(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 105339941 B
(45)授权公告日 2019.03.05

(21)申请号 201480036876.X

(22)申请日 2014.03.26

(65)同一申请的已公布的文献号
申请公布号 CN 105339941 A

(43)申请公布日 2016.02.17

(30)优先权数据
61/824,544 2013.05.17 US
14/044,417 2013.10.02 US

(85)PCT国际申请进入国家阶段日
2015.12.28

(86)PCT国际申请的申请数据
PCT/US2014/031921 2014.03.26

(87)PCT国际申请的公布数据
WO2014/186058 EN 2014.11.20

(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 J·莱格 D·阿兰
刘国洪(托马斯)

(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 边海梅

(51)Int.Cl.
G06F 16/25(2019.01)

(56)对比文件
US 2010/0005366 A1,2010.01.07,
CN 1763696 A,2006.04.26,
CN 100373297 C,2008.03.05,
US 2013103705 A1,2013.04.25,
US 2008281849 A1,2008.11.13,

审查员 刘津

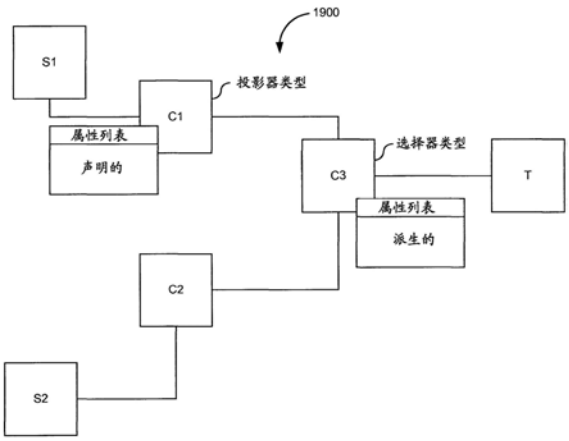
权利要求书4页 说明书21页 附图23页

(54)发明名称

针对ETL映射设计使用投影器和选择器组件类型

(57)摘要

公开了结合了用于简化映射的设计和维护的一种或多种技术的数据集成系统。当组件被添加到现有设计或者从现有设计被除去时,数据集成系统消除了指定所有输入和输出属性的需求。在一个方面中,实现了允许赋值表达式引用上游组件的全部或一部分的组件类型。因此,某些类型的组件的属性可以被传播到下游组件或者以别的方式从上游组件继承,其结果是对映射设计人员的部分需要最小的努力。在代码生成期间,被任何组件投影所需的属性可以基于下游组件的需求而派生。



CN 105339941 B

1.一种用于促进数据映射的生成的方法,所述方法包括:

在一个或多个计算机系统处,接收指定逻辑设计的一个或多个组件的信息,其中所述一个或多个组件当中至少一个组件具有第一类型,并且该信息指定所述逻辑设计的所述一个或多个组件指示改变流经所述逻辑设计的信息的形状的操作,或者指示控制流经所述逻辑设计的信息流但是不改变流经所述逻辑设计的信息的形状的操作;

利用与所述一个或多个计算机系统关联的一个或多个处理器,基于所述逻辑设计中的上游组件,确定所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的对象的数据属性集合;

将具有第一类型的所述对象的所述数据属性集合分组;及

利用与所述一个或多个计算机系统关联的一个或多个处理器,生成指示包括所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的所述数据属性集合的分组的信息,以创建数据集成计划。

2.如权利要求1所述的方法,其中利用与所述一个或多个计算机系统关联的一个或多个处理器,确定所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的数据属性集合包括派生从上游组件可见的一个或多个属性并且将所述一个或多个属性暴露给下游组件。

3.如权利要求1所述的方法,其中接收指定所述逻辑设计的一个或多个组件的信息包括接收指示具有存储在源数据存储仓中的数据的一个或多个属性的源组件的信息。

4.如权利要求1所述的方法,其中接收指定所述逻辑设计的一个或多个组件的信息包括接收指示具有要存储在目标数据存储仓中的数据的一个或多个属性的目标组件的信息。

5.如权利要求1所述的方法,其中生成指示所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的所述数据属性集合的信息包括将属性列表导出到下游组件。

6.如权利要求1所述的方法,还包括:

在所述一个或多个计算机系统处,通过将组件或属性引入到所述逻辑设计中或从所述逻辑设计中去除组件或属性来接收所述逻辑设计中的变化;

利用与所述一个或多个计算机系统关联的一个或多个处理器,确定所述逻辑设计中的变化是否影响所述一个或多个组件当中具有第一类型的至少一个组件;及

基于确定所述逻辑设计中的变化影响所述一个或多个组件当中具有第一类型的至少一个组件,利用与所述一个或多个计算机系统关联的一个或多个处理器,确定对下游组件可见的已更新的数据属性集合。

7.如权利要求1所述的方法,还包括:

在所述一个或多个计算机系统处,通过将组件或属性引入到所述逻辑设计中来接收所述逻辑设计中的变化;

利用与所述一个或多个计算机系统关联的一个或多个处理器,确定所述逻辑设计中的变化是否影响所述一个或多个组件当中具有第一类型的至少一个组件;及

基于确定所述逻辑设计中的变化影响所述一个或多个组件当中具有第一类型的至少一个组件,保留对下游组件可见的数据属性集合。

8.如权利要求1所述的方法,还包括:

在所述一个或多个计算机系统处，接收所述逻辑设计中重命名组件或属性的变化；

利用与所述一个或多个计算机系统关联的一个或多个处理器，确定所述逻辑设计中的变化是否影响所述一个或多个组件当中具有第一类型的至少一个组件；及

基于确定所述逻辑设计中的变化影响所述一个或多个组件当中具有第一类型的至少一个组件，重命名对下游组件可见的数据属性集合。

9.根据权利要求1所述的方法，其中所述数据集成计划被划分成多个执行单元，所述多个执行单元被配置成根据用户的基础架构创建物理计划。

10.根据权利要求1所述的方法，其中所述一个或多个组件包括以下之一：源组件、目标组件、集合组件、表组件、接合组件以及过滤组件。

11.根据权利要求1所述的方法，其中所述逻辑设计定义源组件和目标组件之间的数据流。

12.根据权利要求1所述的方法，还包括利用指示所述数据属性集合的信息来创建所述数据集成计划。

13.一种非临时性计算机可读介质，存储用于促进数据映射生成的计算机可执行代码，所述非临时性计算机可读介质包括：

用于接收指定逻辑设计的一个或多个组件的信息的代码，其中所述一个或多个组件当中至少一个组件具有第一类型，并且该信息指定所述逻辑设计的所述一个或多个组件指示改变流经所述逻辑设计的信息的形状的操作，或者指示控制流经所述逻辑设计的信息流但是不改变流经所述逻辑设计的信息的形状的操作；

用于基于所述逻辑设计中的上游组件，确定所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的对象的数据属性集合的代码；

用于将具有第一类型的所述对象的所述数据属性集合分组的代码；及

用于生成指示包括所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的所述数据属性集合的分组的信息以创建数据集成计划的代码。

14.如权利要求13所述的非临时性计算机可读介质，其中用于确定所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的数据属性集合的代码包括用于派生从上游组件可见的一个或多个属性并且将所述一个或多个属性暴露给下游组件的代码。

15.如权利要求13所述的非临时性计算机可读介质，其中用于接收指定逻辑设计的一个或多个组件的信息的代码包括用于接收指示具有存储在源数据存储仓中的数据的一个或多个属性的源组件的信息的代码。

16.如权利要求13所述的非临时性计算机可读介质，其中用于接收指定逻辑设计的一个或多个组件的信息的代码包括用于接收指示具有要存储在目标数据存储仓中的数据的一个或多个属性的目标组件的信息的代码。

17.如权利要求13所述的非临时性计算机可读介质，其中用于生成指示所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的所述数据属性集合的信息的代码包括用于将属性列表导出到下游组件的代码。

18.如权利要求13所述的非临时性计算机可读介质，还包括：

用于通过将组件或属性引入到逻辑设计中或从逻辑设计中去除组件或属性来接收所述逻辑设计中的变化的代码；

用于确定所述逻辑设计中的变化是否影响所述一个或多个组件当中具有第一类型的至少一个组件的代码；及

用于基于确定所述逻辑设计中的变化影响所述一个或多个组件当中具有第一类型的至少一个组件，确定下游组件可见的已更新的数据属性集合的代码。

19.如权利要求13所述的非临时性计算机可读介质，还包括：

用于通过将组件或属性引入到逻辑设计中或从逻辑设计中去除组件或属性来接收所述逻辑设计中的变化的代码；

用于确定所述逻辑设计中的变化是否影响所述一个或多个组件当中具有第一类型的至少一个组件的代码；及

用于基于确定所述逻辑设计中的变化影响所述一个或多个组件当中具有第一类型的至少一个组件，保留下游组件可见的数据属性集合的代码。

20.如权利要求13所述的非临时性计算机可读介质，还包括：

用于接收逻辑设计中重命名组件或属性的变化的代码；

用于确定所述逻辑设计中的变化是否影响所述一个或多个组件当中具有第一类型的至少一个组件的代码；及

用于基于确定所述逻辑设计中的变化影响所述一个或多个组件当中具有第一类型的至少一个组件，重命名下游组件可见的数据属性集合的代码。

21.一种用于促进数据映射的生成的系统，所述系统包括：

处理器；及

存储器，与所述处理器通信并且被配置为存储指令集合，所述指令集合在被所述处理器执行时，将所述处理器配置为：

接收指定逻辑设计的一个或多个组件的信息，其中所述一个或多个组件当中至少一个组件具有第一类型，并且该信息指定所述逻辑设计的所述一个或多个组件指示改变流经所述逻辑设计的信息的形状的操作，或者指示控制流经所述逻辑设计的信息流但是不改变流经所述逻辑设计的信息的形状的操作；

基于所述逻辑设计中的上游组件，确定所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的对象的数据属性集合；

将具有第一类型的所述对象的所述数据属性集合分组；及

生成指示包括所述一个或多个组件当中具有第一类型的至少一个组件的、对所述逻辑设计中的下游组件可见的所述数据属性集合的分组的信息，以创建数据集成计划。

22.一种数据集成系统，包括：

接收单元，被配置为接收指定逻辑设计的一个或多个组件的信息，其中所述一个或多个组件当中至少一个组件具有第一类型，并且该信息指定所述逻辑设计的所述一个或多个组件指示改变流经所述逻辑设计的信息的形状的操作，或者指示控制流经所述逻辑设计的信息流但是不改变流经所述逻辑设计的信息的形状的操作；

确定单元，被配置为基于逻辑设计中的上游组件，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的对象的数据属性集合；

分组单元，被配置为将具有第一类型的所述对象的所述数据属性集合分组；以及

生成单元，被配置为生成指示包括所述一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的分组的信息，以创建数据集成计划。

23.如权利要求22所述的系统，其中确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合包括派生从上游组件可见的一个或多个属性并且将所述一个或多个属性暴露给下游组件。

24.如权利要求22所述的系统，其中指定逻辑设计的一个或多个组件的信息包括指示具有存储在源数据存储仓中的数据的一个或多个属性的源组件的信息。

25.如权利要求22所述的系统，其中指定逻辑设计的一个或多个组件的信息包括指示具有要存储在目标数据存储仓中的数据的一个或多个属性的目标组件的信息。

26.如权利要求22所述的系统，其中生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的信息包括将属性列表导出到下游组件。

27.如权利要求22所述的系统，其中：

所述接收单元还被配置为通过将组件或属性引入到逻辑设计中或从逻辑设计中去除组件或属性来接收逻辑设计中的变化，

所述确定单元还被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件，以及

基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，所述确定单元还被配置为确定下游组件可见的已更新的数据属性集合。

28.如权利要求22所述的系统，还包括保留单元，其中：

所述接收单元还被配置为通过将组件或属性引入到逻辑设计中来接收逻辑设计中的变化，

所述确定单元还被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件，以及

基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，所述保留单被配置为保留对下游组件可见的数据属性集合。

29.如权利要求22所述的系统，还包括重命名单元，其中：

所述接收单元还被配置为接收逻辑设计中重命名组件或属性的变化，

所述确定单元还被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件，以及

基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，所述重命名单元被配置为重命名对下游组件可见的数据属性集合。

针对ETL映射设计使用投影器和选择器组件类型

背景技术

[0001] 在如今日益快节奏的商业环境中，机构需要使用更专门的软件应用。此外，机构需要确保这些应用在异质硬件平台和系统上的共存并且保证在应用和系统之间共享数据的能力。

[0002] 数据集成是资源密集型过程，它需要具有特殊设计的软件的专用服务器，其中软件将服务器具体地配置为执行从一个系统到另一个系统的数据迁移。在这些领域中，通常导致差的或者低效的性能。

[0003] 因此，期望解决与开发数据集成场景相关的问题，其中一些场景可以在本文进行讨论。此外，期望减少与开发数据集成场景相关的缺陷，其中一些场景可以在本文进行讨论。

发明内容

[0004] 至少为了提供对主题的基本理解，本公开内容的以下部分给出对在本公开内容中找到的一个或多个创新、实施例和/或例子的简化综述。该综述不是要提供对任何特定实施例或例子的排它概述。此外，该综述不是要识别实施例或例子的关键/重要元件或者勾勒本公开内容的主题的范围。因此，该综述的一个目的可以是以简化的形式给出在本公开内容中找到的一些创新、实施例和/或例子，作为对随后给出的更详细描述的序言。

[0005] 在各种实施例中，数据集成系统使用户能够创建独立于平台和技术的逻辑设计。用户可以创建在高级别定义用户想让数据如何在源和目标之间流动的逻辑设计。工具可以鉴于用户的基础设施来分析逻辑设计并且创建物理设计。逻辑设计可以包括对应于设计中每个源和目标、以及诸如连接或过滤的操作和访问点的多个组件。当被转移到物理设计时，每个组件生成对数据执行操作的代码。依赖于底层技术(例如，SQL服务器、Oracle、Hadoop，等等)和所使用的语言(SQL、pig，等等)，由每个组件生成的代码可以不同。

[0006] 在一个方面中，不需要数据集成系统的用户从开始到结束在逻辑设计中的每个组件处指定所有数据属性。数据集成系统提供避免需要完全声明流经逻辑设计的信息的多个组件类型，诸如投影器类型和选择器类型。数据集成系统能够决定在由预定组件类型表示的操作处需要什么属性。这简化了设计和维护两者。在各个方面中，提供了充分利用现有RDBMS资源和能力以避免需要单独的专用ETL服务器来实现改进的性能的数据变换和迁移。

[0007] 在一种实施例中，一种用于促进数据映射的生成的方法包括接收指定逻辑设计的一个或多个组件的信息，其中该一个或多个组件当中至少一个组件具有第一类型。基于逻辑设计中的上游组件，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合。然后生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的信息。在一个方面中，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合可以包括派生从上游组件可见的一个或多个属性并且将该一个或多个属性暴露给下游组件。在另一个方面中，接收指定逻辑设计的一个或多个组件的信息可以包括接

收指示改变流经该逻辑设计的信息的形状的操作的信息。

[0008] 在一些实施例中，接收指定逻辑设计的一个或多个组件的信息可以包括接收指示控制流经该逻辑设计的信息流但是不改变流经该逻辑设计的信息的形状的操作的信息。在更多实施例中，接收指定逻辑设计的一个或多个组件的信息可以包括接收指示具有存储在源数据存储仓中的数据的一个或多个属性的源组件的信息。接收指定逻辑设计的一个或多个组件的信息可以包括接收指示具有要存储在目标数据存储仓中的数据的一个或多个属性的目标组件的信息。

[0009] 生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的信息可以包括将属性列表导出到下游组件。在一种实施例中，通过将组件或属性引入到逻辑设计中或从逻辑设计中去除组件或属性，可以接收逻辑设计中的变化。确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件。基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，可以确定对下游组件可见的已更新的数据属性集合。

[0010] 在更多实施例中，通过将组件或属性引入到逻辑设计中，接收逻辑设计中的变化。可以确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件。基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，可以保留(preserve)对下游组件可见的数据属性集合。在另一个方面中，基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，可以自动重命名对下游组件可见的数据属性集合。

[0011] 在一种实施例中，一种非临时性计算机可读介质存储用于促进数据映射的生成的计算机可执行代码。该非临时性计算机可读介质可以包括用于接收指定逻辑设计的一个或多个组件的信息的代码，其中该一个或多个组件当中至少一个组件具有第一类型；用于基于逻辑设计中的上游组件，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合的代码；以及用于生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合的信息的代码。

[0012] 在另一种实施例中，一种用于促进数据映射的生成的系统可以包括处理器和与处理器通信并且被配置为存储指令集合的存储器，当指令集合被处理器执行时，将处理器配置为接收指定逻辑设计的一个或多个组件的信息，其中该一个或多个组件当中至少一个组件具有第一类型；基于逻辑设计中的上游组件，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合；以及生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合的信息。

[0013] 在一种实施例中，一种用于促进数据映射的生成的系统包括接收单元，被配置为接收指定逻辑设计的一个或多个组件的信息，其中该一个或多个组件当中至少一个组件具有第一类型；确定单元，被配置为基于逻辑设计中的上游组件，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合；及生成单元，被配置为生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合的信息。确定一个或多个组件当中具有第一类型的至少

一个组件的、对逻辑设计中的下游组件可见的属性集合可以包括派生从上游组件可见的一个或多个属性并且将该一个或多个属性暴露给下游组件。

[0014] 指定逻辑设计的一个或多个组件的信息可以包括指示改变流经该逻辑设计的信息的形状的操作的信息。指定逻辑设计的一个或多个组件的信息可以包括指示控制流经该逻辑设计的信息流但是不改变流经该逻辑设计的信息的形状的操作的信息。指定逻辑设计的一个或多个组件的信息可以包括指示具有存储在源数据存储仓中的数据的一个或多个属性的源组件的信息。指定逻辑设计的一个或多个组件的信息可以包括指示具有要存储在目标数据存储仓中的数据的一个或多个属性的目标组件的信息。生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的信息可以包括将属性列表导出到下游组件。

[0015] 在一种实施例中，接收单元还被配置为通过将组件或属性引入到逻辑设计中或从逻辑设计中去除组件或属性来接收逻辑设计中的变化；确定单元还被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件；并且确定单元还被配置为，基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，确定对下游组件可见的已更新的数据属性集合。

[0016] 在一个方面中，该系统还可以包括保存单元，其中接收单元还被配置为通过将组件或属性引入到逻辑设计中来接收逻辑设计中的变化；确定单元还被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件；并且保存单元还被配置为，基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，保存对下游组件可见的数据属性集合。

[0017] 在另一个方面中，该系统还可以包括重命名单元，其中接收单元还被配置为接收逻辑设计中重命名组件或属性的变化；并且确定单元还被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件；并且重命名单元被配置为，基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，重命名对下游组件可见的数据属性集合。

[0018] 除以上部分之外，通过参考本公开内容的剩余部分、任何附图以及权利要求，还应当认识到对本公开内容主题的本质及其等同物(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。

附图说明

[0019] 为了合理地描述和说明在本公开内容中找到的那些创新、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任何要求保护的发明的范围、任何目前描述的实施例和/或例子、或者目前被理解为本公开内容中给出的任何创新的最佳模式的限制。

[0020] 图1是可以结合本发明实施例的系统的简化说明。

[0021] 图2是根据本发明实施例的数据集成系统的框图。

[0022] 图3是可以被用来实现根据本发明实施例的数据集成系统的硬件/软件栈的简化框图。

[0023] 图4是在本发明各种实施例中具有可以为其创建数据集成场景的各种异质数据源

的环境的框图。

[0024] 图5A和5B绘出了可以由数据集成系统执行的常规数据集成处理中的简化数据流。

[0025] 图6A和6B绘出了根据本发明实施例、在可以由数据集成系统执行的下一代数据集成处理中的简化数据流。

[0026] 图7是在根据本发明的一种实施例中数据集成系统的ODI工作室与储存库之间交互的简化框图。

[0027] 图8绘出了根据本发明实施例用于创建数据集成场景的方法的流程图。

[0028] 图9是根据本发明实施例用于创建数据集成场景的用户界面的屏幕截图。

[0029] 图10绘出了根据本发明实施例用于创建映射的方法的流程图。

[0030] 图11是根据本发明实施例用于在数据集成场景中提供映射信息的用户界面的屏幕截图。

[0031] 图12是根据本发明实施例用于在数据集成场景中提供流信息的用户界面的屏幕截图。

[0032] 图13绘出了根据本发明实施例用于创建包的方法的流程图。

[0033] 图14是根据本发明实施例用于在数据集成场景中提供包顺序信息的用户界面的屏幕截图。

[0034] 图15绘出了根据本发明实施例用于部署数据集成场景的方法的流程图。

[0035] 图16A和16B是在根据本发明的一种实施例中的映射的简化框图。

[0036] 图17绘出了根据本发明实施例用于基于组件类型派生(derive)组件属性的方法的流程图。

[0037] 图18是在根据本发明的一种实施例中具有过滤器组件的图16A和16B的映射的简化框图。

[0038] 图19是在根据本发明的一种实施例中的映射的简化框图。

[0039] 图20绘出了根据本发明实施例用于生成组件的方法的流程图。

[0040] 图21是可以被用来实践本发明实施例的计算机系统的简化框图。

[0041] 图22是根据实施例用于促进数据映射的生成的系统的简化框图。

具体实施方式

[0042] 介绍

[0043] 在各种实施例中，数据集成系统使用户能够创建独立于平台和技术的逻辑设计。用户可以创建在高级别定义用户想让数据如何在源和目标之间流动的逻辑设计。工具可以鉴于用户的基础设施来分析逻辑设计并且创建物理设计。逻辑设计可以包括对应于设计中每个源和目标、以及诸如连接或过滤的操作和访问点的多个组件。当被转移到物理设计时，每个组件生成对数据执行操作的代码。依赖于底层技术(例如，SQL服务器、Oracle、Hadoop，等等)和所使用的语言(SQL、pig，等等)，由每个组件生成的代码可以不同。

