(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105593844 B
(45)授权公告日 2019.08.16

(21)申请号 201480051434.2

(22)申请日 2014.08.28

(65)同一申请的已公布的文献号
申请公布号 CN 105593844 A

(43)申请公布日 2016.05.18

(30)优先权数据
61/880,767 2013.09.20 US
61/909,205 2013.11.26 US
14/304,356 2014.06.13 US
14/304,393 2014.06.13 US

(85)PCT国际申请进入国家阶段日
2016.03.18

(86)PCT国际申请的申请数据
PCT/US2014/053138 2014.08.28

(87)PCT国际申请的公布数据
WO2015/041829 EN 2015.03.26

(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚

(72)发明人 耿益璇 E·A·戴维斯

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 罗亚男

(51)Int.Cl.
G06F 16/93(2019.01)
G06F 9/445(2018.01)

(56)对比文件
US 2006036745 A1,2006.02.16,说明书第7-31段,62-117段,附图1-6.
US 2006036745 A1,2006.02.16,说明书第7-31段,62-117段,附图1-6.
US 2011087689 A1,2011.04.14,说明书第52段.
US 2009205013 A1,2009.08.13,说明书第23段.
CN 102656557 A,2012.09.05,全文.
US 2006253830 A1,2006.11.09,全文.
US 2006010134 A1,2006.01.12,全文.
US 7926030 B1,2011.04.12,全文.
US 8131736 B1,2012.03.06,全文.
US 2012159145 A1,2012.06.21,全文.

审查员 曹青

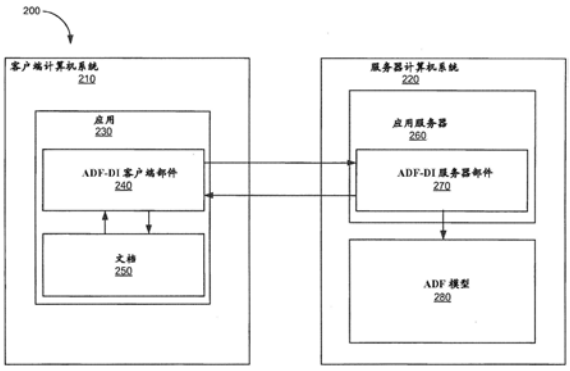
权利要求书3页 说明书20页 附图12页

(54)发明名称

运行时定制基础设施

(57)摘要

在各种实施例中,公开了允许开发人员允许利用桌面集成框架开发的文档的运行时定制的方法、系统和非暂态计算机可读介质。工作簿元数据是描述给定的工作簿如何与特定的web应用集成的一组信息。当工作簿被发布时,元数据可被写入所发布的工作簿的本地高速缓存以及工作簿定义文件中。元数据管理可以由元数据服务支持,从而允许所发布的工作簿的更新和定制独立于所发布的工作簿中的本地高速缓存和工作簿定义文件。



CN 105593844 B

1.一种用于构建应用用户界面的方法,所述方法包括:

在计算机系统,接收识别文档的信息,其中所述文档的内容对于原生应用是原生的,其中所述原生应用包括应用开发框架ADF或者通过所述原生应用的插件与所述应用开发框架ADF通信;

由计算机系统,确定与用于所述文档的元数据关联的规则集,其中所述规则集定义要为所述文档生成的元数据的版本,并且配置要被应用于用于所述文档的元数据的定制;

由计算机系统,基于所确定的规则集,生成用于所述文档的所述元数据,其中所生成的元数据提供所述文档的所述内容的一部分与和数据模型相关联的数据之间的链接,其中所述数据模型由与所述原生应用的所述插件通信的一个或多个基于web的应用提供,其中所述一个或多个基于web的应用由所述计算机系统托管;以及

从计算机系统,将用于所述文档的所生成的元数据传送到所述原生应用,其中该文档在使用所生成的元数据来呈现时,被配置成通过将由所述一个或多个基于web的应用提供的功能扩展到所述原生应用,来为所述一个或多个基于web的应用充当用户界面。

2.如权利要求1所述的方法,其中由计算机系统确定与用于所述文档的元数据关联的规则集包括基于与该文档的用户关联的角色来确定规定要使用的用于该文档的元数据的至少一个规则。

3.如权利要求1所述的方法,其中所述计算机系统包括所述ADF的与所述原生应用的所述插件通信的部件,并且其中所述原生应用的所述插件是所述ADF的客户端侧部件。

4.如权利要求1所述的方法,其中由计算机系统确定与用于所述文档的元数据关联的规则集包括基于时间或日期信息确定规定要使用的用于该文档的元数据的至少一个规则。

5.如权利要求1所述的方法,其中由计算机系统确定与用于所述文档的元数据关联的规则集包括基于组织的部门确定规定要使用的用于该文档的元数据的至少一个规则。

6.如权利要求1所述的方法,其中由计算机系统确定与用于所述文档的元数据关联的规则集包括从元数据管理服务器检索一个或多个规则。

7.如权利要求1所述的方法,其中由计算机系统基于所确定的配置要应用到用于所述文档的元数据的定制的规则集生成用于该文档的元数据包括对用于该文档的基本元数据应用多个元数据定制。

8.一种存储计算机可执行代码的非暂态计算机可读介质,所述计算机可执行代码包括:

用于在计算机系统接收识别文档的信息的代码,其中所述文档的内容对于原生应用是原生的,其中所述原生应用包括应用开发框架ADF或者通过所述原生应用的插件与所述应用开发框架ADF通信;

用于由计算机系统确定与用于所述文档的元数据关联的规则集的代码,其中所述规则集定义要为所述文档生成的元数据的版本,并且配置要被应用于用于所述文档的元数据的定制;

用于由计算机系统基于所确定的规则集生成用于所述文档的所述元数据的代码,其中所生成的元数据提供所述文档的所述内容的一部分与和数据模型相关联的数据之间的链接,其中所述数据模型由与所述原生应用的所述插件通信的一个或多个基于web的应用提供,其中所述一个或多个基于web的应用由所述计算机系统托管;及

用于从计算机系统将用于所述文档的所生成的元数据传送到所述原生应用的代码，其中该文档在使用所生成的元数据来呈现时，被配置成通过将由所述一个或多个基于web的应用提供的功能扩展到所述原生应用，来为所述一个或多个基于web的应用充当用户界面。

9.如权利要求8所述的非暂态计算机可读介质，其中用于由计算机系统确定与用于所述文档的元数据关联的规则集的代码包括基于与该文档的用户关联的角色来确定规定要使用的用于该文档的元数据的至少一个规则的代码。

10.如权利要求8所述的非暂态计算机可读介质，其中所述计算机系统包括所述ADF的与所述原生应用的所述插件通信的部件，并且其中所述原生应用的所述插件是所述ADF的客户端侧部件。

11.如权利要求8所述的非暂态计算机可读介质，其中用于由计算机系统确定与用于所述文档的元数据关联的规则集的代码包括用于基于时间或日期信息确定规定要使用的用于该文档的元数据的至少一个规则的代码。

12.如权利要求8所述的非暂态计算机可读介质，其中用于由计算机系统确定与用于所述文档的元数据关联的规则集的代码包括用于基于组织的部门确定规定要使用的用于该文档的元数据的至少一个规则的代码。

13.如权利要求8所述的非暂态计算机可读介质，其中用于由计算机系统确定与用于所述文档的元数据关联的规则集的代码包括用于从元数据管理服务器检索一个或多个规则的代码。

14.如权利要求8所述的非暂态计算机可读介质，其中用于由计算机系统基于所确定的配置要应用到用于所述文档的元数据的定制的规则集来生成用于该文档的元数据的代码包括用于对用于该文档的基本元数据应用多个元数据定制的代码。

15.一种用于构建应用用户界面的系统，所述系统包括：

处理器；及

存储指令集的存储器，指令集当被处理器执行时，配置处理器以：

接收识别文档的信息，其中所述文档的内容对于原生应用是原生的，其中所述原生应用包括应用开发框架ADF或者通过所述原生应用的插件与所述应用开发框架ADF通信；

确定与用于所述文档的元数据关联的规则集，其中所述规则集定义要为所述文档生成的元数据的版本，并且配置要被应用于用于所述文档的元数据的定制；

基于所确定的规则集，生成用于所述文档的所述元数据，其中所生成的元数据提供所述文档的所述内容的一部分与和数据模型相关联的数据之间的链接，其中所述数据模型由与所述原生应用的所述插件通信的一个或多个基于web的应用提供，其中所述一个或多个基于web的应用由所述系统托管；及

将用于所述文档的所生成的元数据传送到所述原生应用，其中该文档在使用所述元数据来呈现时，被配置成通过将由所述一个或多个基于web的应用提供的功能扩展到所述原生应用，来为所述一个或多个基于web的应用充当用户界面。

16.如权利要求15所述的系统，其中确定与用于所述文档的元数据关联的规则集包括确定规定以下中的一者或多者的至少一个规则：基于与该文档的用户关联的角色的要使用的用于该文档的元数据、以及基于时间或日期信息的要使用的用于该文档的元数据。

17.如权利要求15所述的系统，还包括ADF部件，其中所述ADF部件与所述原生应用的所

述插件通信，并且其中所述原生应用的所述插件是所述ADF的客户端侧部件。

18.如权利要求15所述的系统，其中确定与用于所述文档的元数据关联的规则集包括基于组织的部门确定规定要使用的用于该文档的元数据的至少一个规则。

19.如权利要求15所述的系统，其中确定与用于所述文档的元数据关联的规则集包括从元数据管理服务器检索一个或多个规则。

20.如权利要求15所述的系统，其中生成用于该文档的元数据包括对用于该文档的基本元数据应用多个元数据定制。

运行时定制基础设施

[0001] 对相关申请的交叉引用

[0002] 本申请要求以下申请的权益和优先权：

[0003] 于2013年9月20日提交且标题为“Runtime Customization Infrastructure”的美国临时申请No.61/880,767，

[0004] 于2013年11月26日提交且标题为“Workbook Composer”的美国临时申请No.61/909,205，

[0005] 于2014年6月13日提交且标题为“Runtime Customization Infrastructure”的美国申请No.14/304,356，及

[0006] 于2014年6月13日提交且标题为“Workbook Composer”的美国申请No.14/304,393，

[0007] 这些申请的公开内容通过引用被结合于此，用于所有目的。

背景技术

[0008] 应用是指在执行时执行特定期望任务的软件程序。一般而言，若干应用在包含操作系统、虚拟机(例如，支持Java™编程语言)、设备驱动器等当中一个或多个的运行时环境中执行，如在相关领域中众所周知的。

[0009] 开发人员常常使用应用开发框架(ADF)(它们自己就是应用)用于实现/开发期望的应用。ADF提供可以直接/间接地在应用的开发中使用的预定义的代码/数据模块集合。ADF还可以提供工具，诸如IDE(集成开发环境)、代码生成器、调试器等等，这便于开发人员以更快/更简单的方式编码/实现应用的期望逻辑。

[0010] 一般而言，ADF通过提供应用开发人员可用来定义用户界面和应用逻辑的可重用的部件和集成开发环境来简化应用的开发，例如，通过选择执行期望任务的部件并定义被选部件的外观、行为和交互。一些ADF是基于模型-视图-控制器设计模式，其促进松散耦合及更容易的应用开发和维护。Oracle应用开发框架是利用这种设计模式的ADF的一个例子。

[0011] Oracle ADF包括基于标准的Java Server Faces(JSF)部件库，具有内置的HTML5和Ajax功能。利用这些部件，web部署的用户界面可以利用先前为胖客户端应用预留的功能和交互级别来开发。部件在容易使用的部件集合中提供数据交互、数据可视化和封装的浏览器侧操作，这使得富客户端应用的开发比以前更容易。Oracle ADF还提供了数据绑定框架，它通过IDE中简单的拖放操作简化了到业务服务的绑定。这是在保持企业服务与消耗接口的独立性的同时进行的。利用这种框架，UI开发人员与业务服务层的底层实现绝缘。这使得构建UI的过程真正与业务服务层的实现脱钩，从而更好地定位要在面向服务的体系架构中实现的应用。

[0012] 因此，所期望的是解决与利用应用开发框架构建应用用户界面相关的问题，其中一些可在本文中讨论。此外，所期望的是减少与利用应用开发框架构建应用用户界面相关的缺点，其中一些可在本文中讨论。

发明内容

[0013] 为了至少提供对主题的基本理解，本公开内容的以下部分给出了对在本公开内容中发现的一个或多个创新、实施例和/或例子的简化概述。本概述不是要提供对任何特定实施例或例子的广泛综述。此外，本概述不是要识别实施例或例子的关键/重要元素或者要描绘本公开内容的主题的范围。因此，本概述的一个目的可以是以简化的形式给出在本公开内容中发现的一些创新、实施例和/或例子，作为随后给出的更详细描述的前言。

[0014] 在各种实施例中，公开了允许开发人员对利用桌面集成框架开发的文档的运行时定制的方法、系统和非暂态计算机可读介质。工作簿元数据是描述给定的工作簿如何与特定的web应用集成的一组信息。当工作簿被发布时，元数据可被写入所发布的工作簿的本地高速缓存以及工作簿定义文件中。元数据管理可以被元数据服务处理，从而允许所发布的工作簿的更新和定制独立于所发布的工作簿中的本地高速缓存和工作簿定义文件。

[0015] 在各种实施例中，用于创建为基于web的应用充当用户界面的桌面应用的文档的方法包括从第一应用接收识别文档的信息，其中所述文档的内容是以与第二应用关联的原生应用格式创建的。确定与用于所述文档的元数据关联的规则集，元数据提供所述文档的内容的一部分与和由与第一应用通信的一个或多个基于web的应用提供的数据模型关联的数据之间的链接。基于所确定的配置要应用到用于所述文档的元数据的定制的规则集，生成用于所述文档的元数据。用于所述文档的元数据被传送到第二应用，使得该文档充当用于一个或多个基于web的应用的用户界面。

[0016] 确定与用于所述文档的元数据关联的规则集可以包括确定规定要使用的用于该文档的元数据的版本的至少一个规则。确定与用于所述文档的元数据关联的规则集可以包括基于与该文档的用户关联的角色确定规定要使用的用于该文档的元数据的至少一个规则。确定与用于所述文档的元数据关联的规则集可以包括基于时间或日期信息确定规定要使用的用于该文档的元数据的至少一个规则。确定与用于所述文档的元数据关联的规则集可以包括基于组织的部门确定规定要使用的用于该文档的元数据的至少一个规则。确定与用于所述文档的元数据关联的规则集可以包括从元数据管理服务器检索一个或多个规则。

[0017] 在一些实施例中，基于所确定的配置要应用到用于所述文档的元数据的定制的规则集生成用于该文档的元数据可以包括对用于该文档的基本元数据应用多个元数据定制。

