



(12) 发明专利申请

(10) 申请公布号 CN 101939723 A

(43) 申请公布日 2011. 01. 05

(21) 申请号 200980104571. 7

代理人 胡利鸣 钱静芳

(22) 申请日 2009. 01. 06

(51) Int. Cl.

(30) 优先权数据

G06F 9/06 (2006. 01)

12/027, 990 2008. 02. 07 US

G06F 9/44 (2006. 01)

(85) PCT申请进入国家阶段日

G06F 15/173 (2006. 01)

2010. 08. 06

(86) PCT申请的申请数据

PCT/US2009/030211 2009. 01. 06

(87) PCT申请的公布数据

W02009/099691 EN 2009. 08. 13

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 R·S·金伯利 C·L·安德森

C·A·西泽伯斯基 A·J·摩尔

D·F·伯克斯

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

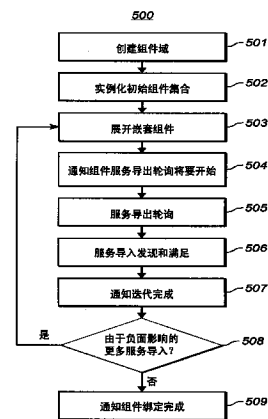
权利要求书 3 页 说明书 10 页 附图 6 页

(54) 发明名称

迭代组件绑定

(57) 摘要

组件域用于定义与该组件域相关联的各组件之间的绑定。组件域跟踪每一组件要导入的各种服务，并且跟踪每一组件导出的各种服务。在运行时而非在编译时间，使用服务导入和导出数据来绑定各组件之间的各种相互关系。因此，取决于可用组件及其特征，绑定可以是不同的。这准许应用程序更加灵活。在一个实现中，绑定是迭代地执行的。例如，如果在将服务导入绑定到服务导出的一次迭代之后，一些组件可能展开它们导出的服务或它们希望导入的服务。



1. 一种包括其上含有计算机可执行指令的一个或多个计算机可读介质 (104) 的计算机程序产品,当所述指令由计算系统 (100) 的一个或多个处理器 (102) 执行时,使所述计算系统执行一种用于合成程序的多个组件的方法,所述方法包括:

创建组件域 (112、210) 的动作 (501);

实例化一个或多个组件 (211) 的初始集合的动作 (502、601),以使得它们与所述组件域相关联,以及以便使所述初始集合中的每一组件所导入的服务 (A、B、C、D、E、F) (如果有的话) 对所述组件域可见,并且以便使所述初始集合中的每一组件所导出的服务 (如果有的话) 也对所述组件域可见;

所述组件域标识一个或多个组件的所述初始集合中的包括嵌套组件 (401、402、411、412) 的任一组件的动作 (602)

对于所述组件的初始集合中的包括一个或多个嵌套组件的每一组件,展开所述包括一个或多个嵌套组件的组件的动作 (503、603),以便所述嵌套组件也在以下程度上对所述组件域可见:使所述嵌套组件所导入的服务 (如果有的话) 对所述组件域可见,以及使所述嵌套组件所导出的服务 (如果有的话) 对所述组件域可见;

所述组件域轮询所述组件的初始集合以寻找所导出的服务的动作 (505); 以及

所述组件域轮询所述组件的初始集合以寻找所导入的服务并使用所述组件的初始集合中的一个组件所导出的服务之一来满足所述组件的初始集合的各组件中的至少另一个组件所导入的服务中的至少一个的动作 (506),其中对包括一个或多个嵌套组件的每一组件的所述展开动作、所述组件域轮询所导出的服务的动作、以及所述组件域轮询所导入的服务并满足所导入的服务中的至少一个的动作是按预定次序发生的。

2. 如权利要求 1 所述的计算机程序产品,其特征在于,所述预定次序是首先对包括一个或多个嵌套组件的每一组件的展开动作,其次所述组件域轮询所导出的服务的动作,以及第三所述组件域轮询所导入的服务并满足所导入的服务中的至少一个的动作。

3. 如权利要求 2 所述的计算机程序产品,其特征在于,所述方法还包括:

在轮询所导出的服务的动作之前,通知所述组件的初始集合中的至少一些组件将要开始轮询所导出的服务的动作。

4. 如权利要求 3 所述的计算机程序产品,其特征在于,所述方法还包括:

在轮询所导入的服务并满足所导入的服务中的至少一个的动作之后,以及向所述组件的初始集合中的至少一些组件通知轮询所导入的服务并满足所导入的服务中的至少一个现在已经完成的动作。

5. 如权利要求 2 所述的计算机程序产品,其特征在于,所述方法还包括:

在轮询所导入的服务并满足所导入的服务中的至少一个的动作之后,向所述组件的初始集合中的至少一些组件通知轮询所导入的服务并满足所导入的服务中的至少一个现在已经完成的动作。

6. 如权利要求 2 所述的计算机程序产品,其特征在于,包括对包含一个或多个嵌套组件的每一组件的展开动作、所述组件域轮询所导出的服务的动作、以及第三所述组件域轮询所导入的服务并满足所导入的服务中的至少一个的动作的集体动作是在轮询的该集体动作的初始迭代中执行的,所述方法还包括:

在所述集体动作的初始迭代中轮询所导入的服务并满足所导入的服务中的至少一个

的动作之后,确定是否存在作为满足所述服务导入中的至少一个的动的结果而添加的其他服务导入的动作;以及

执行所述集体动作的后续迭代的动作。

7. 如权利要求 6 所述的计算机程序产品,其特征在于,所述集体动作是针对多次迭代来迭代地执行的,直至满足了所有服务导入为止。

8. 如权利要求 6 所述的计算机程序产品,其特征在于,所述附加服务导入是作为与所述组件域相关联的用于满足所述组件的初始集合的服务导入的附加组件的结果而添加的,附加服务导入是所述附加组件中的至少一个的服务导入。

9. 如权利要求 6 所述的计算机程序产品,其特征在于,所述附加服务导入是由所述组件的初始集合之一在满足了其服务导入中的另一个之后添加的。

10. 如权利要求 1 所述的计算机程序产品,其特征在于,所述一个或多个计算机可读介质是物理存储器和 / 或存储介质。

11. 一种包括其上含有计算机可执行指令的一个或多个计算机可读介质 (104) 的计算机程序产品,当所述指令由计算系统 (100) 的一个或多个处理器 (102) 执行时,使所述计算系统执行一种用于合成程序的多个组件 (211-216) 的方法 (500),所述方法包括:

创建组件域 (112、210) 的动作 (501),其中所述组件域能够标识在该组件域内实例化的每一组件所导出的服务 (A、B、C、D、E、F) 和所导入的服务;

在所述组件域中实例化组件的初始集合的动作 (502) 以及

针对至少两次迭代来迭代地执行以下动作的动作:

执行所述组件的初始集合中的每一组件的嵌套组件 (401、402、411、412) 展开的动作 (503、603),其中所述嵌套组件展开允许所述组件域标识所述嵌套组件所导出和导入的服务;

所述组件域标识所述组件的初始集合中的包括所述嵌套组件的每一组件所导出的服务 (如果有的话) 的动作 (505)

在标识了所导出的服务后,所述组件域标识所述组件的初始集合中的每一组件所导入的服务 (如果有的话) 的动作 (506)

满足所导入的服务中的至少一个的动作 (506) 以及

确定在所述满足动作之后是否还存在尚待满足的其他所导入的服务的动作 (508),并且如果是,则用相对于所述组件域中的组件的可能展开的集合的下一代来继续所述迭代动作,并且如果否,则结束所述迭代动作。

12. 如权利要求 11 所述的计算机程序产品,其特征在于,所述一个或多个计算机可读介质是物理存储器和 / 或存储介质。

13. 如权利要求 12 所述的计算机程序产品,其特征在于,对于所述迭代中的至少一次,满足所导入的服务中的至少一个的动作包括:

用所述组件域中的另一组件所导出的服务来满足所导入的服务中的至少一个的动作。

14. 如权利要求 12 所述的计算机程序产品,其特征在于,对于所述迭代中的至少一次,满足所导入的服务中的至少一个的动作包括:

