

(12) 发明专利申请

(10) 申请公布号 CN 102043665 A

(43) 申请公布日 2011. 05. 04

(21) 申请号 201010506209. 7

(22) 申请日 2010. 10. 14

(30) 优先权数据

12/579, 199 2009. 10. 14 US

(71) 申请人 SAP 股份公司

地址 德国瓦尔多夫

(72) 发明人 约肯·古尔特勒

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 邵亚丽

(51) Int. Cl.

G06F 9/46 (2006. 01)

G06F 9/54 (2006. 01)

H04L 29/06 (2006. 01)

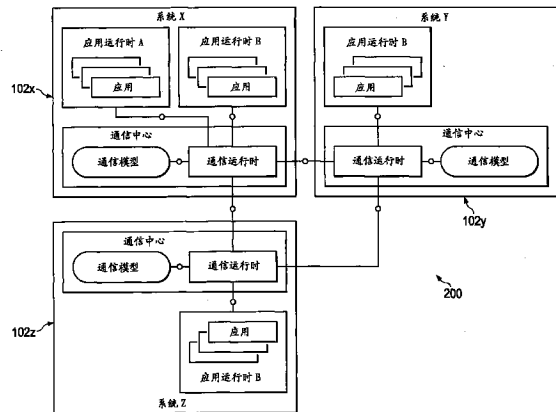
权利要求书 2 页 说明书 7 页 附图 5 页

(54) 发明名称

知晓系统全景的应用间通信基础架构

(57) 摘要

本公开涉及有关知晓系统全景的应用间通信基础架构的系统、方法和软件。该应用间通信基础架构可以实现一个元模型来描述应用如何参与应用间通信。该元模型能够描述应用接受的输入参数及其可能的输出。基于将被披露或以其他方式使所涉及的应用可用的元数据, 存在一种定义应用间的通信流的协议。



1. 一种计算机实现的方法,用于通过知晓全景的应用间通信基础架构进行通信,该方法包括由一个或多个处理器执行的以下步骤:

收集存储在存储器中的用于与异构应用通信的信息;

至少部分地使用所收集的信息产生以异构应用为目标的消息;以及

将该消息传送给事件驱动的应用间通信基础架构,以便随后传输给该异构应用,该应用间通信基础架构能够识别用于该异构应用的通信结构并基于该识别出的结构翻译所述消息。

2. 根据权利要求 1 所述的计算机实现的方法,所述通信结构存储所述异构应用的可用输入和输出。

3. 根据权利要求 1 所述的计算机实现的方法,所述通信结构存储所述异构应用的可用输入端口和输出端口。

4. 根据权利要求 1 所述的计算机实现的方法,所述通信结构包括与所述异构应用分离的披露的文件。

5. 根据权利要求 4 所述的计算机实现的方法,所述披露的文件包括 XML 文件。

6. 根据权利要求 1 所述的计算机实现的方法,所述通信结构存储具体消息的每个参数的数据类型、基数以及强制性。

7. 一种计算机程序产品,用于通过知晓全景的应用间通信基础架构进行通信,该计算机程序产品包括存储计算机可读指令的有形存储介质,所述计算机可读指令在被执行时能够操作用以:

收集存储在存储器中的用于与异构应用通信的信息;

至少部分地使用所收集的信息产生以异构应用为目标的消息;以及

将该消息传送给事件驱动的应用间通信基础架构,以便随后传输给该异构应用,该应用间通信基础架构能够识别用于该异构应用的通信结构并基于该识别出的结构翻译所述消息。

8. 根据权利要求 7 所述的计算机程序产品,所述通信结构存储所述异构应用的可用输入和输出。

9. 根据权利要求 7 所述的计算机程序产品,所述通信结构存储所述异构应用的可用输入端口和输出端口。

10. 根据权利要求 7 所述的计算机程序产品,所述通信结构包括与所述异构应用分离的披露的文件。

11. 根据权利要求 8 所述的计算机程序产品,所述披露的文件包括 XML 文件。

12. 根据权利要求 7 所述的计算机程序产品,所述通信结构存储具体消息的每个参数的数据类型、基数以及强制性。

13. 根据权利要求 7 所述的计算机程序产品,所述应用间通信基础架构和所述异构应用各自在相异的运行环境中的不同处理器上执行。

14. 一种计算机程序产品,用于通过知晓全景的应用间通信基础架构进行通信,该计算机程序产品包括存储计算机可读指令的有形存储介质,所述计算机可读指令在被执行时能够操作用以:

接收来自第一异构应用的消息,该消息以第二异构应用为目标;

动态地识别所存储的用于该第二异构应用的通信结构,该通信结构由该第二异构应用披露;

基于该识别出的结构将所述消息翻译成与该第二异构应用兼容的消息;以及将该翻译后的消息传送给该第二异构应用以进行处理。

15. 根据权利要求 14 所述的计算机程序产品,所述通信结构存储所述异构应用的可用输入和输出。

16. 根据权利要求 14 所述的计算机程序产品,所述通信结构存储所述异构应用的可用输入端口和输出端口。

17. 根据权利要求 14 所述的计算机程序产品,所述通信结构包括与所述异构应用分离的披露的文件。

18. 根据权利要求 14 所述的计算机程序产品,所述通信结构存储具体消息的每个参数的数据类型、基数以及强制性。

19. 根据权利要求 14 所述的计算机程序产品,所述异构应用在相异的运行环境中的不同处理器上执行。

20. 根据权利要求 14 所述的计算机程序产品,所述异构应用基于相异的计算机程序语言,各自采用不同的消息结构。

## 知晓系统全景的应用间通信基础架构

### 技术领域

[0001] 本公开涉及有关知晓系统全景的应用间通信基础架构 (systemlandscape aware inter-application communication infrastructure) 的系统、方法和软件。更具体来说, 本公开包括创建或使用一个元模型 (在各种所涉及的应用中一致地使用) 来描述应用或其它程序如何参与应用间的通信。

