

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁷

G06F 9/46

G06F 15/16

[12] 发明专利说明书

[21] ZL 专利号 93101247.3

[45]授权公告日 2000年12月20日

[11]授权公告号 CN 1059747C

[22]申请日 1993.1.30 [24]颁证日 2000.9.16

[21]申请号 93101247.3

[30]优先权

[32]1992.1.31 [33]GB [31]9202072.6

[73]专利权人 马科尼通讯有限公司

地址 英国英格兰

[72]发明人 S·J·阿特金斯

J·N·弗洛加特 L·W·帕克

[56]参考文献

EP0454610A2 1991.10.30 G06F9/46

US4,007,450 1977.2.8 G06F15/16

US4,714,995 1987.12.22 G06F17/30

审查员 汪涛

[74]专利代理机构 中国专利代理(香港)有限公司

代理人 曹济洪 程天正

权利要求书 2 页 说明书 8 页 附图页数 0 页

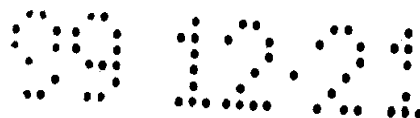
[54]发明名称 面向对象系统及其操作方法

[57]摘要

用客户/服务单位法来解决侧重于面向对象系统中各对象有多重用户的问题是众所周知的事。

本发明公开了另一种方法,即建立一个共享对象,将该共享对象复制出来分派给需要使用该对象的所有系统节点,并编制识别该存放该对象的复制品的诸节点的用户表。操作其中一个节点来更新该对象时,同时也就更新了该用户表所识别的复制品。

ISSN 1008-4274



权 利 要 求 书

1. 一种面向对象系统的操作方法，这种系统包括：
用以存储包括可修改的数据的对象的装置；以及
用以向至少一个远离所述存储对象的装置的进程提供对象的复制品的装置，每个所述至少一个进程包括一个用以存储所述对象的复制品的相应装置：

这种方法包括下列步骤：

向所述至少一个远离所述存储对象的装置的进程的其中至少一个提供对象的复制品；

在所述相应的存储装置中存储对象的复制品，

修改对象的可修改数据；

将有修改的情况通知已经存储了对象的复制品的那个或每个遥远进程，以及

响应来自所述那个或每个遥远进程修改数据的请求，向所述那个或每个遥远进程提供修改数据的复制品。

2. 根据权利要求1的方法，其特征在于还包括下列步骤：

修改所述对象的复制品的可修改数据；以及

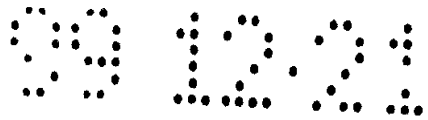
将所述复制品的修改数据传递给对象，从而使修改可修改数据的所述步骤修改对象的可修改数据，以对应于由所述对象的复制品所保存的修改数据。

3. 一种面向对象系统，包括：

用以存储包括可修改数据的对象的装置；

至少一个在远离所述用以存储对象的装置的进程中存储对象复制品的装置；

修改对象的可修改数据的装置；



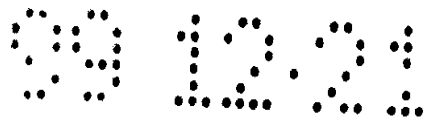
用以通知存储对象的复制品的所述至少一个装置已经修改了对象的可修改数据的数据的装置；以及

响应来自存储对象的复制品的遥远进程对于修改数据的复制品的请求向对象复制品提供对象的修改数据的复制品的装置。

4.根据权利要求3的系统，其特征在于包括：

用以修改对象的复制品的可修改数据的装置；

其中所述用以修改对象的可修改数据的装置修改对象的可修改数据，以对应于对象的复制品的修改数据。



说明书

面向对象系统及其操作方法

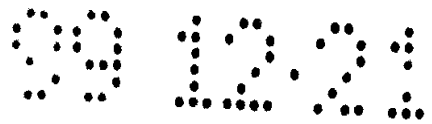
本发明涉及一种面向对象系统。

近年来，软件程序的研制上越来越多地采用一种新的编程技术，这种技术叫做面向对象技术，它适用于软件开发从分析到编码的生存周期的所有阶段。这种技术采用的主要原理是，建立一些对象，这些对象是一些数据和一些根据这些数据起作用的功能的集合体，从而使下列一些情况成立：

