



(12) 发明专利申请

(10) 申请公布号 CN 105027079 A

(43) 申请公布日 2015. 11. 04

(21) 申请号 201480009117. 4

(51) Int. Cl.

(22) 申请日 2014. 03. 18

G06F 9/44(2006. 01)

(30) 优先权数据

13305390. 0 2013. 03. 28 EP

(85) PCT国际申请进入国家阶段日

2015. 08. 17

(86) PCT国际申请的申请数据

PCT/EP2014/055424 2014. 03. 18

(87) PCT国际申请的公布数据

W02014/154534 EN 2014. 10. 02

(71) 申请人 汤姆逊许可公司

地址 法国伊西莱穆利欧市

(72) 发明人 克里斯托夫·马藤斯

多米尼克·夏内

亚历克斯·德·斯迈特

(74) 专利代理机构 北京东方亿思知识产权代理

有限责任公司 11258

代理人 李晓冬

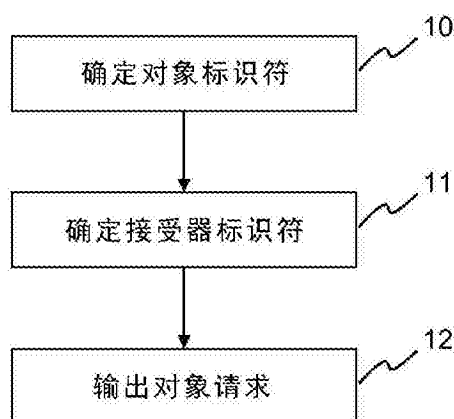
权利要求书2页 说明书6页 附图6页

(54) 发明名称

发布 - 订阅环境中的对象生命周期管理

(57) 摘要

描述了用于发布 - 订阅环境中的对象的解决方案。对象确定单元 (22) 确定 (10) 对象的对象标识符, 而接受器确定单元 (23) 确定 (11) 接受器的接受器标识符, 接受器应当操纵对象。至少包括对象标识符和接受器标识符的对象请求然后被输出 (12) 到发布 - 订阅环境。对象请求的接受器检查 (31) 对对象的操纵是否被需要并根据需要来操纵 (32) 对象。



1. 一种用于请求对对象的操纵的方法,所述方法包括以下步骤:
 - 确定 (10) 所述对象的对象标识符;
 - 确定 (11) 接受器的接受器标识符,所述接受器应当操纵所述对象;以及
 - 输出 (12) 对象请求到发布-订阅环境,所述对象请求至少包括所述对象标识符和所述接受器标识符。
2. 如权利要求 1 所述的方法,其中对所述对象的操纵包括以下各项中的一项:所述对象的创建、所述对象的修改、所述对象的删除、以及所述对象的移动。
3. 如权利要求 1 或 2 所述的方法,其中所述对象请求还包括所述对象的名称和所述对象的特性。
4. 一种被配置为请求对对象的操纵的装置 (20),所述装置 (20) 包括:
 - 对象确定单元 (22),该对象确定单元 (22) 被配置为确定 (10) 所述对象的对象标识符;
 - 接受器确定单元 (23),该接受器确定单元 (23) 被配置为确定 (11) 接受器的接受器标识符,所述接受器应当操纵所述对象;以及
 - 输出设备 (24),该输出设备被配置为输出 (12) 对象请求到发布-订阅环境,所述对象请求至少包括所述对象标识符和所述接受器标识符。
5. 如权利要求 4 所述的装置 (20),其中对所述对象的操纵包括以下各项中的一项:所述对象的创建、所述对象的修改、所述对象的删除、以及所述对象的移动。
6. 如权利要求 4 或 5 所述的装置 (20),其中所述对象请求还包括所述对象的名称和所述对象的特性。
7. 一种在其中存储有指令的计算机可读存储介质,所述指令使得能够请求对对象的操纵,所述指令当被计算机执行时,使得所述计算机:
 - 确定 (10) 所述对象的对象标识符;
 - 确定 (11) 接受器的接受器标识符,所述接受器应当操纵所述对象;以及
 - 输出 (12) 对象请求到发布-订阅环境,所述对象请求至少包括所述对象标识符和所述接受器标识符。
8. 如权利要求 7 所述的计算机可读存储介质,其中对所述对象的操纵包括以下各项中的一项:所述对象的创建、所述对象的修改、所述对象的删除、以及所述对象的移动。
9. 如权利要求 7 或 8 所述的计算机可读存储介质,其中所述对象请求还包括所述对象的名称和所述对象的特性。
10. 一种用于操控对象的方法,所述方法包括以下步骤:
 - 从发布-订阅环境接收 (30) 对象请求,所述对象请求至少包括对象标识符和接受器标识符;
 - 响应于接收 (30) 所述对象请求,检查 (31) 对对象的操纵是否被需要;以及
 - 根据需要来操纵 (32) 所述对象。
11. 如权利要求 10 所述的方法,其中对所述对象的操纵包括以下各项中的一项:所述对象的创建、所述对象的修改、所述对象的删除、以及所述对象的移动。
12. 一种被配置为操控对象的装置 (40),所述装置 (40) 包括:
 - 输入设备 (41),所述输入设备 (41) 被配置为从发布-订阅环境接收 (30) 对象请求,

所述对象请求至少包括对象标识符和接受器标识符；

- 估计单元 (42), 所述估计单元 (42) 被配置为响应于接收 (30) 所述对象请求, 检查 (31) 对对象的操纵是否被需要 ; 以及
- 操纵单元 (43), 所述操纵单元 (43) 被配置为根据需要来操纵 (32) 所述对象。

13. 如权利要求 12 所述的装置 (40), 其中对所述对象的操纵包括以下各项中的一项 : 所述对象的创建、所述对象的修改、所述对象的删除、以及所述对象的移动。

14. 一种在其中存储有指令的计算机可读存储介质, 所述指令使能对对象的操纵, 所述指令当被计算机执行时, 使得所述计算机 :