### 背景技术

[0002] 也可以称为应用 (application) (如商业应用) 的计算机程序在运行环境中运行。在有些情况下, 运行环境是计算机操作系统, 如地址为 1020 102ndAve NE Bellevue, WA 98004 的 Microsoft Corporation (微软公司) 提供的 Microsoft Windows。在有些情况下, 运行环境提供可供运行在该运行环境中的一个或多个应用使用的基本的或必要的以及可选的操作能力或特征。典型地, 运行环境的特征是事先已知的, 而应用则是基于这些预知信息构造或配置的。也就是说, 应用是基于已知将由运行环境提供的特征而特别构造的。更具体地说, 通常所安装的软件来自不同厂商、有不同的版本并使用不同的技术。即便是对于仅涉及单一厂商的全景 (landscape) 来说, 业务也可能运行在范围广泛的不同版本的不同产品上 (并且即使是这些单一厂商的产品也可能基于不同的技术、语言、协议或消息传输结构)。因此, 即使通过动态链接库实现, 各种应用和运行环境之间也存在静态关系。

### 发明内容

[0003] 本公开的一个方面涉及知晓全景的应用间通信基础架构。例如, 用于通过知晓全景的应用间通信基础架构进行通信的计算机程序产品包括存储计算机可读指令的有形存储介质。在一种情况下, 指令可操作用来收集存储在存储器中的用于与异构应用通信的信息, 然后至少部分地使用所收集的信息生成以该异构应用为目标的消息。该计算机程序产品将该消息传送给应用间通信基础架构, 以便随后发送给该异构应用。在此, 该应用间通信基础架构能够识别用于该异构应用的通信结构, 并基于该识别出的结构翻译该消息。

[0004] 在另一种情况下, 应用间通信基础架构从第一异构应用接收消息, 该消息以第二异构应用为目标。该计算机程序产品中的指令可操作用于动态识别所存储的用于该第二异构应用的通信结构, 该通信结构由该第二异构应用披露 (expose)。应用间通信基础架构基于该识别出的结构将消息翻译成与第二异构应用兼容的消息, 然后将翻译后的消息传送给该第二异构应用以进行处理。

[0005] 不同的实施例可以实现或采用以下特征中的一个或多个。具体地说, 通信结构可以存储异构应用的可用输入和输出。在另一例子中, 通信结构可以存储异构应用的可用输入端口 (in port) 和输出端口 (out port)。在又一例子中, 通信结构可以存储具体消息的每个参数的数据类型、基数 (cardinality) 以及强制性 (mandatory)。在再一个例子中, 通信结构可以是与异构应用分离的披露的文件, 该披露的文件是 XML 文件。

[0006] 虽然被一般性地描述为处理和变换相应数据的计算机实现的软件, 但所述方面的

一些或全部可以是计算机实现的方法或进一步包括在用于执行所述功能的相应系统或其它设备中。本公开的这些和其它方面以及实施例的细节在附图和以下描述中阐述。通过说明书和附图,以及通过权利要求,本公开的优点将变得清楚。

### 附图说明

[0007] 图 1 示出根据本公开的一个实现方案的、通过知晓全景的应用间通信基础架构进行通信的示例系统;

[0008] 图 2 示出在图 1 描述的系统中的高层组件的示例配置;

[0009] 图 3A 和 3B 是根据本公开的一个实现方案的示例混搭 (mash-up) 配置;

[0010] 图 4A 和 4B 示出通过知晓全景的应用间通信基础架构传送数据流的示例流程图。

### 具体实施方式

[0011] 本公开的一方面涉及知晓全景的应用间通信基础架构。在特定配置中,该基础架构不局限于客户端一侧或服务器一侧的措施,也不局限于特定系统全景 (system landscape),并且也不局限于任何具体的 UI (用户接口) 技术。更具体地说,该应用间通信基础架构可以实现一个元模型 (一致地应用于所涉及的应用中),用以描述应用如何参与应用间的通信。该元模型可以描述应用接受的输入参数及其可能的输出。在典型的混搭 (mashup) 场景中这也称为各个应用的输入 / 输出。无论如何,该元模型通常不仅仅单纯定义输入 / 输出数据的存在,而且还定义结构 (诸如数据类型或基数) 以及该数据对应用的消息传输来说是否是强制性的 (mandatory) (或可选的 (optional))。元数据将被披露或以其它方式为所涉及的应用可用,基于元数据,存在一种定义应用之间的通信流的协议。在混搭场景中,这可以称为应用之间的“连线” (wiring)。换言之,应用的元模型以及该协议帮助定义应用的连线并实现了对所需的应用间通信的描述。在元模型之外,还可以存在对系统全景 (即,每个所涉及的应用正在哪台机器 / 技术栈 / UI 技术上运行) 的描述,可以将该描述更简单地称为“通信模型”。通信模型可以由为所涉及的应用平台实现的通信运行时 (runtime) 来执行。通信运行时通常是通信中心 (communication hub) 的一部分。如此,通信中心能够彼此连接并由此能够保证基于当前系统场景的最优化通信处理。在特定实施例中,知晓全景的应用间通信基础架构帮助解决将现有的功能 / 应用组合成新的组合应用 / 场景的需求。换句话说,如果企业或其它机构确定需要拥有更宽松耦合的应用场景,就有可能相应地期望一种能够采用现有 (异构) 系统全景的、灵活的应用间通信基础架构。

[0012] 所提到的用于应用间通信的基础架构可被用于许多版本、实例及类型 (例如基于事件) 的应用间通信。根据实现方案,其可以被用作 JAVA 栈、ABAP 栈两者的一部分或者其它语言或组合的一部分。一个示例实现方案涉及混搭场景,在混搭场景中,(潜在地运行于不同 UI 技术之上的) 应用的连接方式使得其组合犹如一个单一的应用。事件驱动的应用间通信基础架构可以帮助实现应用之间的数据流。在本例中,如果存在不同机器和 / 或 UI 技术间的混合,则根据应用间的具体连线,通信将不再必须仅在客户端一侧或服务器一侧执行。

[0013] 现在转向示出的实施例,图 1 示出实现知晓全景的应用间通信基础架构的示例系统 100。系统 100 可以是跨越一个或多个网络,如网络 112,的分布式客户端 / 服务器系统。

在这样的实现方案中,可以使用任何标准的或专有的加密算法以加密格式传送或存储数据。但是系统 100 也可以处于专用的企业环境中——跨越局域网或子网——或者处于其它任何适当的环境中,而不脱离本公开的范围。系统 100 可以包括服务器 102、一个或多个客户端 104 以及网络 112,或可通信地与它们耦合。

