

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 9/54 (2006.01)

G06F 9/46 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200610109253.8

[43] 公开日 2007年1月3日

[11] 公开号 CN 1889048A

[22] 申请日 2006.8.3

[21] 申请号 200610109253.8

[71] 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

[72] 发明人 饶俊 杨明磊 裘洪国

[74] 专利代理机构 北京集佳知识产权代理有限公司
代理人 逯长明

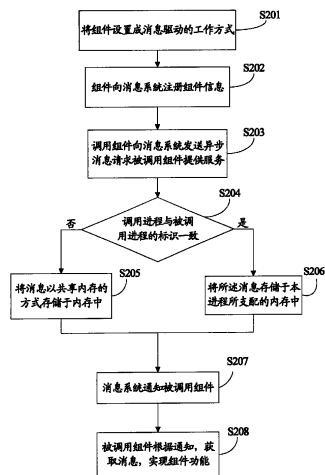
权利要求书 2 页 说明书 13 页 附图 3 页

[54] 发明名称

组件的实现方法以及系统

[57] 摘要

本发明公开了一种组件实现的方法以及系统，本方法包括以下步骤：调用组件向消息系统发送用于请求被调用组件提供服务的消息，所述消息携带调用进程标识、被调用组件标识和被调用进程标识；如果所述消息携带的调用进程标识异于被调用进程标识，则将所述消息存储于所述进程均可访问的内存，否则，将所述消息存储于所述进程所支配的内存；所述消息系统通知所述被调用组件提供服务；所述被调用组件根据通知，获取所述消息，并根据所述消息实现组件功能。本发明实现了组件在线程模型以及进程模型中的重用，并且，使得组件在线程和进程模型中均有高的效率。



1、一种组件的实现方法，其中所述组件采用基于消息驱动的工作方式，包括以下步骤：

调用组件向消息系统发送用于请求被调用组件提供服务的消息，所述消息携带调用进程标识、被调用组件标识和被调用进程标识；

如果所述消息携带的调用进程标识异于被调用进程标识，则将所述消息存储于所述进程均可访问的内存，否则，将所述消息存储于所述进程所支配的内存；

所述消息系统通知所述被调用组件提供服务；

所述被调用组件根据通知，获取所述消息，并根据所述消息实现组件功能。

2、根据权利要求1所述组件的实现方法，其特征是，所述的进程均可访问的内存采用消息队列的内存管理方式。

3、根据权利要求1所述组件的实现方法，其特征是，所述的进程均可访问的内存采用共享内存的内存管理方式。

4、根据权利要求1、2或3所述组件的实现方法，其特征是，在所述发送消息步骤中，所述调用组件以异步的方式发送所述消息。

5、根据权利要求1、2或3所述组件的实现方法，其特征是，所述消息系统通知所述被调用组件提供服务具体是：

所述消息系统获取被调用组件的消息处理入口函数，向所述消息处理入口函数发送携带所述消息内存地址的通知，以指示所述被调用组件获取所述消息。

6、根据权利要求5所述组件的实现方法，其特征是，所述消息系统获取被调用组件的消息处理入口函数，具体包括以下步骤：

组件向消息系统注册组件信息，所述组件信息包括：所述组件标识、组件对应的进程标识以及组件消息处理入口函数；

所述消息系统根据所述被调用组件标识，查询获取所述被调用组件的消息处理入口函数。

7、根据权利要求1、2或3所述组件的实现方法，其特征是，所述组件标识由所述消息系统动态分配。

8、根据权利要求1、2或3所述组件的实现方法，其特征是，所述组件标识由所述组件根据预定的规则自动生成。

9、一种组件的实现系统，包括组件存储器以及消息系统，所述消息系统用于管理所述组件的通信消息，所述组件存储器用于存储组件，其特征是：所述组件存储器与所述的消息系统相连接，所述组件根据所述消息系统中的消息实现组件功能，所述消息系统包括：

两个消息池，用于存储消息，其中第一消息池设置于各进程均可访问的内存中，第二消息池设置于所述各进程所支配的内存中；

消息接收单元，用于接收调用组件发送的用于请求被调用组件提供服务的消息，其中所述消息携带被调用组件标识、调用进程标识、被调用进程标识；

消息分发单元，用于将所述消息分发于所述消息池中，如果所述消息携带的调用进程标识异于所述被调用进程标识，则将所述消息发送至所述第一消息池，否则将所述消息发送至所述第二消息池；