在所述组件域中搜索所述实例化的组件的动作;

确定所述组件域中的实例化的组件都没有导出所述服务的动作;

找出尚未在所述组件域内实例化的导出所述服务的组件的动作；
在所述组件域中实例化所找到的组件的动作；以及
将所找到的组件的实例化所导出的服务与已经在所述组件域内实例化的组件所导入的服务相绑定的动作。

15. 如权利要求 14 所述的计算机程序产品,其特征在于,所述寻找动作包括:
咨询列出组件及其所导出的服务的目录的动作。

16. 如权利要求 11 所述的计算机程序产品,其特征在于,所展开的嵌套组件包括至少两级嵌套组件。

17. 如权利要求 11 所述的计算机程序产品,其特征在于,还包括:
对于迭代中的至少一次,在标识所述组件域中的组件所导出的服务之前并在满足所导出的服务中的至少一个的动作之后通知所述组件的动作。

18. 如权利要求 11 所述的计算机程序产品,其特征在于,还包括:
在迭代的动作完成之后通知所述组件域中的组件的动作。

19. 一种包括其上含有计算机可执行指令的一个或多个计算机可读介质 (104) 的计算机程序产品,当所述指令由计算系统 (100) 的一个或多个处理器 (102) 执行时,使所述计算系统执行一种用于合成程序的多个组件 (211-216) 的方法 (500),所述方法包括:

创建组件域 (112、21) 的动作 (501),其中所述组件域能够标识在该组件域内实例化的每一组件所导出的服务 (A、B、C、D、E、F) 和所导入的服务;

在所述组件域中实例化组件的初始集合的动作 (502);

针对至少两次迭代来迭代地执行以下动作:

执行所述组件的初始集合中的每一组件的嵌套组件 (401、402、411、412) 展开的动作 (503、603),其中所述嵌套组件展开允许所述组件域标识所述嵌套组件所导出和导入的服务;

在标识所述组件域中的组件所导出的服务 (505) 之前通知所述组件的动作 (504),

所述组件域标识所述组件的初始集合中的包括所述嵌套组件的每一组件所导出的服务 (如果有的话) 的动作 (505);

在标识了所导出的服务 (505) 后,所述组件域标识所述组件的初始集合中的每一组件所导入的服务 (如果有的话) 的动作 (506);

满足所导入的服务中的至少一个的动作 (506);以及

通知所述组件域中的组件所述满足动作已完成的动作 (507);

确定在所述满足动作之后是否还存在尚待满足的其他所导入的服务的动作 (508),并且如果是,则用相对于所述组件域中的组件的可能展开的集合的下一迭代来继续所述迭代动作,并且如果否,则结束所述迭代动作;以及

通知所述组件域中的组件所述迭代动作已完成的动作 (509)。

20. 如权利要求 19 所述的计算机程序产品,其特征在于,所述一个或多个计算机可读介质是物理存储器和 / 或存储介质。

迭代组件绑定

[0001] 背景

[0002] 面向对象程序设计将软件构造成对象实例或“组件”的集合，每一个都在定义该组件的方法和属性的类之后模式化。当谈及程序设计时，“组件化”指的是对组件进行定义以便使它们对程序可用。当然，程序的各组件在功能上互连以便实现该程序的更大目的。“合成”指的是对将程序中的对象如何相关在一起进行定义。

[0003] 通常，对象之间的相互关系是使用非常明确地定义程序的各组件之间的关系的代码或声明来实现的。已经使用编码对象之间的相互连接的这种显式合成来在面向对象程序设计领域作出了巨大进步。在常规显式合成中，例如，定义程序的合成（即，各组件之间的相互关系）的代码是在使用源代码创作程序之后编译的。可执行文件因而从其诞生开始就表示程序的各组件和各程序组件之间的各种相互关系。

[0004] 简要概述

[0005] 本文描述的各实施例涉及使用组件域来定义与该组件域相关联的各组件之间的绑定。组件域跟踪每一组件要导入的各种服务，并且跟踪每一组件要导出的各种服务。在运行时而非在编译时间，使用服务导入和导出数据来绑定各组件之间的各种相互关系。因此，取决于可用组件及其特征，绑定可以是不同的。这准许应用程序更加灵活。在一个实施例中，绑定是迭代地执行的。例如，如果在将服务导入绑定到服务导出的一次迭代之后，一些组件可能展开它们导出的服务或它们希望导入的服务。

[0006] 本概述不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。

[0007] 附图简述

[0008] 为了描述可获得本发明的上述和其他优点和特征的方式，将通过引用附图中示出的各具体实施例来呈现以上简要描述的本发明的更具体描述。可以理解，这些附图仅描绘更宽泛原理的各示例实施例，并且因此不被认为是对本发明的范围的限制，将通过使用附图用附加特征和细节来描述和解释这些实施例，附图中：

[0009] 图 1 示出可操作组件域的示例计算系统；

[0010] 图 2 示组件域以及其相关联的逻辑和该组件域中包含的各组件的示例体系结构；

[0011] 图 3 示出组件域中在被绑定后的各组件的示例；

[0012] 图 4 示出与嵌套组件展开相关联的过程流程；

[0013] 图 5 示出用于迭代地绑定组件域中的组件的方法的流程图；

[0014] 图 6 示出用于在实例化具有嵌套组件的组件后展开嵌套组件的方法的流程图；以及

[0015] 图 7 示出用于用服务导出来满足服务导入的方法的流程图。

[0016] 详细描述

[0017] 根据本文描述的各实施例，组件域被用来创建该组件域中的各组件之间的绑定。组件域跟踪每一组件要导入的各种服务，并且跟踪每一组件要导出的各种服务。在运行时而非在编译时间，使用服务导入和导出数据来绑定各组件之间的各种相互关系。因此，取决

于可用组件及其特征,绑定可以是不同的。这准许应用程序更加灵活。在一个实现中,绑定是迭代地执行的。例如,如果在将服务导入绑定到服务导出的一次迭代之后,一些组件可能展开它们导出的服务或它们希望导入的服务。

[0018] 首先,在概括地关于计算系统的一些介绍性讨论之后,将参考图 1 描述基本计算系统。随后,将参考图 2 到图 7 来示出并描述具有组件域逻辑的组件域的各实施例和组件域的操作的各实施例。

[0019] 计算系统现在越来越多地采用各种各样的形式。例如,计算系统可以是手持式设备、电器、膝上型计算机、台式计算机、大型机、分布式计算系统或甚至常规上不被认为是计算系统的设备。在本说明书以及权利要求书中,术语“计算系统”被广义地定义为包括包含至少一个处理器及其上能含有可由处理器执行的计算机可执行指令的存储器的任何设备或系统(或其组合)。存储器可采取任何形式,且可取决于计算系统的性质和形式。计算系统可分布在网络环境上,且可包括多个组成计算系统。

[0020] 参考图 1,在其最基本的配置中,计算系统 100 一般包括至少一个处理单元 102 和存储器 104。存储器 104 可以是物理系统存储器,它可以是易失性的、非易失性的或两者的某种组合。术语“存储器”也可在此处用来指示诸如物理存储介质等非易失性大容量存储。如果计算系统是分布式的,则处理、存储器、和/或存储能力也可以是分布式的。如此处所使用的,术语“模块”或“组件”可以指在计算系统上执行的软件对象或例程。此处描述的不同组件、模块、引擎和服务可被实现为在计算系统上执行的对象或进程(例如,作为分开的线程)。

[0021] 在以下描述中,参考由一个或多个计算系统执行的动作描述各实施例。如果这样的动作由软件实现,则相关联计算系统中执行该动作的一个或多个处理器响应于执行了计算可执行指令而引导计算系统的操作。这样的操作的示例涉及对数据的操纵。计算机可执行指令(和所操纵的数据)可被存储在计算系统 100 的存储器 104 中。

[0022] 计算系统 100 还可包含允许计算系统 100 例如通过网络 110 与其他计算系统通信的通信信道 108。通信信道 108 是通信介质的示例。通信介质一般用诸如载波或其他传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其他数据,并且包括任何信息传递介质。作为示例而非限制,通信介质包括有线介质,诸如有线网络或直接线连接,以及无线介质,诸如声学、无线电、红外线和其他无线介质。如此处所用的术语“计算机可读介质”既包括存储介质又包括通信介质。

