(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 102298607 B
(45)授权公告日 2016.08.17

(21)申请号 201110160095.X

(22)申请日 2011.05.27

(30)优先权数据
12/788,310 2010.05.27 US

(73)专利权人 微软技术许可有限责任公司
地址 美国华盛顿州

(72)发明人 M·卡罗尔 D·诺尔

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 蔡悦

(51)Int.Cl.
G06F 17/30(2006.01)

(56)对比文件
US 2007/0074155 A1,2007.03.29,
US 2005/0177585 A1,2005.08.11,
US 2007/0005619 A1,2007.01.04,
CN 101477547 A,2009.07.08,
US 2005/0228728 A1,2005.10.13,

审查员 李欢

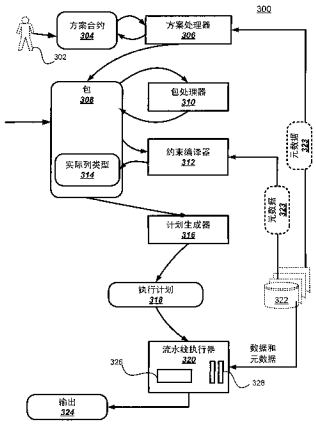
权利要求书3页 说明书11页 附图11页

(54)发明名称

数据集成的方案合约

(57)摘要

本发明公开了数据集成的方案合约。一种用于从输入源提取数据、变换该数据并且将所变换的数据加载到输出目的地的系统和方法。使用包括约束的方案合约来确认输入数据列类型集合，并且将它们转换成实际类型集合。映射该方案合约中的数据允许将输入数据列映射到数据集成组件所使用的数据列。该约束和映射数据在变换具有不同的输入数据列类型集合的数据集合中提供灵活性，同时提供类型的固定集合以供在数据流执行期间使用。灵活性可允许可任选列、一个或多个列集合、列的不同安排、以及每一列的类型定义的变化。



CN 102298607 B

1.一种将来自输入源的输入数据列的输入集合变换成输出数据列的输出集合的基于计算机的方法，所述方法包括：

a)接收包括对应于所述输入集合的一个或多个约束的方案合约，所述方案合约包括对原型列的规范，其中所述原型列由表示可变数目的输出数据列的占位符来指定，并且被配置成向输出数据列的输出集合添加至少一个新的列集合；

b)接收描述所述输入数据列的集合的元数据，所述元数据包括对应于每一输入数据列的类型定义；

c)接收标识所述输入集合的一个或多个数据流的数据流配置数据；

d)执行由所述数据流配置数据指定的每一数据流；以及

e)在接收每一数据流的数据之前，基于所述元数据和所述一个或多个约束来生成列类型集，每一列类型对应于与所述数据流相关联的各输入数据列。

2.如权利要求1所述的基于计算机的方法，其特征在于，所述方案合约包括描述所述输入数据列的一个或多个列与所述方案合约的相关联的约束之间的映射的数据，所述方法还包括采用所述映射将所述输入数据列的每一个与数据集成组件的输入规范相匹配。

3.如权利要求1所述的基于计算机的方法，其特征在于，还包括采用接口以基于输入列类型定义来向一个或多个数据集成组件查询输出列类型。

4.如权利要求1所述的基于计算机的方法，其特征在于，还包括采用接口以基于输入列类型定义来执行对一个或多个数据集成组件的一个或多个查询以寻找输出列类型定义；以及基于所述一个或多个查询来确定所述方案合约是否有效。

5.如权利要求1所述的基于计算机的方法，其特征在于，还包括基于所述列类型的集合来分配一个或多个存储器块；接收来自所述输入源的数据；以及将来自所述输入源的数据存储在所述一个或多个存储器块中。

6.如权利要求1所述的基于计算机的方法，其特征在于，还包括：

a)接收来自另一输入源的输入数据列的另一集合；

b)接收描述数据列的另一集合的其他元数据，所述其他元数据包括不同于与所述输入数据列的集合相关联的对应列类型定义的一个或多个列类型定义；

c)基于所述方案合约来生成对应于所述另一输入源的列类型的另一集合。

7.如权利要求1所述的基于计算机的方法，其特征在于，所述方案合约包括可任选列的规范以及对应于所述可任选列的默认值，所述方法还包括如果所述可任选列从所述数据列的输入集合被省略，则生成具有对应于所述默认值的数据的列。

