 50；只有服务器 50 可以为资源请求提供服务；给定服务程序 50 可以利用一

个或多个其它服务程序50提供的服务来构成它自己的服务;服务程序50不通过面板58为用户提供信息。

应用程序52的特性包括:应用程序52不控制资源;应用程序52不通过ISI接口提供服务;不能通过连通层41用资源请求访问应用程序52;应用程序52用一个或多个服务程序50的服务构造功能;应用程序52可以通过面板58将功能提供给用户,但并不要求它具有这一功能。

考虑到事务处理过程中实体的角色,让服务程序50利用它自己的资源和其它服务程序50的服务构成它自己的服务是有可能的。例如,考虑GSM移动台的存储器搜索功能。在这一例子里,便携式用户接口里有一个应用程序52,它利用面板服务程序60和电话号码簿服务程序50提供的服务构成存储器搜索功能。应用程序52可以请求电话号码簿服务程序50寻找电话号码簿中跟名字相同的电话号码。号码簿处理这一请求,并为发出请求的应用程序52给出一个最接近的结果,或者给出出错信息。

在处理应用程序的请求时,可能要求服务程序50使用它自己的资源,例如,电话自己的号码簿存储器,以及另一个服务程序的资源,例如SIM服务程序50,如果号码簿储存在SIM 27中。在这种情况下,处理应用程序的请求时,号码簿服务程序50自己不得不向SIM服务程序50发出资源请求,以访问SIM 27里储存的号码。发出最初资源请求的应用程序52,不知道也不需要知道,处理它的资源请求的服务程序50不得不也发出资源请求来完成这一任务。

在这种情况下,搞清某一实体在特定事务处理过程中的角色是很有用处的。

事务处理包括一系列的通信,也就是两个实体之间的消息传递。在每一次事务处理中一个实体的角色是客户程序,其它实体的角色则是服务程序。作为客户程序,某一实体发出资源请求、接受响应、指示或者查询请求,或者发送信息响应消息。在事务处理中划分成应用程序52或者服务程序50的实体可以是客户程序。扮演服务程序50的角色的实体为资源请求提供服务、发出查询请求、发送指示消息或者接受信息响应消息。服务程序50的角色只能由划分成服务程序50的实体扮演。应用程序52永远都不会扮演服务程序50的角色。有可能某一实体在一次事务处理过程中扮演服务程序50的角色,例如,为资源请求服务,然后在下一次事务处理中扮演应用程序52的角色,也就是发出资源请求。

应用程序 52 在一个或者多个服务程序 50 的帮助下完成软件的功能。移动台 10 里面和外面的应用程序 52 利用同样的服务程序 50 接口。总之，应用程序 52 利用服务程序 50 提供的资源。通过连通层 41，也就是通信管理器 41A 和 PhoNet，可以访问服务程序 50。应用程序 52 从其它的应用程序 52 或者外部装置接收事件。应用程序 52 可以产生面板 58，从而输出信息给用户并接受用户的输入。

应用程序管理器（AM）54，它在本发明的优选实施方案中的功能被集成在通信管理器 41A 里，负责在任务中管理应用程序 52 和服务程序 50。应用程序管理器 54 可能会有几个实例（instances），每一个都在不同的 OS 任务中运行。应用程序管理器 54 的职责包括：注册新的应用程序 52/服务程序 50；根据请求撤销应用程序 52/服务程序；在任务中管理应用程序 52/服务程序 50 的事件注册；将事件传递给注册应用程序 52/服务程序 50；启动跟应用程序管理器 54 某一实例有关的初始应用程序 52/服务程序 50。

应用程序管理器 54 里的许多应用程序 52/服务程序 50 还被通信管理器 41A 用于特定的任务。即使某一任务只有一个应用程序 52 或者服务程序 50，也要提供应用程序管理器 54 的一个实例。

上述 VAS 应用程序 70 可以分成固定应用程序、动态应用程序和可下载应用程序。固定应用程序是一个永久性的应用程序，从移动台 10 以外接收一些信息，例如用网络 66 发过来的一则消息将一个新的振铃音编入移动台（见图 11）。动态应用程序跟网络 66 有更多的相互作用，例如一个内部浏览器。可下载应用程序包括脚本语言或者目标码，可以通过空中接口从网络 66 向移动台 10 编程。

外部应用程序（MCU 21 以外的）类似于便携式 UI 应用程序，但它是在另一个处理器里运行的。它们用 ISI 消息跟移动台 10 的面板 58 和服务程序 50 通信。这样的例子包括个人通信器类型的应用程序和录音机应用程序。

网络应用程序通过空中接口利用移动台的资源。对移动台资源的使用由例如移动台 10 里的 VAS 应用程序 70 控制。

有各种各样的服务程序 50。总之，服务程序 50 为（所有的）应用程序 52 提供一组服务。服务程序 50 可以在移动台 10 以内或者以外。服务程序 50 可以分成以下几类：

默认服务程序：当连通层 41 被配置成让针对某一资源的所有资源请求都发往这一特定服务程序 50 时，服务程序 50 就是该资源的默认服务程序。

多个服务程序：移动台 10 可能有两个或者更多的相似资源，也就是同一种类型的资源。在这种情况下，每一个资源都有它自己的服务程序 50，这意味着对于移动台 10 和它的附件中某一特定的资源类型，有一个以上的服务程序 50。

例如，可能有多个号码簿。在移动台 10 的个人通信器实施方案里，个人通信器单元和无线电单元中可能有几个号码簿。这些号码簿都有它们自己的服务程序。

又例如，在多模式移动台里，可能有两种呼叫资源，所支持的每一种蜂窝/无绳系统都有一种（例如 AMPS 和 IS-136）。这两种呼叫资源都有它们自己的服务程序 50。

动态服务程序：动态服务程序是在编译/连接以后添加到系统里的服务程序。动态服务程序可以在移动台 10（例如可下载软件）内，也可以在跟移动台 10 连接的其它装置里。

导控器：在移动台 10 和它的附件中有多个服务程序 50，从而给实体分配较高级的控制权，在多个服务程序 50 中选择，例如规定多个服务程序中哪一个是默认服务程序的情况下。在这里该实体就叫做导控器。导控器可以是多个服务程序 50 中的一个，或者它可以是一个独立的实体。

事件服务程序 56，在本发明的这一优选实施方案里它的功能被结合进通信管理器 41A 里，用于处理从服务程序 50 到应用程序 52 的事件分配。这一事件服务程序 56 的功能如下。

应用程序 52 利用事件服务程序 56 功能向某一资源的事件注册（例如来自一个时间保持资源的定时中断）。该事件服务程序 56 还要储存关于哪些应用程序 52 要注册哪些事件的信息。事件服务程序 56 从服务程序 50 接收所有的事件，并将这些事件分发给注册了要接收这些事件的所有应用程序 52。

应用程序 52 利用一组面板 58 来提供它自己的 I/O 功能。面板服务程序 60 的目的是从应用程序 52 提取类型专用信息以提高它们的便携性。应用程序 52 和面板 58 之间的接口是基于应用程序 52 可能不是移动

台 10 这一事实的消息。

这样，面板服务程序 60 用于处理应用程序 52 和面板 58 之间的通信。它提供生成和删除面板 58 的方法。面板服务程序 60 用一个堆栈储存描述当前面板 58 和它们的相对优先级的信息。外部应用程序 52 可以通过从面板服务程序 60 预约面板 58 来使用移动台的小键盘 22 和显示器。外部应用程序 52 可以产生定制面板或者预定面板 58。

现在来看最关心的一个特殊服务程序，系统模式控制（SMC）服务程序是进行多模式移动台控制的一个特殊服务程序。多模式移动台有两重目的：将不同的蜂窝系统（或者无绳系统）合而为一，从而在覆盖范围和容量方面得到最好的性能（移动台 10 只有一个电话号码但支持不同的系统）；将两种不同的独立系统合并成一个，使用户可以用一个终端获得两个系统。在这后一种情况下，移动台 10 在每一个系统里都有一个电话号码。

在 ISA 里，SMC 服务程序的角色是控制对蜂窝（或者无绳）系统软件的选择，这可以在外部实现（MCU 21 以外，例如 DCS 电池组里的 AMPS，如图 11 中的单元 66 所示），也可以在内部实现（在 MCU 21 以内，例如 DECT + GSM 或者 PDS + PHS）。SMC 服务程序不需要包括在单模式移动台内（例如只有 AMPS 或者 GSM）。

SMC 服务程序用于将多模式功能从专用服务程序 50 中分离出来。系统专用服务程序 50 包括，例如，呼叫服务程序、短消息业务（SMS）服务程序、辅助业务（SS）服务程序和网络选择（NS）服务程序 50。这种分离最好通过灵活地配置连通层 41，并为连通层 41 添加一个特殊的路由选择机制来完成，这一机制由路由器 40 来识别和启动。

不同的系统是在不同的系统服务程序 50 里实现的，它们为外部和内部系统提供同样的体系结构。另外，它们将不同的系统完全分离开来，这也会增强系统结构的模块化程度。

图 22A 和 22B 说明的是系统模式控制（SMC）服务程序 50A 的两种实施方案，它结合了系统选择规则，以及它跟应用程序 52 和系统服务程序 50 的连接，特别是第一个系统服务程序 50B 和第二个系统服务程序 50C。在图 22A 的实施方案里，SMC 服务程序 50A 通过连接层 41 来工作，以便控制应用程序 52 跟系统服务程序 50B 和 50C 中的一个的连接。SMC 服务程序 50A 的操作用一个单刀双掷（SPDT）开关 41B 表示。图

22B 的实施方案，它是当前的最佳实施方案，通过连通层 41 将应用程序 52 跟 SMC 服务程序 50A 连接起来，然后同样是通过连通层 41 将 SMC 服务程序 50A 跟系统服务程序 50B 和 50C 中选中的那一个连接起来。

在图 22A 所示的实施方案里，应用程序 52 将一般性的消息发送给系统服务程序 50B 或者 50C。SMC 服务程序 50A 控制这一连通层 41 中的开关 41A，从而间接地影响消息的路由选择。

除了控制路由选择外，SMC 服务程序 50A 还有几个其它的功能。这些功能包括决定哪一个系统应当是活动的（系统的挂起和恢复）；将系统之一关闭的时候初始化所需要的所有功能（例如呼叫转移、自动拨号到一个操作中心）；控制对非活动系统的扫描过程（例如每一个扫描 5 分钟）；控制如何选择不同的系统（根据系统的优先级）；控制图 22A 中实施方案里的连通层 41 开关（控制路由选择）；处理用户控制的活动（例如强制跟一个非活动系统建立呼叫，用户启动的扫描）；以及跟应用程序 52、连通层 41 和系统专用服务程序 50B 和 50C 相互作用，以实现多模式。

总之，应用程序 52 通过 ISI 服务程序接口利用移动台 10 的资源构造它们自己的功能。应用程序 52 向连通层 41 发出资源请求。连通层 41 将资源请求跟当前这一资源活动的服务程序 50 比较，通常是默认服务程序。MCU 21 在连通层 41 里储存并维护着所有这种服务程序的配置表。资源到服务程序的变换的默认配置在编译时就定了下来，但这种变化可以在运行时改变。试图利用不被支持的系统资源的应用程序 52 会在响应消息中收到“不支持”指示。

总结起来，就象前面所说明的那样，连通层 41 负责监测、注册和对所连接的服务程序 50 和应用程序 52 寻址。它为消息选择路由，从资源名称到资源的逻辑地址，从逻辑地址到物理地址。发送方不必知道消息是发往了同一处理器内的一个实体还是发给了附属装置里的一个实体，这都隐藏在物理总线后面。连通层 41 掩盖了 ISI 接口后面的实际物理位置。

应用程序 52 和服务程序 50 利用通信管理器 41A 进行通信。通信管理器 41A 检查通信的目标是否位于同一 OS 任务内。如果是这样，就跟这一目标通信。如果目标不在同一 OS 任务内，通信管理器 41A 就将这一请求传送给 PhoNet 作为一则 ISI 消息。然后 PhoNet 将这一消息发送

给合适的 OS 任务的邮箱，或者如果目标位于另一个装置里，它就将这一消息发送给这一装置，通过适当的媒介，例如串行 IR 总线 18C。响应消息用同样的方式选择路由，或者由通信管理器 41A 直接发送给发送方，或者由 PhoNet 发送给发送方 OS 任务的通信管理器。每一个 OS 任务都有一个通信管理器 41A 的一个实例，其中包含应用程序 52 或者服务程序 50。