[0023] 在图 1 中,计算系统 100 被示为在存储器 104 的非易失性部分内具有组件域 112。如将参考图 2 以及后续附图进一步详细描述,组件域 112 被用作在其中实例化各种组件的逻辑容器。组件域逻辑 114 随后动态地绑定各组件,而非在编译时间绑定这些组件。

[0024] 本发明的范围内的实施例还包括用于携带或其上储存有计算机可执行指令或数据结构的计算机可读介质。这样的计算机可读介质可以是可由通用或专用计算机访问的任何可用介质。作为示例而非限制,这样的计算机可读介质可包括物理存储和/或存储器介质,诸如 RAM、ROM、EEPROM、CD-ROM 或其他光盘存储、磁盘存储或其他磁存储设备、或可用于承载或存储计算机可执行指令或数据结构形式的所需程序代码装置且可由通用或专用计算机访问的任何其他介质。当信息通过网络或另一通信连接(硬连线、无线或硬连线或无线的组合)传输或提供给计算机时,计算机将该连接完全视为计算机可读介质。因此,任何

这样的连接被适当地称为计算机可读介质。上述的组合也应被包括在计算机可读介质的范围内。

[0025] 计算机可执行指令例如包括,使通用计算机、专用计算机、或专用处理设备执行某一功能或某组功能的指令和数据。尽管用对结构特征和/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在此所述的具体特征或动作。相反,在此所述的具体特征和动作是作为实现权利要求的示例形式公开的。

[0026] 图 2 示出组件域 210 和被标记为“组件域逻辑”220 的相关联的组件域管理器。图 2 的组件域 210 可以是例如图 1 是组件域 112。组件域 210 可以是可包含组件实例的任何逻辑容器。组件域 210 的灵活性允许该组件域包含各个类的各种组件实例。组件域 210 中包括的组件的类型将取决于应用程序和该应用程序所使用的构件块。因此,本发明的原理不限于组件域 210 内包括的组件的类型,也不限于该组件域内的组件的数量。然而,出于说明的目的,六个组件 211、212、213、214、215 和 216 被示为包括在组件域 210 内。

[0027] 组件域 210 可以是能够在字面上或逻辑上包含其他组件的任何逻辑容器。例如,组件域 210 可以是能够接纳各组件实例的容器类的实例,而各组件实例的集合是一数据结构。组件域 210 还可以是不直接包含每一组件实例但可能具有与相关联的组件的有关的信息的对象。即使在这种情况下,这些组件也可被认为是逻辑上处于组件域 210 内。例如,在一个实施例中,组件域只包括一组值,可能是一组名-值对。组件域从导出服务的组件接收值,并且导入该服务的组件接收该值。然而,即使在该上下文中,这些组件也可被认为是逻辑上处于组件域内部或之内。在本说明书和权利要求书中,当组件被称为处于组件域“之内”或“内部”时,其意味着该组件被包含在该组件域内或以组件域逻辑 220 能够与该组件域通信并绑定与该组件域相关联的各组件的方式在逻辑上与该组件域相关联。

[0028] 组件域逻辑 220 被示为包括各种机制,包括例如关联机制 222、服务清单机制 223、嵌套组件展开机制 224、具有相关联的匹配规则 226 的匹配机制 225、抽象机制 227、以及具有相关联的目录 229 的编目录机制 228。这些各种机制 222 到 229 被示为分开的机制。然而,这是出于清楚地解释组件域逻辑 220 的功能的目的的。这不是用来表示它们实际上是分开的机制。这些机制中的一些可以组合成单个机制。事实上,组件域逻辑的全部可以是整体式的。在另一个极端,每一机制可以包括多个组件。单个组件还可包括对多个机制都有贡献的逻辑。将简短描述各组件域逻辑 220 机制的操作。

[0029] 然而,首先,这一描述回头参考在组件域 210 内实例化的各组成组件。在所示示例中,各组成组件包括组件 211 到 216。每一组件都是特定类的实例。为了象征性地示出这一原理,每一组件都被示为特定形状。例如,在该示例中,组件 211 被示为正方形。组件 212 和 214 被示为三角形,从而指示它们是同一个类的各自实例,该类不是与组件 211 相同的类。组件 213 被示为圆形,象征它是与组件 211 和 212 不同的类的实例。组件 215 被示为上底长的梯形,象征它是与组件 211 到 214 不同的类的实例。最后,组件 216 被示为下底长的梯形,象征它是与组件 211 到 215 不同的类的实例。

[0030] 每一组件包括可能的一个或多个服务导入以及可能一个或多个服务导出。相对于一组件的“服务导入”是该组件正在请求服务的指示。相对于一组件的“服务导出”是该组件正在提议来提供服务的指示。服务导入和服务导出在图 2 中各自由从对应组件伸出的臂来表示。服务导入由在其端点处具有杯形的臂来表示。服务导出由在其端点处具有球形的

臂来表示。在图 2 的配置中,所有组件 211-216 都是在组件域 210 内实例化的。然而,这些组件都未连接在一起。此外,各组件连接的方式没有提前于运行时来显式地指定。相反,指定了服务导入和导出。在图 2 中,使用从字母 A 到 F 来为各服务抽象地表示各服务的导入和导出。

[0031] 例如,组件 211 具有由臂 a 表示的用于服务 B 的一个服务导出;以及两个服务导入,一个用于服务 A,由臂 i 表示,并且另一个用于服务 A,由臂 j 表示。组件 212 具有两个服务导出,一个用于服务 A,由臂 b 表示,并且一个用于服务 E,由臂 c 表示。组件 212 没有服务导入。组件 213 具有由臂 d 表示的用于服务 F 的一个服务导出;以及两个服务导入,一个用于服务 B,由臂 k 表示,并且另一个用于服务 C,由臂 m 表示。组件 214 与组件 212 是相同的类。在一个实施例中,服务导入和服务导出的类型由类来定义。因此,组件 214 与组件 212 一样,因为它没有服务导出并且具有两个服务导出,一个用于服务 A,由臂 f 表示,并且一个用于服务 E,由臂 e 表示。组件 215 没有服务导入和一个服务导入,其用于服务 D,由臂 g 表示。组件 216 具有由臂 n 表示的用于服务 D 的一个服务导入和由臂 h 表示的用于服务 C 的一个服务导出。

[0032] 组件域 210 示出在合成各组件 211 到 216 之前的这些组件。“合成”指的是各组件将被如何互连的定义。最终,组件域 210 中的各组件将定义其互连。然而,这不是在运行组件域 210 中的各组件所表示的应用程序之前完成的。相反,组件域逻辑 220 将在运行时定义各组件之间的连接。

[0033] 图 3 示出在合成后的各组件 211 到 216。在一个特定匹配操作中,特定服务的服务导出要与同一特定服务的服务导入相耦合。在图 3 中,组件 212 的服务 A 的服务导出连接到组件 211 的服务 A 的服务导入(如通过臂 b 与臂 j 相聚所表示的)。这表示组件 212 使用相关接口将服务 A 提供给组件 211。同一象征适用于图 3 相对于其相应服务所示出的服务导入和导出之间的其他连接。例如,以下臂对被示为连接的,服务 A 的臂 f 和 i、服务 B 的臂 a 和 k、服务 C 的臂 h 和 m、以及服务 D 的臂 g 和 n。组件 212 的服务 E 的服务导出(由臂 c 表示)未连接到任何服务导入,从而表示该服务导出是潜在的。同一象征适用于臂 d 和 e。

[0034] 虽然各组件之间的连接没有提前定义,但指示了各组件的服务导出和导入。服务导出和导入在何处或如何指定是无紧要的。然而,一个可能的位置可以在类定义本身中。例如,考虑以下用 C# 代码编写的被称为“Consumer(消费者)”的类的类定义:

```
[0035] class Consumer
[0036] {
[0037]     [Import(“Consumer Report”)]
[0038]     IReport MyReport {get ;set ;}
[0039]     •
[0040]     •
[0041]     •
[0042] }
```

