



(12) 发明专利

(10) 授权公告号 CN 103049306 B

(45) 授权公告日 2016. 05. 11

(21) 申请号 201210335226. 8

(56) 对比文件

(22) 申请日 2012. 09. 11

CN 1664779 A, 2005. 09. 07,

(30) 优先权数据

US 2003014555 A1, 2003. 01. 16,

61/533, 747 2011. 09. 12 US

US 2010299659 A1, 2010. 11. 25,

13/350, 552 2012. 01. 13 US

审查员 郭涛

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 S·法克斯 J·D·卡普兰

L·普罗塞克 R·M·拜尔斯

R·A·拜因顿

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 段登新

(51) Int. Cl.

G06F 9/455(2006. 01)

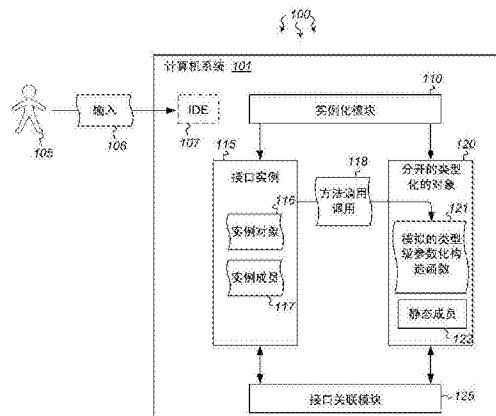
权利要求书2页 说明书7页 附图4页

(54) 发明名称

基于接口的 API 上的静态成员和参数化构造函数函数的模拟

(57) 摘要

本发明涉及基于接口的 API 上的静态成员和参数化构造函数的模拟。实施例涉及在接口实例上提供类型级构造的模拟, 并且涉及维护基于令牌的事件处理系统。在一个场景中, 计算机系统实例化一接口实例, 该接口实例包括实例对象和多个实例成员。该计算机系统还实例化一分开的类型化的对象, 该对象被配置成维护一个或多个模拟的类型级参数化构造函数。该计算机系统随后将经实例化的接口实例与该分开的类型化的对象相关联。该分开的类型化的对象将方法调用从该接口实例路由到各模拟的类型级参数化构造函数和该类型的静态成员。



1. 一种用于在包括存储器和至少一个处理器的计算机系统中在接口实例上提供类型级构造的方法,所述方法包括:

实例化接口实例的动作,所述接口实例包括实例对象和一个或多个实例成员;

实例化分开的类型化的对象的动作,所述分开的类型化的对象被配置成维护一个或多个模拟的类型级参数化构造函数;

将所述实例化的接口实例与所述分开的类型化的对象相关联的动作,其中所述分开的类型化的对象被配置成将方法调用从所述接口实例路由到一个或多个模拟的类型级参数化构造函数和所述类型的静态成员;

运行时标识所述接口实例上的回调方法的动作;

所述运行时创建具有调用方法的委托类型的动作,所述调用方法具有与所述回调方法的签名完全相同的签名;

在确定所述委托已被调用之后,所述运行时将所述回调方法转发到应用二进制接口 ABI 方法的动作;以及

将所述回调方法的结果作为所述委托调用的结果返回的动作,以使得所述运行时提供在接口的 ABI 上的委托 API 体验。

2. 如权利要求 1 所述的方法,其特征在于,所述接口实例被实现为模拟一个或多个类型级操作。

3. 如权利要求 1 所述的方法,其特征在于,多个不同的接口实例与所述分开的类型化的对象相关联以向实现所述接口实例的应用开发者提供统一视图。

4. 如权利要求 1 所述的方法,其特征在于,所述接口实例的所述实例成员被路由到所述模拟的类型级参数化构造函数。

5. 如权利要求 4 所述的方法,其特征在于,所述模拟的类型级参数化构造函数包括一个或多个类型级操作。

6. 如权利要求 1 所述的方法,其特征在于,所述实例成员包括静态成员。

7. 如权利要求 6 所述的方法,其特征在于,所述静态成员被所述分开的类型化的对象模拟。

8. 如权利要求 6 所述的方法,其特征在于,执行所述静态成员的运行时尝试被重定向到所述静态成员的相关联的接口实例。

9. 如权利要求 8 所述的方法,其特征在于,所述重定向允许所述运行时在实现基于接口的调用的所述接口实例的顶上模拟静态成员的存在。

10. 一种用于在包括存储器和至少一个处理器的计算机系统中在接口实例上提供类型级构造的系统,所述系统包括:

用于实例化接口实例的装置,所述接口实例包括实例对象和一个或多个实例成员;

用于实例化分开的类型化的对象的装置,所述类型化的对象被配置成维护一个或多个模拟的类型级参数化构造函数;

用于将所述实例化的接口实例与所述分开的类型化的对象相关联的装置,其中所述分开的类型化的对象被配置成将方法调用从所述接口实例路由到一个或多个模拟的类型级参数化构造函数和所述类型的静态成员;

用于运行时标识所述接口实例上的回调函数的装置;

用于所述运行时创建具有调用方法的委托类型的装置,所述调用方法具有与所述回调方法的签名等同的签名;