[0018] 在一种实施例中，提供了一种存储计算机可执行代码的非暂态计算机可读介质，所述代码用于创建为基于web的应用充当用户界面的桌面应用的文档。该非暂态计算机可读介质包括用于从第一应用接收识别其内容以与第二应用关联的原生应用格式创建的文档的信息的代码，用于确定与用于所述文档的元数据关联的规则集的代码，元数据提供所述文档的内容的一部分与和由与第一应用通信的一个或多个基于web的应用提供的数据模型关联的数据之间的链接，用于基于所确定的配置要应用到用于所述文档的元数据的定制的规则集生成用于所述文档的元数据的代码，以及用于将用于所述文档的元数据传送到第二应用使得该文档充当用于一个或多个基于web的应用的用户界面的代码。

[0019] 在一种实施例中，提供了用于创建为基于web的应用充当用户界面的桌面应用的文档的系统。该系统包括处理器和存储指令集的存储器，指令集当被处理器执行时，配置处理器从第一应用接收识别其内容以与第二应用关联的原生应用格式创建的文档的信息，确定与用于所述文档的元数据关联的规则集，元数据提供所述文档的内容的一部分与和由与

第一应用通信的一个或多个基于web的应用提供的数据模型关联的数据之间的链接，基于所确定的配置要应用到用于所述文档的元数据的定制的规则集，生成用于所述文档的元数据，以及将用于所述文档的元数据传送到第二应用，使得该文档充当用于一个或多个基于web的应用的用户界面。

[0020] 通过参考本公开内容的剩余部分、任何附图和权利要求，除了以上部分，对本公开内容的主题的本质和等同物(以及所提供的任何固有的或明确的优点和改进)的进一步理解也应当被认识到。

附图说明

[0021] 为了合理地描述和说明在本公开内容中发现的那些创新、实施例和/或例子，可以参考一个或多个附图。被用来描述一个或多个附图的附加细节或例子不应当被认为是对在本公开内容中给出的任何所要求保护的发明、任何目前描述的实施例和/或例子或者目前被认为是任何创新的最佳模式的限制。

[0022] 图1是示出在根据本发明的一种实施例中的应用开发框架(ADF)的框图。

[0023] 图2是示出在根据本发明的一种实施例中、用于图1的ADF的桌面集成框架的框图。

[0024] 图3是在根据本发明的一种实施例中、用于利用图2的桌面集成框架设计文档的方法的流程图。

[0025] 图4是在根据本发明的一种实施例中、用于利用图2的桌面集成框架与文档交互的方法的流程图。

[0026] 图5是在根据本发明的一种实施例中、其视图可被底层数据模型驱动的文档部件的屏幕截图的说明。

[0027] 图6是在根据本发明的一种实施例中、用于利用图2的桌面集成框架设计部件的模型驱动方面的方法的流程图。

[0028] 图7是示出在根据本发明的一种实施例中、提供元数据管理的用于图2的ADF的桌面集成框架的框图。

[0029] 图8是示出在根据本发明的一种实施例中、在图2的桌面集成框架之间提供元数据管理的交互的框图。

[0030] 图9是在根据本发明的一种实施例中、用于利用具有元数据管理的图2的桌面集成框架运行文档的方法的流程图。

[0031] 图10是在根据本发明的一种实施例中、用于定制元数据的用户界面的屏幕截图。

[0032] 图11绘出了用于实现其中一种实施例的分布式系统的简化图。

[0033] 图12示出了本发明的各种实施例可以在其中实现的示例性计算机系统。

具体实施方式

[0034] 在下面的描述中，为了解释而阐述具体的细节，以便提供对本发明的实施例的透彻理解。但是，将显而易见的是，各种实施例可以在没有这些特定细节的情况下实践。附图和描述并不旨在是限制性的。

[0035] 介绍

[0036] Java EE是构成许多当今企业应用的基础的标准的、健壮的、可扩展的和安全的平

台。Java EE提供了用于利用Java语言构建多层应用程序的规范集。在过去，在应用的健壮本质与实现它所需的复杂性之间存在直接的关联。但是，随着ADF的出现，诸如Oracle ADF，极其丰富的Java EE应用的实现可以通过遵守标准模式和实践以大大减少的工作量来提供。

[0037] 随着组织构建使用面向服务的体系架构(SOA)原理的复合应用的增加的需求，开发人员不得不创建极其敏捷的应用。在敏捷的应用中实现这些最佳实践通常涉及编写显著数量的基础设施代码，从而对开发人员构建其第一个Java EE应用添加了另一个障碍。除了提供健壮的、高性能的和可维护的应用-Oracle ADF还提供基础设施代码来实现基于敏捷的SOA的应用，由此去除了组织“自己滚(rolling their own)”中所涉及的劳动。

[0038] Oracle ADF还通过Oracle JDeveloper 11g开发工具向Java EE开发提供了可视的和可声明的方法。Oracle ADF实现模型-视图-控制器设计模式，并且利用对诸如对象/关系映射、数据持久化、可重用的控制器层、丰富的Web用户界面框架、到UI的数据绑定、安全性和定制之类的领域的解决方案提供覆盖这种体系架构的所有层的集成解决方案。超出核心基于Web的MVC方法，ADF还与Oracle SOA和WebCenter Portal框架集成，从而简化了完整的复合应用的创建。

[0039] 例如，Oracle ADF通过将服务接口耦合到ADF中的内置业务服务而使得容易开发将数据作为服务暴露的敏捷应用。业务服务实现细节的这种分离在Oracle ADF中是经由元数据执行的。这种元数据驱动的体系架构的使用使得应用开发人员能够专注于业务逻辑和用户体验，而不是服务如何被访问的细节。

[0040] Oracle ADF在ADF模型层中的元数据中存储这些服务的实现细节。这使得开发人员无需修改用户界面就可以交换服务，从而使得应用极其敏捷。此外，创建用户界面的开发人员不需要受业务服务访问细节的打扰。相反，开发人员可以专注于开发应用接口和交互逻辑。创建用户体验可以像将期望的业务服务拖放到可视化页面设计器上并指示什么类型的部件应当代表那个数据一样简单。

[0041] 图1是示出在根据本发明的一种实施例中的应用开发框架(ADF)100的框图。图1是可以结合在本公开内容中给出的一个或多个发明的各种实施例或实现的系统的简化说明。图1可以仅仅是说明本文公开的发明的实施例或实现不应当限制如在权利要求中阐述的任何发明的范围。本领域普通技术人员可以通过本公开内容和本文给出的示教认识到作为附图中所示的其它实施例或实现的其它变体、修改和/或备选方法。

[0042] 作为一个例子，ADF 100可以体现为Oracle ADF。因此，ADF100是基于模型-视图-控制器(MVC)设计模式。MVC应用被分成：1)处理与数据源的交互并且运行业务逻辑的模型层，2)处理应用用户界面的视图层，和3)管理应用流并且充当模型和视图层之间的接口的控制器。将应用分成这三个层简化了部件跨应用的维护和重用。每个层与其它层的独立性导致松散耦合的、面向服务的体系架构(SOA)。

[0043] 在这种实施例中，构成企业应用的模块被示为在ADF 100中，以表示模块是利用ADF开发的，然后在ADF 100的语境中执行。为了简明，未示出ADF的各种内部细节，假设应用是利用Java编程语言和作为JDeveloper 10.1.3的部分可用的Oracle ADF开发的，其中JDeveloper 10.1.3是可从Oracle公司获得的开发工具。但是，以下描述的本发明的特征可以利用编程语言及应用开发框架的任何期望组合来实现，如通过阅读本文提供的公开内容

将对相关领域的技术人员显而易见的。

[0044] 在各种实施例中,ADF 100是以多层形式开发的应用,每一层包含根据预定义的规范实现期望逻辑的代码模块/文件。因此,在一种实施例中,ADF 100使应用作为四层被开发:包含提供应用的用户界面的代码模块/文件的视图层110,包含控制应用流的代码模块的控制器层120,包含为底层数据提供抽象层的数据/代码模块的模型层130,以及包含对来自各种源的数据提供访问并且处理业务逻辑的代码模块的业务服务层140。

[0045] Oracle ADF让开发人员选择当实现每个层时他们更喜欢使用的技术。图1示出了当构建Oracle ADF应用时开发人员可用的各种选项。集成Java EE应用的各种部件并使开发如此灵活的胶水(glue)是Oracle ADF模型层。EJB、Web服务、JavaBeans、JPA/EclipseLink/TopLink对象以及许多其它全都可以被用作用于Oracle ADF模型的业务服务。视图层可以包括利用JSF、Desktop Swing应用和MS Office前端实现的、基于Web的接口,以及用于移动设备的接口。

[0046] 可以认识到,利用这种分层方法的应用的开发常常简化了部件/代码模块跨各种应用的维护和重用。另外,每一层与其它层的独立性导致松散耦合的、面向服务的体系架构(SOA),这在多个/不同系统上部署所开发的业务/企业应用时可能是期望的。

[0047] 一方面,视图层110代表被开发的应用的用户界面。视图层110被示为具有桌面、移动和基于浏览器的视图,其中每个视图提供用户界面的全部或一部分,并且可以以对应于视图类型的各种方式来访问。例如,网页可以由应用响应于接收到包含对应URL的客户端请求而被发送。然后,网页可以由与发出请求的客户端系统关联的显示单元(未示出)上的浏览器显示,由此使发出请求的客户端系统的用户能够与企业应用交互。Oracle ADF支持对业务服务的多通道访问,从而允许业务服务的重用和从Web客户端、客户端-服务器摆动(swing)基于桌面的应用、Microsoft Excel电子表格、诸如智能电话的移动设备等等进行访问。

[0048] 构成视图层(诸如网页)的代码文件/模块可以利用超文本标记语言(HTML)、Java服务器页(JSP)和Java Server Faces(JSF)当中一个或多个来实现。作为替代,用户界面可以利用诸如Swing的Java部件和/或可扩展标记语言(XML)来实现。如进一步指出的,用户界面可以充分利用用户对诸如Microsoft的Word和Excel的桌面应用的经验和熟悉。

[0049] 如以上所指出的,相关的用户开发的代码/数据模块在每一层中提供。但是,每一层通常包含由ADF 100提供的其它预定义的代码/数据模块。一些预定义的模块可以在开发过程中被使用,例如,作为用于开发网页的模板,用于在所开发的代码中包括期望的功能,等等。其它预定义的模块(诸如URL重写模块)可以连同所开发的应用一起部署,并且可以在企业应用的执行期间向用户提供附加的功能(所请求的URL到内部名称的映射)。

[0050] 控制器层120包含控制应用的流的代码模块/文件。每个控制器对象包含根据在视图层110中给出信息的期望方式实现的软件指令和/或数据。期望的方式可以包括当另一网页中的链接被用户点击/选择时特定的网页被显示,当错误在执行期间发生时页面被显示,指示特定数据要被存储/检索,等等。

[0051] 在一方面,控制器层120管理应用流并处理用户输入。例如,当搜索按钮在页面上被点击时,控制器确定要执行什么动作(进行搜索)和导航到什么地方(结果页面)。在JDeveloper中对于基于web的应用有两组控制器选项:标准JSF控制器或扩展JSF控制器功

能的ADF控制器。无论使用哪种控制器，应用流通常都是通过在图上安排页面和导航规则来设计的。应用的流可以被分解成更小的、可重用的任务流；包括非可视部件，诸如方法调用和流中的决策点；并创建在单个包含页面的区域内运行的“页面片段”流。

[0052] 构成控制器层120的代码模块/文件常常被实现为Java servlets(Java小服务程序)，其接收客户端请求并发送期望的网页作为对应的响应。控制器对象还可以被实现为，例如，Apache Jakarta Struts控制器或根据JSF标准。

[0053] 模型层130包含将各种业务服务连接到在其它层中使用它们的对象的数据/代码模块，诸如连接到上面讨论的控制器对象或如图所示直接连接到桌面应用。模型层130的每个抽象数据对象提供可被用来访问在底层业务服务层140中执行的任何类型的业务服务的对应接口。数据对象可以抽象来自客户端的服务的业务服务实现细节和/或向视图部件暴露数据控制方法/属性，从而提供视图与数据层的分离。

[0054] 在一方面，模型层130包括两个部件，数据控制和数据绑定，它们利用元数据文件来定义接口。数据控件抽象来自客户端的业务服务实现细节。数据绑定向UI部件暴露数据控制方法和属性，从而提供视图和模型的彻底分离。由于模型层的元数据体系架构，当将任何类型的业务服务层实现绑定到视图和控制器层时，开发人员得到相同的开发体验。

[0055] Oracle ADF贯穿整个开发过程强调声明式编程范式的使用，以允许用户专注于应用创建的逻辑，而不必进入实现细节。在高级别，用于Fusion(融合)web应用的开发过程通常涉及创建应用工作区。利用向导，用于由开发人员选择的技术所需的库和配置会自动添加并且应用被构造成具有程序包和目录的项目。

[0056] 通过建模数据库对象，任何数据库的在线数据库或离线副本都可以被创建、定义被编辑，以及模式被更新。然后，利用UML建模器，可以为应用创建用例。应用控制和导航也可以被设计。Diagrammers可以被用来可视确定应用控制的流和导航。描述流的底层XML可以被自动创建。资源库可以被用来允许开发人员通过简单地将其拖放到应用中来查看并使用导入的库。从数据库表，可以利用向导或对话框创建实体对象。从这些实体对象，创建要在应用中被页面使用的视图对象。验证规则和其它类型的业务逻辑可以被实现。

[0057] 在这个例子中，业务服务层140管理与数据持久化层的交互。它提供诸如数据持久化、对象/关系映射、事务管理和业务逻辑执行等服务。Oracle ADF中的业务服务层可以在下列任何选项中实现：作为简单的Java类、EJB、Web服务、JPA对象和Oracle ADF业务部件。此外，数据可以直接从文件(XML或CSV)以及REST消费。

[0058] 因此，每个业务服务管理与对应的数据持久化层的交互，并且还提供诸如对象/关系映射、事务管理、业务逻辑执行等服务。业务服务层可以利用简单的Java类、企业Java Beans、web服务等等当中一个或多个来实现。

[0059] 业务部件代表利用例如Oracle ADF业务部件实现的业务服务，以便提供与数据库、Web服务、遗留系统、应用服务器等的交互。在一种实施例中，业务服务层140的业务部件包含应用模块、视图/查询对象和实体对象的混合，它们合作以提供业务服务实现。应用模块可以是UI客户端为了对应用/事务数据起作用而与其通信的事务部件/代码模块。应用模块可以提供可更新的数据模型以及还有与用户事务相关的过程/功能(通常被称为服务方法)。

[0060] 实体对象可以代表数据库表中的对应行并且简化存储在对应行中的数据的操纵

(更新、删除,等等)。实体对象常常封装用于对应行的业务逻辑,以确保期望的业务规则始终如一地被执行。实体对象还可以与其它实体对象关联,以反映存储在底层数据库中的行之间存在的关系。

