



(12) 发明专利申请

(10) 申请公布号 CN 101882073 A

(43) 申请公布日 2010. 11. 10

(21) 申请号 200910137109. 9

(22) 申请日 2009. 05. 04

(71) 申请人 谭家辉

地址 中国香港沙田科技园西大街 5 号企业广场 9 栋 3 层 310C 室

(72) 发明人 谭家辉

(74) 专利代理机构 北京市金杜律师事务所  
11256

代理人 吴立明 庞淑敏

(51) Int. Cl.

G06F 9/44 (2006. 01)

H04L 29/08 (2006. 01)

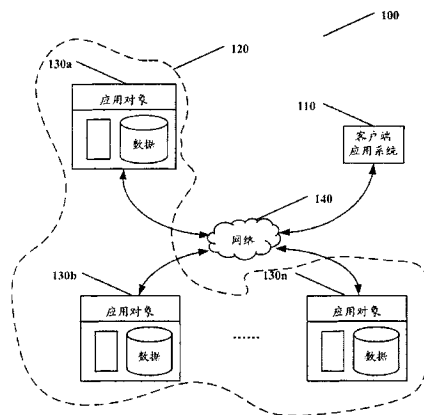
权利要求书 4 页 说明书 15 页 附图 14 页

(54) 发明名称

面向服务的应用系统及其通信方法、创建器和创建方法

(57) 摘要

本发明公开了一种面向服务的应用系统,所述应用系统包括一个或多个应用对象,每个应用对象包括:专用于存储特定类型数据的平面结构数据单元;用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元;以及用于操作该数据单元中的数据和该关系存储单元中的信息的应用代码模块,其中,所述应用代码模块用于响应于网络服务请求,对平面结构数据单元和/或关系存储单元进行操作。本发明还提供了该面向服务的应用系统的通信方法、用于创建该面向服务的应用系统的创建器和方法。通过本发明,可以快速建立和修改该面向服务的应用系统,且在应用系统中不存在信息孤岛,此外还可使用户的业务数据与其供应商、客户和合作方共享,从而为用户带来诸多益处。



1. 一种面向服务的应用系统,所述应用系统包括一个或多个应用对象,每个应用对象包括:

专用于存储特定类型数据的平面结构数据单元;

用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元;以及

用于操作该数据单元中的数据和该关系存储单元中的信息的应用代码模块,

其中,所述应用代码模块用于响应于网络服务请求,对平面结构数据单元和 / 或关系存储单元进行操作。

2. 根据权利要求 1 所述的应用系统,其中,所述应用对象包括平台,该平台是一种应用对象,包括专用于存储所述配置信息的平面结构数据单元,以及其中,该平台中的应用代码模块用于响应于针对所述配置信息的网络服务请求,查找相应的配置信息,以便返回该配置信息。

3. 根据权利要求 2 所述的应用系统,其中,所述平台的关系存储单元用于存储其他应用对象的关系存储单元中的信息。

4. 根据权利要求 1 所述的应用系统,其中,所述应用对象包括数据应用对象,其包括专用于存储业务数据的平面结构数据单元。

5. 根据权利要求 4 所述的应用系统,其中,所述数据应用对象进一步包括专用于存储所述配置信息中与之相关的公共部分的平面结构数据单元。

6. 根据权利要求 1 所述的应用系统,进一步包括中间件,用于把所述配置信息转换为能够由与所述面向服务的应用系统相关的各种客户端应用系统解释的通用格式的信息。

7. 根据权利要求 1 所述的应用系统,进一步包括适配器,用于将来自第三方系统的网络服务请求或所返回的网络服务响应转换成格式适合于所述面向服务的应用系统或者适合于与所述面向服务的应用系统相关的客户端应用系统的信息;和 / 或

将所述面向服务的应用系统或者所述客户端应用系统返回的网络服务响应或者发出的网络服务请求转换成格式适合于第三方系统的信息。

8. 根据权利要求 1 所述的应用系统,其中,所述应用代码模块用于进行以下操作中的一种或者多种:

向平面结构数据单元添加数据;

删除平面结构数据单元中的数据;

更新平面结构数据单元中的数据;

检索平面结构数据单元中的数据;

向关系存储单元添加信息;

删除关系存储单元中的信息;

更新关系存储单元中的信息;以及

检索关系存储单元中的信息。

9. 一种面向服务的应用系统的通信方法,其中,所述面向服务的应用系统包括一个或多个应用对象,每个应用对象包括:

专用于存储特定类型数据的平面结构数据单元;

用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存

储单元 ; 以及

用于操作该数据单元中的数据和该关系存储单元中的信息的应用代码模块,

所述方法包括 : 响应于网络服务请求, 通过应用代码模块对平面结构数据单元和 / 或关系存储单元进行操作。

10. 根据权利要求 9 所述的方法, 其中, 所述应用对象包括平台, 该平台是一种应用对象, 包括专用于存储所述配置信息的平面结构数据单元, 以及其中, 对平面结构数据单元和 / 或关系存储单元进行操作还包括 : 响应于针对所述配置信息的网络服务请求, 通过该平台中的应用代码模块来查找相应的配置信息, 以便返回该配置信息。

11. 根据权利要求 10 所述的方法, 其中, 所述平台的关系存储单元用于存储其他应用对象的关系存储单元中的信息。

12. 根据权利要求 9 所述的方法, 其中, 所述应用对象包括数据应用对象, 其包括专用于存储业务数据的平面结构数据单元。

13. 根据权利要求 12 所述的方法, 其中, 所述数据应用对象进一步包括专用于存储所述配置信息中与之相关的公共部分的平面结构数据单元。

14. 根据权利要求 9 所述的方法, 进一步包括 : 把所述配置信息转换为能够由与所述面向服务的应用系统相关的各种客户端应用系统解释的通用格式的信息。

15. 根据权利要求 9 所述的方法, 进一步包括 : 将来自第三方系统的网络服务请求或所返回的网络服务响应转换成格式适合于所述面向服务的应用系统或者适合于与所述面向服务的应用系统相关的客户端应用系统的信息 ; 和 / 或

将所述面向服务的应用系统或者所述客户端应用系统返回的网络服务响应或者发出的网络服务请求转换成格式适合于第三方系统的信息。

16. 根据权利要求 9 所述的方法, 其中, 对平面结构数据单元和 / 或关系存储单元进行操作包括以下一种或者多种 :

向平面结构数据单元添加数据 ;

删除平面结构数据单元中的数据 ;

更新平面结构数据单元中的数据 ;

检索平面结构数据单元中的数据 ;

向关系存储单元添加信息 ;

删除关系存储单元中的信息 ;

更新关系存储单元中的信息 ; 以及

检索关系存储单元中的信息。

17. 一种用于创建面向服务的应用系统的方法, 包括 :

选择一个或多个应用模板, 其中, 所述应用模板包括用户界面、应用对象和相应动作, 该应用对象包括 :

专用于存储特定类型数据的平面结构数据单元 ;

用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元 ; 以及

用于操作该数据单元中的数据和该关系存储单元中的信息的应用代码模块 ;

针对用户的需求对所述应用模板进行配置, 以生成针对所述用户的配置信息 ; 以及

将所述应用模板和所述配置信息打包,以生成适合用户的需求的应用包。

18. 根据权利要求 17 所述的方法,进一步包括:针对所述应用包执行部署,以便为用户提供所述面向服务的应用系统。

19. 根据权利要求 17 所述的方法,进一步包括:

保存所述应用包,以便于随后创建其他应用;和/或

导出所述应用包,以供外部应用创建人员使用。

20. 根据权利要求 17 所述的方法,其中,选择一个或多个应用模板包括以下操作中的一种或者多种:

从存储库中选择用户界面、应用对象以及相应动作中的一种或者多种;

从存储库中选择预定义的应用包;以及

从外部导入预定义的应用包,

其中,所述预定义的应用包包括用于实现特定功能的多个应用模板,或者包括实现特定功能的多个应用模板以及相关的配置信息。

21. 根据权利要求 17 所述的方法,其中,对所述应用模板进行配置包括以下操作中的一种或者多种:

删除所述用户界面、所述应用对象、所述动作中至少一个;

修改所述用户界面、所述应用对象、所述动作中至少一个;

增加所述用户界面、所述应用对象、所述动作中至少一个;以及

设置网络服务请求所遵循的协议。

22. 根据权利要求 17 所述的方法,其中,所生成的应用包包括:

数据应用对象,其包括专用于存储业务数据的平面结构数据单元;和/或

平台,该平台是一种应用对象,包括专用于存储所述配置信息的平面结构数据单元。

23. 根据权利要求 17 所述的方法,其中所述配置信息包括用户界面信息和/或网络服务请求信息。

24. 一种用于创建面向服务的应用系统的创建器,包括:

选择装置,用于选择一个或多个应用模板,其中,所述应用模板包括用户界面、应用对象和相应动作,该应用对象包括:

专用于存储特定类型数据的平面结构数据单元;

用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元;以及

用于操作该数据单元的数据和关系存储单元中的信息的应用代码模块;

配置装置,用于针对用户的需求对所述应用模板进行配置,以生成针对所述用户的配置信息;以及

打包装置,将所述应用模板和所述配置信息打包,以生成适合用户的需求的应用包。

25. 根据权利要求 24 所述的创建器,进一步包括:

保存装置,用于保存所述应用包,以便于随后创建其他应用;和/或

导出装置,用于导出所述应用包,以供外部应用创建人员使用。

26. 根据权利要求 24 所述的创建器,进一步包括:部署装置,用于针对所述应用包执行部署,以便为用户提供所述面向服务的应用系统。

27. 根据权利要求 24 所述的创建器,其中,所述选择装置进一步用于以下操作中的一种或者多种:

从存储库中选择用户界面、应用对象以及相应动作中的一种或者多种;

从存储库中选择预定义的应用包;以及

从外部导入预定义的应用包,

其中,所述预定义的应用包包括用于实现特定功能的多个应用模板,或者包括实现特定功能的多个应用模板以及相关的配置信息。

28. 根据权利要求 24 所述的创建器,其中,所述配置装置进一步被用于以下操作中的一种或者多种:

删除所述用户界面、所述应用对象、所述动作中至少一个;

修改所述用户界面、所述应用对象、所述动作中至少一个;

增加所述用户界面、所述应用对象、所述动作中至少一个;以及

设置网络服务请求所遵循的协议。

29. 根据权利要求 24 所述的创建器,其中,所生成的应用包包括:

数据应用对象,其包括专用于存储业务数据的平面结构数据单元;和/或

平台,该平台是一种应用对象,包括专用于存储所述配置信息的平面结构数据单元。

30. 根据权利要求 24 所述的创建器,其中,所述配置信息包括用户界面信息和/或网络服务请求信息。

## 面向服务的应用系统及其通信方法、创建器和创建方法

### 技术领域

[0001] 本发明涉及网络应用的技术领域,更具体地,涉及一种面向服务的应用系统、该面向服务的应用系统的通信方法、用于创建该面向服务的应用系统的方法以及用于创建该面向服务的应用的创建器。

### 背景技术

[0002] 在传统的应用开发过程中,通常首先调查用户需求,然后基于用户需求来设计数据库,进行应用编程和用户界面设计。通常,出于成本效益的考虑,在开发这些应用时,总是试图处理诸如企业资源规划、客户关系管理、人力资源管理等不同业务功能中的类似需求。然而,实际上,通过这种方式开发的打包软件应用在实际应用时,难以实现其预期的效果,其主要原因如下。

[0003] 首先,各种业务信息的处理方式不同,难以实现统一化的处理。对于各种业务而言,不管信息是来自不同工艺领域还是来自相同工业领域,对这些信息进行处理的方式通常是不同的,这是因为业务环境和业务实践方面存在巨大差异。

[0004] 其次,软件应用过于繁杂。以上述方式形成的打包软件试图概括出各种不同的业务功能的共同需求。然而,随着软件的使用,软件应用逐步升级,最终导致大多数业务实际上并不需要的功能组件越来越多。

[0005] 再者,存在数据整合问题。针对各种业务开发的打包软件通常是按照不同的概念并针对不同功能设计和开发的,它们彼此之间通常不能通信。需要与多种业务相关的应用的用户(诸如公司或企业等)通常需要购买多种打包软件,而由于这些打包软件之间不能通信,因此对于拥有多个这种打包软件的用户而言,在这些软件之间会形成多个信息孤岛。当公司拥有这样的打包软件时,通常需要花费大量的人力、物力来维护各个软件相关的信息,并且这些信息之间的完整性、一致性也难以得到保证,而且还存在大量冗余。

[0006] 另外,在软件使用过程中不能以具有成本效益的方式来满足用户变化的需求。业务信息通常是易变的,且具有内部关联性,但是基于预定的、刚性数据模型设计的打包软件,不能在使用过程中以期望的、适合的方式来关联这些变化的信息。因此,根据现有技术,难以以较小的成本来满足用户的需求。

### 发明内容

[0007] 有鉴于此,本发明旨在至少部分上解决上述提及的现有技术中开发应用以及使用中的应用所存在的问题。

[0008] 为此,本发明提供了一种面向服务的应用系统、面向服务的应用系统的通信方法、创建面向服务的应用系统的方法和创建器。

[0009] 根据本发明的一个方面,提供了一种面向服务的应用系统,所述应用系统包括一个或多个应用对象,每个应用对象包括:专用于存储特定类型数据的平面结构数据单元;用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单

元；以及用于操作该数据单元中的数据 and 该关系存储单元中的信息的应用代码模块，其中，所述应用代码模块用于响应于网络服务请求，对平面结构数据单元和 / 或关系存储单元进行操作。

[0010] 根据本发明的一个实施方式，所述应用对象包括平台，该平台是一种应用对象，包括专用于存储所述配置信息的平面结构数据单元，以及其中，该平台中的应用代码模块用于响应于针对所述配置信息的网络服务请求，查找相应的配置信息，以便返回该配置信息。

[0011] 根据本发明的另一实施方式，所述平台的关系存储单元用于存储其他应用对象的关系存储单元中的信息。

[0012] 根据本发明的又一实施方式，所述应用对象包括数据应用对象，其包括专用于存储业务数据的平面结构数据单元。

[0013] 根据本发明的再一实施方式，所述数据应用对象进一步包括专用于存储所述配置信息中与之相关的公共部分的平面结构数据单元。

[0014] 根据本发明的另一实施方式，其中所述应用进一步包括中间件，用于把所述配置信息转换为能够由与所述面向服务的应用系统相关的各种客户端应用系统解释的通用格式的信息。