用于在确定所述委托已被调用之后,所述运行时将所述回调方法转发到应用二进制接口 ABI 方法的装置;以及

用于将所述回调方法的结果作为所述委托调用的结果返回的装置,以使得所述运行时提供在接口的 ABI 上的委托 API 体验。

11. 如权利要求 10 所述的系统,其特征在于,使用签名将一个或多个模拟的类型级参数化构造函数与一个或多个本机方法相匹配以检测针对其分派方法调用的适当的接口实例。

## 基于接口的 API 上的静态成员和参数化构造函数的模拟

### 技术领域

[0001] 本发明涉及基于接口的 API 上的静态成员和参数化构造函数的模拟。

[0002] 背景

[0003] 计算机已变得高度集成于工作、家庭、移动设备以及许多其他地方中。计算机可快速且有效地处理大量信息。被设计成在计算机系统上运行的软件应用允许用户执行包括商业应用、学校作业、娱乐等等在内的各种各样的功能。软件应用通常被设计成执行特定的任务，诸如用于草拟文档的文字处理器应用或者用于发送、接收和组织电子邮件的电子邮件程序。

[0004] 这些软件应用通常是使用集成开发环境 (IDE) 设计和创建的。IDE 允许开发者访问协助软件程序的创建的不同元素和构造。不同的 IDE 可能允许使用不同的编程语言。例如，某些 IDE 更适于面向对象的软件开发。这些 IDE 可具有编辑器、编译器、调试器、类浏览器、对象检查器和允许开发者控制和开发应用的每一方面的其他工具。在某些情况下，面向对象的 IDE 允许开发者通过提供参数信息构造对象，并且允许使用具有某些类型的静态成员。

[0005] 简要概述

[0006] 此处描述的实施例涉及在接口实例上提供类型级构造 (type-level construct) 的模拟，并且涉及维护基于令牌的事件处理系统。在一个实施例中，计算机系统实例化一接口实例，该接口实例包括实例对象和多个实例成员。该计算机系统还实例化一分开的类型化的对象，该对象被配置成维护一个或多个模拟的类型级参数化构造函数 (parameterized constructor)。该计算机系统随后将经实例化的接口实例与该分开的类型化的对象相关联。该类型化的对象将方法调用从该接口实例路由到各模拟的类型级参数化构造函数和该类型的静态成员。

[0007] 在另一实施例中，计算机系统维护基于令牌的事件处理系统。该计算机系统实例化多个编程对象的事件订阅表，该表被配置成将令牌映射到事件订阅器和它们订阅的事件。事件订阅表为事件订阅中的每一个维护对订阅者的弱参考。如此，在事件订阅表不强迫订阅者保持存活 (alive) 的情况下跟踪订阅。计算机系统随后确定至少一个事件不再具有任何订阅，并且基于所述确定，将该订阅从事件订阅表中移除。当被所述表跟踪的所有时间订阅均被移除时，所述事件订阅表此时也可被释放 (free)。

[0008] 提供本概述以便以简化形式介绍将在以下详细描述中进一步描述的一些概念。本概述并非旨在标识所要求保护的的主题的关键特征或必要特征，也不旨在用于帮助确定所要求保护的的主题的范围。

[0009] 附加的特征和优点将在以下的描述中被阐述，并且部分地可通过该描述而对本领域技术人员显而易见，或者可通过对本文中的教示的实践来习得。本发明的实施例的特征和优点可以通过在所附权利要求中特别指出的手段和组合来实现并获得。本发明的实施例的特征将从以下描述和所附权利要求书中变得完全显而易见，或者可通过如下所述对本发明的实践而习得。

### [0010] 附图简述

[0011] 为了进一步阐明本发明的各实施例的以上和其他优点和特征,将参考附图来呈现本发明的各实施例的更具体的描述。可以理解,这些附图只描绘本发明的典型实施例,因此将不被认为是对其范围的限制。本发明的实施例将通过使用附图用附加特征和细节来描述和解释,附图中:

[0012] 图 1 示出了本发明的实施例可在其中操作的计算机体系结构,包括在接口实例上提供类型级构造的模拟并维护基于令牌的事件处理系统。

[0013] 图 2 示出了用于在接口实例上提供类型级构造的模拟的示例方法的流程图。

[0014] 图 3 示出了用于维护基于令牌的事件处理系统的示例方法的流程图。

[0015] 图 4 示出了其中在基于令牌的事件处理系统中维护事件订阅的本发明的实施例。

### [0016] 具体描述

[0017] 此处描述的实施例涉及在接口实例上提供类型级构造(type-level construct)的模拟,并且涉及维护基于令牌的事件处理系统。在一个实施例中,计算机系统实例化一接口实例,该接口实例包括实例对象和多个实例成员。该计算机系统还实例化一分开的类型化的对象,该对象被配置成维护一个或多个模拟的类型级参数化构造函数(parameterized constructor)。该计算机系统随后将经实例化的接口实例与该分开的类型化的对象相关联。该类型化的对象将方法调用从该接口实例路由到各模拟的类型级参数化构造函数和该类型的静态成员。