实体（应用程序 52 和服务程序 50）之间的通信是基于消息的。这些消息或者是从任务到任务的 OS 消息，或者是通过某种物理总线（例如 FBUS、IR 链路）发送的消息。这些消息是 ISI 请求、响应和指示。

应用程序 52 和服务程序 50 之间的通信分组如下:

资源请求由应用程序 52 产生，请求使用移动台 10 的一种资源。资源请求是通过连通层 41 从应用程序 52 传送给服务程序 50 的。

响应是由服务程序 50 发送给发出请求的应用程序 52 的，用于发送资源请求引发的操作的结果。

指示（indication），在这里也叫做通知或者事件，是由服务程序 50 发送的，用于将服务程序 50 控制的资源的状态信息传递给一个或者多个应用程序 52。

状态信息可以由服务程序 50 发送，用于将一个时间报告给一个或者多个应用程序 52，其中的事件发生在服务程序 50 控制的资源里。在这种情况下，这一事件不是由应用程序 52 发出的资源请求直接引发的。这种事件的一个例子是呼叫服务程序收到的呼入。

状态信息可以由服务程序 50 发出，用来向一个或者多个应用程序 52 报告一项操作的状态。这一特定的状态信息不是操作的结果，操作的结果是用响应消息给出的。

状态信息也可以根据应用程序 52 向服务程序 50 发出的资源请求引发的操作而发出。操作是在服务程序 50 控制的资源里进行的。

应用程序 52 和服务程序 50 之间的通信还包括如下。

查询请求是根据应用程序 52 发出的资源请求正在进行处理的服务程序 50 发出的。为了继续处理，服务程序 50 需要从发出请求的应用程序 52 获得某些信息。需要的信息可以是，例如，服务程序 50 可能需要通知应用程序 52 发生了一个事件，并跟应用程序 52“握手”（例如请求指示服务程序是否应当继续处理）。查询请求也可以由服务程序 50 发

出，以便从应用程序 52 获得更多的信息，从而能够继续处理。

最后，通知响应（Inform Response）由应用程序 52 发出，作为对服务程序 50 发出的查询请求的响应。通知响应可以包含服务程序 50 需要的额外信息。

所有消息标识符都要附加以下后缀，以说明消息的本质。

_REQ	资源请求
_RESP	资源请求的响应
_IND	指示
_CONS	查询请求
_INF	通知响应

应用程序 52、连通层 41（通信管理器 41A 和 PhoNet）、服务程序 50 和各种子系统之间的通信用图 13、14 和 15 所示的消息序列图来说明。在这些示意图中，应用程序 52 和服务程序 50 或者在 MCU 21 以内，或者在 MCU 21 以外。在所有的图中，描述的都是相同的协议，也就是说，应用程序 52 发送资源请求消息，支持这一资源的服务程序 50 接收资源请求，并将响应发回发出请求的应用程序 52。

这些示意图使用的缩写有：f. call = 功能调用，if = 接口，msg = 消息，r. value = （函数的）返回值，req = 请求，resp = 响应，CM = 通信管理器。

首先看图 13，其中的应用程序 52 和服务程序 50 都在同一 OS 任务内，CM 41A 接收应用程序 52 发出的 ISI 消息形式的资源请求。CM 41A 判断给这一资源请求提供服务的服务程序 50 位于同一 OS 任务内。CM 41A 将资源请求传递给服务程序 50。如果需要，服务程序 50 将这一资源请求解释并变换成子系统 48 的接口功能调用。子系统响应或者是 OS 消息或者是接口功能的返回值. 现在服务程序 50 有了构成响应消息所需要的数据。服务程序 50 将资源请求的响应发回 CM 41A，后者则将这一响应发回发出请求的应用程序 52。

下面参考图 14，其中的应用程序 52 和服务程序 50 位于同一处理器（MCU 21）里，但在不同的 OS 任务里，第一个 CM 41A（CM#1）从应用程序 52 接收 ISI 消息形式的资源请求。CM#1 判断给这一资源请求提供服务的服务程序 50 不在同一 OS 任务内。CM#1 将这一资源请求传递给 PhoNet，后者将这一消息发送给包括这一资源的服务程序 50 的任

务的 OS 邮箱。

这一任务的第二个 CM 41A'（CM#2）读这一个 OS 邮箱，并将这一消息发送给服务程序 50。如果需要，服务程序 50 将这一资源请求解释并变换成目标子系统 48 的接口功能调用。这一子系统的响应或者是一则 OS 消息或者是接口调用的一个返回值。服务程序 50 现在有了构造响应消息所必需的数据。响应消息被发送给 OS 任务，这一 OS 任务包含在最初发送的资源请求消息的报头里（见前面的表 6）。服务程序 50 将资源请求的响应发送回 CM#2，CM#2 将这一响应消息发送给 PhoNet，PhoNet 将这一响应转发给 CM#1。位于应用程序 52 里的 CM#1 收到这一消息，并将它传递给发出请求的应用程序 52。

下面参考图 15，其中的应用程序 52 和服务程序 50 位于不同的处理器里。首先，CM#1 从应用程序 52 接收 ISI 消息形式的资源请求。CM#1 判断为这一资源请求提供服务的服务程序 50 不在同一 OS 任务里。CM#1 将这一资源请求发送给 PhoNet#1。PhoNet#1 将这一资源请求消息发送给包括这一资源的装置。PhoNet#1 发送这一资源请求消息用的是一条合适的链路，例如 IR 链路 18C，如上所述，就象例如该装置向 PhoNet#1 注册时指定的优选链路一样。该装置内的 PhoNet#2 从选中的链路收到这一资源请求消息，并将这一消息发送给包括这一资源的服务程序 50 的任务的 OS 邮箱。这一任务的 CM#2 读邮箱，并将消息发送给服务程序 50。如果需要，服务程序 50 将这一资源请求解释并变换成子系统接口功能调用。子系统的响应或者是一则 OS 消息，或者是接口功能的返回值。服务程序 50 现在有了构成响应消息所需要的数据，这一响应消息被发往资源请求消息的报头里包含的装置和对象。服务程序 50 将响应消息发回 CM#2，CM#2 将响应消息发送给 PhoNet#2，后者则将响应消息发送给最初的装置里的 PhoNet#1。PhoNet#1 收到响应消息，并将它作为 OS 消息发送给目标任务。CM#1 收到这一 OS 消息，并将这一响应消息发回发出请求的应用程序 52。

下面根据本发明的这一方面更详细地介绍智能业务接口（ISI）。

ISI 服务程序 50“拥有”服务程序 50 和客户程序之间的接口（客户程序是使用 ISI 服务程序的一个应用程序 52 或者另外一个服务程序 50）。同样，发送给和来自 ISI 服务程序的消息总是用 ISI 服务程序的所需格式表示，而不是用发出请求的客户程序的格式。这样做保证了 ISI

服务程序接口的定义不依赖于使用它的客户程序，而且所有客户程序都可以使用这一 ISI 服务程序的业务。

ISI 消息的后缀使得消息能够按照使用它们的环境的方式来定义。从后缀可以获得的信息包括：谁是消息的发送方，客户程序还是 ISI 服务程序；谁应当是消息的接收方，客户程序还是 ISI 服务程序；以及要在什么样的环境里发送这一消息。也就是说，这一消息是否是前一消息的结果，或者这一消息是否是一个新的请求。

前面定义了 ISI 服务程序和它的客户程序之间的通信所使用的消息，它包括资源请求、响应、指示、查询请求和通知响应消息。

每一个 ISI 消息都包括一个消息报头。消息报头紧跟 PhoNet 消息报头。所有 ISI 消息之间的消息数据部分的第一部分都相同。

下表给出了基本 ISI 消息的格式。第一栏里的 H 和 D 表示这一字段是在消息的报头部分还是数据部分里。

ISI 消息格式

字段	说明	备注
H	媒介	规定要使用的媒介，例如 PN_MEDIA_MBUS，PN_MEDIA_FBUS
H	接收方装置	接收方的装置地址。发送方地址的第一部分。
H	发送方装置	发送方的装置地址。发送方地址的第一部分。
H	资源	ISI 资源组，就象 ISI 规范里所定义的一样。这在 PhoNet 术语里叫做“功能”。
H	长度	消息里的随后字节数，以接收方对象开始，并且包括接收方对象。这一字段总是按照处理器处理字数据的 LSB/MSB 惯例。在处理器间的通信中，PhoNet 里的媒介驱动程序变换发往 PhoNet 和来自 PhoNet 的固定 MSB、LSB 顺序。
H	接收方对象	接收方对象的内部地址。接收方地址的第二部分。

H	发送方对象	发送方对象的内部地址。发送方地址的第二部分。
D	唯一的事务处理 ID	用于说明消息属于哪一个事务处理。
D	消息 ID	ISI 消息 ID
D	消息数据 1	ISIS 规范里说明的这一消息的消息数据。
D	消息数据 2	消息数据……
D	消息数据 n	消息数据……

这一 ISI 消息规范不必包括每一则消息的完整 ISI 消息描述，因为所有消息的消息报头里的许多字段都是用同样的方式处理的。

消息 ID

描述	参数
消息报头信息	跟上述 PhoNet 消息报头里的一样
资源	RESOURECE_ID
消息报头 int.	跟上述 PhoNet 消息报头里的一样
唯一的事务处理 ID	
消息 ID	MESSAGE_ID
消息数据	

为每一个 ISI 消息都规定了以下 ISI 消息字段。

资源 – 这一字段描述消息所属的资源（以及服务程序 50），例如 ISI_PND（电话号码本），ISI_CALL，ISI_MENU 等等。

唯一的事务处理 ID（UTID） - 这一字段用于标识消息所属的事务处理。

消息 ID – 这一字段描述资源内的某一特定消息.

消息数据 – 这些字段是消息的内容，见具体的消息规范。

作为一个例子，以及为了给一则资源请求消息构造一则 ISI 消息，资源请求（消息类型_REQ）使用连通层 41 的“按源选择路由”功能，将资源请求从发出请求的应用程序 52 发往服务于这一请求的适当的服务程序 50。这意味着构造 ISI 资源请求消息时，发出请求的应用程序 52 不必知道接收方的对象地址。

下表说明构造用于生成资源请求消息的 ISI 消息的具体信息。

字段	说明	备注
H	媒介	PN_MEDIA_ROUTING_REQ
H	接收方装置	PN_DEV_HOST
H	发送方装置	发出资源请求的应用程序 52 的装置 ID。
H	资源	ISI 资源组，就象为所需消息定义的一样。
H	长度	消息长度，作为紧跟的字节的字节数。
H	接收方对象	PN_OBJ_ROUTING_REQ
H	发送方对象	发出资源请求的应用程序 52 的对象 ID。
D	唯一的事务处理 ID	发出请求的应用程序 52 产生的唯一的事务处理 ID。
D	消息 ID	ISI 消息 ID，如同合适的 ISI 规范里为所需消息定义的一样。它的后缀是_REQ。
D	消息数据 1	消息数据，跟合适的 ISI 规范为所需消息定义的一样。
D	消息数据 2	消息数据……
D	消息数据 n	消息数据……

又例如，为了给所有其它类型的消息构造 ISI 消息，_RESP、_IND、_CONS 和_INF 类型的消息都利用连通层 41 的“根据接收方的地址选择路由”功能。这意味着发送消息的实体（应用程序 52 或者服务程序 50）必须事先知道接收方的地址。总之，由于事务处理或者以前的事务处理中早些时候的通信，发送方会知道接收方的地址。由于这些消息类型的实质，以前的通信或者以前的事务处理会已经发生，就象前面图 13~15 的例子中所描述的那样。

下表给出了构造 IS 消息的信息，这一消息将被直接发送给接收方。

字段	说明	备注
H	媒介	PN_MEDIA_ROUTING_REQ
H	接收方装置	接收这一消息的实体的装置 ID。装置 ID 可以从以前来自这一实体的消息的报头中“发送方装置”字段里找到。
H	发送方装置	发出资源请求的应用程序 52 的装置 ID。

H 资源	ISI 资源组，就象适当的 ISI 规范里为所需消息定义的一样。虽然这一消息不是由功能选择路由的，但仍然包括了这一字段，因为消息是用资源和消息 ID 的组合来标识的。消息 ID 的范围限于某一特定的资源。
H 长度	消息长度，作为随后的字节的字节数。
H 接收方对象	接收消息的实体的对象 ID。这一对象 ID 可以在以前来自这一实体的消息的报头里“发送方装置”字段里找到。
H 发送方对象	发出资源请求的应用程序 52 的对象 ID。
D 唯一的事务处理 ID	唯一的事务处理 ID。这一 UTID 或者是由发送实体产生的，或者是从早些时候的事务处理中的消息里或者是从以前的事务处理中拷贝出来的。
D 消息 ID	ISI 消息 ID，如同合适的 ISI 规范里为所需消息定义的一样。
D 消息数据 1	消息数据，如同合适的 ISI 规范里为所需消息定义的一样。
D 消息数据 2	消息数据……
D 消息数据 n	消息数据……