消息发送单元，用于根据所接收的消息，向所述被调用组件发送通知，指示被调用组件访问所述消息池获取所述消息。

10、根据权利要求9所述组件的实现系统，其特征是，所述进程均可访问的内存采用消息队列的内存管理方式。

11、根据权利要求9所述组件的实现系统，其特征是，所述进程均可访问的内存采用共享内存的内存管理方式。

12、根据权利要求9、10或11所述组件的实现系统，其特征是，所述消息系统进一步包括：

组件注册单元，用于组件向所述系统注册组件信息，其中所述组件信息包括组件标识，以及组件的消息处理入口函数。

13、根据权利要求9、10或11所述组件的实现系统，其特征是，所述组件实现系统进一步包括：

组件标识生成单元，与所述组件存储器相连接，用于按照预定的规则生成所述组件的组件标识。

组件的实现方法以及系统

技术领域

本发明涉及计算机领域，特别的涉及一种组件的实现方法以及系统。

背景技术

组件是一种可重复使用的对象或程序，这种对象或程序可执行特定的功能，且设计为可使用其他组件以及应用程序。随着软件规模的越来越大，组件化的开发方式也越来越得到广泛的应用。使用组件化的软件开发方式不但能够大大减少软件功能开发的重复性，还大大地增加了软件开发的灵活性。

线程模型以及进程模型是目前常用的两种软件运行模型，由于两种模型各有各的优点，因而各有各的应用场合。表一所示为线程模型以及进程模型的特点对照表：

项目	线程模型	进程模型
调度开销	小	由于需要切换表页表，刷新高速缓存，而造成调度开销较大
地址空间	共享，线程间可以互相访问对方的地址空间	隔离，进程间不能访问对方的地址空间

表一

由表一可以看出，线程模型由于调度开销小，实时性高，因而适合对实时性要求高，而安全性要求不严格的场合；而进程模型的最大优点是地址空间隔离，安全性好，因而特别适合对实时性要求相对较低，而对安全要求高的场合。

因而，对于组件的设计来说，由于组件的运行模型可能采用进程模型，亦可能采用线程模型，因而，如何实现组件在进程模型以及线程模型下的组件重用，是组件设计的一关键技术。

为了解决上述的技术问题，现有技术通常采用以下的两种技术方案：第一

种是，基于函数调用方式的直接函数调用和远程过程调用（Remote Procedure Call，简称 RPC）方式相结合方式；第二种是，基于队列的进程和线程无缝切换的组件实现方式。

对于第一种技术方案，对于进程内的线程间的调用使用直接函数调用方式，而对于进程间的调用使用 RPC 方式。

RPC 采用客户机与服务器的模式，请求程序就是一个客户机，而服务提供程序就是一个服务器。其基本执行过程如下：

在客户端，调用进程发送一个包含进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息的到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用过程接收答复信息，获得进程结果，然后调用执行继续进行下一任务。

然而基于 RPC 的实现方式，客户端在调用信息发出后，必须阻塞等待答复回应，而不能处理其他的事务，因而造成系统运行速度较慢。

另外的，RPC 必须使用接口定义语言（Interface Definition Language，简称 IDL）来定义函数接口，然后使用编译器来生成客户端的桩（Stub）和服务器的框架（Skeleton）。客户端的程序与桩一起编译链接，服务器的程序和框架一起编译链接，客户端的桩和服务器的框架之间实现远程通讯。RPC 实现的复杂性也一定程度的限制了该技术的应用。

第二种技术方案是目前应用较为广泛的一种技术方案，在许多支持线程模型和进程模型的操作系统上，队列是一种比较常用的通信方式，例如 Vxwork 上队列（queue）、Linux 上的消息（Message，简称 msg）等等。

以 Linux 操作系统为例，为了实现组件在 Linux 操作系统，即可以运行在线程模型，又可以运行在进程模型中，将组件设计成队列作为组件的输入和输出。组件的输入为队列中的数据，组件的输出为写入到其他队列中的数据，这个队列的信息可以在输入的数据中指定。由于 Linux 操作系统上的队列 msg 是一种进程间的通信机制，通过将组件设计成以队列输入为驱动的消息处理方