[0018] 在另一实施例中,计算机系统维护基于令牌的事件处理系统。该计算机系统实例化多个编程对象的事件订阅表,该表被配置成将令牌映射到事件订阅器和它们订阅的事件。事件订阅表为事件订阅中的每一个维护对订阅者的弱参考。如此,在事件订阅表不强迫订阅者保持存活(alive)的情况下跟踪订阅。计算机系统随后确定至少一个事件不再具有任何订阅,并且基于所述确定,将该订阅从事件订阅表中移除。当被所述表跟踪的所有时间订阅均被移除时,所述事件订阅表此时也可被释放(free)。

[0019] 以下讨论现涉及可以执行的多种方法以及方法动作。应当注意,虽然这些方法动作可能是按一定次序讨论的,或者是在流程图被描绘为是按照特定顺序进行的,然而并非必然需要特定的次序,除非特别声明,或者是在一个动作被执行之前因为该动作依赖于另一动作的完成而需要的情况。

[0020] 本发明的各实施例可包括或利用专用或通用计算机,该专用或通用计算机包括诸如例如一个或多个处理器和系统存储器等计算机硬件,如以下更详细讨论的。本发明范围内的各实施例还包括用于承载或存储计算机可执行指令和/或数据结构的物理和其他计算机可读介质。这样的计算机可读介质可以是可由通用或专用计算机系统访问的任何可用介质。以数据形式存储有计算机可执行指令的计算机可读介质是计算机存储介质。承载计算机可执行指令的计算机可读介质是传输介质。由此,作为示例而非限制,本发明的各实施例可包括至少两种显著不同的计算机可读介质:计算机存储介质和传输介质。

[0021] 计算机存储介质包括 RAM、ROM、EEPROM、CD-ROM、基于 RAM、闪存、相变存储器(PCM)、或其它类型的存储器的固态驱动器(SSD)、或者其他光盘存储、磁盘存储或其他磁存储设备、或可用于以计算机可执行指令、数据或数据结构形式存储所期望的程序代码装置且可被通用或专用计算机访问的任何其他介质。

[0022] “网络”被定义成允许在计算机系统和 / 或模块和 / 或其他电子设备之间传输电子数据的一个或多个数据链路和 / 或数据交换机。当信息通过网络(硬连线、无线、或者硬连线或无线的组合)被传输或提供给计算机时,该计算机将该连接适当地视为传输介质。传输介质可以包括如下的网络:所述网络可以用于运送计算机可执行指令形式或数据结构形式的数据或所期望的程序代码装置,并且可以被通用或专用计算机访问。上述的组合也应被包括在计算机可读介质的范围内。

[0023] 此外,在到达各种计算机系统组件之后,计算机可执行指令或数据结构形式的程序代码装置可从传输介质自动传输到计算机存储介质(或反之亦然)。例如,通过网络或数据链路接收到的计算机可执行指令或数据结构可被缓冲在网络接口模块(例如,网络接口卡或“NIC”)内的 RAM 中,然后最终被传输给计算机系统 RAM 和 / 或计算机系统处的较不易失性的计算机存储介质。因而,应当理解,计算机存储介质可被包括在还利用(或甚至主要利用)传输介质的计算机系统组件中。

[0024] 计算机可执行(或计算机可解释)指令例如包括致使通用计算机、专用计算机、或专用处理设备执行某个功能或某组功能的指令。计算机可执行指令可以是例如二进制代码、诸如汇编语言之类的中间格式指令、或甚至源代码。尽管用结构特征和 / 或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于上述特征或动作。相反,上述特征和动作是作为实现权利要求的示例形式而公开的。

[0025] 本领域的技术人员将理解,本发明可以在具有许多类型的计算机系统配置的网络计算环境中实践,这些计算机系统配置包括个人计算机、台式计算机、膝上型计算机、消息处理器、手持式设备、多处理器系统、基于微处理器的或可编程消费电子设备、网络 PC、小型计算机、大型计算机、移动电话、PDA、寻呼机、路由器、交换机等等。本发明还可在分布式系统环境中实践,在该环境中本地和远程计算机系统通过网络(经由硬连线数据链路、无线数据链路、或者经由硬连线和无线数据链路的组合)链接起来,每个计算机系统都执行任务(例如,云计算、云服务等)。在分布式系统环境中,程序模块可位于本地和远程存储器存储设备中。

[0026] 图 1 示出了可在其中采用本发明的原理的计算机体系结构 100。计算机架构 100 包括计算机系统 101。计算机系统 101 可以是任何类型的本地或分布式计算机系统,包括云计算系统。计算机系统可包括用于执行各种不同任务的各种不同模块。例如,实例化模块 110 可被配置成实例化接口实例(例如,115)、类型化的对象(例如,120)或其他元素。计算机系统可被配置成运行集成开发环境(IDE),该集成开发环境允许程序开发者开发应用。IDE 107 可从开发者 105 接收用户输入 106,所述用户输入指示要由该 IDE 以及计算机系统的其他模块提供的各种功能。这些模块的功能将在下面进一步描述。

[0027] 一般而言,此处描述的实施例允许运行时(runtime)将接口实例 115 与类型 120 相关联,该接口实例随后可用于模拟参数化构造函数 121 的效果以及该类型的静态成员。类似地,运行时可提供对象 120 的实例,该实例可用于访问对该运行时所提供的类型的类型级操作。