- 从发布 - 订阅环境接收 (30) 对象请求, 所述对象请求至少包括对象标识符和接受器标识符 ;
- 响应于接收 (30) 所述对象请求, 检查 (31) 对对象的操纵是否被需要 ; 以及
- 根据需要来操纵 (32) 所述对象。

15. 如权利要求 14 所述的计算机可读存储介质, 其中对所述对象的操纵包括以下各项中的一项 : 所述对象的创建、所述对象的修改、所述对象的删除、以及所述对象的移动。

发布 - 订阅环境中的对象生命周期管理

技术领域

[0001] 本发明涉及用于对象生命周期管理的解决方案。更具体地,本发明涉及用于发布 - 订阅环境中的对象生命周期管理的解决方案。

背景技术

[0002] 在软件架构领域,发布 - 订阅 (P/S) 是一种消息传递模式,其中消息的发送者 (称为发布者) 不会对消息编程来直接发送到特定的接收者 (称为订阅者)。相反,所发布的数据被多播,而不知道可能有什么订阅者 (如果有任何订阅者的话)。类似地,订阅者订阅具体数据,并且仅接收感兴趣的消息,而不知道有什么发布者 (如果有任何发布者的话)。

[0003] 图 1 中示出了示例性 P/S 框架。除了 P/S 实体之外,描绘了 P/S 主题之间的关系,以及应用所使用的对象实例。注意到,图 1 中使用了 DDS 术语 (DDS:数据分发服务)。

[0004] 如果发布者和订阅者所链接到的对象 (即,主题实例) 经历变化 (例如,由于应用以某种方式对对象进行处理,或者由于环境的变化),则有从发布者到订阅者的发布动作。

[0005] 在上述 P/S 环境中,存在对对象进行处理 (特别是对对象生命周期) 的问题。

[0006] 考虑下面的对象生命周期方法:

- [0007] • 创建对象;
- [0008] • 更新对象;
- [0009] • 删除对象;
- [0010] • 将对象移动到另一上下文,即,删除一个上下文中的对象,在另一上下文中创建相同的对象;

[0011] • 如果可能的话,同时执行上述方法的组合,例如,相结合地移动并更新对象。

[0012] 实际上, P/S 环境是面向数据的,而以上方法是方法导向的。此外,只要对象不被创建,方法就不能作用于对象。另一个问题是,应当有发起 (issue) 方法的实体 (角色),以及执行方法的实体 (角色)。

发明内容

[0013] 本发明的目标是提出改进的解决方案用于操纵发布 - 订阅环境中的对象。

[0014] 根据本发明的一个方面,用于请求对对象的操纵的方法包括以下步骤:

- [0015] - 确定对象的对象标识符;
- [0016] - 确定应当对对象进行操纵的接受器的接受器标识符;以及
- [0017] - 输出对象请求到发布 - 订阅环境,对象请求至少包括对象标识符和接受器标识符。

[0018] 相应地,被配置请求对对象的操纵的装置包括:

- [0019] - 被配置为确定对象的对象标识符的对象确定单元;
- [0020] - 被配置为确定应当对对象进行操纵的接受器的接受器标识符的接受器确定单元;以及

[0021] - 被配置为输出对象请求到发布 - 订阅环境的输出, 对象请求至少包括对象标识符和接受器标识符。

[0022] 类似地, 计算机可读存储介质中存储有使得能够请求对对象的操纵的指令, 当指令被计算机执行时, 使得计算机进行以下操作:

[0023] - 确定对象的对象标识符;

[0024] - 确定接受器的接受器标识符, 接受器应当操纵对象; 以及

[0025] - 输出对象请求到发布 - 订阅环境, 对象请求至少包括对象标识符和接受器标识符。

[0026] 优选地, 对象请求还包括对象的名称和对象的特性。

[0027] 根据本发明的另一方面, 用于操控对象的方法包括以下步骤:

[0028] - 从发布 - 订阅环境接收对象请求, 对象请求至少包括对象标识符和接受器标识符;

[0029] - 响应于接收对象请求, 检查对对象的操纵是否被需要; 以及

[0030] - 根据需要来操纵对象。

[0031] 相应地, 被配置为操控对象的装置包括:

[0032] - 被配置为从发布 - 订阅环境接收对象请求的输入, 对象请求至少包括对象标识符和接受器标识符;

[0033] - 被配置为响应于接收对象请求, 检查对对象的操纵是否被需要的估计单元; 以及

[0034] - 被配置为根据需要来操纵对象的操纵单元。

[0035] 类似地, 计算机可读存储介质中存储有使能请求对对象的操纵的指令, 当指令被计算机执行时, 使得计算机进行以下操作:

[0036] - 从发布 - 订阅环境接收对象请求, 对象请求至少包括对象标识符和接受器标识符;

[0037] - 响应于接收对象请求, 检查对对象的操纵是否被需要; 以及

[0038] - 根据需要来操纵对象。

[0039] 本发明描述了用于 P/S 环境中的对象生命周期的框架。它定义了该框架中的相关角色, 并定义了用于执行所有对象生命周期方法的一次请求机制。更具体地, 描述了两个角色, 即发起方法的对象管理器的角色和执行方法的对象提供器的角色。创建、更新、删除和移动方法被结合到单个 OBJECT_REQUEST 方法中。

[0040] 该框架允许对象管理器控制或管理对象, 对象管理器不在它自己控制的环境中对对象进行管理, 但在由对象提供器控制的环境中对对象进行管理。

[0041] 所提出的解决方案允许在需要动态对象创建和处理的环境中在面向数据的控制系统中使用用于对象生命周期管理的方法。由于可以仅用一种方法来覆盖完整的对象生命周期, 因此解决方案更加简单。同时, 所提出的方案描述了框架。同样地, 它可以被重复用于要求动态对象处理的许多不同的应用。这只是关于将通用对象生命周期角色映射到由具体应用提供的具体功能的问题。