式；同样该消息队列也可以用于进程内的线程间的通信。本技术方案无论组件的应用软件采用进程或线程的模型，不用改变组件的本身，而实现了组件的重用。

然而，由于本技术方案对于线程间的通信也采用了与进程间通信的队列方式，造成不必要的内存拷贝，访问队列的开销增大，造成线程间通信的效率大大减低。

发明内容

本发明要解决的技术问题是提供一种组件的实现方法，实现组件在线程模型以及进程模型中的重用，并且，使得组件在线程和进程模型中均有高的效率。

本发明要解决的技术问题是还提供一种组件的实现系统，实现组件在线程模型以及进程模型中的重用，并且，使得组件在线程和进程模型中均有高的效率。

为解决上述第一技术问题，本发明目的是通过以下技术方案实现的：

一种组件的实现方法，其中所述组件采用基于消息驱动的工作方式，包括以下步骤：

调用组件向消息系统发送用于请求被调用组件提供服务的消息，所述消息携带调用进程标识、被调用组件标识和被调用进程标识；

如果所述消息携带的调用进程标识异于被调用进程标识，则将所述消息存储于所述进程均可访问的内存，否则，将所述消息存储于所述进程所支配的内存；

所述消息系统通知所述被调用组件提供服务；

所述被调用组件根据通知，获取所述消息，并根据所述消息实现组件功能。

本发明所述方法中，优选地，所述的进程均可访问的内存采用消息队列的内存管理方式。

本发明所述方法中，优选地，所述的进程均可访问的内存采用共享内存的内存管理方式。

本发明所述方法中，优选地，在所述发送消息步骤中，所述调用组件以异步的方式发送所述消息。

本发明所述方法中，优选地，所述消息系统通知所述被调用组件提供服务具体是：

所述消息系统获取被调用组件的消息处理入口函数，向所述消息处理入口函数发送携带所述消息内存地址的通知，以指示所述被调用组件获取所述消息。

本发明所述方法中，优选地，所述消息系统获取被调用组件的消息处理入口函数，具体包括以下步骤：

组件向消息系统注册组件信息，所述组件信息包括：所述组件标识、组件对应的进程标识以及组件消息处理入口函数；

所述消息系统根据所述被调用组件标识，查询获取所述被调用组件的消息处理入口函数。

本发明所述方法中，优选地，所述组件标识由所述消息系统动态分配。

本发明所述方法中，优选地，所述组件标识由所述组件根据预定的规则自动生成。

为解决上述第二技术问题，本发明目的是通过以下技术方案实现的：

一种组件的实现系统，包括组件存储器以及消息系统，所述消息系统用于管理组件之间的通信消息，所述组件存储器用于存储组件，所述组件存储器与所述的消息系统相连接，所述组件根据所述消息系统中的消息实现组件功能，所述消息系统包括：

两个消息池，用于存储消息，其中第一消息池设置于各进程均可访问的内存中，第二消息池设置于所述各进程所支配的内存中；

消息接收单元，调用组件发送的用于请求被调用组件提供服务的消息，其中所述消息携带被调用组件标识、调用进程标识、被调用进程标识；

消息分发单元，用于将所述消息分发于所述消息池中，如果所述消息携带

的调用进程标识异于所述被调用进程标识,则将所述消息发送至所述第一消息池,否则将所述消息发送至所述第二消息池;

消息发送单元,用于根据所接收的消息,向所述被调用组件发送通知,指示被调用组件访问所述消息池获取所述消息。

本发明所述系统中,优选地,所述进程均可访问的内存采用消息队列的内存管理方式。

本发明所述系统中,优选地,所述进程均可访问的内存采用共享内存的内存管理方式。

本发明所述系统中,优选地,所述消息系统进一步包括:

组件注册单元,用于组件向所述系统注册组件信息,其中所述组件信息包括组件标识,以及组件的消息处理入口函数。

本发明所述系统中,优选地,所述组件实现系统进一步包括:

组件标识生成单元,与所述组件存储器相连接,用于按照预定的规则生成所述组件的组件标识。

以上第一技术方案可以看出,本发明与现有技术相比较,由于本发明在组件实现时,对于进程间的通信,则将该消息存储于所述进程均可访问的内存中,使得各进程可以相互访问进程间的通信消息;而对于进程内线程间的通信,则将消息存储于本进程所支配的内存中。可见本发明实现了组件在进程模型以及线程模型下的无缝切换重用,提高了组件的重用性,同时,保证了进程间或进程内的线程间的通讯均有高的通信效率。

进一步的,对于进程间的通信,该所述进程均可访问的内存采用消息队列管理方式,使得各进程以消息队列的方式进行通信。使得进程间的通信简单,实施方便。

进一步的,对于进程间的通信,则将消息存储于采用共享内存的管理方式的内存,避免了进程间通信的消息的内存拷贝,相对于采用消息队列的方式而言,进一步的加快了进程间通信的效率。

进一步的，调用组件采用异步的方式发送消息，那么组件在发送消息后，可以继续执行其他任务，而无须阻塞等待应答消息，因此进一步的提高了通信效率。

以上第二技术方案可以看出，本发明与现有技术相比较，由于本发明消息系统在内存中设置了采用两种内存管理方式的消息池，一种为进程间可互相访问的消息池，另一种为分布设置与个进程内所支配的内存中的消息池。对于进程间的通信，则将消息存储于进程间可互相访问的消息池；而对于进程内线程间的通信，则将消息存储于设置于本进程内的消息池。可见本发明实现了组件在进程模型以及线程模型下的无缝切换重用，提高了组件的重用性，同时，保证了进程间或进程内的线程间的通讯均有高的通信效率。

进一步的，该进程间可互相访问的消息池采用消息队列的管理方式，使得各进程以消息队列的方式进行通信。使得进程间的通信简单，实施方便。

进一步的，该进程间可互相访问的消息池采用消息队列的管理方式，进一步的加快了进程间通信的效率。

另外，由于本发明所提供的系统规模小，占用资源少，复杂度低，在实际中易于实施。

附图说明

图1为本发明实施例1的方法流程示意图；

图2为本发明实施例2的方法流程示意图；

图3为本发明实施例3的消息系统结构示意图；

图4为本发明实施例4的消息系统结构示意图。

图号说明：

300: 消息系统；	301: 消息接收单元；	302: 组件注册单元；
303: 消息分发单元；	304: 消息发送单元；	305: 第一消息池；
306: 第二消息池；	307: 组件标识生成单元；	400: 消息系统；