8.如权利要求1所述的基于计算机的方法，其特征在于，所述方案合约包括表示可变数量的列的原型列的规范，所述方法还包括：

a)确定对应于所述原型列的数据列的输入集合的一个或多个列；

b)在一个或多个所确定的列中确定一个或多个列；以及

c)提供所确定的一个或多个列作为对数据集成组件的输入。

9.如权利要求1所述的基于计算机的方法，其特征在于，还包括：

a)执行对所述方案合约的一个或多个约束的设计确认；以及

b)在每一数据流的执行之前，执行对应于每一输入数据列的所述类型定义的运行时确认。

10.一种用于便于将输入数据变换成输出数据的基于计算机的系统，所述系统包括：

a)一个或多个数据集成组件，每一数据集成组件被配置成接收输入数据列并且基于所述输入数据列来生成输出数据列；

b)方案处理器，被配置成接收包括一个或多个约束和列映射数据的方案合约，并且确认所述方案合约，所述方案合约包括对原型列的规范，其中所述原型列由表示可变数目的输出数据列的占位符来指定，并且被配置成向输出数据列的输出集合添加至少一个新的列集合；

c)约束编译器，被配置成接收所述一个或多个约束，接收描述来自输入源的数据列的元数据，并且生成实际列类型集合，以供所述一个或多个数据集成组件使用来变换所述输入数据列；

系统被配置成采用所述方案合约，以供变换输入数据的第一集合以及变换输入数据的第二集合，所述输入数据的第一集合以及所述输入数据的第二集合各自具有与输入列类型规范的其他集合不同的相应的输入列类型规范集合。

11.如权利要求10所述的基于计算机的系统，其特征在于，所述输入数据的第一集合具有第一数量的数据列，所述输入数据的第二集合具有不同于所述第一数量的数据列的第二数量的数据列。

12.如权利要求10所述的基于计算机的系统，其特征在于，所述约束编译器被配置成采用所述列映射数据将所述输入数据列与所述实际列类型的集合相匹配。

13.如权利要求10所述的基于计算机的系统，其特征在于，所述约束编译器包括用于确认所述方案合约的装置。

14.如权利要求10所述的基于计算机的系统，其特征在于，所述约束编译器被配置成通过查询所述数据集成组件来确认所述方案合约。

15.一种用于便于将输入数据变换成输出数据的方法，所述方法包括：

a)接收一个或多个列约束集合；

b)接收描述来自输入源的输入列集合的元数据，所述输入列集合包括至少一个原型列，其中所述原型列由表示可变数目的输出数据列的占位符来指定，并且被配置成向输出数据列的输出集合添加至少一个新的列集合；

c)基于所述列约束来确认所述元数据；

d)基于所述列约束将所述元数据转换成实际列类型集合；以及

e)向一个或多个数据集成组件提供所述实际列类型以便于变换从所述输入源接收的数据。

16.如权利要求15所述的方法，其特征在于，还包括接收列映射数据；将元数据转换成实际列类型集合，包括基于所述列映射数据将所述输入列集合中的每一个映射到对应的实际列类型。

17.如权利要求15所述的方法，其特征在于，还包括使得所述一个或多个数据集成组件能够变换从另一数据源接收的其他数据，所述其他数据具有在类型上不同于所述输入列集合的另一输入列集合。

18.如权利要求15所述的方法，其特征在于，还包括基于所述元数据来确定第一数据集成组件的输出数据类型，以及基于所述输出数据类型和第二数据集成组件的输入数据类型

来确定所述元数据和所述约束集合的有效性。

数据集成的方案合约

技术领域

[0001] 本发明涉及计算机领域，尤其涉及计算机领域中的数据处理。

背景技术

[0002] 相关数据库通常包括一个或多个表。每一表具有一个或多个记录，并且表的每一记录具有一个或多个字段。表的记录被称为行，而字段被称为列。每一列具有相关联的元数据，该元数据为每一记录描述在字段中的数据的类型、大小或其他属性。方案包括每一表的每一列的元数据以及每一表的其他规范，诸如排序字段、键等等。

[0003] 提取、变换和加载系统(ETL)是从特定数据源提取数据、变换数据以将其转换成所需状态并且将所变换的数据加载到指定目的地的基于计算机的系统。ETL系统可用于各种环境中。例如，异构系统可具有按第一格式、方案或安排来存储的某些数据，以及使用不同的格式、方案或安排的系统的其他部分。ETL系统可被用于集成两个子系统。变换可包括诸如重新格式化、排序、过滤、对数据的列进行组合、或其他类型的修改。ETL系统的输入数据和输出数据各自具有一种方案。输入方案与输入方案可以是相同或不同的。

[0004] ETL所采用的输入和输出方案通常是固定的。为适应各方案中的改变，开发者可按需修改方案。在某些系统中，可由ETL动态地处理方案的各部分。然而，动态过程可能导致数据流实现中的低效。例如，存储器块对于特定数据流可能不是最佳的。

发明内容

[0005] 提供本发明内容以便以简化形式介绍将在以下的具体实施方式中进一步描述的一些概念。本发明内容并不旨在标识出所要求保护的主题的关键特征或必要特征，也不旨在用于限定所要求保护的主题的范围。

[0006] 简言之，系统、方法和各组件用于便于从输入源提取数据、变换数据并且将所变换的数据加载到输出目的地。在一个实施方式中，方案合约包括用于验证输入数据列类型集合并且将它们转换成实际类型集合的约束。在一个实施方式中，将数据映射在方案合约中允许将输入数据列映射到数据集成组件所使用的数据列。