[0042] 为了更好地理解, 将在下面的描述中参考附图更详细地解释本发明。应当理解的是, 本发明不限于该示例性实施例, 并且所指定的特征还可以方便地被合并和 / 或被修改,

而不背离由所附权利要求定义的本发明的范围。

附图说明

- [0043] 图 1 示意性地示出示例性 P/S 框架；
- [0044] 图 2 描绘根据本发明的、用于请求对对象的操纵的方法；
- [0045] 图 3 示意性地示出被配置为执行图 2 的方法的装置；
- [0046] 图 4 描绘根据本发明的、用于对对象的操纵的方法；
- [0047] 图 5 示意性地示出被配置为执行图 4 的方法的装置；
- [0048] 图 6 示出根据本发明的、具有对象管理器和对象提供器的 P/S 框架；
- [0049] 图 7 示出 OBJECT_REQUEST 从对象管理器到对象提供器的转移；
- [0050] 图 8 给出关于根据本发明的对象处理机制的概览；
- [0051] 图 9 示出使用所提出的对象生命周期管理实现的 CREAT 动作；
- [0052] 图 10 示出使用所提出的对象生命周期管理实现的 UPDATE 动作；
- [0053] 图 11 示出使用所提出的对象生命周期管理实现的 MOVE 动作；
- [0054] 图 12 示出使用所提出的对象生命周期管理实现的 DELETE 动作；

具体实施方式

[0055] 图 2 描绘根据本发明的、用于请求对对象的操纵的方法。在第一步骤中，确定 (10) 对象的对象标识符。还确定 (11) 接受器 (recipient) 的接受器标识符，接受器应当操纵对象。至少所确定的对象标识符和接受器标识符然后被包括在对象请求中，对象请求被输出 (12) 到发布 - 订阅环境中。取决于所期望的操纵，对象请求还包括对象的名称和对象的特性。

[0056] 图 3 中示意性地示出被配置为执行图 2 的方法的装置 20。装置 20 包括输入设备 (input) 21，输入设备 21 用于接收请求以发起用于对对象的操纵的请求。输入设备 21 例如是用户接口或者应用。对象确定单元 22 确定 (10) 对象的对象标识符，而接受器确定单元 23 确定 (11) 应当对对象进行操纵的接受器的接受器标识符。所确定的对象标识符和接受器标识符然后随同其它数据（如果需要的话）被包括在对象请求中，对象请求经由输出设备 (output) 24 被输出 (12) 到发布 - 订阅环境中。当然，装置 20 的不同元件同样可被组合成单个单元，或被实现为在处理器上运行的软件。

[0057] 图 4 描绘根据本发明的、用于操纵对象的方法。从发布 - 订阅环境中接收 (30) 对象请求，对象请求至少包括对象标识符和接受器标识符。响应于接收 (30) 对象请求，对象请求的接受器检查 (31) 是否需要操纵对象。如果需要的话，根据需要来操纵 (32) 对象。

[0058] 图 5 中示意性地示出被配置为执行图 4 的方法的装置 40。装置 40 包括输入设备 41，输入设备 41 被配置为从发布 - 订阅环境接收 (30) 对象请求，对象请求至少包括对象标识符和接受器标识符。响应于接收 (30) 对象请求，估计单元 42 检查 (31) 是否需要操纵对象。如果需要的话，操纵单元 43 根据需要来操纵 (32) 对象。在适当的情况下，被操纵的对象经由输出设备 44 被输出到发布 - 订阅环境。当然，装置 40 的不同元件同样可被组合成单个单元，或被实现为在处理器上运行的软件。

[0059] 根据本发明的 P/S 对象生命周期框架描述了被动态创建、修改、删除并移动的对

象在 P/S 环境中如何被处理。如图 6 中所示出的, 框架预见两个角色。对象管理器履行向对象提供器发送生命周期请求的角色。对象提供器对请求进行反应并执行所请求的针对特定对象的动作, 即, 创建、修改、删除或移动。对象提供器是对象的容器, 并在对象管理器的请求下处理对象的生命周期。多个对象管理器可以发出请求到多个对象提供器。此外多个对象提供器可以接收来自多个对象管理器的请求。

[0060] 对象管理器通常是用户经由用户接口进行控制的应用, 其对位于设备上的数据或者在设备上的应用中的数据进行动作。对象提供器是应用或设备内的控制单元, 或者是设备本身, 对一组对象进行分组和控制。对象提供器应对象管理器的请求进行动作。不同的对象提供器无需知道彼此的存在。它们是彼此独立的。

[0061] 为了给出关于可以是什么对象的一些示例, 在便签处理应用 (“便利贴”) 中, 对象是在任何特定的时间处可用的不同的便签。在文档处理应用中, 对象是对不同文档的表示。在多媒体应用中, 对象是由应用控制的媒体端点。

[0062] 下面将参考图 7 来解释单个对象生命周期请求, 单个对象生命周期请求表现了所有的对象生命周期方法。该请求的示例性格式如下:

[0063] OBJECT_REQUEST(Object_ID, Object_Provider_ID, Name, parameters(value))

[0064] 对象由 “主题” 名称和 Object_ID 来标识。特定对象的特性由与对象相关的参数来描述, 参数具有值。在 P/S 域内, Object_ID 号和 Object_Provider_ID 号是唯一的。

[0065] 例如, 如果对象是便签, 则每个便签可能具有名称, 名称是数字或标题, 特性可以指示尺寸、字体、最大字符。

[0066] 注意, 在 P/S 环境中, OBJECT_REQUEST 被认为是对象 / 标题本身, 当对象管理器改变参数值时, 对象 / 标题的参数值被发布。订阅 OBJECT_REQUEST 的所有对象提供器将接收请求。仅被寻址的对象提供器对请求采取行动。

[0067] 用于不同的生命周期动作的对象生命周期请求的使用如下:

[0068] CREATE(创建)

[0069] OBJECT_REQUEST(Object_ID = j, Object_Provider_ID = p, Name = n, parameters = value)