- a. 各对象仅通过互通信息彼此进行通信。
- b. 对象的使用者无需知道任何有关对象内部的执行程序，只需知道对象（通过接收有效信息）所能提供的服务。
- c. 所选择的对象应该与处理中的问题有一定的关系。一般说来，面向对象软件系统包含一些对象，这些对象是其软件待监控或控制的系统内各机构的概括。

本简单的说明书不是想全面说明作为一项技术的面向对象法，而是作为理解本发明的基础。虽然这里对对象下的定义是按软件的术语进行的，但我们也可以把对象视为软件与硬件的混合体，例如，给某一对象传递一个信息可能会产生影响受控制中的硬件状态的作用——对象的“软件”部分事实上仅仅是对象的“硬件”部分的接口机构。

鉴于电信系统中的对象只借助于互通信息进行通信，因而在建立面向对象的系统时给体系结构方面提供了很大的灵活性。这就产生了这样的问题，即无论在单个进程中存在的各对象之间互通信息，在同一个机器的个别进程中存在的各对象之间互通信息，



还是在各不同机器上的各对象之间互通信息，差别都不大。在各情况下，仅仅是互通信息的机构不同而已。这意味着，各对象、各进程和各机器三者之间再也不存在任何严密的关系。

“进程”一词在软件领域中应用面极其普遍，它是指软件的那一个与同一机器中的其它进程同时进行的部分，就是说，各指令在一个进程中的执行与各指令在另一个进程中的开始执行或完成无关。这种同时性通常是采用多重任务操作系统来共用系统各进程之间的处理时间达到的。如果不提供这种共用机理，则每个机器只能有一个进程。

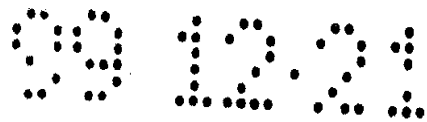
在面向对象的系统中，单个对象拥有多个使用者是常有的事。若一个对象的使用者是同一个进程内的一些对象，则不牵涉到同时性问题，而各信息是顺次到达的。若对象的使用者是不同进程中的一些对象，则必须考虑如何处理同时存取的问题。

虽然下面只将提到各项进程，但进程一词确实也可适用于分布在一个以上的机器上的对象。

上述问题原先是在采用一般的客户/服务单元模式的基础上提出的。在这个方法中，许多并行对象待使用的对象系配在其各自的进程中的，这个进程叫做服务单元进程。于是其它一些进程（现在叫做客户进程）可以将信息传递到服务单元进程中的对象，以便利用其性能。各项要求的同时性由服务单元进程接口处某些形式的排队机构处理，该排队机构是作为进程间信息传递机构的一部分而设的。

虽然这种机构行得通，但却具在许多缺点：

a. 服务单元进程的作用是通过串行化处理同时性问题的，即只将各信息排排队，逐次进行处理，实际上抛弃了试图达到的同时性。这方面的容许程度取决于所研究的实际系统，但在那些采



用多重机器达到同时性的系统上，这种方案会增加各客户等待服务单元回应的闲置时间，从而达不到最佳的运行效能。

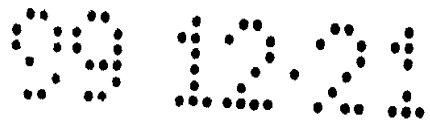
b. 信息在同一个进程内的各对象之间传递总是比在不同进程中各对象之间传递快得多。客户/服务单元法使系统不得不使用较慢的机构。若对象通信能尽量限制在一个进程中进行，则在运行效能方面取得很好的改进。

c. 一般说来，面向对象设计会产生大量需要由许多不同的进程使用的对象。客户/服务单元法兼顾了两种极端的作法：一种作法是把所有这些对象放进单个服务单元的进程中，另一种作法是把各对象放进自己的进程中。使用单个服务单元进程会使信息涌塞，从而使系统的运行慢到让服务单元实行排队轮候的服务方式那样的速度，而对象的数目通常是多到实际上因系统资源上的限制而不能将进程分派到各对象中的程度。在大系统中是不可能获得与所要求的系统运行效能相称的令人满意的折衷方案的。