[0014] 服务器 102 包括可操作用以接收、发送、处理和存储与系统 100 相关的数据的电子计算设备。一般来说,图 1 仅提供了可以用于本公开的计算机的一个示例。每个计算机一般都旨在包含任何适当的处理设备。例如,尽管图 1 示出了可用于本公开的一个服务器 102,但系统 100 也能够使用服务器以外的计算机来实现,或者使用服务器池来实现。实际上,服务器 102 可以是任何计算机或处理设备,如刀片式服务器、通用个人计算机 (PC)、Macintosh、工作站、基于 Unix 的计算机,或任何其它合适的设备。换言之,本公开考虑了通用计算机以外的计算机以及没有常规操作系统的计算机。服务器 102 可被适配为执行任何操作系统,包括 Linux、UNIX、Windows Server,或任何其它合适的操作系统。根据一个实施例,服务器 102 还可以包括 web 服务器和 / 或邮件服务器,或与它们可通信地耦合。

[0015] 服务器 102 可以包括本地存储器 120。存储器 120 是任何有形的计算机可读存储器,其可包括任何易失性或非易失性存储器形式的存储器或数据库模块,包括但不限于磁介质、光介质、随机存取存储器 (RAM)、只读存储器 (ROM)、可移动介质,或任何其它合适的本地或远程存储组件。所示出的存储器 120 除其它项目以外可以包括知晓全景的应用间通信基础架构 130 以及多个应用 140,这些将在以下详述。此外,存储器 120 可以包括一个或多个运行环境 135。存储器 120 还可以包括其它类型的数据,如环境和 / 或应用描述数据、用于一个或多个应用的应用数据,以及涉及 VPN 应用或服务、防火墙策略、安全或访问日志、打印或其它报告文件、超文本标记语言 (HTML) 文件或模板、相关及不相关的软件应用或子系统等等的数据。因此,存储器 120 也可以被看作是数据仓库,如来自一个或多个应用的本地数据仓库。

[0016] 更具体地说,应用 140 是任何应用、程序、模块、过程或其它能够使用本身也是软件的知晓全景的应用间通信基础架构或有助于使用知晓全景的应用间通信基础架构的软件。不管具体实现方案如何,“软件”可以包括软件、固件、连线的或编程的硬件,或包含在有形计算机可读介质上的它们的任何组合。实际上,形式变换模块 130 可以用任何适当的计算机语言来编写或描述,包括 C、C++、Java、Visual Basic、汇编语言、Perl、任何合适版本的 4GL,等等。例如,形式变换模块 130 可以是复合应用,其一些部分可以被实现为 Enterprise Java Beans (EJBs),或者,设计时 (design-time) 组件可以具有将运行时实现 (run-time implementation) 生成到不同平台中的能力,所述平台如 J2EE (Java 2 Platform, Enterprise Edition, Java 2 平台企业版)、ABAP (Advanced Business Application Programming, 高级商务应用编程) 对象,或 Microsoft 的 .NET。应该理解,虽然图 1 所示应用 140 为单个模块,但应用 140 也可以包括大量其它子模块,或者也可以替代为通过各种对象、方法或其它过程实现各种特征和功能的单个多任务模块。此外,虽然被示为在服务器 102 的内部,但与应用 140 相关的一个或多个过程可以被远程地存储、引用或执行。例如,应用 140 的一部分可以是远程调用的 web 服务,而应用 140 的另一部分则可以是为在远程客户端 104 进行处理而捆绑 (bundle) 的接口对象。再有,应用 140 可以是其它软件模块或企业应用的子代 (child) 或子模块 (未示出) 而不脱离本公开的范围。

[0017] 为了使用知晓全景的应用间通信基础架构,多个应用 140——不管是在同一计算机上还是跨过网络 112——通过知晓全景的应用间通信基础架构 130 相互可通信地耦合。该耦合例如可以使用 HTTP 连接或任何其它合适的、可工作用以以期望格式传送消息的通信通道来实现。应用间通信基础架构 130 通常包括事件驱动模块,该模块在运行时能够识别目标异构应用的通信结构,并基于该识别出的结构翻译消息。例如,应用间通信基础架构 130 可以从第一异构应用接收消息,该消息的目标为第二异构应用。然后,应用间通信基础架构 130 动态地识别用于第二异构应用的、所存储的通信结构,该通信结构是由该第二异构应用披露的。应用间通信基础架构 130 基于该识别出的结构将所述消息翻译成与该第二异构应用兼容的消息,并将翻译后的消息传送给第二异构应用以进行处理。

[0018] 根据一些实现方案,通信结构 145 可以是一个或多个可扩展标记语言 (XML) 文档文件或部分或结构化查询语言 (“SQL”) 语句或脚本语言的脚本或扩展语言的形式。例如,通信结构 145 可以被格式化、存储、或定义为文本文件、虚拟存储器存取方法 (Virtual Storage Access Method, VSAM) 文件、平面文件、Btrieve 文件、逗号分隔值 (comma-separated-value, CSV) 文件、内部变量或一个或多个库中的各种数据结构。除了 XML,其它示例的脚本语言可以包括 JavaScript、层叠样式表单 (Cascading Style Sheets, CSS)、HTML、异步 JavaScript 和 XML (asynchronous JavaScript and XML, AJAX),等等。无论具体语言或格式如何,应理解的是,通信结构 145 可以是与具体应用 140 紧密相关的文件、由具体应用 140 披露的在物理上分离的文件,或与用于多个应用 140 的知晓全景的应用间通信基础架构 130 相关联的单个文件。

[0019] 以下示出的示例通信结构 145 的形式是用于假设域名 xyz.com 的 XML 脚本。具体地说,在这样的基础架构中使用的协议可以是例如基于 XML 的。例如可以在应用间发送“命令”或其它消息的通信结构 145 的一个 XML 部分可能遵循以下示例格式:

```
[0020] <? xml version = " 1.0" encoding = " iso-8859-1" ? >
[0021]     <ccp:CommandsIn xmlns:ccp = " http://www.xyz.com/namespaces/ccp" >
[0022]         <ccp:Command name = " FireInport" >
[0023]             <ccp:Port name = " airline" >
[0024]                 <atom:feed xmlns:atom = " http://www.w3.org/2005/
Atom" >
[0025]                     <atom:event>
[0026]                         <CARRID>
[0027]                             UA
[0028]                         </CARRID>
[0029]                     </atom:event>
[0030]                 </atom:feed>
[0031]             </ccp:Port>
[0032]         </ccp:Command>
[0033]     </ccp:CommandsIn>
```