[0007] 在一个实施方式中，映射组件允许对可选列或多组一个或多个列的指定。输入列可按各种方式来安排，并且可被映射到数据集成组件所使用的数据列。输入列的安排、名称、或类型定义可以变化。

[0008] 系统允许使用不同的输入数据源来重新使用方案合约，其中每一输入数据源的列类型规范是不同的。示例差别包括输入列的类型定义的变化。例如，来自不同源的对应列可具有与其他源不同的串长度。

[0009] 在一个实施方式中，在接收每一数据流的数据之前，生成列类型集合，每一列类型对应于与该数据流相关联的相应的输入数据列。生成可基于受约束的方案合约集合以及对应于输入数据列的元数据。

[0010] 为了实现前述及相关目的，在这里结合以下描述及附图来描述系统的某些说明性

方面。然而，这些方面仅指示了可采用本发明的原理的各种方法中的少数几种，且本发明旨在包括所有这样的方面及其等效方面。通过结合附图考虑本发明的以下详细描述，本发明的其它优点以及新颖的特征将变得显而易见。

附图说明

[0011] 参考以下附图来描述本发明的非限制性且非穷尽性实施方式。在各附图中，除非另外指明，否则在全部附图中相同的附图标记指代相同的部分。

[0012] 为了帮助理解本发明，将参考以下与附图相关联地阅读的具体实施方式，附图中：

[0013] 图1是其中可部署此处描述的机制的数据集成环境的框图；

[0014] 图2是其中可部署此处描述的机制的示例数据集成系统的框图；

[0015] 图3是示出可实现提取、变换和加载系统的示例系统的框图；

[0016] 图4是示出便于数据集成系统中的灵活的约束规范的过程的示例实施方式的流程图；

[0017] 图5是示出生成包括约束规范的包的过程的示例实施方式的流程图；

[0018] 图6是示出执行包括约束规范的包的过程的示例实施方式的流程图；

[0019] 图7是示出执行数据流的过程的示例实施方式的流程图；

[0020] 图8是示出对实际数据类型执行运行时验证的示例过程的流程图；

[0021] 图9A-B示出约束规范与物理输入源之间的映射的示例；以及

[0022] 图10是示出计算设备的一个实施方式的框图，它示出可用于执行此处描述的功能的所选择的计算设备的组件。

具体实施方式

[0023] 下文中将参考附图来更全面地描述本发明的各示例实施方式，附图构成实施方式的一部分且在其中作为示例示出了可在其中实践本发明的各特定示例实施方式。然而，本发明可被实现为许多不同的形式并且不应被解释为被限于此处描述的各实施方式；相反，提供这些实施方式以使得本公开变得透彻和完整，并且将本发明的范围完全传达给本领域技术人员。特别地，本发明可被实现为方法或设备。因此，本发明可采用完全硬件实施方式、完全软件实施方式或者结合软件和硬件方面实施方式的形式。因此，以下详细描述并非是局限性的。

[0024] 贯穿说明书和权利要求书，下列术语采用此处显式相关联的含义，除非该上下文在其他地方另有清楚指示。如此处所使用的，短语“在一个实施方式中”尽管它可以但不一定指前一实施方式。此外，如此处所使用的，短语“在另一个实施方式中”尽管它可以但不一定指一不同的实施方式。因此，可以容易地组合本发明的各实施方式而不背离本发明的范围或精神。类似地，如此处所使用的，短语“在一个实现中”尽管它可以但不一定指相同的实现，并且可以组合各种实现的技术。

[0025] 另外，如此处所使用的，术语“或”是包括性“或”运算符，并且等价于术语“和/或”，除非上下文清楚地另外指明。术语“基于”并非穷尽性的并且允许基于未描述的其他因素，除非上下文清楚地另外指明。另外，在本说明书全文中，“一”、“一种”和“所述”的含义包括复数引用。“在……中”的含义包括“在……中”和“在……上”。

[0026] 此处所描述的组件可以从其上具有数据结构的各种计算机可读介质来执行。组件可通过本地或远程过程诸如按照具有一或多个数据分组(例如,来自一个通过信号与本地系统、分布式系统中的另一组件交互或跨诸如因特网的网络与其它系统交互的组件的数据)的信号来通信。例如,根据本发明的各实施方式,软件组件可被存储在非瞬态计算机可读存储介质上,包括但不限于:专用集成电路(ASIC)、紧致盘(CD)、数字多功能盘(DVD)、随机存取存储器(RAM)、只读存储器(ROM)、软盘、硬盘、电可擦除可编程只读存储器(EEPROM)、闪存或记忆棒。