为解决上述(a-c)的问题，我们提出了本发明的共享对象模式。

若各不同进程中的各对象想利用它们之间的另一个对象，则这个对象可视为共享对象。如果不建立一个服务单元进程，也可以在每一个想使用的进程中建立该共享对象的复制品。可以认为所有传递到该共享对象的信息都处在同一个进程中，此外，队列上的各信息不是串行化的，即保留了同性。这样做看起来极为理想，只是为使该共享对象成为共享对象，各进程中的各复制品必须完全相同。

对象的特点在于其内部数据和其方法（根据到来的信息作用到其内部数据的功能）。对象的内部数据通常是不断变化的，是随时间而变的。对象的方法则是变化较小，通常在系统建立时（或



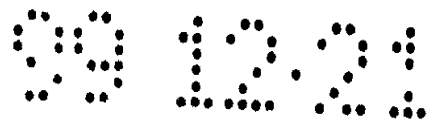
更早的时间) 就确定下来了, 且在系统的使用寿命中是不变的。因此复制共享对象时, 必须保持一致的仅仅是内部数据而已。若在一个进程内使用的方法只引起对共享对象内部数据的读出操作, 则没有问题。但如果方法起作用后促使内部数据发生变化, 即引起书写操作, 则必须改变共享对象的每一个复制品, 以保持一致性。

这只能通过进程之间互通信息实现。这时必须制定数据结构并维持这种结构, 这个数据结构详细说明了目前应用共享对象各进程。在其中一个进程修改共享对象的内部数据时, 可以利用此数据表改造或更新其它进程所保存的复制品。

在(英国剑桥 1989 年 8 月第 4 期第 32 卷的计算机杂志第 323 - 332 页)题为“阿见那的持久对象的处理(The Treatment of Persistent Objects in Arjuna)”的论文中讨论了一种在若干物理远程节点上可以复制对象的面向对象装置的计划。这个计划涉及配置容错的分布系统, 这些系统相对地抵抗个别节点的故障和消息的损失。持久对象存储在非易失性存储器中。要启动对象就必须将它从非易失性存储器复制到易失性存储器。然后评估易失性存储器中的复制品。如果存取涉及修改数据, 当存取正常终止时用修改的易失性复制品更新非易失性复制品。如果易失性复制品在使用时发生紧急事故, 就可以用非易失性复制品恢复操作。

如果一个节点希望使用存储在另一个节点上的共享对象, 那么在需要只读操作时, 直接形成和评估本地复制品, 从而不需要客户服务单元进程。

然而, 如果需要发生写操作, 则不形成本地复制品。而是在产生写操作的过程和对象的所有复制品之间必须形成个别客户服务单元进程。



这样就构成了复制的对象，以在功能上等效于单个对象，任何变化经过其相应的客户服务单元进程立即传送到所有的复制品。在这样保持一致性的同时，需要为每个复制品提供专用的客户服务单元进程，允许可以改变数据。

同时性借助于共享对象的多重复制品保持下来，因而本发明的共享对象法解决了同时性的问题。进程之间互通信息的方式目前仍在使用，只是只有当共享对象法会使内部数据发生变化时才应用这个方式，所有其它方法则使用快速的进程内互通信息的方式。这一下就消除了各对象往各服务单元进程的分派问题和与此有关的运行效能问题。这并不是说，不应采用客户/服务单元模式，仍然有某些情况使用这种方式比共享对象法切合实际。另外，共享对象法也有一些潜在的缺点，即：

a. 若许多进程共享一个对象，则必须保持大的数据表。（而在客户/服务单元方法中，这种情况就成了信息涌塞现象）。

b. 为确保始终如一地更新共享对象的内部状态，必须将对象加以闭锁，即确保不同时更新来自两个不同进程的数据。

本发明的第一方面提供一种面向对象系统的操作方法，这种系统包括：

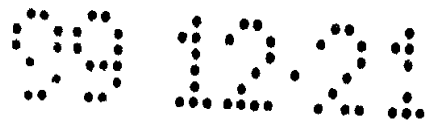
用以存储包括可修改的数据的对象的装置；以及

用以向至少一个远离所述存储对象的装置的进程提供对象的复制品的装置，每个所述至少一个进程包括一个用以存储所述对象的复制品的相应装置：

这种方法包括下列步骤：

向所述至少一个远离所述存储对象的装置的进程的其中至少一个提供对象的复制品；

在所述相应的存储装置中存储对象的复制品，



修改对象的可修改数据;