ISI 是一个异步消息传递系统。用 ISI 消息通信的实体可以是分布式的。实体可以在同一个 OS 任务里，在同一个处理器的不同 OS 任务里，或者在不同的处理器里。实体可以同时发送消息给一个或者多个实体和以及从这一个或者多个实体接收消息。应用程序 52 可以使用许多服务程序 50 的业务，服务程序 50 也可以将它们的服务提供给许多应用程序 52。

这一接口的异步本质意味着实体无法知道什么时候能够收到它们发出的消息的响应。由于可能存在分布式的实体，每一条消息的消息传输延迟都可能不同，它取决于目的地的位置和传送消息的媒介。应用程序 52 和服务程序 50 之间的关系可能意味着在任何时刻某一实体都在跟许多实体通信。

由于上述特性，这一 ISI 系统支持应用程序 52 和服务程序 50 管理它们自己的通信，从而可以轻松地应付异步通信、分布式通信和同时多

个通信所带来的效应。

ISI 利用事务处理来管理消息的传递。事务处理是两个实体之间的一个通信（也就是消息）序列。在每一个事务处理中，一个实体扮演客户程序的角色，其它实体则扮演服务程序的角色。客户程序角色由发出资源请求、接收响应、指示或者查询请求或者发送通知响应消息的实体扮演。划分成应用程序 52 或者服务程序 50 的实体可以在事务处理中扮演客户程序的角色。服务程序角色由服务于资源请求、发出查询请求、发送指示消息或者接收通知响应消息的实体扮演。服务程序角色只能由划分成服务程序 50 的实体扮演。根据定义，应用程序 52 永远不能扮演服务程序角色。

一个实体有可能在一次事务处理中扮演服务程序的角色，例如服务于资源请求，然后在下一次事务处理中扮演客户程序的角色，也就是发出资源请求。

本发明中与 ISI 消息有关的方法包括以下类型的事务处理:

资源请求事务处理;

事件指示事务处理;

事件指示事务处理和继后操作; 和

查询请求事务处理。

当应用程序 52 发送一则资源请求消息时，资源请求事务处理就开始。如上所述，资源请求消息有连通层 41 从发出请求的应用程序 52 发送给提供服务的适当的服务程序 50。然后服务程序 50 满足这一请求。当服务程序 50 处理这一请求时，它可以用一则指示消息响应发出请求的应用程序 52，将请求的进程告诉这一发出请求的应用程序 52，或者它可以向发出请求的应用程序 52 发出查询请求，并接收作为应答的通知响应。当服务程序 50 用一则响应消息响应发出资源请求的应用程序 52 时，资源请求事务处理就结束。

例子 1

图 16 描述的是一个示例性资源请求事务处理，其中的应用程序 52 向电话号码簿（PND）服务程序 50 发出资源请求。PND 服务程序 50 响应这一请求，给发出请求的应用程序 52 发去响应消息。这样就终止了资源请求事务处理。

例子 2

图 17 描述的是一个示例性的资源请求事务处理，其中的应用程序 52 向呼叫服务程序 50 发出一个资源请求。呼叫服务程序 50 响应这一请求。呼叫服务程序 50 通过发送指示消息（例如呼叫进程、呼叫告警、呼叫已连接等等）向发出请求的应用程序 52 报告请求的处理进程。最后，资源请求得到满足，呼叫服务程序 50 发送一则响应（RESP）消息给发出请求的应用程序 52。这样就结束了资源请求事务处理。

例子 3

图 18 描述的是一个示例性的资源请求事务处理，其中的应用程序 52 向 PND 服务程序 50 发出一个资源请求。PND 服务程序 50 响应这一请求。在响应这一请求的过程中，PND 服务程序 50 发现它必须检查是不是要继续当前发出请求的应用程序 52，因为它必须读外部存储器。PND 服务程序 50 向应用程序 52 发出一个查询请求（CONS）。应用程序 52 用一个通知响应（INF）消息响应发出查询请求的服务程序 50，允许其读取外部存储器。然后 PND 服务程序 50 可以完成资源请求的处理。PND 服务程序 50 发送一则响应消息给发出请求的应用程序 52。这样就结束了资源请求事务处理。

当服务程序 50 用指示消息的形式发送事件指示给一个或者多个应用程序 52 时，就开始事件指示事务处理。在此以前应用程序 52 已经注册，以接收指示。事件指示从发生事件的服务程序 50 发送给应用程序 52。当事件指示消息被传递给注了册以接收事件指示的应用程序 52 时，事件指示事务处理就结束。

例子 4

图 19 描述的是一个示例性事件指示处理，它开始于服务程序 50 发送 Call_Coming_IND 给应用程序 52。然后完成这一事务处理。在此以前，应用程序 52 已经注册，以便从服务程序 50 接收事件指示。

如图 19 所示，事件指示事务处理在事件指示消息成功地传递给应用程序 52 以后结束。然而，事件指示事务处理可能会继续下去。在这种情况下，这一事务处理就是有继后操作的事件指示事务处理。

接收事件指示消息的应用程序 52 可能会希望通过发出资源请求对指示消息做出反应，还可能希望将随后的资源请求事务处理跟它收到的事件指示事务处理联系起来。这可能确实如此，因为服务程序 50 正在发送许多事件指示，应用程序 52 想搞清楚它应当对哪一个指示“应答”。

在这种情况下，随后的资源请求事务处理被看作有继后操作的事件指示事务处理的一部分。当服务程序 50 发送响应消息给应用程序 52 发出的资源请求时，就认为有继后操作的事件指示事务处理已经结束。

应当指出，应用程序 52 可能想通过发出资源请求来对指示消息做出反应，但它可能不希望将资源请求直接跟指示消息联系起来。在这种情况下就发出随后的资源请求，作为正常的资源请求事务处理。

例子 5

图 20 描述的是一个有继后操作的示例性事件指示事务处理，当安全（SEC）服务程序 50 发送一个 SEC_Code_Required_IND 消息给应用程序 52 时它就开始，应用程序 52 事先已经注册，以便接收事件指示。然后接收事件指示的应用程序 52 之一发出一个跟事件指示绑在一起的资源请求。这一资源请求是一个 SEC_Code_Verify_REQ。SEC 服务程序 50 用一个响应消息（SEC_Code_Verify_ OK_RESP）响应发出请求的应用程序 52。这样就结束了有继后操作的事件指示事务处理。

查询请求事务处理开始于服务程序 50 发送一个查询请求给应用程序 52，这一查询请求不是正在进行的资源请求事务处理的一部分。连通层 41 将这一查询请求消息从发出查询请求的服务程序 50 传递给适当的应用程序 52。应用程序 52 完成必要的处理。当应用程序 52 用一则通知响应消息响应发出查询请求的服务程序 50 时，查询请求事务处理就结束。

例子 6

图 21 描述的是一个示例性查询请求事务处理，它开始于菜单服务程序 50 发送一则查询请求消息给应用程序 52 时，这一应用程序 52 事先已经注册，以便使用菜单服务程序 50 的业务。应用程序 52 用一个对发出查询的服务程序 50 的通知响应响应查询请求。这样就结束了查询请求事务处理。在注册以后，可以看到应用程序 52 发送一个 MENU_Entry_Add_REQ（一个资源请求）给菜单服务程序 50，作为响应，从菜单服务程序 50 接收一个 MENU_Entry_Add_Ok_RESP。在用户选择了菜单以后，例如通过移动台 10 的小键盘 22 和显示器 20 的用户接口，实际的查询请求事务处理就此开始。

如上所述，每一事务处理都有一个唯一的事务处理 ID（UTID）。事务处理所涉及的实体用 UTID 来识别给特定事务处理的消息。这使得应

用程序 52 和服务程序 50 能够同时在有一个以上的实体的一个以上的事务处理中活动，而同时仍然能够据此比较请求和响应消息。

在每一事务处理中，一个通信实体被指定为事务处理的始发方。始发方负责为事务处理分配 UTID，并将分配的 UTID 包括在事务处理的第一则消息里。然后始发方和其它通信实体在这一事物处理的所有随后消息里都使用分配的 UTID。

始发方是根据要发生的事务处理的类型来指定的，并且是事务处理中的第一个通信实体。下表给出了所有事务处理类型里的始发方。

事务处理	始发方
资源请求	发出请求的应用程序 52
事件指示	指明服务程序 50
有继后操作的事件指示	指明服务程序 50
查询请求	发出查询请求的服务程序 50

希望通过 ISI 通信的每一个实体都维持着自己的 UTID 计数。无论什么时候一个实体分配一个 UTID 给事务处理，它都将自己的 UTID 计数器加一，并在开始这一事务处理的消息的报头的唯一的事务处理 ID 字段里将它包括进去。UTID 是例如一个 8 位的值。MSB 用于说明这一 UTID 是否为一个客户程序或者一个服务程序 50 所拥有。其余的 7 位包括了实体的 UTID 计数器值。在开始的时候，所有的实体都可能将它们的 UTID 计数器置零。

实体可以用 UTID、其它通信实体的地址、资源和消息 ID 来识别一则消息以及它所属的事务处理。在多数情况下，只需要这些元素中的少数来识别一则消息属于哪一事务处理。下表给出了几个例子。

事务处理	应用程序进行识别用的是	服务程序 50 进行识别用的是
资源请求	AO UTID + 资源	AO UTID + 发送方地址
事件指示	SO UTID + 资源	SO UTID + 资源
有继后操作的事件指示	SO UTID + 资源	SO UTID + 发送方地址
查询请求	SO UTID + 资源	SO UTID + 资源

由此可知，本发明人已经描述了一种接口，用于在无线电电话和多个附属装置之一之间进行通信，这多个附属装置有几个物理连接媒介可供选择（例如 IR、RF 或者有线）。本发明的特性包括有两层的 PhoNet

接口:(1)路由器层,用于跟附属装置交换消息,以控制接口功能,并跟踪链路状态;和(2)连接层,用于将路由器层消息转换成适合于物理连接的格式。连接层还包括:(1)控制单元,(2)媒介模块,和(3)帧过滤器。ISI(智能业务接口)是一个消息传递概念,它利用 PhoNet 接口为消息选择路由。有不同类型物理连接的单个附属装置能够通过 PhoNet/ISI 相互通信。同样,电话的单个软件模块能够通过 PhoNet/ISI 通信。电话的 PhoNet/ISI 接口和附属装置的 PhoNet/ISI 有一种主/从关系。PhoNet/ISI 是一种接口体系结构,它支持模块化的软件体系结构,ISA。PhoNet 是所有类型附属装置的一种通用电话接口,它通常都是跟电话的一个有线或者无线连接器连接,或者安装在电池组里。PhoNet 这两个分开的层提供了"即插即用"功能。拥有不同于电话已经支持的物理连接的新的附属装置可以通过在连接层载入一个新的媒介而跟电话实现连接。

虽然介绍本发明时采用了优选实施方案,但是应当明白,对于本领域的技术人员而言可以对这些技术进行许多修改。例如,本发明并不局限于这里公开的具体消息格式、协议、字段大小和字段顺序。本发明也不局限于这里公开的服务程序、子系统、资源和应用程序。还有,可以采用也可以不采用 FBUS、MBUS 和无线总线 18A~18C,同样,可以采用也可以不采用无线和有线以外的其它媒介。例如,在某些应用中可以采用光纤。