[0027] 如此处所用的术语计算机可读介质既包括非瞬态存储介质又包括通信介质。通信介质通常以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其他数据,并包括任何信息传递介质。作为示例而非限制,通信介质包括有线介质,如有线网络或直接线连接,以及诸如声学、无线电、红外线及其他无线介质之类的无线介质。

[0028] 图1是其中可部署此处所描述的机制的示例数据集成环境100的框图。可将各实施方式部署在各种环境中;环境100提供一个这样的示例。实施方式100可以是较大的数据集成环境的一部分。

[0029] 如图所示,环境100包括输入数据源102。数据源102可包括一个或多个文件、数据库、存储器结构、网络资源、数据流等、或其组合。数据源102用作提取、变换和加载系统(ETL)104的输入。ETL 104接收来自数据源102的数据,对数据执行各种操作,并且提供数据输出106。数据输出106可具有数据源102的特性,诸如被实现在文件、数据库等中。

[0030] 数据源102可具有输入方案,该输入方案提供各种表和包含在表中的列的属性。数据输出106可具有输出方案,该输出方案提供各种表和包含在表中的列(或更具体地是由ETL104输出的列)的属性。ETL 104可包括在将数据提供到目的地之前执行输入数据的一个或多个变换的一系列操作。由此,数据可以说是随着数据前进或被变换而从每一输入源流过ETL。

[0031] 某些变换可导致输出方案与输入方案不同。示例变换可串接两个串列以产生第三列。输入方案可包括各自的串长度为30的两列的规范。输出方案可用串长度为60的单个列替换这两列。另一示例变换可对表的行进行排序,产生等效于输入方案的每一列的方案。输入方案与输出方案之间的多种配置和关系是可能的。

[0032] 图2是其中可部署此处所描述的机制的示例数据集成系统200的框图。系统200更详细地示出环境100的各方面。

[0033] 如图所示,系统200包括输入源的三种类型:文件202、数据库204以及其他源205。输入源的这三种类型可表示环境100的输入数据源102。系统200还示出ETL系统的三个组件。源组件208、变换组件214以及目的地组件220可以是ETL 104的组件。在一个配置中,源组件208可从输入源中提取数据;变换组件214可修改、概括或以其他方式处理从源组件208接收的数据;目的地组件220可将输出数据加载到诸如文件224、数据库226或其他目的地数据228等输出数据存储中。ETL系统此处还被称为数据集成系统。源组件208、变换组件214以及目的地组件220被称为数据集成组件。

[0034] 在所示的示例系统200中,源组件208、变换组件214以及目的地组件220中的每一个具有其自身的输入接口和输出接口。这些输入接口和输出接口与对应的接口或数据连

接，以便在数据源102与数据输出106之间形成数据流。如图所示，外部列接口206向源组件208提供输入，并且接收来自文件202、数据库204或其他源205的数据。源组件208的输出接口210与变换组件214的输入接口212连接。变换组件214的输出接口216与目的地组件220的输入接口218连接。外部列222可以是与输出源：文件224、数据库226或其他目的地数据228连接的目的地组件220的输出。

[0035] 源组件208、变换组件214以及目的地组件220可形成流水线，其中数据从数据源被接收、由这些组件中的每一个来处理、并且随后输出到数据输出。可以同时执行该流水线内的操作。例如，目的地组件220可能正处理从变换组件214接收的数据，同时变换组件正处理从源组件208接收的数据，而源组件208同时处理从数据源接收的新数据。尽管图2示出简单的配置，但可通过将源组件208、变换组件214或目的地组件220的附加实例添加到ETL系统中来配置更复杂的系统。

[0036] 在某些配置中，源组件、变换组件或目的地组件可执行其他组件的功能。例如，变换组件可导入来自外部源的数据并且将其与存储器中的数据结合，由此用作源组件和变换组件的组合。在另一示例中，变换组件可将数据写入外部目的地。在某些配置中，源组件、变换组件或目的地组件可具有多个输入或多个目的地。例如，这可被用于结合各输入，或将一个输入拆分成多个输出。

[0037] 图3是示出可实现诸如图1的ETL 104等ETL的示例数据集成系统300的框图。图3只是合适的系统配置的一个示例，并且不旨在对本发明的使用范围或功能提出任何限制。因此，可采用各种系统配置而不背离此处描述的机制的范围或精神。

[0038] 系统300的组件可以分布在一个或多个计算设备间，这些计算设备中的每一个通过采用诸如IP、TCP/IP、UDP、HTTP、SSL、TLS、FTP、SMTP、WAP、蓝牙、WLAN等各种有线或无线通信协议中的一种或多种来彼此通信。

[0039] 计算设备可以是专用或通用计算设备。示例计算设备包括大型计算机、服务器、刀片服务器、个人计算机、便携式计算机、通信设备、消费电子产品等。图9示出了可被用于实现系统300或其各部分的计算设备的示例实施方式。