[0044] 在一个方面中，不需要数据集成系统的用户从开始到结束在逻辑设计中的每个组件处指定所有数据属性。数据集成系统提供避免需要完全声明流经逻辑设计的信息的多个组件类型，诸如投影器类型和选择器类型。数据集成系统能够决定在由预定组件类型表示的操作处需要什么属性。这简化了设计和维护。

[0045] 图1是可以结合在本公开内容中发现的任何创新、实施例和/或例子的实施例或者结合到其中的系统100的简化说明。图1仅仅是说明了结合本发明的实施例并且不限制如权利要求中所述的本发明的范围。本领域普通技术人员将认识到其它变体、修改和备选方案。

[0046] 在一种实施例中，系统100包括一个或多个用户计算机110(例如，计算机110A、110B和110C)。用户计算机110可以是通用个人计算机(仅仅作为例子，包括运行任何适当特点(flavor)的微软公司的Windows™和/或Apple公司的Macintosh™操作系统的个人计算机和/或膝上型计算机)和/或运行各种市售UNIX™或类UNIX操作系统当中任何一种的工作站计算机。这些用户计算机110还可以具有各种应用当中任何一种，这些应用包括被配置为执行本发明方法的一个或多个应用，以及一个或多个办公室应用、数据库客户端和/或服务器应用，及web浏览器应用。

[0047] 作为替代，用户计算机110可以是能够经由网络(例如，以下描述的通信网络120)通信和/或显示和导航网页或者其它类型的电子文档的任何其它电子设备，诸如瘦客户端计算机、启用互联网的游戏系统，和/或个人数字助理。虽然示例性系统100被示为具有三个用户计算机，但是可以支持任何数量的用户计算机或设备。

[0048] 本发明的某些实施例在联网环境中操作，该联网环境可以包括通信网络120。通信网络120可以是本领域技术人员熟悉的任何类型的网络，这些网络可以利用任何各种商用协议(包括但不限于TCP/IP、SNA、IPX、AppleTalk等)来支持数据通信。仅仅作为例子，通信网络120可以是局域网(“LAN”)，包括但不限于以太网、令牌环网等；广域网；虚拟网络，包括但不限于虚拟专用网(“VPN”)；互联网；内联网；外联网；公共交换电话网(“PSTN”)；红外线网络；无线网络，包括但不限于依据IEEE 802.11协议套件、本领域中已知的Bluetooth™协议和/或任何其它无线协议操作的网络；和/或这些和/或其它网络的任意组合。

[0049] 本发明的实施例可以包括一个或多个服务器计算机130(例如，计算机130A和130B)。每个服务器计算机130可以被配置为具有包括但不限于以上讨论的任何一种操作系统，以及任何市售服务器操作系统。每个服务器计算机130还可以运行一个或多个应用，这些应用可以被配置为向一个或多个客户端(例如，用户计算机110)和/或其它服务器(例如，服务器计算机130)提供服务。

[0050] 仅仅作为例子，其中一个服务器计算机130可以是web服务器，仅仅作为例子，web服务器可以被用来处理来自用户计算机110的对网页或其它电子文档的请求。web服务器还可以运行各种服务器应用，包括HTTP服务器、FTP服务器、CGI服务器、数据库服务器、Java服务器，等等。在本发明的一些实施例中，web服务器可以被配置为提供可以在用户计算机110当中一个或多个上的web浏览器中操作的网页，以执行本发明的方法。

[0051] 在一些实施例中，服务器计算机130可以包括一个或多个文件和/或应用服务器，这可以包括可由在用户计算机110和/或其它服务器计算机130当中一个或多个上运行的客户端访问的一个或多个应用。仅仅作为例子，服务器计算机130当中一个或多个可以是能够响应于用户计算机110和/或其它服务器计算机130而执行程序或脚本的一个或多个通用计算机，包括但不限于web应用(在一些情况下，这可以被配置为执行本发明的方法)。

[0052] 仅仅作为例子，web应用可以被实现为以任何编程语言以及任何编程/脚本语言的组合所写的脚本或程序，这些编程语言诸如Java、C或C++和/或任何脚本语言，诸如Perl、Python或TCL。(一个或多个)应用服务器还可以包括数据库服务器，包括但不限于可从

Oracle,Microsoft、IBM等商购的那些,数据库服务器可以处理来自其中一个用户计算机110和/或另一服务器计算机130上运行的数据库客户端的请求。

[0053] 在一些实施例中,应用服务器可以动态地创建网页,用于显示根据本发明的实施例的信息。由应用服务器提供的数据可以被格式化为网页(例如,包括HTML、XML、Javascript、AJAX等)和/或可以经由web服务器被转发到其中一个用户计算机110(例如,如上所述)。类似地,web服务器可以从其中一个用户计算机110接收网页请求和/或输入数据和/或将网页请求和/或输入数据转发到应用服务器。

[0054] 根据更多的实施例,服务器计算机130当中的一个或多个可以充当文件服务器和/或可以包括实现由其中一个用户计算机110和/或另一服务器计算机130上运行的应用所结合的本发明方法所必需的文件当中的一个或多个。作为替代,如本领域技术人员将认识到的,文件服务器可以包括所有必需的文件,从而允许这种应用被用户计算机110和/或服务器计算机130当中的一个或多个远程调用。应当注意的是,依赖于特定于实现的需求和参数,本文关于各种服务器(例如,应用服务器、数据库服务器、web服务器、文件服务器,等等)所描述的功能可以由单个服务器和/或多个专用服务器执行。

[0055] 在某些实施例中,系统100可以包括一个或多个数据库140(例如,数据库140A和140B)。(一个或多个)数据库140的位置是任意的:仅仅作为例子,数据库140A可以驻留在服务器计算机130A(和/或用户计算机110当中的一个或多个)本地的存储介质上(和/或驻留在其中)。作为替代,数据库140B可以远离用户计算机110和服务器计算机130当中任何一个或全部,只要它可以与这些计算机当中一个或多个通信就可以(例如,经由通信网络120)。在特定的一组实施例中,数据库140可以驻留在本领域技术人员熟悉的存储区域网络(“SAN”)中。(同样,用于执行用户计算机110和服务器计算机130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相应的计算机上和/或远程存储。)在一组实施例中,数据库140当中一个或多个可以是适于响应于SQL格式的命令而存储、更新和检索数据的关系型数据库。例如,数据库140可以由数据库服务器控制和/或维护,如上所述。

[0056] 数据集成概述

[0057] 图2是根据本发明实施例的数据集成系统200的简化框图。图2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数据集成系统200的简化说明。图2仅仅是说明本文所公开的发明的实施例或实现不应当限制如权利要求中所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以及本文给出的示教认识到对附图中示出的那些实施例或实现的其它变化、修改和/或备选方案。

[0058] 在这种实施例中,数据集成系统200包括信息源202、信息集成204和信息目的地206。一般而言,信息从信息源202流到信息集成204,由此信息可以被信息目的地206消费、获得或者以别的方式使用。数据流可以是单向的或双向的。在一些实施例中,在数据集成系统200中可以存在一个或多个数据流。

[0059] 信息源202表示被配置为提供数据来源的一个或多个硬件和/或软件元件。信息源202可以提供对数据的直接或间接访问。在这种实施例中,信息源202包括一个或多个应用208和一个或多个储存库210。

[0060] 应用208表示传统应用,诸如桌面、被托管的、基于web的或者基于云的应用。应用208可以被配置为为了一个或多个预定目的而接收、处理和维护数据。应用208的一些例子

包括客户关系管理(CRM)应用、金融服务应用、政府和风险合规应用、人力资本管理(HCM)、采购应用、供应链管理应用、项目或投资组合管理应用，等等。应用208可以包括被配置为用于以各种人类可读和机器可读的格式操纵和导出应用数据的功能，如本领域中已知的。应用208还可以访问存储库210中的数据并在存储库210中存储数据。

[0061] 储存库210表示被配置为提供对数据的访问的硬件和/或软件元件。储存库210可以提供数据的逻辑和/或物理分区。储存库210还可以提供报告和数据分析。储存库210的一些例子包括数据库、数据仓库、云存储，等等。储存库可以包括通过集成来自一个或多个应用208的数据所创建的中央储存库。存储在储存库210中的数据可以从操作系统上载。在源中可用之前，数据可以传递通过附加的操作。

[0062] 信息集成204表示被配置为提供数据集成服务的一个或多个硬件和/或软件元件。可以在信息集成204中提供直接或间接的数据集成服务。在这种实施例中，信息集成204包括数据迁移212、数据入库214、主数据管理216、数据同步218、联合220和实时消息传送222。可以理解，信息集成204可以包括与这里所示那些不同的提供数据集成功能的一个或多个模块、服务或其它附加元件。

[0063] 数据迁移212表示被配置为提供数据迁移的一个或多个硬件和/或软件元件。一般而言，数据迁移212提供用于在存储类型、格式或系统之间传送数据的一个或多个过程。数据迁移212通常提供手动或编程选项来实现迁移。在数据迁移过程中，一个系统上的数据或者由一个系统提供的数据被映射到另一个系统，从而提供用于数据提取和数据加载的设计。数据迁移可以涉及一个或多个阶段，诸如创建将第一系统的数据格式关联到第二系统的格式和需求的一个或多个设计的设计阶段，在其中从第一系统读取数据的数据提取阶段，数据清除(cleansing)阶段，以及其中数据被写到第二系统的数据加载阶段。在一些实施例中，数据迁移可以包括确定数据在上述任何阶段中是否被准确处理的数据验证阶段。

[0064] 数据入库214表示被配置为提供用于报告和数据分析的数据库的一个或多个硬件和/或软件元件。数据仓库通常被视为数据的中央储存库，它是通过集成来自一个或多个不同源的数据创建的。数据入库214可以包括数据的当前存储以及历史数据的存储。数据入库214可以包括典型的基于提取、变换、加载(ETL)的数据仓库，由此暂存层(staging layer)、数据集成层和访问层包含(house)关键功能。在一个例子中，暂存层或暂存数据库存储从一个或多个不同的源数据系统中每一个当中提取的原始数据。集成层通过变换来自暂存层的数据来集成不同的数据集，从而常常将这种变换后的数据存储在操作数据存储仓(ODS)数据库中。然后，集成的数据被移动到常常被称为数据仓库(data warehouse)数据库的另一个数据库。数据可以被布置成分层的组(常常被称为维度)并且被布置成事实和汇总(aggregate)事实。可以提供访问层来帮助用户或其它系统检索数据。数据仓库可以被细分为数据集市(data mart)，由此每个数据集市存储来自仓库的数据的子集。在一些实施例中，数据入库214可以包括商业智能工具，用于提取、变换并将数据加载到储存库中的工具，以及用于管理和检索元数据的工具。

[0065] 主数据管理216表示被配置为管理数据的主拷贝的一个或多个硬件和/或软件元件。主数据管理216可以包括一致地定义和管理主数据的一组过程、管理(governance)、策略、标准和工具。主数据管理216可以包括用于除去重复数据、标准化数据并且结合规则以便消除不正确的数据进入系统以便创建主数据的权威来源的功能。主数据管理216可以提

供用于收集、汇总、匹配、整合、质量保证、持久化和在整个组织中分发数据以确保正在进行的信息的维护和应用使用中的一致性和控制的过程。

[0066] 数据同步218表示被配置为同步数据的一个或多个硬件和/或软件元件。数据同步218可以提供在从源到目标的数据中建立一致性并且反之亦然。数据同步218还可以提供数据随时间推移的连续协调(harmonization)。

[0067] 联合220表示被配置为整合来自组成源的数据视图的一个或多个硬件和/或软件元件。联合220可以透明地将多个自治的数据库系统映射到单个联合数据库。组成数据库可以经由计算机网络互连并且可以在地理上分散。联合220提供了合并几个完全不同的数据库的备选方案。例如,联合数据库或虚拟数据库可以提供所有组成数据库的合成物。联合220可能无法提供完全不同的组成数据库的真正数据集成,而仅仅是在视图中提供。

[0068] 联合220可以包括提供统一用户界面的功能,从而使用户和客户端能够利用单个查询在多个不连续的数据库中存储和检索数据——即使组成数据库是异质的。联合220可以包括将查询分解为子查询以用于提交到相关的组成数据并且合成子查询的结果集的功能。联合220可以包括对子查询的一个或多个封装(wrapper),以便将子查询转化为适当的查询语言。在一些实施例中,联合220是自治组件的集合,这些自治组件通过公布导出模式和访问操作使它们的数据能够可由该联合的其它成员获得。

[0069] 实时消息传送222表示被配置为提供受制于实时约束(例如,从事件到系统响应的操作期限)的消息传送服务的一个或多个硬件和/或软件元件。实时消息传送222可以包括保证动作或响应在严格时间限制内的功能。在一个例子中,实时消息传送222可以负责从一个数据库取一些订单和顾客数据,将其与保存在文件中的一些雇员数据结合,然后将集成的数据加载到Microsoft SQL服务器2000数据库中。因为订单需要在它们到达的时候被分析,所以实时消息传送222可以尽可能接近实时地将订单传递至目标数据库并且只提取新的和改变的数据,以保持工作量(workload)尽可能小。

[0070] 信息目的地206表示被配置为存储或消费数据的一个或多个硬件和/或软件元件。在这种实施例中,信息目的地206可以提供对数据的直接或间接访问。在这种实施例中,信息目的地206包括一个或多个应用224和一个或多个储存库226。

[0071] 应用224表示传统应用,诸如桌面、被托管的、基于web的或基于云的应用。应用224可以被配置为为了一个或多个预定目的而接收、处理和维护数据。应用224的一些例子包括客户关系管理(CRM)应用、金融服务应用、政府和风险合规应用、人力资本管理(HCM)、采购应用、供应链管理应用、项目或投资组合管理应用,等等。应用224可以包括被配置为以各种人类可读和机器可读的格式操纵和导入(import)应用数据的功能,如本领域已知的。应用224还可以访问储存库226中的数据并在储存库226中存储数据。

[0072] 储存库226表示提供对数据的访问的硬件和/或软件元件。储存库226可以提供数据的逻辑和/或物理分区。储存库226还可以提供报告和数据分析。储存库226的一些例子包括数据库、数据仓库、云存储,等等。储存库可以包括通过集成来自一个或多个应用226的数据创建的中央储存库。存储在储存库226中的数据可以通过信息集成204被上载或导入。在使数据在目的地可用之前,数据可以传递通过附加的操作。

[0073] 数据集成系统

[0074] 图3是可以被用来实现根据本发明实施例的数据集成系统200的硬件/软件栈的简

化框图。图3仅仅是说明本文所公开的发明的实施例或实现不应当限制如权利要求中所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以及本文给出的示教认识到对附图中示出的那些实施例或实现的其它变化、修改和/或备选方案。在根据这种实施例的数据集成系统200中找到的组件的一个例子可以包括，ORACLE数据集成器(ORACLE DATA INTEGRATOR)，其是由位于加州Redwood Shores的Oracle提供的ORACLE融合(ORACLE FUSION)中间件产品系列的成员。ORACLE DATA INTEGRATOR是基于Java的应用，它使用一个或多个数据库执行基于集合的数据集成任务。此外，ORACLE DATA INTEGRATOR可以提取数据，通过Web服务和消息提供变换后的数据，并且创建对面向服务的体系架构中的事件做出响应并在面向服务的体系架构中创建事件的集成过程。ORACLE DATA INTEGRATOR是基于ELT[提取-加载和变换]体系架构而不是常规的ETL[提取-变换-加载]体系架构的。用于ORACLE DATA INTEGRATOR的用户手册的拷贝附到本公开内容并且通过引用被结合于此，用于所有目的。

[0075] 在各种实施例中，数据集成系统200提供新的声明设计方法，以定义数据变换和集成过程，从而产生更快更简单的开发和维护。因此，数据集成系统200使声明规则与实现细节分离。数据集成系统200还为数据变换和验证过程的执行提供独特的E-LT体系架构(提取-加载变换)。实施例中的这种体系架构消除了对单独ETL服务器和专用引擎的需求。在一些实施例中，数据集成系统200代替地充分利用RDBMS引擎的固有能力。

[0076] 在一些实施例中，数据集成系统200集成在一个或多个中间件软件包中(诸如ORACLE FUSION MIDDLEWARE平台)，并且变成中间件栈的组件。如图3中所绘出的，数据集成系统200可以提供运行时组件作为Java EE应用。

[0077] 在这个例子中，数据集成系统200的一个组件是储存库302。储存库302表示被配置为存储关于IT基础设施、所有应用的元数据、项目、场景和执行日志的配置信息的硬件和/或软件元件。在一些方面，储存库302的多个实例可以在IT基础设施中共存，例如，开发、QA、用户、验收和生产。储存库302被配置为允许交换元数据和场景的几个单独的环境(例如：开发、测试、维护和生产环境)。储存库302还被配置为充当版本控制系统，其中对象被归档并被指定了版本号。

[0078] 在这个例子中，储存库302由至少一个主储存库304和一个或多个工作储存库306组成。为在数据集成系统200中使用而开发或配置的对象可以存储在这些储存库类型之一当中。一般而言，主储存库304存储以下信息：包括用户、简档和权限的安全信息，包括技术、服务器定义、模式、语境、语言等的拓扑信息，以及版本化和归档的对象。一个或多个工作储存库306可以包含真正开发的对象。

[0079] 几个工作储存器可以在数据集成系统200中共存(例如，具有单独的环境或者匹配特定的版本生命周期)。这一个或多个工作储存库306存储用于模型的信息，包括模式定义、数据存储仓结构和元数据、字段和列定义、数据质量约束、交叉引用、数据沿袭(lineage)，等等。这一个或多个工作储存库306还可以存储项目(包括商业规则、包、过程、文件夹、知识模块、变量等)以及场景执行(包括场景、调度信息和日志)。在一些方面中，这一个或多个工作储存库306可以只包含执行信息(通常用于生产目的)，并且被指定为执行储存库。

[0080] 在各种实施例中，储存库302存储一个或多个ETL项目。ETL项目定义或以别的方式指定对源或目标中的数据的数据属性建模的一个或多个数据模型。ETL项目还提供数据质

量控制以及定义用于移动和变换数据的映射。数据完整性控制确保数据的整体一致性。对于由特定源或目标强加的约束和声明规则，应用数据不是总有效。例如，订单可能被发现没有顾客，或者订单行没有产品，等等。数据集成系统200提供用于检测这些约束违背并且为了回收或报告目的而存储它们的工作环境。

[0081] 在数据集成系统200的一些实施例中，存在两种不同类型的控制：静态控制和流控制。静态控制暗示被用来验证应用数据的完整性的规则的存在。这些规则当中的一些(被称为约束)已经在数据服务器中实现(利用主键、引用约束，等等)。数据集成系统200允许附加约束的定义和检查，而无需在源中直接声明它们。流控制涉及实现其自己的声明规则的变换和集成过程的目标。在将数据加载到目标之前，流控制根据这些约束验证应用的进入的数据。流控制过程一般被称为映射。

[0082] ETL项目可以被自动化到可以为了在运行时环境中执行而部署的包中。因此，通过顺序化包中的不同步骤(映射、过程等)的执行并且通过产生包含用于这些步骤当中每一个的就绪(ready-to-use)代码的生产场景来实现数据集成流的自动化。包通常由组织成执行图的一系列步骤组成。包是被用来生成生产场景的主要对象。它们表示数据集成工作流并且可以执行作业，诸如像：开始对数据存储仓或模型的逆向工程设计过程、向管理员发送电子邮件、下载文件并且解压该文件、定义映射必须以其执行的次序，以及定义利用变化的参数对执行命令迭代的循环。