405: 第一消息池。

具体实施方式

本发明的核心思想是，将组件设计成消息驱动的工作方式，在组件的调用时，如果该调用为进程间的调用，则将消息存储在进程间可互相访问的内存中；如果该调用为进程内的线程间的调用，则将消息存储在本进程所支配的内存中。

为了使本领域的技术人员更好的理解本发明的技术方案，下面结合附图以及具体实施例进行详细的描述：

实施例1：

图1所示为本实施例的方法流程示意图，如图示：本方法包括以下步骤：

步骤S101：将组件设置成消息驱动的工作方式。

其中该消息驱动的方式具体是，如果组件要使用其它线程或其他进程的组件提供服务，那么这个组件向消息系统发送请求消息，该组件发送请求消息后，可以继续执行其他任务，提供服务的组件根据该请求消息，执行相应的功能，并返回应答消息到发送请求消息的组件，发送请求消息的组件接收到提供服务的组件返回的应答消息后，处理执行的结果。

步骤S102：组件向消息系统注册组件信息。

每个组件均有唯一的标识，称为组件标识。该组件标识的确定既可以根据同一的规划来确定和分配各个组件的组件标识，还可以通过预定义组件标识的规则，使各组件根据该规则自行生成各组件标识；还可以在组件运行时，由消息系统动态为运行的各个组件分配各组件标识。

组件在开始运行时，向消息系统注册组件的信息，该信息包括唯一标识组件的组件标识、组件的消息处理入口函数、以及组件的其它信息，比如组件所对应的进程标识，及组件消息处理入口函数所在的进程标识等。

步骤S103：调用组件向消息系统发送异步消息请求被调用组件提供服务。

当某组件需要调用其它进程或其他线程的组件提供服务时，则通过异步的

方式，向消息系统发送消息，该消息携带需要提供服务的被调用组件的组件标识、以及提供服务的被调用进程的进程标识，消息系统接收到该消息后，根据消息携带的被调用组件标识，获取该组件标识所对应的注册信息，获取该提供服务的组件的消息处理入口函数以及其他信息。

步骤S104: 如果消息携带的调用进程以及被调用进程的标识一致，则执行步骤S105；否则，执行步骤S106。

如果消息所携带的调用进程标识以及被调用进程标识一致，则当前的调用为进程内线程间的调用，则执行步骤S106；否则，当前的调用为进程间的调用，执行步骤S105。

步骤S105: 将消息以消息队列的方式存储于内存中。

如果当前的调用为进程间的调用，则消息系统将该消息以消息队列的方式存储于内存中。将消息设计成队列作为组件的输入和输出。组件的输入为队列中的消息，组件的输出可以为写入到其他队列中的数据。

步骤S106: 将所述消息存储于本进程所支配内存中。

如果当前的调用为进程内的线程间的调用，则消息系统将该消息存储在本进程所支配的内存中。

步骤S107: 消息系统通知被调用组件。

消息系统接收到所述消息后，根据消息携带的被调用的组件标识，获取该被调用组件所注册的消息入口函数，并向被调用组件的消息入口处理函数，发送携带该消息地址的通知，通知被调用组件访问内存中的消息，以使组件按照消息指示，执行相应的进程或线程，实现组件功能。

步骤S108: 被调用组件根据通知，获取消息，实现组件功能。

被调用组件的消息处理入口函数，根据通知携带的消息地址，访问内存获取该消息。

如果该消息以消息队列的方式存储在内存中，那么被调用组件的消息处理入口函数根据消息地址，将该消息拷贝到接收任务的消息接收区域中，然后读

取该消息;

如果该消息存储在本进程的内存中,则被调用组件消息处理入口函数根据通知携带的消息地址,直接访问本进程内存,读取该消息。

被调用组件的消息处理入口函数获取该消息后,根据该消息指示,执行相应的进程或线程。

如果该调用需要被调用组件需要在被调用进程或线程执行完毕后,向调用组件返回结果的,则进一步的,如果该调用为进程间的调用时,被调用组件将结果以消息队列的方式,发送至消息系统的内存队列中,并且,消息系统向调用组件发送携带该结果消息的地址或队列标识的通知,通知调用组件根据该结果消息的地址或队列标识,从消息队列中将该结果消息拷贝至调用组件的消息接收区域内,调用组件的消息处理入口函数在本消息接收区域内读取该结果;如果该调用为进程内线程间的调用时,则被调用组件结果以消息的方式,向消息系统发送并存储至本进程内存中,并且,消息系统向调用组件发送携带该结果消息的地址通知,通知调用组件根据该结果消息的地址,访问本进程内存读取该结果消息。调用组件读取该结果消息后,处理执行的结果。

由以上可以看出,由于进程内线程间的空间是可以相互访问的,因而本发明,对于进程内线程间的调用消息,直接存储于本进程的内存;只对进程间的调用消息,采用消息队列的内存管理方式存储。