[0015] 根据本发明的又一实施方式，其中所述应用进一步包括适配器，用于将来自第三方系统的网络服务请求或所返回的网络服务响应转换成格式适合于所述面向服务的应用系统或者适合于与所述面向服务的应用系统相关的客户端应用系统的信息；和 / 或将所述面向服务的应用系统或者所述客户端应用系统返回的网络服务响应或者发出的网络服务请求转换成格式适合于第三方系统的信息。

[0016] 根据本发明的再一实施方式，其中，所述应用代码模块用于进行以下操作中的一种或者多种：向平面结构数据单元添加数据；删除平面结构数据单元中的数据；更新平面结构数据单元中的数据；检索平面结构数据单元中的数据；向关系存储单元添加信息；删除关系存储单元中的信息；更新关系存储单元中的信息；以及检索关系存储单元中的信息。

[0017] 根据本发明的另一方面，提供了一种面向服务的应用系统的通信方法，其中，所述面向服务的应用系统包括一个或多个应用对象，每个应用对象包括专用于存储特定类型数据的平面结构数据单元、用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元、以及用于操作该数据单元中的数据 and 该关系存储单元中的信息的应用代码模块，所述方法包括：响应于网络服务请求，通过应用代码模块对平面结构数据单元和 / 或关系存储单元进行操作。

[0018] 根据本发明的又一方面，提供了一种用于创建面向服务的应用系统的方法，包括：选择一个或多个应用模板，其中，所述应用模板包括用户界面、应用对象和相应动作，该应用对象包括：专用于存储特定类型数据的平面结构数据单元、用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元、以及用于操作该数据单元中的数据 and 该关系存储单元中的信息的应用代码模块；针对用户的需求对所述应用模板进行配置，以生成针对所述用户的配置信息；以及将所述应用模板和所述配置信息打包，以生成适合用户的需求的应用包。

[0019] 根据本发明的一个实施方式，所述方法进一步包括：针对所述应用包执行部署，以

便为用户提供所述面向服务的应用系统。

[0020] 根据本发明的另一实施方式,所述方法进一步包括:保存所述应用包,以便于随后创建其他应用;和/或导出所述应用包,以供外部应用创建人员使用。

[0021] 根据本发明的又一实施方式,其中,选择一个或多个应用模板包括以下操作中的一种或者多种:从存储库中选择用户界面、应用对象以及相应动作中的一种或者多种;从存储库中选择预定义的应用包;以及从外部导入预定义的应用包,其中,所述预定义的应用包包括用于实现特定功能的多个应用模板,或者包括实现特定功能的多个应用模板以及相关的配置信息。

[0022] 根据本发明的再一实施方式,其中,对所述应用模板进行配置包括以下操作中的一种或者多种:删除所述用户界面、所述应用对象、所述动作中至少一个;修改所述用户界面、所述应用对象、所述动作中至少一个;增加所述用户界面、所述应用对象、所述动作中至少一个;以及设置网络服务请求所遵循的协议。

[0023] 根据本发明的另一实施方式,其中,所生成的应用包包括:数据应用对象,其包括专用于存储业务数据的平面结构数据单元;和/或平台,该平台是一种应用对象,包括专用于存储所述配置信息的平面结构数据单元。

[0024] 根据本发明的又一实施方式,其中,所述配置信息包括用户界面信息和/或网络服务请求信息。

[0025] 根据本发明的又一方面,还提供了一种用于创建面向服务的应用系统的创建器,包括:选择装置,用于选择一个或多个应用模板,其中,所述应用模板包括用户界面、应用对象和相应动作,该应用对象包括:专用于存储特定类型数据的平面结构数据单元、用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元、以及用于操作该数据单元的数据和关系存储单元中的信息的应用代码模块;配置装置,用于针对用户的需求对所述应用模板进行配置,以生成针对所述用户的配置信息;以及打包装置,将所述应用模板和所述配置信息打包,以生成适合用户的需求的应用包。

[0026] 根据本发明,该面向服务的应用系统主要由多个应用对象构成,并且优选地包括用于存储配置信息的称作“平台”的应用对象。因此,基于这种结构,可以通过预先定义的应用模板来以快速、简单的方式(通过点击、拖放等简单的鼠标操作)来创建该面向服务的应用系统。

[0027] 另外,可以对选择的预定义打包文件进行修改以满足特定用户的需求,也可以通过略微修改原先为其他用户创建的应用包来为另一用户创建类似的应用。此外,还可以容易地修改已有的面向服务的应用系统来适应客户新的需求。

[0028] 根据本发明,相同类型的业务数据(诸如联系人信息)存储在逻辑上为单个表或者单个文件的同一平面结构数据单元中,但该平面结构数据单元、该应用对象甚至整个网络应用系统均可以位于网络中的多个位置,换句话讲,可以位于多个物理位置或者多个域中。根据本发明,可以根据业务需求将各个平面结构数据单元、各个应用对象或整个网络应用布置在单个位置或者多个位置,或布置在单个域中或者多个域中。例如,对于小型公司,可以将它们存储在单个位置,对于大型的国际性公司,则可以将其提供在多个位置。另外,根据本发明,存储在该单一逻辑表中的业务数据可以由针对不同业务的多个应用共享,这与利用不同概念针对不同业务分别开发打包软件传统方式不同,根据该传统方式,打包软



件是根据预定的、刚性数据模型基于不同理念针对不同业务而设计的,这些打包软件之间通常是不能通信的,更不能共享业务数据。所以,与传统方式相比,在本发明的网络应用系统中,业务数据不会出现信息孤岛。因此,不会存在信息整合、数据完整性等方面的问题,使得用户从维护数据完整性等繁杂的工作中解脱出来,减少了成本,提供了效率。

[0029] 另外,根据本发明,不同的客户端应用系统(例如,用户的客户端应用系统、该用户的提供商的客户端应用系统、该用户的客户的客户端应用系统和该用户的合作方的客户端应用系统)可以对应于一个面向服务的应用系统中的不同应用对象子集,这例如可以通过装载不同的配置信息来实现。因而,可以实现用户的业务数据与其提供商、客户、合作方等的适当共享。因此,本发明不但为用户的公司带来了诸多益处,而且为用户的整个公司链带来了方便。

[0030] 根据本发明的优选实施方式,还可以通过适配器使得本发明的网络应用系统中的客户端应用系统和/或面向服务的应用系统与第三方系统的相互通信。从而,可以将用户已有的业务系统与本发明的网络应用系统结合起来使用,从而节约成本。另外,通过使用适配器可以使得本发明的网络应用系统可以适应于任何变化,使得本发明的网络应用系统可以利用诸如 REST 方式或者 SOAP 协议等适当方式与第三方系统进行数据交换,因此可以进一步避免信息孤岛的问题。

#### 附图说明

[0031] 通过对结合附图所示出的实施方式进行详细说明,本发明的上述以及其他特征将更加明显,在本发明附图中,相同的标号表示相同或相似的部件。在附图中,

[0032] 图 1 示意性地示出了根据本发明的一个实施方式的网络应用系统;

[0033] 图 2a 至图 2d 示意性地示出了根据本发明的应用对象的示例;

[0034] 图 3 示意性地示出用于说明根据本发明的关系存储单元的图示;

[0035] 图 4 示意性地示出了根据本发明的一个实施方式的网络应用系统的通信流程图;

[0036] 图 5 示意性地示出了根据本发明的另一实施方式的网络应用系统;

[0037] 图 6 示意性地示出了根据本发明的又一实施方式的网络应用系统;

[0038] 图 7 分别示出了第三方系统返回的 XML 文件和本发明的网络应用系统所能力理解的文件格式;

[0039] 图 8 分别示出了第三方系统和本发明的客户端应用系统的网络服务请求和响应所使用的协议的示例性格式;

[0040] 图 9 示意性地示出了根据本发明的再一实施方式的网络应用系统;

[0041] 图 10 示意性地示出了根据本发明的又一实施方式的网络应用系统;

[0042] 图 11 示意性地示出了根据本发明的一个实施方式的用于创建面向服务的应用系统的方法的流程图;

[0043] 图 12 示意性地示出了根据本发明的一个实施方式的用于创建面向服务的应用系统的创建器的方框图;

[0044] 图 13 示意性地示出了根据本发明的另一实施方式的用于创建面向服务的应用系统的创建器的方框图;

[0045] 图 14 示意性地示出了根据本发明的又一实施方式的用于创建面向服务的应用系

统的创建器的方框图；

[0046] 图 15 示意性地示出了根据本发明的再一实施方式的用于创建面向服务的应用系统的创建器的方框图；以及

[0047] 图 16 示意性地示出了可以实现根据本发明的实施方式的计算设备的结构方框图。

### 具体实施方式

[0048] 接着,将参考附图通过实施方式来描述本发明提供的面向服务的应用系统、该面向服务的应用系统的通信方法、用于创建面向服务的应用系统的方法及用于创建面向服务的应用系统的创建器。

[0049] 首先,为了更加清楚和方便地描述根据本发明的面向服务的应用系统及通信方法,将通过参考图 1 至图 10 描述根据本发明的实施例的网络应用系统及其通信方法,来描述面向服务的应用系统及其通信方法。

[0050] 参考图 1,图 1 示意性地示出了根据本发明的一个实施方式的网络应用系统。如图 1 所示,网络应用系统 100 包括客户端应用系统 110、面向服务的应用系统 120,其中,该面向服务的应用系统 120 包括应用对象 130a、130b... 130n。客户端应用系统 110 与面向服务的应用 120 之间通过网络 140 通信。

[0051] 网络 140 可以是任何类型的网络,或者是由能够彼此通信的设备组成的网络的组合。例如,网络 140 可以是传输控制协议 / 网际协议 (TCP/IP) 网络,诸如统称被称作是“因特网”的全球互联网。

[0052] 客户端应用系统 110 可以是能够操控直接或间接运行在任何操作系统上的内容和布局的任何网络客户端应用。该网络客户端应用 110 可以是利用或者不利用网络浏览器来运行的富互联网应用 (RIA),例如 Flash 应用、Java 应用、JavaFX 应用、Adobe Flex/AIR 应用、Microsoft Silverlight 应用等,其中富互联网应用可以运行专用的网络浏览器插件上,或者经由保证程序安全运行的砂盒 (sandbox) 来独立运行;或者,可以是例如 Javascript 应用、VBScript 应用等在网络浏览器上运行的脚本应用;或者,可以是以例如 .NET、Java、C、C++、Ruby、Python 等各种编程技术开发的直接在操作系统上运行的其他应用。并且该网络客户端应用可以任何基于网络的应用,其基于 Windows 操作系统,基于 Linux 系统,或者基于其他任何适当的操作系统。该客户端应用系统 110 甚至可以链接到任何适当站点 (例如公司网站、博客或者是阿里巴巴或易趣等大商业网站等) 中的小应用,以方便使用。例如,用户可以将其产品目录以及在线查询和支付“挂”到它们的合作方网站上以方便使用。

[0053] 该客户端应用系统 110 被实现为向面向服务的应用系统 120 发出网络服务请求,诸如请求面向服务的应用系统 120 中相应的应用对象以添加数据、删除数据、检索数据、更新数据等,以及接收来自应用对象的网络服务响应,诸如返回的检索数据等。

[0054] 客户端应用系统 110 可以被配置为通过装载配置信息而开始操作。该配置信息决定了客户端应用系统 110 如何与用户交互以及如何发起网络服务请求。关于配置信息将在下文中详细描述。

[0055] 客户端应用系统 110 可以运行在任何设备或者系统上,只要该设备或系统能够支持该客户端应用系统与面向服务的应用系统之间的通信。该设备或者系统可以是个人计算

机,例如台式计算机;手持计算设备,诸如像是 iPhone、黑莓、安装了 Window Mobile 系统的手机、智能手机等的移动电话;便携式计算设备,诸如笔记本电脑、上网本电脑;以及工作站、计算设备网络等。

[0056] 面向服务的应用系统 120 包括一个或多个应用对象 130a、130b... 130n。在本发明中,“应用对象”也可以称作是“构件块”服务器应用,可以通过这些“构建块”服务器应用的不同组合来构建针对不同功能需求的解决方案。下面,将参考图 2a 至图 2d 来详细描述用于构成面向服务的应用系统 120 的应用对象。

[0057] 参考图 2,图 2a 至图 2d 示意性地示出了根据本发明的应用对象的示例。如图 2a 所示,应用对象 200 可以包括平面结构数据单元 201、关系存储单元 202 以及应用代码模块 203。其中平面结构数据单元 201 可以专用于存储特定类型数据,诸如联系人信息、订单信息、项目信息、与应用相关的配置信息等中的一种;关系存储单元 202 可以用于存储该平面结构数据单元 201 中的数据与其他应用对象中平面结构数据单元中的数据之间的关系;而应用代码模块 203 可以用于操作该平面结构数据单元 201 中的数据以及关系存储单元 202 中的信息,诸如向平面结构数据单元 201 和 / 或关系存储单元 202 中添加数据等。

[0058] 如图 2b 所示,应用对象 200 可以运行在支持系统 210 上,该支持系统 210 是指以下特征的组合:允许应用对象 200 在其上运行的任何平台,例如,操作系统;允许关系存储单元 202 和平面结构数据单元 201 驻留在其上的任何数据库管理系统,例如 RDBMS, Oracle, DB2, MySQL, MS SQL 等;允许利用诸如 JavaScript、VBScript、.NET、Java、PHP、C、C++、Ruby、Python 等各种编程技术开发的应用代码模块在其上运行的应用框架;以及允许任何平台、数据库管理系统以及应用框架运行的设备或者系统。