[0043] 在方括号中,类 Consumer 声明其导入被称为“Consumer Report(消费者报告)”的服务。方括号的内容不影响 C# 中的编译对象的功能。相反,对该编译类而言,该内容被编写成伴随编译类型或成员的、以不作为编译类型或成员默认执行的一部分但可以从外部检

查并操作的方式对该编译类型或成员进行描述的元数据。在该示例中,在要实例化类的对象时,能以其预编译或后编译形式来引用类定义,以标识遵循该类的特定对象的服务导入。

[0044] 在服务导入与服务导出之间的一个示例匹配中,服务名要精确匹配,但也构想了其他匹配规则。以下表示用 C# 表示的类定义,其中该类导出被称为“Consumer Report”的服务:

```
[0045] class ReportEnvelope
[0046] {
[0047]     [Export(“Consumer Report”)]
[0048]     IReport Content {get ;set}
[0049]     •
[0050]     •
[0051]     •
[0052] }
```

[0053] 在此,括号中的元数据表达式声明类 ReportEnvelope(报告信封)导出“Consumer Report”服务。例如,在 ReportEnvelope 类的对象与 Consumer 类的对象相连接时,Consumer 对象可以从 ReportEnvelope 对象得到报告。

[0054] 在该示例中,不管类本身是已编译还是预编译的,都可以对其进行引用以确定特定对象的服务导出和服务导入。在另一实施例中,对象可被配置成提供组件域逻辑可以用来询问各对象以寻找服务导入和导出标识的标识。

[0055] 现在将详细描述组件域管理器 220 的各种机制 222-229。同样,这些机制可以按功能来划分,并且不表示组件域管理器 220 中的任何实际对象边界。如以下将提到的,各机制 222-229 中的一些可以偶尔使用通常由系统中提供的底层运行时所提供的实例化机制。该实例化机制具有使用类定义来实例化遵循该类的组件的能力。

[0056] 关联模块 222 能够将实例化的组件与组件域 210 进行关联。例如,在图 2 中,组件 211 到 216 被示为在组件域 210 内相关联,象征它们不仅仅是实例化,而是以它们与组件域 210 相关联的方式来实例化的。将在运行时对与组件域 210 相关联的组件进行特定绑定处理。各组件与组件域的关联可以在每一组件的实例化的时刻或在某一稍后时刻进行。

[0057] 服务清单机制 223 对与组件域相关联的组件所导出和导入的服务进行索引。服务清单机制 223 可以具有用于发现这些服务的机制。例如,在一个实施例中,在实例化特定类的组件时,服务清单机制 223 可以访问类定义来评估所导入或导出的服务。在另一实施例中,可以将各组件设计成对发现请求进行响应。例如,各组件可以提供用来将其导出和导入的服务通知给查询实体的应用程序接口。或者,该组件可被设计成在实例化后立即将其服务导入和导出写到特定位置。

[0058] 例如,参考图 2,服务清单机制可以在概念上维护下表 1:

[0059]

组件标识符	导出的服务	导入的服务
211	B	A, A

212	A, E	
213	F	B, C
214	A, E	
215	D	
216	C	D

[0060]

[0061] 表 1

[0062] 服务清单机制 223 不必将信息表示成表,但表是用于向读者示出可以维护的信息的类型的有用图示。此外,服务清单机制 223 可以不知道特定组件所导出或导入的服务的全部。

[0063] 嵌套组件展开机制 224 评估组件域中的一组件是否在其中嵌套了组件。例如,图 4 表示图 2 的组件之一(即,组件 215)在箭头 410 所表示的展开之前包括嵌套组件的分层结构。具体而言,图 4 中的组件 215 在展开之前包括两个子组件 401 和 402。这些子组件之一 401 在展开之前包括其自己的子组件 411 和 412。

[0064] 嵌套组件展开机制 224 被配置成发现嵌套组件,并将它们展开以便嵌套组件所导出和导入的服务也对绑定操作可见。例如,在展开 410 之后,示出了各嵌套组件 401、402、411 以及 412 而其导出的各个服务(在组件 411 的情况下是服务 A 和 E)被示为对绑定可见。另外,导入的各服务(组件 401 的服务 A、组件 402 的服务 B、以及组件 412 的服务 C)被示为对绑定可见。因此,在展开后,这些嵌套组件被服务清单机制 223 所识别。

[0065] 匹配机制 225 实际地对绑定这些组件的请求进行响应。为了做到这一点,匹配机制 225 遵循一组匹配规则 226。匹配机制 225 遵循匹配规则 226 以进行匹配并将服务导入绑定到服务导出。例如,在图 3 中,组件域 210 中的各个组件被示为绑定在一起。

[0066] 匹配规则可以是默认的一组隐式匹配规则,或可以存在超控默认匹配规则的其他匹配规则。例如,可能默认服务导入到服务导出的匹配是通过寻找精确的服务名匹配来进行的。例如,在上述类定义示例中,类“Consumer”具有被称为“Consumer Report”的服务导入。另一方面,类“ReportEnvelope”具有完全相同的名称“Consumer Report”的服务导出。因此,根据精确服务名匹配规则,ReportEnvelope 的实例的服务导出可以绑定到 Consumer 的实例的服务导入。

[0067] 然而,可以应用其他服务匹配规则。例如,在一些匹配规则中可以忽略大小写。在其他规则中,可能存在被认为彼此匹配的一群服务名。例如,可能存在意大利语的服务名、法语的服务名等,它们各自可被认为是匹配的。根据另一组匹配规则,如果服务导入或导出不是由名称来指定的,则匹配机制 224 实际上可以在结构上对照服务导出的成员名、类型和参数来评估服务导入的成员名、类型和参数。

[0068] 服务匹配机制 225 还可以采取步骤以不仅在逻辑上匹配各组件,还可至少对服务导入和导出的实际功能绑定有所贡献。在一个实施例中,一旦服务匹配机制 225 找到匹配,

则服务匹配机制 225 可以从服务导出组件获取所请求的服务,并将该服务提供给服务导入组件。

[0069] 在另一实施例中,服务匹配机制 225 可以向服务导入组件提供一句柄数据结构。服务导入机制随后可以使用该句柄来完成与服务导出机制的绑定。服务匹配机制可以任选地允许服务导入机制接受或拒绝该匹配。例如,服务导入组件可以通知服务匹配机制该匹配是否是可接受的,这之后将发生实际绑定。如果服务未被认为是可接受的,可能是服务匹配机制找到了备份匹配等,直至找到可接受匹配并被服务导入组件接受为止。

[0070] 例如,句柄数据结构可包括服务导入组件可以用来在其导入与匹配组件所提供的导出之间执行绑定操作的工厂。该句柄还可包括包含服务导入组件可以用来确定是否要触发工厂来执行绑定的信息的元数据(此后称为“选择元数据”)。在一个实施例中,服务导入组件可以不使用选择元数据,而可改为使用工厂来自动地接受绑定。然而,服务导入组件可以另选地使用选择元数据来评估所提议的导出对服务导入组件而言是否是可接受的。如果服务导入组件选择接受绑定,则服务导入组件可以使用工厂来触发该系统完成绑定。

[0071] 在其他情况下,可存在到特定服务导入的多个可能的服务导出匹配。在这种情况下,句柄可包括用于每一可能的服务导出的工厂和选择元数据。服务导入组件随后可以使用该元数据来不选择服务导出中的任何一个或可能选择服务导出中的一子集来进行导入。服务导入组件随后可以使用所选服务导出的对应工厂来完成与所选服务导出中的每一个的绑定。

[0072] 例如,如以上示例中,假定 Consumer 组件要导入名为“Consumer Report”的服务。如果存在提供名为“Consumer Report”的服务的多个组件,则可以对 Consumer 组件给予每一可能的服务导出匹配的工厂和选择元数据。选择元数据可包括例如 1) 生成该 Consumer Report 的日期,以及 2) 与 Consumer Report 有关的产品类的标识。如果 Consumer 组件只对名为“Consumer Report”的一个服务导出感兴趣,则 Consumer 组件可以使用选择元数据来标识最近生成的并且指示与 Consumer 组件最相关的产品类的 Consumer Report 服务导出。