因而,本发明解决了现有技术中由于无论调用是进程间或线程内的调用,均使用消息队列内存管理方式存储,而导致当当前调用为进程内线程间的调用时,线程模型的优点得不到发挥得问题,保证了当前调用为进程内线程间的调用时,线程模型的优点得以发挥,大大提高了组件调用的实现效率。

实施例2:

图2所示为本实施例的方法流程示意图,如图示:本方法包括以下步骤:

步骤S201: 将组件设置成消息驱动的工作方式。

步骤S202: 组件向消息系统注册组件信息。

步骤S203: 调用组件向消息系统发送异步消息请求被调用组件提供服务。

以上步骤以实施例1中的步骤S101、S102、S103同理。

步骤S204: 如果消息携带的调用进程以及被调用进程的标识一致, 则执行步骤S205; 否则, 执行步骤S206。

步骤S205: 将消息以共享内存的方式存储于内存中。

如果当前的调用为进程间的调用, 则消息系统将该消息存储于系统预设的采用共享内存的方式的内存中。

共享内存管理方式的内存是通过直接操作内存映射文件来进行的, 而内存映射文件是进行数据共享的最低层机制, 共享内存可以以较小的开销获取较高的性能。

步骤S206: 将所述消息存储于本进程所支配的内存中。

如果当前的调用为进程内的线程间的调用, 根据进程模型的空间地址特点, 进程内的地址空间可以互相访问, 则消息系统将该消息存储在本进程的内存中。

步骤S207: 消息系统通知被调用组件。

本步骤与实施例1中的步骤S107同理。

步骤S208: 被调用组件根据通知, 获取消息, 实现组件功能。

被调用组件的消息处理入口函数, 根据通知携带的消息地址, 访问内存获取该消息。如果该消息以共享内存的方式存储在内存中, 则通过直接操作内存映射文件来进行的; 如果该消息是存储于本进程所支配的内存中, 则直接在进程内访问该内存地址读取该消息。

被调用组件的消息处理入口函数获取该消息后, 根据该消息指示, 执行相应的进程或线程。

如果该调用需要被调用组件需要在被调用进程或线程执行完毕后, 向调用组件返回结果的, 则进一步的, 如果该调用为进程间的调用时, 被调用组件将

结果以共享内存的方式存储于内存中，并且，消息系统向调用组件发送携带该结果消息的地址通知，通知调用组件根据该结果消息的地址，通过直接操作内存映射文件读取该结果消息，调用组件的消息处理入口函数读取该结果；如果该调用为进程内线程间的调用时，则被调用组件界结果以消息的方式向消息系统发送并存储至本进程的内存中，并且，消息系统向调用组件发送携带该结果消息的地址通知，通知调用组件根据该结果消息的地址，进程内访问内存地址读取该结果消息。调用组件读取该结果消息后，处理执行的结果。

由以上可以看出，本发明对于进程内线程间的调用，由于进程内线程间的空间是可以相互访问的，因而对线程间通信的消息采用普通内存的管理方式来存储管理，保持了线程模型的效率高的优点；对于进程间的调用，由于进程间的空间是隔离的，因而对进程间通信的消息采用共享内存的方式来存储管理，相对于实施例1中的进程间的队列方式，减少了内存拷贝的次数，提高了效率。特别的对于消息包含大量数据时，本发明技术方案的效果更加显著。

实施例3:

图3所示为本实施例的消息系统结构示意图，如图示，消息系统300:

组件注册单元302，与存储组件的组件存储器相连接，用于组件向消息系统注册组件信息，包括组件的组件标识，组件的消息处理入口函数、以及组件对应的进程或线程数以及组件对应的进程标识或线程标识等组件的其他信息。消息系统根据组件标识识别组件。

消息接收单元301，与组件存储器相连接，用于接收组件向消息系统发送的消息。

当某组件中的进程或线程需要调用其它组件或本组件内的进程或线程时，向消息系统的消息接收单元301发送由被调用组件（即可以为该本组件，亦可以为其它组件）处理的消息，该消息包含调用组件以及被调用组件的组件标识、调用进程以及被调用进程的进程标识。

消息分发单元303，与消息接收单元301、第二消息池306、第一消息池305

分别相连接,用于对消息接收单元301所接收的消息分发到第一消息池305或第二消息池306。

消息分发单元303根据消息接收单元301所接收的消息携带的调用进程以及被调用进程标识,如果该调用进程以及被调用进程标识不一致,则表明该调用为进程间的调用,消息分发单元303将该消息发送至基于队列方式的第一消息池305中存储;如果该调用进程以及被调用进程标识一致,则表明当前调用为进程内线程间的调用,消息分发单元303将该消息发送至设置于本进程内的第二消息池306中存储。

第一消息池305,与消息分发单元303以及消息发送单元304分别连接,该消息池采用基于消息队列的内存管理方式,用于对消息分发单元303分发的消息进行存储管理。

第二消息池306,与消息分发单元303以及消息发送单元304分别连接,该消息池由设置于各进程所支配的内存部分或全部所组成,用于对消息分发单元303分发的消息进行存储管理。