[0040] 图3包括虚线中的用户302，以便指示用户302不是系统300的一部分，但可与该系统交互。在一个实施方式中，用户302将方案合约304中的一个或多个约束提供给系统300。方案合约可包括约束集合，尽管它可包括附加数据。约束本身可以是类型规范。在某些实施方式中，约束可以是通用类型规范。例如，串的约束可适应各种类型或长度的串类型。整数的约束可适应各种整数类型。在一个实施方式中，方案合约包括映射数据以供将输入数据与一个或多个数据集成组件所使用的规范相匹配。

[0041] 映射数据可指定处理可变的列数量的方式。例如，约束可指定任何类型的一个或多个列，或任何类型的零或多个列。另一示例约束可指定串的一个或多个列。约束可指定单个可选列，以及类型约束和默认值。如果可选输入列被省略，则可将默认值和类型约束插入到数据流。

[0042] 某些映射数据可提供规范，以便启用输入列与数据集成组件所处理的列之间的映射，同时保留列的语义信息。例如，映射机制可通过名称来指定列。如果对应的列位于相对于其他列的、数据集成组件期望之外的不同位置，则该对应的列可由列名称规范来映射。在某些实施方式中，名称规范可通过诸如使用通配字符或其他机制来概括，以便匹配输入列

名称。例如，约束可指定被称为“AB*”的一个或多个串列，并且映射到以字符“AB”开始的列。

[0043] 在一个实现中，内容映射可提供物理列引用与内部引用之间的对应关系。当系统的组件被改变但提供了等效的列时，映射机制便于每一列的自动重新映射。在一个实施方式中，系统可向用户呈现期望的列和实际输入列的语义上下文，并且使得用户能够指示一个或多个列的列映射。图9提供列映射的示例。

[0044] 在方案合约304中的约束可按多种形式中的任意一种存在，并且可按各种方式来生成。在一个实施方式中，方案处理器306接收来自对应于图1的输入数据源102的输入源322的元数据323。以虚线示出的输入源322和元数据323在系统300的外部，但可作为输入提供给系统。元数据323可包括从输入源322接收的数据的实际类型。方案处理器306可便于来自实际类型的约束的规范。在一个实施方式中，方案处理器306可指示每一列的实际类型，并且使得用户302能够用方案合约304中的约束类型替换实际类型。

[0045] 在一个实施方式中，方案处理器306可执行处理以确定可从元数据323中得到的且对系统有效的约束。例如，可检查下游组件输入，并且确定可使实际类型成为更通用的约束。例如，下游输入可指示一般串长度为50的列的规范。对应的实际类型可指定统一字符编码串的长度为30。方案处理器306可确定将实际类型改变为一般串的约束。方案处理器306可向用户302呈现所提议的约束类型，并且便于接受或改变所提议的类型。约束的细节可以变化。约束的某些示例是：长度为50的串；串；任何精度数；任何类型；可任选的列；或一个或多个列。

[0046] 在一个实施方式中，方案处理器306可分析方案合约304中的约束，以便基于下游组件的输入或输出接口来确定该约束是否有效。在一个实施方式中，每一数据集合组件可用输入约束规范来配置。在一个实施方式中，可通过确定数据集成组件输入约束规范是否可适应数据流的对应的约束来执行确认。例如，如果下游组件将整数类型指定为数据流的输入，则数值类型的约束类型可被认为无效。在一个实施方式中，ETL的每一数据集成组件可实现使方案处理器306能够查询其输入或输出接口的接口。在一个实施方式中，组件接口可提供基于一个或多个对应输入数据类型来输出的数据类型。可在下游数据集成组件的后续查询中使用输出类型数据。可使用这一信息来确认方案合约304中的约束的每一类型。

[0047] 在一个实施方式中，方案处理器306可向用户302提供警告或出错消息，以便指示无效或可能无效的类型规范。由此，用户302可执行任何数量的迭代，以便产生方案合约304中的约束。

[0048] 在所示的实施方式中，方案处理器306可将方案合约304转换成包308。在一个实施方式中，包308可以是可扩展标记语言(XML)文件，尽管可使用各种其他格式。在一个实施方式中，包308可以是对象模型的形式。在一个实施方式中，可使用诸如XML等第一格式，随后该第一格式被转换成诸如对象模型等第二格式。

[0049] 包可包括控制流规范以及一个或多个数据流规范。控制流元素可包括提供结构的一个或多个容器、提供功能的任务、以及将容器和任务连接到排序的控制流的优先级约束。控制流任务可定义并且执行提取数据、应用变换且加载数据的数据流。

[0050] 如示例系统300中所示，可从另一源接收包308。例如，XML文件可通过另一过程来生成并且向系统300提供，而无需方案处理器306的帮助。某些实现可由此允许从内部或外部过程接收包308。