[0034] 在此,转发 (forward) 命令“FireInport”,并且输入端口 (inport) 的具体数据以 ATOM 兼容的格式定义。在其之后的 XML 纲要 (XML schema) 在结尾定义“通信模型”。当然,

应该理解,以上仅仅是出于说明的目的,并非想要限制本公开的范围。换言之,以上示例脚本是为了帮助提供上下文和描述,其它脚本可以实现与之不同的功能和功能性而仍在本公开的范围。

[0035] 服务器 102 还可以包括处理器 125。处理器 125 执行指令,如前述软件,并操纵数据以执行服务器 102 的操作,并且服务器 102 例如可以是中央处理单元 (CPU)、刀片式服务器、专用集成电路 (ASIC) 或现场可编程门阵列 (FPGA)。尽管图 1 示出的是服务器 102 中的单个处理器 125,但根据特定的需要也可以使用多个处理器 125,并且提到处理器 125 时即意味着在适当的情况下包括多个处理器 125。在所示出的实施例中,处理器 125 执行应用 140 和应用间通信基础架构 130。

[0036] 服务器 102 还可以包括用于在客户端 - 服务器或其它分布式环境中通过网络 112 与其它计算机系统,如客户端 104,通信的接口 117。在特定实施例中,服务器 102 通过接口 117 接收来自内部或外部发送者的数据以存储在存储器 120 中和 / 或由处理器 125 处理。一般来说,接口 117 包括以适当组合的软件和 / 或硬件编码并可工作用以与网络 112 通信的逻辑。更具体来说,接口 117 可以包括支持一个或多个与通信网络 112 相关联的通信协议的软件或可操作用以传送物理信号的硬件。

[0037] 网络 112 便利计算机服务器 102 和任何其它本地或远程计算机 (如客户端 104) 之间的无线或有线通信。网络 112 可以是企业或安全网络的全部或一部分。在另一示例中,网络 112 可以是仅仅在服务器 102 和客户端 104 之间的跨越有线线路或无线链接的 VPN。这样的示例无线链接可以经由 802. 11a、802. 11b、802. 11g、802. 20、WiMax,等等。虽然所示出的是单一的或连续的网络,但网络 112 可以在逻辑上划分为各种分支网络或虚拟网络而不脱离本公开的范围,只要网络 112 的至少一部分能够便利服务器 102 和至少一个客户端 104 之间的通信即可。例如,服务器 102 可以通过一个子网可通信地耦合到数据仓库,同时通过另一个子网可通信地耦合到某个客户端 104。换言之,网络 112 包括任何可工作用以便利系统 100 中的各种计算组件之间的通信的内部或外部网、网络、子网或它们的组合。网络 112 例如可以在网络地址间传送网际协议 (Internet Protocol, IP) 分组、帧中继帧、异步传输模式 (ATM) 信元、语音、视频、数据,以及其它适当的信息。网络 112 可以包括一个或多个局域网 (LAN)、无线接入网 (RAN)、城域网 (MAN)、广域网 (WAN)、被称为互联网的全球计算机网络的全部或部分,和 / 或位于一个或多个地点的任何其它通信系统。在特定实施例中,网络 112 可以是用户可以通过特定的本地或远程客户端 104 访问的安全网络。

[0038] 客户端 104 可以是任何可工作用以使用任何通信链路与服务器 102 或网络 112 连接或通信的计算设备。在高层,每个客户端 104 至少包括或执行 GUI (图形用户接口) 136,并且每个客户端包括可工作用以接收、发送、处理和存储任何与系统 100 相关联的合适的数据的电子计算设备。应理解的是,可以有任何数量的客户端 104 与服务器 102 可通信地耦合。此外,“客户端 104”和“用户”可以适当地交换使用而不会脱离本公开的范围。此外,为易于说明,每个客户端 104 被描述为由一个用户使用。但本公开也考虑到了许多用户可以使用一台计算机或一个用户可以使用多台计算机的情况。如在本公开中使用的,客户端 104 旨在包括个人计算机、触摸屏终端、工作站、网络计算机、信息亭、无线数据端口、智能手机、个人数据助理 (PDA)、在这些设备或其它设备中的一个或多个处理器,或任何其它合适的处理设备。例如,客户端 104 可以是可工作用以与外部或非安全网络无线连接的 PDA。在



另一个例子中,客户端 104 可以包括便携式计算机,该便携式计算机包括如键区、触摸屏、鼠标或其它能够接受信息的设备的输入设备,以及传达与服务器 102 或客户端 104 的操作相关联的信息,包括数字数据、可视信息或 GUI 136,的输出设备。输入设备和输出设备都可以包括固定的或可移动的存储介质,如计算机磁盘、CD-ROM,或其它能够通过显示器、即 GUI 的客户端部分或应用接口 136 从客户端 104 的用户接收输入并向客户端 104 的用户提供输出的合适的介质。

[0039] GUI 136 包括可工作用以允许客户端 104 的用户与系统 100 的至少一部分为任何合适的目的(如查看应用或其它事务数据)而接口的图形用户接口。例如,GUI 136 可以呈现有关通过应用间通信基础架构 130 在异构的应用 140 间通信的业务信息的各种视图。一般来说,GUI 136 向特定用户提供由系统 100 提供的或在系统 100 内通信的数据的有效且用户友好的表示。GUI 136 可以包括多个具有交互栏、下拉列表和由用户操作的按钮的可定制的画面或视图。GUI 136 还可以呈现多个入口(portal)或应用程序控制面板(dashboard)。例如,GUI 136 可以显示允许用户输入和定义用于一个或多个模块的搜索参数的安全网页。应理解,术语“图形用户接口”可用作单数或复数,以描述一个或多个图形用户接口以及特定图形用户接口的每个显示。实际上,不脱离本公开的范围,在适当的情况下,提到 GUI 136 可以指应用 130 的前端或组件,以及通过客户端 104 可访问的特定接口。因此,GUI 136 考虑了任何在系统 100 中处理信息并有效地向用户呈现结果的图形用户接口,如通用 web 浏览器或触摸屏。服务器 102 可以通过 web 浏览器(如 Microsoft Internet Explorer 或 Netscape Navigator)从客户端 104 接受数据并利用网络 112 向浏览器返回适当的 HTML 或 XML 响应。

