

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 1/08 (2006.01)

H04L 29/06 (2006.01)



[12] 发明专利说明书

专利号 ZL 200610067456.5

[45] 授权公告日 2010 年 3 月 24 日

[11] 授权公告号 CN 100596050C

[22] 申请日 2006.3.27

[21] 申请号 200610067456.5

[73] 专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼岛

[72] 发明人 钱志龙 程立 李磊

[56] 参考文献

US2002/0150045A1 2002.10.17

WO2005/109804A2 2005.11.17

US2005/0234914A1 2005.10.20

CN1149919A 1997.5.14

审查员 岳跃平

[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

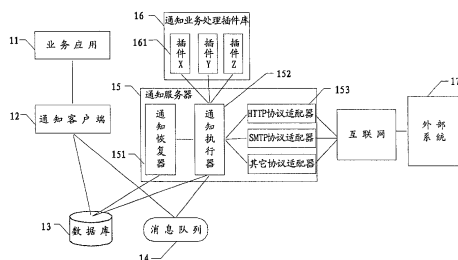
权利要求书 2 页 说明书 13 页 附图 4 页

[54] 发明名称

一种可靠的系统间消息通知方法和系统

[57] 摘要

本发明公开一种可靠的系统间消息通知方法和系统，能够保证消息通知的可靠到达。该方法包括以下步骤：将通知消息持久存储；发送存储的新通知消息或重试的通知消息；若通知消息发送成功，则删除存储的本条通知消息；否则对该通知消息设置重试时间，并更新存储；定时检查等待重试的通知消息，如果存在重试时间到期的通知消息，则执行发送步骤。所述方法和系统支持不同系统间的多协议传输，接收方不需要实现复杂的交互协议就能够可靠接收通知消息，适合广泛应用于互联网。此外，还支持多业务处理，可作为一个通用的业务应用平台。同时，还可以进行多业务和多协议的灵活扩展。



1、一种系统间消息通知方法，其特征在于，包括以下步骤：

将通知消息存储于数据库中；

根据所述通知消息的类型找到相应的业务插件，由相应的业务插件对所述通知消息进行预处理得到外部系统通知地址、通知协议和通知参数；

根据通知协议选择合适的协议适配器，将通知消息内容、外部系统通知地址和通知参数交给协议适配器以向外部系统发送通知消息；

业务插件根据是否有返回结果，以及返回结果的格式和内容是否有效决定本次发送是否需要重试；

若通知消息发送不成功，则对该通知消息设置重试时间，并更新存储；

检查等待重试的通知消息，如果存在重试时间到期的通知消息，则执行发送；

其中，在所述向外部系统发送通知消息前还包括：判断所述通知消息的事务处理是否完成，如果完成则触发该通知消息的发送；否则注册一个事务同步器，在通知消息的事务处理完成后再触发该通知消息的发送。

2、根据权利要求1所述的一种系统间消息通知方法，其特征在于，还包括：同时发送新通知消息以及重试的通知消息。

3、根据权利要求1所述的一种系统间消息通知方法，其特征在于：所述检查为定时检查。

4、根据权利要求2所述的一种系统间消息通知方法，其特征在于：所述新通知消息的发送是受到触发后执行。

5、根据权利要求1所述的一种系统间消息通知方法，其特征在于：所述的重试时间之间的间隔随着重试次数的增加而增大。

6、根据权利要求1所述的一种系统间消息通知方法，其特征在于，所述的重试时间到期包括：重试时间小于或等于定时检查的时间点。

7、一种系统间消息通知系统，其特征在于，包括：

数据库，用于保存通知消息；

消息队列；

通知客户端，用于将通知消息存放到所述数据库中，以及通过所述消息队列触发通知执行器即时发送通知消息；

设置在所述通知客户端的事务同步器，负责在通知消息的事务处理完成后通过所述消息队列触发通知执行器；

至少一种业务插件；

通知执行器，用于根据通知消息的类型选择相应的业务插件，由相应的业务插件对所述通知消息进行预处理得到外部系统通知地址、通知协议和通知参数；

至少一种协议适配器，每种协议适配器支持一种传输协议，用于系统间的数据传输；

其中，通知执行器还用于根据通知协议选择合适的协议适配器，将通知消息内容、外部系统通知地址和通知参数交给协议适配器以向外部系统发送通知消息；

所述业务插件还用于根据是否有返回结果，以及返回结果的格式和内容是否有效决定本次发送是否需要重试；

若通知消息发送不成功，则通知执行器对所述通知消息设置重试时间，并更新所述数据库中的对应通知消息；

通知恢复器，用于检查等待重试的通知消息，如果存在重试时间到期的通知消息，则通知所述通知执行器。

8、根据权利要求7所述的一种系统间消息通知系统，其特征在于：所述的通知恢复器定时运行。

一种可靠的系统间消息通知方法和系统

技术领域

本发明涉及一种数据传输领域，特别涉及一种可靠的系统间消息通知方法和系统。

背景技术

随着互联网技术的快速发展，以消息通知作为交互模式的互联网应用系统也越来越多。目前一种典型的消息通知应用是网上银行与商户之间的消息通知。它的应用场景是，用户从商户网站向网银提交交易订单，然后用户在网银完成实际支付，支付完成之后，网银将用户支付成功的结果以消息通知的形式返回给商户，由商户完成诸如发货等后续交易操作。支付结果的消息通知能否可靠送达商户，将直接影响用户的交易流程，因此该消息通知送达的可靠性是非常关键的。