因此,虽然介绍本发明时涉及到了上述优选实施方案,但是对于本领域里的技术人员而言,显然可以更改本发明的形式和内容,而不会偏离本发明的范围和实质。

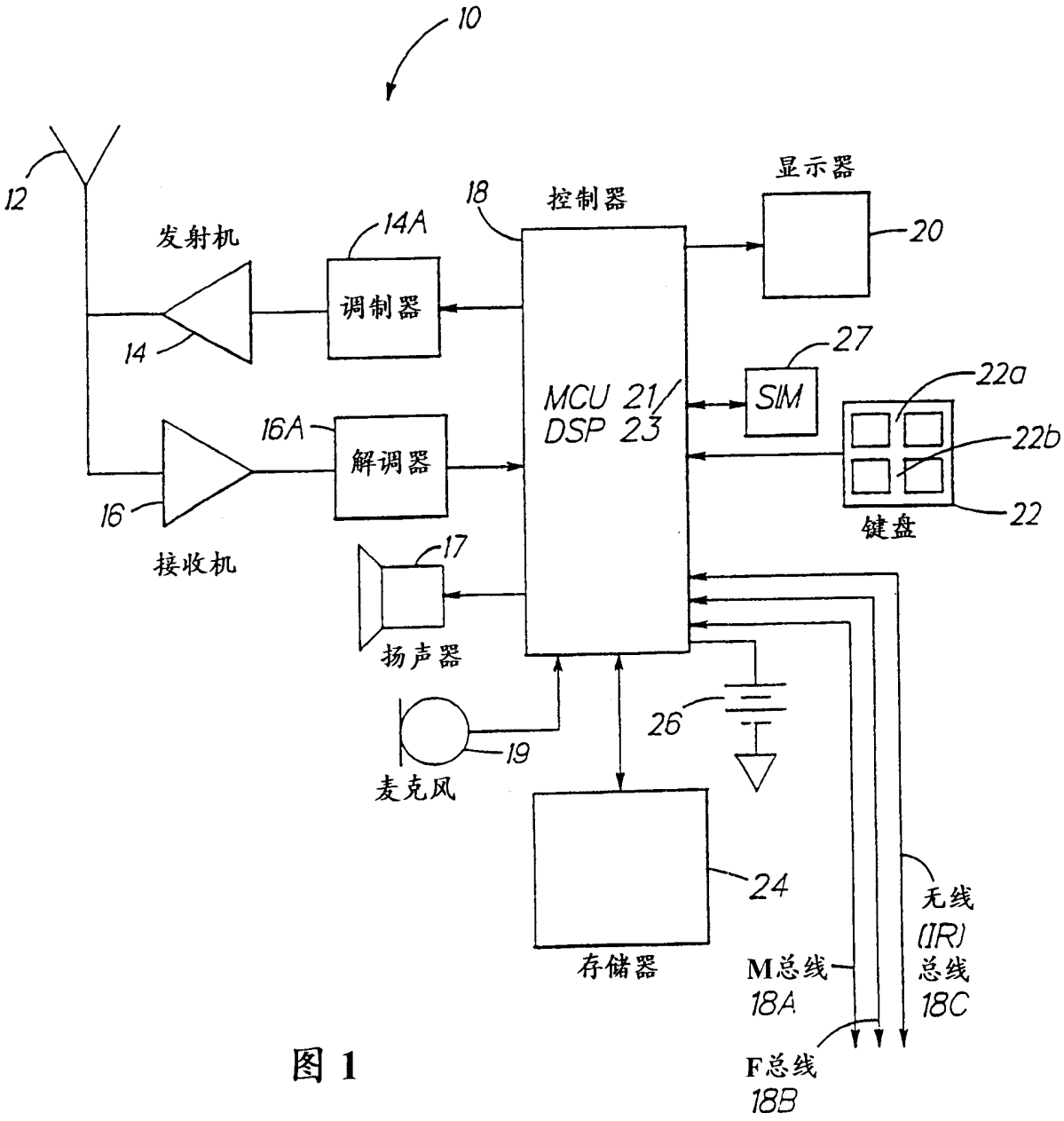

10
12
发射机
14A
调制器
14
18
控制器
显示器
20
27
SIM
MCU 21/
DSP 23
22a
22b
22
键盘
16A
解调器
16
接收机
17
扬声器
19
麦克风
26
24
存储器
M总线
18A
F总线
18B
无线
(IR)
总线
18C