[0059] 应用对象 200 所包括的平面结构数据单元 201 可以是诸如简单的 XML 文件,或者是诸如 MySQL、MS SQL、Oracle、DB2 等关系数据库中的单个逻辑表结构(尽管在物理上该平面结构数据单元可以通过在不同的位置储存的表来实现)。根据本发明,每个应用对象中的平面结构数据单元 201 专用于存储一种特定类型的数据,即,不同类型的数据存储在不同应用对象的平面结构数据单元中,且不同应用对象的平面结构数据单元中存储的数据类型互不相同。此外,与传统的关系数据库还不同的是,平面结构数据单元 201 中存储的数据与其他应用对象的平面结构数据单元中存储的数据之间的关系并不是通过诸如在表中加入诸如外键来表示,而是通过应用对象 200 中包括的关系存储单元 202 来表示,该关系存储单元将在后文中详细描述。

[0060] 需要说明的是,每个应用对象 200 可以具有相同结构,即,每个应用对象均可包括平面结构数据单元 201、关系存储单元 202 和应用代码模块 203。但是,各个平面结构数据单元 201 可以具有不同内部结构,例如,单个逻辑表形式的平面结构数据单元 201 可以具有不同类型和不同数量的字段,以便存储不同类型的数据,诸如联系人信息、订单信息、项目信息、与面向服务的应用系统对应的客户端应用系统的配置信息等。

[0061] 如图 2c 所示,应用对象 200a 中的平面结构数据单元 201a 专用于存储客户端应用系统的配置信息,其中,该配置信息可以供客户端应用系统获取以便启动客户端应用系统。因此,在这种意义上讲,也可以将该应用对象 200a 称作为“平台”。而如图 2d 所示,应用对象 200b 中的平面结构数据单元 201b 专用于存储普通的业务数据时,充当为客户端应用系统提供业务数据服务的功能,因此也可以将其称为“数据应用对象”。

[0062] 应用对象 200 中包括的关系存储单元 202 在本发明中起到了表示各个应用对象中平面结构数据单元的数据之间的关系的的作用。下面,将参考图 3 对本发明的关系存储单元进行更加详细的描述。

[0063] 参考图 3,图 3 示出了用于说明根据本发明的关系存储单元的图示,其中关系存储单元是链接表。在图 3 中,示出了数据应用对象 310 中包括的平面结构数据单元 311 和的链接表 312、数据应用对象 320 中包括的平面结构数据单元 321 和链接表 322 以及平台 330 中的链接表 332。需要说明的是,为了清楚起见,没有示出数据应用对象 310 和 320 中的应用代码模块以及平台中的平面结构数据单元和应用代码模块。

[0064] 如图所示的链接表 312 和 322 分别用于存储平面结构数据单元 311 和 321 与其他应用对象中的平面结构数据单元之间的链接关系。根据本发明的实施方式,如果一个平面结构数据单元的数据与其他应用对象中平面结构数据单元中的数据具有链接关系,则在与这两个平面结构数据单元对应的链接表中将会具有用于表示该链接关系的链接记录。另外,优选地,在平台 330 的中心链接表 332 中也保存与之相应的链接记录,利用该链接记录,可以在数据应用对象中的链接记录丢失时进行记录恢复。

[0065] 例如,如果平面结构数据单元 311 中的数据记录 A 与平面结构数据单元 321 中的数据记录 B 链接,则在与平面结构数据单元 311 对应的链接表 312 中则存在这样的链接记录,其包括作为主键的被链接记录 A 的标识、其中记录 B 被链接的数据应用对象 320 的标识以及被链接记录 B 的标识;与平面结构数据单元 321 对应的链接表 322 中,也会存在相应的链接记录,其包括作为主键的被链接记录 B 的标识、其中记录 A 被链接的数据应用对象 310 的标识以及被链接记录 A 的标识;以及优选地,在平台 330 中的中心链接表 332 中也可以存储对应的链接记录,该链接记录包括其中记录 A 被链接的数据应用对象 310 的标识、被链接记录 A 的标识、其中记录 B 被链接的数据应用对象 320 的标识以及被链接记录 B 的标识。

[0066] 在需要将两个记录链接起来时,可以通过分别在与这两个平面结构数据单元对应的两个链接表中存储这样的两条记录、以及优选地还在平台的中心链接表中加入如上所述的链接记录,来建立两个记录的连接关系,从而将两条记录链接起来。

[0067] 这样,如果用户想通过客户端应用系统查看与某条记录相关的链接项目,则可在与存储该条记录的平面结构数据单元对应的链接表中,查找主键为该条记录的标识的所有链接记录。然后,可以根据查找到的链接记录中包括的、其中记录被链接的应用对象的标识,向该应用对象发出网络服务请求,以便获取被链接的记录,该被链接记录具有的标识是查找到链接记录中所包括的被链接记录的标识。

[0068] 需要说明的是,链接表只是用于存储各个平面结构数据单元的数据之间关系的关系存储单元的一个实例。本发明并不局限于此,也可以采用其他任何适当的技术来存储这些链接记录,诸如可以采用像是 XML 文件等文件形式,只要能够保存下这些链接信息即可。关于实现链接记录存储的其他技术相关的细节,对于本领域技术人员是已知的,此处不再赘述。

[0069] 接下来,将继续参考图 2 来描述根据本发明的应用对象。

[0070] 返回图 2,应用代码模块 203 配置用于响应于诸如来自客户端应用系统或者其他应用对象的网络服务请求(例如,REST 方式或者 SOAP 协议的请求),来操作平面结构数据单元 201 中的数据和/或诸如链接表的关系存储单元 202 中所存储的信息的模块。可以

根据实现所述支持系统 210 的技术,利用诸如 .NET、JAVA、PHP、C、C++、Ruby, Python 等的适当技术来开发应用代码模块 203。该应用代码模块 203 可以包括对数据进行基本操作的基本方法,诸如创建、检索、更新和删除平面结构数据单元中的数据和链接表中的链接信息等,但并不局限于此。并且,这些基本方法可以被设计为支持诸如 SOAP 协议和 / 或 REST 方式等任何格式的网络服务请求,以便响应于网络服务请求对平面结构数据单元中的数据或关系存储单元中的信息进行操作。在该面向服务的应用系统包括平台的情况下,可以将平台中的应用代码模块配置为响应于针对所述配置信息的网络服务请求,查找相应的配置信息,以便返回该配置信息。

[0071] 需要说明,对于采用不同技术的平面结构数据单元,所述基本方法的实现也不同,诸如对于 XML 文件形式的平面结构数据单元,基本方法中包括诸如 Xquery 等 XML 查询语言,而对于关系数据库中的表形式的平面结构数据单元,基本方法可以包括结构化查询语言 (SQL),这是本领域技术人员公知的,此处将不再赘述。

[0072] 另外,需要说明的是,对于所有应用对象而言,所有的应用代码模块可以都是完全相同的,这样有助于加速部署过程。然而,本发明并不局限于此,而是可以针对各个应用对象配置不同的应用代码模块。

[0073] 此外,还需要说明的是,根据本发明的应用对象(即平台或者数据应用对象)中的每一个均可以位于一台或者多台设备上,并且一个或者多个应用对象也可以位于单个设备上。

[0074] 在上文中,对根据本发明的一个实施方式的网络应用系统进行了描述,下面将参考图 4 来描述根据本发明的网络应用系统的通信方法。

[0075] 参考图 4,首先,在步骤 401 发出针对配置信息的请求,以便启动该客户端应用系统。

[0076] 根据本发明的实施方式,客户端应用系统被配置成通过载入配置信息来启动。该配置信息以包括用户界面信息和 / 或网络服务请求信息。用户界面信息决定了客户端应用系统如何与用户交互,即应该向用户呈现什么样的界面。网络服务请求信息决定了如何发起网络服务请求,即应该以什么上的协议发起请求,应该发起什么请求。对于不同的用户,配置信息可以是不同的,对于不同类型的用户设备也可以具有不同的配置信息,但是本发明并不局限于此。

[0077] 该请求应该具有用于存储配置信息的实体可以解释的格式,诸如可以是支持 SOAP 协议和 / 或 REST 格式等任何格式的网络服务请求,或者其他格式的请求。

[0078] 接着,在步骤 402,响应于针对配置信息的请求,返回包含该配置信息的响应。

[0079] 在本发明的一个实施方式中,配置信息存储在面向服务的应用系统中的平台 130a 的平面结构数据单元中。平台 130a 中的应用代码模块响应于该请求,查找所请求的配置信息,并将返回所找到的配置信息。

[0080] 然后,在步骤 403,基于所述配置信息启动客户端应用系统。根据所接收到的配置信息中的用户界面信息,客户端应用系统被启动,并为用户显示相应的用户界面,以便进行交互。

[0081] 接下来,在步骤 404,客户端应用系统发出网络服务请求。

[0082] 在用户通过客户端应用系统所呈现的用户界面进行交互的过程中,可以通过触发

用户界面上的组件,来发出网络服务请求。该网络服务请求可以基于配置信息中的网络服务请求信息而发出,其是与该组件对应的、具有网络服务请求信息中所定义的格式的网络服务请求。

[0083] 然后,在步骤 405,响应于该网络服务请求,通过应用代码模块来执行相应操作,以便返回相应的网络服务响应。应用代码模块被配置为能够理解该网络服务请求,然后根据所要求的网络服务,对平面结构数据单元和 / 或诸如链接表的关系存储单元执行基本操作,诸如创建、删除、查询和更新等。然后,可以返回与该网络服务请求对应的网络服务响应。

[0084] 需要说明的是,尽管对于本发明而言,优选地是在平台中存储配置信息,然而本发明并不局限于此,而是可以在网络中的其他位置存储该配置信息,只要该信息能够由客户端应用系统来获取,该配置信息甚至还可以存储在客户端应用系统所在的用户设备中。

[0085] 另外,根据本发明,可以在一个网络请求中对一个数据应用对象执行操作,例如,向其添加数据。然而,也可以对若干数据应用对象执行多个操作,例如可以在添加数据的同时建立连接关系。

[0086] 接下来,将参考图 5 来描述根据本发明的另一实施方式的网络应用系统。

[0087] 参考图 5,图 5 示出了根据本发明的另一实施方式的网络应用系统 500。在图 5 所示的网络应用系统中,包括在诸如台式计算机、膝上型计算和手持设备等各种不同硬件设备上运行的不同客户端应用系统 510a、...、510m。在面向服务的应用系统 520 中的平台 530a 中,存储了与各个客户端应用系统 510a、...、510m 对应的配置信息。在客户端应用系统 510 启动之前,客户端应用系统会向平台 530a 发送针对配置信息的网络服务请求,平台 530a 会响应于该网络服务请求,向该客户端返回与之相应的配置信息。

[0088] 然而,由于硬件设备的限制,通常在诸如手持设备上的用户设备中需安装简化版的客户端系统,而此时可能会出现不能解释来自平台 530a 的配置信息等问题。为此,在面向服务的应用系统 520 中,还包括有中间件 550,用于允许在不同硬件设备上运行的客户端应用系统 510a、... 510m 可以访问本发明的面向服务的应用系统 520。中间件 550 与应用对象类似,可以运行在前述的支持系统 210 上,并且与应用对象类似,中间件也 550 可以根据实现所述支持系统 210 的技术,利用诸如 .NET、JAVA、PHP、C、C++、Ruby、Python 等的适当技术来开发。

[0089] 在客户端应用系统不能解释配置信息时,可以请求中间件 550 的辅助,中间件可以将所述配置信息转换为客户端应用系统能够理解格式的信息,诸如在客户端应用系统利用浏览器来运行时,可以转换为通用 HTML 标签。在根据本发明的另一实施方式中,在客户端应用系统不能解释配置信息时,可以向平台 530a 发出需要执行格式转换的网络服务请求,或者在客户端应用系统发出针对配置信息的网络请求中要求执行格式转换。平台 530a 会将配置信息发送给中间件 550,以便进行格式转换。然后,中间件 550 可以将经过格式转换的配置信息发送到相应的客户端应用系统,或者平台通知相应的客户端应用系统从中间件 550 处获取配置信息。在本发明的又一实施方式中,客户端应用系统可以直接向中间件 550 请求配置信息;中间件 550 基于该配置请求与平台 530a 通信,并从其获取相应的配置信息;然后,中间件 550 对所获取的配置信息执行数据格式转换,并将经过格式转换的配置信息发送给客户端应用系统。

[0090] 在上述实施例中,中间件 550 是一个单独部件,然而需要说明的是,中间件 550 也可以与平台 530a 集成在一起。这样,客户端应用系统直接向平台 530a 请求配置,如果需要转换则可以由包括在平台 530a 中的中间件进行格式转换,不需要则可以绕过该中间件经由网络 540 将配置信息直接发送给客户端应用系统。

[0091] 此外,图 6 还示出了根据本发明的又一实施方式的网络应用系统 600。在图 6 示出的网络应用系统 600 中,除了客户端应用系统和相应的面向服务的应用系统外,还存在第三方系统 680。该第三方系统例如是与客户端应用系统的用户属于共同体的其他方(诸如该公司的客户、供应商、合作方等)的系统,或者该公司中的其他部门的系统等。第三方系统通常是与本发明的系统不同的应用,因此它们所采用的网络服务请求很可能与本发明的基于平面结构数据单元的面向服务的应用系统不同。第三方系统的网络服务请求以及他们所能解释的响应通常是具有结构层次的,而由于本发明是基于平面结构数据单元,所以本发明的网络服务请求和返回的网络服务响应均是无层次。如图 7 所示,XMLm 701 例如可以是第三方所返回的响应,该文件中的数据信息是具有层次结构关系的,而与之对应的 XML1702a、XML2702b 和中心链接 702c 中的每个均是无结构层次的文件,即平面 XML 文件,XML1702a 和 XML2702b 之间的关系(层次结构)通过中心链接 702c 来表示。因此,这两种格式的信息通常是不能为彼此所理解的。另外,在第三方系统与客户端应用系统通信时,即使他们的网络服务请求和响应同样是基于本发明的平面结构数据单元,但他们所使用的协议也可能不同。图 8 以示例性的方式示出了这种情况,例如客户端应用系统所使用的网路服务请求是如 801 所示的 REST 方式的请求,而第三方系统所使用的网络服务请求是如 802 所示的 SOAP 协议的请求。因此,在这种情况下,这也可能造成第三方系统与本发明的网络应用系统彼此之间的通信障碍。