[0040] 图 2 示出系统 100 中的高层组件的示例配置 200。具体来说,图 2 示出三个分别在相应的应用运行时内执行一个或多个应用 140 的示例系统(X、Y 和 Z)。此外,每个系统都实现了包括(或链接到)通信运行时模块和通信模型的通信中心。在该实现方案中,可以将其它应用收集在混搭(mashup)场景中,混搭场景更容易跨越相异的运行时实现动态数据流。替代通过远程调用在本地系统上处理所述数据流,可以由远程系统通过通信元模型(部分地或全部地)实现该数据流。换言之,本地系统可以通过通信基础架构知晓远程异构应用,然后通信基础架构允许远程应用以期望的协议接收数据消息。图 3 示出更详细的配置,其中示出包括连线(wiring)实现通信元模型的扩展的混搭场景。例如,ABAP 可以表示包括 CHIP(芯片)模块或 applet(小应用程序)的后端系统。通信元模型允许连线存在于前端和后端之一或两者之上。换言之,Java 前端可以知晓一个或多个后端系统正在执行 ABAP 运行时,但不需要到 ABAP 系统各种组件的“硬连线”链路。通过这种方式,ABAP 系统可以更有效地管理 ABAP 应用或子模块间的消息和数据流,即便它们是由能够极大减少往返并提高性能的 Java 系统产生的。

[0041] 对于运行的一个方面,图 4A 是示出示例计算机过程 400 的流程图,计算机过程 400 用于在系统 X(如图 2 中描述的系统)中的应用 A1 和应用 B1 之间进行通信。为表达清楚起见,以下在图 2 所示的环境 200 的上下文中一般地描述方法 400。但是应理解,方法 400 也可以替代地通过任何其它合适的系统、环境或合适的系统和环境的组合来执行。

[0042] 所示方法 400 于步骤 402 开始,该步骤通过定义通信模型(即适当的元数据)使得能够在设计时中进行该通信。在本例中,该通信模型定义从应用 A1 至应用 B1 的数据流。

尽管未示出,但在该数据流中可以包含任何种类的数据变换器,例如将应用 A1 披露的地址数据变换成应用 B1 需要的纬度 / 经度数据。

[0043] 在步骤 404,确定数据流要被执行或正被执行。这一执行步骤可通过点击 A1 中的按钮、改变 A1 中的表选择或任何其它合适的运行时动作来触发。然后,在步骤 406,调用通信运行时。下一步,在步骤 408 通信运行时了解所定义的通信模型并检查是否有为 A1 定义的、外发的 (outgoing) 数据流。在步骤 410,通信运行时处理该数据流 (包括数据变换步骤),在步骤 412,最终 (在合适的时候) 将数据转发给应用运行时 B。应用运行时 B 将该数据转发给应用 B1,并且应用 B1 被该新数据更新。

[0044] 对于运行的另一方面,图 4B 是使出示例计算机过程 450 的流程图,计算机过程 450 用于在运行于不同系统 (如图 2 所描述的环境) 上的应用之间的应用间通信。为表述清楚起见,以下在图 2 所示的环境 200 的上下文中一般地描述方法 450。例如,利用图 2 所示的环境,可以有定义为从系统 Y 上的应用 B1 至系统 X 上的应用 A1 再到系统 X 上的应用 B2 的三步数据流。但是应理解,方法 450 也可以替代地用任何其它合适的系统、环境或合适的系统和环境的组合来执行。

[0045] 对于过程 450 来说,所示出的第一步骤包括在步骤 452 在设计时中定义元数据 / 通信模型。在这种情况下,为了在运行时实现最优化的且知晓系统全景的处理,要在步骤 454 使该通信模型为数据流所涉及的合适的分布式系统可用。可以通过任何适当的方式来实现这种可用性,包括将在设计时的数据复制到所涉及的系统 (即复制到所涉及的通信运行时)、使用中央仓库来存储通信模型 (以使在本例中通信运行时通过适当的远程调用访问该通信模型)、以及将数据存储在一个系统中并在运行时将数据传递给数据流所涉及的系统。

[0046] 在示例的运行时,在步骤 454,数据流从系统 Y 上的应用 B1 开始 (如在以上的简单例子中那样通过任何种类的用户交互开始)。在步骤 456,应用运行时 B 将其转发给通信运行时。在步骤 458,通信运行时检查通信模型并确定数据流是远程的。一旦确定,就在步骤 460 调用目标应用本地的或以其他方式与目标应用关联的通信运行时,从而如果需要可以将其启动。在步骤 462,当前通信运行时将其转发给系统 X 上的目标通信运行时。在步骤 464,目标通信运行时利用可用的通信模型来处理数据。由于通信模型在系统 X 上也可用,因此系统 X 上的通信运行时能够处理数据流中的下两个步骤而不用反向调用系统 Y 上的在前的通信运行时。如此,跨越两个系统上运行的三个应用处理了数据流,但这种处理方式尽可能地避免了远程调用。

[0047] 虽然本公开使用多个数据流和处理流以及相关描述来举例说明与所公开的各种方法和技术相关的示例技术,但系统 100 考虑了使用或实现任何合适的技术以用于执行这些和其他任务。应理解,这些技术仅为说明的目的,可以在任何适当的时间执行所描述的或类似的技术,包括并发执行、单独执行或组合执行。此外,这些流程中的许多步骤可以同时发送和 / 或以与所示出和描述的不同的顺序发生。再有,系统 100 可以使用具有附加步骤、较少步骤和 / 或不同步骤的过程和方法,只要这些过程和方法仍然是合适的。简言之,尽管根据一定的实现方案和一般关联的方法对本公开进行了描述,但是这些实现方案和方法的变更和置换对本领域技术人员来说是显而易见的。因此,其它实现方案也在本公开的范围内。