[0061] 桌面集成

[0062] ADF桌面集成将Oracle应用开发框架扩展到像Microsoft Excel的桌面应用的世界中。应用开发人员可以快速开发集成的文档,诸如电子表格和其它基于桌面的应用的文档,以允许用户访问和编辑关键业务数据。这种框架与每个web应用的安全性和业务逻辑基础设施无缝地集成。它还允许最终用户无需到网络的实时连接就编辑他们的数据。一旦重新连接,ADF桌面集成就可以对照应用的后端透明地上传和验证所有用户更改。因此,ADF桌面集成允许开发人员将由基于web的应用提供的功能扩展到桌面应用。最终用户也可能更喜欢ADF桌面集成,因为它在用户偏好的桌面应用中提供了承担信息管理任务的熟悉的用户界面,诸如容易和无缝地执行复杂计算或上传大量数据。

[0063] 图2是示出在根据本发明的一种实施例中、用于图1的ADF 100的桌面集成框架200的框图。桌面集成框架200可以结合本公开内容中给出的一个或多个发明的各种实施例或实现。桌面集成框架200仅仅是说明本文公开的发明的实施例或实现不应当限制如在权利要求中阐述的任何发明的范围。本领域普通技术人员可以通过本公开内容和本文给出的示教认识到作为附图中所示的其它实施例或实现的其它变体、修改和/或备选方法。

[0064] 在这个例子中,桌面集成框架200包括客户端计算机系统210和服务器计算机系统220。客户端计算机系统210代表被配置为提供对应用230的访问和/或托管应用230的硬件和/或软件元件。客户端计算机系统210可以被体现为个人计算机系统、膝上型计算机、平板计算机、移动设备,等等。客户端计算机系统210可以包括在一个或多个计算机上执行的一个或多个操作系统、应用、浏览器,等等。客户端计算机系统210可以仅仅是说明本文公开的发明的实施例或实现不应当限制如在权利要求中阐述的任何发明的范围。本领域普通技术人员可以通过本公开内容和本文给出的示教认识到作为附图中所示的其它实施例或实现的其它变体、修改和/或备选方法。

[0065] 应用230代表允许用户生成、编辑或以其它方式与文档进行交互的一个或多个软件元件。应用230的一些例子是文本编辑器、文字处理应用、电子表格应用、图像编辑和操纵程序,等等。在各种实施例中,桌面集成框架200利用特定于桌面应用,诸如像Microsoft Word和Microsoft Excel的Microsoft Office产品,的配置操作。

[0066] 应用230还包括或以其它方式与ADF-DI客户端部件240的通信并创建文档250。ADF-DI客户端部件240代表将由基于web的或其它网络可访问的应用提供的功能扩展到应用230的一个或多个软件元件。例如,ADF-DI客户端部件240允许最终用户利用与应用230关联的熟悉的用户界面来利用文档250承担通常通过访问服务器计算机系统220执行的信息管理任务。这些任务可以由服务器计算机系统220托管的、基于web的或其它网络可访问的应用来执行或处理。在各种实施例中,由在应用230中执行的这种信息管理任务操纵的数据与服务器计算机系统220同步。

[0067] 文档250代表电子信息的一个或多个计算机数据文件或单元。文档250可以包括文本、图像、音频、视频,以及其它多媒体信息。文档250还可以与特定于应用230的元数据关联。文档250(或应用230)可以提供用于创建、交互和管理与文档250关联的内容的原生功

能。在各个方面，应用230提供用于与应用230的功能或文档250的内容交互的一个或多个接口。

[0068] 服务器计算机系统220代表被配置为提供对应用服务器260的访问和/或托管应用服务器260的硬件和/或软件元件。服务器计算机系统220可以被体现为本地服务器计算机系统、云服务，等等。服务器计算机系统220可以包括在一个或多个计算机上执行的一个或多个操作系统、服务器、服务、应用，等等。服务器计算机系统220仅仅是说明本文公开的发明的实施例或实现不应当限制如在权利要求中阐述的任何发明的范围。本领域普通技术人员可以通过本公开内容和本文给出的示教认识到作为附图中所示的其它实施例或实现的其它变体、修改和/或备选方法。

[0069] 应用服务器260代表允许用户与基于web的或基于网络的应用进行交互的一个或多个软件元件。应用服务器260的一些例子是或者不考虑应用的功能是什么就提供创建应用-服务器实现的通用方法的软件框架，或者是特定实现实例的服务器部分。在各种实施例中，应用服务器260利用特定于Java平台企业版或定义一组核心API和Java应用服务器的特征的Java EE的配置操作。应用服务器260可以包括servlets、JavaServer Pages、企业JavaBeans，等等。应用服务器260仅仅是说明本文公开的发明的实施例或实现不应当限制如在权利要求中阐述的任何发明的范围。本领域普通技术人员可以通过本公开内容和本文给出的示教认识到作为附图中所示的其它实施例或实现的其它变体、修改和/或备选方法。

[0070] ADF-DI服务器部件270代表一个或多个服务器部件，诸如应用服务器260的部分。一般而言，ADF-DI客户端部件240既充当视图层110又充当控制层120，并且与部分地充当模型层130的ADF-DI服务器部件270通信，以同步数据并执行由应用服务器260托管的应用中的业务逻辑或者利用ADF模型280与应用服务器260通信。如以上所讨论的，模型层130代表与由应用230内的ADF-DI客户端部件240给出的当前视图相关的数据值，连同模型级的业务规则、安全性和对照数据值使用的应用逻辑。在这个例子中，ADF-DI客户端部件240和ADF-DI服务器部件270允许最终用户利用与应用230关联的熟悉的用户界面来利用文档250承担访问ADF模型280的视图/控制器任务。

[0071] 在一方面，开发人员利用ADF-DI客户端部件240在应用230中工作，以便在设计模式下创建文档250。开发人员利用应用230的原生工具以期望的方式构造和格式化文档250。然后，开发人员可以利用ADF-DI客户端部件240向文档250添加部件，以便将文档250与与应用服务器260集成。部件的一些例子是输入部件(例如，表单部件)、输出部件、标签、列表、按钮、图像、表格，等等。

[0072] 然后，被添加到文档250的部件被映射到由ADF-DI服务器部件270或通过其提供的对应的模型。例如，文本框部件可以被映射，以便在文档250中提供到由模型层130暴露的ADF-DI服务器部件270或通过其提供的人模型的姓名属性的输入/输出机制。

[0073] 在一方面，部件是提供可被许多应用使用的(或可以被同一应用多次使用的)功能的可重用实体。部件可以嵌在文档250中并且它们可以具有零个或多个可视表示。不具有可视表示的部件不显示，但可以提供一些其它功能。部件通常提供一个或多个接口，诸如编程接口、数据绑定接口，以及可视接口。部件200可以具有一个或多个可视表示。如下面进一步描述的，部件可以具有被底层模型驱动的可视表示。

[0074] 在一方面，部件可以在设计时指定任意数量的视图，其中任何一个都可以在运行

时被显示。视图组件是在运行时实际显示的视图集。对于应用或部件，视图组件包括在某个时间点被选择用于显示的视图合成物中的视图。

[0075] 一旦所有期望的部件都被包括并映射到可让应用服务器260和ADF模型280访问的数据和/或模型元数据，文档250就可以“被发布”或者以其它方式在应用服务器260上可用。应用服务器260可以提供到所发布的文档的下载链接，从而使用户能够经由浏览器访问文档并开始在应用230中工作，以查看、创建和/或操纵数据，诸如存储在服务器计算机系统220可访问的数据库中的数据。在各种实施例中，所发布的文档与文档元数据分开存储，其中文档元数据定义部件、数据映射，以及任何开发人员与文档关联的逻辑。在一些实施例中，所发布的文档包括所有文档元数据。

[0076] 图3是在根据本发明的一种实施例中、用于利用图2的桌面集成框架200设计文档的方法300的流程图。当被逻辑机器，诸如计算机系统或信息处理设备，的中央处理单元(CPU或处理器)执行时，图3中所绘的方法300中的实现或处理可以由软件(例如，指令或代码模块)执行，由电子设备或专用集成电路的硬件部件执行，或者由软件和硬件元件的组合执行。图3中所绘的方法300开始于步骤310。

[0077] 在步骤320中，文档被接收。一般而言，文档是由原生地创建这种文档的应用或者利用以原生格式创建文档的库创建的。根据图2，用户可以在应用230中打开现有的文档或创建新文档，诸如在Microsoft Excel中创建新的电子表格。文档开发人员可以利用原生或非原生工具以任何期望的方式编辑、构造或格式化文档。

[0078] 在步骤330中，文档元数据被接收。文档元数据包括被ADF-DI客户端部件240用来基于从应用服务器260获得的数据呈现文档的内容的信息。在一方面，文档元数据识别包括在文档中的每个部件以及部件如何绑定到在业务服务层140中可用的数据或元数据。文档元数据还可以提供访问信息、静态数据，以及由开发人员提供的任何其它逻辑或数据操纵信息。如以上所讨论的，开发人员可以利用ADF-DI客户端部件240向文档250添加部件，以便将文档250与应用服务器260集成。

[0079] 在各种实施例中，ADF-DI客户端部件240提供了表达构建器，其包括用于可被添加到文档的一些或全部部件的一个或多个特性。每个特性可以定义其对应的部件的行为的一方面。例如，特性可以指定被映射到部件的模型或对象和/或对应于该部件的模型或对象的一个或多个属性。在另一个例子中，特性可以指定文档250的各方面，诸如表列标题、工作表格式栏命令，等等。

[0080] 在步骤340中，文档和文档元数据被发布。如以上所讨论的，所发布的文档可以与文档元数据分开存储。在一些实施例中，所发布的文档可以包括文档元数据的全部或部分。一般而言，所发布的文档包括至少使ADF-DI客户端部件240能够初始化文档并且从ADF-DI服务器部件270请求附加信息以便在运行时为用户呈现文档250的内容的足够多元数据。图3在步骤350结束。

[0081] 在运行时期间，用户下载所发布的文档250并且利用应用230打开它。在一种实施例中，ADF-DI客户端部件240已作为应用插件或模块被安装。然后，ADF-DI客户端部件240可以检测到文档250已被创作为包括框架部件并且联系ADF-DI服务器部件270，以请求文档元数据、实际数据，以及需要被执行的任何逻辑。例如，ADF-DI客户端部件240可以首先从或通过从ADF-DI服务器部件270检索定义哪些部件将被包括和在哪里包括它们的文档元数据。

ADF-DI客户端部件240可以从或通过从ADF-DI服务器部件270检索来自ADF模型280的、被选部件将对其使用或以其它方式对其操作的数据。ADF-DI客户端部件240还可以从或通过从ADF-DI服务器部件270检索与文档250关联的任何逻辑。最后，ADF-DI客户端部件240可以随后利用文档元数据、实际数据和逻辑呈现文档250的内容。

[0082] 因此，用户可以检索文档模板并且基于由ADF-DI客户端部件240执行的处理和从应用服务器260获得的数据让文档内容被自动更新和格式化。然后，用户可以利用与应用230关联的熟悉的用户界面来利用文档250承担任务。

[0083] 在各方面，当用户与文档250交互或操纵其时，ADF-DI客户端部件240和ADF-DI服务器部件270可以保持通信，以相应地发送和接收更新。在文档250的一个或多个部件中对模型层130中对应模型的数据所作的改变可以在ADF模型280中持久化。

[0084] 图4是在根据本发明的一种实施例中、用于利用图2的桌面集成框架200与文档交互的方法400的流程图。当被逻辑机器，诸如计算机系统或信息处理设备，的中央处理单元(CPU或处理器)执行时，图4中所绘的方法400中的实现或处理可以由软件(例如，指令或代码模块)执行，由电子设备或专用集成电路的硬件部件执行，或者由软件和硬件元件的组合执行。图4中所绘的方法400开始于步骤410。

[0085] 在步骤420中，文档被接收。根据图2，用户可以与应用服务器260交互，以检索期望的文档，诸如通过点击文档链接。该文档可以被下载或以其它方式传送到客户端计算机系统210并在应用230中打开。

[0086] 在步骤430中，文档元数据被处理。如以上所讨论的，文档元数据包括被ADF-DI客户端部件240用来基于从应用服务器260获得的数据呈现文档的内容的信息。因此，ADF-DI客户端部件240确定哪些部件将被添加到文档和在哪里添加。ADF-DI客户端部件240还确定每个部件使用要什么数据，以及应用由开发人员定义的任何逻辑。

[0087] 在步骤440中，文档数据被处理。如以上所讨论的，所发布的文档可以与被文档使用的文档元数据和实际数据分开存储。一般而言，所发布的文档包括至少使ADF-DI客户端部件240能够初始化文档并且从ADF-DI服务器部件270请求附加信息以便为用户呈现文档250的内容的足够多元数据。

[0088] 在步骤450中，文档被呈现。如以上所讨论的，ADF-DI客户端部件240可以从或通过从ADF-DI服务器部件270检索来自ADF模型280的、被选部件将对其使用或以其它方式对其操作的数据。ADF-DI客户端部件240还可以从或通过从ADF-DI服务器部件270检索与文档250关联的任何逻辑。最后，ADF-DI客户端部件240可以随后利用文档元数据、实际数据和逻辑呈现文档250的内容。

[0089] 在步骤460中，确定对文档的更新是否存在。对于对文档的更新为什么可以存在，可以有各种原因。当用户与文档250交互或操纵其时，ADF-DI客户端部件240和ADF-DI服务器部件270可以保持通信，以相应地发送和接收更新。在文档250的一个或多个部件中对模型层130中对应模型的数据所作的改变可以在ADF模型280中持久化。在一些实施例中，由用户进行的交互可能需要新的数据集。因而，方法400的流程返回到步骤440，以处理在步骤450中呈现的附加文档数据。图4在步骤470结束。

[0090] 模型驱动的方面

[0091] 在各种实施例中，桌面集成框架200允许开发人员包括其视图和数据由对应的模

型或模型属性驱动的部件文档250。在一方面，部件可以提供工具提示，这包括关于当用户的鼠标在数据的一部分之上时或者当数据被选择时被显示部件的所呈现内容的有用信息。工具提示可以经由模型或对象来定义，并且在文档的部件被呈现时自动配置。在另一方面，列表部件的元素可以利用与模型或对象关联的一个或多个属性的预先存在的值来填充。因此，不需要开发人员指定当文档被呈现时部件给出的视图中的值。

[0092] 图5是在根据本发明的一种实施例中、其视图可被底层数据模型驱动的文档部件的屏幕截图500的说明。在这个例子中，树510提供可以在文档250中给出的一个或多个视图的列表。在该例中，EmpView 520包括一个或多个部件，诸如被标记为“Empno”、“Ename”、“Job”、“Mgr”等等的文字、列表、图像、日期，等等。

[0093] 在一方面，与部件530关联的雇员对象的底层数据模型或属性被标记为“Hiredate”。该数据模型或属性被配置为存储与给定雇员被组织雇用的日期关联的日期值。在各种实施例中，部件530的视图或行为可以由该属性存储缺乏来自开发人员的进一步配置的日期值的事实来驱动。例如，在ADF-DI客户端部件240呈现具有对应于部件530的单元的Excel工作簿时，ADF-DI客户端部件240可以配置该单元，使得，当用户选择存储表示雇员雇用日期的值时，日期弹出窗口在Excel的原生特征之内或之外提供，从而允许用户选择雇用的新日期或者修改雇用的现有日期。