[0028] 从而,实施例允许将一组接口实例与一类型相关联,从而向运行时提供模拟在基于接口的应用二进制接口顶上的类型级操作。在某些情况下,用于使基于接口的应用二进制接口适应于面向对象的系统的其他实现限制可由运行时对实例方法执行的操作的种类。

此处描述的实施例为静态成员和构造函数提供参数。

[0029] 运行时可将接口 115 的一个或多个实例与类型 120 相关联。当该运行时执行一具有参数的类型的构造函数时,该运行时扫描该接口以寻找具有一组匹配的参数的方法并返回所指的类型的实例。随后,使用返回值作为新对象,通过向所选择的方法分派,该运行时模拟该构造函数。这与构造运行时可调用包装器(wrapper)的其他技术不同,所述其他技术简单地通过调用单一公知 API 来模拟构造。

[0030] 实施例还允许运行时通过使用与类型相关联的接口实例来模拟静态成员。每个静态成员与接口上的一绑定到定义所述静态成员的类型成员相关联。当该运行时尝试执行静态成员时,访问反而被重定向到其相关联的接口。这种重定向允许该运行时在应用二进制接口的顶上模拟静态成员的存在,其要求基于接口的调用。至少在某些情况下,其他版本的运行时可调用包装器技术不允许模拟静态成员,反而要求该运行时暴露仅具有实例成员的对象模型。

[0031] 类似地,该运行时可将类型级操作模拟为对由该运行时向该应用二进制接口提供的对象的实例操作。该运行时将创建单一类型对象,该单一类型对象实现用于该类型上的参数化构造函数和静态成员的接口。当调用被分派到表示参数化构造函数的接口方法时,该运行时将该调用分派到与被调用的接口方法具有相同方法参数的构造函数并返回结果。当对表示一类型的静态成员的接口方法做出调用时,由运行时提供的对象调用所指的类型的静态成员来满足该接口请求。

[0032] 实施例还描述了运行时可用来管理担当事件源的运行时可调用包装器的寿命的不同方法。这些方法可用于仅通过该运行时保持事件源包装器存活(alive),只要托管(host)在该运行时内的事件订阅者需要它。

[0033] 在一个实施例中,该运行时维护对绑定于该事件源的运行时可调用包装器的每个事件订阅者的一组弱参考。在事件订阅表中维护这些参考。当该事件源被订阅时,对该订阅者的弱参考被添加到该组弱参考中。类似地,当事件订阅被终止时,其相应的弱参考被从该组活动订阅者移除。当该运行时相信它能够最终化该事件订阅表时,它首先检查该组弱参考来看是否有任何弱参考仍存活。如果这些参考中的任何一个存活,则与该事件订阅表相关联的事件订阅表不再被考虑为由该运行时最终化的候选。

[0034] 用于事件处理的另一寿命管理技术由维护每个事件源运行时可调用包装器的参考计数的运行时实现。从该系统参考此事件源递增此参考计数。当此参考计数为非 0 时,该运行时维护对该事件源的强参考,从而防止它够资格被垃圾收集。当参考计数击中 0 时,该运行时释放其对该事件源的强参考。随后当不再被运行程序需要时,与该事件源相关联的事件订阅表被允许变得够格被垃圾收集。

[0035] 而且,实施例提供了一种用于运行时在表示回调方法的接口上模拟委托对象的方式。其他实现可能要求委托抽象被打断而客户端代码直接对应用二进制接口(ABI)做出回调。如此处所述,为了模拟委托,该运行时标识该接口实例上的回调方法。随后它创建具有“调用”方法的委托类型,该“调用”方法具有与该回调方法的签名完全相同的签名。当该委托被直接调用或经由事件被调用时,该运行时将该委托转发到底层 ABI 方法。该回调方法的结果随后被作为委托调用的结果返回。以此方式,该运行时提供接口的 ABI 上的委托 API 体验。

[0036] 类似地,运行时可为 ABI 调用者提供供使用的接口,该接口映射到由该运行时托管的委托的实例。该运行时将创建实现具有单一“调用”方法的新接口的对象,该方法具有与该委托的签名相匹配的签名。当外部调用者对该对象的接口方法做出调用时,该运行时调用它正包装的委托。该委托调用的结果随后作为该方法调用的结果被传递。这允许该运行时提供在委托上的基于接口的 ABI。下面,分别针对图 2 和图 3 的方法 200 和 300,进一步解释这些概念。

[0037] 考虑到以上描述的系统 and 体系结构,参考图 2 和图 3 的流程图将更好地理解根据所公开的主题实现的方法。为了解释简明起见,这些方法被示出和描述为一系列框。然而,应该理解和了解,所要求保护的主体不受框的次序的限制,因为一些框可按不同的次序进行和 / 或与此处所描绘和描述的其他框同时进行。此外,并非全部所示的框都是实现下面所述的方法所必需的。