[0051] 在一个实施方式中，包处理器310可接收包308并将其变换成运行时使用的形式。包处理器310可对包含在包308内的方案执行设计确认。在一个实现中，这一确认基本上可与由方案处理器306执行的设计确认类似。在某些配置中，可跳过方案处理器306的设计确认，诸如在从另一源接收包308的配置中。由此，包处理器310的设计确认可重复某些确认，或可执行先前未执行过的确认。注意，尽管对于这一讨论，方案处理器306和包处理器310被示为不同组件，但在某些实现中，可至少部分地将它们组合成一个或多个组件。

[0052] 在一个实施方式中，包处理器310可开始对包308的执行。这可包括对与工作流不同的一个或多个任务的执行。例如，对应于包308的任务可包括将一个或多个电子邮件或其他消息发送给指定收件人，经由FTP或其他协议传送一个或多个文件，执行指定脚本，或其他动作。在一个实现中，包处理器310可启动工作流过程。该工作流过程可包括对约束编译器312的执行。

[0053] 在一个实施方式中，约束编译器312接收在包308内指定的约束，并基于从输入源322接收的元数据323将其变换成实际列类型314。元数据323可指示对应数据的实际类型。在一个实现中，约束编译器312可执行在包308中指定的列约束的运行时确认，确定各种数据集成组件是否被配置成处理对应类型。如果约束有效，则约束编译器312可基于元数据323和包308生成实际列类型314。在一个实现中，可将实际列类型314包含在包308内，尽管在某些实现中，可不同于包308。

[0054] 作为示例，诸如变换组件314等数据集成组件可接收包括具有统一字符编码串30的实际类型规范的第一列以及具有统一字符编码串40的实际类型规范的第二列。变换组件214可具有指定串输入的约束。系统可确定基于串接这两列的操作让变换组件具有统一字符编码串70的输出类型规范。

[0055] 实际列类型314包括要从输入源322接收的每一输入类型的类型规范。在一个实施方式中，约束编译器312可确定对每一数据集成组件的每一输入列的实际列类型，并且将这一信息包括在实际列类型314中。由此，可确定并包括对应于外部列206、输入接口212或输入接口218的输入列规范。

[0056] 注意，可使用由此处所描述的系统基于第一输入源322创建的包308来处理第二输入源322。第二输入源322可具有于第一输入源322不同的类型。当执行第二输入源322时，实际列类型314可以与当处理第一输入源322时生成的实际列类型相同或不同。

[0057] 在一个实施方式中，计划生成器316可接收包括实际列类型314的包308，并且基于包308和实际列类型314来生成执行计划318。执行计划318可包括表维度、要分配的存储器块、要采用的多个线程以及线程之间的关系、要调用的过程以及过程的时序、工作流的一个或多个图、或其他这样的数据的规范。执行计划318可用作工作流执行的蓝图。执行计划318可由任何一个或多个数据集成组件使用。在一个实现中，在从输入源322接收数据之前生成执行计划318。它可在执行工作流期间保持固定。具有固定的执行计划提供了可改进性能、减少资源使用或提供其他益处的操作效率。

[0058] 在一个实施方式中，流水线执行器320可接收执行计划318，并且分配诸如存储器326、线程328、数据结构、对象等资源。流水线处理器320可通过诸如源组件208、变换组件214或目的地组件220等各种组件来启动或管理工作流的执行。输出320可通过流水线执行来产生。输出324可包括数据输出106。它还可包括向用户呈现的数据，诸如状态数据、消息

等。在一个实施方式中，每一数据集成组件实现便于此处描述的机制的接口。具体地，数据集成组件可接收指定供执行的实际类型的一个或多个列的方法调用。可跟踪对输出列的改变，并将其传播到下游组件。

[0059] 图4是示出便于数据集成系统中的方案合约的过程400的示例实施方式的流程图。在一个实施方式中，过程400的至少某些动作是由图3的系统300的组件执行的。

[0060] 过程400的所示部分可在框402启动，在那里包被生成，该包包括一个或多个灵活的约束或映射数据。包308是这样的包的一个示例。框402的动作在图5以及相关联的讨论中更详细地示出和描述。该过程可前进至循环404，该循环为输入源的每一集合以及对应执行迭代。如此处讨论的，可使用具有包括一个或多个不同列类型的不同元数据的一个或多个输入源集合来重复使用由此处描述的机制生成的包。循环404的每一迭代与数据集成执行以及对应的数据源集合相对应。在所示实施方式中，循环404包括块406-408并且由块410终止。

[0061] 该过程可前进至框406，在那里接收输入数据源的新集合。该过程可前进至框408，在那里执行当前的包。图6示出用于执行包的过程的示例实施方式。

[0062] 该过程可前进至框410，并且基于系统配置或用户的控制动作可选择地执行循环404的另一迭代。在退出循环404之后，该过程可前进至完成框412，并且退出或返回到调用程序。