[0083] 场景被设计为将源组件(映射、包、过程、变量)投产。场景来自对这个组件的代码的生成(SQL、shell等)。一旦被生成，源组件的代码就被冻结并且该场景被存储在储存库302中，诸如工作储存库306当中的一个或多个。场景可以被导出，然后导入不同的生产环境中。

[0084] 在各种实施例中，数据集成系统200以被Java图形模块和调度代理访问的模块化方式围绕储存库302组织。图形模块可以被用来设计和建立存储在储存库302中的一个或多个集成过程。管理员、开发人员和操作人员可以使用开发工作室访问储存库302。代理可以被用来调度和协调与存储在储存库302中的集成过程关联的一组集成任务。例如，在运行时，部署在桌面、web服务上或者以别的方式与源通信的代理协调一个或多个集成过程的执行。代理可以检索存储在主储存库304中的代码、连接到各个源和目标系统，并且编排(orchestrate)整体数据集成过程或场景。

[0085] 在这种实施例中，数据集成系统200包括桌面308，该桌面308可以包括以上讨论的图形模块和/或代理当中一个或多个。桌面308表示一个或多个桌面或工作站计算设备，诸如个人计算机、笔记本电脑、上网本电脑、平板电脑，等等。桌面308包括Java虚拟机(JVM)310和Oracle数据集成器(ODI)工作室312。Java虚拟机(JVM)310是可以执行Java字节码的虚拟机。JVM 310最经常被实现为在现有操作系统上运行，但是也可以被实现为直接在硬件上运行。JVM 310提供Java字节码可以在其中被执行的运行时环境，从而启用诸如运行时web服务(WS)314和代理316的特征。JVM310可以包括Java类库，实现Java应用编程接口(API)的一组标准的类库(在Java字节码中)，以及构成Java运行时环境(JRE)的其它元件。

[0086] 代理316被配置为调度和协调与存储在储存库302中的一个或多个集成过程关联的一组集成任务。例如，在运行时，代理协调集成过程的执行。代理可以检索存储在主储存库304中的代码，连接到各个源和目标系统，并且编排整个数据集成过程或场景。

[0087] 再次参考图3,ODI工作室312包括被配置为设计数据集成项目的硬件和/或软件元件。在这个例子中,ODI工作室312包括被用来创建和管理数据集成项目的四个图形模块或导航器,即,设计人员模块318、操作人员模块320、拓扑模块322和安全模块324。设计人员模块318是被配置为定义数据存储仓(表、文件、Web服务等)、数据映射和包(集成步骤的集合,包括映射)的模块。在各种实施例中,设计人员模块318定义用于数据变换和数据完整性的声明规则。因此,项目开发在设计人员模块318中发生。此外,数据库和应用元数据在设计人员模块318中被导入和定义。在一种实施例中,设计人员模块318使用元数据和规则来生成数据集成场景或加载生产计划。一般而言,设计人员模块318被用来设计数据完整性检查并且建立变换,诸如像:现有应用或数据库的自动逆向工程设计、变换和集成映射的图形开发和维护、映射中数据流的可视化、自动文档生成以及所生成的代码的定制。

[0088] 操作人员模块320是被配置为查看和管理生产集成作业的模块。因此,操作人员模块320管理和监视生产中的数据集成过程并且可以显示带错误计数、已处理的行数、执行统计、被执行的实际代码等等的执行日志。在设计时,开发人员还可以联系设计人员模块318使用操作人员模块320以用于调试目的。

[0089] 拓扑模块322是被配置为创建和管理到数据源和代理的连接的模块。拓扑模块322定义基础设施的物理和逻辑体系架构。基础设施或项目管理员可以通过拓扑模块322在主储存库中注册服务器、数据库模式和目录以及代理。安全模块324是被配置为管理用户和他们的储存库特权的模块。

[0090] 一般而言,用户或过程与设计人员模块318交互,以便为源和目标326创建具有一个或多个数据集成过程的数据集成项目。每个数据集成过程包括至少一个数据集成任务。在一些实施例中,数据集成任务由指示什么数据位要被变换并与其它位组合以及数据如何被真正提取、加载等的技术细节的一组商业规则来定义。在优选实施例中,数据集成任务是利用构建数据映射的声明方法指定的。映射是填充被称为目标的一个数据存储仓的对象,其数据来自被称为源的一个或多个其它数据存储仓。一般而言,源数据存储仓中的列通过映射被链接到目标数据存储仓中的列。映射可以被添加到包以作为包步骤。如以上所讨论的,包定义数据集成作业。包是在项目之下创建的并且由有组织的步骤序列组成,每个步骤可以是映射或过程。包可以具有一个入口点和多个出口点。

[0091] 在一些实施例中,当创建新映射时,开发人员或技术业务人员与设计人员318交互,以便首先定义哪些数据要被集成并且哪些业务规则应当被使用。例如,开发人员可以指定哪些表要被连接,哪些过滤器要被应用,以及哪些SQL表达式要被用来变换数据。被使用的SQL的特定方言(dialect)由代码要在其上执行的数据库平台确定。然后,在单独的步骤中,技术人员可以与设计人员318交互,以选择最高效的方式来提取、组合,然后集成这种数据。例如,技术人员可以使用特定于数据库的工具和设计技术,诸如增量加载、批量加载实用工具、渐变维度和更改数据的捕获。

[0092] 在这种实施例中,可以为源和目标326创建映射。源和目标326可以包括一个或多个传统应用328、一个或多个文件/XML文档330、一个或多个应用332、一个或多个数据仓库(DW)、商业智能(BI)工具和应用、企业过程管理(EPM)工具和应用334,以及一个或多个JVM 336(包括运行时web服务340和代理342)。

[0093] 图4是在本发明各种实施例中具有可以为其创建数据集成场景的各种异质数据源

的环境400的框图。在这个例子中，环境400包括ODI工作室312和储存库302。储存库302包含生成集成场景400所需的所有元数据。用户或过程与ODI工作室312交互，以利用数据完整性控制402和声明规则404创建集成场景400。

[0094] 订单应用406表示用于跟踪顾客订单的应用。创建“订单应用”数据模型来表示存储在订单应用406中的数据以及任何数据完整性控制或条件。例如，“订单应用”数据模型可以基于超结构化查询语言数据库(HSQLDB)映射并且包括五个数据存储仓，SRC_CITY、SRC_CUSTOMER、SRC_ORDERS、SRC_ORDER_LINES、SRC_PRODUCT和SRC_REGION。

[0095] 参数文件408表示从生产系统发出的平面文件(例如，ASCII)，该文件包含销售代表的列表以及分成年龄范围的年龄段。在这个例子中，创建“参数”数据模型来表示该平面文件中的数据。例如，“参数”数据模型可以基于文件接口并且包括两个数据存储仓，SRC_SALES_PERSON和SRC_AGE_GROUP。

[0096] 销售管理应用410表示用于跟踪销售的应用。销售管理应用410可以是用来自订单应用406和参数文件408的数据的变换填充的数据仓库。创建“销售管理”数据模型来表示存储在销售管理应用410中的数据以及任何数据完整性控制或条件或变换。例如，“销售管理”数据模型可以基于超结构化查询语言数据库(HSQLDB)映射并且包括六个数据存储仓，TRG_CITY、TRG_COUNTRY、TRG_CUSTOMER、TRG_PRODUCT、TRG_PROD_FAMILY、TRG_REGION和TRG_SALE。

[0097] 图5A和5B绘出了可以由数据集成系统200执行的常规数据集成处理中的简化数据流。在这个例子中，来自订单应用406、参数文件408和一个或多个其它可选的或附加源的数据流经针对销售管理应用410的传统ETL过程。数据变换发生在单独的ETL服务器500中。该场景需要专用或专有资源，导致较差的性能，并造成高成本。

[0098] 图6A和6B绘出了根据本发明实施例、在可以由数据集成系统200执行的下一代数据集成处理中的简化数据流。在这个例子中，来自订单应用406、参数文件408和一个或多个其它可选的或附加的数据源的数据流经针对销售管理应用410的E-LT过程。数据变换充分利用现有资源，从而导致更高的性能和效率。如上所述，现有的ETL系统需要专用和/或专有的基础设施来执行数据变换。这样做部分地是为了适应未知的用户基础设施。例如，在不知道正在使用什么类型的数据库的情况下，现有的ETL系统不能预料在给定系统中什么变换操作将可用。但是，这导致未充分利用资源，这些资源诸如能够在没有任何专用和/或专有的基础设施的情况下执行适当的数据变换的用户的现有数据库和服务器。

[0099] 根据实施例，本发明通过使用户能够根据用户的特定需求定制数据集成过程来充分利用用户的现有基础设施。例如，当设计数据集成计划时，它可以被分成可由单个系统执行的离散部分，称为执行单元。一旦数据集成计划已经被分成多个执行单元，就可以基于用户的基础设施和系统资源向用户呈现物理计划。这个计划可以由用户进一步定制，以改变哪些用户系统执行哪些执行单元。例如，可以向用户呈现其中连接操作对第一数据库执行的计划，并且用户可以通过将连接操作移动到第二数据库来定制该计划。

[0100] 如图6B中所示，这导致不依赖于作为现有ETL系统的特征的独立变换服务器的提取-加载-变换(E-LT)体系架构。相反，如上所述，可以在用户的现有基础设施上执行数据变换。E-LT体系架构为用户提供更大的灵活性，同时降低与获取并维护专用变换服务器相关联的成本。

[0101] 再次参考图3,可以使用代理来调度和协调与集成过程相关联的一组集成任务。例如,在运行时,代理协调集成过程的执行。代理可以检索存储在主储存库304中的代码,连接到各个源和目标系统,并且编排整体的数据集成过程或场景。在各种实施例中,存在两种类型的代理。在一个例子中,独立的代理(诸如代理316)被安装在桌面308上。在另一个例子中,应用服务器代理可以被部署在应用服务器326上(诸如部署在Oracle WebLogic服务器上的Java EE代理)并且可以受益于应用服务器层特征,诸如用于高可用性需求的群集。在还有另一个例子中,代理(诸如代理342)可以被部署在源和目标326上。

[0102] 在这种实施例中,数据集成系统200包括可包括以上讨论的代理中的一个或多个代理的应用服务器344。应用服务器344表示一个或多个应用服务器、web服务器,或被托管的应用。在这个例子中,应用服务器344包括FMW控制台346、小服务程序(servlet)容器348、web服务容器350,以及数据源连接池352。

[0103] FMW控制台346表示被配置为管理应用服务器344的各方面的一个或多个硬件和/或软件元件,其中所述各方面诸如与小服务程序容器348、web服务容器350和数据源连接池334相关的信息。例如,FMW控制台346可以是被用来管理Oracle WebLogic服务器域的基于浏览器的图形用户界面。FMW控制台346可以包括如下功能:配置、启动和停止WebLogic服务器实例,配置WebLogic服务器群集,配置WebLogic服务器服务(诸如数据库连接(JDBC)和消息传送(JMS)),配置安全参数(包括创建和管理用户、组和角色),配置和部署Java EE应用序,监视服务器和应用的性能,查看服务器和域日志文件,查看应用部署描述符,以及编辑选定的运行时应用部署描述符元件。在一些实施例中,FMW控制台346包括在生产中向FMW控制台346提供对数据集成过程的访问ODI插件354,并且可以显示带错误计数、处理的行数、执行统计数据、所执行的实际代码等等的执行日志。

[0104] 小服务程序容器348表示被配置为扩展应用服务器344的能力的一个或多个硬件和/或软件元件。小服务程序最常被用来处理或存储从HTML表单提交的数据,提供诸如数据库查询的结果的动态内容,并且管理在无状态HTTP协议中不存在的状态信息,诸如将物品填充到适当顾客的购物车中。小服务程序通常是Java EE中符合Java小服务程序API的Java类,其中Java小服务程序API是Java类可以通过其对请求作出响应的协议。为了部署和运行小服务程序,小服务程序容器348被用作与小服务程序交互的web服务器的组件。因此,小服务程序容器348可以扩展由web服务容器350的公共web服务356和数据服务358提供的功能,以及对由数据源连接池352提供的数据池的访问。小服务程序容器348还负责管理小服务程序的生命周期,将URL映射到特定的小服务程序并且确保URL请求者具有正确的访问权限。

[0105] 在这个例子中,小服务程序容器348包括与ODI SDK 362关联的Java EE应用360、ODI控制台364,以及与Java EE代理368关联的运行时web服务366。ODI SDK 362提供用于数据集成和ETL设计的软件开发工具包(SDK)。ODI SDK 362使工作中常见并且很重复的工作自动化,从而允许用户把重复的任务脚本化。

[0106] ODI控制台364是提供对储存库302的web访问的Java企业版(Java EE)应用。ODI控制台364被配置为允许用户浏览设计时对象,包括项目、模型和执行日志。ODI控制台364可以允许用户查看流映射、追踪所有数据的源并且甚至深入到字段级,以理解被用来建立数据的变换。此外,最终用户可以通过ODI控制台364启动和监视场景执行。在一个方面中,ODI控制台364为管理员提供查看和编辑诸如数据服务器、物理和逻辑模式的拓扑对象以及管

理储存库302的能力。

[0107] 数据场景设计和开发

[0108] 如以上所讨论的，设计场景来将源组件(映射、包、过程、变量)投产。场景来自对这个组件的代码(SQL、shell等)的生成。场景可以被导出，然后导入不同的生产环境。

[0109] 图7是在根据本发明的一种实施例中数据集成系统的ODI工作室与储存库之间交互的简化框图。在图7所示的实施例中，图3的ODI工作室312使用元数据和规则来生成用于生产的数据集成场景700。一般而言，设计人员模块318被用来设计数据完整性检查并建立变换，诸如像：现有应用或数据库的自动逆向工程设计、变换和集成映射的图形开发和维护、映射中数据流的可视化、自动文档生成，以及所生成的代码的定制。

[0110] 图8绘出了根据本发明实施例用于创建数据集成场景的方法800的流程图。图8中所绘出的方法800的实现方式或其中的处理可以在软件(例如，指令或代码模块)被诸如计算机系统或信息处理设备的逻辑机器的中央处理单元(CPU或处理器)执行时由软件执行、由电子设备或专用集成电路的硬件组件执行，或者由软件元件和硬件元件的组合执行。图8中所绘出的方法800开始于步骤810。

[0111] 在各种实施例中，用户可以启动与ODI工作室312的设计人员模块318的会话并且连接到储存库302。用户可以与一个或多个用户接口特征交互，以创建新的数据集成项目或者从存储在例如主储存库304中的现有数据集成项目选择。一般而言，设计人员模块318被用来管理元数据、设计数据完整性检查，并建立变换。在各种实施例中，通过设计人员模块318被处理的主要对象是模型和项目。数据模型包含数据源或目标(例如，表、列、约束、描述、交叉引用等)中的所有元数据。项目包含用于源或目标的所有加载和变换规则(例如，映射、过程、变量等)。

[0112] 在步骤820中，创建一个或多个数据模型。在步骤830中，创建一个或多个项目。图9是根据本发明实施例用于创建数据集成场景的用户界面900的屏幕截图。在这个例子中，导航面板910显示信息并且包括用于与项目交互的功能。导航面板920显示信息并且包括用于与数据模型交互的功能。如以上所讨论的，用户不仅可以创建数据模型，而且可以为数据模型中的数据开发任何数据完整性检查。此外，用户可以为项目指定映射、过程、变量，其为将数据从源加载到目标的流中的数据提供数据完整性以及变换。在步骤840中，生成一个或多个数据集成场景。图8在步骤850结束。

[0113] 图10绘出了根据本发明实施例用于创建映射的方法1000的流程图。图10中所绘的方法1000的实现方式或其中的处理可以在软件(例如，指令或代码模块)被诸如计算机系统或信息处理设备的逻辑机器的中央处理单元(CPU或处理器)执行时由软件执行、由电子设备或专用集成电路的硬件组件执行，或者由软件元件和硬件元件的组合执行。图10中所绘出的方法1000开始于步骤1010。

[0114] 在步骤1020中，接收目标数据存储仓信息。例如，用户可以与设计人员模块318的一个或多个用户接口特征交互，以提供目标数据存储仓信息。在一种实施例中，用户可以将包括来自导航面板910的一个或多个数据模型的目标数据存储仓信息拖放到映射或流面板上，其中该映射或流面板可视地表示选定的数据模型的各方面以及任何关联的变换或数据完整性检查。

[0115] 在步骤1030中，接收源数据存储仓信息。例如，用户可以与设计人员模块318的一

个或多个用户接口特征交互，以提供源数据存储仓信息。在一种实施例中，用户可以将包括来自导航面板910的一个或多个数据模型的源数据存储仓信息拖放到与目标数据存储仓信息相同的映射或流面板上，其中该映射或流面板可视地表示选定的数据模型的各方面以及任何关联的变换或数据完整性检查。

[0116] 在各种实施例中，源数据存储仓信息和目标数据存储仓信息可以包括一个或多个数据模型以及可选的操作。操作的一些例子可以包括一个或多个数据集操作(例如，联合、连接、交集等)、数据变换、数据过滤操作、约束、描述、交叉引用、完整性检查，等等。在更多实施例中，这些操作当中的一些可以在设计人员模块318中被预先配置并可视表示。在其它实施例中，可以提供定制的操作，从而允许用户指定实现操作的逻辑、映射等。

[0117] 在步骤1040中，接收映射信息。例如，用户可以与设计人员模块318的一个或多个用户接口特征交互，以便将源数据存储仓信息映射到目标数据存储仓信息。在一种实施例中，用户可以可视地将源数据存储仓信息中的数据元件的属性与目标数据存储仓信息中的数据元件的属性连接。这可以通过匹配源数据存储仓信息和目标数据存储仓信息中表的列名来完成。在更多实施例中，可以使用一种或多种自动映射技术来提供映射信息。

[0118] 图11是根据本发明实施例用于在数据集成场景中提供映射信息的用户界面1100的屏幕截图。在这个例子中，源组件的属性被映射到目标组件的属性。

[0119] 再次参考图10，在步骤1050中，接收数据加载策略。数据加载策略包括关于来自源数据存储仓信息的实际数据如何在提取阶段被加载的信息。可以在设计人员318的流选项卡(tab)中定义数据加载策略。在一些实施例中，依赖于映射的配置，可以为流自动计算数据加载策略。

[0120] 例如，可以为流建议一个或多个知识模块。知识模块(KM)是跨不同的技术实现可重用变换和ELT(提取、加载和变换)策略的组件。在一个方面中，知识模块(KM)是代码模板。每个KM可以专用于整个数据集合过程中的单独任务。KM中的代码看起来接近它将利用替代方法被执行的形式，从而使其能够一般性地被许多不同的集成作业使用。被生成和执行的代码是从在设计器模块318中定义的声明规则和元数据派生的。其一个例子是通过变化数据捕获(change data capture)从Oracle数据库10g提取数据并且将变换后的数据加载到Oracle数据库11g中的分区事实表中，或者从Microsoft SQL服务器数据库创建基于时间戳的提取物并且将这种数据加载到Teradata企业级数据仓库中。

[0121] KM的能力在于它们的可重用性和灵活性——例如，可以为一个事实表开发加载策略，然后该加载策略可以应用到所有其它事实表。在一个方面中，使用给定KM的所有映射继承对该KM所作的任何改变。在一些实施例中，提供了五种不同类型的KM，它们当中每一个覆盖从源到目标的变换过程中的一个阶段，诸如集成知识模块(IKM)、加载知识模块(LKM)，以及检查知识模块CKM。

[0122] 参考图4，用户可以定义在环境400中从SRC_AGE_GROUP、SRC_SALES_PERSON文件以及从SRC_CUSTOMER表中检索数据的方式。为了定义加载策略，用户可以选择对应于SRC_AGE_GROUP文件的加载的源集合并且选择LKM文件到SQL，以实现从文件到SQL的流。在一个方面中，LKM负责从远程服务器向暂存区域(staging area)加载源数据。

[0123] 在步骤1060中，接收数据集成策略。在定义了加载阶段之后，用户定义策略，以采用所加载数据到目标的集成。为了定义集成策略，用户可以选择目标对象并且选择IKM SQL

增量更新。IKM负责将最终变换后的数据写到目标。当IKM被启动时，它假设用于远程服务器的所有加载阶段都已经执行了它们的任务，诸如所有远程源数据集都已经被LKM加载到暂存区域中，或者源数据存储仓在与暂存区域相同的数据服务器上。