图 1

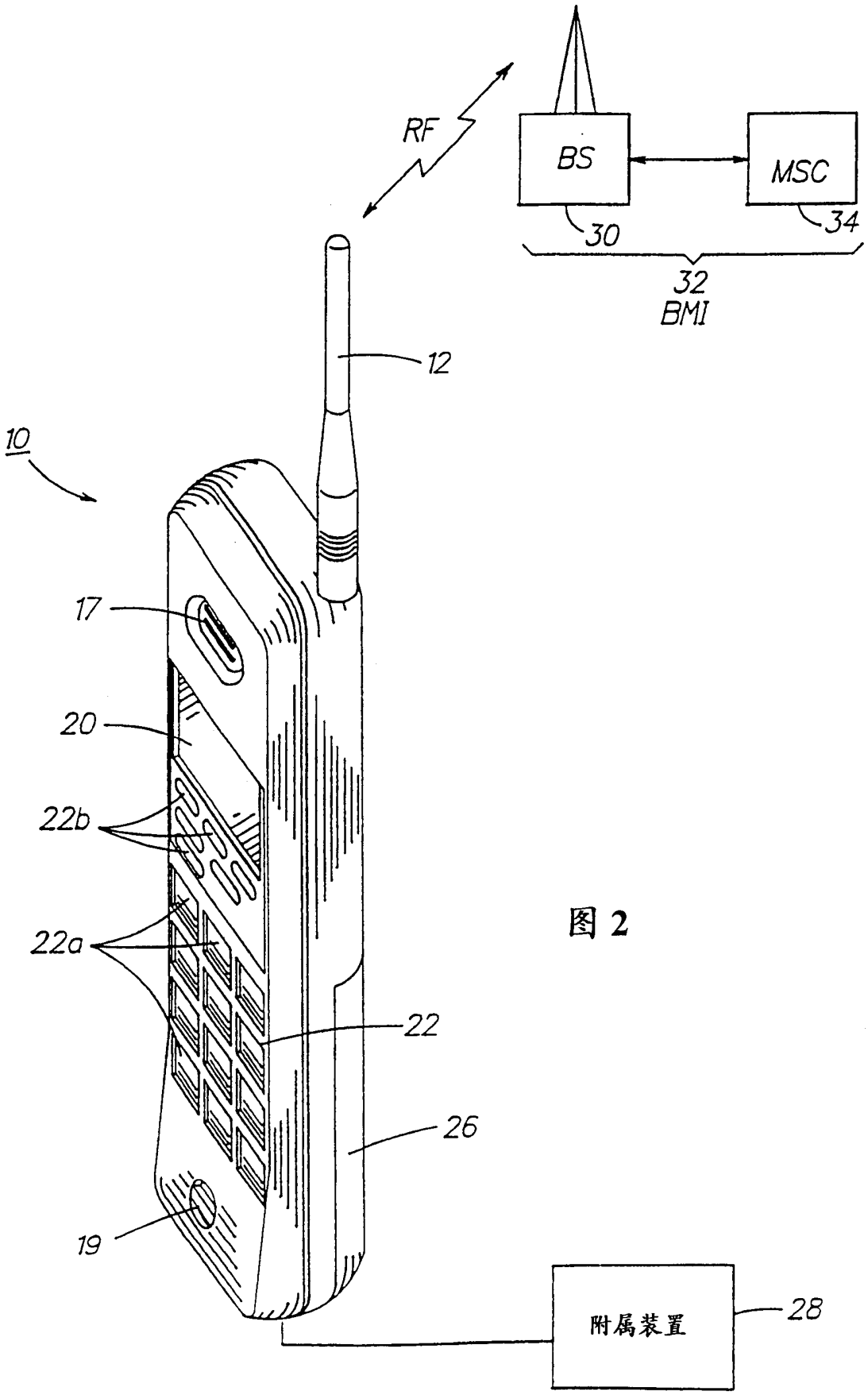


图 2

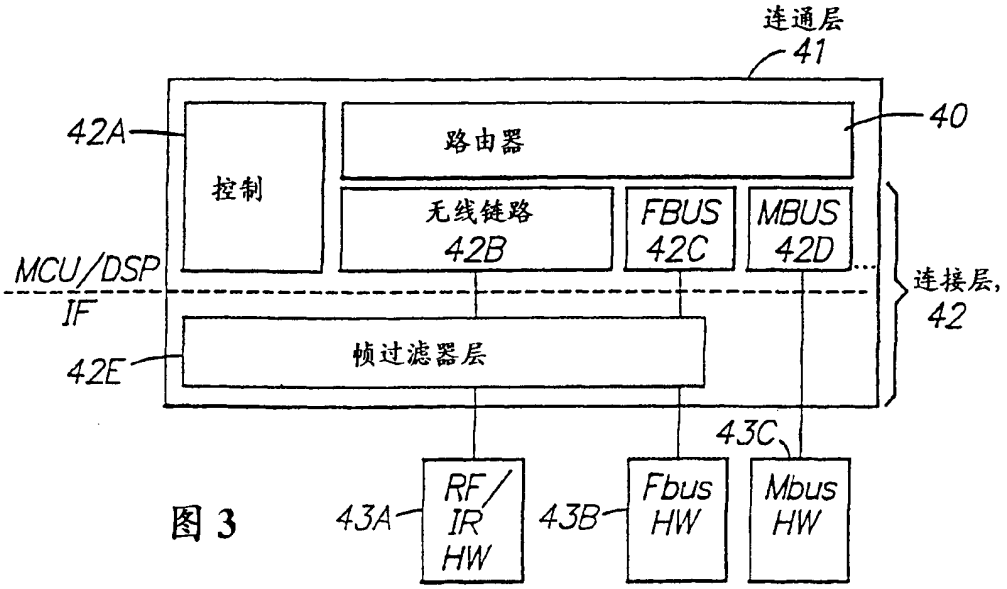


图 3

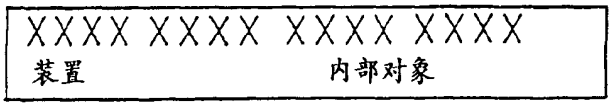


图 4

主管道

装置	对象媒介	
04	01	IR
04	02	bus
04	06	IR

管道目标

装置	对象
03	01
06	02
06	06

图 5

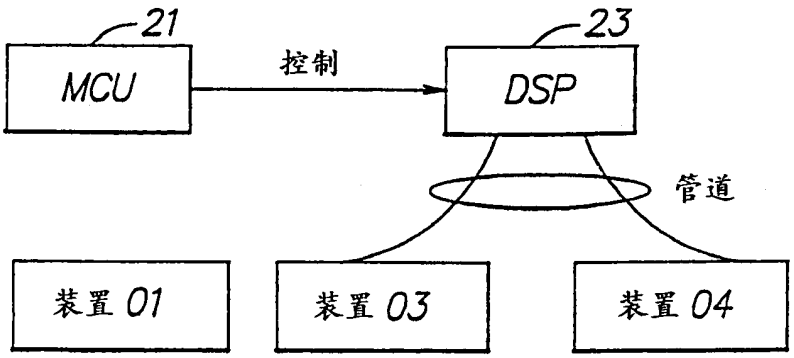

21
MCU
控制
23
DSP
管道
装置 01
装置 03
装置 04

图 6

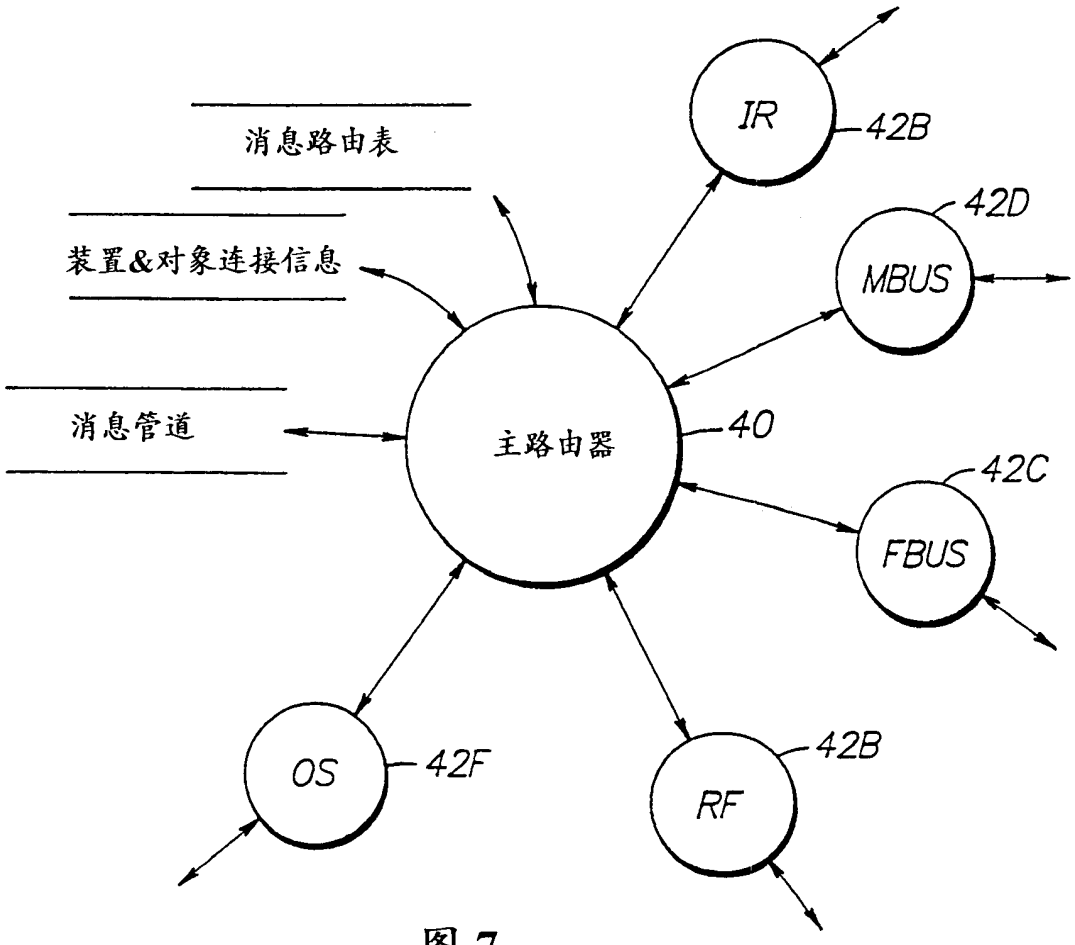

消息路由表
装置&对象连接信息
消息管道
主路由器
40
IR
42B
MBUS
42D
FBUS
42C
OS
42F
RF
42B

图 7

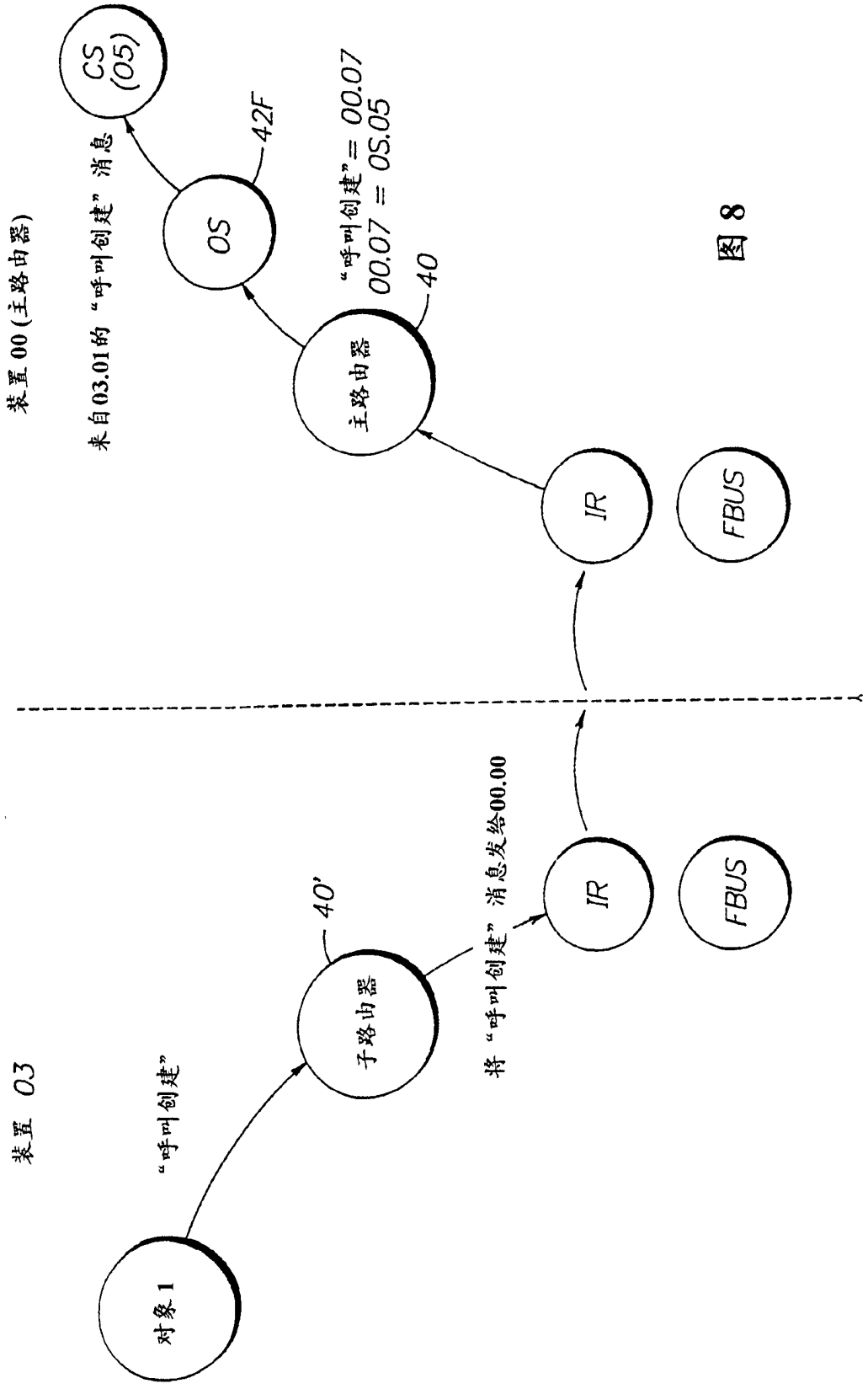

装置 03
对象 1
"呼叫创建"
子路由器
40'
将"呼叫创建"消息发给00.00
IR
FBUS
装置 00 (主路由器)
来自03.01的"呼叫创建"消息
主路由器
40
"呼叫创建"= 00.07
00.07 = 05.05
OS
42F
CS
(05)
IR
FBUS

图 8

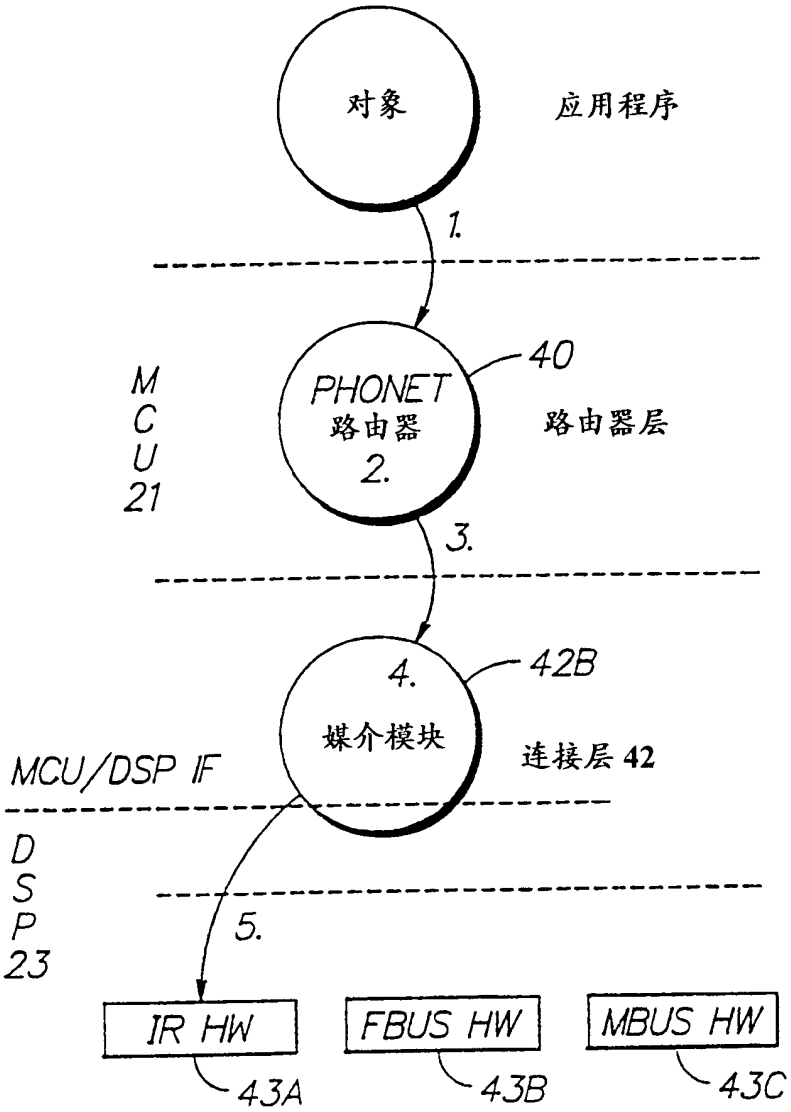

对象
应用程序
1.
PHONET
路由器
2.
40
路由器层
MCU
21
3.
4.
媒介模块
42B
连接层 42
MCU/DSP IF
DSP
23
5.
IR HW
FBUS HW
MBUS HW
43A
43B
43C

图 9

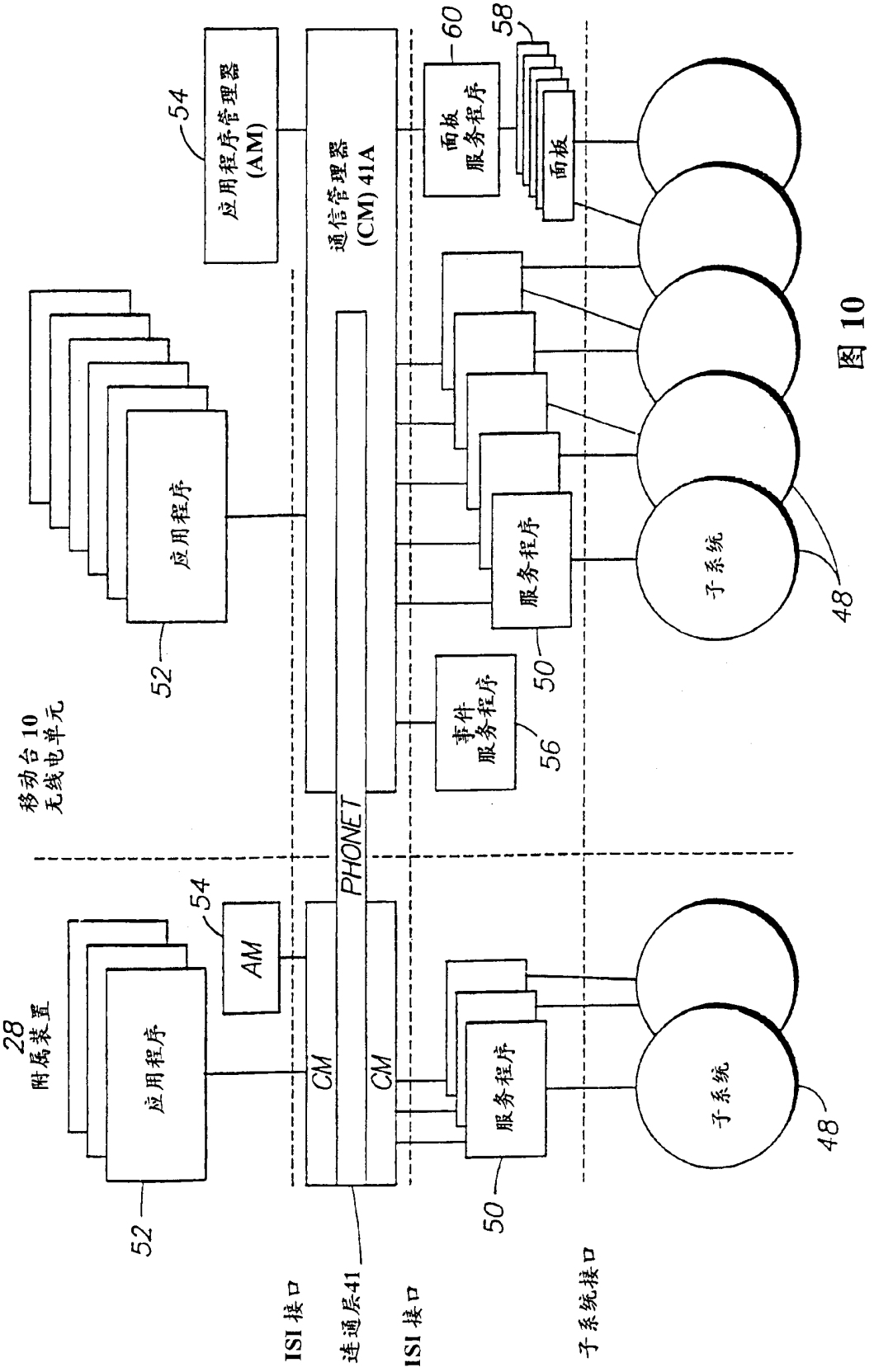

移动台 10
无线电单元
28
附属装置
应用程序
52
AM
54
CM
CM
PHONET
连通层41
ISI 接口
ISI 接口
应用程序
52
应用程序管理器
(AM)
54
通信管理器
(CM) 41A
面板
服务程序
60
面板
58
事件
服务程序
56
服务程序
50
服务程序
50
子系统
48
子系统
48
子系统接口