[0073] 在一些情况下,组件域中可能不存在提供与特定服务导入相匹配的服务导出的组件。在这种情况下,绑定将仍然可能执行。现在将描述处理其中最初不能对特定服务导入执行绑定的情况的多种方式。

[0074] 在一种情况下,服务导入可具有相关联的“重要性”属性等,其指定对总体应用程序的功能而言满足该服务导入有多紧急。如果重要性高,则应当在某一位置找到服务导出。然而,如果重要性低,则可能可以在根本不绑定该服务的情况下通过该应用程序。可存在各种中间重要性级别,这可以管控绑定操作将在什么程度上尝试寻找服务导出。

[0075] 在一个实施例中,如果在组件域内没有找到特定服务导入的服务导出,抽象机制 227 可能从另一组件域中找出另一组件。因此,多个组件域可能在单个计算系统上运行,或可由该计算系统通过网络访问。在这种情况下,抽象机制 227 可以咨询那些其他组件域服务清单机制(或可能编目录机制 228),直至找到提供与要绑定的服务导入相对应的服务导出的组件。在一个实施例中,抽象机制 227 可以将本身模拟成提供服务导出的组件。服务导入因而可以绑定到抽象机制 227 本身。当服务导入组件对服务的抽象机制 227 进行调用时,抽象机制 227 可以与外部组件进行接口以执行所请求的服务。

[0076] 在一个实施例中,该组件域可以与多个相关组件域相关联。抽象机制 227 可以搜索所有这些相关组件域以寻找适当的服务导出。在其他实施例中,抽象机制 227 可以只搜索相关组件域中的一些。例如,如果组件域 210 是组件域的分层结构树中的节点,则抽象机制 227 可能只搜索表示祖先家系中的节点的组件域,而非搜索同辈组件域。抽象组件 227 因而可以对多个不同服务导出的服务导出进行抽象,同时执行与各外部组件进行接口以执行该服务所必需的底层处理。

[0077] 作为替换或补充,如果组件域内的已经实例化的组件不能满足服务导入,则编目录机制 228 可以使用目录 229 来找出当前未被实例化的提供特定服务导出的组件。例如,目录 229 可包含各个类定义,每一个类定义都具有描述该类的对象所导出的服务的元数据。基于对相关联的元数据的评估,编目录机制 228 可以找出提供所需服务导出的特定类定义。编目录机制 228 随后可以选择该类的要在组件域 210 内实例化的外部组件。或者,可以向导入该服务的组件提供描述被编目的类所导出的服务的元数据,以使得导入组件可以作出其自己的关于所导出的服务是否令人满意的决定。如果其是令人满意的,则导入该服务的组件可以使该类的一组件被实例化。例如,还可以向导入该服务的组件提供访问允许实例化导出该服务的组件的工厂的能力。

[0078] 在描述了图 2 的组件域逻辑 220 的各个功能之后,现在将参考图 5 到 7 来描述各个过程。

[0079] 图 5 示出了用于合成程序的各组件的方法 500 的流程图。在创建组件域(动作 501)后,实例化组件的初始集合并将其与该组件域相关联(动作 502)。例如,参考图 2,可以用相关联组件域逻辑 220 的一些或全部来创建组件域 210。组件域中的组件的初始集合可以像一个一样少,但可以多得多而没有限制。在图 2 的示例中,组件域中存在六个组件 211 到 216。然而,组件的初始集合中的组件的数量对本发明的更宽泛原理而言并不重要。组件的初始集合中的每一组件以及应在该组件域中实例化的任何后续组件都与该组件域相关联,以便使这些组件中的任一个所导入的服务(如果有的话)和所导出的服务(如果有的话)对该组件域可见。

[0080] 作为这一实例化过程的一部分,展开嵌套组件(动作 503)。图 6 示出了用于实例化组件的方法 600 的流程图。首先实例化组件本身(动作 601)。如果该组件不包含任何嵌套组件(判定框 602 中的“否”),则不进行展开。另一方面,如果存在嵌套组件(判定框 602 中的“是”),则展开这些嵌套组件(动作 603),以便这些嵌套组件也在以下程度上对该组件域可见:使该嵌套组件所导入的服务(如果有的话)对该组件域可见,以及使该嵌套组件所导出的服务(如果有的话)对该组件域可见。这一展开的示例参考图 4 来示出。这一展开可以在组件域中实例化具有一个或多个嵌套组件的组件的任何时候来执行。作为替换或补充,可以在绑定每次进行迭代时在预定时间展开这些组件。例如,在图 3 中,嵌套组件的展开在每一次迭代的开始时发生。

[0081] 返回图 5,一旦展开了所有组件,则组件域逻辑可任选地通知组件的初始集合的至少一些组件域逻辑将轮询组件以寻找所导出的服务(动作 504)。组件域可以通过采取其被编程来采取的任何动作来准备就其服务导出而被轮询来进行响应。例如,组件可以通过程序将另一组件实例化到组件域中。该组件能以其他方式将其状态置于适当的准备就绪以接收服务导出请求。

[0082] 接着,组件域逻辑轮询组件域中的每一组件以寻找该组件所提供的服务导出(动作 505)。从这一列表中,可以制定服务导出数据。接着,组件域逻辑可以轮询每一组件以寻找服务导入,并可以用服务导出列表来满足这些服务导入中的每一个(动作 506)。如以下参考图 7 进一步详细地解释的,如果未找到满足服务导入的服务导出,则组件域逻辑可以搜索目录来寻找提供该服务的类,并且随后在适当的情况下使该编目类的组件被实例化。

[0083] 如果存在可满足特定服务导入的多个服务导出,并且多个导出对该服务导入而言是可接受的,则使所有这些服务导出作为服务导出的集合来对该服务导入可用。还存在以下情况:存在可满足服务导入的比该服务导入所想要的更多的服务导出。例如,可能存在可满足指示了其只想要一个或不超过一个服务导出的服务导入的多个服务导出。或者,对指定其想要不超过 4 个服务导出的服务导入而言可能存在 10 个服务导出可用。在这种情况下,使用冲突解决来标识如何处理这些情况。

[0084] 一个可任选冲突解决步骤是在存在比服务导入所想要的更多的服务导出的情况下简单地使合成过程失败。在这种情况下,异常报告可以对关于该失败的原因给出一些指导以完成合成过程。另一选项是具有用于指定特定匹配的可接受性的规则,并且使更可接受的服务导出与服务导入相匹配,使得将适当数量的服务导出与该服务导入绑定。如果没有匹配的服务导出,则服务导入还可以指定空匹配是否是可接受的。这可取决于该服务的导入所启用的功能的重要性。

[0085] 例如,参考图 7,在搜索组件域内的组件(动作 701)时,确定该组件域内是否存在满足特定服务导入的服务导出(判定框 702)。如果找到了提供该服务导出的服务组件(判定框 702 找到“是”),则使用该组件域内找到的组件的服务导出来满足服务导入(动作 703)。

[0086] 另一方面,如果在组件域内未找到满足特定服务导入的服务组件(判定框 702 中的“否”),则在该组件域的外面找到导出该适当服务的组件(动作 704)。这可以通过例如咨询列出组件类及其导出的服务的目录来实现。

[0087] 取决于该组件是要在组件域内实例化(判定框 705 中的“实例化”),还是要由组件域逻辑进行抽象(判定框 705 中的“抽象”),可以随后采取不同的动作。

[0088] 如果要进行组件的实例化,则找到的组件要在组件域中实例化(动作 706),并且该新实例化的组件所导出的服务要与该组件域中预先存在的组件的服务导入进行绑定(动作 707)。如果要进行抽象,则组件域逻辑的抽象机制可以对找到的组件所提供的服务导出进行抽象(动作 708)。