[0092] 因此,为了使得第三方能够访问该面向服务的应用系统,本发明提供了一种网络服务适配器,将来自第三方系统的网络服务请求或所返回的网络服务响应转换成格式适合于所述面向服务的应用系统或者适合于与所述面向服务的应用系统对应的客户端应用系统的信息;和/或将所述面向服务的应用系统或者所述客户端应用系统返回的网络服务响应或者发出的网络服务请求转换成格式适合于第三方系统的信息。例如,可以将 REST 方式的请求转换为能够被理解的 SOAP 协议的请求,将本发明的系统返回的无层次的 XML 文件整合为第三系统能够理解的具有层级结构的 XML 文件,反之亦然。

[0093] 这样,就使得第三方能够与客户端应用系统和/或数据应用对象通信。在用户已经拥有了一些业务系统的情况下,通过网络服务适配器可以将已有的这些业务系统与本发明的网络应用系统整合起来,从而节约了用户公司的资源,避免了重复投资造成的浪费。

[0094] 需要说明的是,该适配器可以如图适配器 670a 所示,作为一个单独的部件而存在;也可以如图适配器 670b 所示打包到任何面向服务的应用系统中。与应用对象和中间件类似,该适配器 670a 和 670b 可以运行在前述的支持系统 210 上,该适配器也可以根据实现所述支持系统 210 的技术,利用诸如 .NET、JAVA、PHP、C、C++、Ruby、Python 等的适当技术来开发。关于网路服务适配器的具体信息,可参照与本申请同日提交的申请人为李新宇、申请号为“”、发明名称为“用于对网络服务通信进行适配的方法、适配器和适配器系统”的专利申请。

[0095] 此外,图 9 还示出了根据本发明的再一实施方式的网络应用系统。根据如图 9 所

示的实施方式,并非将所有的配置信息都集中保存在平台 930a 的数据单元中,而是可以仅在平台 930a 中保存对于客户端应用系统特定的配置信息,而对于所有客户端应用系统都具有的公共配置信息,可以在相应的数据应用对象 930b...930n 中建立一个专用于存储公共配置信息的平面结构数据单元,以便存储与之相关的配置信息,如图 9 所示。

[0096] 此外,图 10 还示出了根据本发明的又一实施方式的网络应用系统。在图 10 所示的网络应用系统中,存在第一客户端应用系统 1010a1、用于存储与该第一客户端应用系统 1010a1 相关的配置信息的第一平台 1030a1 以及与第一客户端应用系统 1010a1 对应的第一面向服务的应用系统 1050a1;以及第二客户端应用系统 1010a2、存储与该第二客户端应用系统 1010a2 相关的配置信息的第二平台 1030a2 以及与第二客户端应用系统 1010a2 相关的第二面向服务的应用系统 1050a1。第一客户端应用系统和第二客户端应用系统可以是属于共同体的各方,例如某公司与该公司的客户、供应商或合作方等。在根据本发明的网络应用系统中,第一客户端应用系统 1010a1 可以通过网络 1040 来向面向服务的应用系统 1050a2 中的应用对象发出网络服务请求,面向服务的应用系统 1050a2 中的应用对象可以响应于该网络服务请求为第一客户端应用系统 1010a1 提供相应的网络服务响应,同样第二客户端应用系统 1010a2 可以通过网络 1040 访问第一面向服务的应用系统 1050a1 中的应用对象。因此,可以在这样的网络应用系统中实现数据共享和数据交换。

[0097] 上面参考图 1 至图 10 以示例性的方式对本发明的网络应用系统及其通信方法的若干实施方式进行了详细描述。然而,需要说明的是,当客户端应用系统和面向服务的应用系统运行在同一台设备上时,实际上可以不存在网络 140、540、640、940 和 1040。

[0098] 另外,需要说明的是,虽然在某些实施方式中示出了一个客户端应用系统,实际上可以存在与相同的面向服务的应用系统的若干个客户端应用系统。该网络应用系统也可以包括分别与多个客户端应用系统对应的多个不同的面向服务的应用系统,并且可以存在分别与多个客户端对应的多个平台。

[0099] 还需要说明的是,客户端应用系统和面向服务的应用系统也可以运行该网络中的一个或多个设备上;在网络 140、540、640、940 和 1040 是因特网的情况下,客户端应用系统和面向服务的应用系统还可以运行在一个或多个域中。

[0100] 另外,需要说明的是,驻留在用户设备(或系统)中的所有客户端应用系统可以是依据支持系统的技术而开发的通用应用,通过装载不同的配置信息,这些客户端应用系统可以满足用户的不同需求。但是本发明并不局限于此,所有客户端应用系统也可以共用相同的配置信息。

[0101] 上文中,通过描述本发明的网络应用系统及其通信方法,来描述了本发明的面向服务的应用系统和客户端应用系统及他们的通信方法。然而,需要说明的是,本发明的各个面向服务的应用系统和各个客户端应用系统可以由一个软件提供商提供,也可能由分开的多个软件提供商来提供,只要面向服务的应用系统和客户端应用系统之间遵循相同的通讯协议。因此,本发明分开保护了根据本发明的面向服务的应用系统和客户端应用系统。

[0102] 在下文中,将参考图 11 来描述发明提供的用于创建面向服务的应用系统的方法和创建器。

[0103] 参考图 11,图 11 示出了根据本发明的一个实施方式的用于创建面向服务的应用系统的方法流程图。



[0104] 如图 11 所示,在步骤 1101,选择一个或多个应用模板。

[0105] 根据本发明的实施方式,提供了一个或多个应用模板以加速应用开发过程。该应用模板包括设计好的用户界面、应用对象和相应动作。用户界面是将通过例如浏览器以页面形式呈现给用户以便与用户进行交互的界面,其上有可以提供给用户的各种组件。动作是与用户界面上提供的各种组件(按钮、选择框、文本框等)相关联的操作,是响应于用户界面上组件的激活而执行的动作,诸如显示对应页面,发出对数据进行操作的网路服务请求等。

[0106] 应用对象如上文所述包括:专用于存储特定类型数据的平面结构数据单元;用于存储该平面结构数据单元中的数据与其他应用对象中平面结构数据单元的数据之间关系的关系存储单元;和用于操作该平面结构数据单元的数据以及关系存储单元中的信息的应用代码模块。应用对象通常是网络服务请求所针对的对象。其中,在应用对象用于存储应用配置信息时,也可以称作为平台;用于存储普通的数据信息时,可以称其为数据应用对象。

[0107] 在根据本发明的一个实施方式中,提供有存储库,该存储库中用于存储应用模板。因此,可以从存储库中选择用户界面、应用对象以及相应动作中的一种或者多种。

[0108] 在根据本发明的另一实施方式中,应用模板是应用包的形式,该应用包包括用于实现特定功能的多个应用模板。优选地,应用包中还进一步包括相关的配置信息。在这种情况下,如果要创建与该应用包所实现的功能类似的面向服务的应用系统,则可以从存储库中选择该应用包,以便进一步加速面向服务的应用系统的创建过程。

[0109] 在根据本发明的又一实施方式中,可以通过导入部件从外部导入该应用包,以便加速面向服务的应用系统开发过程。

[0110] 接下来,在步骤 1102,针对用户的需求对所述应用模板进行配置,以生成针对所述用户的配置信息。

[0111] 对于不同的用户而言,所需的面向服务的应用系统可能不同,因此可能需要对所选的应用模板进行配置,即对这些应用模板进行调整以便适合用户需要。

[0112] 可以对所选择应用模板中包括的用户界面、应用对象和动作等进行修改。例如,可以在用户界面上添加组件以及与之对应的动作等;添加应用对象;修改应用对象之间的链接关系;修改平面结构数据单元的内部结构,诸如增删字段等。此外,还可以修改用户界面上的呈现的图标、图片等,以便创建个性化的应用。

[0113] 例如,如果选择的是应用包,则可以删除该应用包中不需要的用户界面、应用对象和相关动作;另外,也增加没有包括在该应用包中的用户界面、应用对象和相关动作,从而满足用户的各种特定需求。

[0114] 此外,还可以指定网络服务请求所使用的相关协议,诸如 REST 方式或者 SOAP 协议,图 8 中示例性地示出了基于这两种协议的请求和响应。

[0115] 对应用模板进行了配置后,可以得到相关的配置信息,该配置信息包括与要呈现给用户的界面相关的信息,即用户界面信息,和网络服务请求信息。用户界面信息决定了客户端应用系统如何与用户交互,即应该像用户呈现什么样的界面。网络服务请求信息决定了如何发起网络服务请求,即应该以什么上的协议发起请求,应该发起什么请求。基于所述配置信息,就可以为用户提供特定的服务。

[0116] 根据本发明的一个实施方式,生成的配置信息可以存储在一个应用对象中,此时

该应用对象可以称作是“平台”，相应地，其他用于存储业务数据的对应对象可以称作是“数据应用对象”。然而，本发明并不局限于此，配置信息可以存储在可由客户端应用系统获取的任何适当位置，并且配置信息也可以以任何适当的方式存储。

[0117] 特别需要说明的是，根据现有技术的应用开发方法，由预定的、刚性数据模型设计的打包软件不能以简单、经济的方式在应用投入使用后将需要关联的信息关联起来。而根据本发明可以容易地实现这一点。例如，可以原有用户界面上添加一个新的链接按钮，并通过简单的点击、拖放等操作将与需要链接的两个信息对应的两个应用对象关联起来，从而建立了与该按钮相关联的动作。然后，可以更新配置信息。此后，在用户通过装载新的配置信息来启动客户端应用系统时，就可以在用户界面上显示“链接”操作按钮，以使用户将所需的信息链接起来，而无需改变两个应用对象本身。类似地，还可以根据用户需求方便地对网络应用进行其他更改，诸如解除应用对象的链接关系等。

[0118] 接下来，在步骤 1103，将所述应用模板和所述配置信息打包，以生成适合用户的需求的应用包。

[0119] 在本发明的一个实施方式中，进一步地对该应用包执行部署，以便将该应用包提供给用户。

[0120] 在本发明的另一实施方式中，为了加速随后面向服务的应用系统创建过程，可以进一步将生成的应用包保存到存储库中以备以后开发程序使用。

[0121] 在本发明的又一实施方式中，为了方便应用包的共享，可以进一步将生成的应用包保存到存储库中，以供外部应用创建人员使用。

[0122] 然而，需要说明的是，执行部署的步骤并非是必需的。例如，创建应用的人员可以根据存储库中原有的应用模板来创建一些自定义的应用包存储在存储库中，以方便随后使用。在这种情况下，并不需要部署步骤。

[0123] 下面，将参考图 12 至图 15 来描述本发明提供了用于创建面向服务的应用系统的创建器。

[0124] 图 12 示出根据本发明的一个实施方式的用于创建面向服务的应用系统的创建器。如图所示，本发明的创建器 1200 包括：选择装置 1201，配置装置 1202，和打包装置 1203。

[0125] 选择装置 1201 用于选择一个或多个应用模板。所述应用模板包括用户界面、应用对象和相应动作。应用对象包括：专用于存储特定类型数据的平面结构数据单元；用于存储该数据单元中的数据与其他应用对象中数据单元的数据之间关系的关系存储单元；以及用于操作该数据单元的数据和关系存储单元中的信息的应用代码模块。配置装置 1202 用于针对用户的需求对所述应用模板进行配置，以生成针对所述用户的配置信息。打包装置 1203 用于将所述应用模板和所述配置信息打包，以生成适合用户的需求的应用包。

[0126] 在根据本发明的一个实施方式中，所述选择装置 1201 进一步用于以下操作中的一种或者多种：从存储库中选择用户界面、应用对象以及相应动作中的一种或者多种；从存储库中选择预定义的应用包；以及从外部导入预定义的应用包，其中，所述预定义应用包包括用于实现特定功能的多个应用模板，或者包括实现特定功能的多个应用模板的多个应用模板以及相关的配置信息。

[0127] 在根据本发明的另一实施方式中，所述配置装置 1202 进一步被用于以下操作中

的一种或者多种；删除所述用户界面、所述应用对象、所述动作中至少一个；修改所述用户界面、所述应用对象、所述动作中至少一个；增加所述用户界面、所述应用对象、所述动作中至少一个；以及设置所述动作中的网络服务请求所遵循的协议。

[0128] 在根据本发明的又一实施方式中，所生成的应用包包括：数据应用对象，其包括专用于存储业务数据的平面结构数据单元；和/或平台，该平台是一种应用对象，包括专用于存储所述配置信息的平面结构数据单元。

[0129] 在根据本发明的再一实施方式中，所述配置信息包括用户界面信息和/或网络服务请求信息。

[0130] 此外，图 13、图 14 和图 15 还示出根据本发明的另一些实施方式。

[0131] 如图所示，图 13、图 14、图 15 中的创建器 1300、1400 和 1500 分别包括：选择装置 1301、1401 和 1501，其对应于图 12 中的选择装置 1201，用于选择一个或多个应用模板；配置装置 1302、1402 和 1502，其对应于图 12 中的配置装置 1202，用于针对用户的需求对所述应用模板进行配置，以生成针对所述用户的配置信息；和打包装置 1303、1403 和 1503，其对应于图 12 中的打包装置 1203，用于将所述应用模板和所述配置信息打包，以生成适合用户的需求的应用包。

[0132] 与图 12 所示的创建器 1200 不同的是，图 13 所示的创建器 1300 进一步包括部署装置 1304，用于针对所述应用包执行部署，以便为用户提供所述面向服务的应用系统；图 14 所示的创建器 1400 进一步包括保存装置 1405，用于保存所述应用包，以便于随后创建其他应用；图 15 所示的创建器 1500 进一步包括导出装置 1506，用于导出所述应用包，以供外部应用创建人员使用。

[0133] 需要说明的是，在上文中示出的各种特征，可以进行组合，只要这些组合并不违背本发明的原理。例如，在根据本发明的又一实施方式中，可以包括上述图 13、14 和图 15 中的部署装置 1304、导入装置 1405 和导出装置 1506。

[0134] 根据本发明，面向服务的应用系统主要由多个应用对象构成，并且优选地包括用于存储配置信息的称作“平台”的应用对象。因此，基于这种结构，可以通过预先定义的应用模板来以快速、简单的方式（通过点击、拖放等简单的鼠标操作）来创建面向服务的应用系统。