[0038] 图 2 示出了用于在接口实例上提供类型级构造的模拟的方法 200 的流程图。现在将频繁参考环境 100 的组件和数据来描述方法 200。

[0039] 方法 200 包括实例化接口实例的动作,该接口实例包括实例对象和一个或多个实例成员(动作 210)。例如,实例模块 110 可实例化接口实例 115。尽管图 1 中仅示出一个接口实例,然而将理解,可实例化基本上任何数量的接口实例。实例 115 包括至少一个实例对象 116,以及一个或多个实例成员 117。至少在一些情况下,这些实例成员可包括静态成员。取决于实现,每个实例化的接口实例可包括不同数量的实例对象和 / 或(静态)实例成员。一旦被实例化之后,该接口实例可被实现以模拟类型级操作。

[0040] 方法 200 进一步包括实例化分开的类型化的对象 120 的动作,其中该分开的类型化的对象 120 被配置成维护一个或多个模拟的类型级参数化构造函数 121 (动作 220)、以及将实例化的接口实例 115 与分开的类型化的对象 120 相关联的动作,其中该分开的类型化的对象被配置成将方法调用 118 从该接口实例路由到该类型的一个或多个模拟的类型级参数化构造函数 121 和静态成员 122 (动作 230)。在某些情况下,如前面提到的,多个不同接口实例可被模块 110 按顺序实例化或同时实例化。这些实例化的接口实例中的每一个可被(例如,接口关联模块 125) 与分开的类型化的对象 120 相关联以向实现该接口实例的应用开发者提供统一视图。从而,例如,如果开发者 105 正在使用 IDE 107 开发应用,在该 IDE 中使用的接口实例可与分开的类型化的对象相关联以为开发者提供统一视图。

[0041] 在某些实施例中,该接口实例的实例成员 117 可被路由到模拟的类型级参数化构造函数 120。这些模拟的类型级参数化构造函数可包括一个或多个类型级操作。这些操作可被该接口实例使用。模拟的类型级参数化构造函数还可被用于模拟该接口实例上的委托对象。该委托对象可表示各种回调方法。

[0042] 如同前面提到的,可使用签名将模拟的类型级参数化构造函数 121 与本机方法(native method)相匹配。此处所使用的本机方法是指计算机系统 101 或计算机系统 101 的操作系统可本机地(natively)访问的方法。将模拟的类型级参数化构造函数和本机方法相匹配以检测要对其分派方法调用的适当接口实例。然后,在接收对表示指定类型的静态成员 122 的接口方法的调用之后,该分开的类型化的对象 120 调用该指定类型的实例成员 117 来满足该接口方法调用。所述静态成员可被分开的类型化的对象 121 模拟。执行所述静态成员的任何运行时尝试可被重定向到所述静态成员的相关联的接口实例 115。所述



重定向允许该运行时在实现基于接口的调用 118 的接口实例的顶上模拟静态成员的存在。该运行时从而可标识该接口实例上的回调方法,创建具有调用方法的委托类型(该调用方法具有与该回调方法的签名完全相同的签名),确定该委托已被调用,将该回调方法转发到适当的应用二进制接口(ABI)方法,并且将回调的结果作为委托调用的结果返回。如此,该运行时提供接口的 ABI 上的委托 API 体验。

[0043] 相应地,此处描述的实施例可提供接口实例上的类型级构造的模拟。如此,通过具有维护类型级构造的分离的类型化的对象,可提供参数化构造函数和静态成员。单个接口实例从而可将其方法调用通过模拟的类型级参数化构造函数路由。

[0044] 图 3 示出了用于维护基于令牌的事件处理系统的方法 300 的流程图。现在将频繁参照图 4 的环境 400 的组件和数据来描述方法 300。

[0045] 方法 300 包括实例化多个编程对象的事件订阅表的动作,该表被配置成将一个或多个令牌映射到事件订阅器和它们订阅的事件(动作 310)。例如,实例化模块 440 可建立事件订阅表 445 并将其作为基于令牌的事件处理系统的一部分来维护。该事件订阅表可包括用于不同编程对象 450 的多个不同的事件订阅(例如,446A 和 446B)。如此,当事件在特定编程对象上发生时,对该事件的订阅者被通知。所述事件订阅具有订阅者(447A/447B)和用于每个订阅者的相关联事件。例如,事件订阅 446A 中的订阅者 447A 订阅事件 A (448A) 和事件 B (448B)。事件订阅者 447B 订阅事件 A (448B)、事件 C (448C) 和事件 D (448D)。将要理解,每个订阅者可订阅基本上任何数量的不同事件。

[0046] 方法 300 还可包括事件订阅表为所述事件订阅中的每一个维护到订阅者的弱参考的动作,使得所述订阅能被跟踪而事件订阅表不必迫使订阅者保持存活(动作 320)。事件订阅表 445 可为每个事件订阅(446A/446B) 建立并维护对订阅者(447A/447B) 的弱参考 441。该弱参考跟踪该订阅而不迫使订阅者保持存活(因此将参考标识为“弱”参考)。相应地,事件订阅表将不迫使订阅者保持存活来维护该参考。