[0063] 图5是示出用于生成包括约束规范或映射数据的包(诸如图3的包308)的过程500的示例实施方式的流程图。过程500或其变型可执行图4的框402的至少某些动作。过程500的所示部分可在框502启动，在那里系统便于方案合约的生成。如此处讨论的，诸如方案处理器306的系统组件可执行诸如从输入源接收元数据、分析该元数据、以及使得用户能够将实际方案变换成约束集合的动作。这可包括将至少某些类型规范自动地修改成对应的约束规范。它可包括向用户呈现规范，以及接收命令以修改规范。

[0064] 该过程可前进至框504，在那里可基于方案合约以及诸如源组件208、变换组件214和目的地组件220等数据集成系统的配置来执行设计确认。该过程可前进至框506，在那里，基于设计确认来选择性地发出警告或出错消息。

[0065] 如箭头507所示，在某些实施方式中，该过程可回到框502以执行约束生成的附加便利。框502、504和506可以是迭代过程的一部分，在那里用户生成并修改一个或多个约束或映射数据。

[0066] 该过程可从框506回到框508，在那里流程规范被保存到包中，数据流规范包括一个或多个约束规范或映射数据。该过程可前进至完成框510，并且退出或返回到调用程序，诸如过程400。

[0067] 图6是示出用于执行包括约束规范或映射数据的包的过程600的示例实施方式的流程图。过程600或其变型可执行图4的框408的至少某些动作。过程600的所示部分可在框602启动，在那里可开始包的执行。这可包括启动如在包中指定的各种动作，诸如对脚本等的执行。该过程可前进至框604，在那里可执行对包的设计确认。尽管在图6中未示出，但在一个实施方式中，如果出错未找到，则该过程可退出或提示用户输入命令。

[0068] 该过程可前进至循环606，该循环为在包中指定的每一数据流迭代。包可包括一个或多个数据流规范。取决于配置和系统能力，可同时执行对某些数据流的执行。某些数据流

可取决于其他数据流并被顺序地执行。对某些数据流可存在同时执行和顺序执行的组合。尽管循环606被示为顺序循环，但可以理解，可至少部分地并行执行循环606的多个迭代。在每一迭代中，对应的数据流被称为当前数据流。

[0069] 该过程可前进至框608，在那里执行当前的数据流。图7示出用于执行数据流的过程的示例实施方式。该过程可前进至框610，并且基于系统配置或用户的控制动作可选择地执行循环606的另一迭代。在退出循环606之后，该过程可前进至完成框612，并且退出或返回到调用程序，诸如过程500。

[0070] 图7是示出用于执行数据流的过程700的示例实施方式的流程图。过程700或其变型可执行图6的框608的至少某些动作。在一个实施方式中，过程700的至少一部分动作可由图3的流水线执行器320来执行或管理。过程700的所示部分可在框702启动，在那里从诸如输入源322等数据源检索诸如图3的元数据323等外部元数据。这一元数据可包括来自数据源的每一列的数据的实际类型。

[0071] 该过程可前进至框704，在那里确认对应于当前数据流的实际类型，并且生成实际数据类型。在一个实施方式中，执行实际类型的运行时确认的过程可查询接收数据流的每一数据集成组件，以便确认每一输入类型并指定对应的输出类型。随后可提供对应的输出类型作为到下一下游数据集成组件的输入。如果确认成功，则框704的动作可生成每一数据集成组件的输入的实际列类型。当前列类型可因此基于来自组件查询过程的元数据或响应而被确定。图8示出在运行时确认数据流并生成实际列类型的示例过程。

[0072] 该过程可前进至框708，在那里基于实际列类型生成执行计划。在某些实施方式，执行计划可指定要分配的存储器块的大小、操作的顺序、要分配的多个执行线程、执行线程之中的过程分布、要分配或解除分配的其他资源、或要作为数据流执行的一部分来执行的其他动作。执行计划用作后续执行的蓝图。在一个实施方式中，在接收对应的数据流的数据之前，执行框704和708的动作，包括生成实际列类型以及生成执行计划。

[0073] 该过程可前进至框710，在那里执行对数据流的执行。这可包括从输入数据源检索数据、分配存储器、线程或其他资源、调度和执行各种过程等。该过程可前进至完成框712，并且退出或返回到调用程序，诸如过程600。

[0074] 图8是示出执行实际数据类型的运行时生成和确认的示例过程800的流程图过程800或其变型可执行图7的框704的至少某些动作。过程800的所示部分可在框802启动，在那里可检索数据流的布局。数据流布局可指定构成数据流以及其他信息的组件的顺序。

[0075] 该过程可前进至框804，在那里检索对应于当前输入源的实际数据类型。例如，它们可被接收为图3的元数据323的一部分。该过程可前进至循环806，该循环按数据流的次序为数据流的每一数据集成组件循环。循环806包括框808-812并由框814终止。每一迭代的数据集成组件被称为当前组件。迭代的顺序和当前组件可基于在框802检索的数据流布局。