但是，互联网是不可靠的，它的不可靠性既体现在网络本身连通性与延时的不可预料性，也体现在连接在互联网上的软硬件设备可用性的不可预料性。因此，基于互联网的消息通知有一定的丢失机率。但电子商务中的信息流对可靠性的要求非常高，丢失关键的消息通知轻则阻碍商业活动的顺利开展，重则带来商务活动参与方的资金财产损失。从目前的实际情况看，各种银行与商户系统间的消息通知都没有做到可靠送达。据 iResearch 的 2005 年的统计数据显示，目前国内的第三方网上支付平台中绝大多数有较高的掉单率（即用户在网银支付成功的结果没有及时通知到第三方支付平台），其中，掉单率最高的网上支付平台达到 20%。

WS-Reliable Messaging（Web Services Reliable Messaging Protocol，Web Service 可靠消息协议）和 WS-Reliability（Web Service Reliability，Web Service 可靠性）是两种面向 Web Service 的可靠消息协

议。其中, WS-Reliable Messaging 是由 IBM、BEA、Microsoft 和 TIBCO 等公司提出的一种可以在分布式系统中进行可靠消息传输的协议。而 WS-Reliability 则是由 OASIS 组织的 Web Service 可靠消息 (WSRM) 技术委员会提出的一种使 SOAP (Simple Object Access Protocol) 简单对象访问协议消息可以确保送达的、无重复的基于 SOAP 的协议。本质上, 这两种协议都是基于对消息编序, 并使用滑动窗口协议对消息进行确认来实现消息的可靠传输的。

HTTPLR 协议是爱尔兰的 Propylon 有限公司在 2005 年提出的, 它是一种建立在 HTTP 协议之上的应用层协议, 旨在为 HTTP 协议提供确保送达和单次送达的消息传输服务。HTTPLR 针对消息上行 (从客户端到服务器端) 和下行 (从服务器端到客户端) 都提供了可靠消息的机制, 但该机制不涉及消息发送端和消息接收端的可用性、组件的健壮性、消息的持久存储、以及消息的重试和超时机制, 而只是涉及消息交换的双方如何就消息是否已送达达成一致的认识。

在现实情形中, 无论是 WS-ReliableMessaging、WS-Reliability 还是 HTTPLR 协议, 都未在互联网环境下得到广泛应用。究其原因, 有以下几点: 一是这些协议本身尚未完全成熟, 有些尚处于草案阶段; 二是尽管在理论上描述了可靠消息传输的机制, 但它们并未描述实现的具体机制; 三是它们的应用环境有局限性, 比如 HTTPLR 应用于 HTTP 协议之上, 而 WS-Reliable-Messaging 和 WS-Reliability 协议是面向 Web Service 的; 四是它们对客户端和服务器的实现都有比较复杂的要求, 而在当前的互联网环境下, 要求消息交换的双方都实现同一种复杂协议往往是困难的。

发明内容

本发明要解决的技术问题是提供一种可靠的系统间消息通知方法和系统, 适用于任何协作的系统之间, 尤其解决了系统之间通过互联网进行消息通知

时，由于网络、服务器或者程序系统不稳定造成的通知消息丢失且不可恢复的问题。

为了解决上述问题，本发明提供了一种可靠的系统间消息通知方法，该方法包括以下步骤：

将通知消息存储；

发送存储的通知消息；

若通知消息发送不成功，则对该通知消息设置重试时间，并更新存储；

检查等待重试的通知消息，如果存在重试时间到期的通知消息，则执行发送步骤。

所述可靠的系统间消息通知方法还可以包括，同时发送新通知消息以及重试的通知消息。优选的，所述检查为定时检查。所述新通知消息的发送是受到触发后执行。

执行所述发送步骤前还可以包括：判断所述消息的事务处理是否完成，如果完成则触发该消息的发送；否则注册一个事务同步器，在消息的事务处理完成后再触发该消息的发送。

所述可靠的系统间消息通知方法还包括：根据所述通知消息选择相应的协议适配器进行消息通知的发送；以及根据所述通知消息选择相应的业务插件进行消息通知的预处理以及返回的通知结果的处理。优选的，所述的重试时间间隔随着重试次数的增加而增大。所述的重试时间到期包括：重试时间点小于或等于定时检查的时间点。

本发明还提供了一种可靠的系统间消息通知系统，该系统包括：

数据库，用于保存通知消息；

通知客户端，用于将通知消息存放到所述数据库中，并触发消息通知立即发送；

通知执行器，用于发送通知消息，若通知消息发送不成功，则对应所述

通知消息设置重试时间，并更新所述数据库中的对应通知消息；

通知恢复器，用于检查等待重试的通知消息，如果存在重试时间到期的通知消息，则通知所述通知执行器。

所述的通知恢复器定时运行。

所述可靠的系统间消息通知系统还可以包括：消息队列，用于触发所述通知执行器即时发送通知消息。

所述可靠的系统间消息通知系统还可以包括：设置在所述通知客户端的事务同步器，负责在消息的事务处理完成后再通过所述消息队列触发通知执行器。优选的，所述通知客户端嵌入在消息的业务系统中。

所述可靠的系统间消息通知系统还可以包括：至少一种协议适配器，每种协议适配器支持一种传输协议，用于系统间的实际数据传输；以及至少一种业务插件，用于通知消息发送前的预处理和返回的通知结果的处理。优选的，由所述业务插件判断决定通知消息的发送是否成功。