[0047] 在某些情况下,计时器 445 可用于确定该订阅已保持存活了指定时间量。然后,基于所确定的时间量,可将该订阅移除。在其他情况下,开发者可简单地移除对一段时间的订阅,或开发者可使用订阅移除模块 460 来移除事件所绑定到的编程对象。不管原因如何,事件可能到达这样一点:它不再具有任何订阅者。在该点,事件订阅表可确定该事件不具有任何订阅(动作 330)。

[0048] 在此订阅移除期间,事件订阅表不保持订阅者或事件存活。而且,至少在某些情况下,在确定对于给定事件没有更多订阅要跟踪之后,事件订阅表可自动移除它自身。还应当注意,至少在某些实施例中,编程对象、计时器 455 和订阅移除模块 460 可以用用户代码 451 实现,这与以系统代码实现的实例化模块 440 和事件订阅表不同。然而,要注意,计算机体系结构 400 的每个部分可在基本上任何类型的硬件和 / 或软件环境中实现。

[0049] 在某些实施例中,可实现参考计数来代替弱参考。用于事件处理的此技术由维护每个事件源运行时可调用包装器的参考计数的运行时实现。在这些情况下,可实现订阅者和该订阅者所订阅的事件之间的强参考。只要在事件订阅者被添加到事件源时,该参考计数被递增,而移除订阅者使该参考计数递减。当确定该参考计数为非 0 时,对事件源的强参考被自动维护。该强参考防止事件源被垃圾收集。当确定参考计数为 0 时,该运行时释放其对该事件源的强参考。这允许该事件源在不再被其相关联的运行中的程序需要时够格被

垃圾收集。

[0050] 相应地,描述了提供接口实例上的类型级构造模拟的方法、系统和计算机程序产品。此外,提供了维护基于令牌的事件处理系统的方法、系统和计算机程序产品。

[0051] 本发明可具体化为其它具体形式而不背离其精神或本质特征。所描述的实施例在所有方面都应被认为仅是说明性而非限制性的。因此,本发明的范围由所附权利要求书而非前述描述指示。落入权利要求书的等效方案的含义和范围内的所有改变被权利要求书的范围所涵盖。