[0135] 另外，可以对选择的预定义打包文件进行修改以满足特定用户的需求，也可以通过略微修改原先为其他用户创建的应用包来为另一用户创建类似的应用。此外，还可以容易地修改已有的面向服务的应用系统来适应客户新的需求。

[0136] 根据本发明，相同类型的业务数据（诸如联系人信息）存储在逻辑上为单个表或者单个文件的同一平面结构数据单元中，但该平面结构数据单元和该应用对象甚至整个网络应用系统均可以存在于网络中的多个位置，换句话说讲，可以存在与多个物理位置或者多个域中。根据本发明，可以根据业务需求将各个平面结构数据单元、各个应用对象或整个网络应用系统布置在单个位置或者多个位置，或布置在单个域中或者多个域中。例如，对于小型公司，可以将它们存储在单个位置，对于大型的国际性公司，则可以将其提供在多个位置。另外，根据本发明，存储在该单一逻辑表中的业务数据可以由针对不同业务的多个应用共享，这与利用不同概念针对不同业务分别开发打包软件传统方式不同，根据该传统方式，打包软件是根据预定的、刚性数据模型基于不同理念、针对不同业务而设计的，这些打包软

件之间通常是不能通信的,更不能共享业务数据。所以,与传统方式相比,在本发明的网络应用系统中,业务数据不会出现信息孤岛。因此,不会存在信息整合、数据完整性等方面的问题,使得用户从维护数据完整性等繁杂的工作中解脱出来,减少了成本,提供了效率。

[0137] 另外,根据本发明,不同的客户端应用系统(例如,用户的客户端应用系统、该用户的提供商的客户端应用系统、该用户的客户的客户端应用系统和该用户的合作方的客户端应用系统)可以对应于一个面向服务的应用系统中的不同应用对象子集,这例如可以通过装载不同的配置信息来实现。因而,可以实现用户的业务数据与其提供商、客户、合作方等的适当共享。因此,本发明不但为用户的公司带来了诸多益处,而且为用户的整个公司链带来了方便。

[0138] 根据本发明的优选实施方式,还可以通过适配器使得本发明的网络应用系统中的客户端应用系统和/或面向服务的应用系统与第三方系统的相互通信。从而,可以将用户已有的业务系统与本发明的网络应用系统结合起来使用,从而节约成本。另外,通过使用适配器可以使得本发明的网络应用系统可以适应于任何变化,使得本发明的网络应用系统可以利用诸如 REST 方式或者 SOAP 协议等适当方式与第三方系统进行数据交换,因此可以进一步避免信息孤岛的问题。

[0139] 下面,将参考图 16 来描述可以实现本发明的计算机设备。图 16 示意性示出了可以实现根据本发明的实施方式的计算设备的结构方框图。

[0140] 图 16 中所示的计算机系统包括 CPU(中央处理单元)1601、RAM(随机存取存储器)1602、ROM(只读存储器)1603、系统总线 1604、硬盘控制器 1605、键盘控制器 1606、串行接口控制器 1607、并行接口控制器 1608、显示器控制器 1609、硬盘 1610、键盘 1611、串行外部设备 1612、并行外部设备 1613、显示器 1614 以及网络适配器 1615。在这些部件中,与系统总线 1604 相连的有 CPU 1601、RAM1602、ROM 1603、硬盘控制器 1605、键盘控制器 1606、串行接口控制器 1607、并行接口控制器 1608、显示器控制器 1609 以及网络适配器 1615。硬盘 1610 与硬盘控制器 1605 相连,键盘 1611 与键盘控制器 1606 相连,串行外部设备 1612 与串行接口控制器 1607 相连,并行外部设备 1613 与并行接口控制器 1608 相连,显示器 1614 与显示器控制器 1609 相连,而网络接口控制器 1615 允许该计算机系统访问诸如互联网的计算机网络。

[0141] 图 16 所述的结构方框图仅仅为了示例的目的而示出的,并非是对本发明的限制。在一些情况下,可以根据需要添加或者减少其中的一些设备,例如用于实现本发明的计算机设备可以并不具有显示器、键盘等输入/输出设备。

[0142] 此外,本发明的实施方式可以以软件、硬件或者软件和硬件的结合来实现。硬件部分可以利用专用逻辑来实现;软件部分可以存储在存储器中,由适当的指令执行系统,例如微处理器或者专用设计硬件来执行。

[0143] 虽然已经参考目前考虑到的实施方式描述了本发明,但是应该理解本发明不限于所公开的实施方式。相反,本发明旨在涵盖所附权利要求的精神和范围之内所包括的各种修改和等同布置。以下权利要求的范围符合最广泛解释,以便包含所有这样的修改及等同结构和功能。