本发明适用于任何协作的系统之间，与现有互联网间可靠消息交换的技术相比，本发明具有的优点是：

1、系统之间通过互联网进行消息通知时，由于本发明将通知消息持久存储，如存储到所述数据库中，直到该消息发送成功，避免因为软、硬件系统故障而丢失该通知任务，保证了消息通知的可靠到达。

2、所述方法和系统针对互联网上的短期和中长期通信故障模式提出现实、高效的超时和消息重发解决方案。随着重试次数的增加，其重试时间的间隔也越来越长。重试时间间隔的变化，兼顾了通知恢复的及时性和效率。对于短期的通信故障，系统能够在故障恢复之后及时将消息通知送达接收方；对于中长期通信故障，系统能够避免多次重试的开销。

3、本发明通过扩展协议适配器，能够适用于各种消息传输协议，而不是局限于一、两种协议，适用于广泛的互联需求。由于这种多协议的支持，这

种方法和系统在实现上是非对称的，消息交互的一方（如消息的接收方）不需要实现复杂的交互协议，只需要根据业务的要求返回消息处理的结果，就能够接收来自消息提供方的可靠消息。

4、本发明所述方法和系统能够作为一个通用的平台，通过扩展业务插件，同时支撑各种不同业务应用的运转，并且可以灵活地扩展新的业务。

附图说明

下面结合附图和具体实施方式对本发明作进一步详细的说明。

图1是本发明一种可靠的系统间消息通知系统的系统结构示意图；

图2是本发明一种可靠的系统间消息通知方法的步骤流程图；

图3是消息通知的注册流程图；

图4是消息通知的发送流程图；

图5是消息通知的恢复流程图；

图6是重试间隔确定方法的概念模式图。

具体实施方式

基于互联网的电子商务在当今社会中占据了越来越重要的地位。越来越多的商业实体通过互联网组织信息流、资金流和物流，以消息通知作为交互模式的互联网应用系统也越来越多。以银行为例，用户在银行完成支付后，银行需要将用户支付的情况通过消息通知的形式告知商户系统；以第三方安全交易平台为例，当用户在交易平台上推进交易流程后，交易平台需要将交易的当前状态通知相关的外部商户系统。这些都是典型的基于互联网的消息通知。本发明适用于任何协作的系统之间，尤其适用于广泛的互联网。

参照图1，是本发明一种可靠的系统间消息通知系统的结构示意图。业务应用11可以为某个商业应用，它是消息通知过程的发起者。外部系统17

可以为某个商业系统，它是消息的接收者。本发明所提供的通知系统用于发送所述业务应用 11 发来的通知消息，并能够确保该通知消息通过互联网可靠到达所述的外部系统 17。所述通知系统可以包括：

数据库 13，用于保存待发送的新通知消息和等待重发的通知消息；

通知客户端 12，用于将通知消息存放到所述数据库中，并通过消息队列 14 触发消息通知立即发送；

通知执行器 152，用于发送新通知消息或重试的通知消息，若通知消息发送成功，则从数据库 13 中删除本条通知消息；否则对应所述通知消息设置重试时间，并更新数据库 13 中的对应通知消息；

通知恢复器 151，用于检查等待重试的通知消息，如果存在重试时间到期的通知消息，则通知所述通知执行器 152。

如图所示，数据库 13 可以是任何通用的关系型数据库，用以保存待发送的新通知消息和曾经发送不成功等待重发的通知消息。通知客户端 12 可以为供业务应用 11 使用的一个客户端模块，嵌入到消息的业务系统中，业务应用 11 可以使用通知客户端 12 注册一个通知消息。一旦通知消息注册成功，系统就能够确保该通知消息到达消息的接收方，即使此时消息接收方不在线，或者网络暂时无法连通。所述注册可以为，通知客户端 12 将业务应用 11 发来的通知消息存放到数据库中，并通过消息队列 14 触发消息即时发送。所述业务系统是指针对业务应用 11，进行相应的业务处理的系统。结合网上银行与商户之间的消息通知实例，所述消息的事务处理是指用户在网上银行进行实际支付，若支付完成则事务处理提交，否则还处于事务处理中；所述通知消息指用户的支付结果，即是否完成支付。所述事务是一个技术上的专有名词，表示具有 ACID 特性的一组操作，事务的 ACID 特性是实现消息通知与业务操作 100%一致性的关键。当创建事务时，应确保事务具有某些可以自行管理的特性，这些特性就称为 ACID 特性。ACID 是指：原子性(Atomicity)、

一致性(Consistency)、隔离性(Isolation)和持久性(Durability)。

将通知客户端 12 嵌入到消息的业务系统中，可以在同一个数据库事务中完成业务处理以及通知任务的注册，完全可以避免业务处理和通知任务不一致的情况。所述一致为一通知消息只有当事务处理完成提交后，该通知消息才注册成功，进行即时发送。所述数据库事务中的数据库可以是系统中保存消息通知请求的数据库。一般说来，这个数据库与业务系统保存业务数据的数据库是同一个。如果是同一个数据库，则可以避免使用复杂低效的分布式事务机制就能实现通知消息的注册与业务数据的处理位于同一数据库事务中。但也可以使用不同的数据库，并通过分布式事务机制保证通知消息注册与业务数据处理的一致性。

消息队列 14 是一种标准的用于系统间异步消息通讯的中间件。通过使用消息队列，消息的发送过程和接收过程之间是异步的，同时使得消息发送方和消息接收方不直接进行通讯，而是通过消息队列进行间接通讯。这样做最大程度地减小了消息发送方和消息接收方之间的相互依赖，使得消息发送方与消息接收方之间可以相对独立地进行工作。目前的消息队列一般都能够做到当接收到消息之后，只要消息接收方工作正常，可以即时将消息传递给消息接收方。本实施例中，消息队列 14 用以触发通知消息的即时发送。