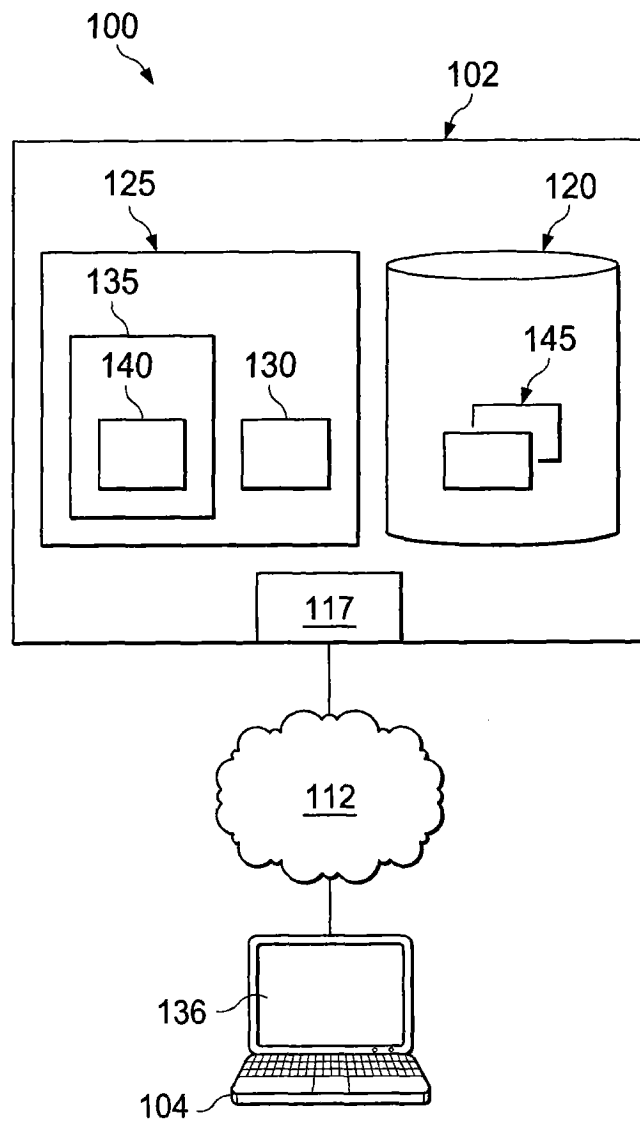


图 1

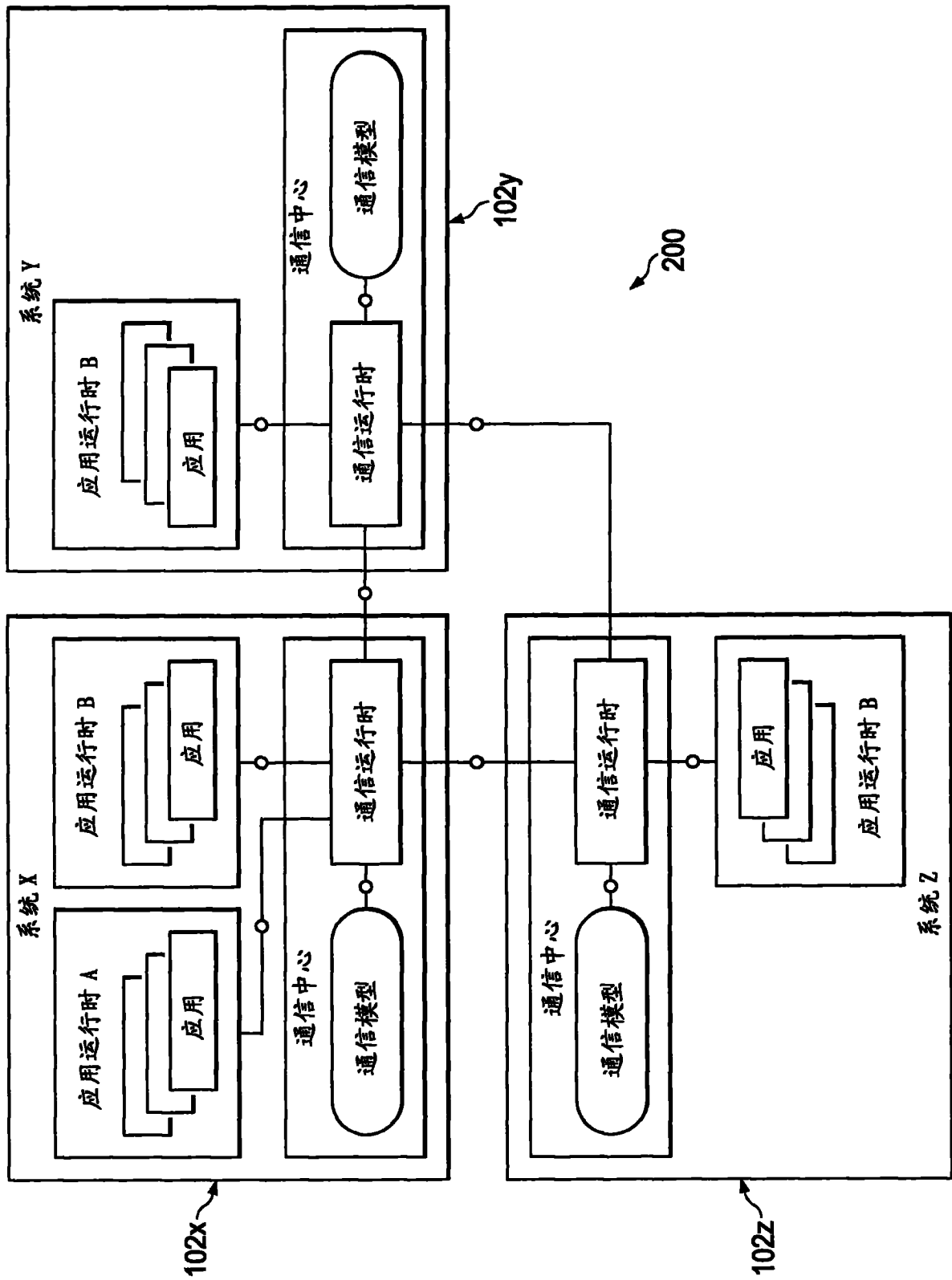