消息发送单元304,与第一消息池305、第二消息池306、以及组件注册单元302分别相连接,用于根据第一消息池305或第二消息池306中的消息,在组件注册单元302中查询获取该消息携带的被调用组件标识所标识的组件的消息处理入口函数,并向该被调用组件的消息处理入口函数发送携带该消息队列标识(只对于第一消息池305中的消息)或者消息内存地址(对于第一消息池305或第二消息池306中的消息)的通知,通知被调用组件的消息处理入口函数根据通知携带的消息队列标识或消息内存地址,获取该消息。

值得说明的是,在本实施例中,与组件一一对应的组件标识既可以由操作系统自动生成,亦可以由各组件本身根据预定的组件标识生成规则自动生成。

由本实施例的描述可以看到,由于本发明的消息系统对于进程间以及进程内线程间的通信消息,分别设置了采用了两个不同的消息池,使得该消息系统可以适应进程、线程或混合模型,大大提高了组件的重用度。

进一步的，由于进程间空间地址隔离，因而本组件实现系统中的消息系统对于进程间的通信消息，采用消息队列的内存管理方式存储管理，而对于进程内线程间的通信消息，由于进程内线程间的地址空间可以互相访问，使得线程间通信的优点得以发挥。本发明使得进程间或线程间的通信均有高的效率。

最后，本系统规模小，需占用资源少，复杂度低，在实际中易于实现。

实施例4:

图4所示为本实施例的消息系统结构示意图，如图示，本组件实现系统中的消息系统400的结构与实施例3中的消息系统300基本相同，所不同的是，本消息系统400中使用基于共享内存管理方式的第一消息池405替换了以及消息队列的内存管理方式的第一消息池305；并且，本消息系统400进一步的包括了组件标识生成单元307，与组件存储器相连接，用于动态或静态的生成惟一标识组件的组件标识。

本实施由于对于使用了基于共享内存管理方式的第一消息池405替换了以及消息队列的内存管理方式的第一消息池305，对于进程间通信的消息，使用基于共享内存的第一消息池405进行存储管理。由于基于共享内存方式的消息池管理相对与实施例3中的基于消息队列的消息池管理，进一步减少内存拷贝的次数，进一步的提高进程间通信的效率，提高进程间调用的效率。

另外，由于消息系统进一步包含了组件标识生成单元307，使得消息系统的功能更加完整，应用更加方便。

以上对本发明所提供的组件的实现方法以及系统进行了详细介绍，本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本发明的限制。