通知服务器 15 可以为一台或一组独立运行的，用以专门负责通知消息发送和通知消息重发的服务器，可以包括通知恢复器 151、通知执行器 152 以及多种协议适配器 153。其中，通知恢复器 151 为负责调度未成功发送的通知消息在恰当的时间进行重新发送的模块；通知执行器 152 为负责执行实际的通知消息发送的模块；多种协议适配器 153 中，每种协议适配器支持一种传输协议，通过互联网完成与外部系统 17 的实际数据传输。通知业务处理插件库 16 为各种业务插件 161 的管理器。业务插件 161 负责通知消息发送前的预处理和返回的通知结果处理，所述预处理和结果处理均与业务应用紧密相

关，不同的业务应用，其处理过程也不相同。

参照图 2，是本发明一种可靠的互联网系统间消息通知方法的步骤流程图。当业务应用需要向外部系统发送一通知消息时，执行以下步骤即可可靠送达：

步骤 s1，业务应用将自己的通知需求封装成一个消息通知请求，发送至通知客户端。

步骤 s2，通知客户端将所述通知消息进行注册。一旦通知消息注册成功，业务应用就可以继续其它业务处理，所述系统就能够确保该通知消息到达消息的接收方，即使此时消息接收方不在线，或者网络暂时无法连通。

步骤 s3，当通知消息注册成功后，通知系统进行通知消息的发送。若发送成功，则消息通知任务结束；否则继续步骤 s4。

步骤 s4，进行消息发送的恢复处理，调度未成功发送的通知消息在恰当的时间进行重新发送，继续步骤 s3。

在此过程中，步骤 s4 与步骤 s1、步骤 s2 和步骤 s3 可以同时执行。下面分别对步骤 s2、s3 和 s4 进行详细说明。

参照图 3，是消息通知的注册流程图，包括以下步骤：

步骤 a1，通知客户端接收所述业务应用发来的消息通知请求；所述消息通知请求可以包括消息通知的内容。

步骤 a2，通知客户端将所述通知请求存放到数据库中，直到通知消息发送成功，这样可以避免在发送过程中由于网络、服务器或程序系统不稳定造成的通知消息丢失且不可恢复的问题。

步骤 a3，通知客户端判断所述消息的事务处理是否完成提交，若还处于事务处理中，则继续执行步骤 a4、步骤 a5 和步骤 a6；若事务已完成提交，则通知客户端自动触发消息队列，直接执行步骤 a6。结合网上银行与商户之间的消息通知实例，所述消息的事务处理是指用户在网上银行进行实际支

付，若支付完成则事务处理提交，否则还处于事务处理中；所述通知消息指用户的支付结果，即是否完成支付。本实施例中，由于通知客户端嵌入到业务系统中，能够在业务事务内完成将通知任务注册到数据库中，这样避免使用复杂的机制就实现了业务处理和通知注册的一致性。

步骤 a4，通知客户端向事务管理器注册一个事务同步器。所述事务管理器是对事务进行管理的软件，包括事务同步器。所述事务同步器设置在通知客户端，用于实现在事务提交之后通过消息队列向通知服务器发送一个立即发送通知消息的请求，可以保证通知的及时性。

步骤 a5，当事务完成提交之后，事务同步器自动触发消息队列。

步骤 a6，消息队列向通知执行器发送一个立即发送通知消息的请求，驱动通知执行器进行发送，保证了通知的即时性。

参照图 4，是消息通知的发送流程图，包括以下步骤：

步骤 b1，通知执行器接收到通知客户端通过消息队列发来的新消息通知请求，或者消息恢复器发来的重试消息通知请求。如果同时收到消息队列发来的新消息通知请求和消息恢复器发来的重试消息通知请求，则优选的，通知执行器执行多线程处理，可以同时发送新通知消息以及重试的通知消息。当然，也可以采用确定优先级进行发送。

步骤 b2，通知执行器根据消息通知请求的类型，从业务插件库中找到相应的业务插件，将消息通知请求交给业务插件进行预处理，得到实际的外部系统通知地址、通知协议和通知参数。

步骤 b3，通知执行器根据通知协议选择合适的协议适配器，将通知消息内容、通知地址和通知参数交给协议适配器进行实际的消息发送。

步骤 b4，如果网络、双方服务器与系统均正常工作，则消息通过互联网到达外部系统，外部系统接收到消息并处理后返回处理结果；否则所述通知执行器能够自动检测到通知消息未到达外部系统，继续步骤 b8 和 b9。

步骤 b5, 通知执行器将返回结果交给业务插件, 由业务插件完成相应的业务处理。在上述例子中, 当收到外部商户的返回结果之后, 业务插件需要将交易的状态变为已发货, 并且记录发货单的详情, 通知用户发货情况。由于通知服务器本身是通用的, 与具体业务无关, 因此这些处理是由业务插件来完成的。

步骤 b6, 业务插件判断本次发送是否需要重试。业务插件是根据是否有返回结果, 以及返回结果的格式和内容是否有效决定。仍以上述例子为例, 如果返回结果中包含有效的商户发货信息, 则业务插件判断消息通知是成功的; 如果没有返回结果, 或者返回结果格式无效, 或者外部商户在返回结果中直接表明暂时无法处理, 需要过一段时间重试, 则业务插件判断消息通知未成功, 需要重试。