[0094] 在另一方面，与部件540关联的雇员对象的底层数据模型或属性被标记为“Deptno”。该数据模型或属性被配置为存储与雇员关联的部门或团队的标识符。在各种实施例中，部件540的视图或行为可以由该属性存储缺乏来自开发人员的进一步配置的、在数据模型中指定的多个预定值之一的事实来驱动。例如，在ADF-DI客户端部件240呈现具有对应于部件540的单元的Excel工作簿时，ADF-DI客户端部件240可以配置该单元，使得，当用户选择存储表示该雇员被分配到的部门或团队的值时，下拉列表在Excel的原生特征之内或之外提供，从而允许用户从得自数据模型的部门或团队的预定列表中选择。

[0095] 图6是在根据本发明的一种实施例中、用于利用图2的桌面集成框架200设计部件的模型驱动方面的方法600的流程图。当被逻辑机器，诸如计算机系统或信息处理设备，的中央处理单元(CPU或处理器)执行时，图6中所绘的方法600中的实现或处理可以由软件(例如，指令或代码模块)执行，由电子设备或专用集成电路的硬件部件执行，或者由软件和硬件元件的组合执行。图6中所绘的方法600开始于步骤610。

[0096] 在步骤620中，部件规范被接收。一般而言，部件规范指的是规定部件如何被定义的信息。根据图2，开发人员可以在应用230中打开现有的文档或创建新文档，诸如在Microsoft Excel中创建新的电子表格。然后，开发人员可以利用原生或非原生工具以任何期望的方式编辑、构造或格式化文档。此外，开发人员可以从多个预定的部件中选择并且将那些部件添加到文档250。

[0097] 在步骤630中，数据绑定规范被接收。一般而言，数据绑定规范指的是指定部件如何与数据交互的信息，诸如其来源，等等。在各种实施例中，除了部件规范，ADF-DI客户端部件240还利用这种信息进一步配置部件。在一方面，ADF-DI客户端部件240基于与部件关联的一个或多个模型或对象识别每个部件将如何给出一个或多个视图。ADF-DI客户端部件240可以与ADF-DI服务器部件270交互，以检索模型的各方面、值，等等，以配置部件、任何关联的视图，以及关联的行为。在各种实施例中，ADF-DI客户端部件240可以响应于用户使用

上面讨论的表达构建器而接收部件规范和数据绑定规范。

[0098] 在步骤640中，部件的模型驱动方面被确定。ADF-DI客户端部件240可以与ADF-DI服务器部件270交互，以检索被认为是ADF模型280中的属性特性或暗示的模型的各方面、值，等等，以配置部件、任何关联的视图，以及关联的行为。配置信息可以存储在可与所发布的文档关联的文档元数据中。

[0099] 在一个例子中，向用户给出的标签常常与对应于该标签的数据对象的名称不同。例如，如果存在称为“EmpName”的属性，但在UI中，开发人员期望显示“雇员姓名”。大多数UI框架允许开发人员指定用户友好的标签。但是，在需要出现的每个地方指定用户友好的标签是低效的。“模型驱动”方法是在模型级将用户友好的标签与EmpName属性关联。然后，想要呈现EmpName的每个UI元素(页面，工作表，等等)将在设计时间接地“引用指”EmpName标签，并在运行时动态获取它。该方法适用于数据对象的各种不同的潜在属性。其它的例子包括“只读”、“强制”，等等。图6在步骤650结束。

[0100] 元数据管理

[0101] 在一些实施例中，定义工作簿集成的元数据可以由元数据服务进行管理。图7是示出在根据本发明的一种实施例中、提供元数据管理的用于图2的ADF的桌面集成框架200的框图。在这个例子中，文档元数据可以在设计时和运行时都被存储在内部储存库295中。在储存库295中，元数据可以在文档250发布之后进行定制，诸如在客户站点。ADF-DI服务器部件270与元数据服务器290通信，以基于管理规则集提供一个或多个版本的文档元数据。管理规则可以定义指示文档利用什么版本的元数据的标准和/或条件，诸如时间、日期、用户角色、组织，等等。这允许工作簿定义方案随时间而改变，诸如当更多的特征被添加时。

[0102] 因此，在各种实施例中，桌面集成框架200允许开发人员指定工作簿是可定制的。开发人员可以配置工作簿元数据在明确定义的工作簿的初始化点从元数据服务器290检索。工作簿元数据可以为当前用户的定制环境定制。在另一方面，运行时工作簿可以利用从元数据服务器290检索的定制的元数据呈现。

[0103] 在各种实施例中，ADF-DI客户端部件240向ADF-DI服务器部件270转发该文档元数据，ADF-DI服务器部件270又具有存储工作簿定义文件(基本文档元数据)和定制的元数据服务器290。开发人员可以选择使用基于文件的或数据库储存库，用于管理基本文档和/或定制。

[0104] 图8是示出在根据本发明的一种实施例中、在图7的桌面集成框架之间提供元数据管理的交互的框图。在方框810中，启用定制的工作簿以这样一种方式发布，使得推断出的元数据路径可以被用来在类路径(classpath)上定位对应的工作簿定义文件。例如，给定设计时工作簿：

[0105] “/PROJECT_ROOT/src/apps/workbooks/myDTWorkbook”，当其被发布时，其工作薄定义文件在

[0106] “/PROJECT_ROOT/src/apps/workbooks/myDTWorkbook.xlsx-workbooks-definition.xml”生成。

[0107] 当应用被部署时，这个工作簿定义文件可以由类路径储存库利用路径“/apps/workbooks/myDTWorkbook.xlsx-workbook-definition.xml”找出。因此，为了使运行时工作簿可以利用推断出的元数据路径加载其工作簿元数据，它应当被发布到

[0108] “/PROJECT_ROOT/public_html/oracle/apps/workbooks/”文档夹。如以上所讨论的，开发人员可以选择使用基于文件的或数据库储存库，用于管理基本文档和/或定制。

[0109] 在方框820中的工作簿初始化后，ADF-DI客户端部件240向服务器计算机系统220发出对当前工作簿的元数据的请求。在方框830中，ADF-DI服务器部件270利用MDS会话(例如，经由元数据服务器290来自当前ADFContext的标准MDS会话)检索工作簿元数据。基于与MDS会话关联的定制语境，元数据服务器290将任何及所有地址应用到工作簿元数据。在方框820中，定制的工作簿元数据被返回到ADF-DI客户端部件240，并且被写入工作簿的元数据高速缓存。在方框860中，高速缓存的定制的工作簿元数据被用来初始化运行时工作簿。

[0110] 图9是在根据本发明的一种实施例中、用于利用包括一个或多个定制的图2的桌面集成框架200运行文档的方法900的流程图。当被逻辑机器，诸如计算机系统或信息处理设备，的中央处理单元(CPU或处理器)执行时，图9中所绘的方法900中的实现或处理可以由软件(例如，指令或代码模块)执行，由电子设备或专用集成电路的硬件部件执行，或者由软件和硬件元件的组合执行。图9中所绘的方法900开始于步骤910。

[0111] 在步骤920中，文档和初始元数据被接收。例如，用户可以请求从应用服务器260下载所发布的文档250。在步骤930中，元数据定制被请求。在一方面，文档下载可以对文档250完成，在这个时候，文档250可以利用应用230打开。然后，ADF-DI客户端部件240可以检测到文档250需要被呈现。ADF-DI客户端部件240联系ADF-DI服务器部件270并且根据文档的设计获得文档元数据和要被呈现的任何实际数据。

[0112] 在各方面，在工作簿初始化时，ADF-DI客户端部件240向服务器计算机系统220发出对当前工作簿的元数据的请求。ADF-DI服务器部件270从当前ADFContext经由元数据服务器290获得标准MDS会话，并且利用这个MDS会话检索工作簿元数据。基于与MDS会话关联的定制语境，元数据服务器290将任何及所有定制应用到工作簿元数据。定制的工作簿元数据被返回到ADF-DI客户端部件240。定制的工作簿元数据被用来初始化运行时工作簿。

[0113] 在步骤940中，文档和定制的元数据被处理。ADF-DI客户端部件240确定部件的放置并且利用由ADF-DI服务器部件270在步骤950中提供的实际数据将它们作为文档250的内容呈现。图9在步骤960结束。

[0114] 工作簿创作

[0115] 在各种实施例中，定制可以利用各种工具和接口产生。公开了允许最终用户提供利用桌面集成框架开发的文档的运行时定制的方法、系统和非暂态计算机可读介质。工作簿元数据是描述给定的工作簿如何与特定的Web应用集成的信息集。当工作簿被发布时，元数据可被写入所发布的工作簿的本地高速缓存以及工作簿定义文件中。元数据管理可以被元数据服务处理，从而允许所发布的工作簿的更新和定制独立于所发布的工作簿中的本地高速缓存和工作簿定义文件。工作簿定制编辑器被用来执行集成的工作簿的各种定制。

[0116] 图10是在根据本发明的一种实施例中、用于定制元数据的用户界面1000的屏幕截图。在这个例子中，用户界面1000提供从给定的工作簿中选择集成的工作表的能力，查看和选择选定工作表上的部件的能力，或者编辑或删除所选部件的能力。在一些实施例中，用户界面1000可以提供添加新工作表的能力，添加新部件的能力，或者编辑现有部件的高级特性，诸如动作集，的能力。

[0117] 在操作的一个例子中，web应用已售出并安装在Acme公司。Acme的业务分析师，

Ralph,回顾了集成的工作簿并确定需要一些调整。Ralph登录到该web应用并导航到用户界面1000。Ralph决定创建基本工作簿的新定制。Ralph从表中的标准65列删除了5列,因为他知道,Acme公司并不需要/想要那些列。删除节省了定制。匹配与工作簿定制关联的定制语境的定制现在可让用户获得。这种用户将看到具有60列的表。

[0118] 在另一个例子中,用户界面1000构成ADF任务流的一部分,其允许经授权的用户编辑给定的集成工作簿的各方面。如以上所讨论的,用户界面1000可以从选定的工作簿提供集成的工作表的列表。用户界面1000可以包括使用户能够选择或删除工作簿和/或工作表的功能。用户界面1000可以包括显示或以其它方式与选定的工作表关联的部件的列表。用户界面1000可以包括使用户能够选择、删除和编辑部件的功能。用户界面1000可以包括使用户能够编辑部件的特性的功能,诸如输入文本、输出文本、标签、值的列表。例如,文档250的开发人员可以指定部件的哪些方面可被查看(仅视图),诸如部件ID、值、注释,等等,或者可编辑,诸如样式、工具提示、只读、位置。开发人员可以指定部件的哪些方面是可编辑的,诸如对于图像部件-来源,ShortDesc,位置等,对于按钮部件-标签,位置等。在另一方面,开发人员可以指定表部件的哪些方面是可查看的(仅视图)和可编辑的。一些例子包括只读表-只删除部件,不可编辑;或者允许位置、列的编辑(是否删除、编辑、重新排序),单元/标题的样式、工具提示、可见、标题标签的编辑,并且防止特定列的删除。

[0119] 在各方面,主机应用为用户界面1000提供各种信息,诸如要编辑的工作簿(包括元数据路径和工作簿名)和定制语境。此外,主机应用可以负责控制用户界面1000的哪个用户可以编辑哪些(如果有的话)工作簿。

[0120] 结论

[0121] 在一些图中绘出的系统可以在各种配置中提供。在一些实施例中,系统可以被配置为分布式系统,其中系统的一个或多个部件跨云计算系统中的一个或多个网络分布。

[0122] 图11绘出了用于实现其中一种实施例的分布式系统1100的简化图。在所示的实施例中,分布式系统1100包括一个或多个客户端计算设备1102、1104、1106和1108,其被配置为经一个或多个网络1110执行和操作客户端应用,诸如web浏览器、专有客户端(例如,Oracle Form)等等。服务器1112可以经由网络1110与远程客户计算设备1102、1104、1106和1108通信耦合。

[0123] 在各种实施例中,服务器1112可以适于运行由系统的一个或多个部件提供的一个或多个服务或软件应用。在一些实施例中,这些服务可以作为基于web的或云服务或者依据软件即服务(SaaS)模型向客户计算设备1102、1104、1106和/或1108的用户提供。操作客户端计算设备1102、1104、1106和/或1108的用户又可以利用一个或多个客户端应用与服务器1112交互,以利用由这些部件提供的服务。

[0124] 在该图中所绘出的配置中,系统1100的软件部件1118、1120和1122被示为在服务器1112上实现。在其它实施例中,系统1100的一个或多个部件和/或由这些部件提供的服务也可以由一个或多个客户端计算设备1102、1104、1106和/或1108实现。然后,操作客户端计算设备的用户可以利用一个或多个客户端应用来使用由这些部件提供的服务。这些部件可以在硬件、固件、软件或其组合中实现。但应当认识到的是,各种不同的系统配置是可能的,这些配置可以与分布式系统1100不同。因此,图中所示的实施例是用于实现实施例系统的分布式系统的一个例子并且不是要限制。

[0125] 客户端计算设备1102、1104、1106和/或1108可以是便携式手持设备(例如，iPhone®、蜂窝电话、iPad®、计算平板、个人数字助理(PDA))或可穿戴设备(例如，GoogleGlass®头戴式显示器)，运行诸如Microsoft WindowsMobile®的软件，和/或诸如iOS、Windows Phone、Android，BlackBerry 10、Palm OS等各种移动操作系统，并且启用互联网、电子邮件、短消息服务(SMS)、BlackBerry®或其它的通信协议。客户端计算设备可以是通用个人计算机，作为例子，包括运行各种版本的Microsoft Windows®、Apple Macintosh®和/或Linux操作系统的个人计算机和/或便携式计算机。客户端计算设备可以是运行任何各种市售UNIX®或类UNIX操作系统当中任意一种，包括但不限于各种GNU/Linux操作系统，诸如像Google Chrome OS，的工作站计算机。作为替代，或附加地，客户端计算设备1102、1104、1106和1108可以是任何其它电子设备，诸如能够经(一个或多个)网络1110通信的瘦客户端计算机、启用互联网的游戏系统(例如，具有或不具有Kinect®手势输入设备的Microsoft Xbox游戏控制台)，和/或个人消息传送设备。

[0126] 虽然示例性分布式系统1100被示为具有四个客户端计算设备，但任何数量的客户端计算设备可以被支持。其它设备，诸如具有传感器的设备等，可以与服务器1112交互。

[0127] 分布式系统1100中的(一个或多个)网络1110可以是本领域技术人员熟悉的、可以利用各种市售协议当中任意一种支持数据通信的任何类型的网络，其中协议包括但不限于TCP/IP(传输控制协议/网际协议)、SNA(系统网络体系结构)、IPX(互联网数据包交换)、AppleTalk，等等。仅仅作为例子，(一个或多个)网络1110可以是局域网(LAN)，诸如基于以太网、令牌环等的LAN。(一个或多个)网络1110可以是广域网和互联网。它可以包括虚拟网络，包括但不限于虚拟专用网(VPN)、内联网、外联网、公共交换电话网(PSTN)、红外网络、无线网络(例如，依据电气和电子学研究所(IEEE)802.11协议套件、Bluetooth®和/或任何其它无线协议当中任意一种操作的网络)；和/或这些和/或其它网络的任意组合。