图 10

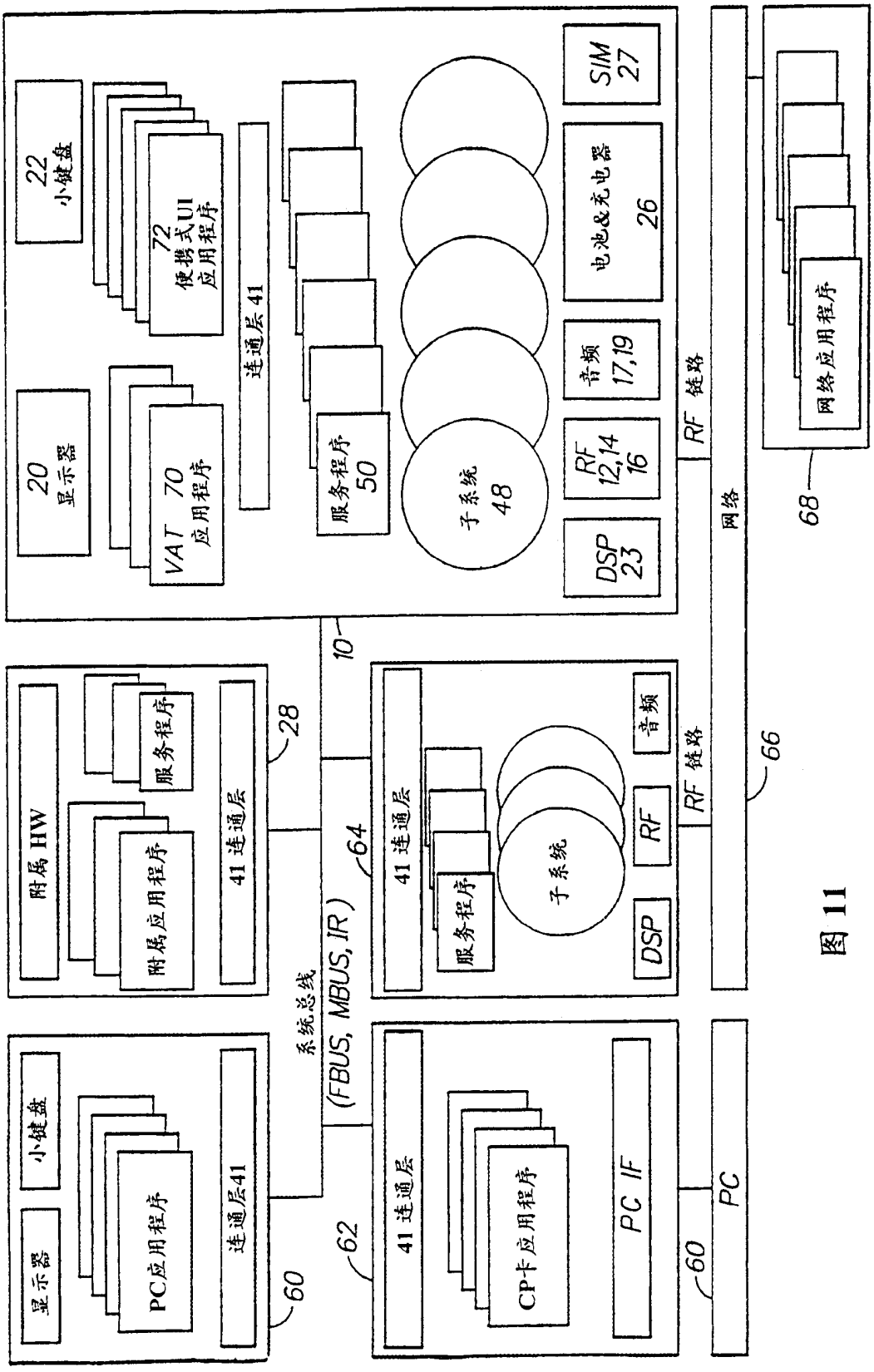

显示器
小键盘
PC应用程序
连通层41
60
系统总线
(FBUS, MBUS, IR)
62
41 连通层
CP卡应用程序
PC IF
60
PC
附属 HW
附属应用程序
服务程序
41 连通层
28
10
64
41 连通层
服务程序
子系统
DSP
RF
音频
RF 链路
66
网络
20
显示器
22
小键盘
72
便携式UI
应用程序
VAT 70
应用程序
连通层 41
服务程序
50
子系统
48
DSP
23
RF
12,14
16
音频
17,19
电池&充电器
26
SIM
27
RF 链路
网络
68
网络应用程序

图 11

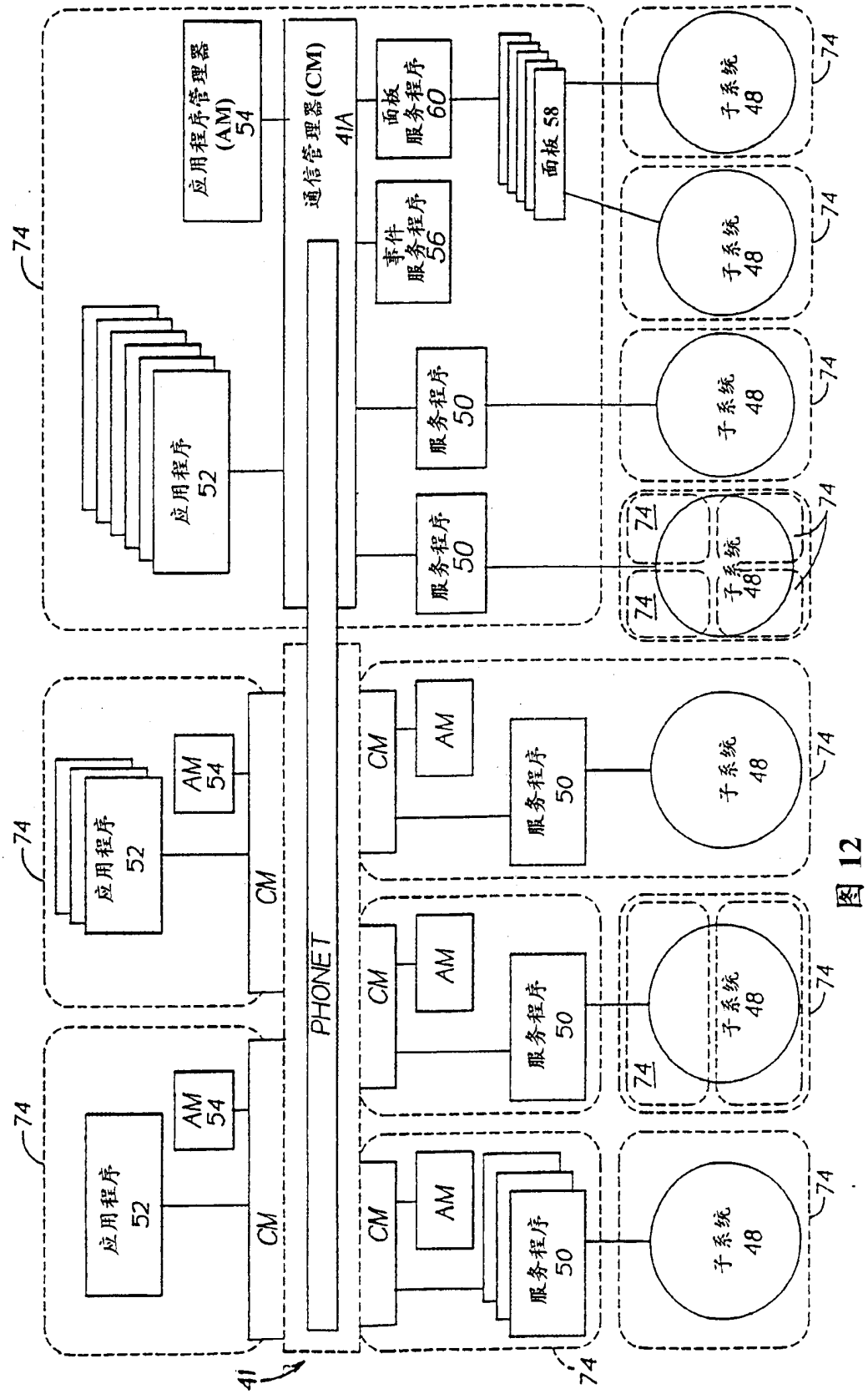

74
应用程序管理器 (AM) 54
通信管理器(CM) 41A
面板服务程序 60
事件服务程序 56
面板 58
应用程序 52
服务程序 50
子系统 48
AM 54
CM
PHONET
41

图 12

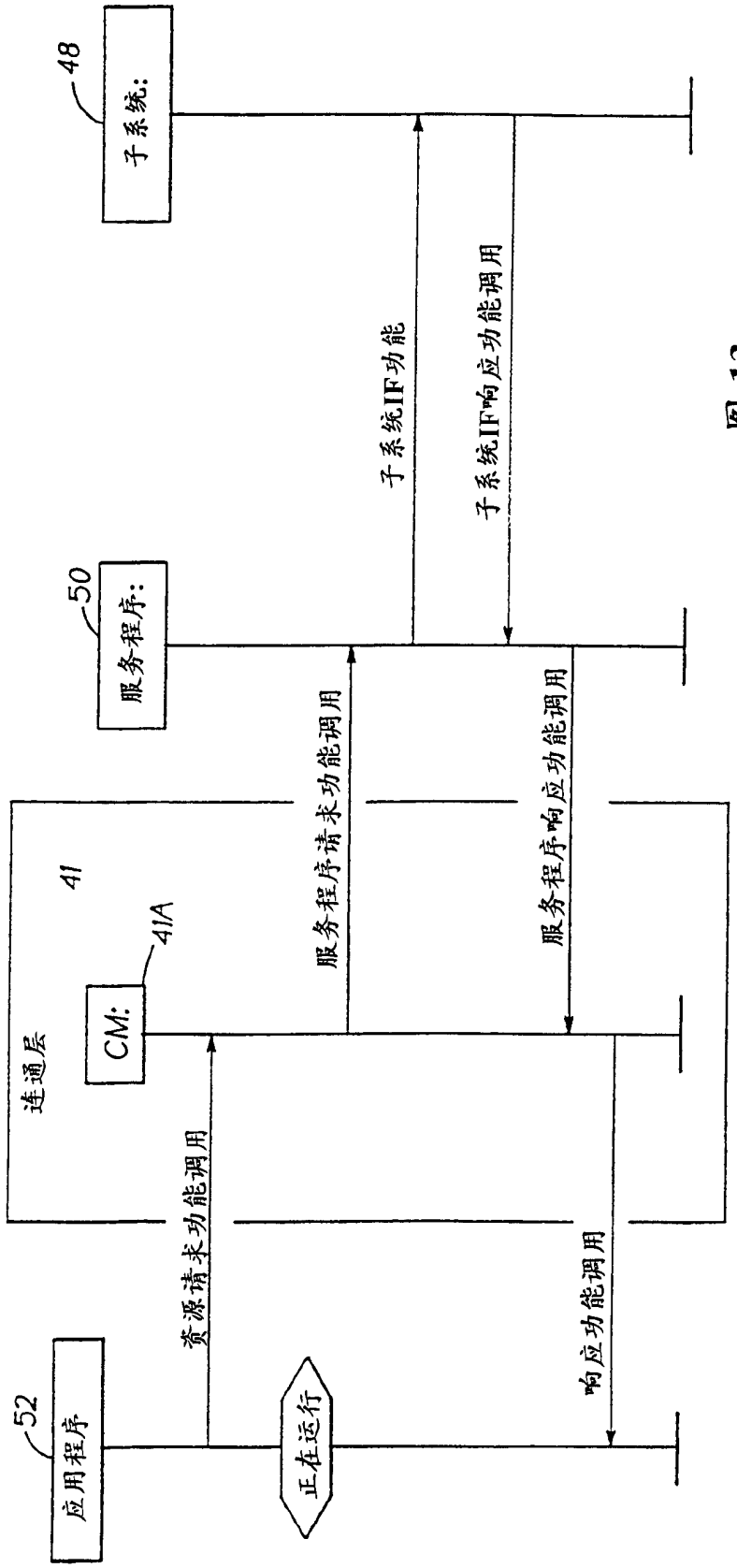

48
子系统:
子系统IF功能
子系统IF响应功能调用
50
服务程序:
服务程序请求功能调用
服务程序响应功能调用
41
连通层
41A
CM:
资源请求功能调用
响应功能调用
52
应用程序
正在运行