因为消息发送不成功的可能原因不但包括网络或系统故障, 还取决于业务处理是否能够成功, 而通用的通知服务器可以判定网络与系统的故障, 但无法判定业务处理是否成功, 因此将判定职责交给业务插件去完成。当然, 明显的网络与系统故障是通知服务器本身能够判断的, 因此, 当发生这种情况时, 由通知服务器直接决定需要重试。

步骤 b7, 如果业务插件表明本次消息不需要重试, 则通知执行器从数据库中删除本条消息通知请求, 成功完成通知发送。

步骤 b8, 如果由于网络等原因发送的通知消息未到达外部系统, 或业务插件表明本次消息需要重试, 则通知执行器根据重试策略计算下一次重试需要等待的时间间隔, 并用该间隔加上当前时间作为下一次消息发送重试的时间点。

步骤 b9, 对于等待重新发送的通知消息, 通知执行器根据计算的重试时间, 更新数据库中对应消息通知请求的发送时间, 并放弃本轮通知, 等待通知恢复器的重新调度。

在上述通知发送流程中，通过扩展业务插件，可以实现在同一个消息通知系统中支持各种类型的业务通知；通过扩展协议处理器，可以实现在同一个消息通知系统上支持不同协议的通知。此外，通知的接收者不需要实现任何特殊的可靠消息通知协议，只需要根据业务的要求返回消息处理的结果，就能够可靠接收来自消息提供方的消息。

参照图 5，是消息通知的恢复流程图。当所述通知消息需要重新发送时，进入通知恢复流程。优选的，通知恢复器定时运行，在每一轮运行中，执行以下步骤：

步骤 c1，通知恢复器等待定时时间，当时间点到开始运行。

步骤 c2，从数据库中查找重试时间到期的消息通知请求。检查需要重试的消息通知请求，将步骤 b8 计算所得的重试时间与本轮的定时时间相比较，若有重试时间点小于或等于定时时间点的情况，则存在重试时间到期的消息通知请求，继续步骤 c3；否则放弃本轮重试，继续步骤 c4。

步骤 c3，对于重试时间到期的通知消息，向通知执行器发送重试请求，逐条交给通知执行器进行通知发送。

步骤 c4，等待下一定时时间点，进行下一轮运行。

通知恢复流程的定时时间一般是固定的，比如每隔 1 分钟运行一次。定时时间间隔的大小关系到通知恢复的及时性，因此，这个时间间隔一般设置得越小越好，但必须在通知服务器能够承受的性能范围内。此外，这个定时时间间隔与消息通知重试的间隔并不相同，消息通知重试的间隔时间是通过重试间隔确定方法针对每一条通知消息单独计算和设置的。

在通知发送流程的步骤 b8 中，所述的重试策略可以采用常用的固定重试时间间隔，本发明优选采用下述的一种重试间隔确定方法进行重试时间间隔的确定。所述重试间隔确定方法，是针对互联网上消息通知失败的常见原因设计的，意在用尽可能少的重试次数、尽可能及时地将消息通知发送到消息

接收方。

互联网上跨系统的消息通知失败的可能原因有以下几种：网络暂时繁忙，造成传输协议超时；网络短暂中断；对方服务器暂时繁忙，造成不响应请求；对方系统存在 BUG，造成不响应请求或者请求处理出错；对方服务器宕机，造成不响应请求；网络长时间故障，造成长达数小时以至于数天的中断；对方系统长时间故障，造成长达数小时以至于数天不可用；对方系统不存在或者永久关闭。

从上述原因可见，造成消息通知不成功的原因既有可能在数分钟内消失或缓解，也有可能长达数小时或者数天。消息通知的最佳重试策略应为：当原因可以在数分钟内消失或缓解时，消息通知能够及时地发送给对方；而当原因需要数小时或者数天才能消失时，系统也能将消息通知发送给对方，而且不做太多无谓的尝试。因此，本发明优选采用下述的一种重试间隔确定方法进行重试时间间隔的确定。

参照图 6，是所述重试间隔确定方法的概念模式图。对所述确定方法说明如下：

假定有一组编号从 1 到 $n+1$ 的盒子，1 到 n 号盒子上都绑定一个定时器，盒子的编号越大，定时间隔越长，例如第一次间隔 2 分钟，第二次间隔 5 分钟，第 3 次间隔 10 分钟等等。第 $n+1$ 号盒子上没有定时器，认为消息接收方永久不可达，放弃自动重试，等待人工恢复。

一个消息通知任务第 i 次 ($i = 1..n$) 发送不成功，则进第 i 个盒子；

第 i 个盒子的定时器每次触发都会重试本盒子中的所有消息通知任务；

当一个消息通知任务第 $n+1$ 次发送不成功时，则进第 $n+1$ 个盒子，由于第 $n+1$ 个盒子上没有绑定定时器，因此系统将会放弃对该消息通知任务的自动重试，转入人工处理。

此外，本发明中的通知系统针对不同类型外部系统的特点，对盒子上的

定时器进行了调整，使之符合相应外部系统的故障模式。

采用图 6 所示的重试间隔确定方法，如果消息通知任务重试的次数较少，说明造成消息通知发送不成功的原因在很短时间内消失了，则由于每次重试的时间间隔也较小，故可以保证故障恢复时消息通知的及时发送；如果消息通知任务重试的次数较多，说明造成消息通知发送不成功的原因属于长期通信故障，则后来重试的时间间隔也较长，可以有效避免太多无效的重试尝试，减少系统资源占用，但也可以保证消息通知的确定送达。即上述重试间隔确定方法确保了短期通信故障能够及时恢复消息通知，而对于长期通信故障则能够避免大量无效的通知尝试，兼顾了通知恢复的及时性和效率。