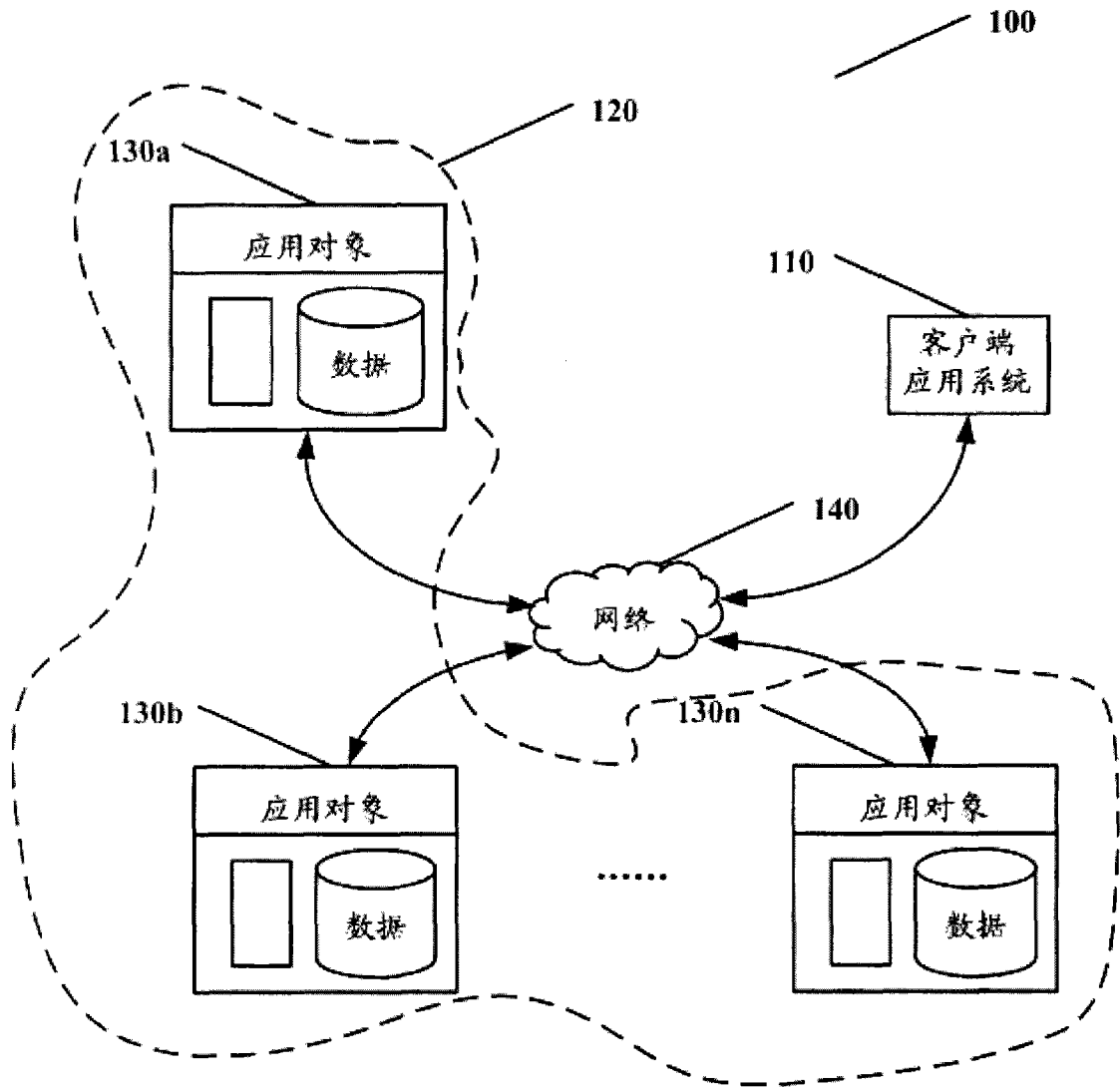


图 1

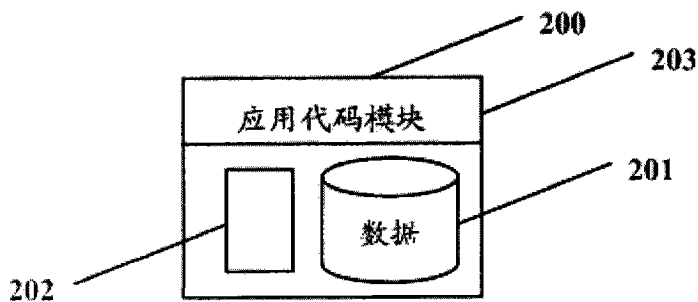


图 2a

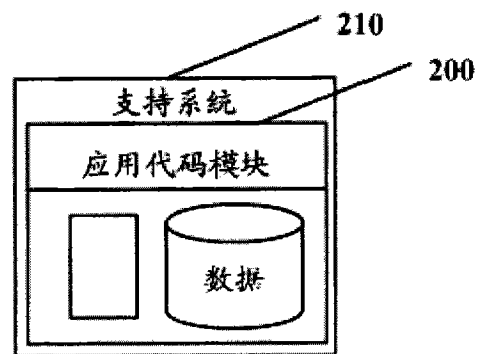


图 2b

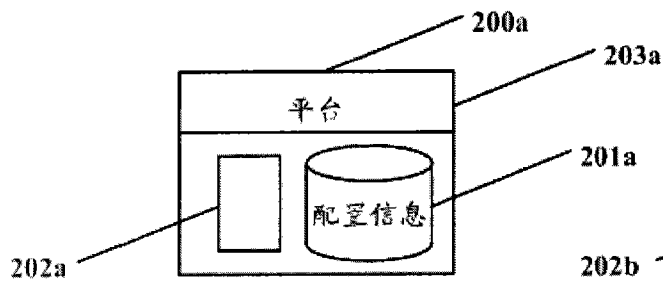


图 2c

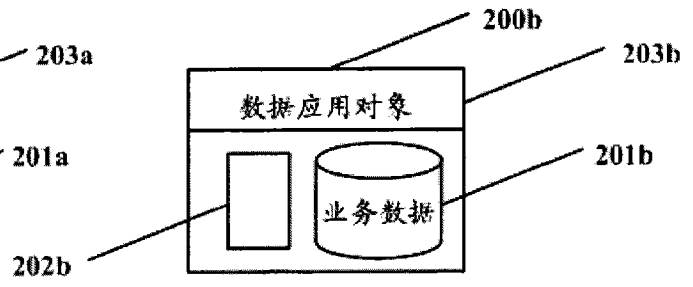


图 2d

图 2c 图 2d

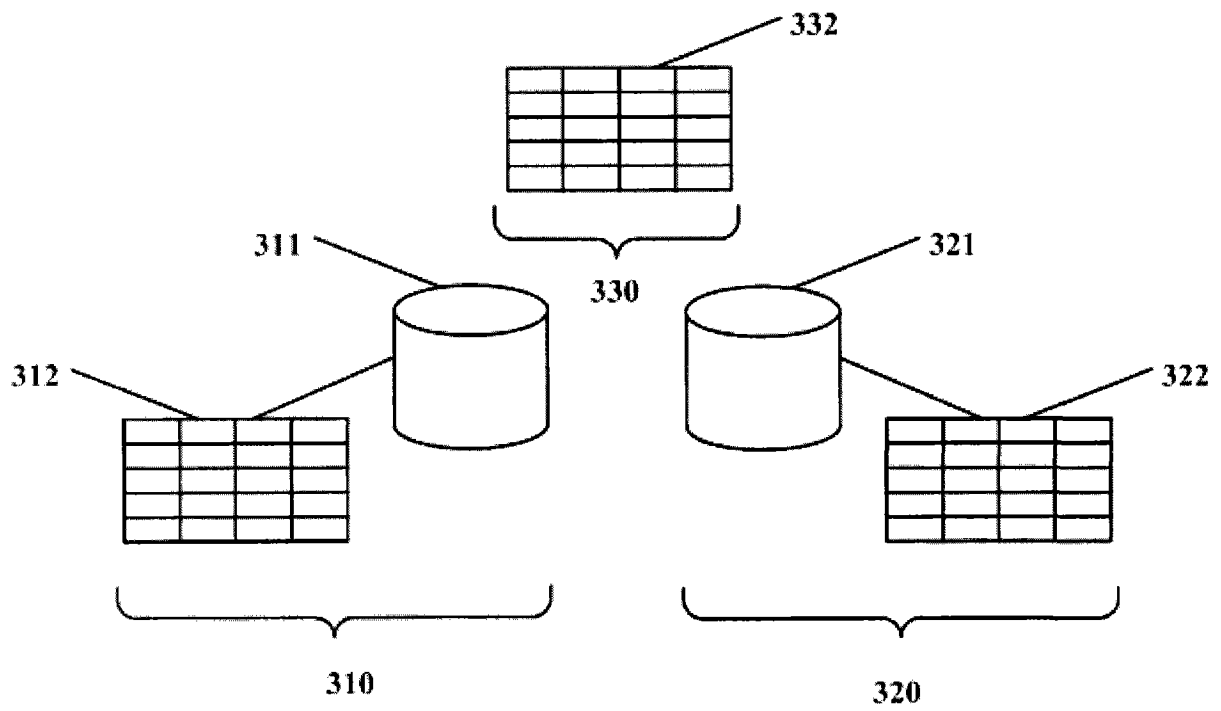


图 3