[0124] 在步骤1070中，接收数据控制策略。一般而言，CKM负责检查数据集的记录与所定义的约束一致。CKM可以被用来维护数据完整性并参与整体数据质量主动推荐(initiative)。CKM可以以两种方式被使用。首先，检查现有数据的一致性。这可以在任何数据存储仓上或者在映射内进行。在这种情况下，被检查的数据是当前在数据存储仓中的数据。在第二种情况下，目标数据存储仓中的数据在其被加载之后被检查。在这种情况下，CKM在写到目标之前模拟目标数据存储仓对结果流的约束。

[0125] 图12是根据本发明实施例用于在数据集成场景中提供流信息的用户界面1200的屏幕截图。

[0126] 在步骤1080中，映射被生成。图10在步骤1090结束。

[0127] 数据集成场景包和部署

[0128] 如以上所讨论的，数据集成流的自动化可以在数据集成系统中通过顺序化包中的不同步骤(映射、过程等)的执行并且通过产生包含用于这些步骤当中每一个的就绪代码的生产场景来实现。包由组织成执行图的一系列步骤组成。包是被用来生成生产场景的主要对象。场景被设计为将源组件(映射、包、过程、变量)投产。场景来自对这种组件的代码(SQL、shell等)的生成。场景可以被导出，然后导入不同的生产环境。

[0129] 图13绘出了根据本发明实施例用于创建包的方法的流程图。图13中所绘的方法1300的实现方式或其中的处理可以在软件(例如，指令或代码模块)被诸如计算机系统或信息处理设备的逻辑机器的中央处理单元(CPU或处理器)执行时由软件执行、由电子设备或专用集成电路的硬件组件执行，或者由软件元件和硬件元件的组合执行。图13中所绘出的方法1300开始于步骤1310。

[0130] 在步骤1320中，接收包步骤信息。包步骤信息包括识别步骤、元件、属性、组件等的信息。在一个例子中，用户可以与设计人员模块318的一个或多个用户接口特征交互，以创建、识别或以其它方式指定用于包的一个或多个步骤。在一种实施例中，一个或多个组件被选择并放在图上。这些组件在包中被示为步骤。

[0131] 在步骤1330中，接收包步骤顺序信息。包步骤顺序信息包括识别步骤的次序、依赖性等的信息。一旦步骤被创建，步骤就被排序或重新排序为数据处理链。在一个例子中，用户可以与设计人员模块318的一个或多个用户接口特征交互，以提供包的一个或多个步骤的顺序或排序。数据处理链可以包括被定义为第一步骤的独特步骤。一般而言，每个步骤具有一个或多个终止状态，诸如成功或失败。处于一些状态(诸如失败或成功)的步骤后面可以跟着另一个步骤或者包的结束。在一个方面中，在一些状态的情况下(诸如失败)顺序信息可以定义重试次数。在另一个方面中，包可以具有几个可能的终止步骤。

[0132] 图14是根据本发明实施例用于在数据集成场景中提供包步骤顺序信息的用户界面的屏幕截图。

[0133] 在步骤1340中，包被生成。图13在步骤1350结束。

[0134] 如以上所讨论的，数据集成流的自动化可以在数据集成系统中通过顺序化包中不同步骤(映射、过程等)的执行来实现。然后，可以为生产场景产生包含用于这些步骤当中每

一个的就绪代码的包。在各种实施例中，包被部署成在生产环境中自动运行。

[0135] 图15绘出了根据本发明实施例用于部署数据集成场景的方法1500的流程图。图15中所绘的方法1500的实现方式或其中的处理可以在软件(例如，指令或代码模块)被诸如计算机系统或信息处理设备的逻辑机器的中央处理单元(CPU或处理器)执行时由软件执行、由电子设备或专用集成电路的硬件组件执行，或者由软件元件和硬件元件的组合执行。图15中所绘出的方法1500开始于步骤1510。

[0136] 在步骤1520中，检索集成场景。在一种实施例中，包是从储存库302中检索的。在步骤1530中，集成场景被部署到一个或多个代理。在步骤1540中，集成场景由这一个或多个代理执行。在一个方面中，集成场景可以以几种途径被执行，诸如从ODI工作室312、从命令行或者从web服务执行。场景执行可以例如经由操作人员模块320等被查看和监视，如以上所讨论的。图15在步骤1550结束。

[0137] 投影器类型和选择器类型的使用

[0138] 在大多数数据集成系统中，映射需要构成映射的部分的所有输入和输出属性的明确定义。在典型的基于流的ETL工具中，在属性级做出连接器。这导致非常简明的映射模型。但是，这也产生了大量对象并且由于属性级连接器的数目而使构造和维护映射变得繁琐。

[0139] 在各种实施例中，数据集成系统200结合了用于使映射的设计和维护变得容易的一种或多种技术。组件可以被简单地添加到现有设计，而不需要指定所有输入和输出属性，并且允许组件级的连接器被重新路由(reroute)。在一个方面中，实现允许赋值表达式引用上游组件的全部或一部分的组件。因此，某些类型的组件的属性可以被传播到下游组件或以其它方式从上游组件继承，其结果是对映射设计人员的部分需要最小的努力。

[0140] 在一个方面中，数据集成系统200最小化管理属性级连接性的需求。例如，数据集成系统200可以归类映射的组件(诸如接合器、集合、表、过滤器等)。类别的一些例子是投影器和选择器。一般而言，投影器是影响流经映射的数据的形状的组件。选择器是控制数据的流但不从根本上改变流的形状的组件。在各种实施例中，选择器类型的组件被配置为透明，因为上游组件的属性是可见的。投影器类型的组件被配置为不透明，因为上游组件的属性是不可见的，只有那些在数据流的形状中可用。

[0141] 因此，需要每个属性被连接到下游组件的属性，以便能够看到来自上游的数据，在各种实施例中，数据集成系统200使用户能够直接引用任何上游属性直到并且包括最近的投影器的属性。因此，数据集成系统200大大地简化了映射的设计和维护。数据集成系统200还使得向现有的设计添加组件变得简单，通常只需要组件级连接器被重新路由(reroute)。

[0142] 图16A和16B是在根据本发明的一种实施例中的映射的简化框图。在这个例子中，图16A的映射1600包括表示数据源SRC_EMP的组件1610、表示数据源SRC_DEPT的组件1620和表示数据目标TGT_EMPDEPT的组件1630。为了更新数据目标TGT_EMPDEPT，对于数据源SRC_EMP和SRC_DEPT的需要连接。表示连接(JOIN)的组件1640被添加到映射1600，该组件连接到作为输入组件的1610和1620和作为输出的组件1630。组件1640被配置为提供连接表达式，诸如(SRC_EMP.DEPTNO＝SRC_DEPT.DEPTNO)。

[0143] 在传统的数据集成系统中，映射1600需要构成表示JOIN的组件1640的部分的所有输入和输出属性的明确定义。与此相反，在各种实施例中，映射开发人员可以定义数据目标TGT_EMPDEPT的列如何直接从流经组件1640并且因此对组件1630可见的、由组件1610表示

的数据源SRC_EMP的属性以及由组件1620表示的数据源SRC_DEPT的属性填充。例如，目标列TGT_EMPDEPT.NAME的赋值可以引用SRC_EMP.ENAME，而不需要参考组件1640的属性。

[0144] 图17绘出了根据本发明实施例用于基于组件类型派生组件属性的方法1700的流程图。图17中所绘的方法1700的实现方式或其中的处理可以在软件(例如，指令或代码模块)被诸如计算机系统或信息处理设备的逻辑机器的中央处理单元(CPU或处理器)执行时由软件执行、由电子设备或专用集成电路的硬件组件执行，或者由软件元件和硬件元件的组合执行。图17中所绘出的方法1700开始于步骤1710。

[0145] 在步骤1720中，确定组件的组件类型。如以上所讨论的，一些类型的组件影响流经映射的数据的形状，而其它类型的组件控制数据的流但不从根本上改变流的形状。在步骤1730中，基于组件类型来确定下游组件的属性。例如，如果组件控制数据的流但并不从根本上改变流的形状，则数据集成系统200可以基于上游组件的属性派生一组属性。在步骤1740中，派生的组件被暴露给下游组件。图17在步骤1750结束。因此，目标列TGT_EMPDEPT.NAME的赋值可以透明地引用SRC_EMP.ENAME，而无需参考组件1640的属性。

[0146] 数据集成系统200还使得向现有的设计添加组件变得加单，通常只需要组件级连接器被重新路由。例如，如果过滤器组件要被添加到设计中，则改变组件级连接器将不需要改变某些下游组件的属性赋值。

[0147] 图18是在根据本发明的一种实施例中具有过滤器组件1800的映射1600的简化框图。在这个例子中，映射1600包括被放在组件1610和组件1640之间的过滤器组件1800。为了将过滤器组件1800添加到映射1600中(例如，使用过滤器SRG_EMP.SAL>3000)，用户只需要将过滤器组件1800添加到组件1610和组件1640之间的图。组件级连接器将需要被重新链接，使得过滤器组件1800连接到作为输入的组件1610和作为输出的组件1640。这种改变将不需要对映射1600中任何下游赋值的改变。在传统的流工具中，通过引入新组件，列级的一切都需要重新链接。

[0148] 图19是在根据本发明的一种实施例中的映射1900的简化框图。在这个例子中，映射1900包括表示数据源的组件S1、表示数据源的组件S2以及表示数据目标的组件T。组件C1、C2和C3被配置为或者影响流经映射的数据的形状或者控制数据的流但不从根本上改变流的形状。在这个例子中，组件C1是投影器类型。因此，组件C1具有声明的一组属性(其中一些是从组件S1的一个或多个属性声明的)，如从下游组件(例如，如下所述的组件C3)看到的。组件C3具有选择器类型。因此，组件C3具有派生的一组属性(其中一些是从组件C1和C2的一个或多个属性派生的)，如从下游组件(例如，组件T)看到的。

[0149] 图20绘出了根据本发明实施例用于生成组件的方法2000的流程图。图20中所绘的方法2000的实现方式或其中的处理可以在软件(例如，指令或代码模块)被诸如计算机系统或信息处理设备的逻辑机器的中央处理单元(CPU或处理器)执行时由软件执行、由电子设备或专用集成电路的硬件组件执行，或者由软件元件和硬件元件的组合执行。图20中所绘出的方法2000开始于步骤2010。

[0150] 在步骤2020中，接收组件定义。例如，组件定义可以包括规则、操作、过程、变量、顺序，等等。在步骤2030中，接收组件类型。例如，如果组件改变流中数据的形状，则该组件可以以一种方式被归类。如果组件控制数据的流但不从根本上改变流的形状，则该组件可以以另一种方式被归类。在步骤2040中，组件被生成并可以被数据集成系统200使用。图20在

步骤2050结束。

[0151] 因此，数据集成系统200使用户能够创建独立于平台和技术的逻辑设计。用户可以创建在高级别定义用户想让数据如何在源和目标之间流动的逻辑设计。工具可以鉴于用户的基础设施来分析逻辑设计并且创建物理设计。逻辑设计可以包括对应于设计中每个源和目标的多个组件，和诸如连接或过滤的操作，以及访问点。每个组件当被转移到物理设计时生成对数据执行操作的代码。依赖于底层技术(例如，SQL服务器、Oracle、Hadoop，等等)和所使用的语言(SQL、pig，等等)，由每个组件生成的代码可以不同。

[0152] 因此，不需要数据集成系统的用户从开始到结束在逻辑设计中的每个组件处指定所有数据属性。数据集成系统提供避免需要完全声明流经逻辑设计的信息的多个组件类型，诸如投影器类型和选择器类型。数据集成系统200能够决定在由预定组件类型表示的操作处需要什么属性。这简化了设计和维护。

[0153] 结论

[0154] 图21是可以被用来实践本发明的实施例的计算机系统2100的简化框图。如图21中所示，计算机系统2100包括经由总线子系统2120与多个外围设备通信的处理器2110。这些外围设备可以包括包含存储器子系统2140和文件存储子系统2150的存储设备子系统2130、输入设备2160、输出设备2170，以及网络接口子系统2180。

[0155] 总线子系统2120提供让计算机系统2100的各种组件和子系统彼此如预期地通信的机制。虽然总线子系统2120示意性地示为单条总线，但总线子系统的备选实施例可以利用多条总线。

[0156] 存储设备子系统2130可以被配置为存储提供本发明的功能的基本编程和数据结构。提供本发明的功能的软件(代码模块或指令)可以存储在存储设备子系统2130中。这些软件模块或指令可以由(一个或多个)处理器2110执行。存储设备子系统中2130还可以提供用于存储根据本发明被使用的数据的储存库。存储设备子系统2130可以包括存储器子系统2140和文件/盘存储子系统2150。

[0157] 存储器子系统2140可以包括多个存储器，包括用于在程序执行期间存储指令和数据的主随机存取存储器(RAM)2142，以及其中存储固定指令的只读存储器(ROM)2144。文件存储子系统2150为程序和数据文件提供持久性(非易失性)存储，并且可以包括硬盘驱动器，连同关联的可移动介质的软盘驱动器、光盘只读存储器(CD-ROM)驱动器、DVD、光盘驱动器、可移动介质盒，以及其它类似的存储介质。

[0158] 输入设备2160可以包括键盘、诸如鼠标、轨迹球、触摸板或图形输入板的定点设备、扫描仪、条形码扫描仪、结合到显示器中的触摸屏、诸如语音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。一般而言，术语“输入设备”的使用意在包括用于将信息输入计算机系统2100的所有可能类型的设备和机制。

[0159] 输出设备2170可以包括显示器子系统、打印机、传真机，或者诸如音频输出设备的非视觉显示器，等等。显示子系统可以是阴极射线管(CRT)、诸如液晶显示器(LCD)的平板设备，或投影设备。一般而言，术语“输出设备”的使用意在包括从计算机系统2100输出信息的所有可能类型的设备和机制。

[0160] 网络接口子系统2180提供到其它计算机系统、设备和诸如通信网络2190的网络的接口。网络接口子系统2180充当从计算机系统2100接收数据和从计算机系统向其它系统发

送数据的接口。通信网络2190的一些例子是专用网络、公共网络、租用线路、互联网、以太网、令牌环网、光纤网络等。

[0161] 计算机系统2100可以是各种类型，包括个人计算机、便携式计算机、工作站、网络计算机、大型机、信息站或任何其它数据处理系统。由于计算机和网络的不断变化的本质，在图21中所绘出的计算机系统2100的描述仅仅是作为用于说明计算机系统的优选实施例的具体例子。具有比图21中所绘的系统更多或更少组件的许多其它配置是可能的。

[0162] 图22是根据实施例用于促进数据映射的生成的系统2200的简化框图。图22中的系统2200中所包括的单元可以在可被逻辑机器(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)执行的软件(例如，指令或代码模块)中实现，在电子设备或专用集成电路的硬件组件中实现，或者在软件元件和硬件元件的组合中实现。

[0163] 如图22中所示，除其它的之外，数据集成系统2200可以包括接收单元2210、确定单元2220和生成单元2230。接收单元2210可以被配置为接收指定逻辑设计的一个或多个组件的信息，其中该一个或多个组件当中至少一个组件具有第一类型。确定单元2220可以被配置为基于逻辑设计中的上游组件，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合。生成单元2230可以被配置为生成指示所述一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的信息。

[0164] 根据实施例，确定一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的数据属性集合可以包括派生从上游组件可见的一个或多个属性并且将该一个或多个属性暴露给下游组件。

[0165] 根据实施例，指定逻辑设计的一个或多个组件的信息可以包括指示改变流经该逻辑设计的信息的形状的操作的信息。

[0166] 根据实施例，指定逻辑设计的一个或多个组件的信息可以包括指示控制流经该逻辑设计的信息流但是不改变流经该逻辑设计的信息的形状的操作的信息。

[0167] 根据实施例，指定逻辑设计的一个或多个组件的信息可以包括指示具有存储在源数据存储仓中的数据的一个或多个属性的源组件的信息。

[0168] 根据实施例，指定逻辑设计的一个或多个组件的信息可以包括指示具有要存储在目标数据存储仓中的数据的一个或多个属性的目标组件的信息。

[0169] 根据实施例，生成指示一个或多个组件当中具有第一类型的至少一个组件的、对逻辑设计中的下游组件可见的属性集合的信息可以包括将属性列表导出到下游组件。

[0170] 根据实施例，接收单元2210还可以被配置为通过将组件或属性引入到逻辑设计中或从逻辑设计中去除组件或属性来接收逻辑设计中的变化。确定单元2220还可以被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件。基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，确定单元2220还可以被配置为确定下游组件可见的已更新的数据属性集合。

[0171] 根据实施例，数据集成系统2200还可以包括保留单元2240。接收单元2210还可以被配置为通过将组件或属性引入到逻辑设计中来接收逻辑设计中的变化。确定单元2220还可以被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件。基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个

组件，保留单元2240还可以被配置为保留对下游组件可见的数据属性集合。

[0172] 根据实施例，数据集成系统2200还可以包括重命名单元2250。接收单元2210还可以被配置为接收逻辑设计中重命名组件或属性的变化。确定单元2220还可以被配置为确定逻辑设计中的变化是否影响一个或多个组件当中具有第一类型的至少一个组件。基于确定逻辑设计中的变化影响一个或多个组件当中具有第一类型的至少一个组件，重命名单元2250可以被配置为重命名对下游组件可见的数据属性集合。

[0173] 虽然已经描述了本发明的具体实施例，但是各种修改、变更、替换构造和等同物也包括在本发明的范围之内。所描述的发明并不限于某些特殊数据处理环境中的操作，而是可以自由地在多种数据处理环境中操作。此外，虽然已经利用特定的一系列事务和步骤描述了本发明，但是对本领域技术人员应当显然，本发明的范围不限于所描述的事务和步骤序列。

[0174] 另外，虽然已经利用硬件和软件的特定组合描述了本发明，但是应当认识到，硬件和软件的其它组合也在本发明的范围之内。本发明可以仅在硬件中，或仅在软件中，或使用它们的组合来实现。

[0175] 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。但是，显而易见的是，在不背离如权利要求中阐述的本发明的更宽的精神和范围的情况下，可以对其进行添加、减少、删除以及其它修改和变化。

[0176] 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何一个的各种实施例可以在软件、固件、硬件或者其组合中以逻辑的形式实现。逻辑可以作为适于指示逻辑机器的中央处理单元(CPU或处理器)执行一组步骤的一组指令存储在机器可存取存储器、机器可读制品、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之上，其中这组步骤可以在本公开内容中给出的发明的各种实施例中公开。当代码模块利用计算机系统或信息处理设备的处理器变得可操作时，逻辑可以构成软件程序或计算机程序产品的一部分，当逻辑被执行时，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、硬件或者其组合中实现所给出的一个或多个发明的各种实施例的所公开操作或功能当中任何一个的其它方式、变化、修改、备选方案和/或方法。

[0177] 其示教可以在本公开内容中给出的那些发明当中任何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照示例性说明和具体附图描述，因此所描述的方法和/或具体结构的各种修改和适配会对本领域技术人员变得显然。依赖于本公开内容和在本文发现的这些示教并且所述示教通过其可以推动本领域的所有这种修改、适配或变化都应当被认为在其示教可以在本公开内容中存在的一个或多个发明的范围之内。由此，本描述和附图不应当在限制性的意义上考虑，因为应当理解，公开内容中给出的发明不是要以任何方式限定到具体说明的那些实施例。

[0178] 因此，以上描述和任何附图、说明和图示意在说明而不是限制。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中所示的那些实施例来确定，而是应当参考未决的权利要求连同它们的完全范围或等同物来确定。