[0128] 服务器1112可以由一个或多个通用计算机、专用服务器计算机(作为例子，包括PC(个人计算机)服务器、UNIX®服务器、中档服务器、大型计算机、机架式服务器，等等)、服务器群、服务器集群，或者任何其它适当的布置和/或组合组成。在各种实施例中，服务器1112可以适于运行在前面公开内容中所描述的一个或多个服务或软件应用。例如，服务器1112可以对应于用于执行上面根据本公开内容的实施例所描述的处理的服务器。

[0129] 服务器1112可以运行包括上面讨论的操作系统当中任意一种在内的操作系统，以及任何市售的服务器操作系统。服务器1112还可以运行各种附加的服务器应用和/或中间层应用当中任意一种，包括HTTP(超文本传输协议)服务器、FTP(文档传输协议)服务器、CGI(公共网关接口)服务器、JAVA®服务器、数据库服务器，等等。示例性数据库服务器包括但不限于从Oracle、Microsoft、Sybase、IBM(国际商业机器)等等可商购的那些。

[0130] 在一些实现中，服务器1112可以包括一个或多个应用，以分析和整合从客户端计算设备1102、1104、1106和1108的用户接收的数据馈送和/或事件更新。作为例子，数据馈送和/或事件更新可以包括，但不限于，Twitter®馈送、Facebook®更新或者从一个或多个第三方信息源和连续数据流接收到的实时更新，其可以包括与传感器数据应用、金融报价机、网络性能测量工具(例如，网络监视和流量管理应用)、点击流分析工具、汽车交通监

视等相关的实时事件。服务器1112还可以包括一个或多个应用，以经由客户端计算设备1102、1104、1106和1108的一个或多个显示设备显示数据馈送和/或实时事件。

[0131] 分布式系统1100还可以包括一个或多个数据库1114和1116。数据库1114和1116可以驻留在各个位置。作为例子，数据库1114和1116中的一个或多个可以驻留在服务器1112本地的非暂态存储介质上(和/或驻留在服务器1112中)。作为替代，数据库1114和1116可以远离服务器1112，并且经基于网络的或专用的连接与服务器1112通信。在一组实施例中，数据库1114和1116可以驻留在存储区域网络(SAN)中。类似地，用于执行服务器1112所具有的功能的任何必要的文件都可以适当地本地存储在服务器1112上和/或远程存储。在一组实施例中，数据库1114和1116可以包括适于响应于SQL格式的命令而存储、更新和检索数据的关系数据库，诸如由Oracle提供的数据库。

[0132] 图12示出了本发明的各种实施例可以在其中实现的示例性计算机系统1200。该系统1200可以被用来实现以上描述的任何计算机系统。如图所示，计算机系统1200包括经由总线子系统1202与多个外围子系统通信的处理单元1204。这些外围子系统可以包括处理加速单元1206、I/O子系统1208、存储子系统1218和通信子系统1224。存储子系统1218包括有形的计算机可读存储介质1222和系统存储器1210。

[0133] 总线子系统1202提供用于使计算机系统1200的各种部件和子系统如预期的那样彼此通信的机制。虽然总线子系统1202被示意性地示为单条总线，但是总线子系统的备选实施例可以利用多条总线。总线子系统1202可以是若干种类型的总线结构中的任何一种，包括存储器总线或存储器控制器、外围总线、以及利用任何各种总线体系结构的局部总线。例如，这种体系结构可以包括工业标准体系结构(ISA)总线、微通道体系结构(MCA)总线、增强型ISA(EISA)总线、视频电子标准协会(VESA)局部总线和外围组件互连(PCI)总线，其可以被实现为按IEEE P1386.1标准制造的Mezzanine总线。

[0134] 可以被实现为一个或多个集成电路(例如，常规微处理器或微控制器)的处理单元1204控制计算机系统1200的操作。一个或多个处理器可以被包括在处理单元1204中。这些处理器可以包括单核或多核处理器。在某些实施例中，处理单元1204可以被实现为一个或多个独立的处理单元1232和/或1234，其中在每个处理单元中包括单核或多核处理器。在其它实施例中，处理单元1204也可以被实现为通过将两个双核处理器集成到单个芯片中形成的四核处理单元。

[0135] 在各种实施例，处理单元1204可以响应于程序代码执行各种程序并且可以维护多个并发执行的程序或进程。在任何给定的时间，要被执行的程序代码中的一些或全部代码可以驻留在(一个或多个)处理器1204中和/或存储子系统1218中。通过适当的编程，(一个或多个)处理器1204可以提供上述各种功能。计算机系统1200可以附加地包括处理加速单元1206，其可以包括数字信号处理器(DSP)、专用处理器，等等。

[0136] I/O子系统1208可以包括用户界面输入设备和用户界面输出设备。用户界面输入设备可以包括键盘、诸如鼠标或轨迹球的定点设备、结合到显示器中的触摸板或触摸屏、滚动轮、点击轮、拨盘、按钮、开关、键盘、具有语音命令识别系统的音频输入设备、麦克风以及其它类型的输入设备。用户界面输入设备可以包括，例如，运动感测和/或手势识别设备，诸如Microsoft Kinect®运动传感器，其使得用户能够通过利用手势和语音命令的自然用户

界面控制诸如Microsoft Xbox® 1260游戏控制器的输入设备并与之交互。用户界面输入设备也可以包括眼睛姿势识别设备，诸如从用户检测眼睛活动(例如，当拍拍摄照片和/或做出菜单选择时的“眨眼”)并且将眼睛姿势转换为到输入设备(例如，Google Glass®)中的输入的Google Glass®眨眼检测器。此外，用户界面输入设备可以包括使用户能够通过语音命令与语音识别系统(例如，Siri®导航器)交互的语音识别感测设备。

[0137] 用户界面输入设备也可以包括，但不限于，三维(3D)鼠标、操纵杆或指向棒、游戏面板和绘图板，以及音频/视频设备，诸如扬声器、数码相机、数码摄像机、便携式媒体播放器、网络摄像头、图像扫描仪、指纹扫描仪、条形码阅读器3D扫描仪、3D打印机、激光测距仪和视线跟踪设备。此外，用户界面输入设备可以包括，例如，医学成像输入设备，诸如计算机断层扫描、磁共振成像、正电子发射断层摄影术、医疗超声设备。用户界面输入设备也可以包括，例如，诸如MIDI键盘、数字乐器等的音频输入设备。

[0138] 用户界面输出设备可以包括显示子系统、指示灯，或者诸如音频输出设备的非可视显示器，等等。显示子系统可以是阴极射线管(CRT)、诸如利用液晶显示器(LCD)或等离子显示器的平板设备、投影设备、触摸屏，等等。一般而言，术语“输出设备”的使用意在包括用于从计算机系统1200向用户或其它计算机输出信息的所有可能类型的设备和机制。例如，用户界面输出设备可以包括，但不限于，可视地传达文本、图形和音频/视频信息的各种显示设备，诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备，以及调制解调器。

[0139] 计算机系统1200可以包括包含软件元素、被示为当前位于系统存储器1210中的存储子系统1218。系统存储器1210可以存储可加载并且可在处理单元1204上执行的程序指令，以及在这些程序执行期间所产生的数据。

[0140] 依赖于计算机系统1200的配置和类型，系统存储器1210可以是易失性的(诸如随机存取存储器(RAM))和/或非易失性的(诸如只读存储器(ROM)、闪存存储器，等等)。RAM通常包含可被处理单元1204立即访问和/或目前正被处理单元1204操作和执行的数据和/或程序模块。在一些实现中，系统存储器1210可以包括多种不同类型的存储器，诸如静态随机存取存储器(SRAM)或动态随机存取存储器(DRAM)。在一些实现中，诸如在启动期间，包含有助于在计算机系统1200的元件之间传送信息的基本例程的基本输入/输出系统(BIOS)，通常可以被存储在ROM中。作为例子，但不是限制，系统存储器1210也示出了可以包括客户端应用、web浏览器、中间层应用、关系数据库管理系统(RDBMS)等的应用程序1212，程序数据1214，以及操作系统1216。作为例子，操作系统1216可以包括各种版本的Microsoft Windows®，Apple Macintosh® 和/或Linux操作系统、各种商用UNIX®或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统、Google Chrome®操作系统，等等)和/或诸如iOS、Windows® Phone、Android® OS、BlackBerry® 10OS和Palm® OS操作系统的移动操作系统。

[0141] 存储子系统1218也可以提供用于存储提供一些实施例的功能的基本编程和数据结构的有形计算机可读存储介质。当被处理器执行时提供上述功能的软件(程序、代码模块、指令)可以被存储在存储子系统1218中。这些软件模块或指令可以被处理单元1204执

行。存储子系统1218也可以提供用于存储根据本公开内容被使用的数据的储存库。

[0142] 存储子系统1218也可以包括可被进一步连接到计算机可读存储介质1222的计算机可读存储介质读取器1220。与系统存储器1210一起并且，可选地，与其相结合，计算机可读存储介质1222可以全面地表示用于临时和/或更持久地包含、存储、发送和检索计算机可读信息的远程、本地、固定和/或可移除存储设备加存储介质。

[0143] 包含代码或代码的部分的计算机可读存储介质1222也可以包括本领域已知或使用的任何适当的介质，包括存储介质和通信介质，诸如但不限于，以用于信息的存储和/或传输的任何方法或技术实现的易失性和非易失性、可移除和不可移除介质。这可以包括有形的计算机可读存储介质，诸如RAM、ROM、电可擦除可编程ROM(EEPROM)、闪存存储器或其它存储器技术、CD-ROM、数字多功能盘(DVD)或其它光学储存器、磁带盒、磁带、磁盘储存器或其它磁存储设备，或者其它有形的计算机可读介质。这也可以包括非有形的计算机可读介质，诸如数据信号、数据传输，或者可以被用来发送期望信息并且可以被计算系统1200访问的任何其它介质。

[0144] 作为例子，计算机可读存储介质1222可以包括从不可移除的非易失性磁介质读取或写到其的硬盘驱动器、从可移除的非易失性磁盘读取或写到其的磁盘驱动器、以及从可移除的非易失性光盘，诸如CD ROM、DVD和Blu-Ray®盘或其它光学介质，读取或写到其的光盘驱动器。计算机可读存储介质1222可以包括，但不限于，Zip®驱动器、闪存卡、通用串行总线(USB)闪存驱动器、安全数字(SD)卡、DVD盘、数字音频带，等等。计算机可读存储介质922也可以包括基于非易失性存储器的固态驱动器(SSD)，诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等，基于易失性存储器的SSD，诸如固态RAM、动态RAM、静态RAM，基于DRAM的SSD，磁阻RAM(MRAM)SSD，以及使用基于DRAM和闪存存储器的SSD的组合的混合SSD。盘驱动器及其相关联的计算机可读介质可以为计算机系统1200提供计算机可读指令、数据结构、程序模块及其它数据的非易失性存储。

[0145] 通信子系统1224提供到其它计算机系统和网络的接口。通信子系统1224用作用于从其它系统接收数据和从计算机系统1200向其它系统发送数据的接口。例如，通信子系统1224可以使计算机系统1200能够经由互联网连接到一个或多个设备。在一些实施例中，通信子系统1224可以包括用于访问无线语音和/或数据网络的射频(RF)收发器组件(例如，利用蜂窝电话技术，诸如3G、4G或EDGE(用于全球演进的增强型数据速率)的先进数据网络技术，Wi-Fi(IEEE 802.11系列标准)，或其它移动通信技术，或其任意组合)、全球定位系统(GPS)接收器组件和/或其它组件。在一些实施例中，作为无线接口的附加或者替代，通信子系统1224可以提供有线网络连接(例如，以太网)。

[0146] 在一些实施例中，通信子系统1224也可以代表可以使用计算机系统1200的一个或多个用户接收结构化和/或非结构化数据馈送1226、事件流1228、事件更新1230等形式的输入通信。

[0147] 作为例子，通信子系统1224可以被配置为实时地从社交网络和/或其它通信服务的用户接收数据馈送1226，诸如Twitter®馈送、Facebook®更新、诸如丰富站点摘要(RSS)馈送的web馈送和/或来自一个或多个第三方信息源的实时更新。

[0148] 此外，通信子系统1224也可以被配置为接收连续数据流形式的数据，这可以包括

本质上可以是连续的或无界的没有明确终止的实时事件的事件流1228和/或事件更新1230。产生连续数据的应用的例子可以包括，例如，传感器数据应用、金融报价机、网络性能测量工具(例如，网络监视和流量管理应用)、点击流分析工具、汽车流量监视，等等。

[0149] 通信子系统1224也可以被配置为向一个或多个数据库输出结构化和/或非结构化数据馈送1226、事件流1228、事件更新1230，等等，这一个或多个数据库可以与耦合到计算机系统1200的一个或多个流式数据源计算机通信。

[0150] 计算机系统1200可以是各种类型之一，包括手持式便携式设备(例如，iPhone®蜂窝电话、iPad®计算平板、PDA)、可穿戴设备(例如，Google Glass®头戴式显示器)、PC、工作站、大型机、信息站，服务器机架，或任何其它数据处理系统。

[0151] 由于计算机和网络的不断变化的本质，在该图中绘出的计算机系统1200的描述仅仅要作为具体的例子。具有比该图中绘出的系统更多或更少组件的许多其它配置是可能的。例如，定制的硬件也可以被使用和/或特定的元素可以用硬件、固件、软件(包括applets)或它们的组合来实现。另外，也可以采用到诸如网络输入/输出设备之类的其它计算设备的连接。基于本文提供的公开内容和示教，本领域普通技术人员将认识到实现各种实施例的其它方式和/或方法。

[0152] 在前面的说明书中，本发明的各方面参考其具体实施例进行了描述，但本领域技术人员将认识到，本发明不限于此。上述发明的各个特征和各方面可以被单独或联合使用。另外，在不背离本说明书的更广泛精神和范围的情况下，实施例可以在除本文所述的那些之外的任何数目的环境和应用中被使用。因此，本说明书和附图应当被认为是说明性而不是限制性的。

[0153] 其示教可以在本公开内容中给出的一个或多个发明当中任意一个的各种实施例可以以软件、固件、硬件或其组合中的逻辑的形式来实现。逻辑可以存储在机器可访问的存储器、机器可读的制品、有形的计算机可读介质、计算机可读存储介质或其它计算机/机器可读介质当中或其上，作为适于指示逻辑机器的中央处理单元(CPU或处理器)执行可以在本公开内容中给出的发明的各种实施例中公开的步骤集的指令集。逻辑可以构成软件程序或计算机程序产品的一部分，因为，当其被执行时，代码模块变得利用计算机系统或信息处理设备的处理器可操作，以执行在本公开内容中给出的发明的各种实施例中的方法或过程。基于本公开内容和本文提供的示教，本领域普通技术人员将认识到用于在软件、固件、硬件或其组合中实现所给出的一个或多个发明的各种实施例的任何所公开的操作或功能的其它途径、变体、修改、备选方案和/或方法。