[0089] 返回到图 5,一旦绑定的初始迭代完成,则通知各组件绑定迭代完成(动作 507)。在这一阶段,这些组件可以响应于初始绑定来使新服务导入或导出可见。

[0090] 在这一实施例中,首先执行嵌套组件的展开,接着进行服务导出的轮询,并且接下来进行服务导入的轮询和每一服务导入的满足。然而,只要这些步骤是以预定次序执行的,并且各组件了解该次序,则这些组件可以基于该组件处于该预定次序中的位置来采取适当的动作。例如,在以上示例中,各组件要在服务导出轮询之前首先展开嵌套组件。此外,服务要通过例如通过程序来实例化其他组件来处于使其自身准备好轮询的位置。最后,一旦初始绑定完成,各组件知道迭代可能重复,并且因此他们有机会展示新服务导入和导出以用于下一代。

[0091] 返回到图 5, 随后确定来自最后迭代的绑定操作是否造成将需要绑定操作的另一迭代的任何负面影响 (判定框 508)。例如, 在绑定操作中, 新组件可能已被实例化到组件域中。例如, 组件可能是在目录中找到的, 并且被实例化到组件域中。这些新实例化的组件自身可以具有服务导入。此外, 在组件绑定了服务时, 其可以发现它可以提供新服务导出, 并可能需要展示新服务导入。

[0092] 如果需要重新迭代 (判定框 508 中的“是”), 则组件域逻辑再一次展开任何新实例化的组件的任何嵌套组件 (动作 503) (如果在初始迭代期间未展开), 并随后通知每一组件将要开始服务导出轮询 (动作 504)。要执行重新迭代的这一判定可以在绑定了组件的初始集合之后很久来执行。例如, 程序的新版本可能变得可用, 从而对每一组件而言都需要重新绑定操作。

[0093] 如果不需要重新迭代 (判定框 508 中的“否”), 则通知这些组件绑定操作完成 (动作 509)。组件的这一合成是在运行时而非在编译时间执行的, 并且是基于服务导入和导出的松散标识的。这允许添加新组件, 并允许修改旧组件, 且在发货之后添加到该系统。例如, 在新组件变得对特定程序可用时, 可以更新目录。在程序接下来运行时, 可以诸如例如通过网络从程序供应商来从远程位置获得提供特定服务的组件。

[0094] 在一些情况下, 迭代可被终止。例如, 在尝试用导出来满足导入的过程中 (在动作 506 中), 可能发现一些导入不能使用当前可用组件集合来可接受地满足。在这种情况下, 该过程可以终止。计算系统可以自动地或可能在用户帮助下或可能遵循一些策略规则, 可随后通过从该组件域移除一个或多个组件和 / 或向该组件域添加一个或多个组件来改变组件集合。随后, 合成过程可以再次重新启动。在重新启动合成过程时, 如果一些导入已经被绑定到一些导出, 则重新绑定可以通过不尝试重新绑定到另一导出来可任选地尊重该绑定。或者, 合成过程可以不对现有绑定给予尊重的情况下重新开始。在这两个极端之间, 存在可以给予现有绑定的各种不同级别的尊重。

[0095] 本发明可被具体化为其他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。从而, 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应被权利要求书的范围所涵盖。