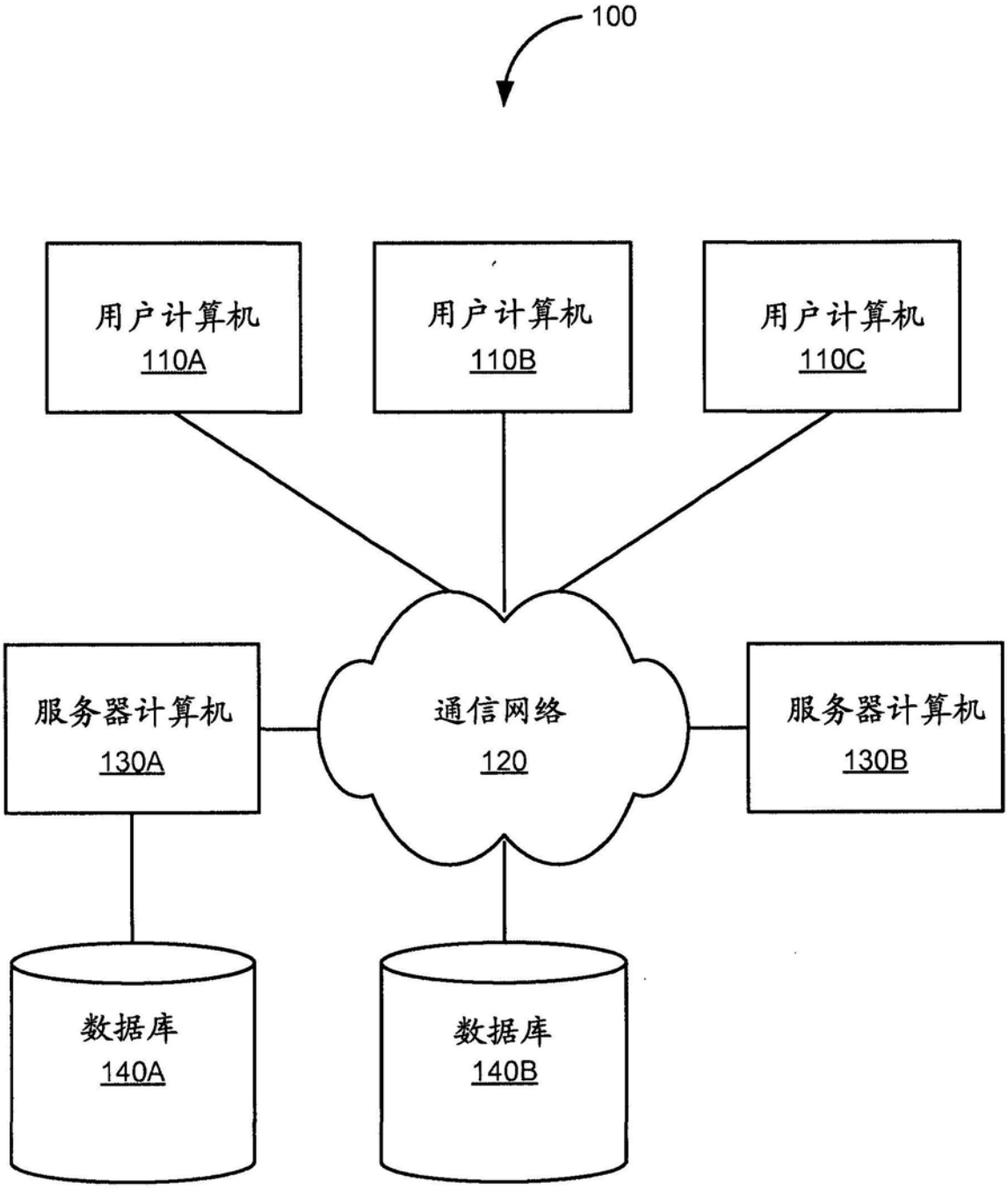

100
用户计算机
110A
用户计算机
110B
用户计算机
110C
服务器计算机
130A
通信网络
120
服务器计算机
130B
数据库
140A
数据库
140B

图1

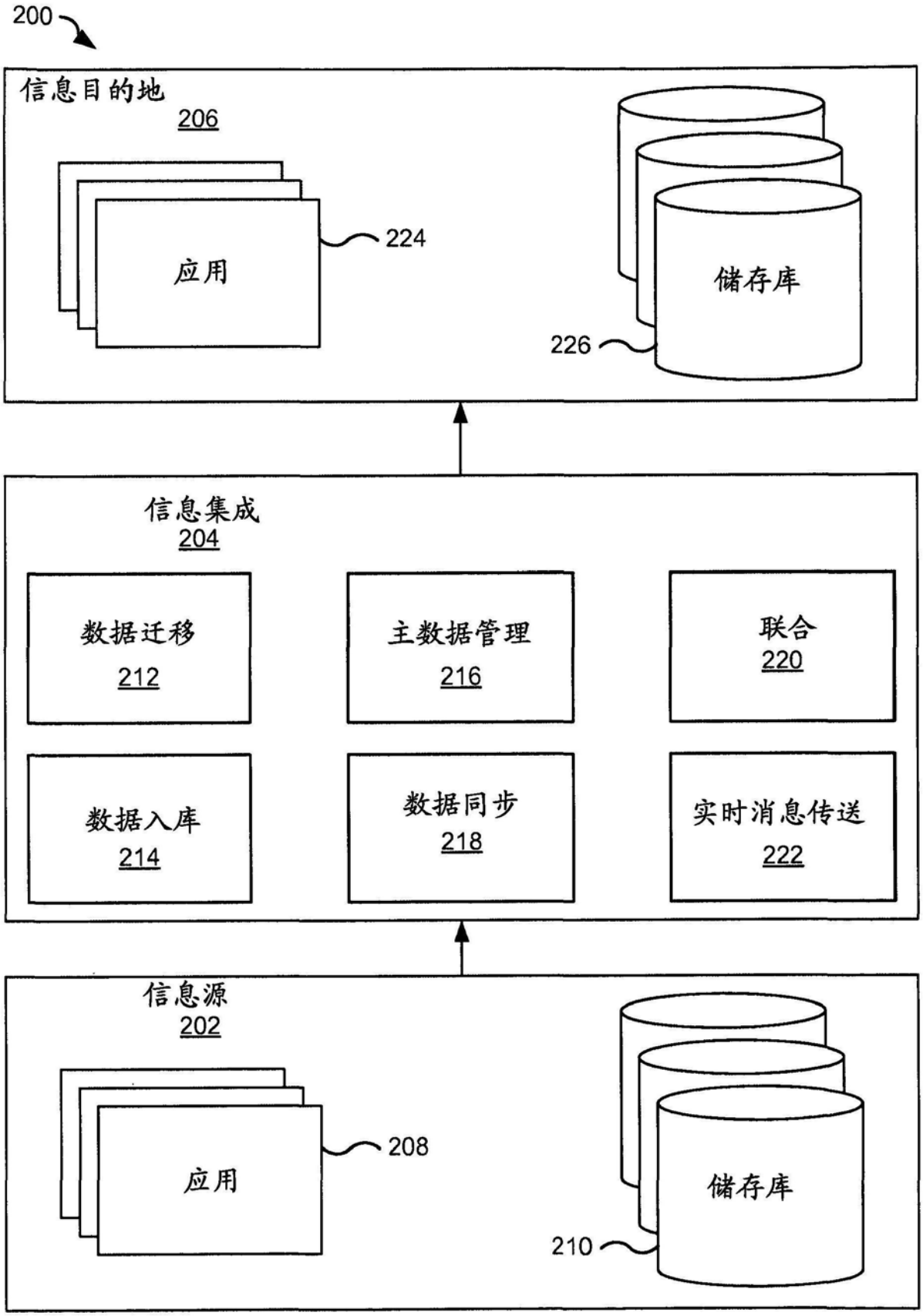

200
信息目的地
206
应用
224
储存库
226
信息集成
204
数据迁移
212
主数据管理
216
联合
220
数据入库
214
数据同步
218
实时消息传送
222
信息源
202
应用
208
储存库
210

图2

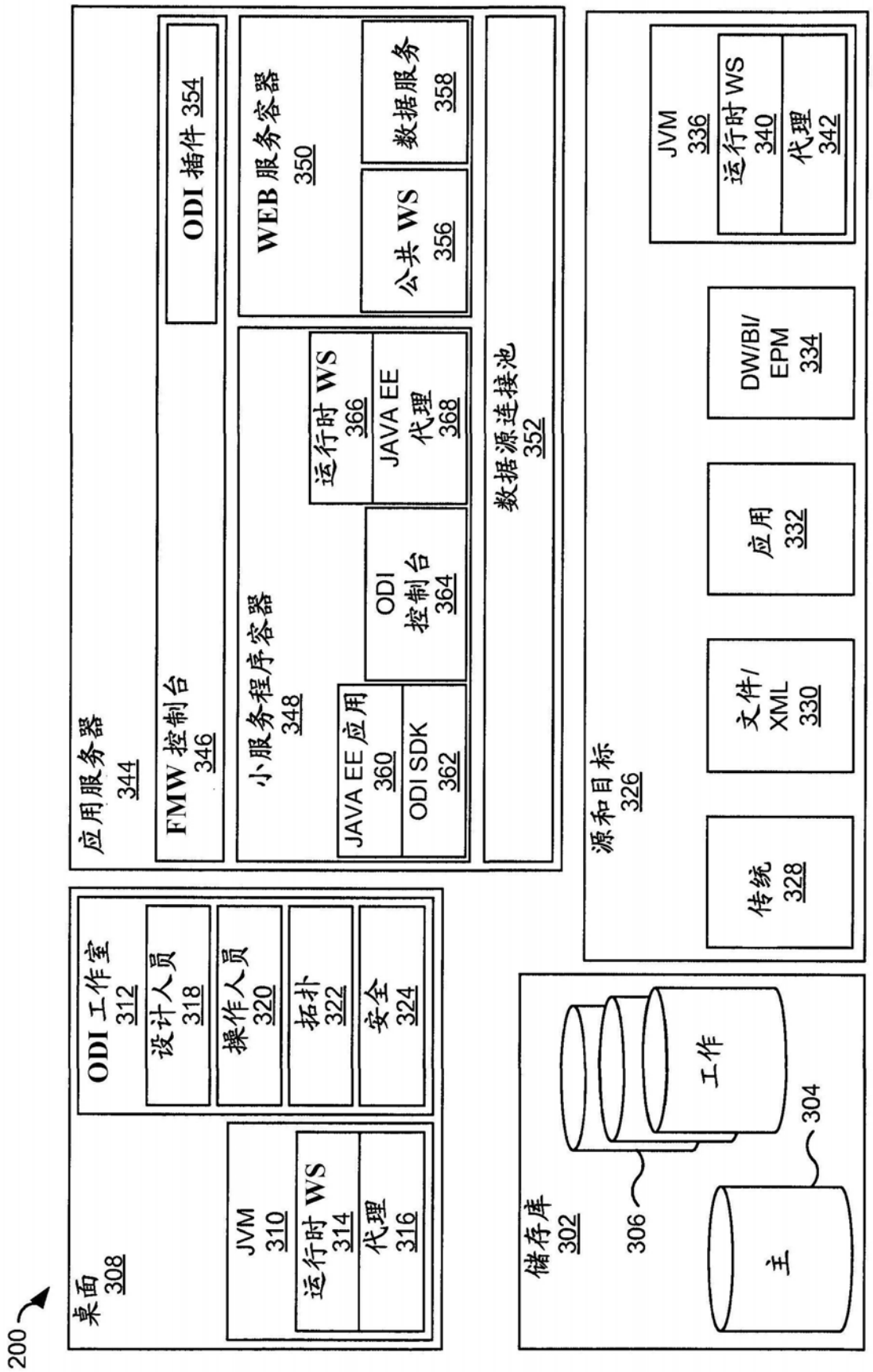

200
桌面
308
JVM
310
运行时 WS
314
代理
316
ODI 工作室
312
设计人员
318
操作人员
320
拓扑
322
安全
324
应用服务器
344
FMW 控制台
346
ODI 插件 354
小服务程序容器
348
JAVA EE 应用
360
ODI SDK
362
ODI
控制台
364
运行时 WS
366
JAVA EE
代理
368
WEB 服务容器
350
公共 WS
356
数据服务
358
数据源连接池
352
储存库
302
306
主
工作
304
源和目标
326
传统
328
文件/
XML
330
应用
332
DW/BI/
EPM
334
JVM
336
运行时 WS
340
代理
342