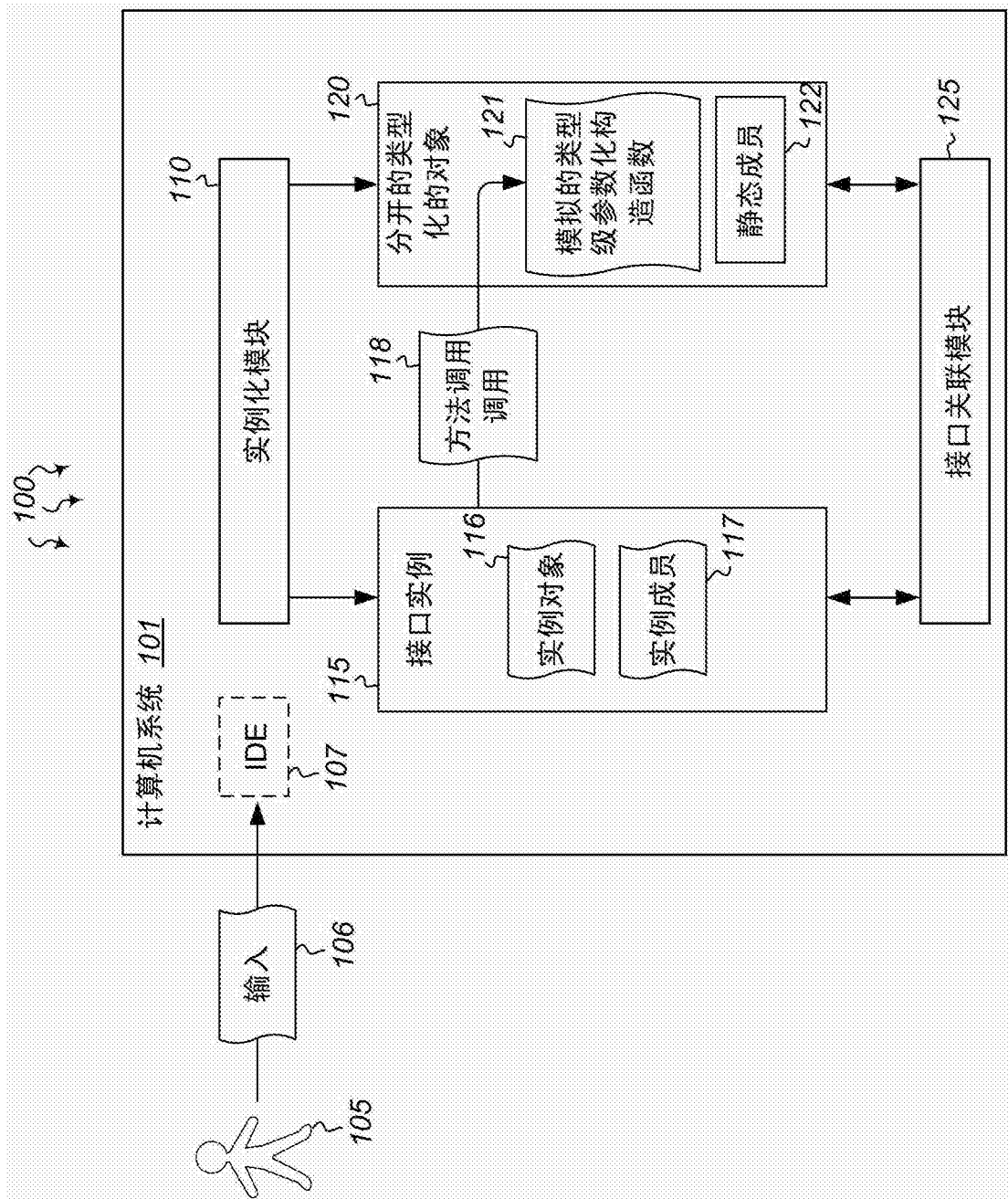


图 1

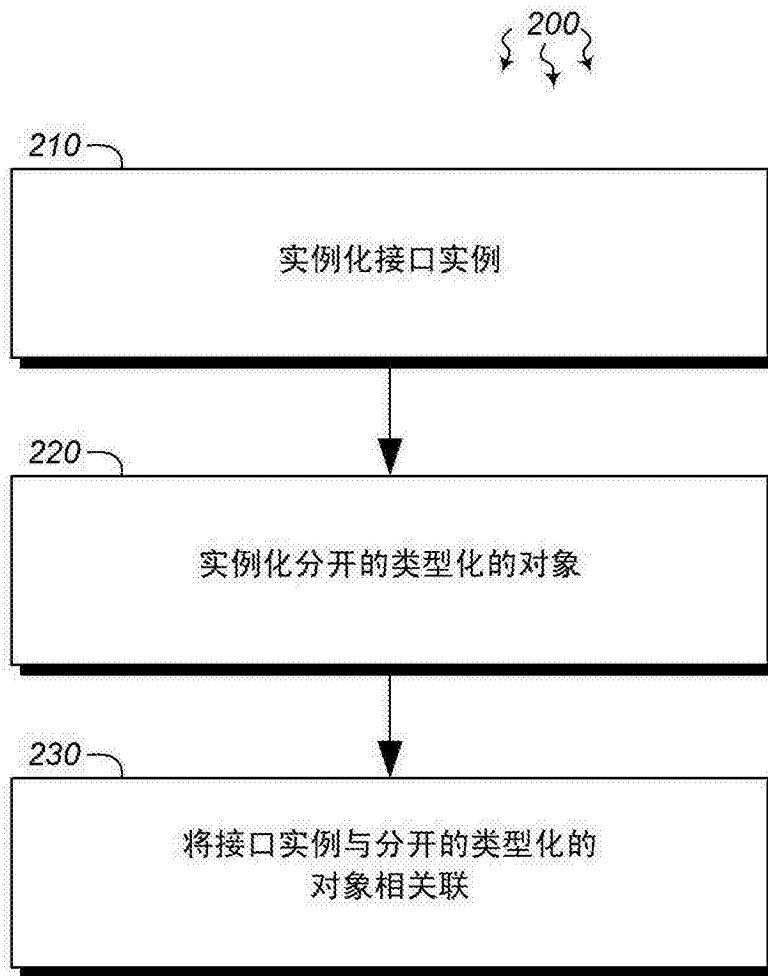


图 2