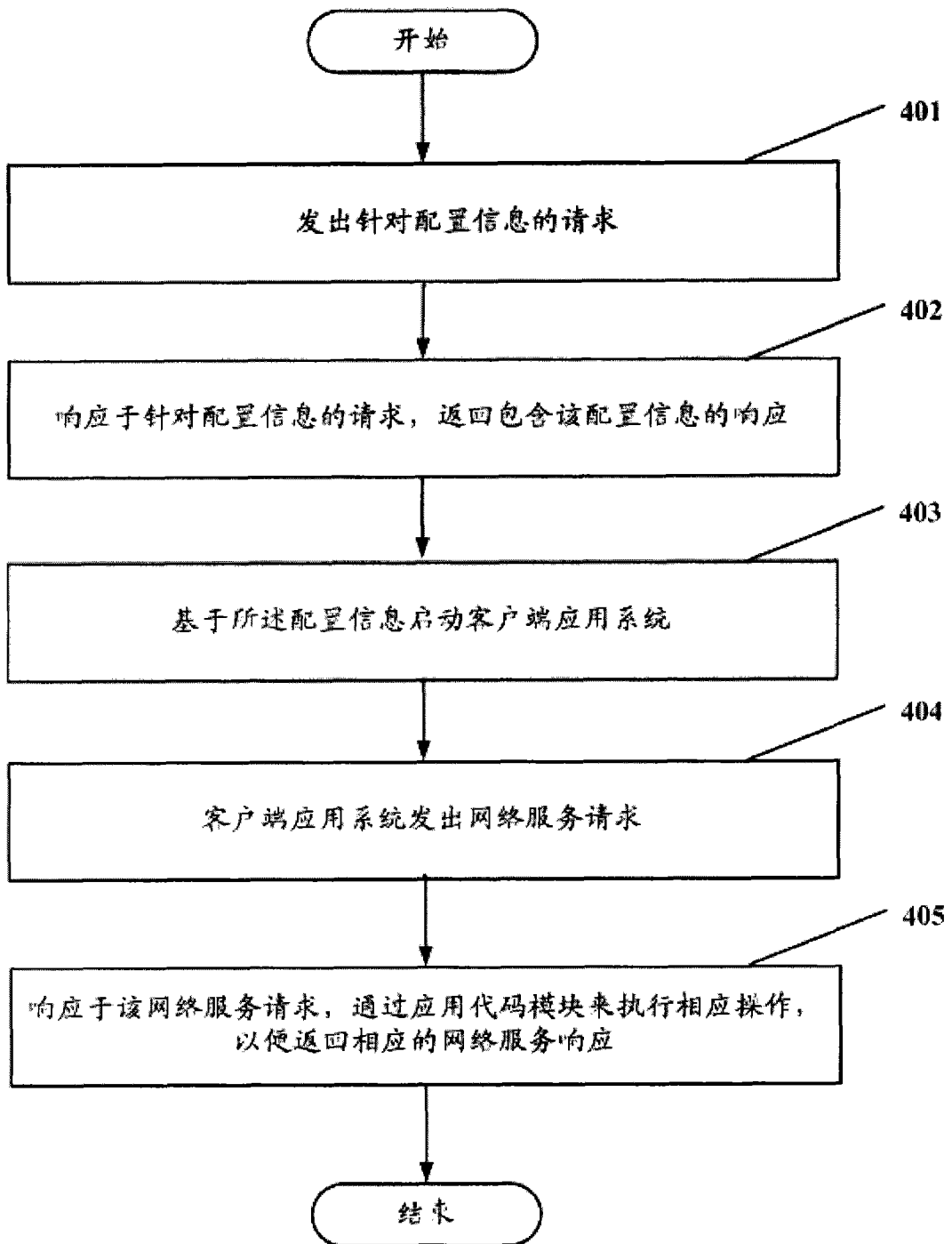


图 4

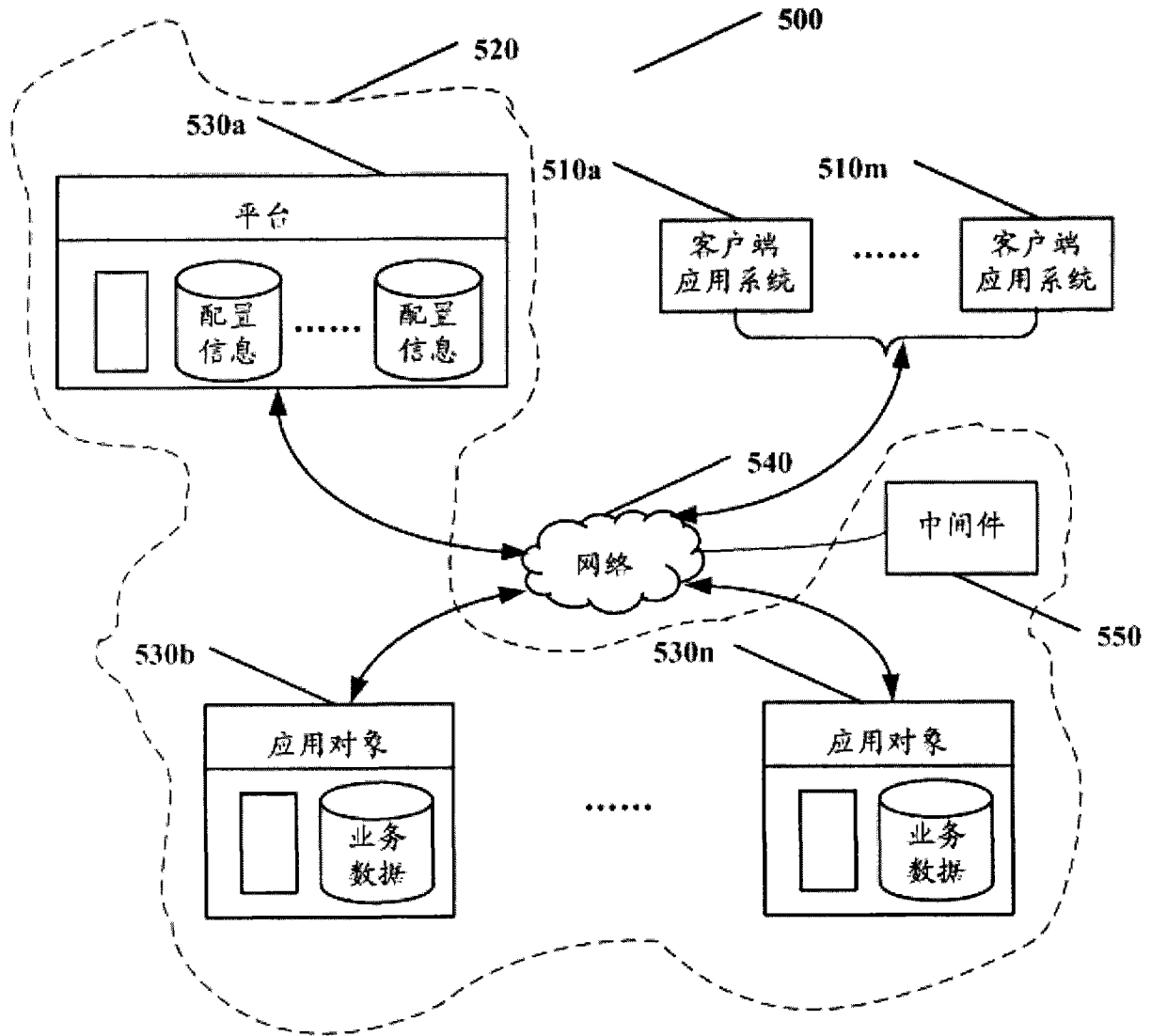


图 5



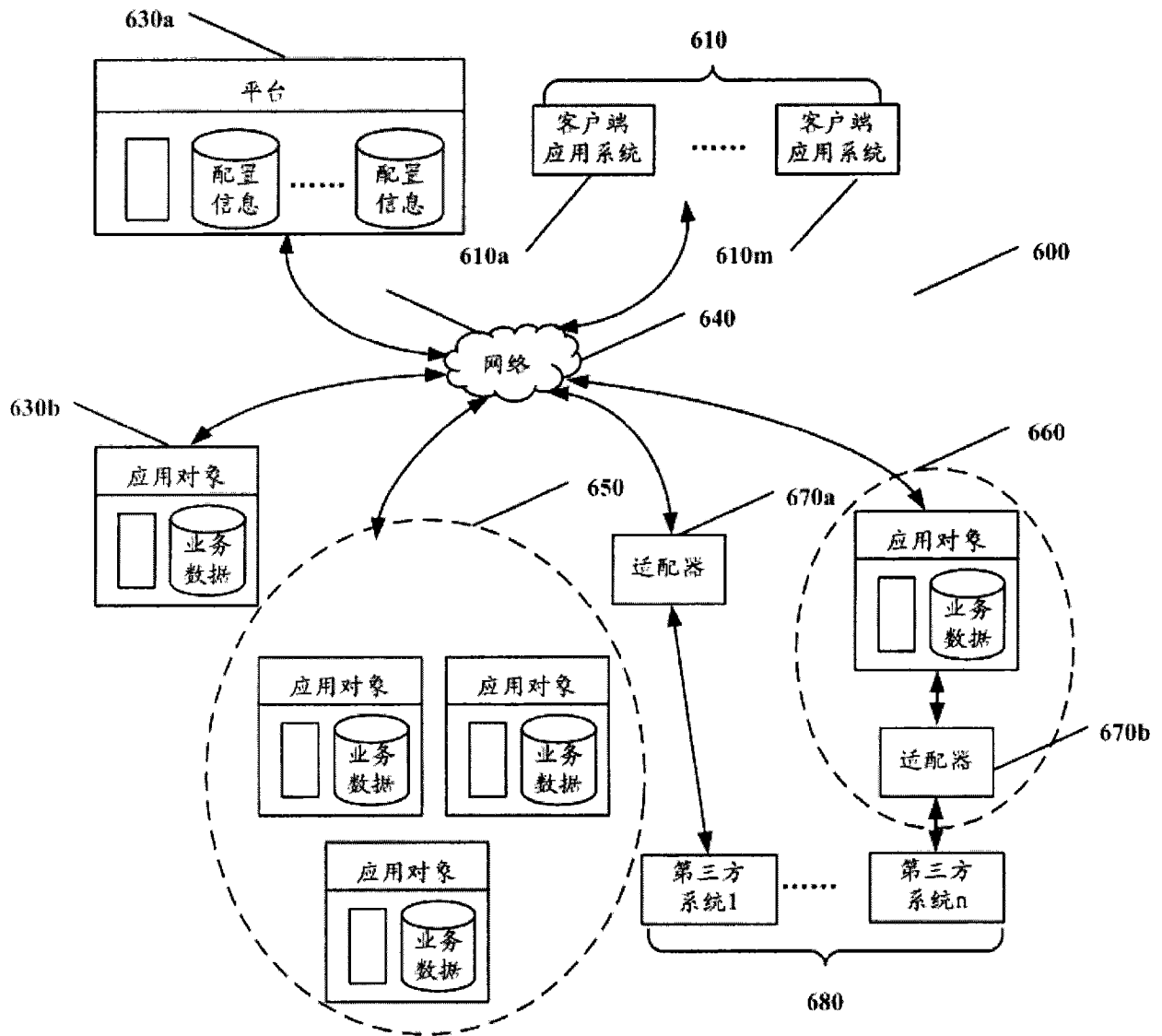


图 6

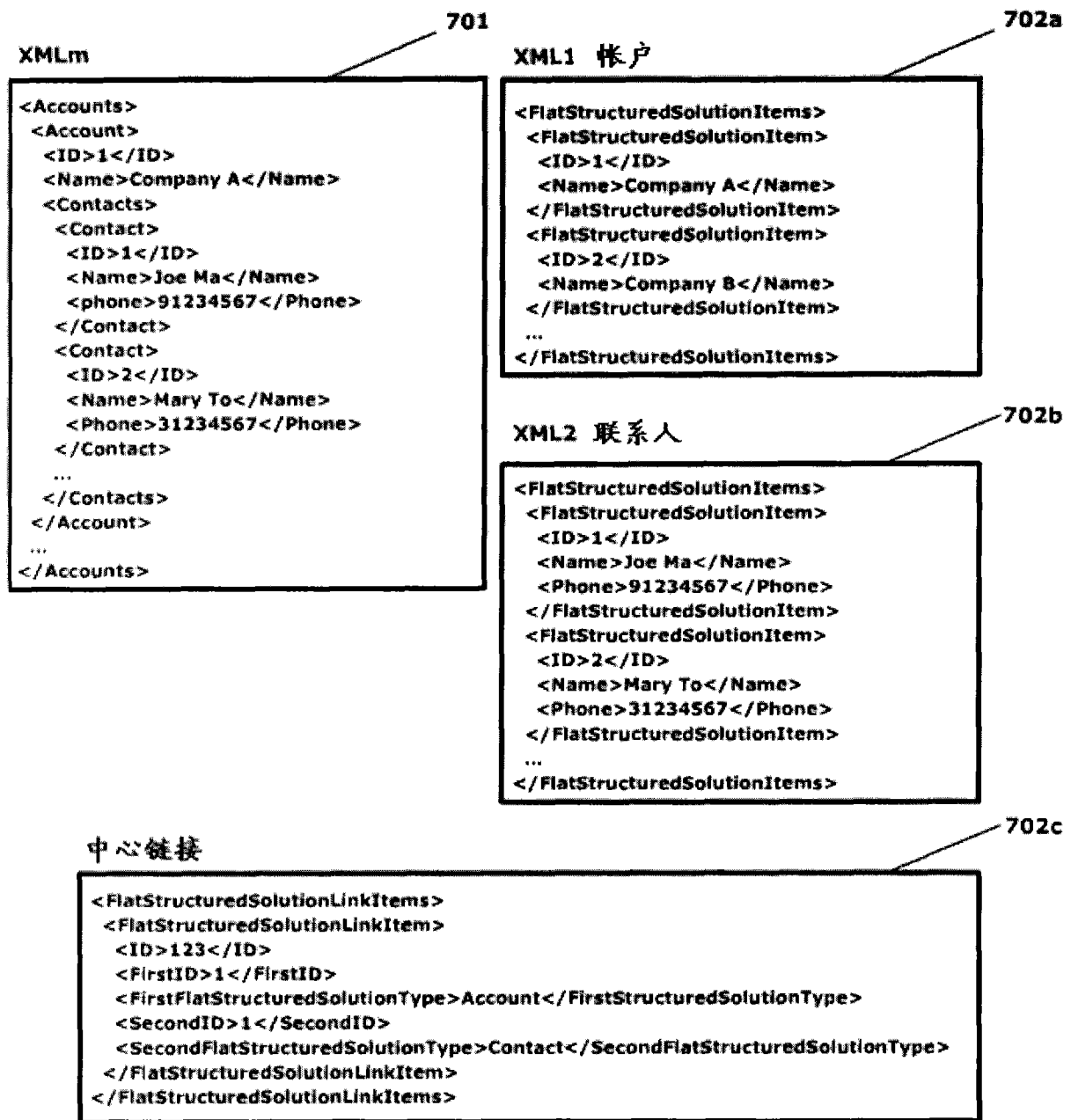


图 7

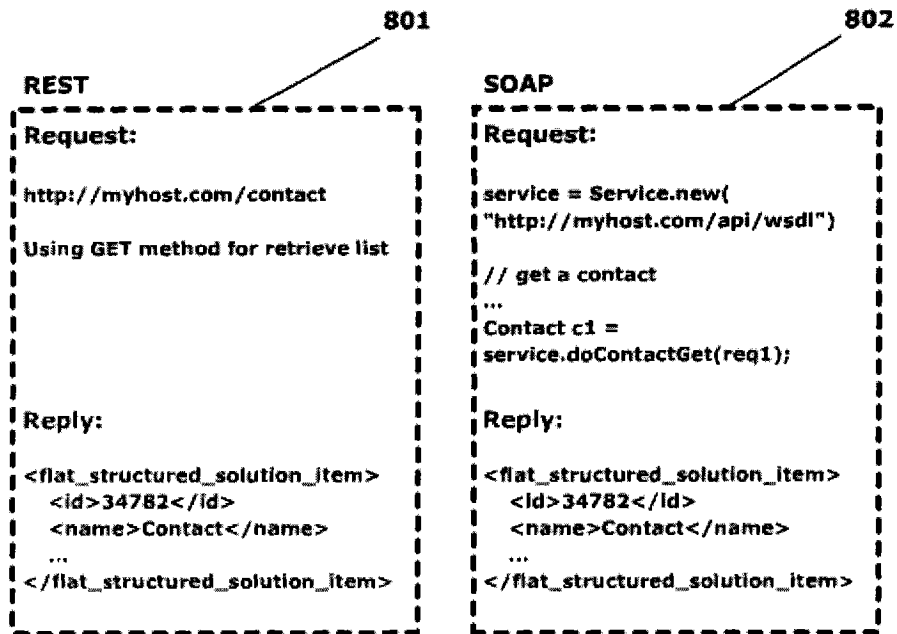


图 8

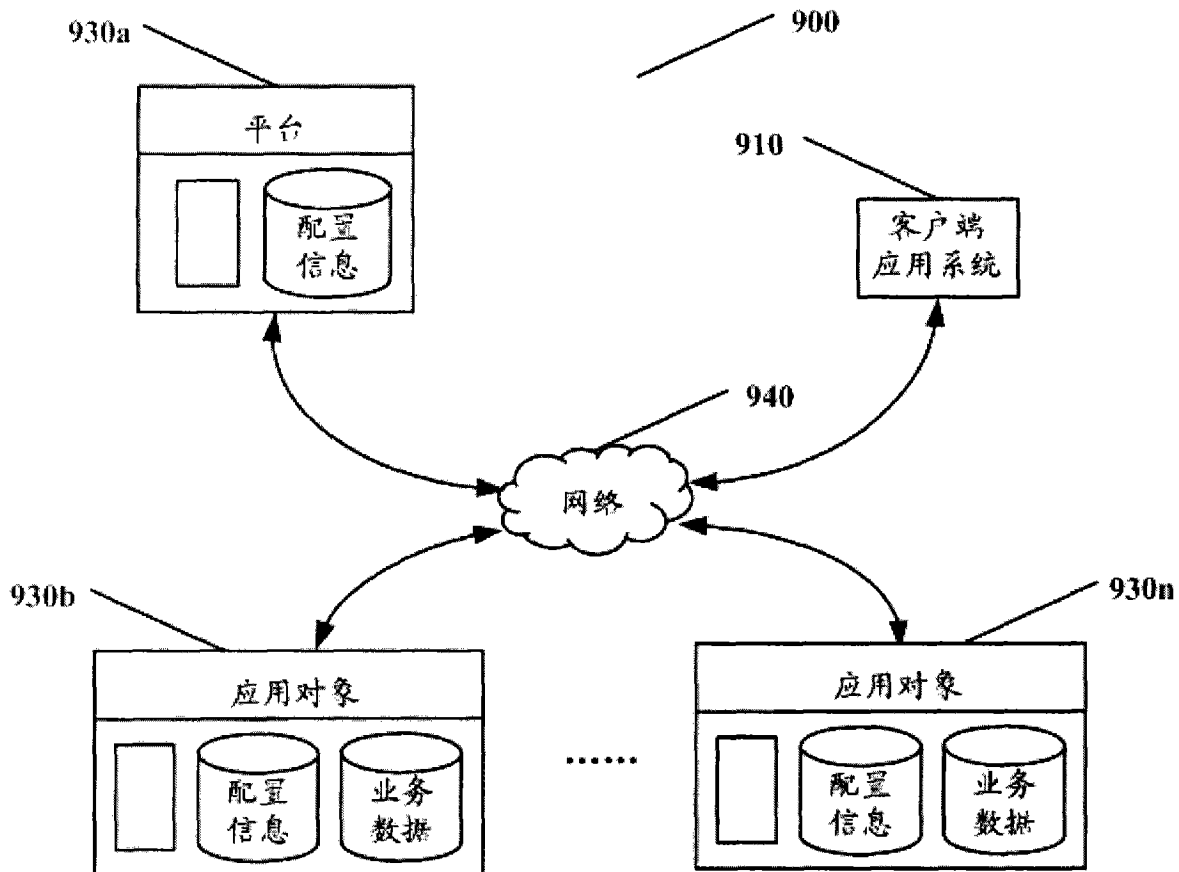


图 9

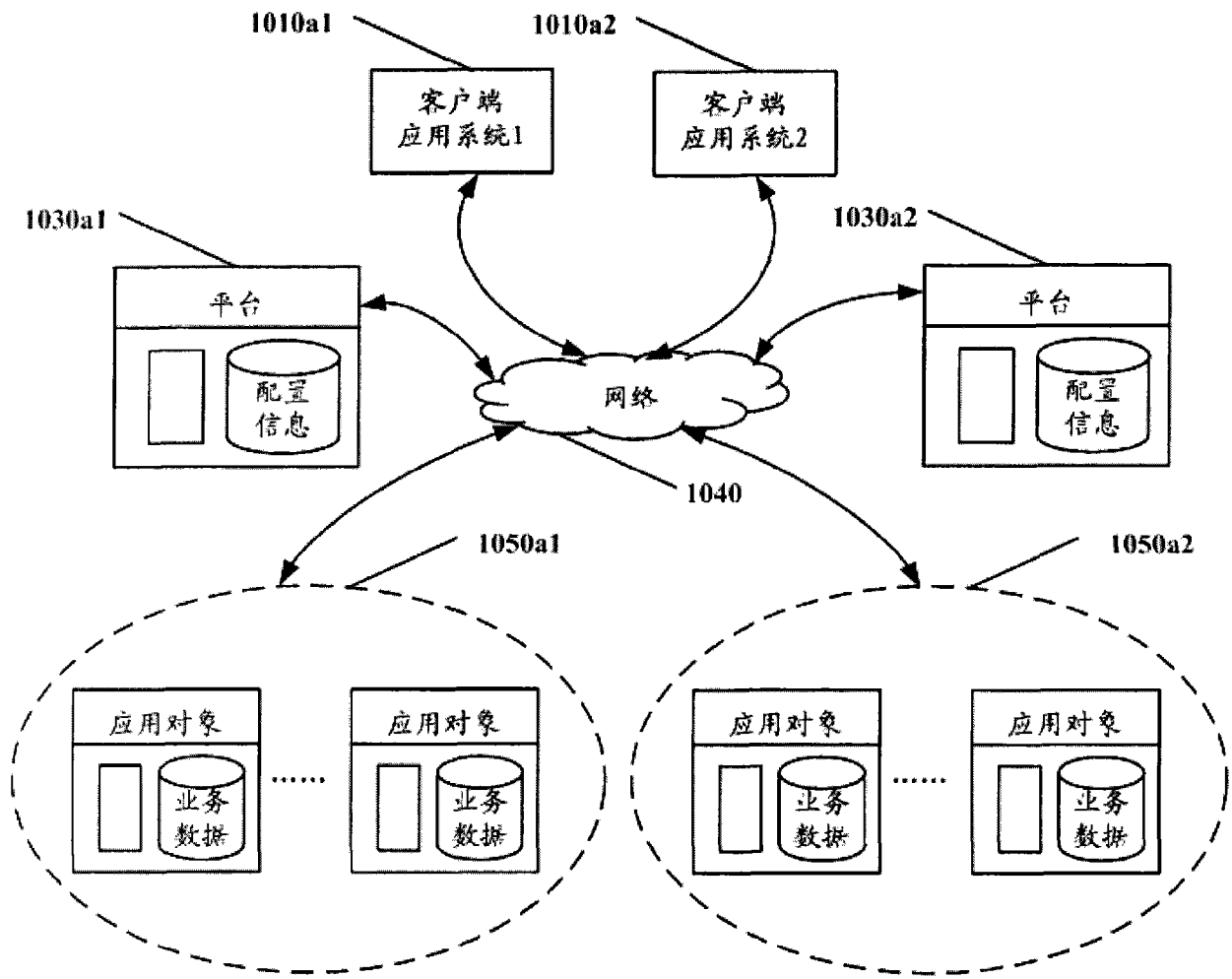