将有修改的情况通知已经存储了对象的复制品的那个或每个
遥远进程, 以及

响应来自所述那个或每个遥远进程修改数据的请求, 向所述
那个或每个遥远进程提供修改数据的复制品。

当在一个进程中形成对象的图象时, 记录进程使用对象的情
况。当任何进程对象的参考复制品时, 通知所有有对象图象的进
程。每个进程于是可以通过重读参考数据而重新调整对象自身的
显示。

本发明的第二方面提供一种面向对象系统, 包括:

用以存储包括可修改数据的对象的装置;

至少一个在远离所述用以存储对象的装置的进程中存储对象复
制品的装置;

修改对象的可修改数据的装置;

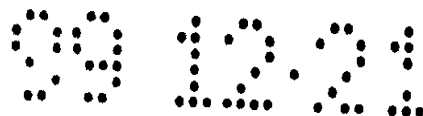
用以通知存储对象的复制品的所述于少一个装置已经修改了对
象的可修改数据的数据的装置; 以及

响应来自存储对象的复制品的遥远进程对于修改数据的复制品
的请求向对象复制品提供对象的修改数据的复制品的装置。

可以通过现时实际上存取共享对象的进程更新共享对象的数
据。使用通知书机构(Bouing)通知其它已经改变了对象的进程, 必
要时, 这些进程于是负责重读共享对象的数据。Bouing 机构不传送
数据。

现在以实例说明本发明的内容。

客户/服务单元法一般是在与某些形式的进程间通信(IPC) 的
客户和服务单元进程中采用为数不多的代码(有时叫做信息段)。
客户进程中代码的用途纯粹是传达对服务单元进程中的对象作为



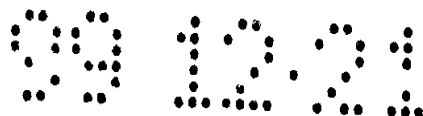
传递给采用 IPC 机构的服务单元进程的信息的任何要求。IPC 机构会将各信息在服务单元进程排排队，由服务单元进程中的小量代码读取该队列，并将各要求送往该对象，将它们加以处理。整个过程可以想象为一条管线，该管线从其一端输入来自一个或多个客户进程的各项要求，并在管线的另一端将这些要求传送给服务单元进程中合适的对象。全部处理这些要求（而不是仅仅只传送它）所需的活动是在服务单元进程中进行的。

分享对象法的采用使客户/服务单元“流水线”的机构成了在进程内采用分享对象局部处理各项要求的机构。只有当这些要求修改共享对象的内部数据时才会利用进程之间的通信来确保对象的其它复制品跟上共享对象的变化。

有待在一个系统中共享的对象必须在汇编时照此加以鉴定。这是通过确定想要显示该特性的各对象可继承的共享对象的类别达到的。每当建立一个共享对象时，在所要求的进程内就建立起该对象的复制品。若该对象是第一次建立的，则建立另外两个项目；该对象的数据其复制品是连同写入要求进程的标识符所在的用户表(used-by table)在共用存储器中建立的。该共享对象其后由其它进程的建立促使在其内部数据会是存储在共用存储器中的数据的真正复制品的请求进程内建立该对象的复制品，用户表则用请求进程的 ID(标识符)加以更新。

共用存储器可以视为可从任何进程加以访问的全局数据区。

每当提出关于共享对象的要求时，该要求就由进程局部加以处理。若该要求促使该共享对象的内部数据的修改，则更新共享对象数据的共用存储复制品以反映该变化。除此之外，还需要用经修改的共享对象给每一个其它的进程发送信号表明已发生了变化，使它们能使它们共享对象的各局部复制品与存储在共用寄存



器中经修改的数据重新组合。这个信号发送的过程叫做更新广播（或 BOU），它是利用用户表给所有作为共享对象的用户登记的所有进程发送信号。含有共享对象的各进程必须含有确保 BOU 信号必然引起必要的更新的 BOU 信号处理器。

当然进程有可能在某时候再不需要共享某一对象。发生这种情况时，可从进程除去该共享的对象，并修改用户表，以删除各请求进程的标识。