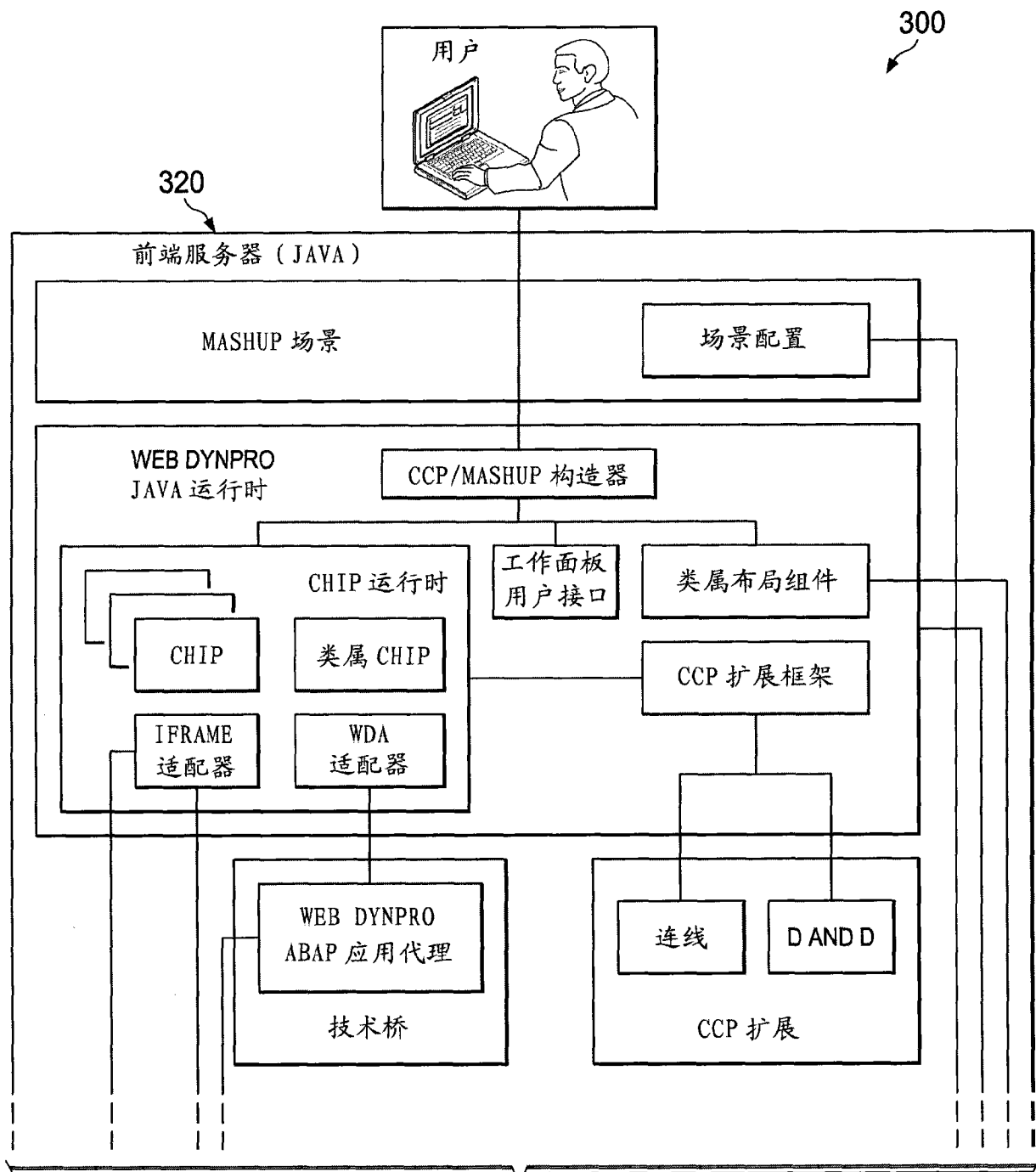
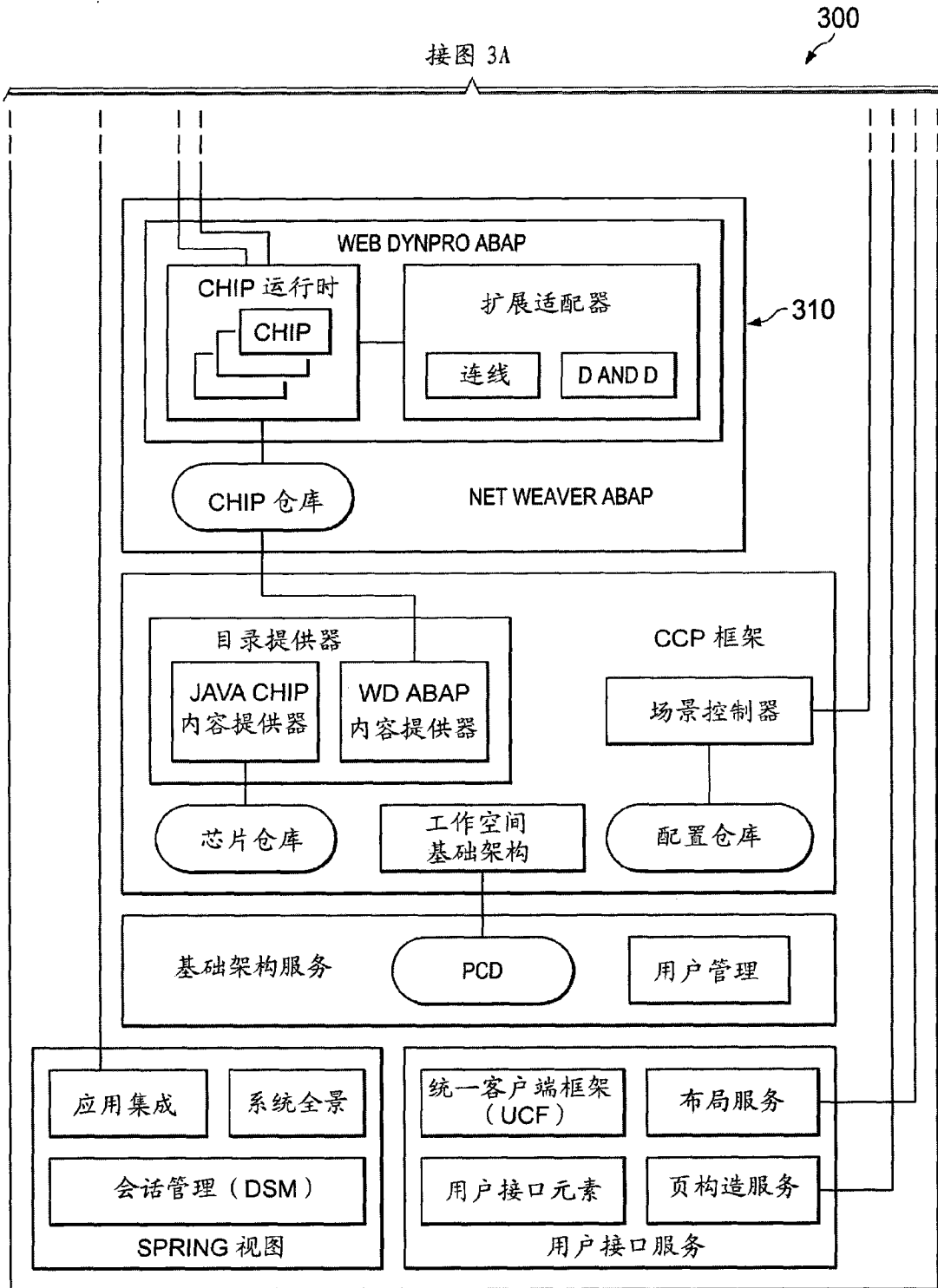