图3

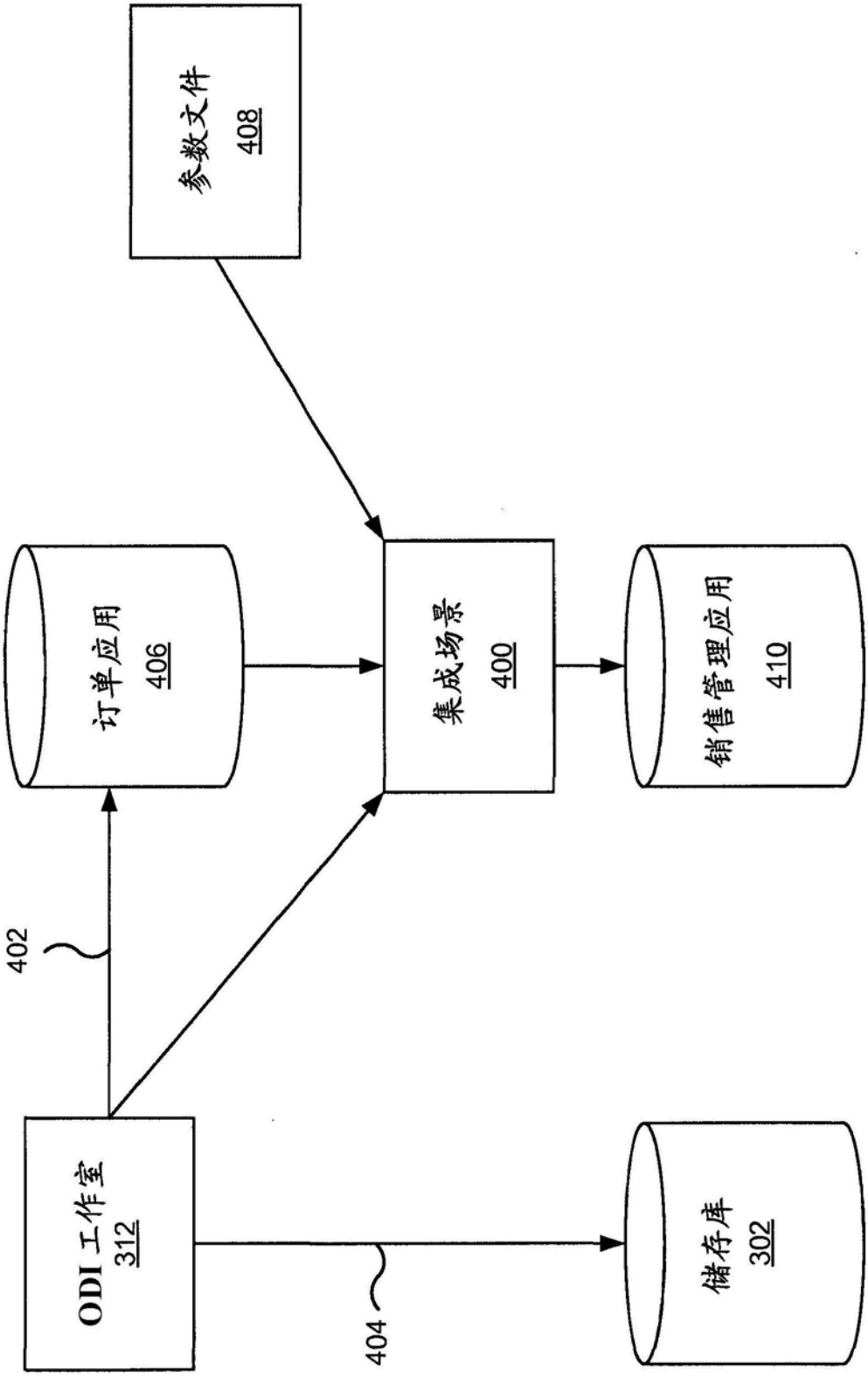

参数文件
408
订单应用
406
集成场景
400
销售管理应用
410
402
ODI 工作室
312
404
储存库
302

图4

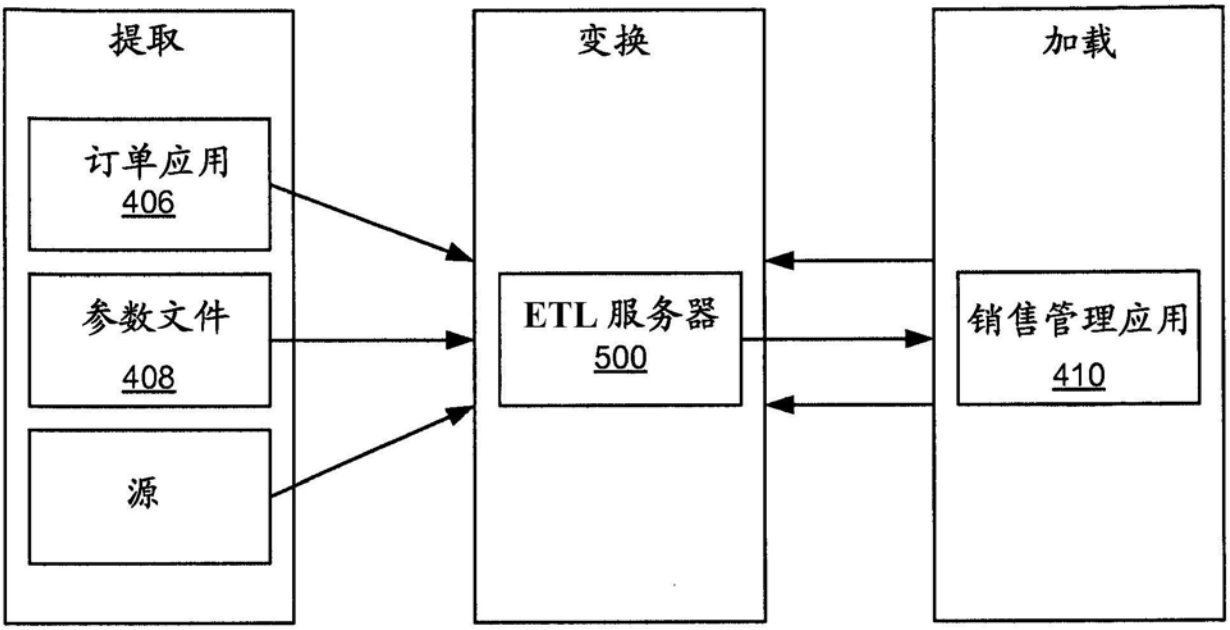

提取
订单应用
406
参数文件
408
源
变换
ETL 服务器
500
加载
销售管理应用
410

图5A

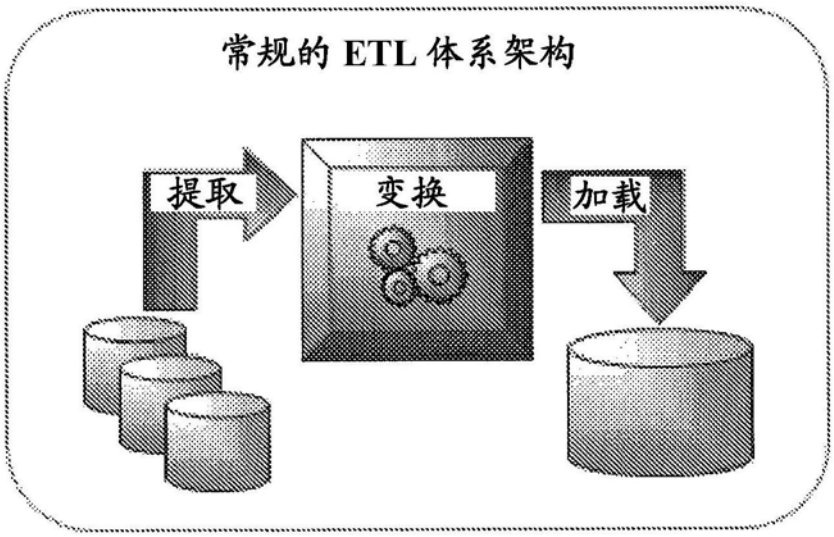

常规的 ETL 体系架构
提取
变换
加载

图5B

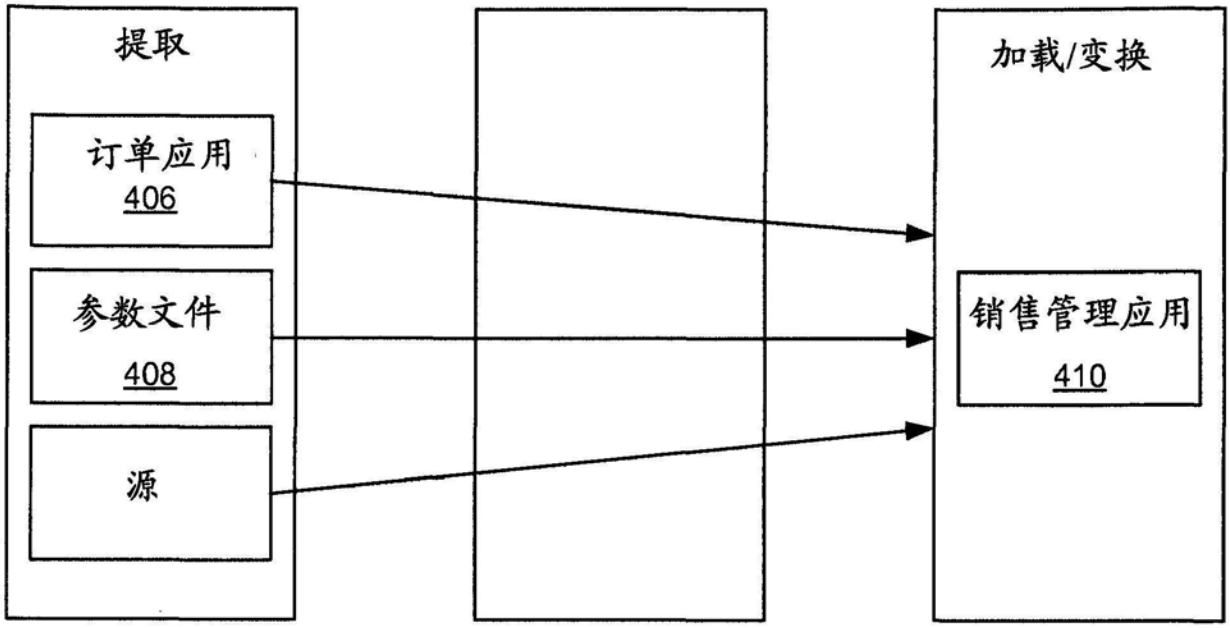


图6A

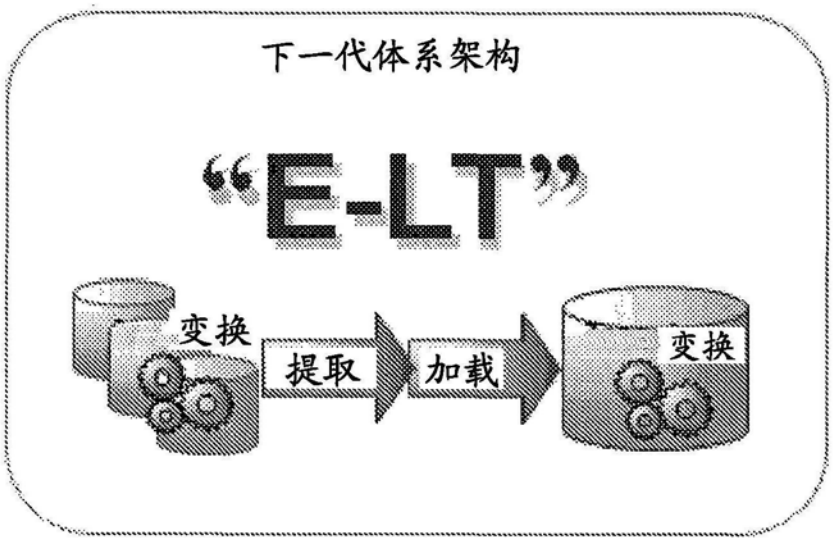


图6B

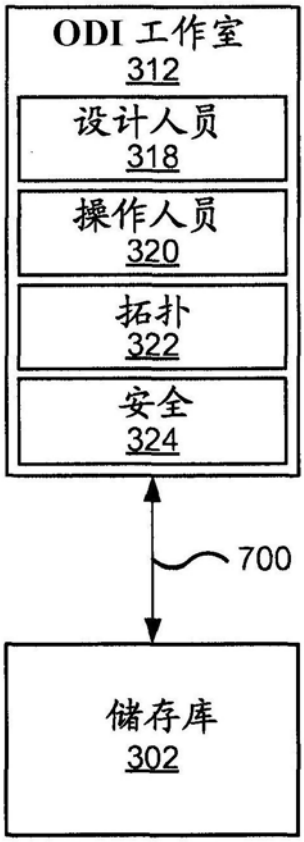


图7

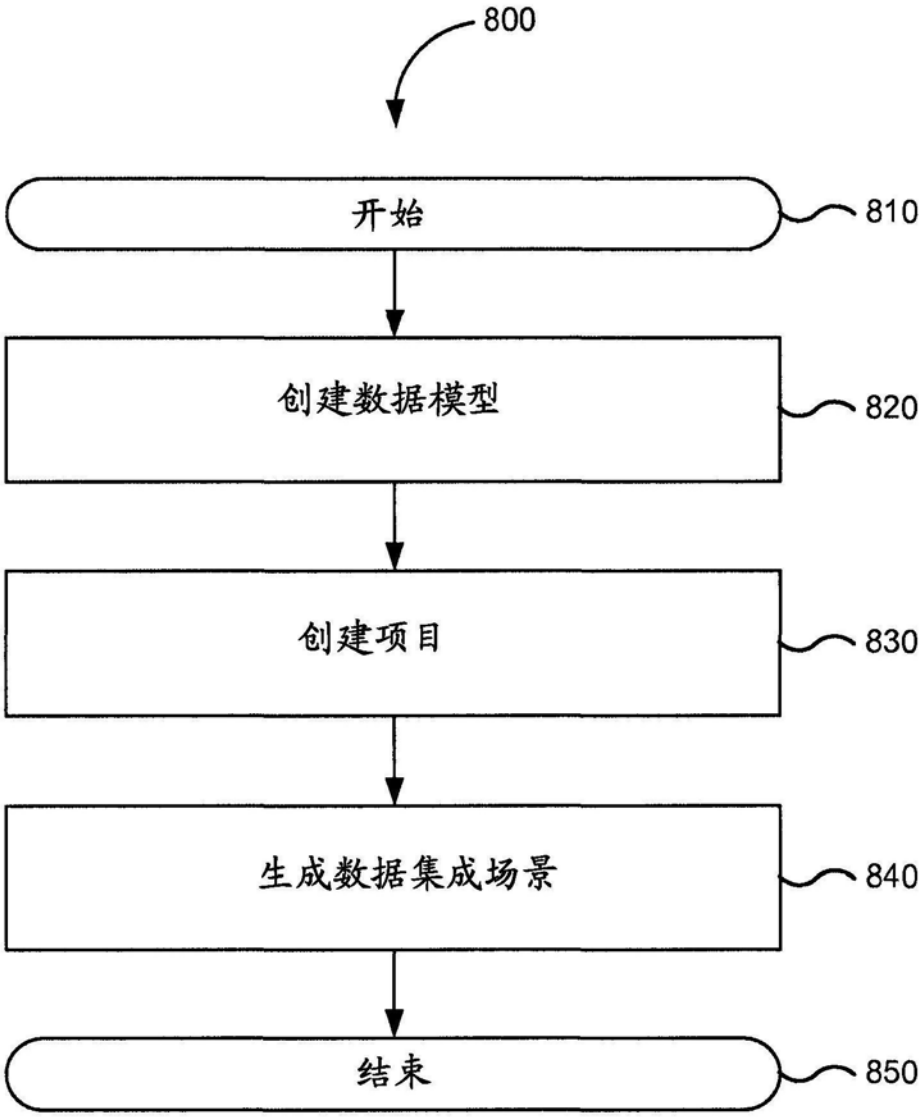


图8

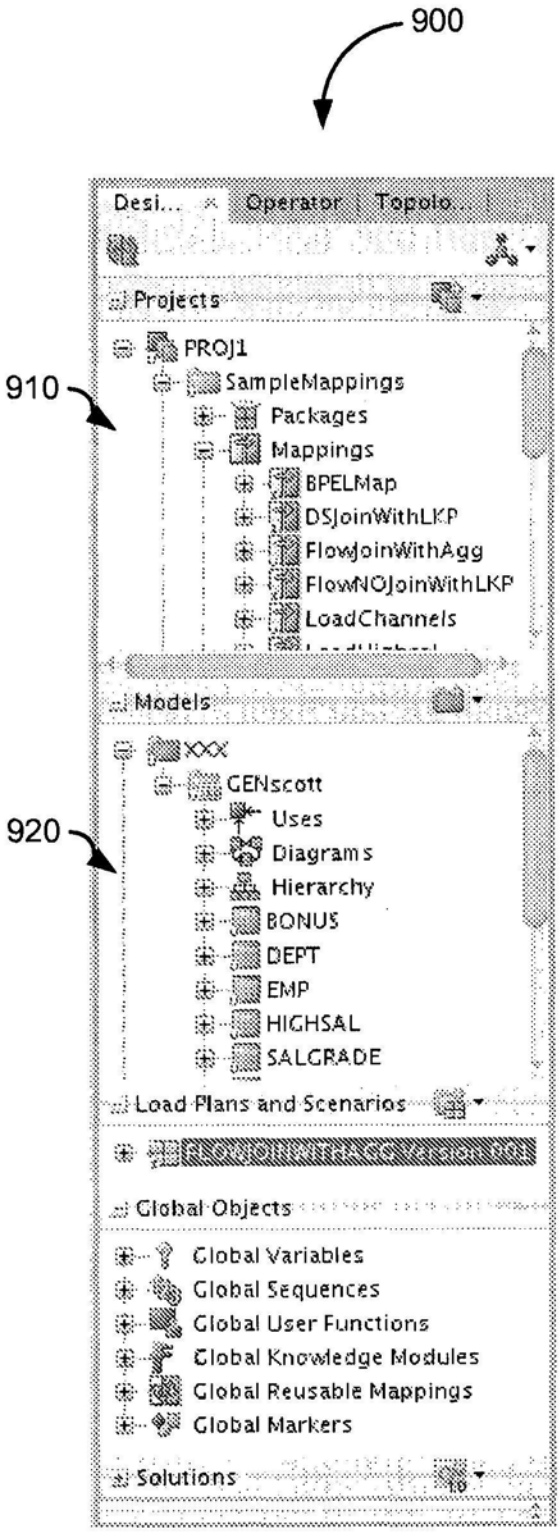

900
910
920
Desi...
Operator
Topolo...
Projects
PROJ1
SampleMappings
Packages
Mappings
BPELMap
DSJoinWithLKP
FlowJoinWithAgg
FlowNOJoinWithLKP
LoadChannels
Models
xxx
GENscott
Uses
Diagrams
Hierarchy
BONUS
DEPT
EMP
HIGHSAL
SALGRADE
Load Plans and Scenarios
Global Objects
Global Variables
Global Sequences
Global User Functions
Global Knowledge Modules
Global Reusable Mappings
Global Markers
Solutions

图9

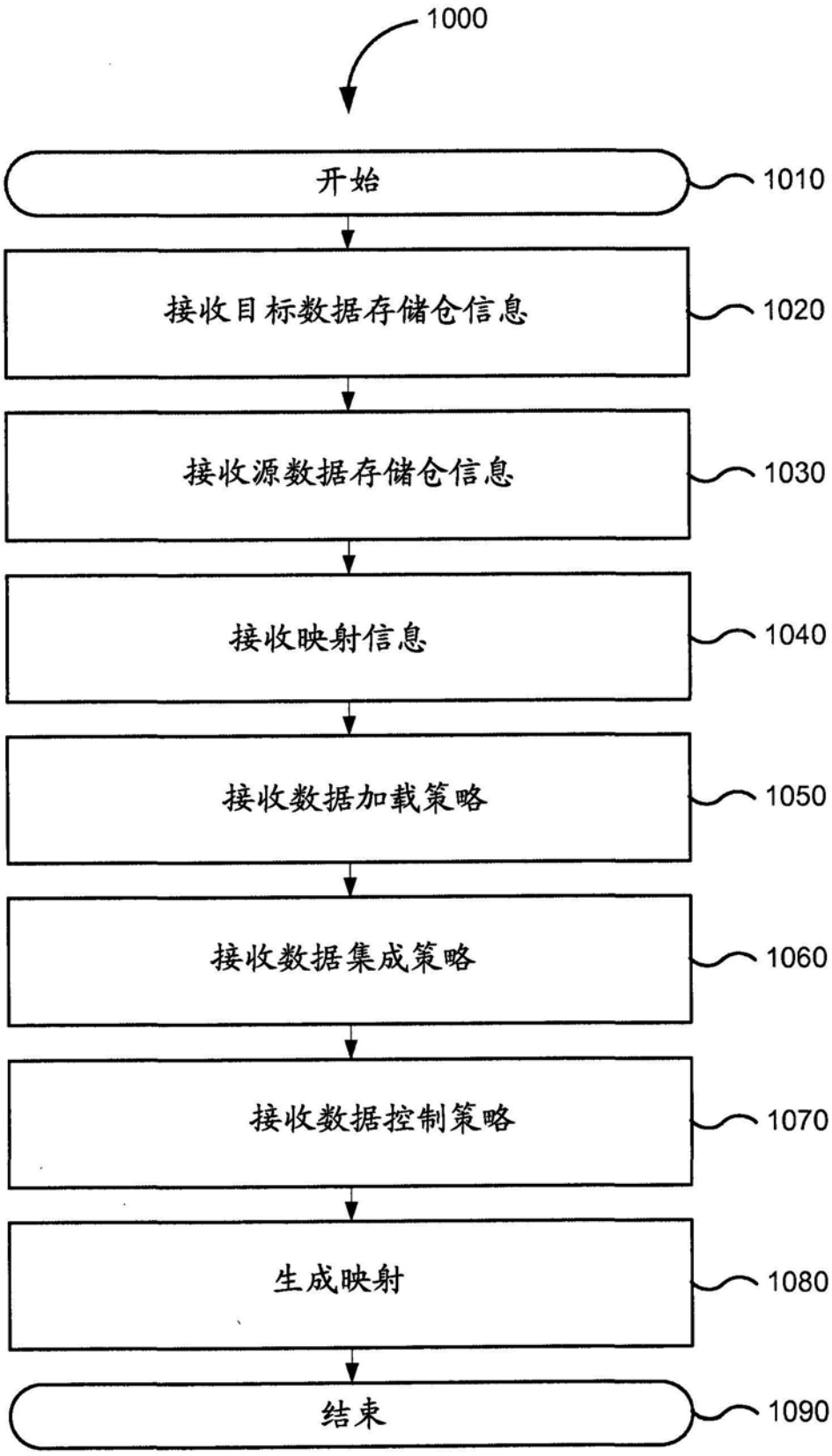

1000
开始
1010
接收目标数据存储仓信息
1020
接收源数据存储仓信息
1030
接收映射信息
1040
接收数据加载策略
1050
接收数据集成策略
1060
接收数据控制策略
1070
生成映射
1080
结束
1090

图10

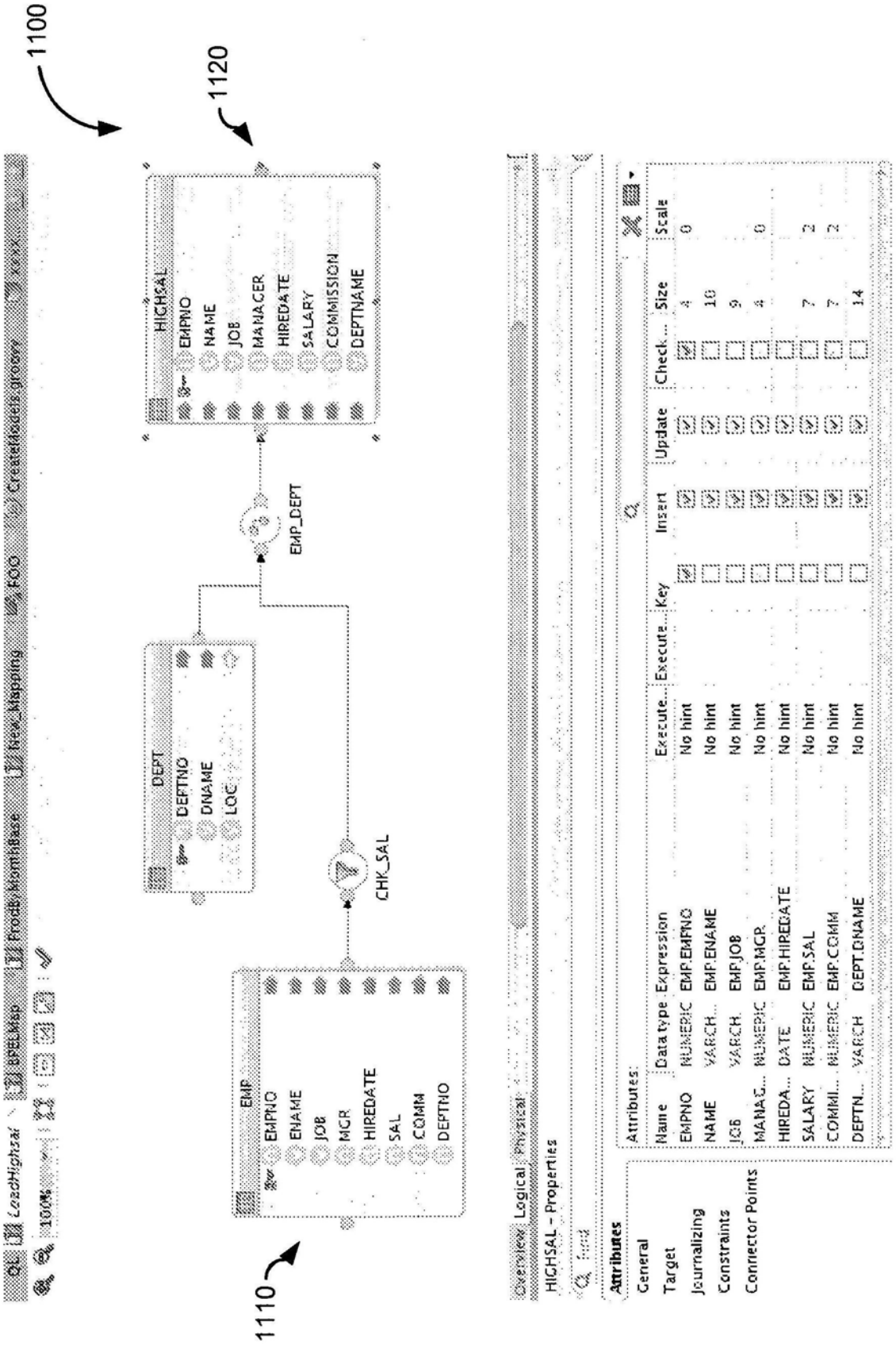

1100
1120
1110
HIGHSAL
EMPNO
NAME
JOB
MANAGER
HIREDATE
SALARY
COMMISSION
DEPTNAME
DEPT
DEPTNO
DNAME
LOC
EMP
EMPNO
ENAME
JOB
MGR
HIREDATE
SAL
COMM
DEPTNO
EMP_DEPT
CHK_SAL
LoadHighsal
BPELMap
FromDbToHsmlBase
New_Mapping
FOO
CreateModels.groovy
Overview
Logical
Physical
HIGHSAL - Properties
Attributes
General
Target
Journalizing
Constraints
Connector Points
Attributes:
Name
Data type
Expression
Execute...
Execute...
Key
Insert
Update
Check...
Size
Scale
EMPNO
NUMERIC
EMP.EMPNO
No hint
NAME
VARCH...
EMP.ENAME
No hint
JOB
VARCH...
EMP.JOB
No hint
MANAG...
NUMERIC
EMP.MGR
No hint
HIREDA...
DATE
EMP.HIREDATE
No hint
SALARY
NUMERIC
EMP.SAL
No hint
COMMI...
NUMERIC
EMP.COMM
No hint
DEPTN...
VARCH
DEPT.DNAME
No hint