[0154] 其示教可以在本公开内容中给出的一个或多个发明当中任意一个的所公开的例子、实现和各种实施例仅仅是说明性的，以便以合理的清晰度向本领域技术人员传达本公开内容的示教。由于这些实现和实施例可以参考示例性说明或具体的图进行描述，因此所描述的方法和/或具体结构的各种修改或适应会对本领域技术人员变得显然。依赖于本公开内容和其中发现的这些示教，或者所述示教通过其推动本领域的，所有此类修改、适应或变体，都应当被认为在其示教在本公开内容中给出的一个或多个发明的范围内。因此，本描述和附图不应当在限制的意义上考虑，因为应当理解，在本公开内容中给出的发明决不是要限于具体说明的那些实施例。

[0155] 因此，以上描述和任何附图、说明和图示都是说明性而不是限制性的。因此，本公开内容中给出的任何发明的范围不应当简单地参考以上描述和图示中示出的那些实施例来确定，而是应当参考未决的权利要求连同其完整范围或等同物来确定。

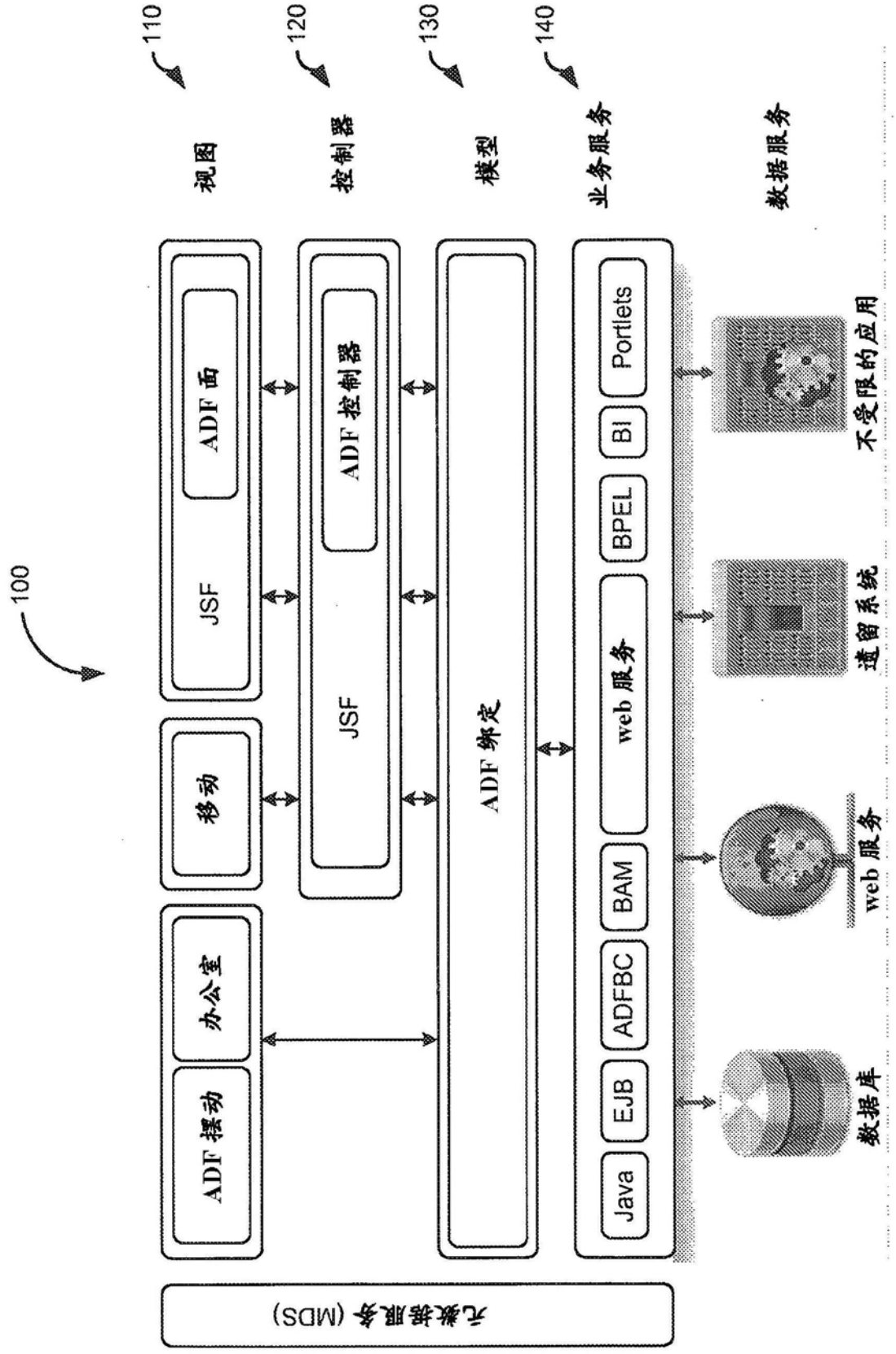

100
110
120
130
140
视图
控制器
模型
业务服务
数据服务
ADF 面
JSF
ADF 控制器
JSF
ADF 绑定
Portlets
BI
BPEL
web 服务
BAM
ADFBC
EJB
Java
移动
办公室
ADF 摆动
元数据服务 (MDS)
不受限的应用
遗留系统
web 服务
数据库

图1

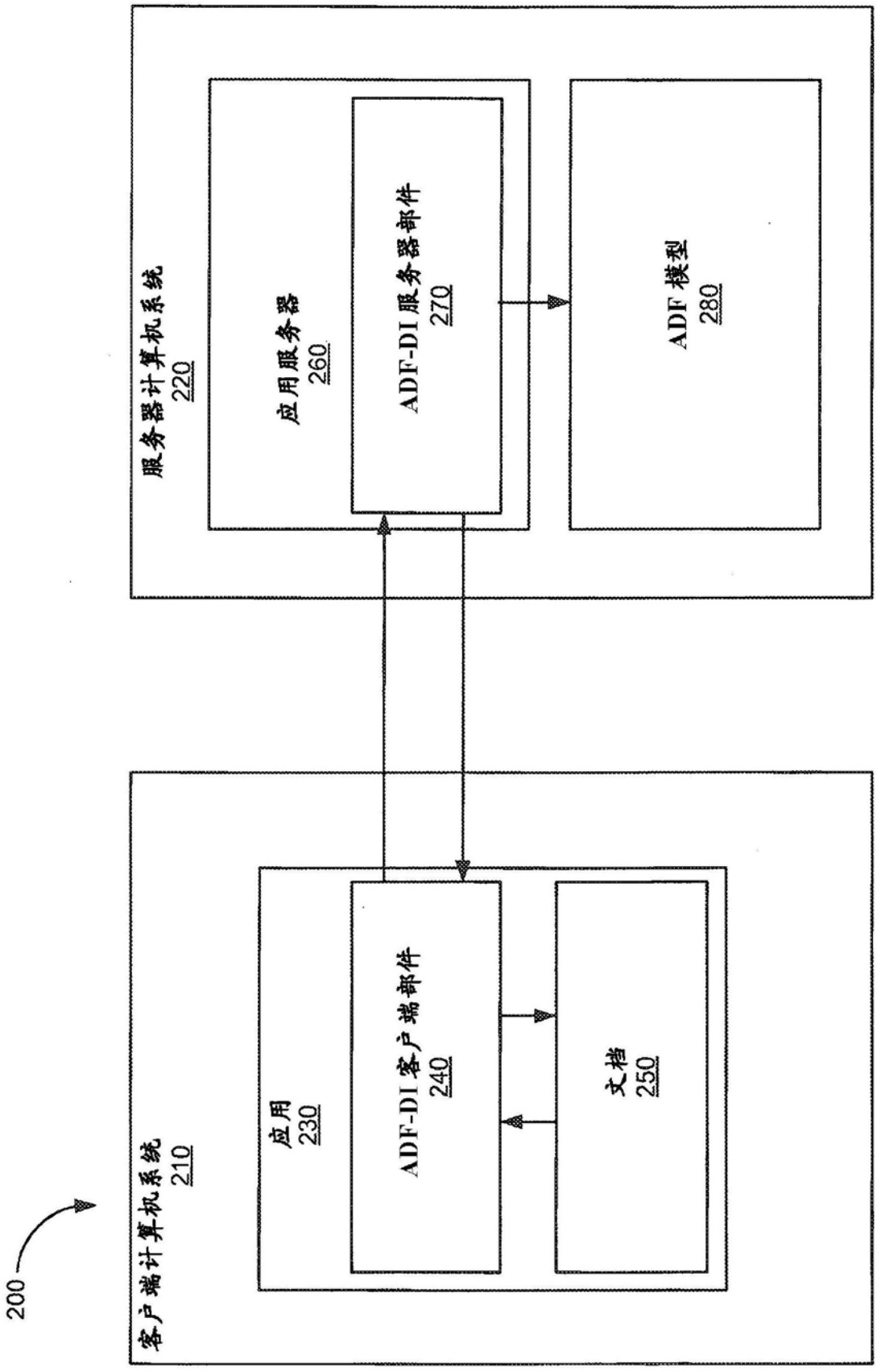

200
客户端计算机系统
210
应用
230
ADF-DI 客户端部件
240
文档
250
服务器计算机系统
220
应用服务器
260
ADF-DI 服务器部件
270
ADF 模型
280