图 13

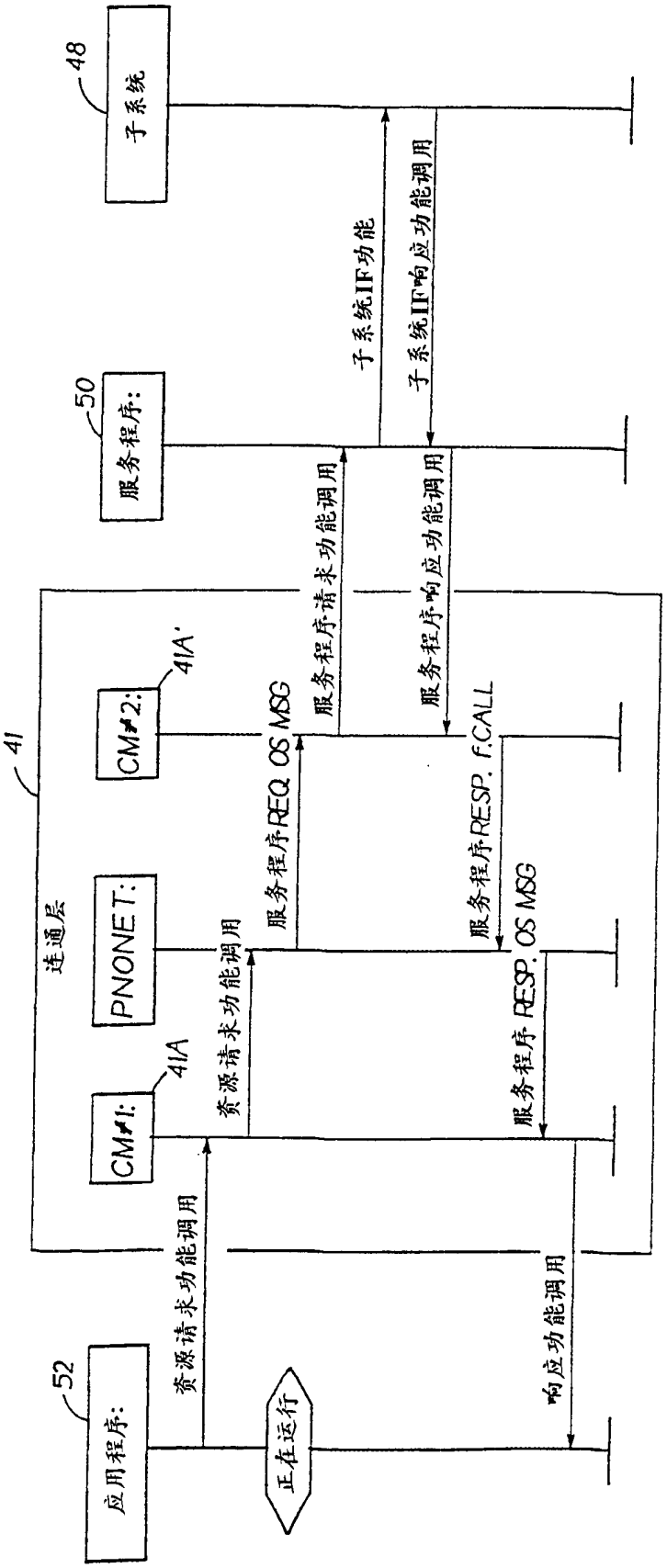

48
子系统
50
服务程序:
子系统IF功能
子系统IF响应功能调用
服务程序请求功能调用
服务程序响应功能调用
41
连通层
CM#2:
41A'
PNONET:
CM#1:
41A
服务程序REQ OS MSG
资源请求功能调用
服务程序RESP. F.CALL
服务程序 RESP. OS MSG
资源请求功能调用
响应功能调用
52
应用程序:
正在运行

图 14

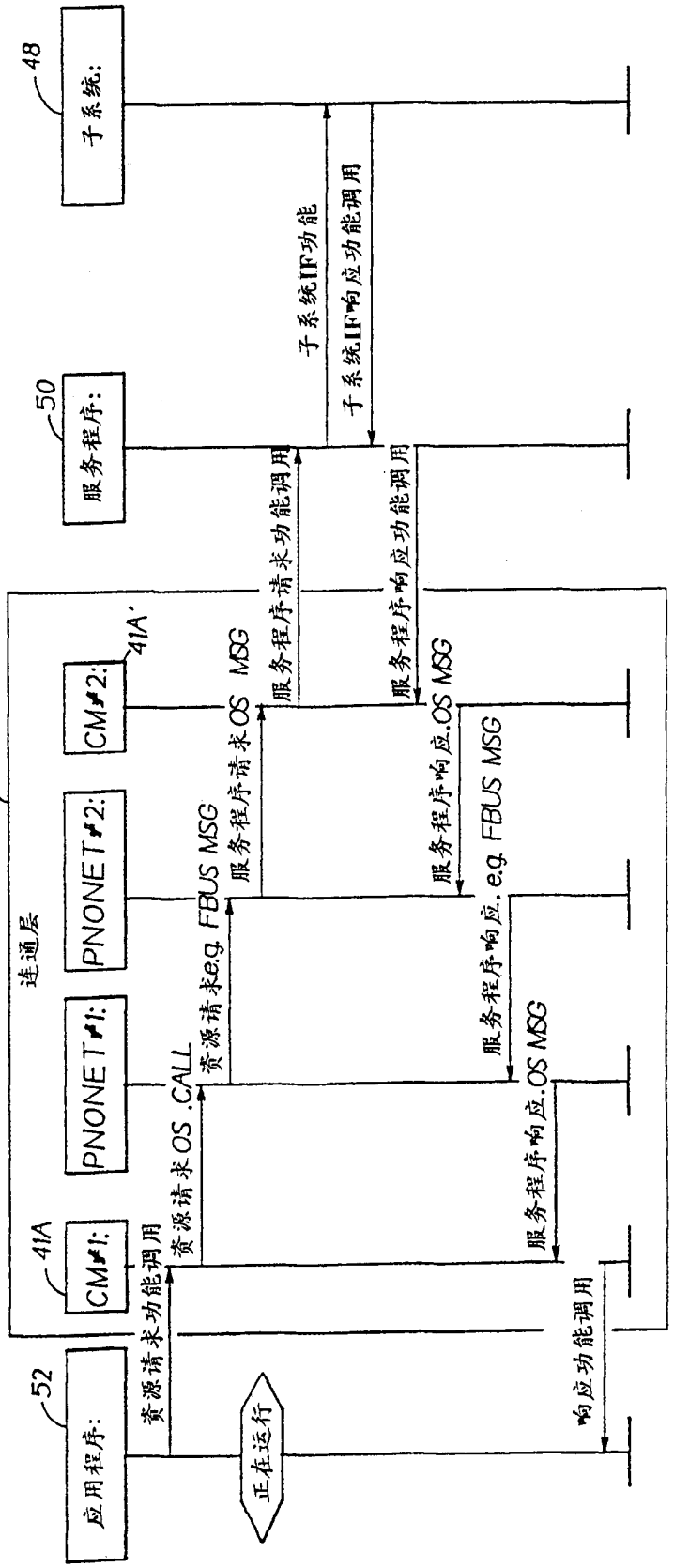

48
子系统:
50
服务程序:
41
连通层
41A'
CM#2:
PNONET#2:
PNONET#1:
41A
CM#1:
52
应用程序:
正在运行
子系统IF功能
子系统IF响应功能调用
服务程序请求功能调用
服务程序响应功能调用
服务程序请求OS MSG
服务程序响应.OS MSG
资源请求e.g. FBUS MSG
服务程序响应. e.g. FBUS MSG
资源请求OS .CALL
服务程序响应.OS MSG
资源请求功能调用
响应功能调用

图 15

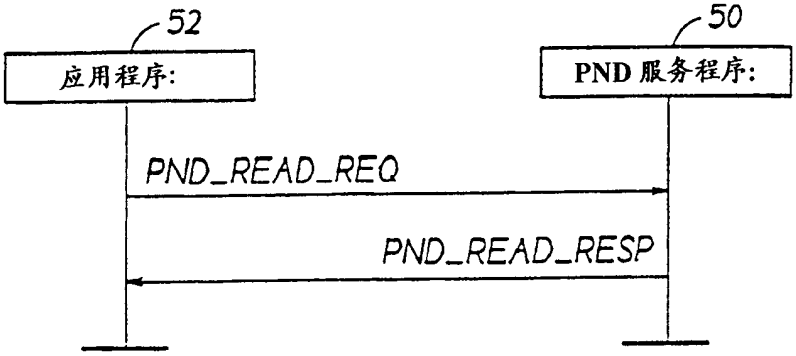

52
应用程序:
50
PND 服务程序:
PND_READ_REQ
PND_READ_RESP

图 16

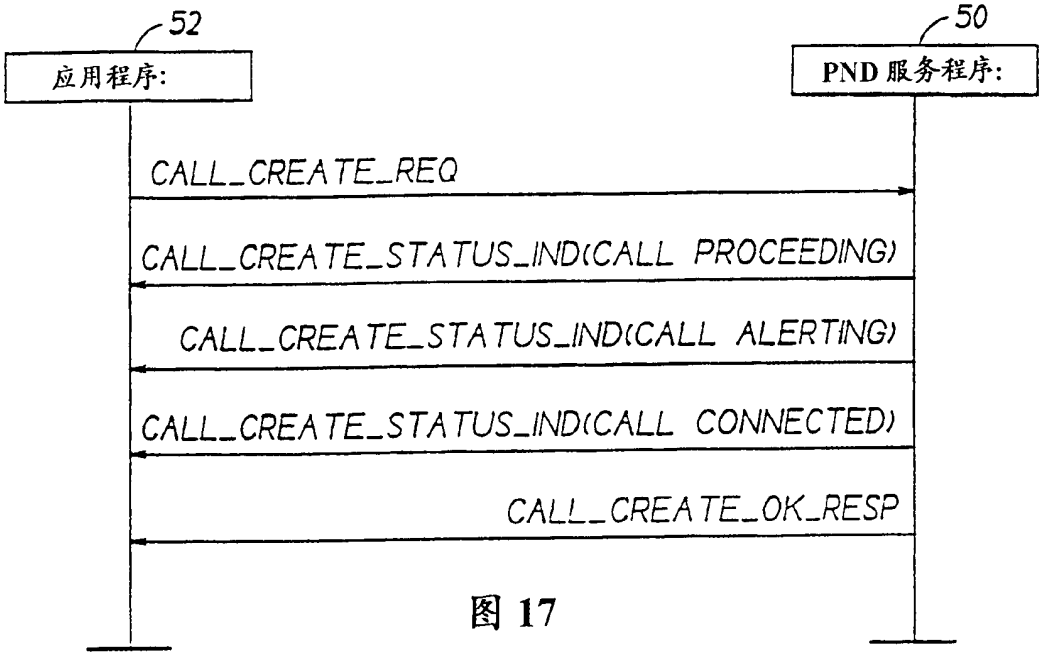

52
应用程序:
50
PND 服务程序:
CALL_CREATE_REQ
CALL_CREATE_STATUS_IND(CALL PROCEEDING)
CALL_CREATE_STATUS_IND(CALL ALERTING)
CALL_CREATE_STATUS_IND(CALL CONNECTED)
CALL_CREATE_OK_RESP