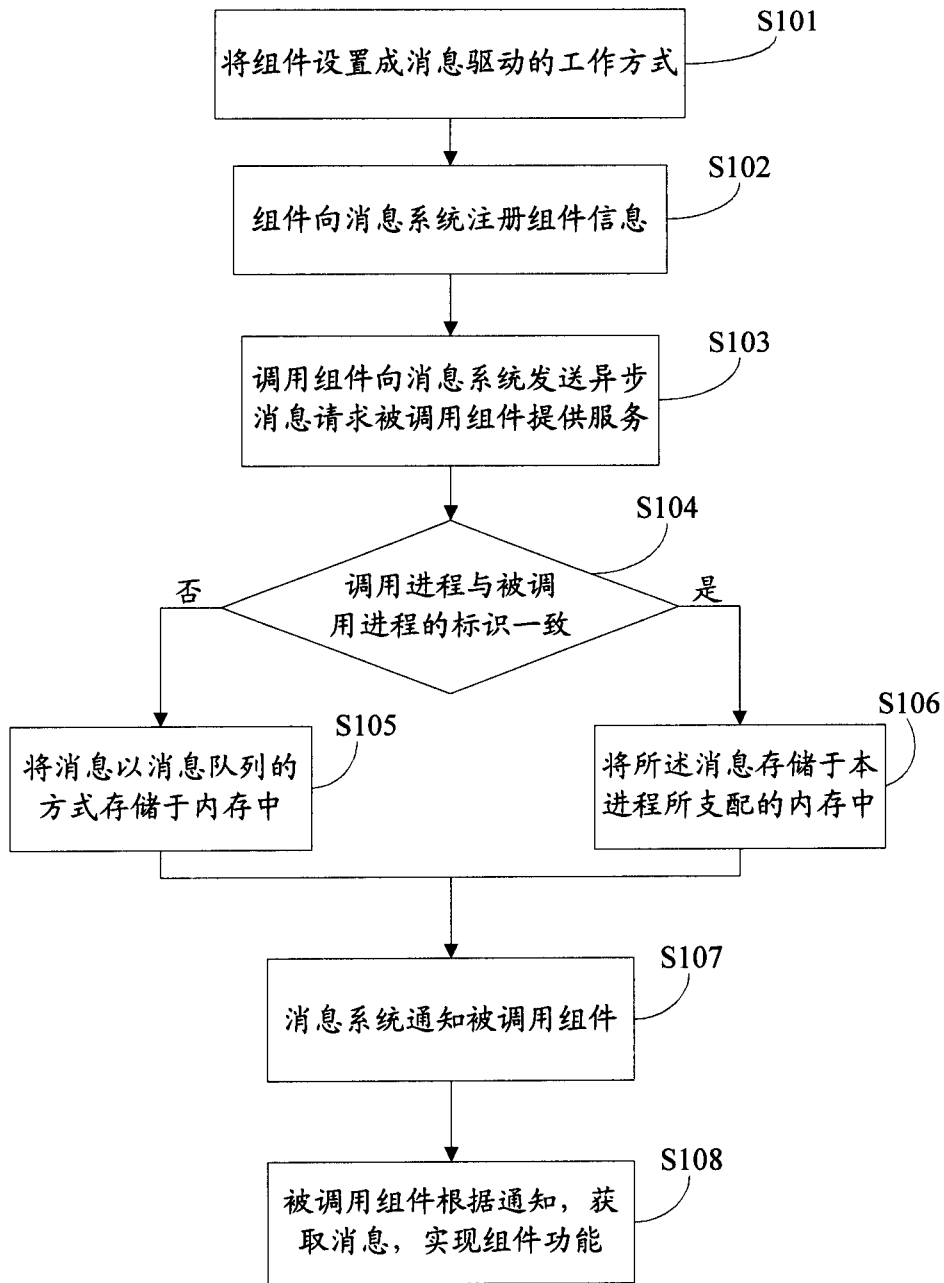


图 1

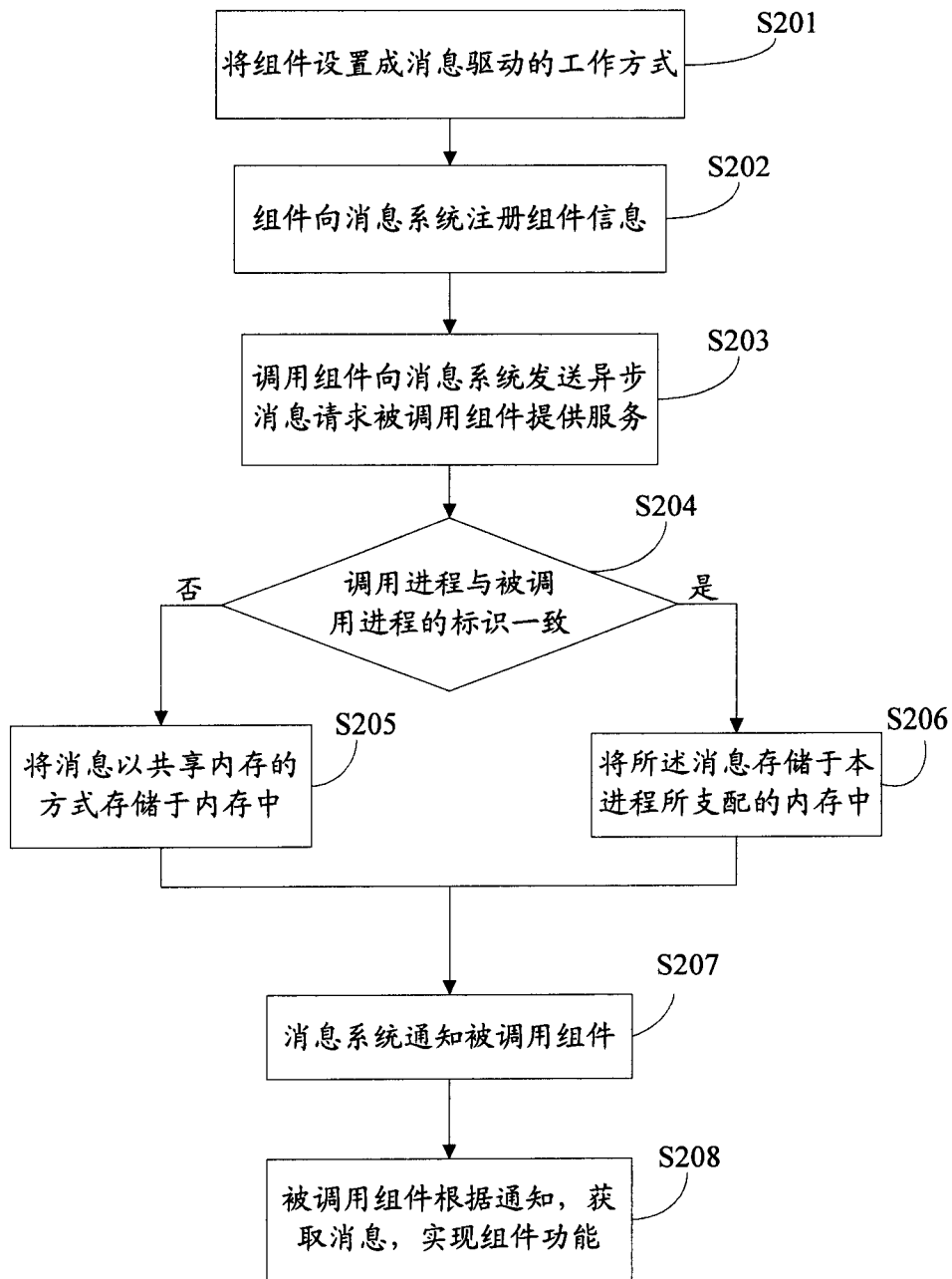


图 2