图2

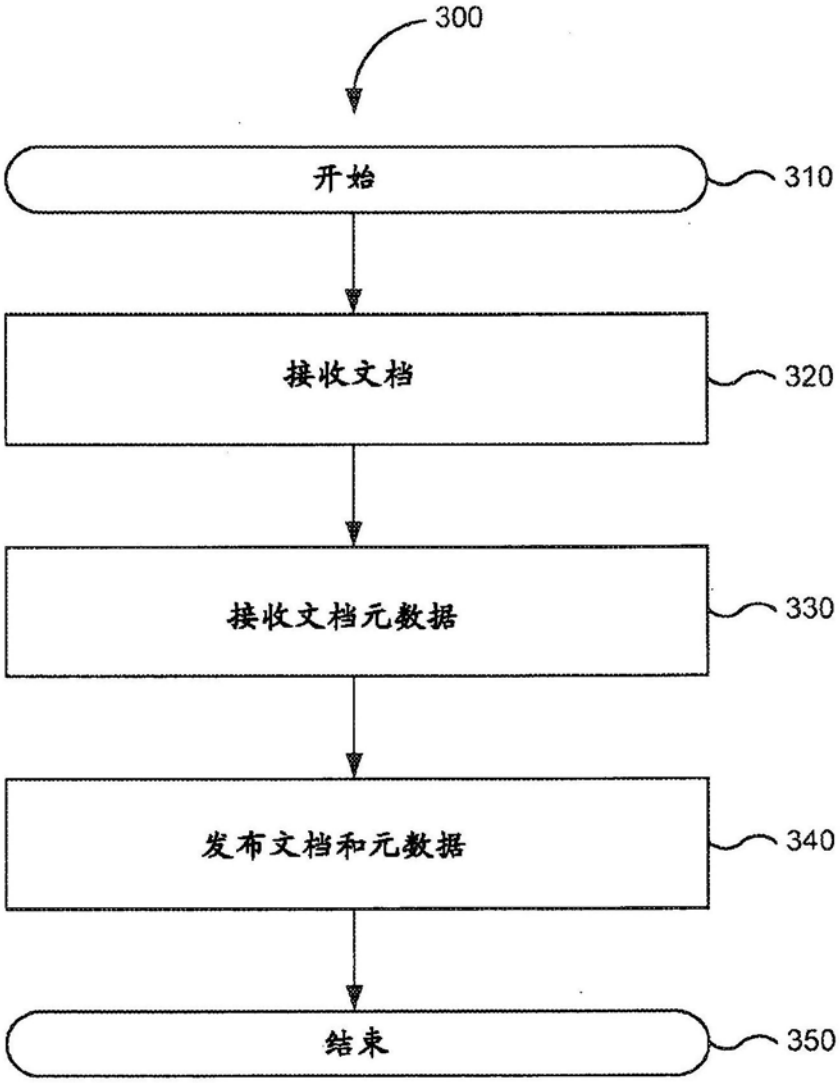


图3

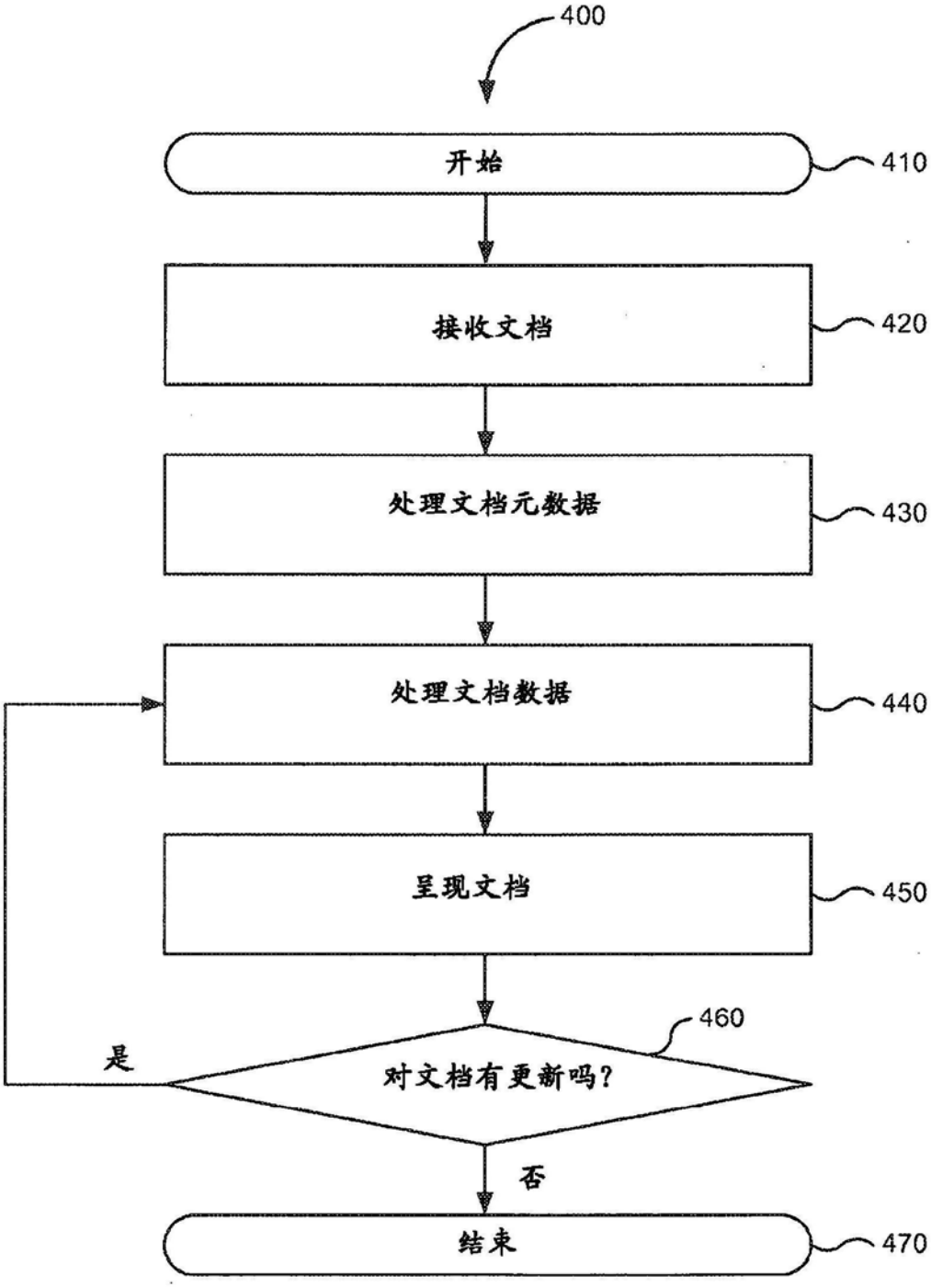

400
开始
410
接收文档
420
处理文档元数据
430
处理文档数据
440
呈现文档
450
460
对文档有更新吗?
是
否
结束
470

图4

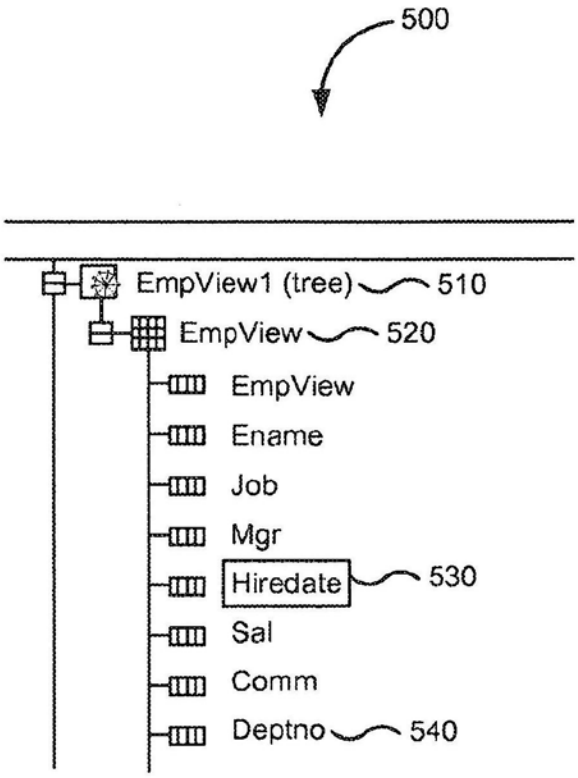

500
EmpView1 (tree)
510
EmpView
520
EmpView
Ename
Job
Mgr
Hiredate
530
Sal
Comm
Deptno
540

图5

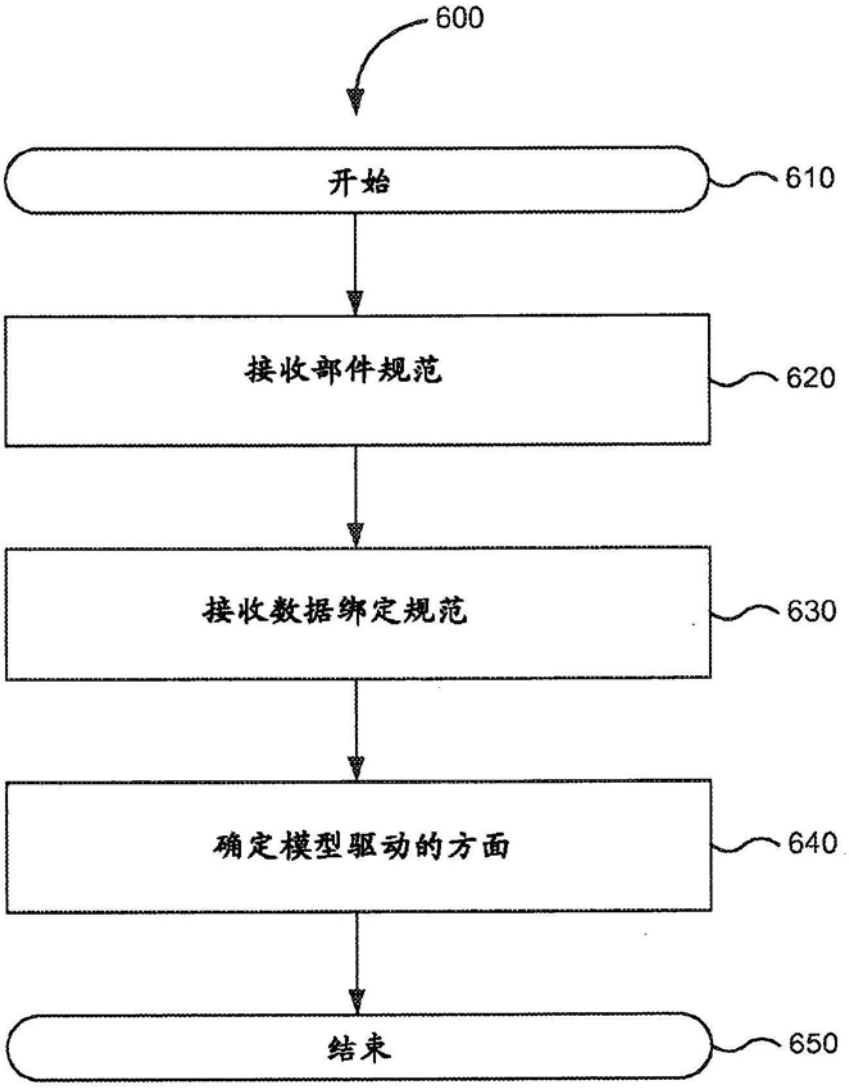


图6

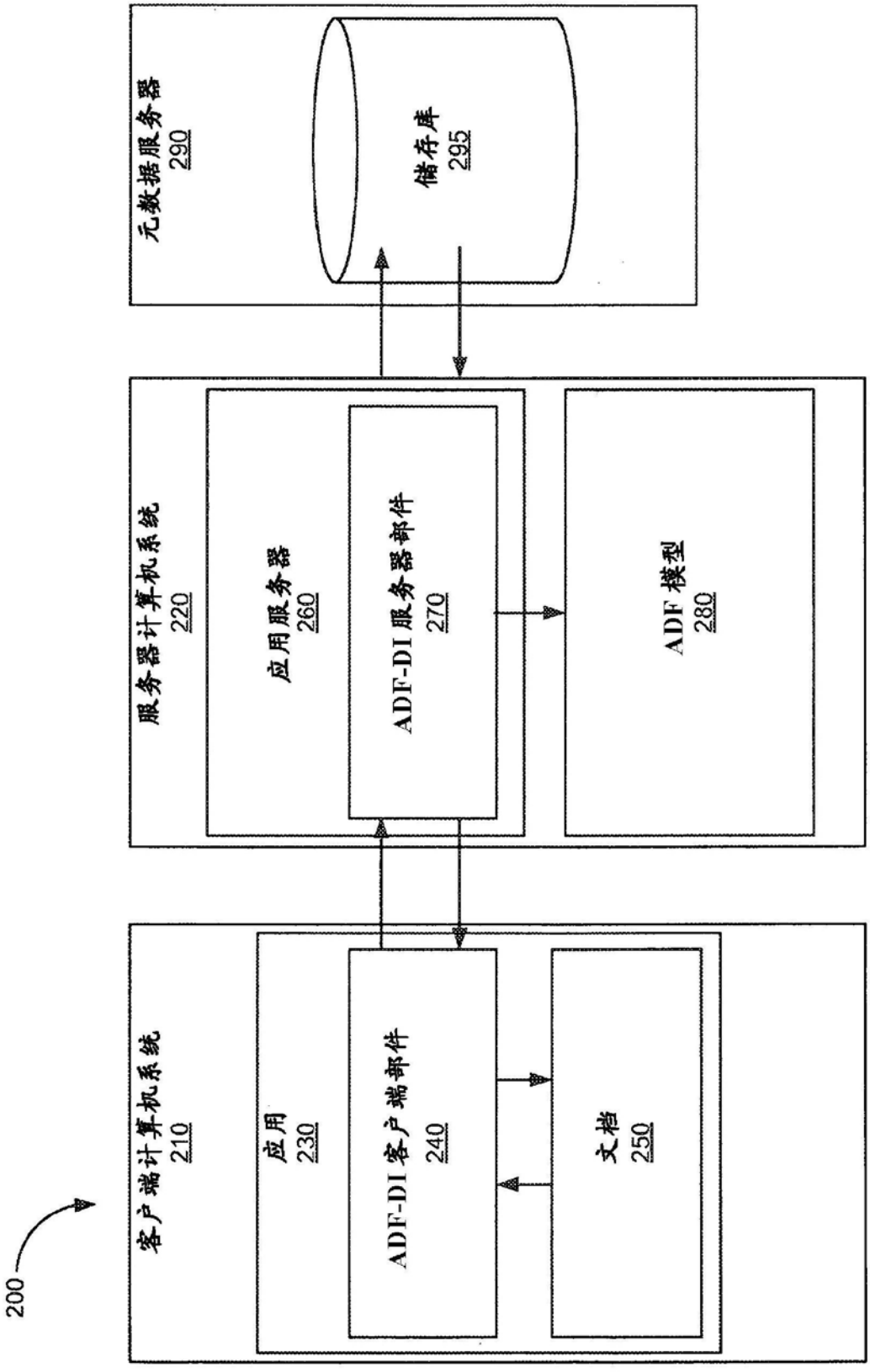

元数据服务器
290
储存库
295
服务器计算机系统
220
应用服务器
260
ADF-DI 服务器部件
270
ADF 模型
280
客户端计算机系统
210
应用
230
ADF-DI 客户端部件
240
文档
250
200

图7

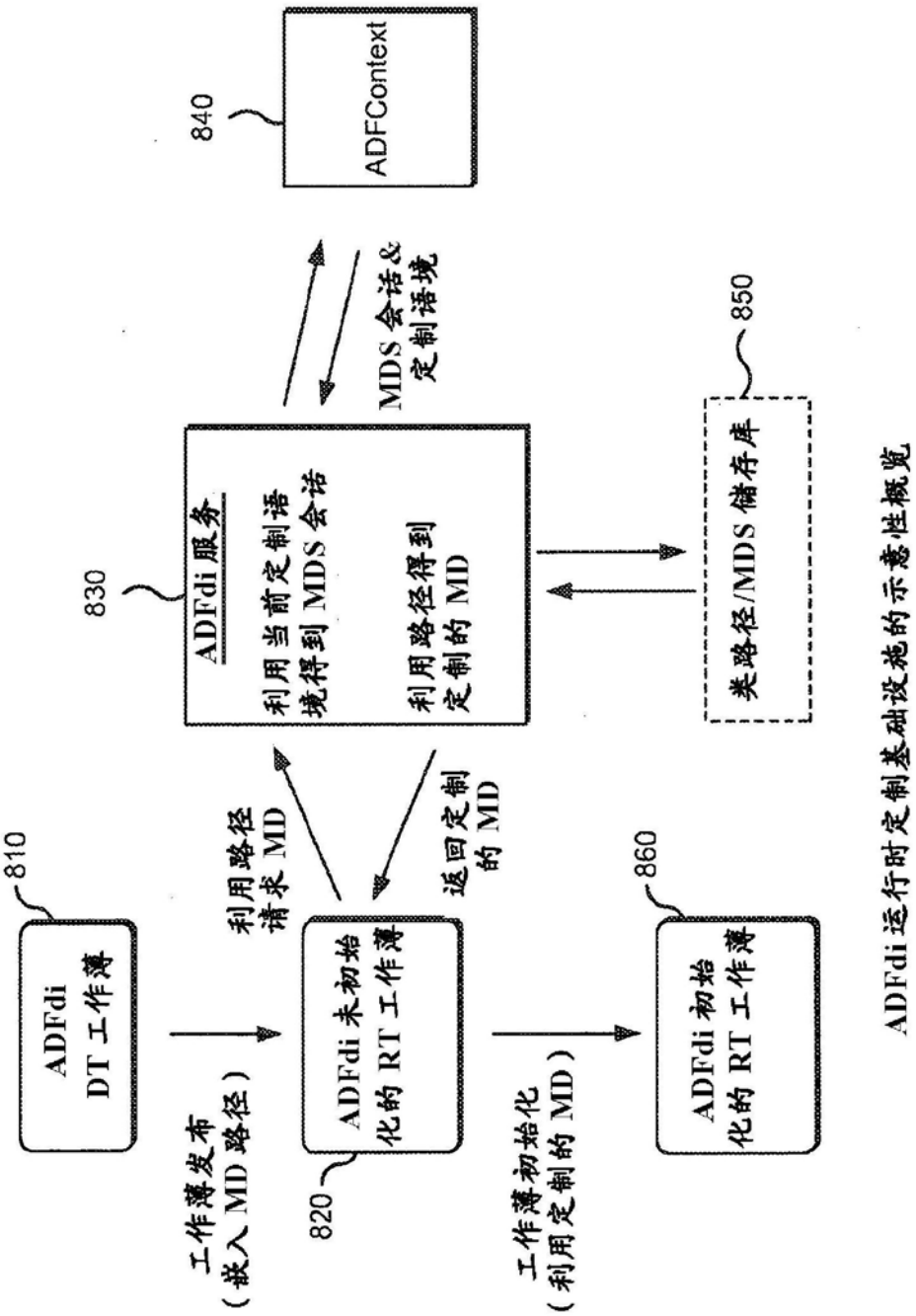

840
ADFContext
MDS 会话&
定制语境
850
830
ADFdi 服务
利用当前定制语境得到 MDS 会话
利用路径得到定制的 MID
类路径/MDS 储存库
利用路径请求 MID
返回定制的 MID
810
860
ADFdi DT 工作薄
ADFdi 未初始化的 RT 工作薄
ADFdi 初始化的 RT 工作薄
工作薄发布（嵌入 MID 路径）
820
工作薄初始化（利用定制的 MID）
ADFdi 运行时定制基础设施的示意性概览