[0070] Object_ID j 是不存在的对象 (在任何对象提供器中)。对象提供器 p 在它的区域中创建该对象作为对请求的反应。OBJECT_REQUEST 中的值被分配为对象的参数。

[0071] DELETE(删除)

[0072] OBJECT_REQUEST(Object_ID = j, Object_Provider_ID = void)

[0073] 对象提供器 p 知道它在它的区域中具有对象 j。对象提供器 p 删除对象 j 作为对请求的反应。

[0074] UPDATE(更新)

[0075] OBJECT_REQUEST(Object_ID = j, Object_Provider_ID = p, Name = n, parameters = other value(其他值))

[0076] 对象提供器 p 知道它在它的区域中具有对象 j。对象提供器 p 利用所指示的新值更新对象 j 的参数作为对请求的反应。

[0077] MOVE(移动)

[0078] OBJECT_REQUEST(Object_ID = j, Object_Provider_ID = q, Name = n,

parameters = value)

[0079] 对象提供器 p 知道它在它的区域中具有对象 j。由于请求指明对象 j 现在属于对象提供器 q, 因此对象提供器从它的区域中删除对象 j。对象提供器 q 利用所指示的参数的值在它的区域中创建对象 j。

[0080] 图 8 给出关于根据本发明的对象处理机制的概览。

[0081] 经由对象管理器中的用户接口和 / 或应用, 输入 (1) 对象请求动作 (创建、更新……)。对象管理器中的 OBJECT_REQUEST 对象中的 OBJECT_REQUEST 对象数据被相应地改变。OBJECT_REQUEST 对象然后通过 P/S 域被发布 (2)。订阅了该 OBJECT_REQUEST 主题的所有对象提供器读取 OBJECT_REQUEST 对象。目标对象提供器看到请求并按所请求的来创建、更新、删除 (3) 所针对的对象。注意, 移动意味着一个对象提供器中的创建和另一对象提供器中的删除。该对象发布 (4) 它的状态和特性 (被创建、被更新、被删除)。注意, 对于移动, 有两个通知 (即, 被删除的和被创建的) 通过 P/S 域。最后, 在对象管理器用户接口或应用上示出 (5) 请求的结果。

[0082] 下面还使用便利贴应用的示例对本发明进行解释。便利贴被附到虚拟墙上。便利贴应用可以显示和操纵 (创建、更新、删除、移动) 来自存在于它的环境中的任何墙的便签。此外, 便利贴应用本身可以提供它自己的虚拟墙。

[0083] 如先前已经指出的, 在便利贴应用的交互中, 两个角色被标识。墙提供器起到虚拟墙的作用, 并将拥有被附到它的墙上的便签, 即负责发布。便签用户接口向用户显示便签, 并将控制呈现给用户以操纵便签。便利贴应用可以同时扮演两个角色。可以想象的是, 一些无头设备 (headless device) (例如, 网关) 可以实现仅用作墙提供器的应用。

[0084] 便利贴应用的数据模型包括三个主要部分, 即虚拟墙 (便签被附到该虚拟墙上) 的表示、便签本身的表示和便签操控请求的表示。这些便签操控请求包括创建便签、改变便签的文本、将便签移动到不同的墙以及删除便签。

[0085] 便签请求 (对应于上述 OBJECT_REQUEST) 的字段的值的每种可能的组合具有含义。每个请求或者与给定的墙提供器不相关, 或者可解释为创建、删除或更新请求。

[0086] 从便签用户接口的角度看, 解释同样简单。为了创建新的便签, 使用新的唯一的便签 ID (对应于 OBJECT_ID), 选择具有它的正确的墙 ID (对应于 OBJECT_PROVIDER_ID) 的墙并发布请求。对于更新现有便签, 用相应的便签 ID、当前的墙 ID 和新的消息来发布请求。为了移动便签, 用便签 ID、我们想要移动到的墙的墙 ID 以及现有消息来发布请求。实际上, 新墙的创建动作和旧墙的删除动作被结合在一个请求中。对于删除便签, 用便签 ID 和空的墙 ID 来发布请求, 消息 (对应于“特性”) 在本情况下不相关。

[0087] 图 9 示出了 CREATE (创建) 动作:

[0088] 1、用户创建具有消息“你好 (hello)”的新便签, 该新的便签将被附到 ID 为 1234 的墙上。便签用户接口选取便签的唯一 ID (在本情况中, 1), 并发布便签请求 (NoteRequest)。

[0089] 2、在接收到便签请求事件时, 墙提供器 1234 将它解释为针对便签的创建的请求。墙提供器 3456 忽略该请求, 因为该请求与它的墙无关。

[0090] 3、墙提供器 1234 然后向便签主题发布新的便签实例。

[0091] 4、便签用户接口被通知主题上的变化、发现新的便签并更新它的用户接口。

[0092] 图 10 示出了 UPDATE (更新) 动作:

[0093] 1、用户决定将便签的消息从“你好”变为“世界 (World)”。便签用户接口发布适当的便签请求。

[0094] 2、在接收到便签请求事件时，墙提供器 1234 将它解释为针对现有便签的更新请求。墙提供器 3456 忽略该请求，因为该请求与它的墙无关。

[0095] 3、墙提供器 1234 向便签主题发布经更新的便签实例。

[0096] 4、便签用户接口被通知主题的变化、发现经更新的便签并更新它的用户接口。

[0097] 图 11 示出 MOVE (移动) 动作：

[0098] 1、用户将便签移动到墙 3456。便签用户接口发布适当的便签请求。

[0099] 2、在接收到便签请求事件时，墙提供器 1234 将它解释为针对现有便签的删除请求。墙提供器 3456 将同一事件解释为针对新的便签的创建请求。

[0100] 3、墙提供器 1234 从主题中移除它的便签实例，墙提供器 3456 向主题发布新的便签实例。

[0101] 4、便签用户接口被通知主题上的变化、发现经更新的便签并更新它的用户接口。

[0102] 最后，图 12 示出 DELETE (删除) 动作：

[0103] 1、用户删除便签。便签用户接口发布适当的便签请求：它选择空的墙 ID (0)。

[0104] 2、在接收到便签请求事件时，墙提供器 1234 忽略该请求，因为该请求与它的墙无关。墙提供器 3456 将该请求解释为删除请求。