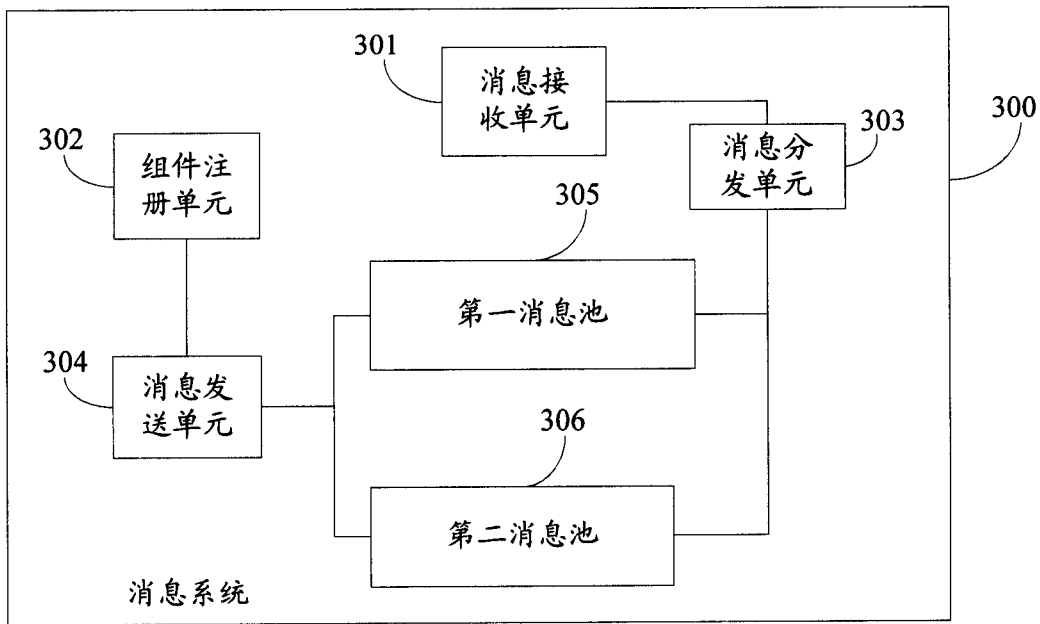


图 3

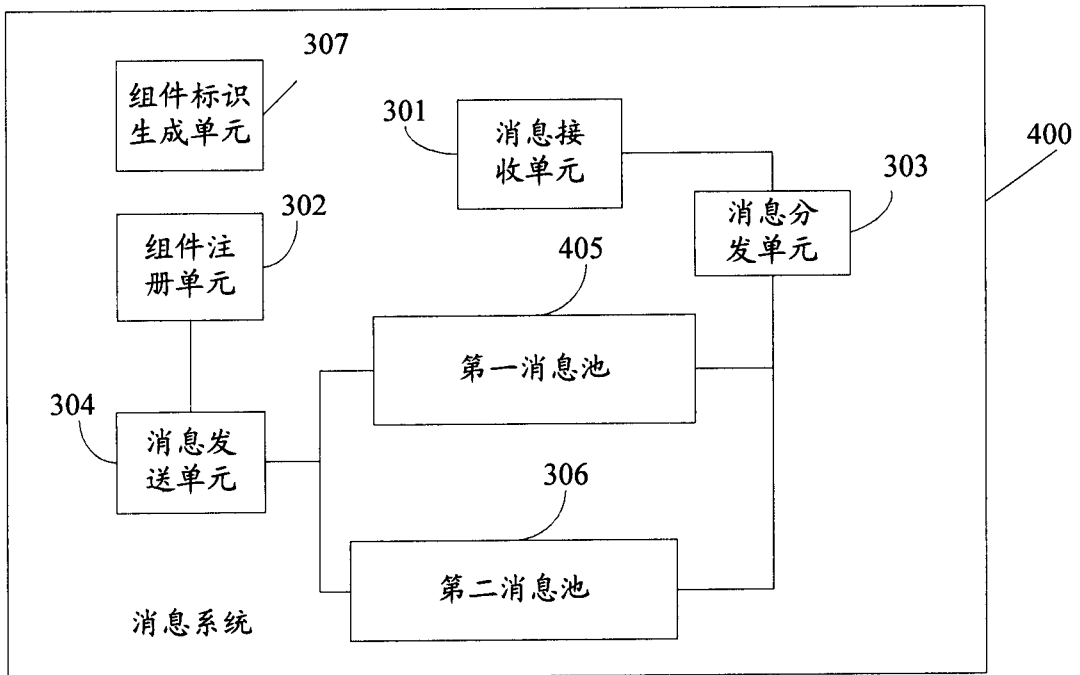


图 4