图 2



转图 3B

图 3A



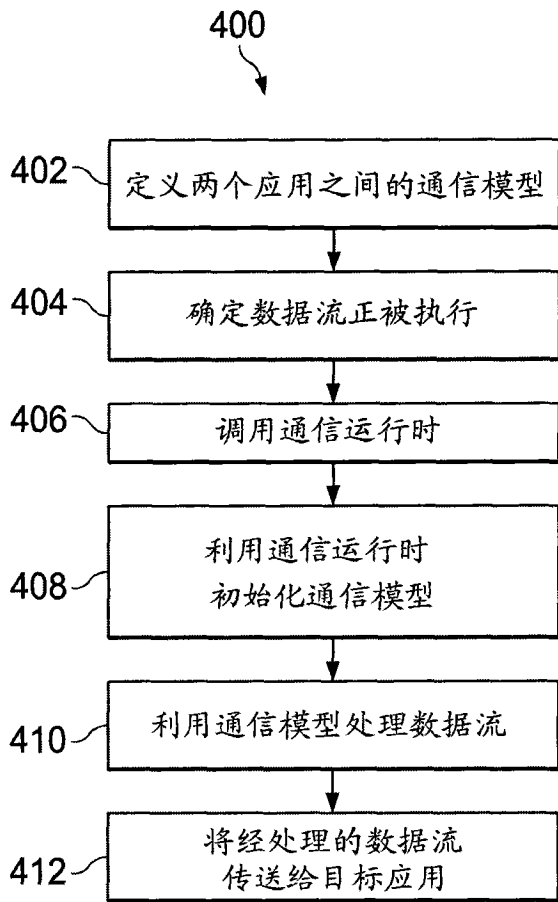


图 4A

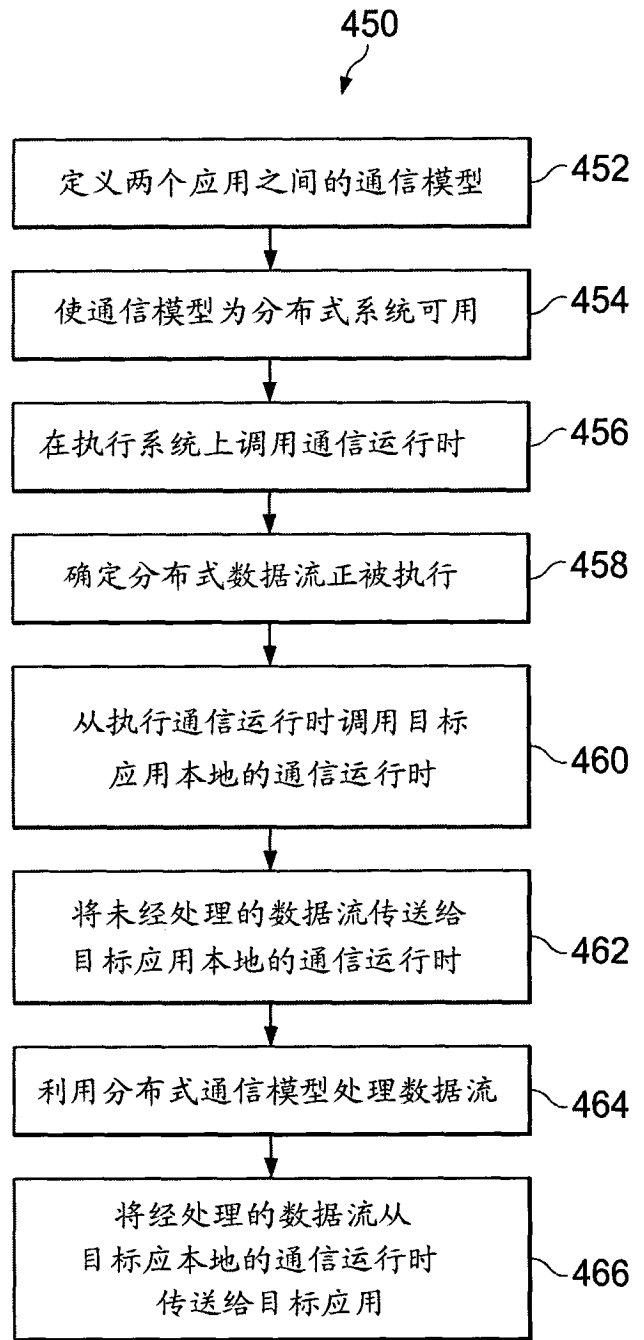


图 4B