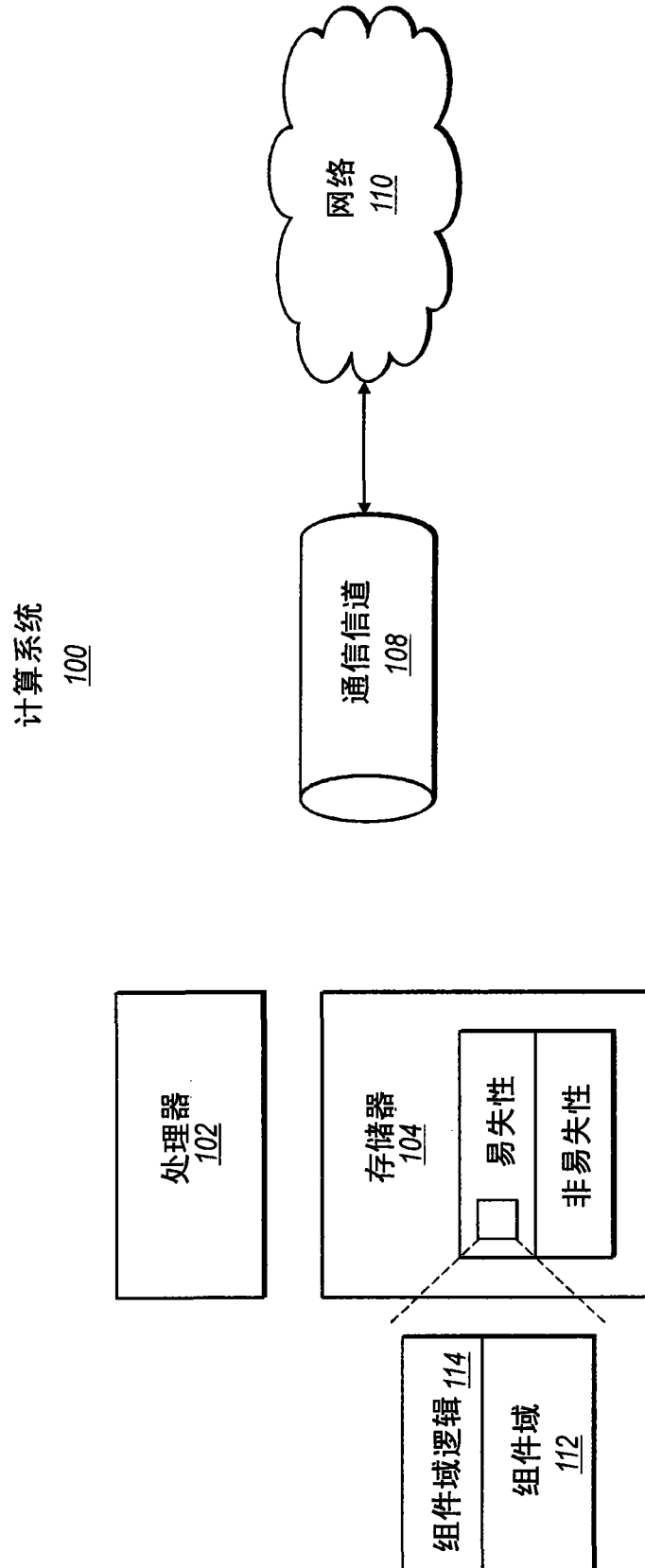


图 1

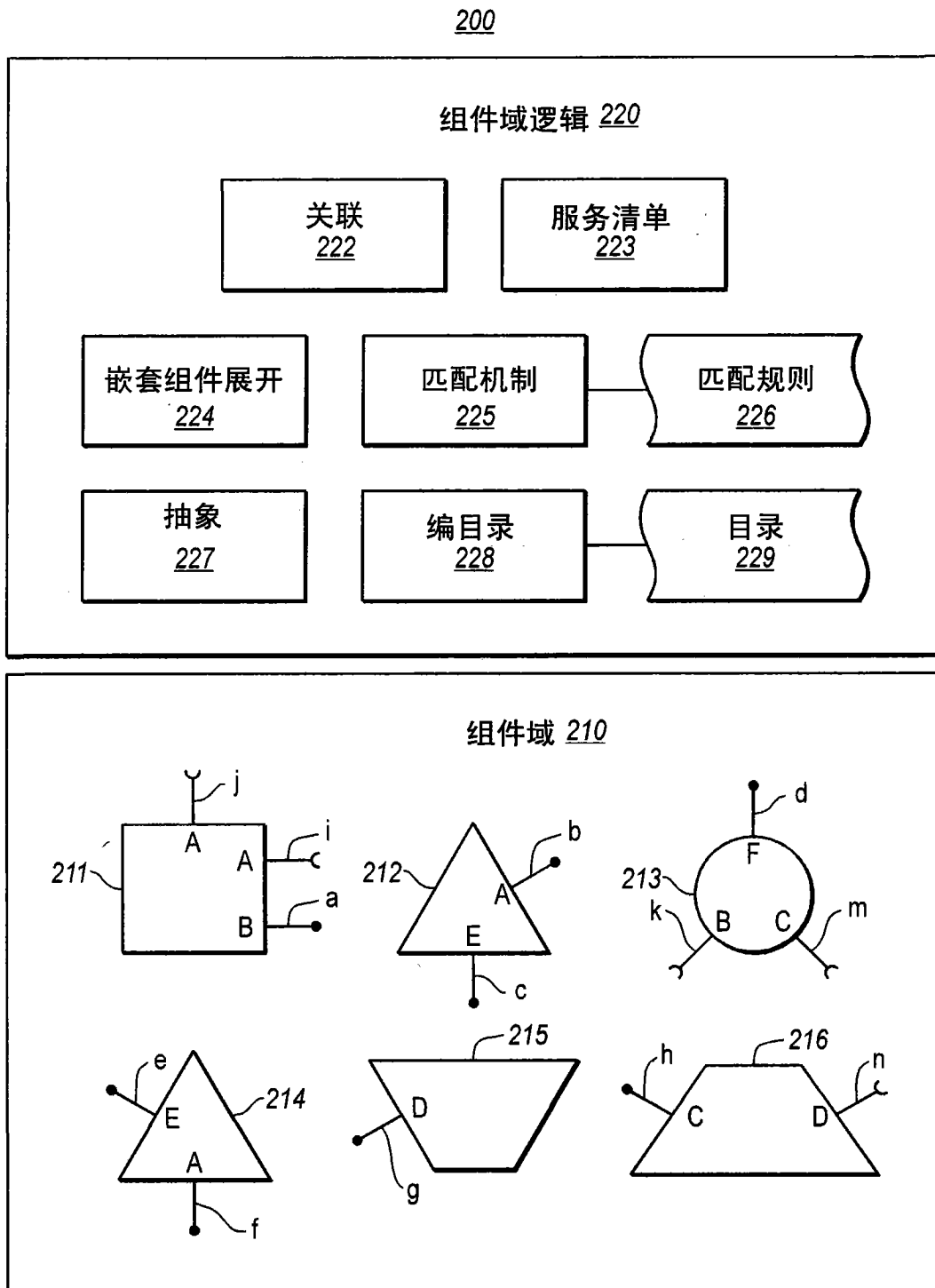


图 2

300

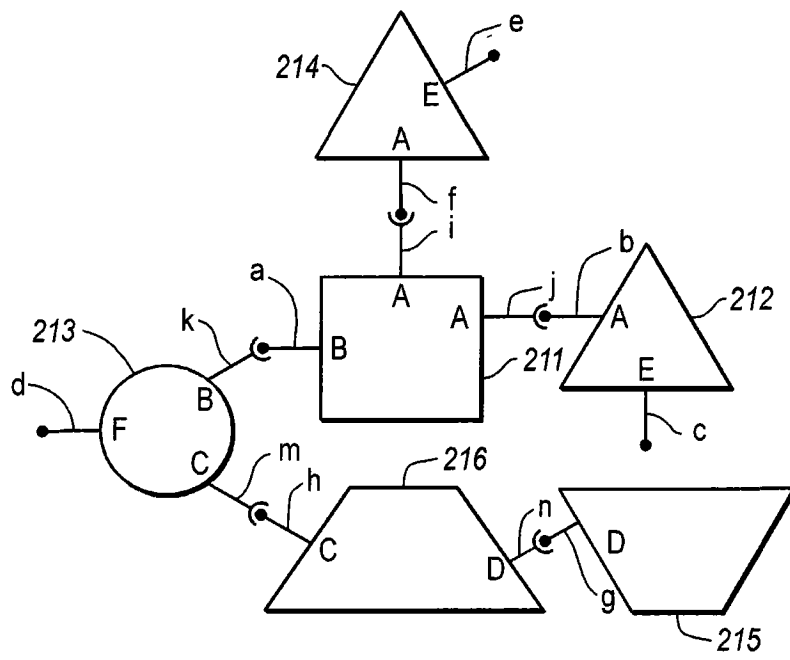


图 3