图 17

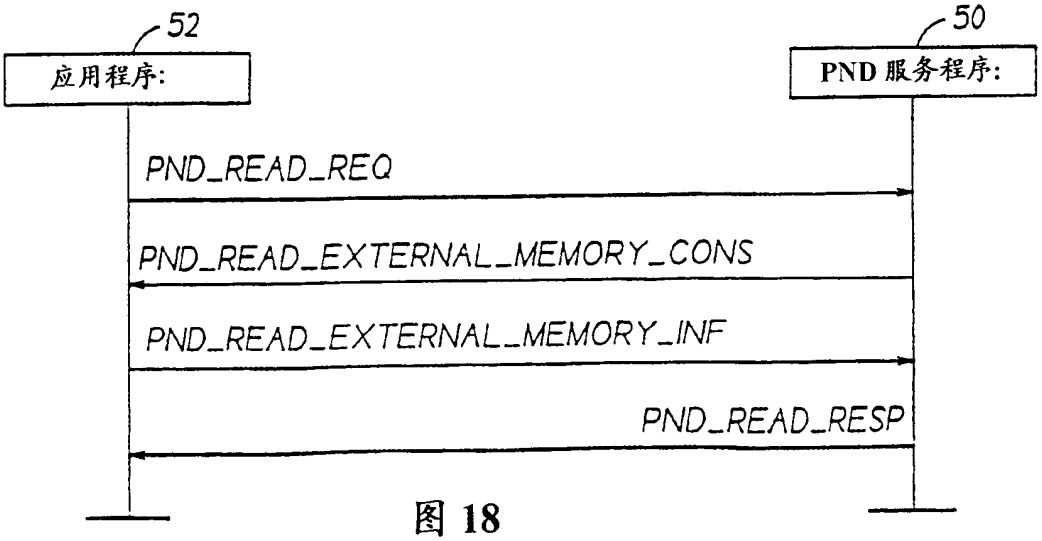

52
应用程序:
50
PND 服务程序:
PND_READ_REQ
PND_READ_EXTERNAL_MEMORY_CONS
PND_READ_EXTERNAL_MEMORY_INF
PND_READ_RESP

图 18

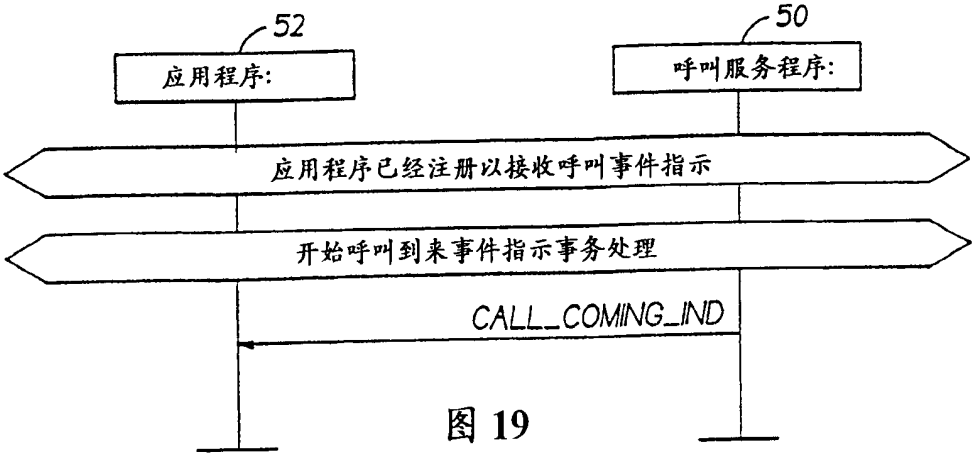

52
应用程序:
50
呼叫服务程序:
应用程序已经注册以接收呼叫事件指示
开始呼叫到来事件指示事务处理
CALL_COMING_IND

图 19

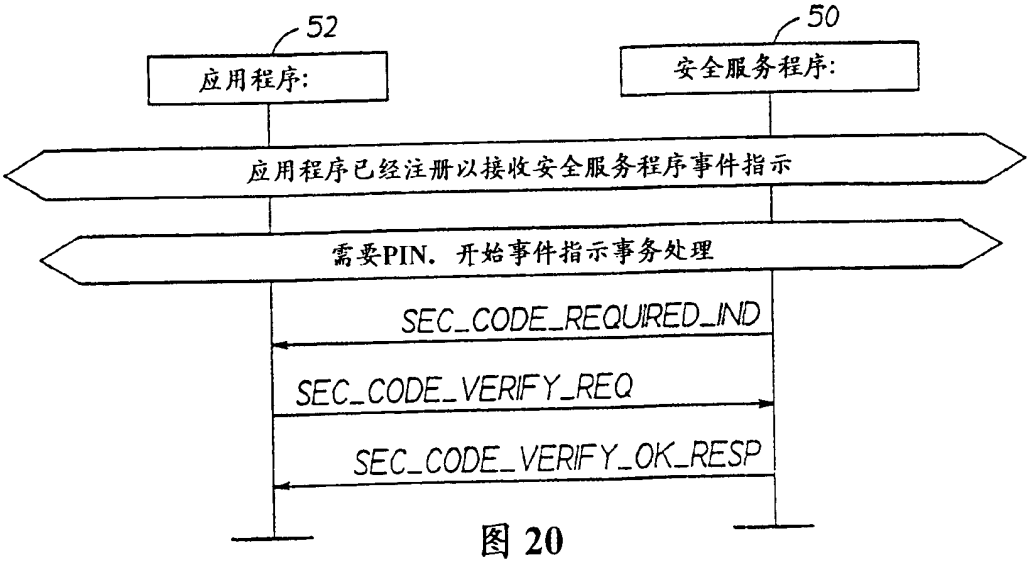

52
应用程序:
50
安全服务程序:
应用程序已经注册以接收安全服务程序事件指示
需要PIN. 开始事件指示事务处理
SEC_CODE_REQUIRED_IND
SEC_CODE_VERIFY_REQ
SEC_CODE_VERIFY_OK_RESP

图 20

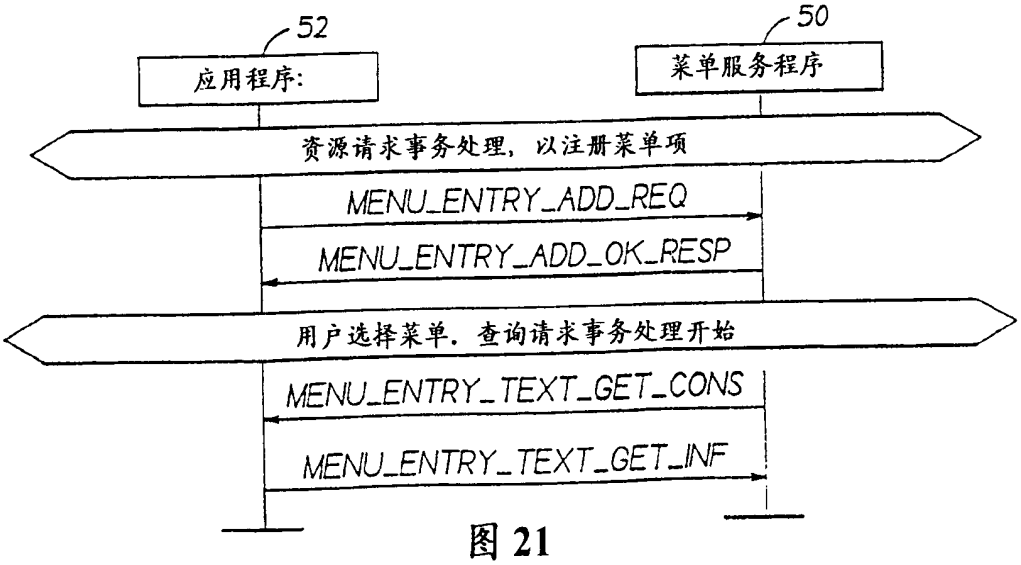

52
应用程序:
50
菜单服务程序
资源请求事务处理，以注册菜单项
MENU_ENTRY_ADD_REQ
MENU_ENTRY_ADD_OK_RESP
用户选择菜单. 查询请求事务处理开始
MENU_ENTRY_TEXT_GET_CONS
MENU_ENTRY_TEXT_GET_INF

图 21

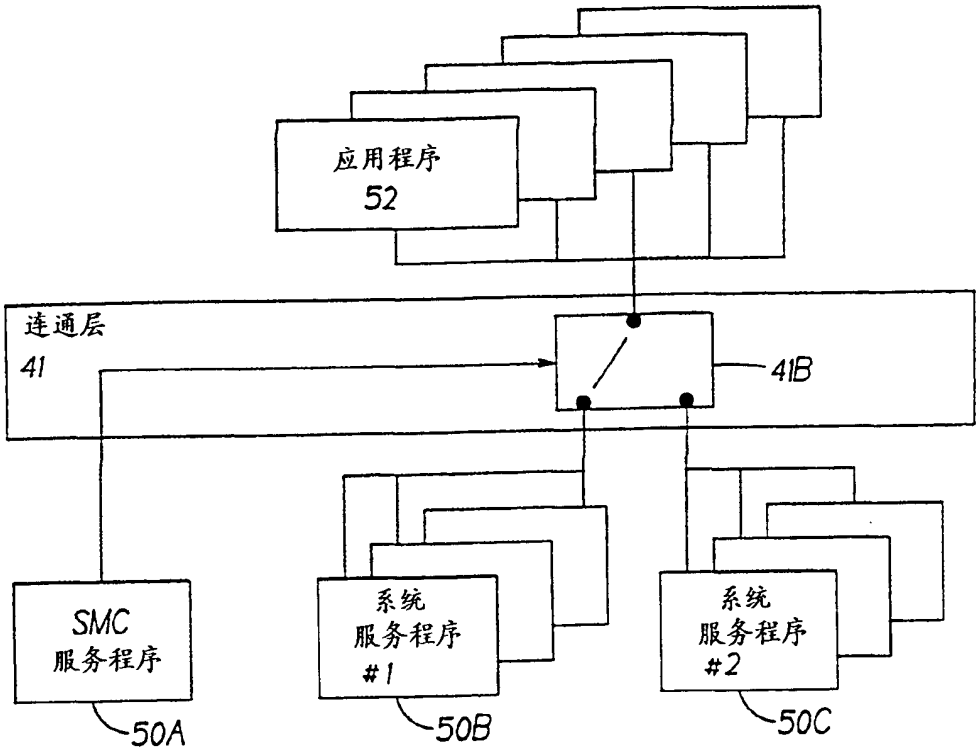

应用程序
52
连通层
41
41B
SMC
服务程序
50A
系统
服务程序
#1
50B
系统
服务程序
#2
50C

图 22A

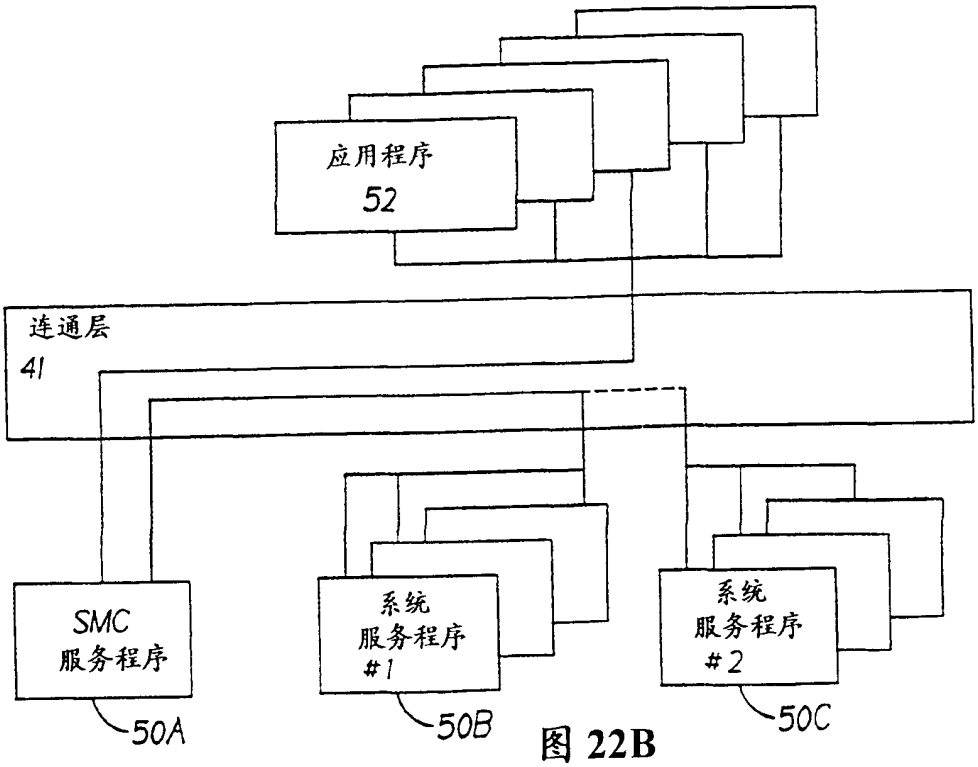

应用程序
52
连通层
41
SMC
服务程序
50A
系统
服务程序
#1
50B
系统
服务程序
#2
50C

图 22B