图11

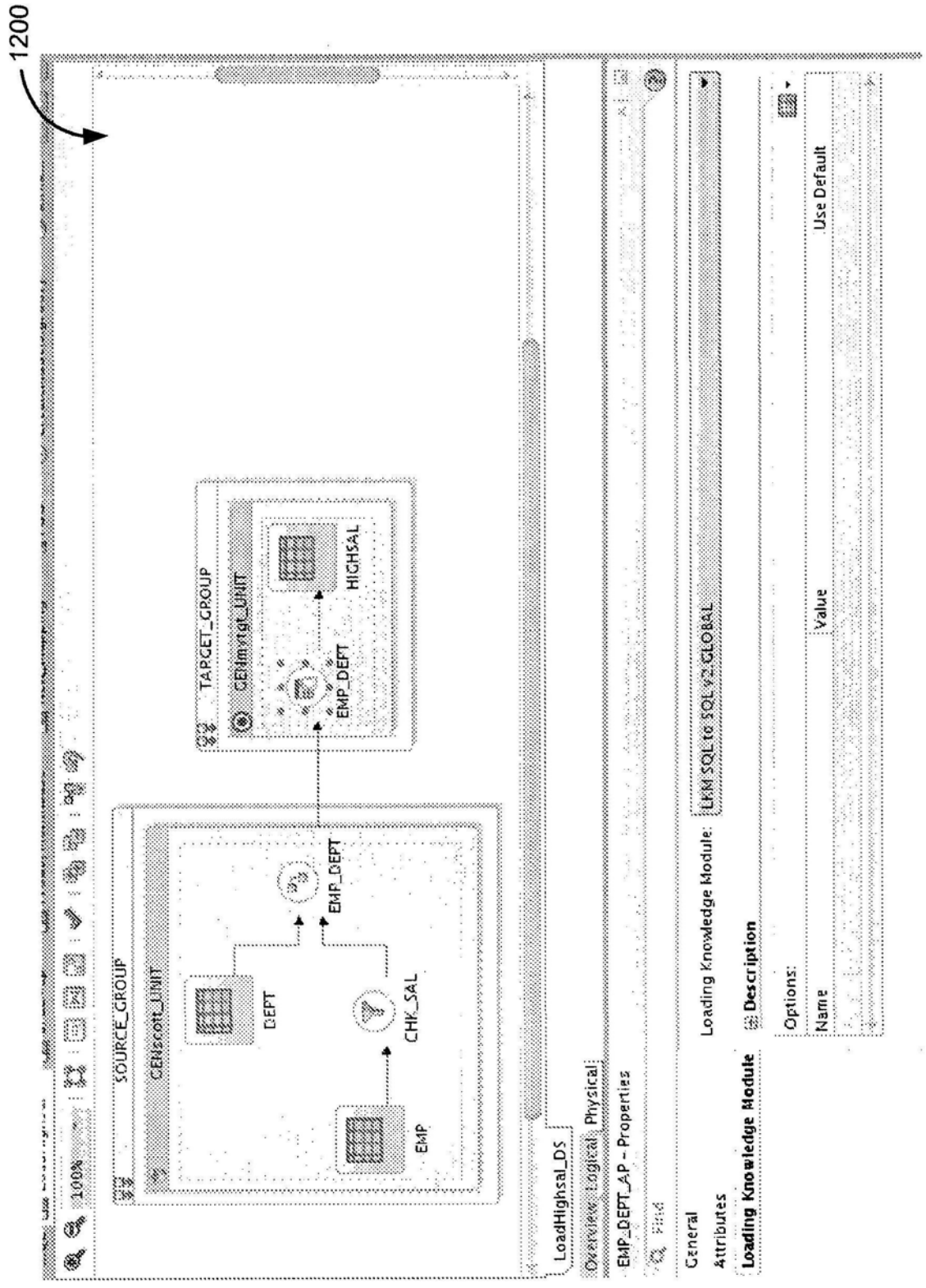

1200
SOURCE_GROUP
CEMscott_UNIT
DEPT
EMP_DEPT
CHK_SAL
EMP
TARGET_GROUP
CEImtgt_UNIT
HIGHSAL
EMP_DEPT
LoadHighsal_DS
Overview
Logical
Physical
EMP_DEPT_AP - Properties
General
Attributes
Loading Knowledge Module
Loading Knowledge Module:
LKM SQL to SQL v2 GLOBAL
Description
Options:
Name
Value
Use Default

图12

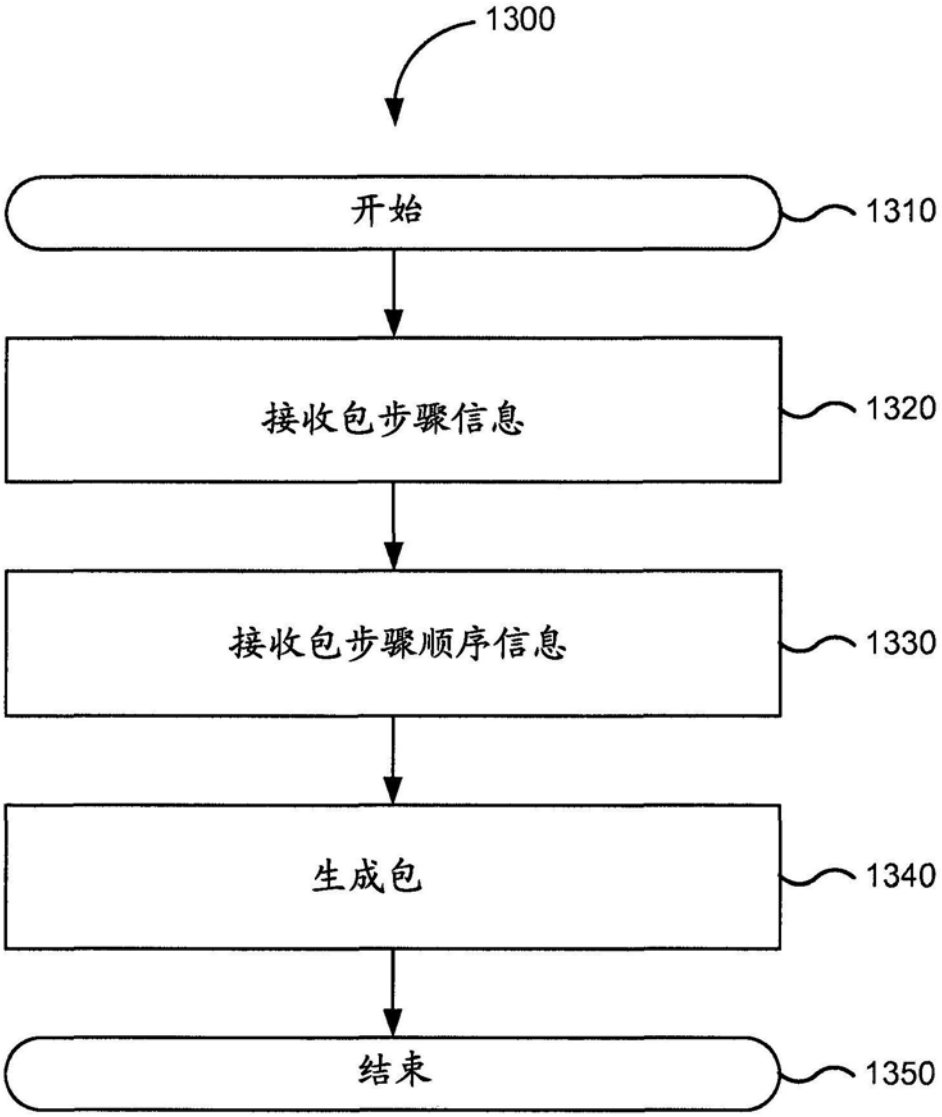


图13

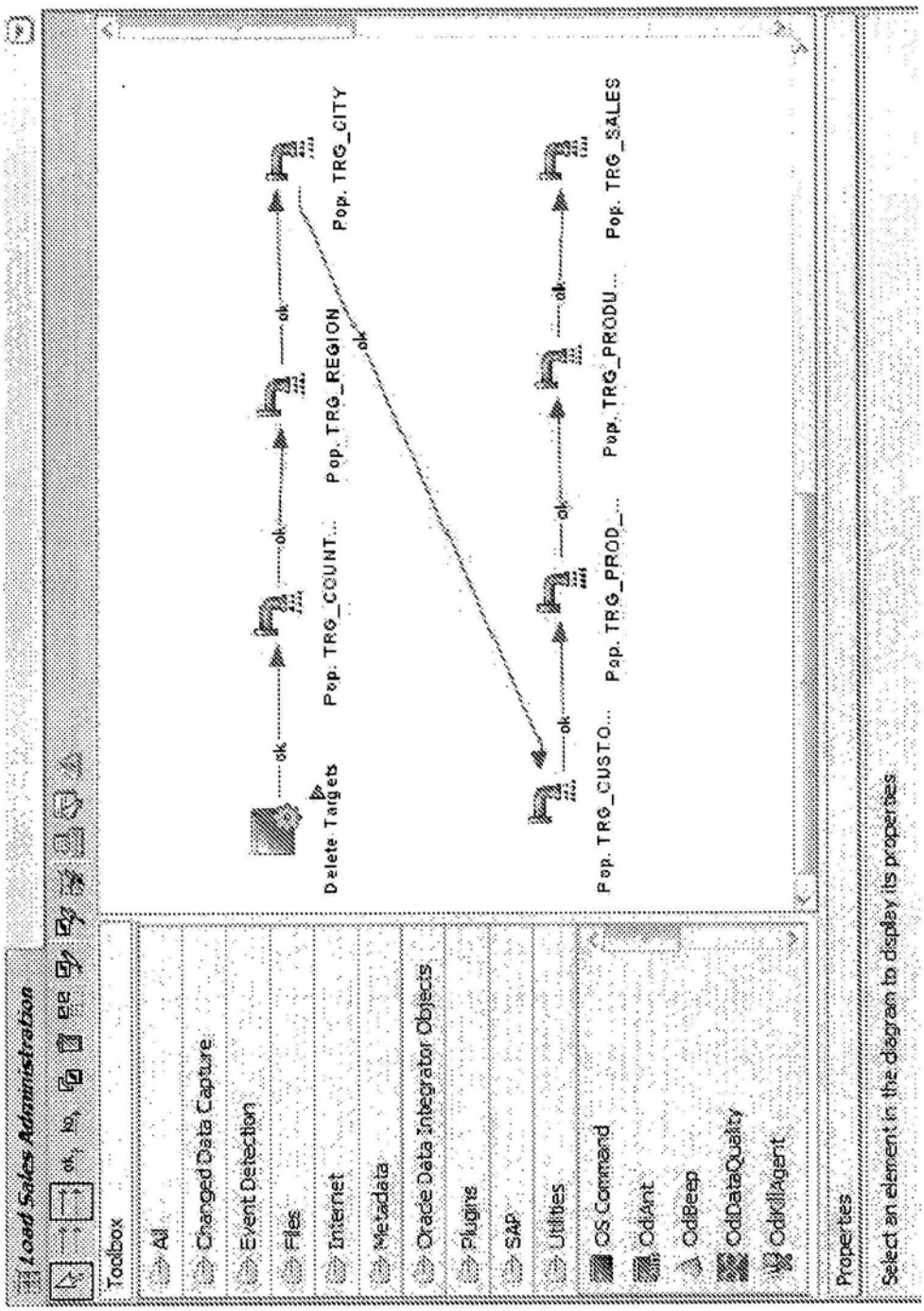

Load Sales Administration
Toolbox
All
Changed Data Capture
Event Detection
Files
Internet
Metadata
Oracle Data Integrator Objects
Plugins
SAP
Utilities
OS Command
OdiAnt
OdiBeep
OdiDataQuality
OdiKillAgent
Properties
Select an element in the diagram to display its properties
Delete Targets
Pop. TRG_COUNT...
Pop. TRG_REGION
Pop. TRG_CITY
Pop. TRG_CUSTO...
Pop. TRG_PROD_...
Pop. TRG_PRODU...
Pop. TRG_SALES
ok