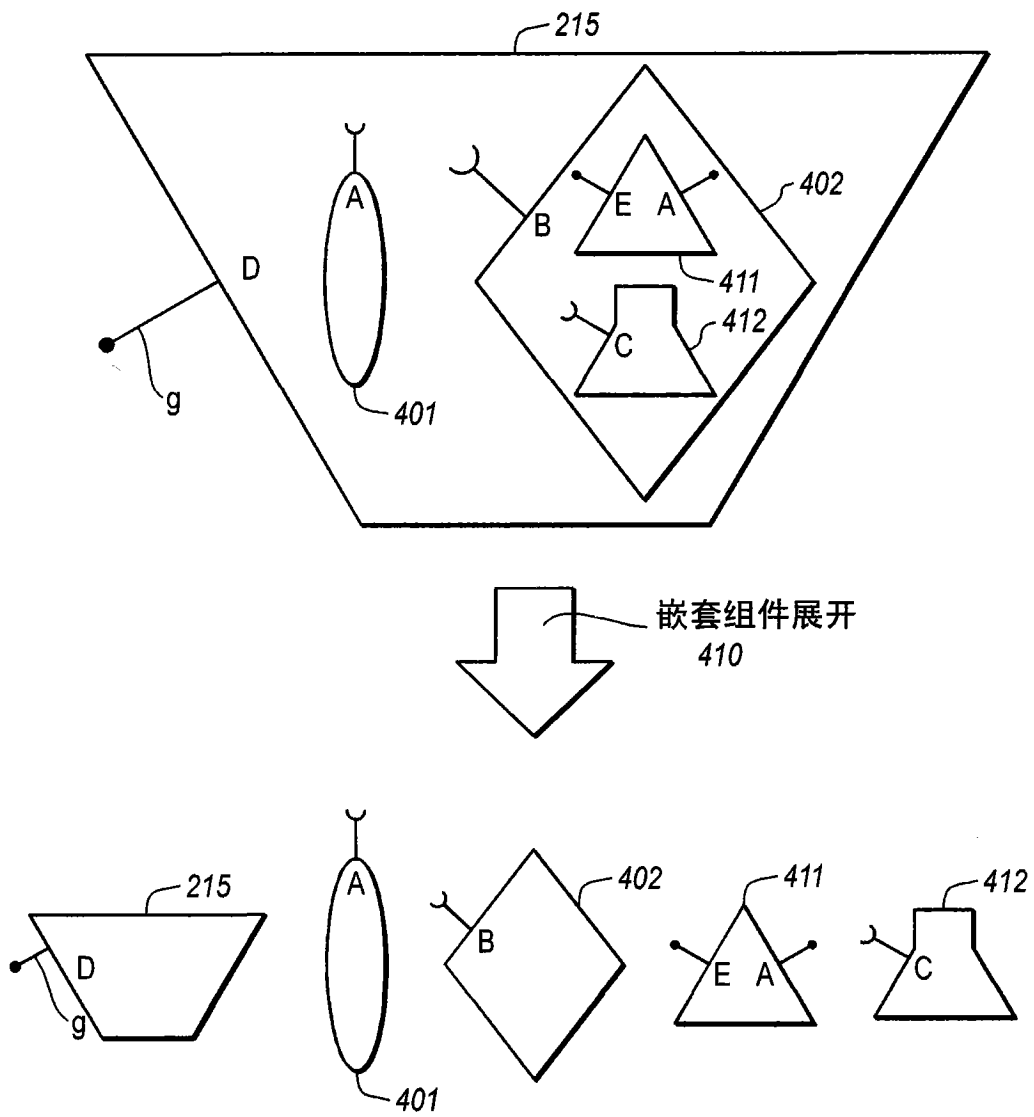


图 4

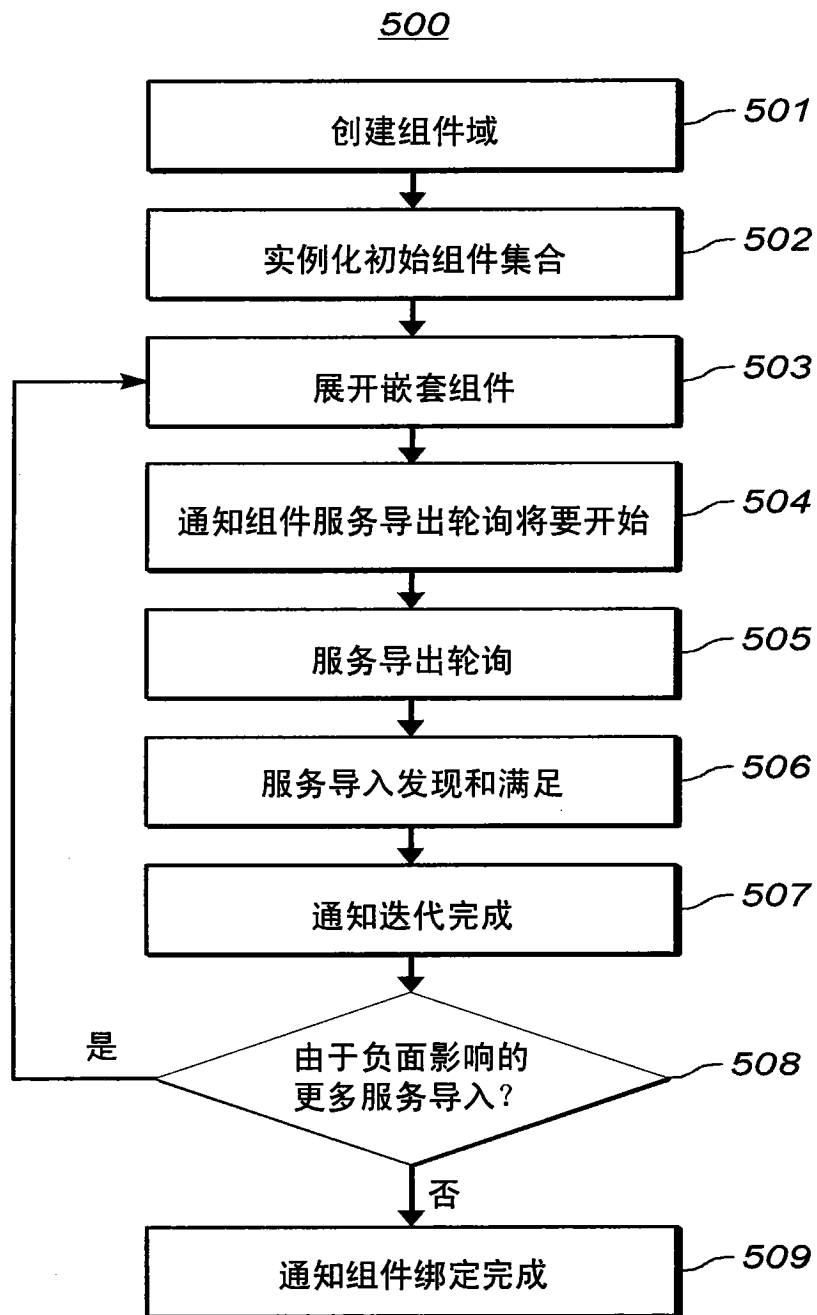


图 5

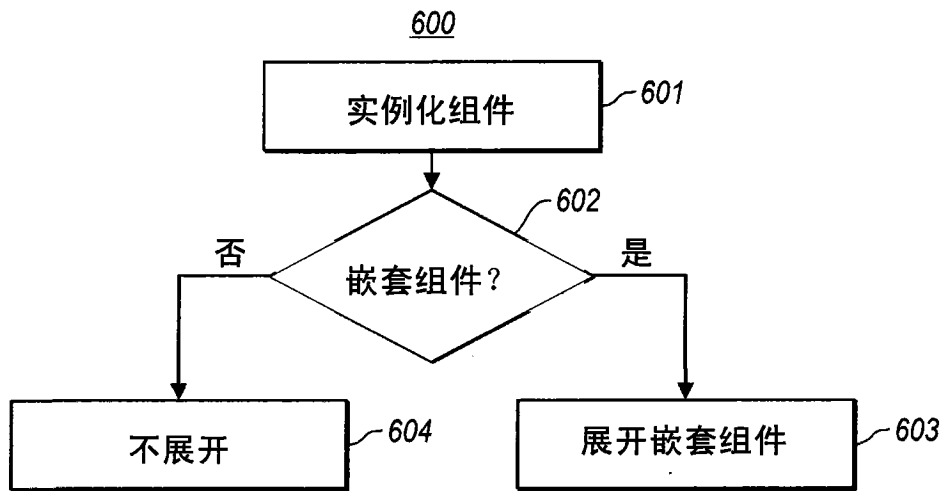


图 6

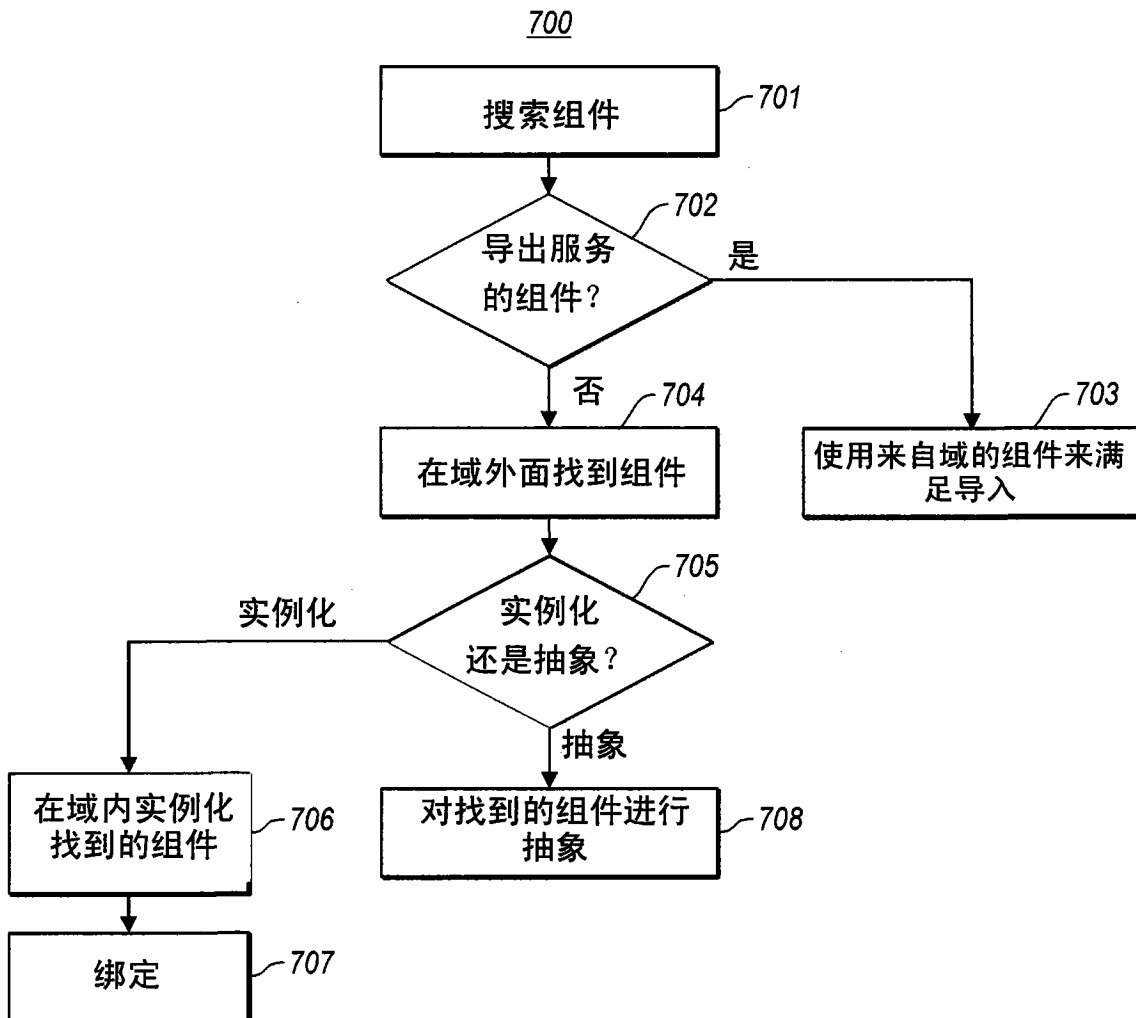


图 7