[0105] 3、墙 3456 从它的主题中移除它的便签。

[0106] 4、便签用户接口被通知主题上的变化、发现便签实例已经从主题中被移除并更新它的用户接口。

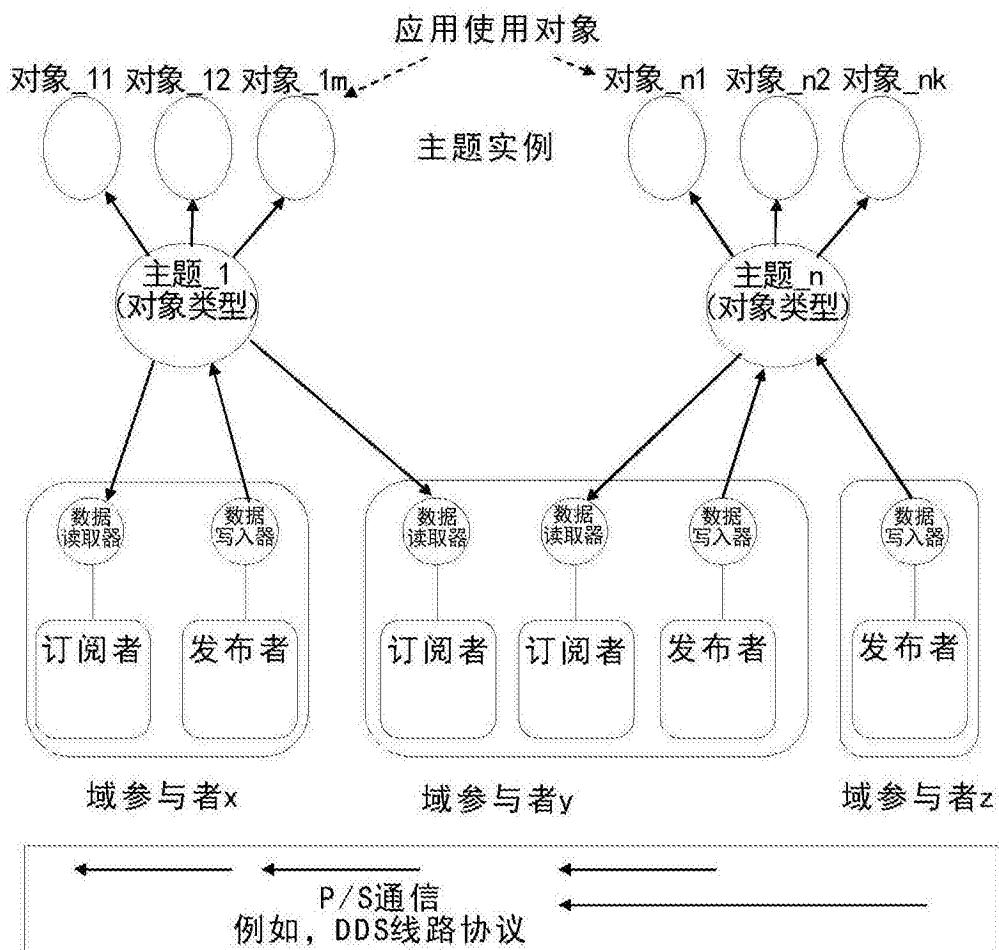


图 1

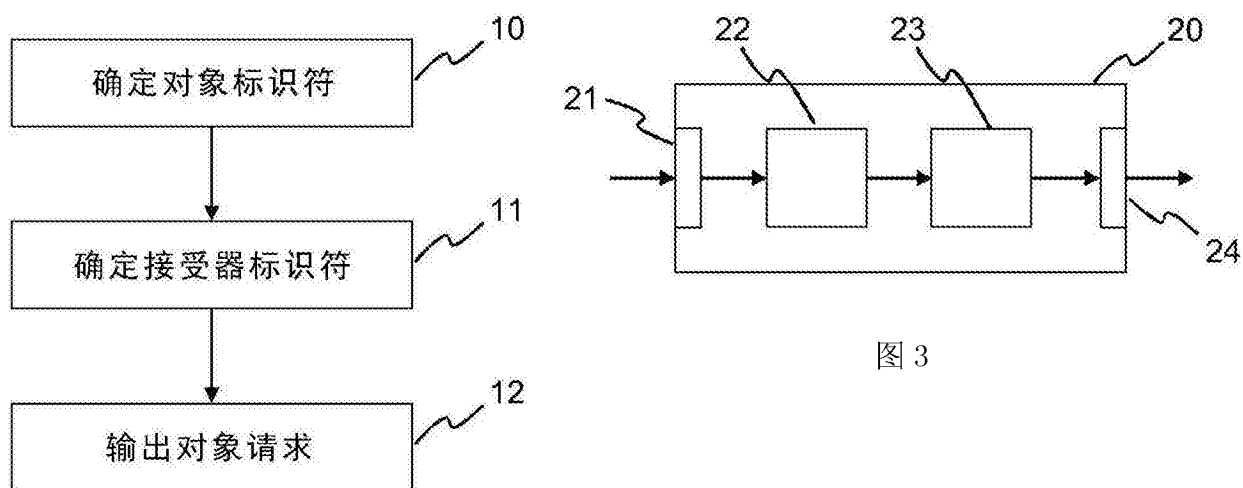


图 3

图 2

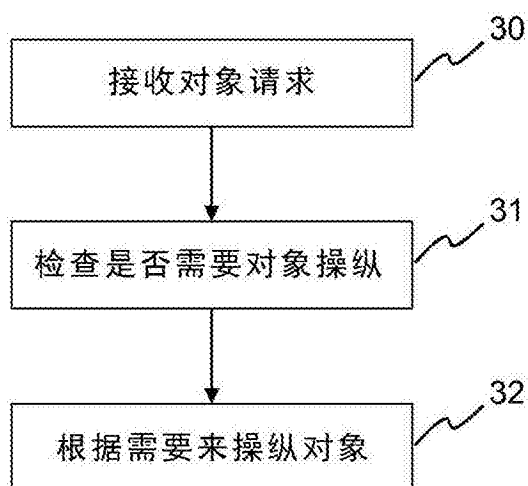


图 4

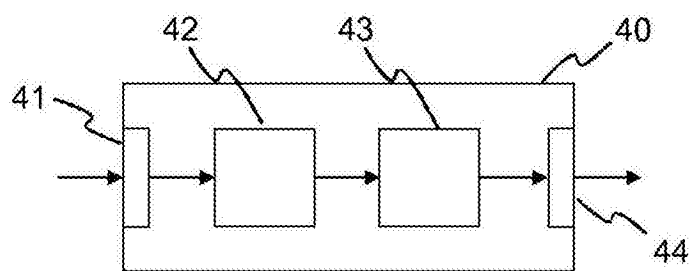


图 5

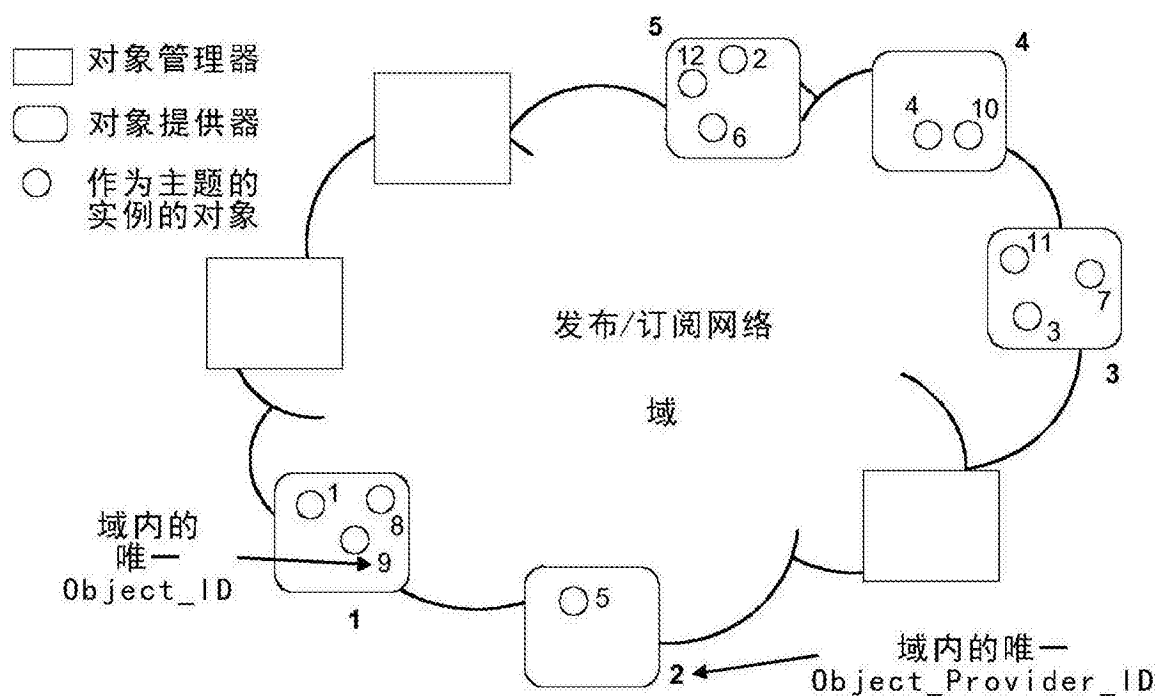