图14

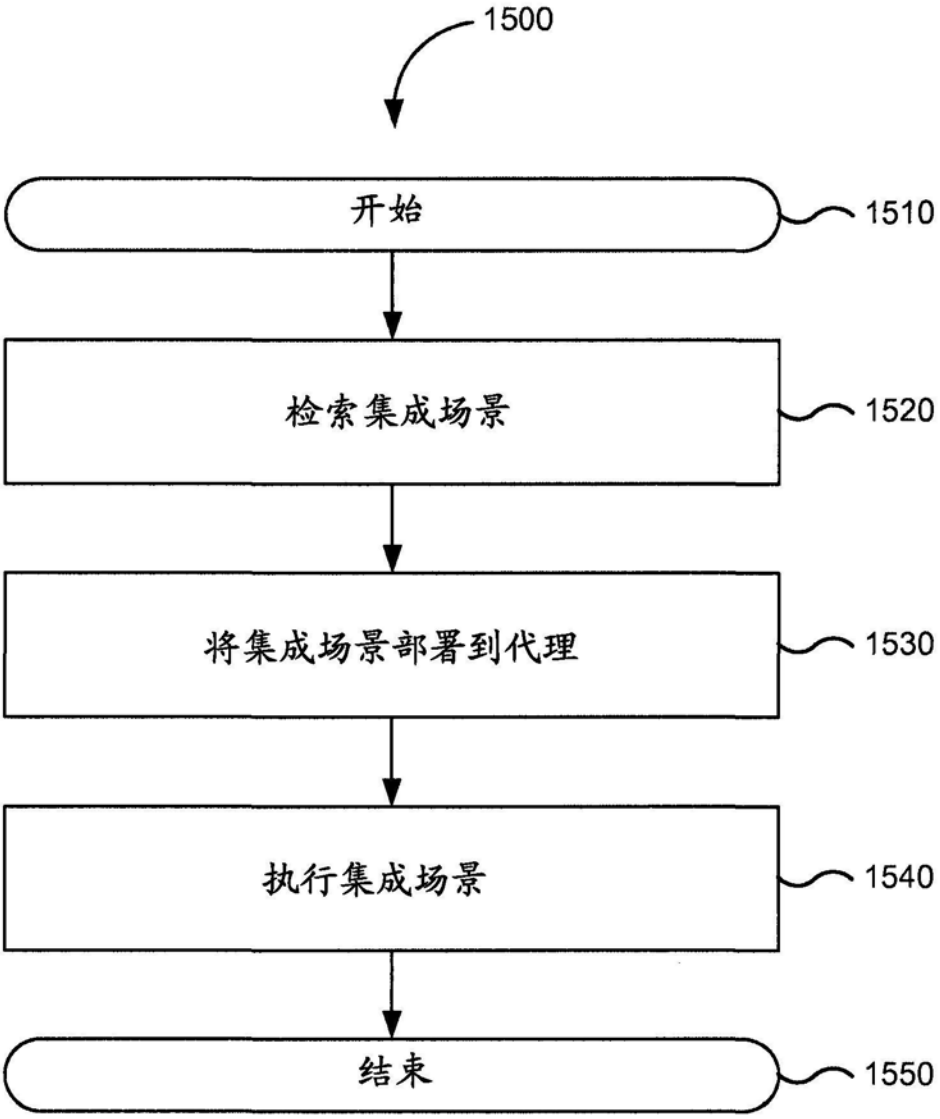

1500
开始
1510
检索集成场景
1520
将集成场景部署到代理
1530
执行集成场景
1540
结束
1550

图15

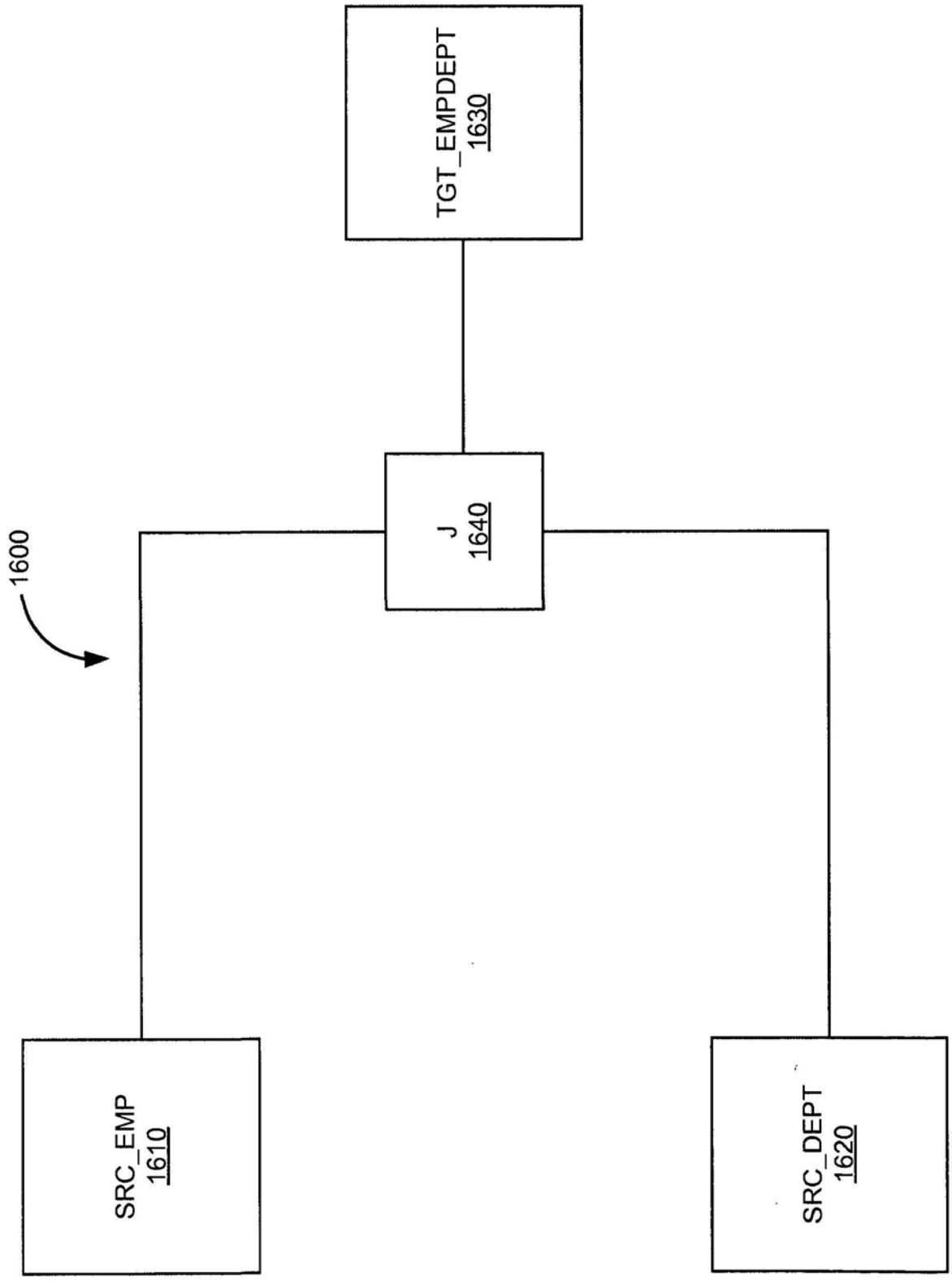


图16A

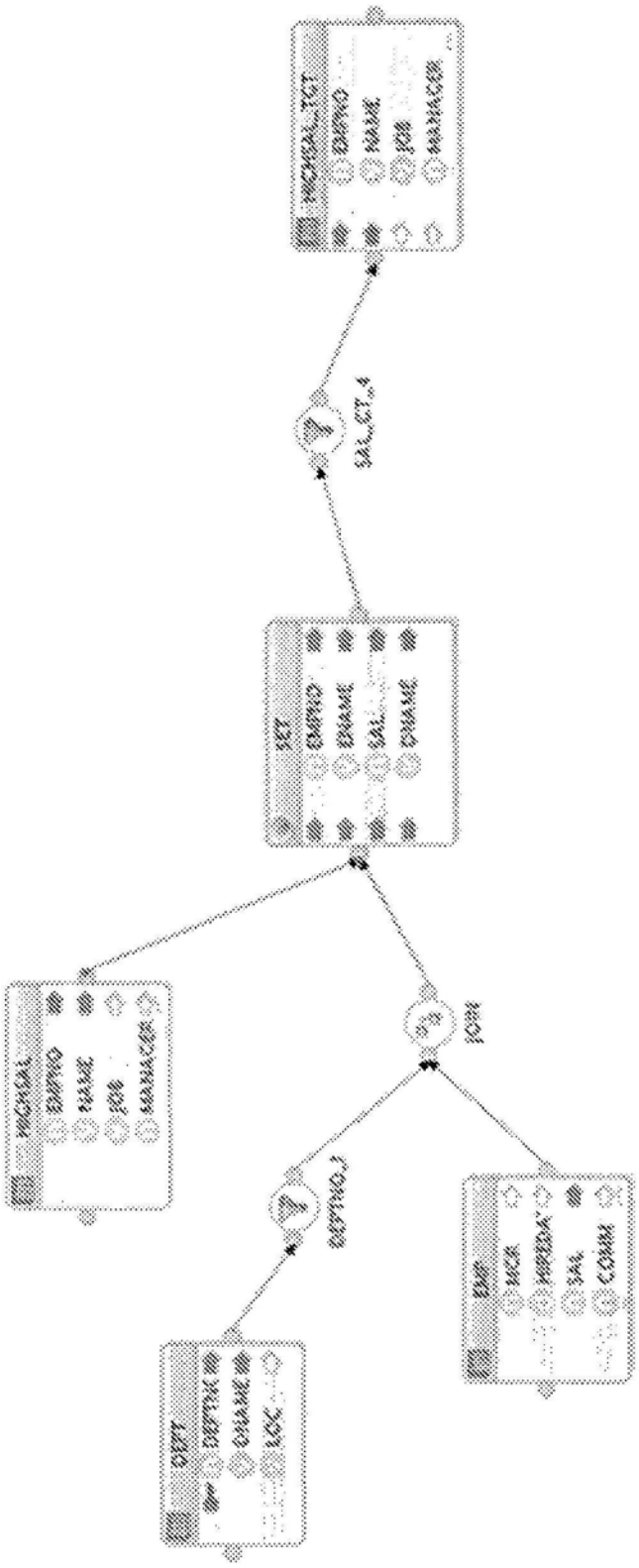


图16B

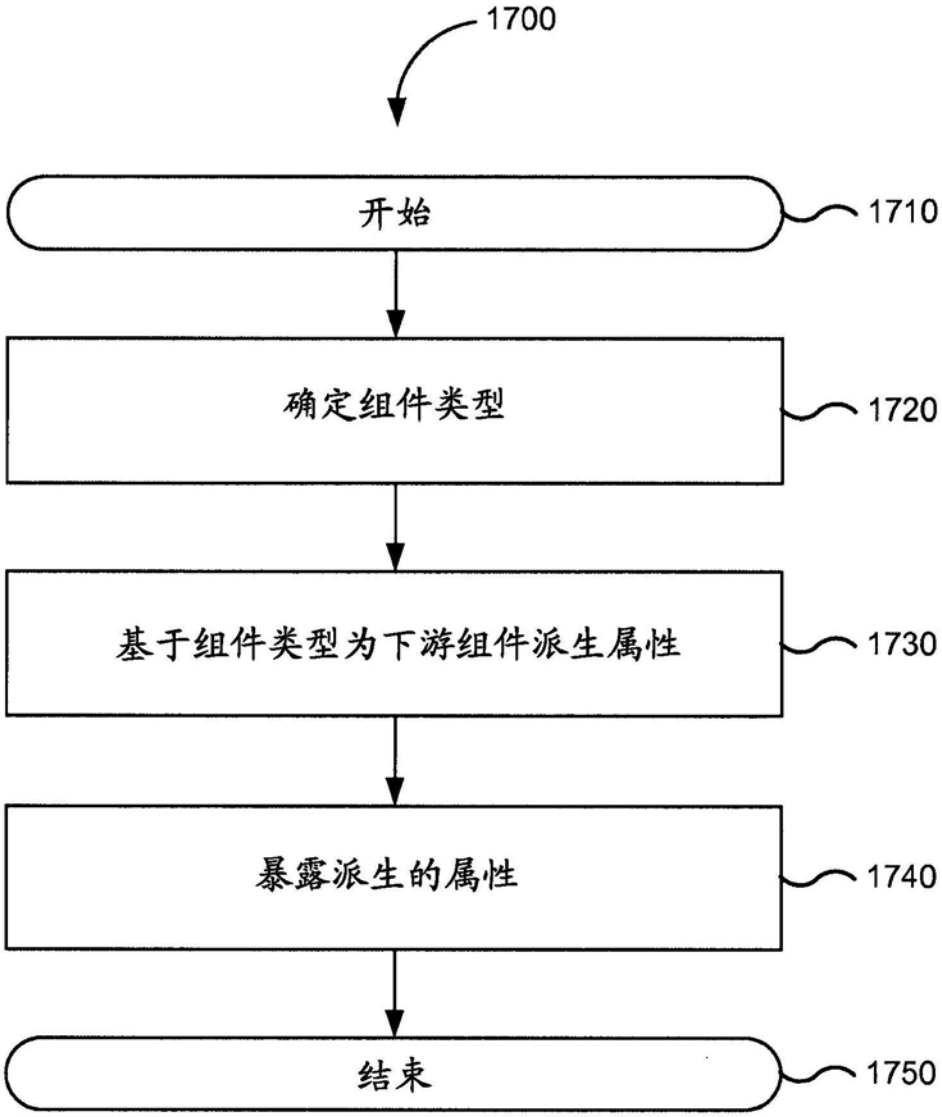

1700
开始
1710
确定组件类型
1720
基于组件类型为下游组件派生属性
1730
暴露派生的属性
1740
结束
1750

图17

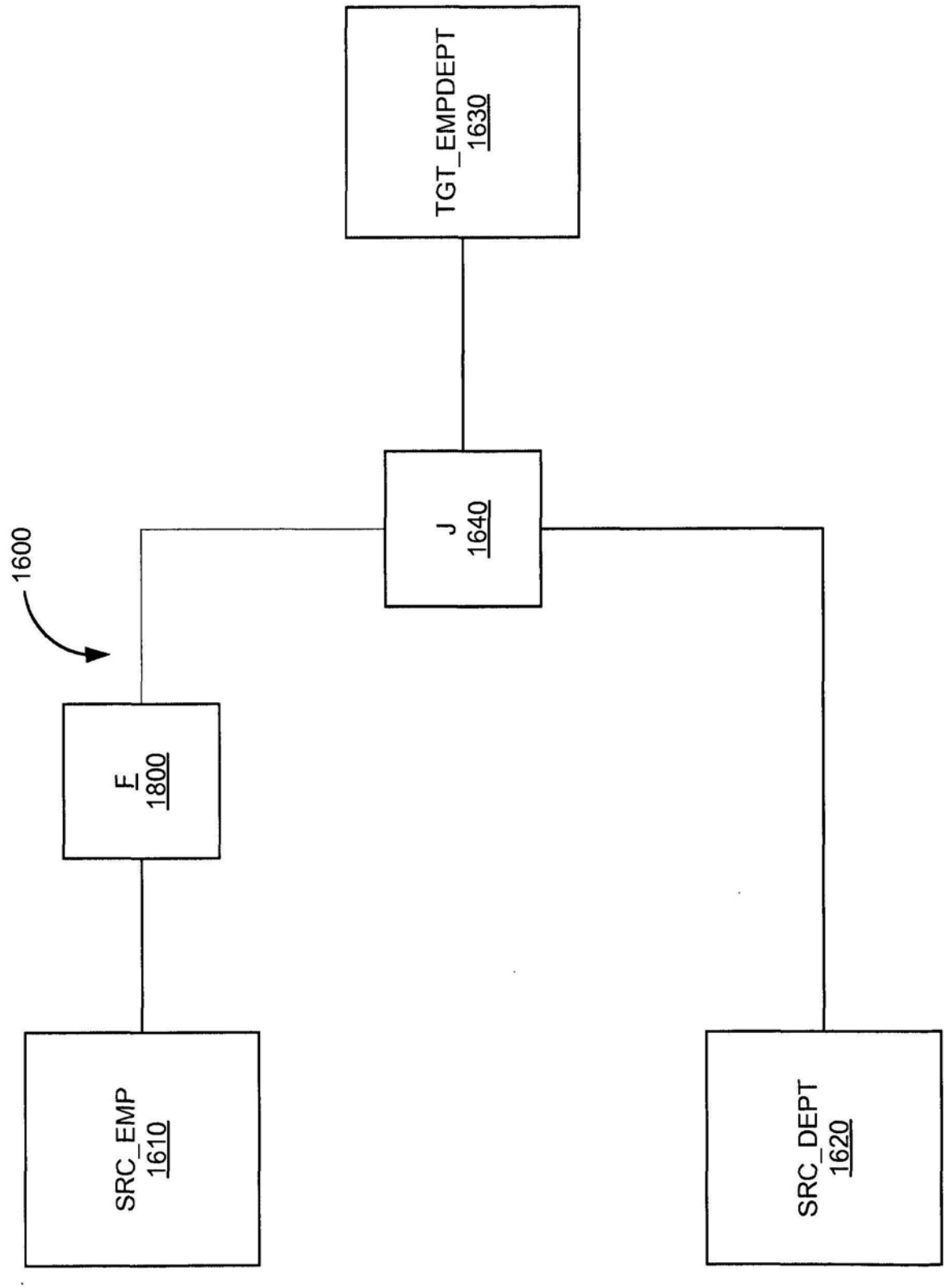


图18

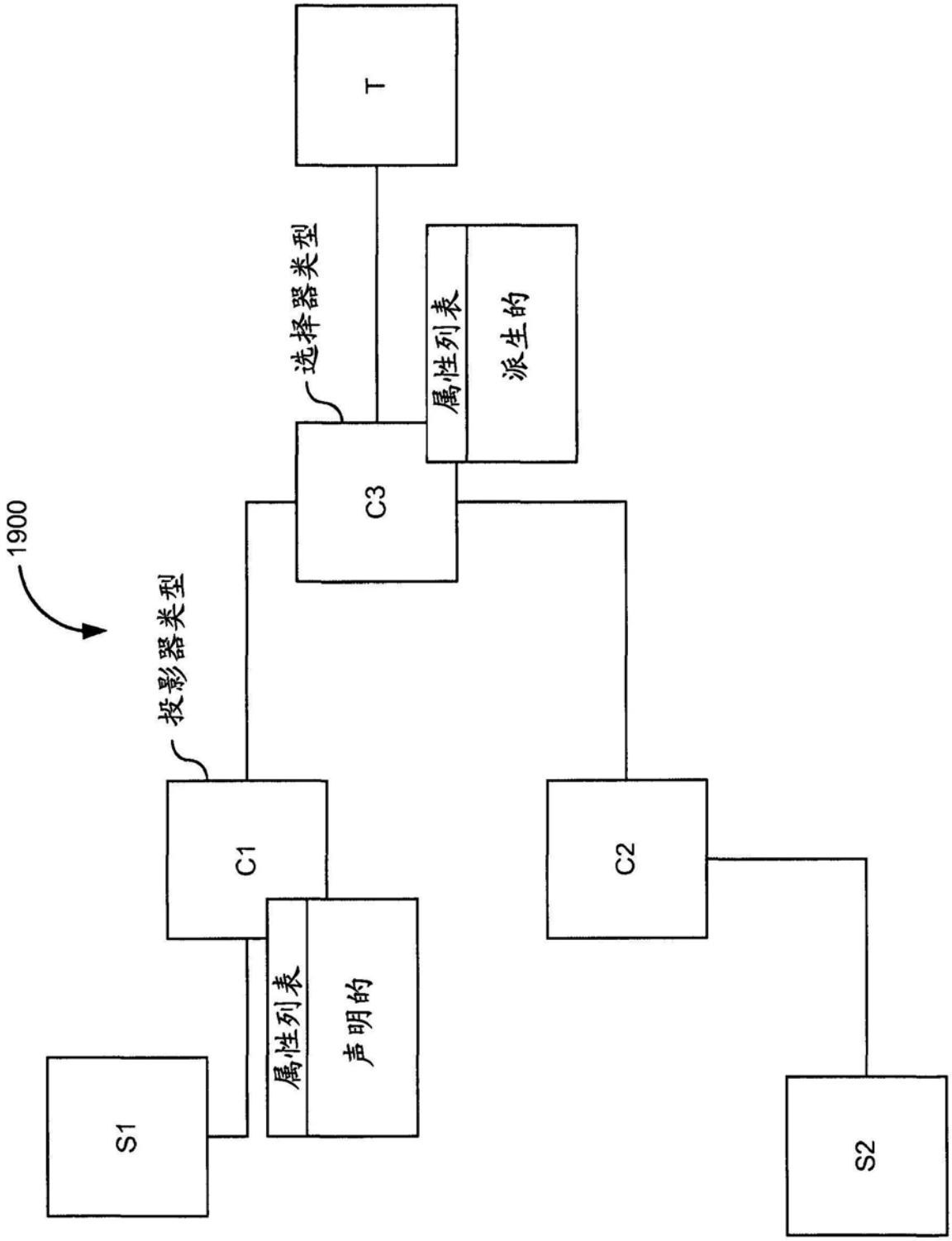


图19

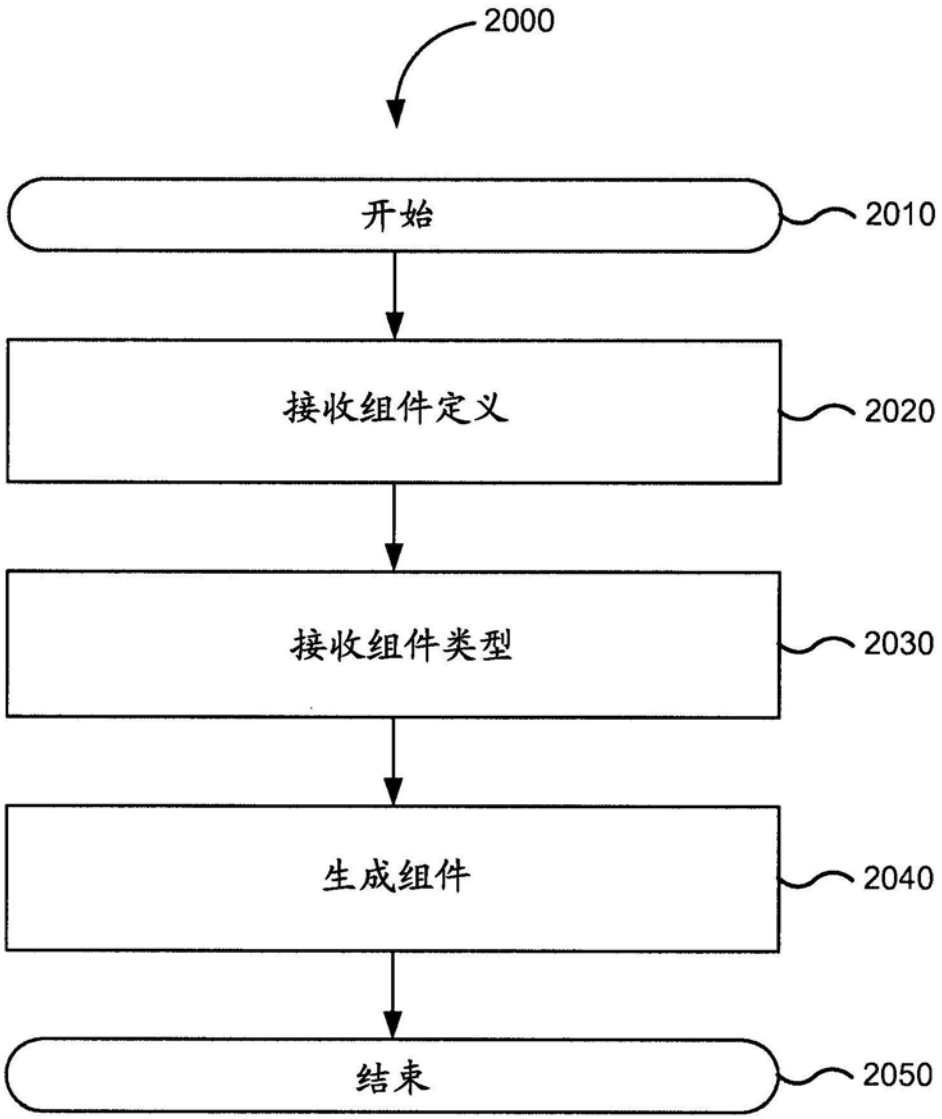


图20

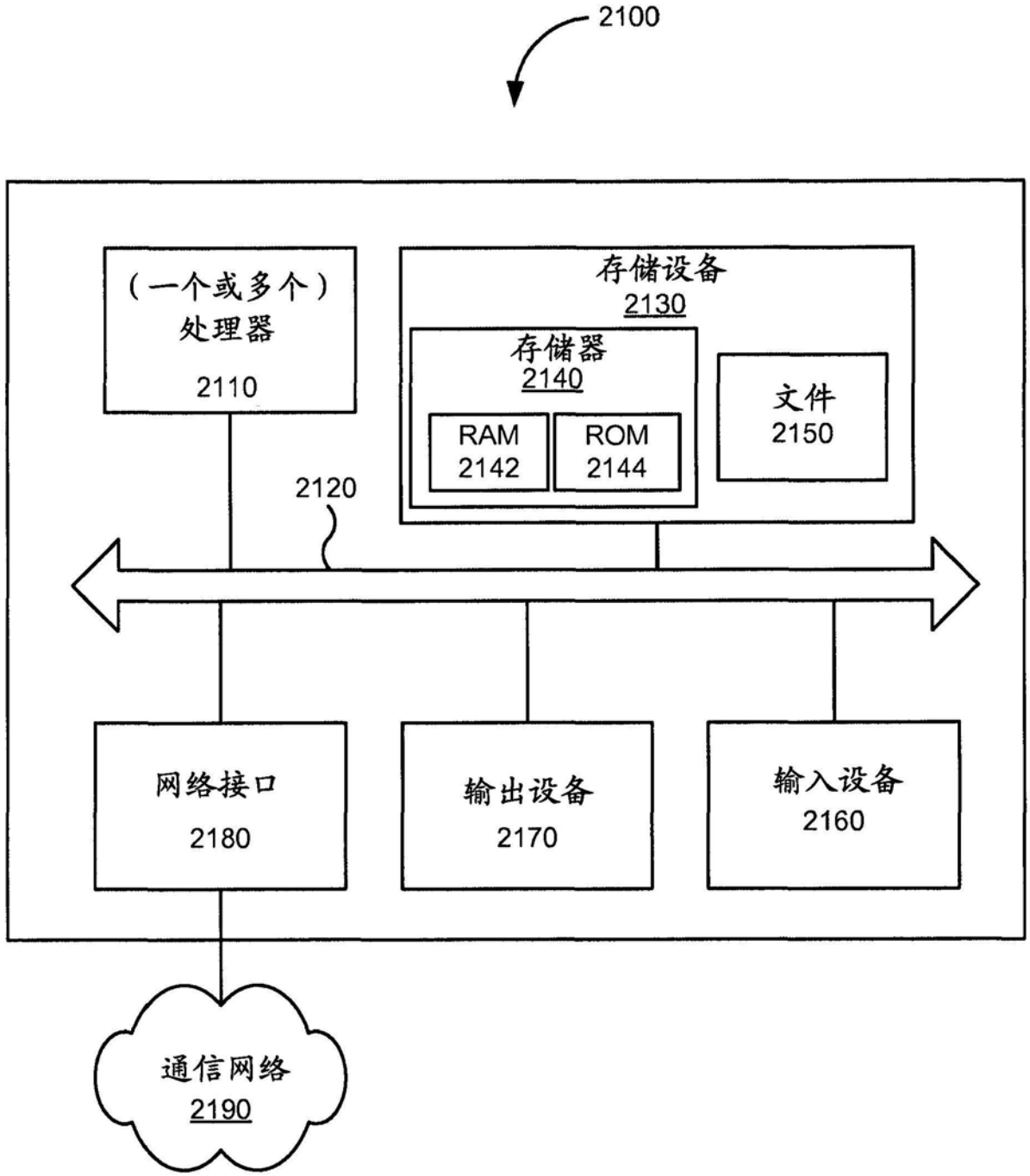

2100
（一个或多个）
处理器
2110
存储设备
2130
存储器
2140
RAM
2142
ROM
2144
文件
2150
2120
网络接口
2180
输出设备
2170
输入设备
2160
通信网络
2190

图21

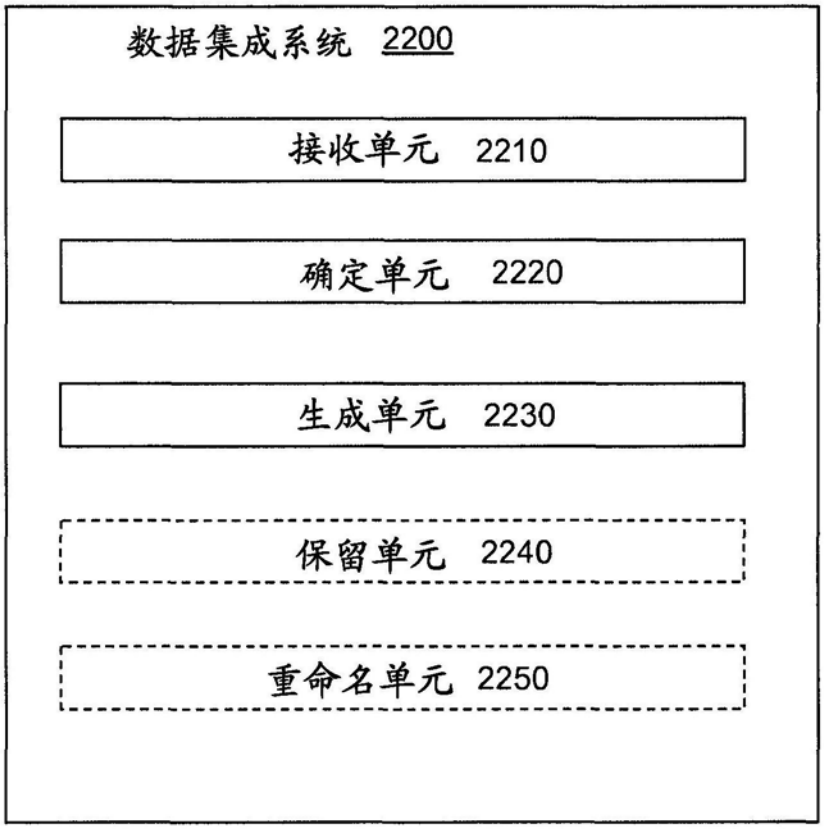

数据集成系统 2200
接收单元 2210
确定单元 2220
生成单元 2230
保留单元 2240
重命名单元 2250

图22