上述本发明一种可靠的系统间消息通知方法和系统，适用于任何协作的系统之间，尤其解决了系统之间通过互联网进行消息通知时，由于网络内在的不可靠性和系统软、硬件故障造成的通知消息丢失且不可恢复的问题，确保消息的可靠、及时到达。本发明支持不同系统间的多协议传输，接收方不需要实现复杂的交互协议就能够可靠接收通知消息，适合广泛应用于互联网。所述方法和系统还支持多业务处理，可作为一个通用的业务应用平台。同时，可以进行多业务和多协议的灵活扩展。

以上对本发明所提供的一种可靠的系统间消息通知方法和系统进行了详细介绍，本文仅以优选实施例对本发明进行说明。特别是，本发明所述的方法和系统适用于任何协作的系统之间，并不局限在基于互联网的系统之间，只是优选保证了互联网间的可靠消息通知，更适合广泛应用于互联网。所以，并不能因此即局限本发明的权利范围，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。

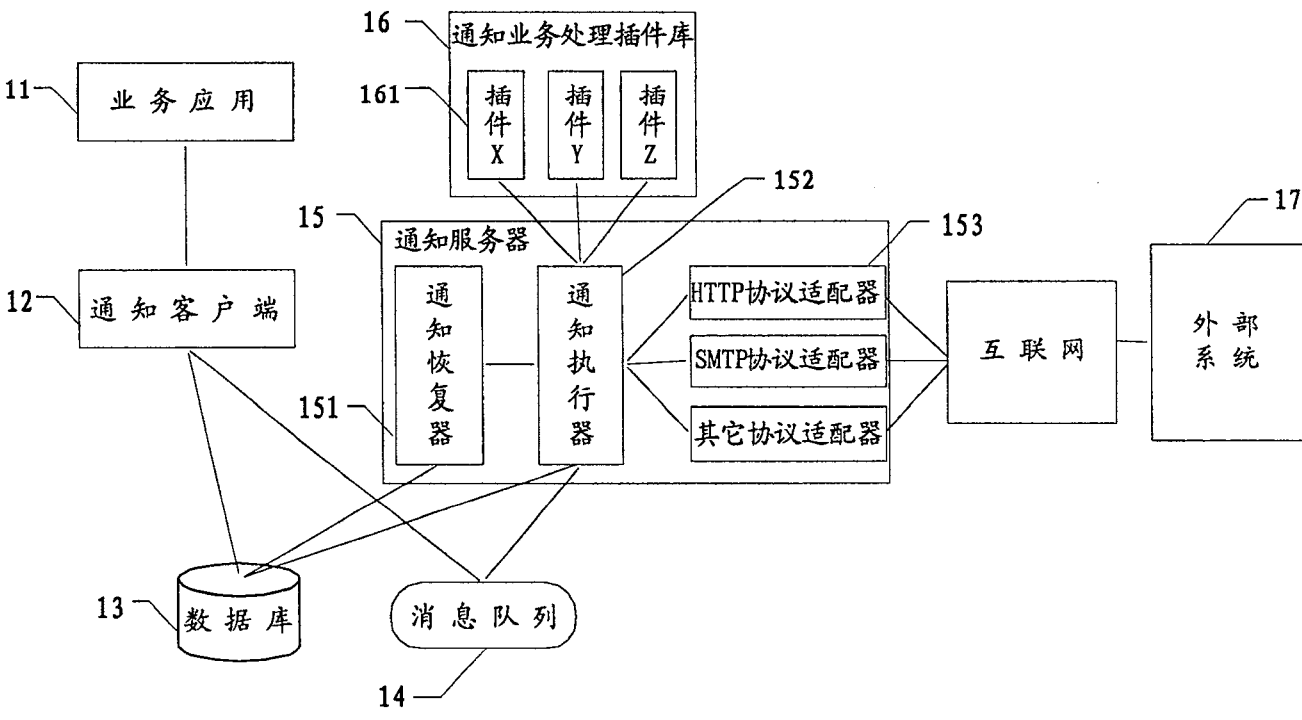


图 1