图 6

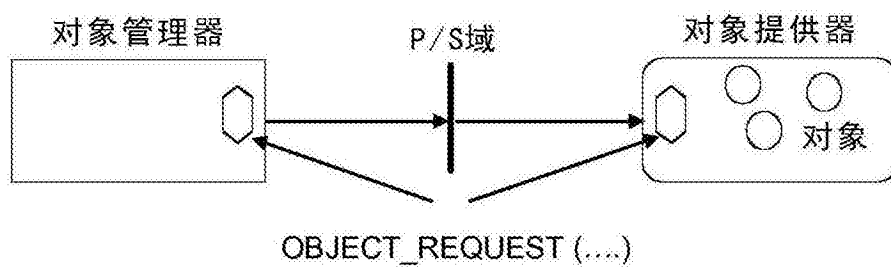


图 7

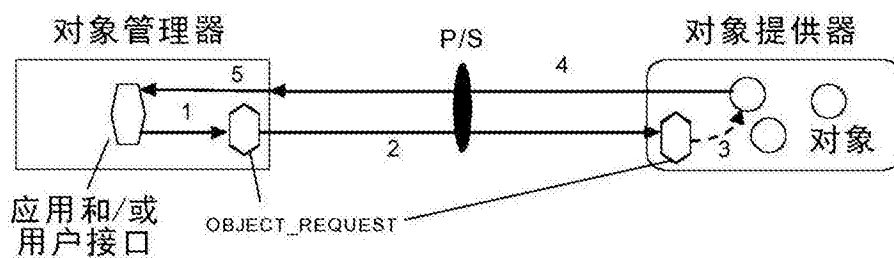


图 8

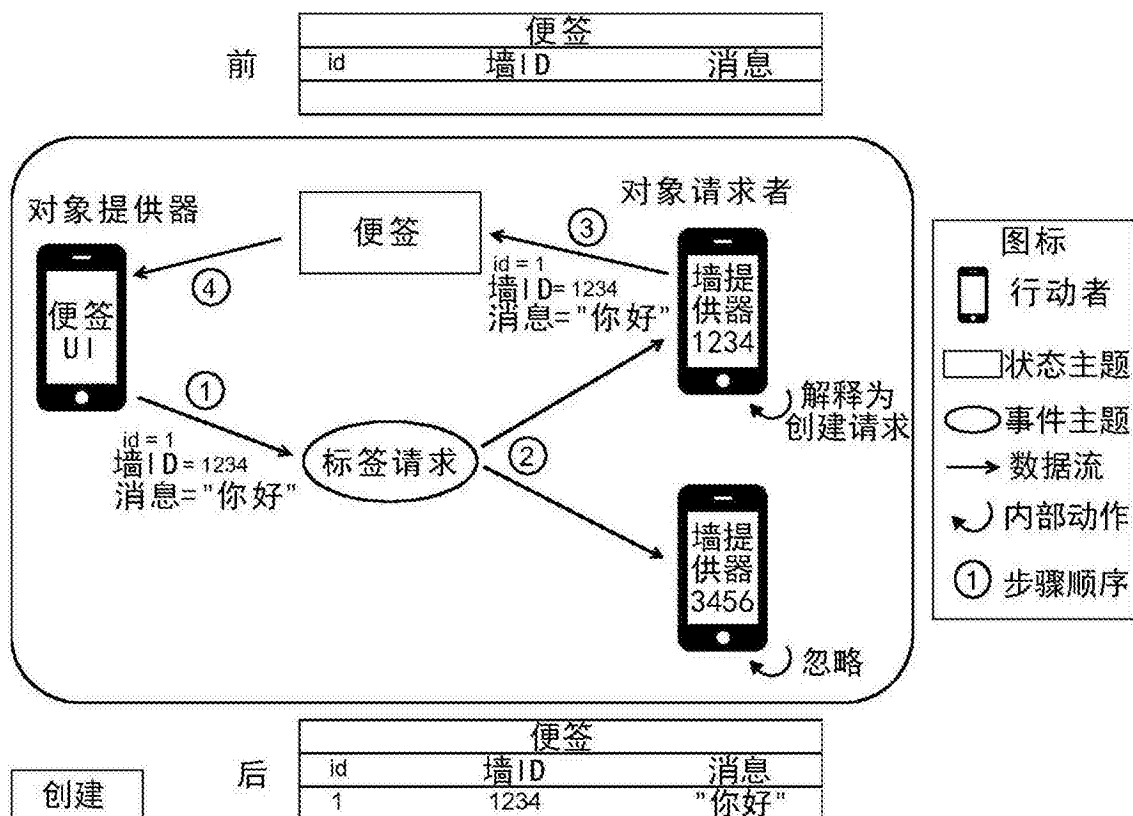


图 9

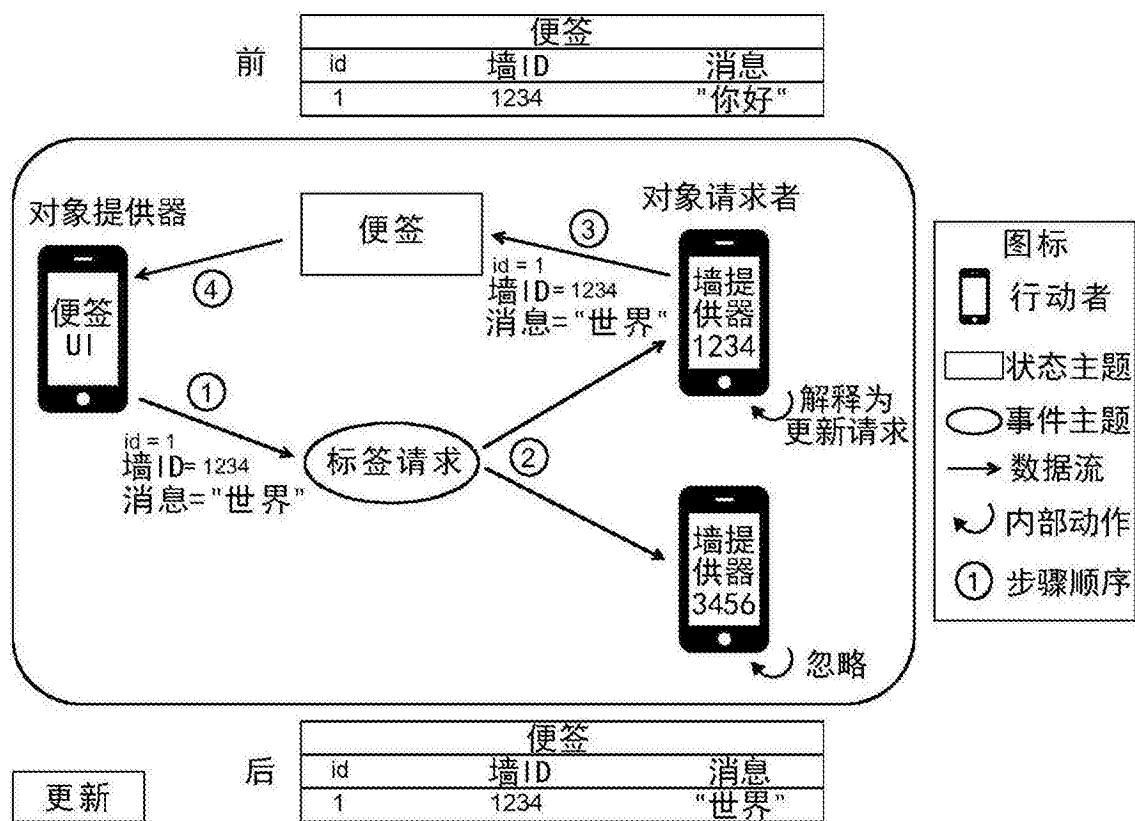


图 10

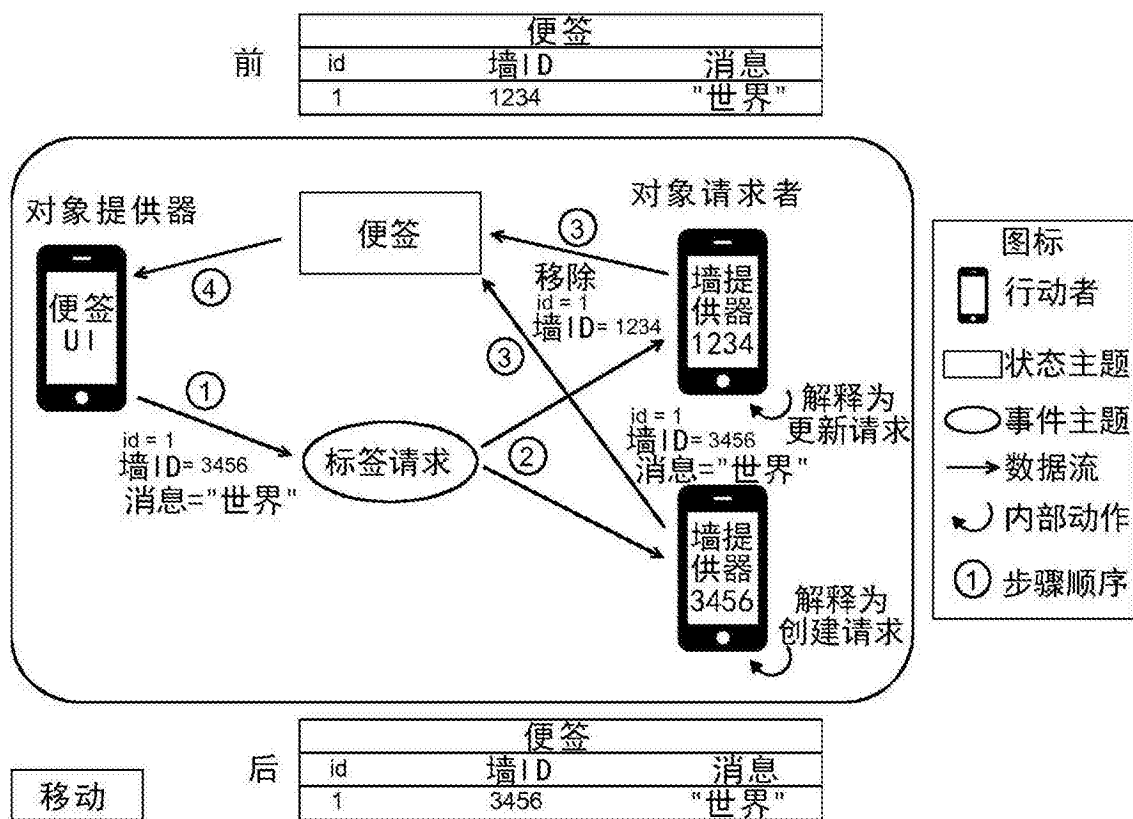


图 11

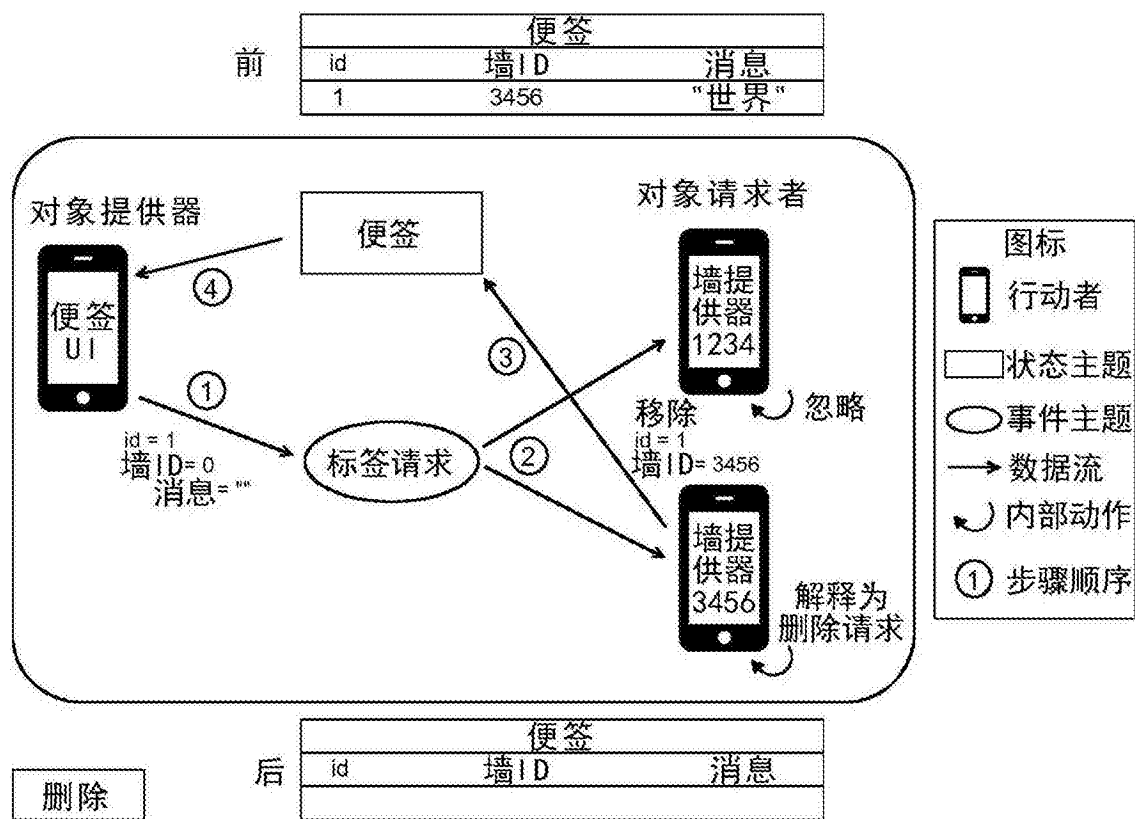


图 12