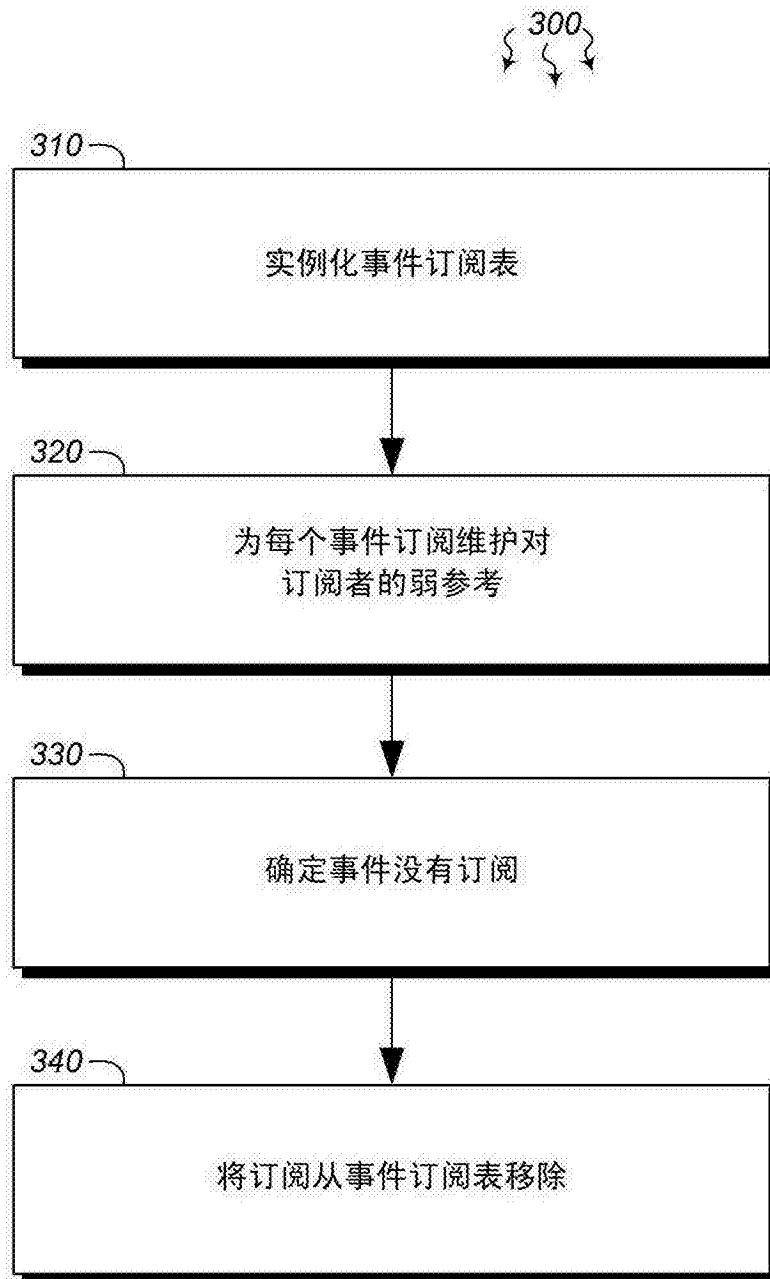


图 3

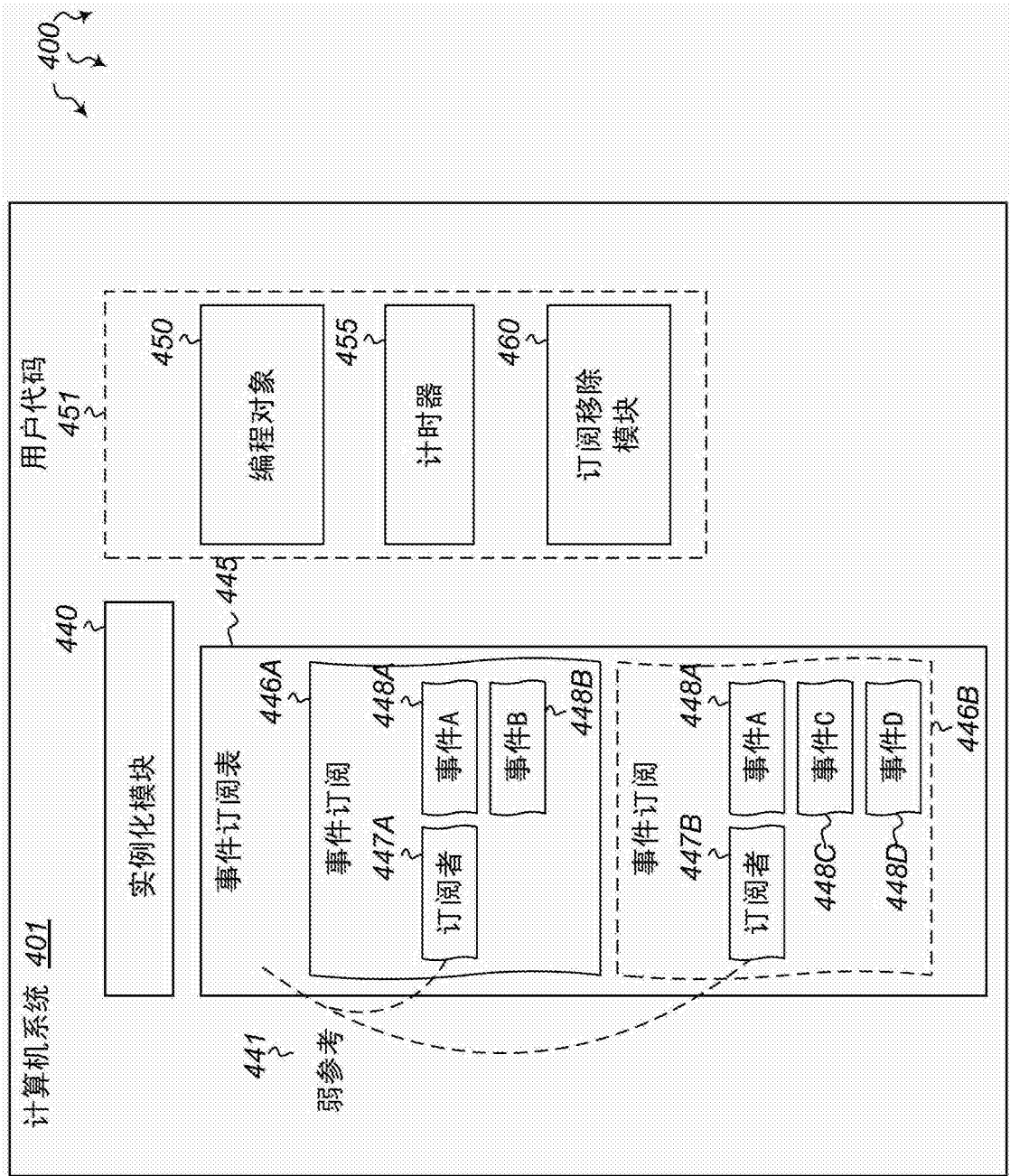


图 4