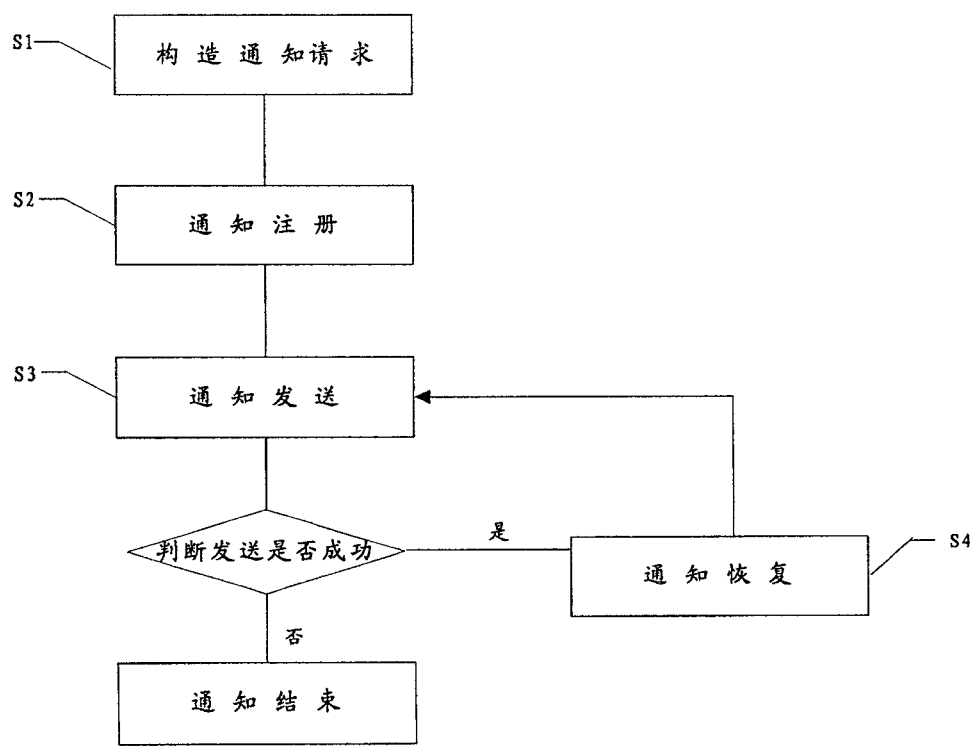


图 2

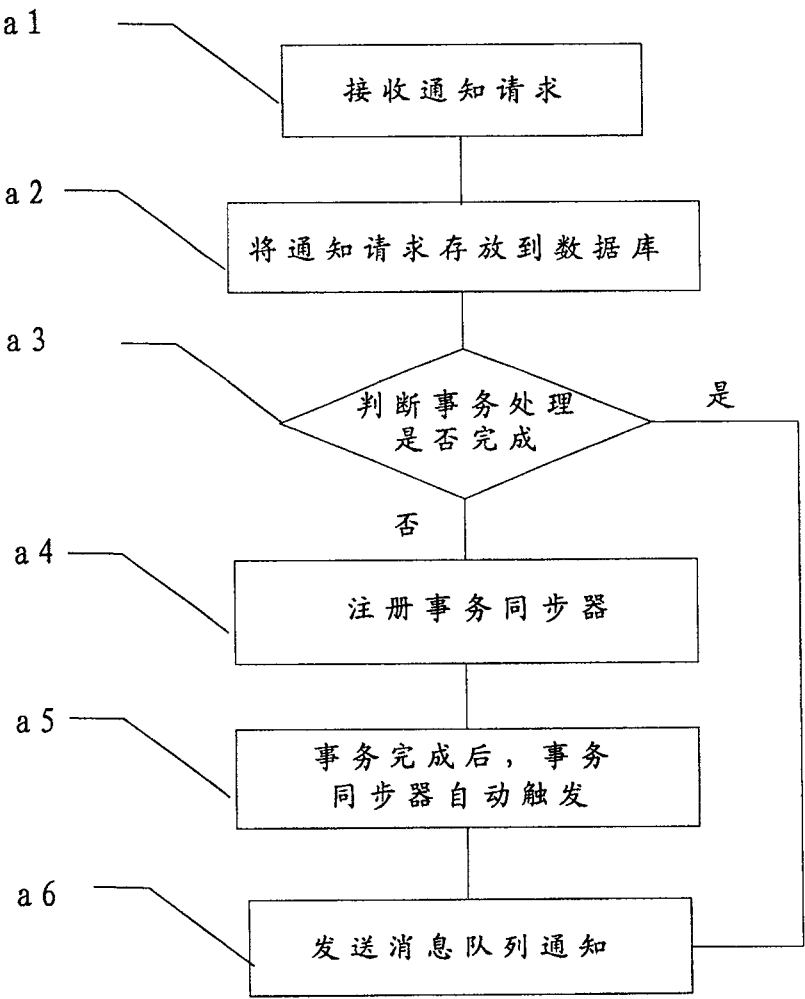


图 3

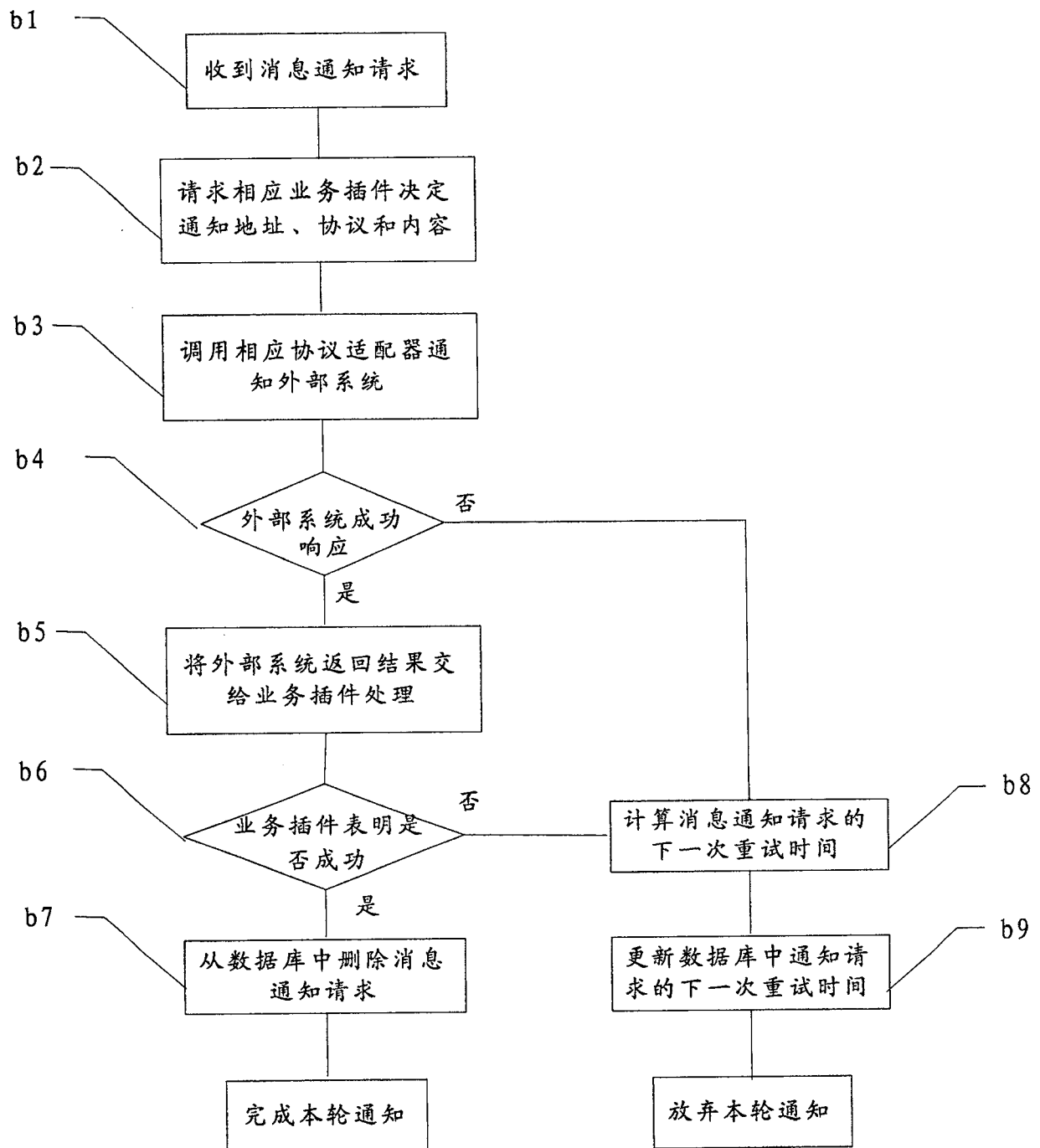


图 4

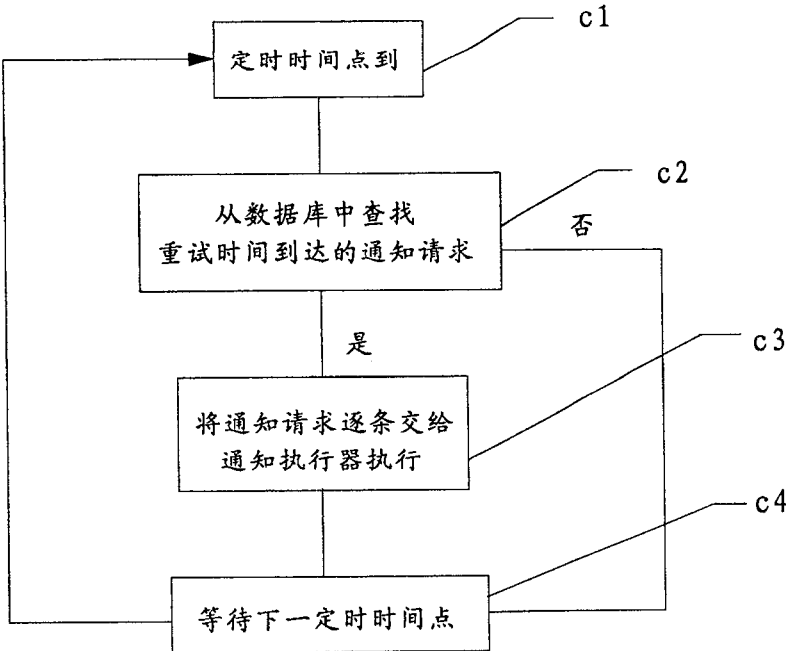


图 5

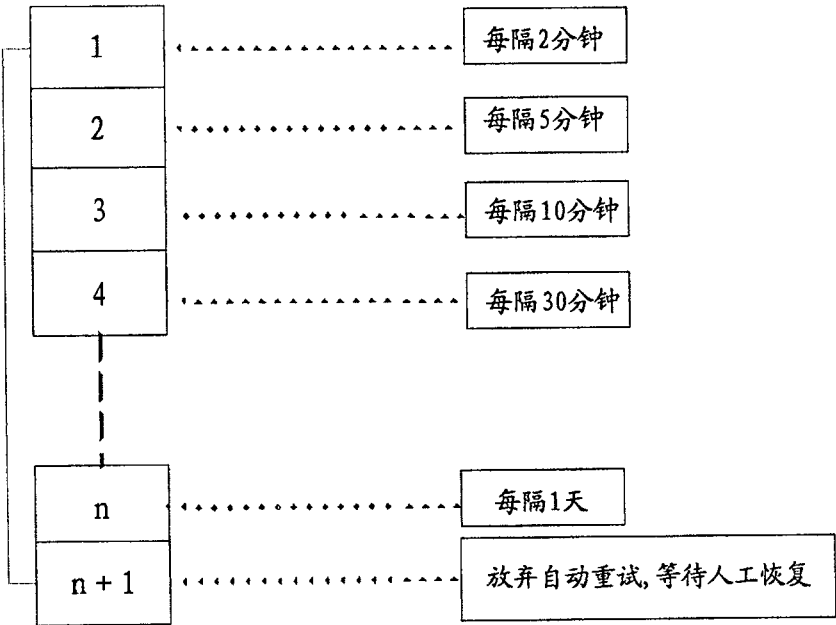


图 6