图 10

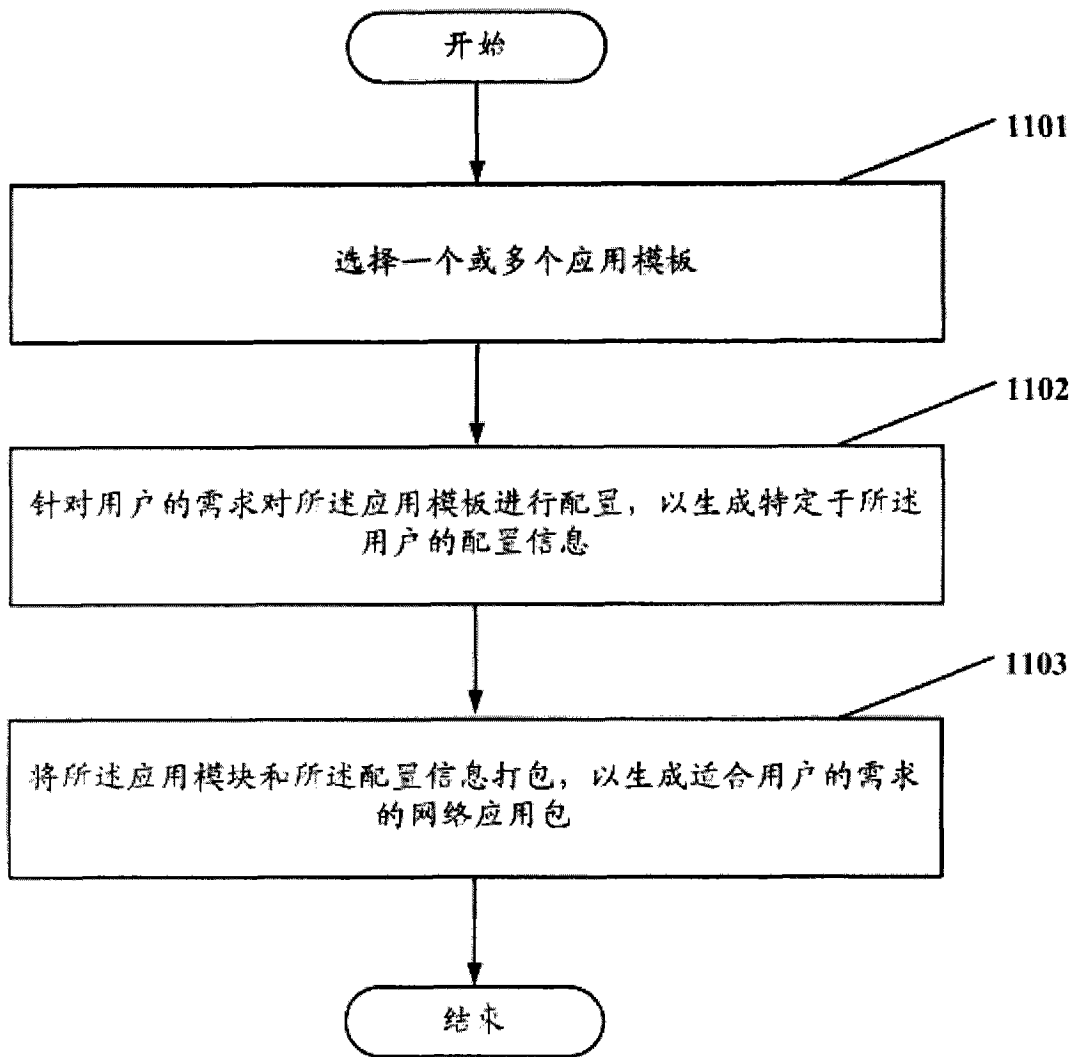


图 11

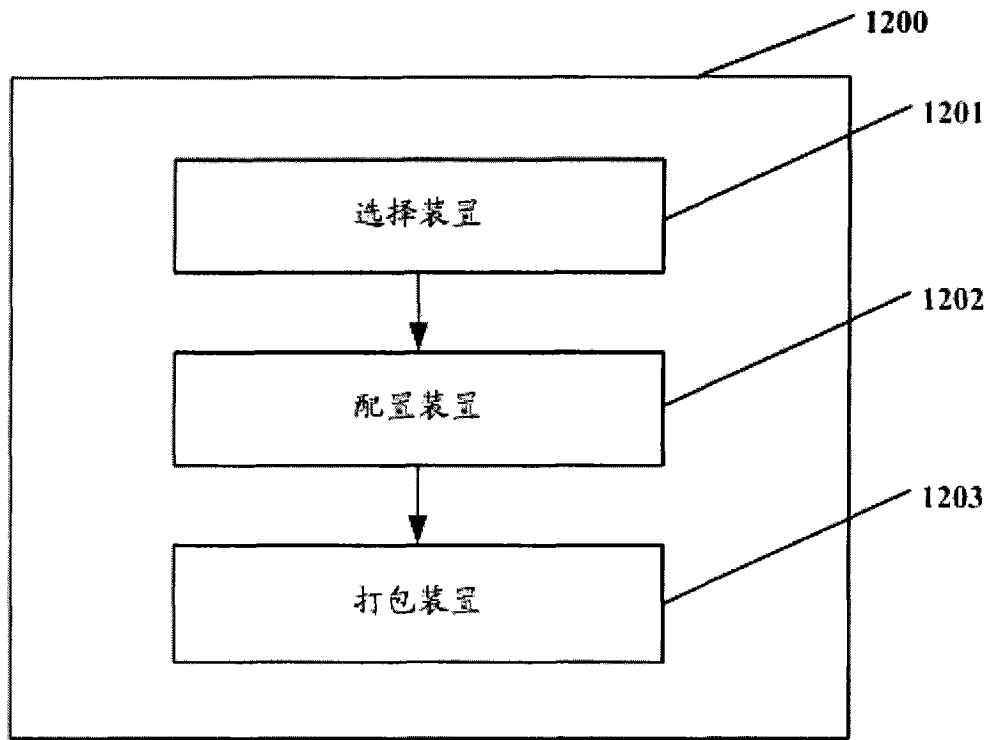


图 12

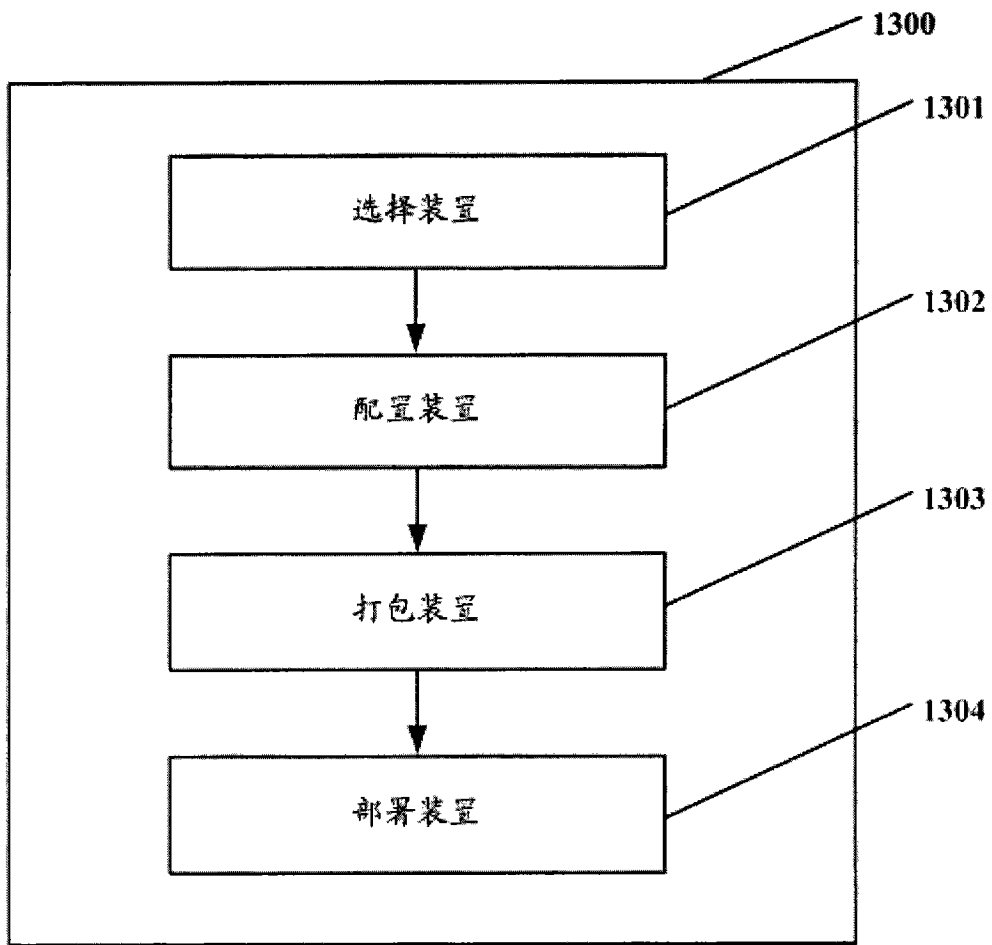


图 13

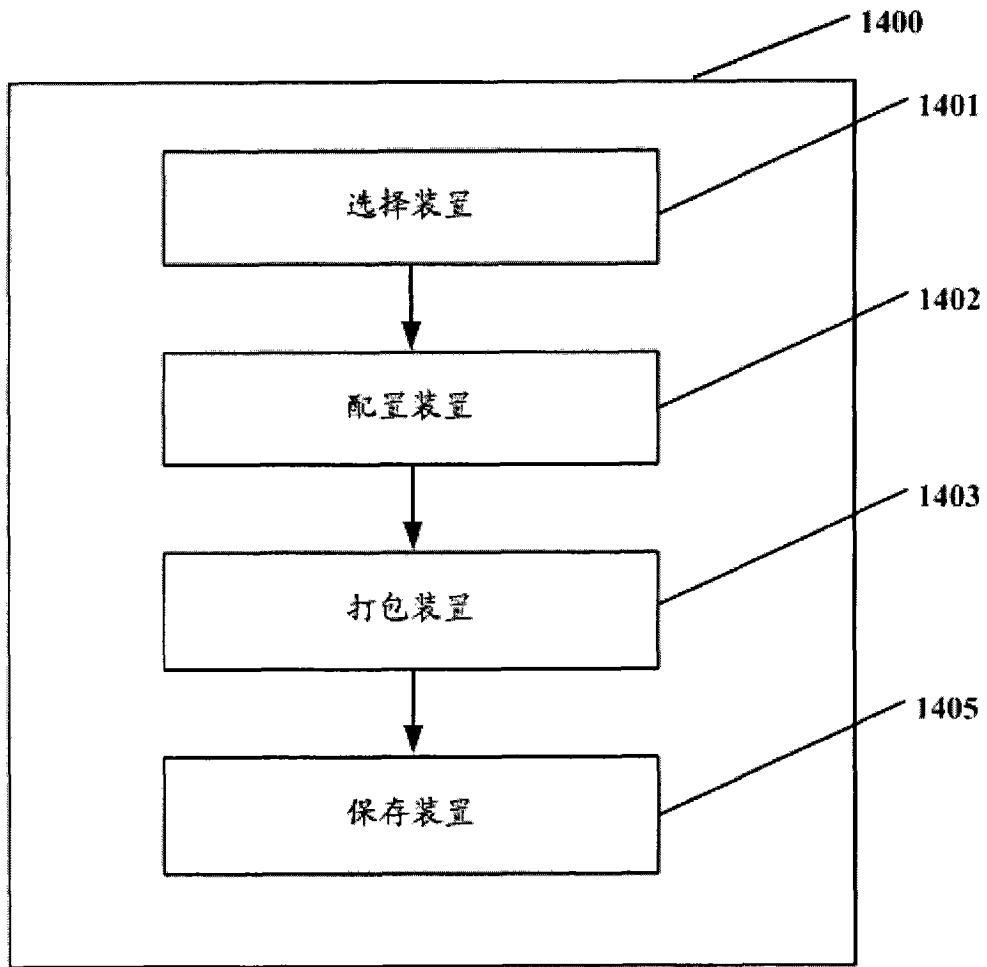


图 14



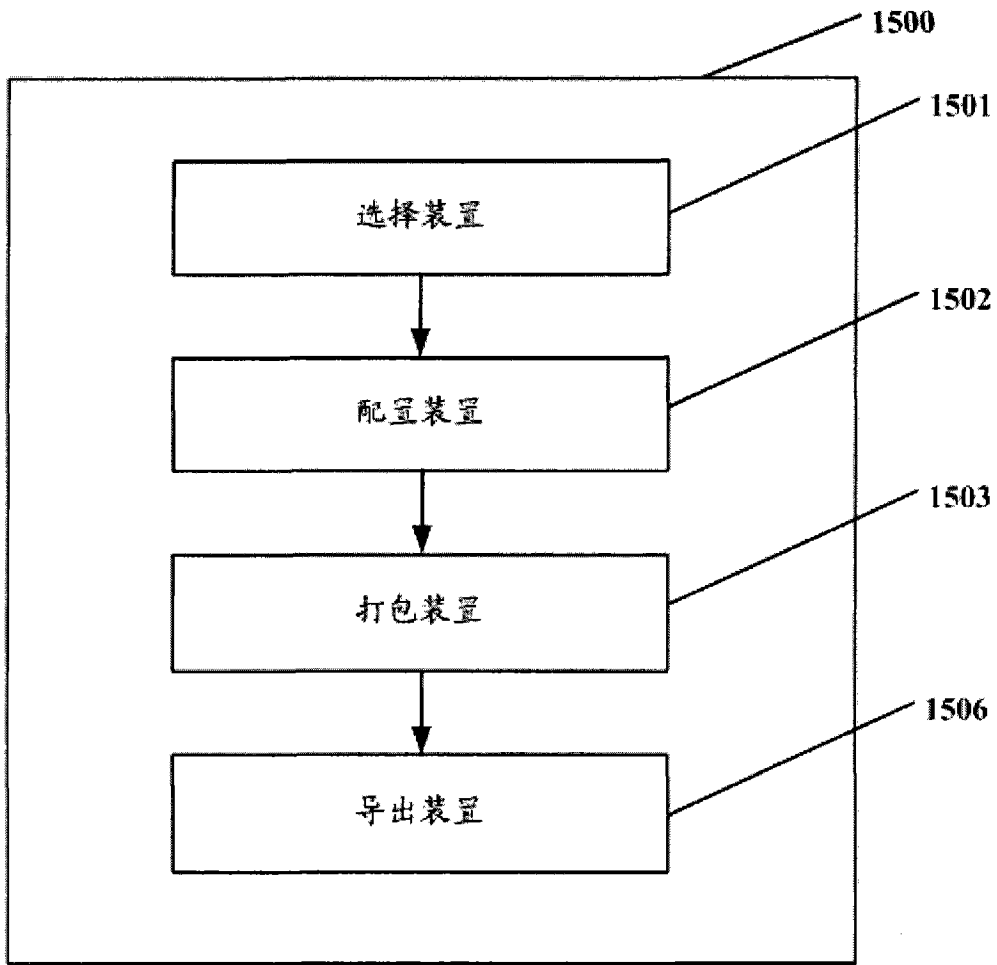


图 15

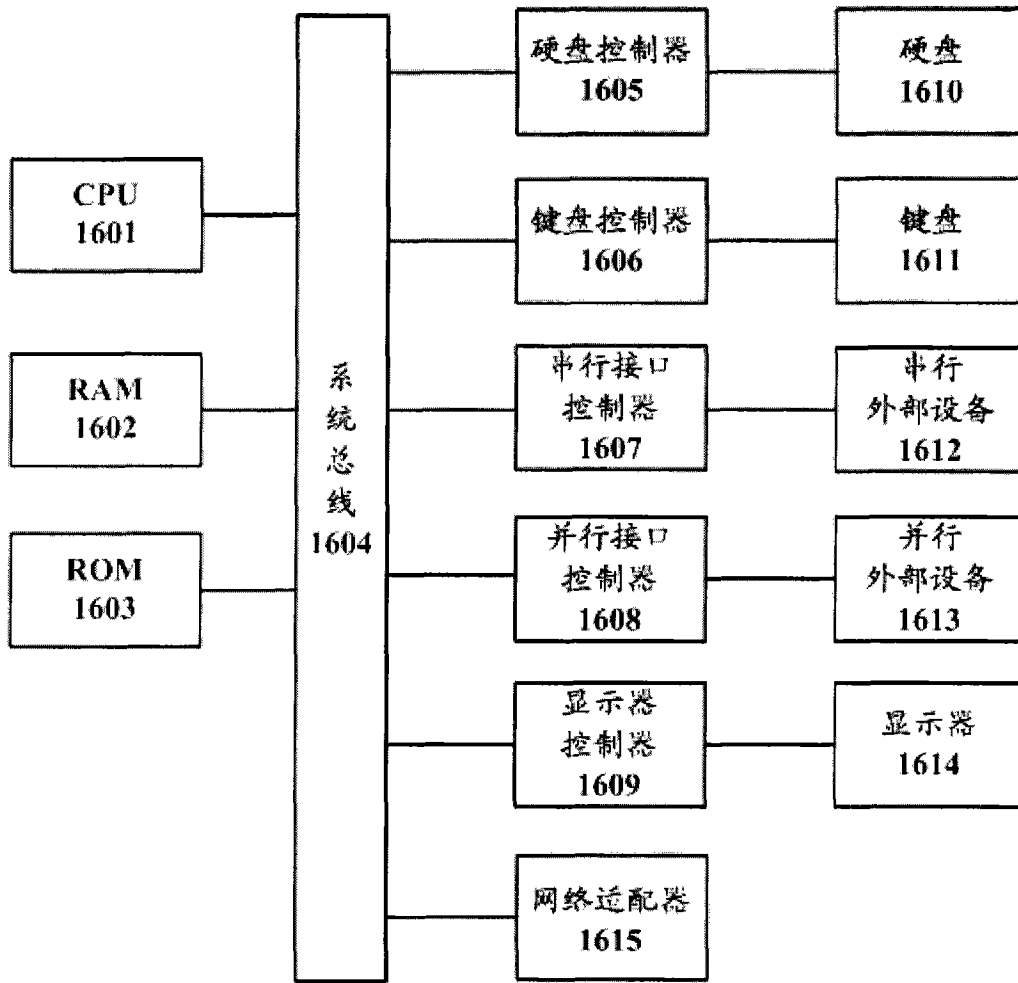


图 16