图8

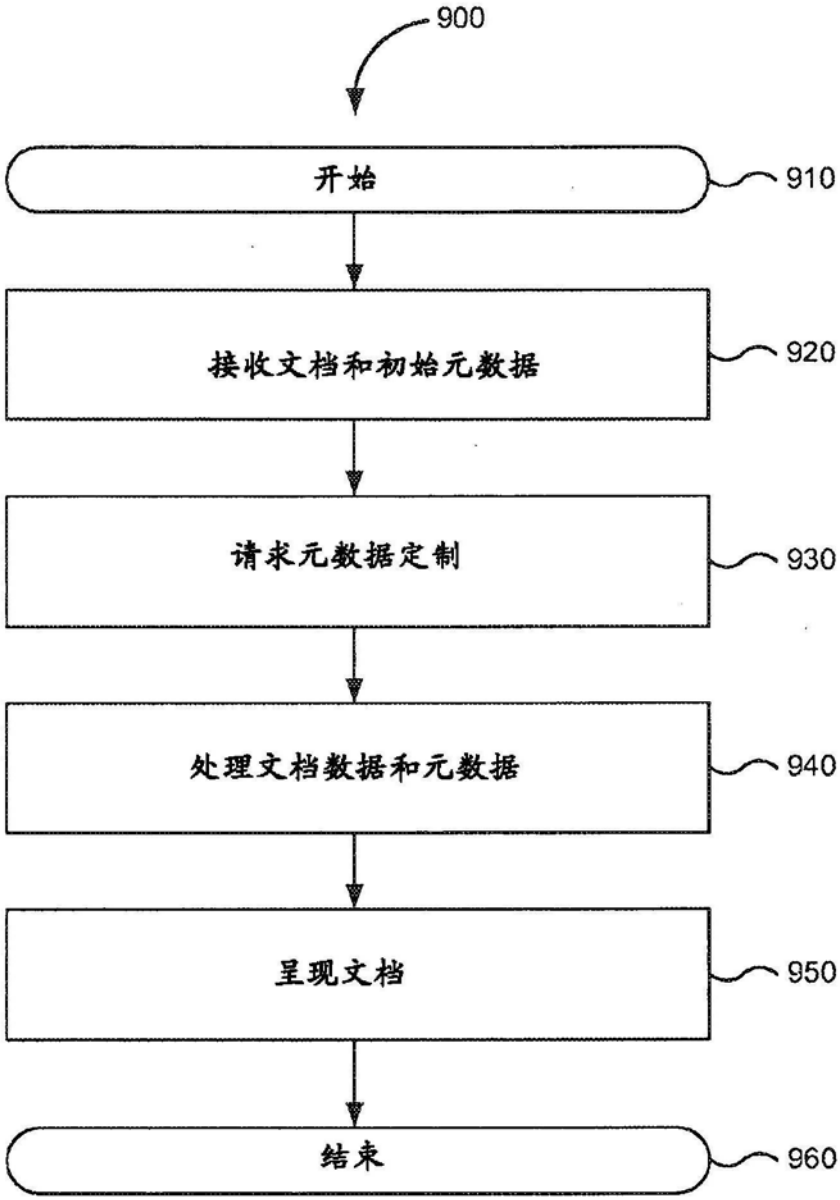

900
开始
910
接收文档和初始元数据
920
请求元数据定制
930
处理文档数据和元数据
940
呈现文档
950
结束
960

图9

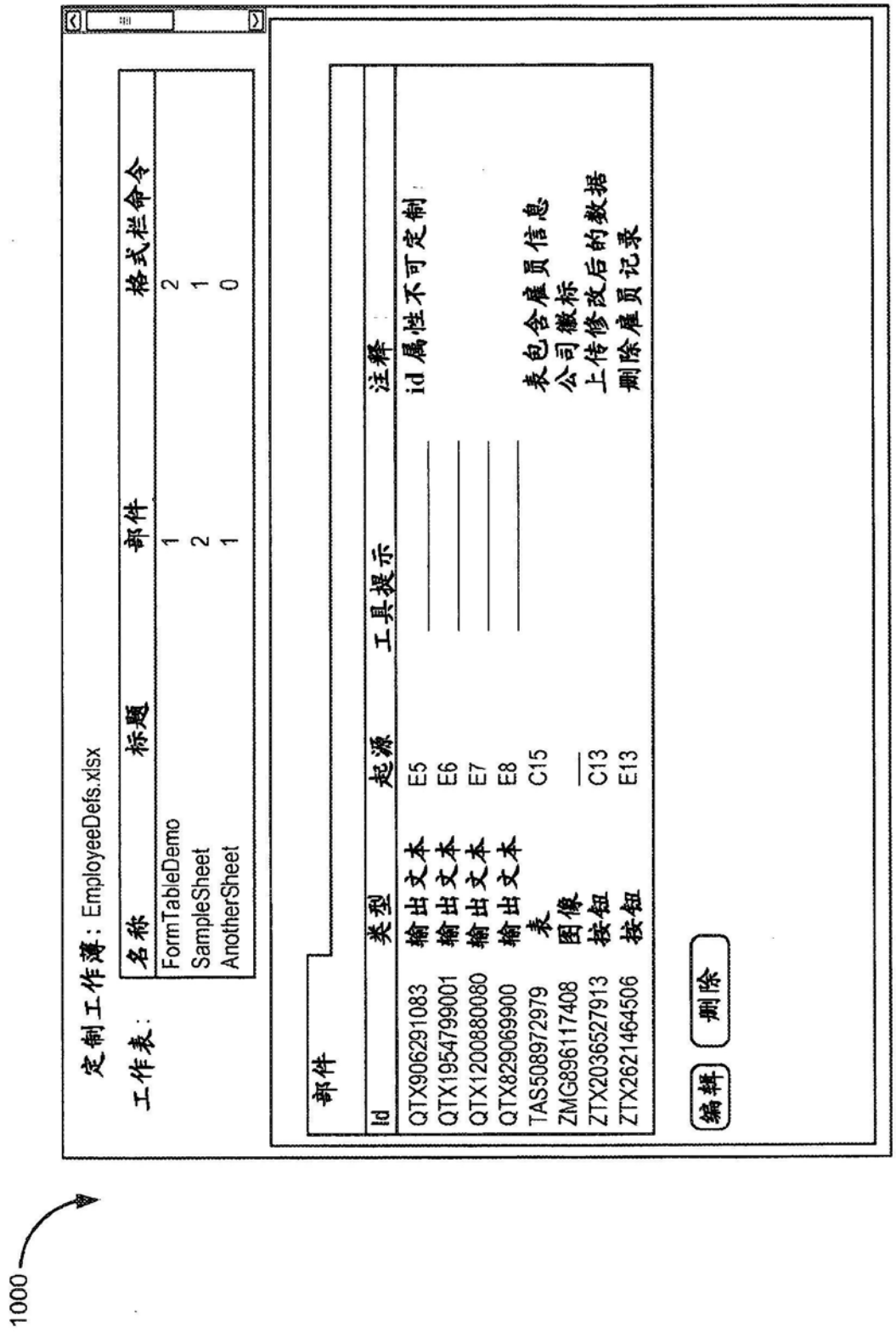

1000
定制工作簿：EmployeeDefs.xlsx
工作表：
名称 标题 部件 格式栏命令
FormTableDemo 1 2
SampleSheet 2 1
AnotherSheet 1 0
部件
Id 类型 起源 工具提示 注释
QTX906291083 输出文本 E5 id 属性不可定制
QTX1954799001 输出文本 E6
QTX1200880080 输出文本 E7
QTX829069900 输出文本 E8
TAS508972979 表 C15 表包含雇员信息
ZMG896117408 图像 公司徽标
ZTX2036527913 按钮 C13 上传修改后的数据
ZTX2621464506 按钮 E13 删除雇员记录
编辑
删除

图10

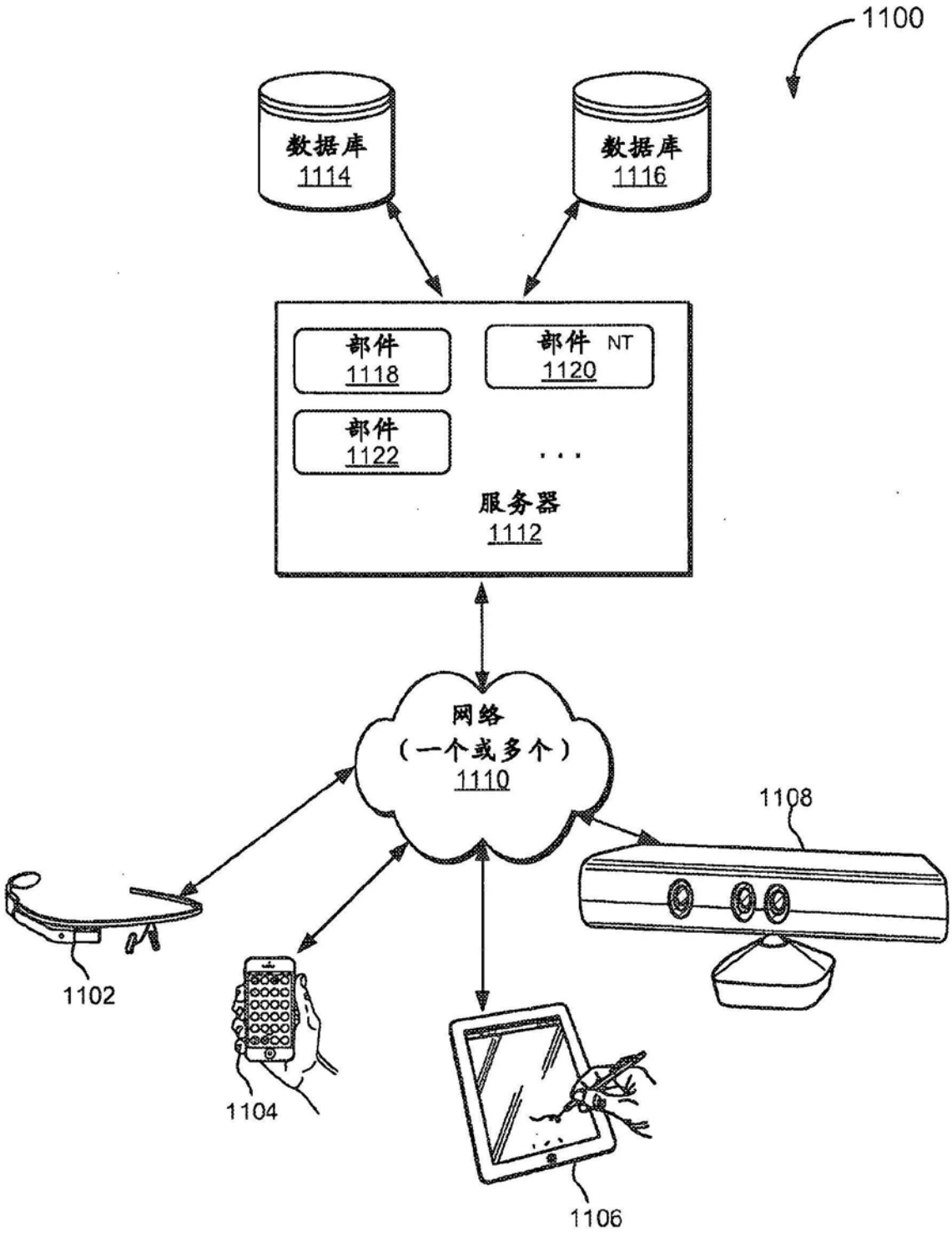

1100
数据库
1114
数据库
1116
部件
1118
部件 NT
1120
部件
1122
. . .
服务器
1112
网络
（一个或多个）
1110
1108
1102
1104
1106

图11

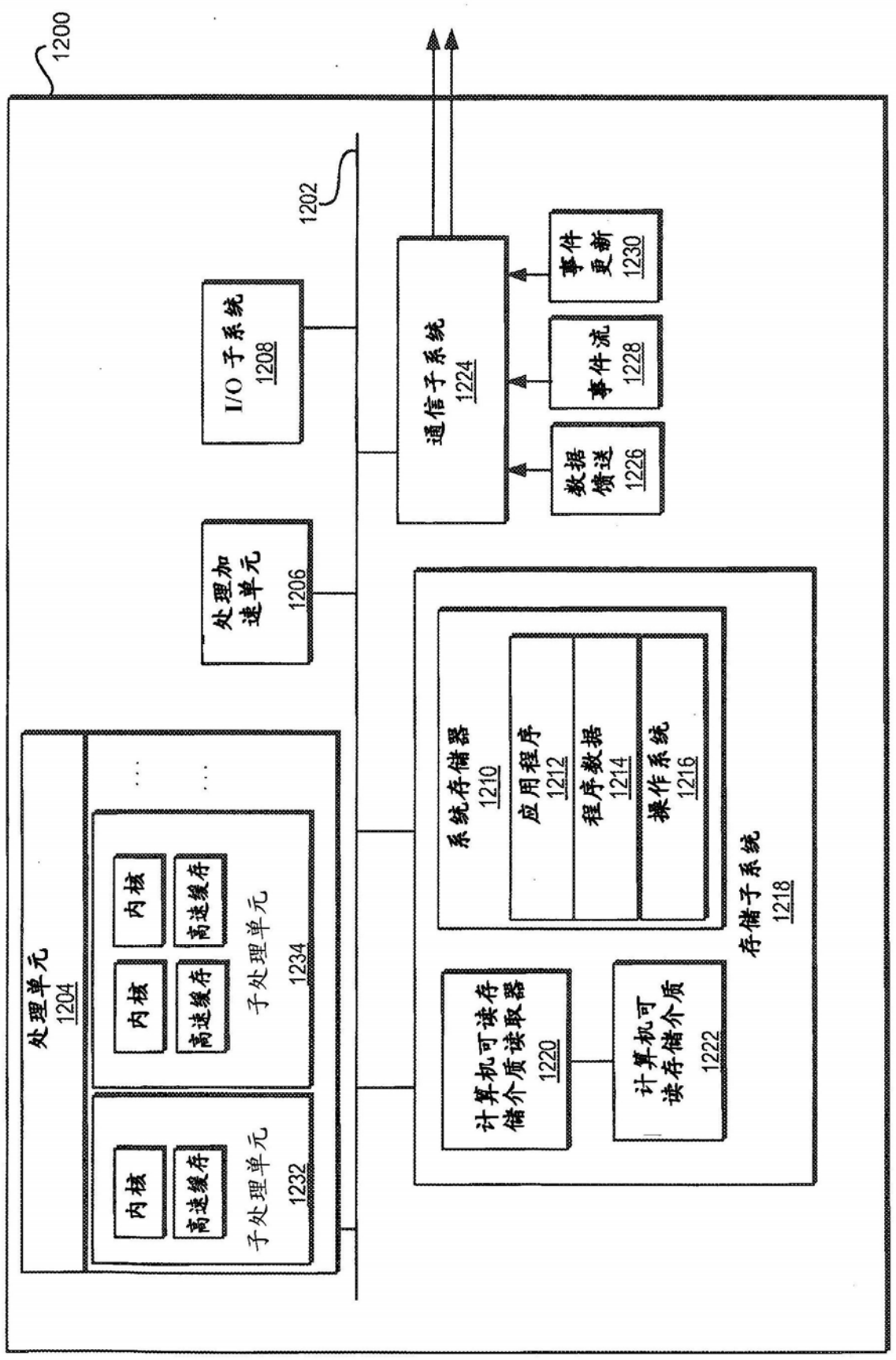

1200
1202
处理单元
1204
内核
高速缓存
子处理单元
1232
内核
高速缓存
子处理单元
1234
内核
高速缓存
I/O 子系统
1208
处理加速单元
1206
通信子系统
1224
事件更新
1230
事件流
1228
数据馈送
1226
存储子系统
1218
系统存储器
1210
应用程序
1212
程序数据
1214
操作系统
1216
计算机可读存储介质读取器
1220
计算机可读存储介质
1222

图12