[0076] 该过程可前进至框808，在那里向当前组件提供一个或多个输入类型。例如，如果当前组件接收两个数据列，则系统可向该组件提供每一数据列的实际数据类型。该过程可前进至框810，在那里当前组件可基于其配置来确认实际输入数据类型。如果组件不能正确地处理输入数据类型，则它可返回一无效状态。尽管未示出，但在某些实现中，如果返回无效状态，则循环806可退出并且该过程可退出。

[0077] 该过程可前进至框812，在那里可从当前组件检索一个或多个输出数据类型。当前

组件可确定基于输入类型的输出类型，并且返回这些输出类型。如此处讨论的，输出列的数量可与输入列的数量不同。随后可向后续组件提供所返回的输出类型作为循环806的后续迭代，该后续组件要接收这些输出类型一个或多个作为其输入类型。

[0078] 在一个实现中，数据流的每一数据集成组件可实现包括接收一个或多个输入类型并返回一个或多个输出类型以及确认状态的方法的接口。由此，在某些实现中，框808、810和812的动作可采用这一接口。在各种实现中，可使用其他接口或机制来执行框808-812的动作，并且控制各数据集成组件与流水线执行器320之中的交互。

[0079] 该过程可前进至框814，并且基于系统配置、每一确认的状态或用户的控制动作可选择地执行循环806的另一迭代。循环806可对数据流的每一下游组件重复。在退出循环806之后，该过程可前进至完成框816，并且退出或返回到调用程序，诸如过程700。

[0080] 在运行时确认的示例中，变换组件可被配置成接收两个输入串列并且输出串接的串列。串接的串列的长度取决于输入列的长度。输入串列的每一个可具有无指定长度的串类型的约束。从这一变换组件接收输入的目的地组件可被指定为接收不长于60个字符的串。由约束编译器312的运行时确认可接收指示第一输入串的实际长度为40以及第二输入串的实际长度为15的元数据。变化组件可指示输入类型是有效的以及串的输出列类型的长为55。对下游数据集成组件的后续查询可指示输入列具有长度为55的串类型。

[0081] 使用相同示例，使用相同约束的另一源输入可提供指示第一输入串列的长度为40以及第二输入串列的长度为30的元数据。变换组件可指示输入列的组合是无效的，因为所组合的长度将超过其60的限制。用户随后可重新配置该系统，或根据需要以其他方式执行修改。由此，设计类型规范的灵活性允许输入源类型的变型，尽管这可导致某些配置的运行时无效性。

[0082] 图9A-B示出约束规范与物理输入源之间的映射900的示例。这些示出此处描述的映射与编译机制。

[0083] 表904示出示例数据集成组件的约束规范。该规范可包括三个列引用："输入1"、"输入2"以及"输入3"，它们具有的相应数据类型约束为整数、串、以及串＞＝100。表902示出输入数据源的三列的实际数据类型。"ID"、"名称"以及"描述"三列分别映射到数据集成组件的列引用"输入1"、"输入2"以及"输入3"。箭头930示出示例映射。输入列的数据类型相对于数据集成组件的约束是有效的。

[0084] 表906示出输入数据源的所修改的实际数据类型。修改可由于输入数据源的改变、将不同的数据源用作输入数据源、数据流的中间改变、或环境中的另一改变。与表902相比，表906包括插入在被称为"描述"的列之前的第三位置且被称为"分类"的附加列。

[0085] 表908示出与表904相同的约束规范。箭头932示出表908的列引用与表906的列引用之间的映射。如图所示，如表902与表904之间的原始映射那样，列"输入3"继续被映射到列"描述"。如此处讨论的，诸如列名称的使用、内部映射或来自用户的规范等稳健的映射机制可使得系统能够在输入数据源的列改变之后维护正确的列映射。

[0086] 表910示出输入数据中另一改变。与表902的输入数据相比，表910的输入数据包括与表906相似的附加列，即"分类"。箭头934示出按与箭头932所示的相似方式来保留原始映射。表910还示出列"名称"的实际类型中的改变。在表910中，数据类型是VarChar(40)，它指示长度为40。在表902中，"名称"的数据类型是VarChar(30)。由此，这一字段的大小增加了。

如表912所示，数据集成组件的对应列的约束是“任何串”，该“任何串”可适应列VarChar(30)。如图所示，约束的使用由此允许输入数据类型的至少某些改变。

[0087] 图9B示出映射机制的另一示例。如表916所示，数据集成组件包括列引用“输入1”、“输入2”以及“输入3”，它们具有相应的数据类型约束即整数、串以及串＞＝100。列引用“输入4”具有默认值为空的“任何日期”的对应约束。另一列引用“输入5”具有的对应约束为“*”。

[0088] 在一个实施方式中，约束可指定列是可任选的，并且由此可从输入源中省略。约束可指定列被省略的配置中使用的默认值。表916的“输入4”的约束提供了这样的规范的一个示例，尽管各种实现可使用指定默认列或默认值的其他格式或方式。

[0089] 在一个实施方式中，数据集成组件可提供用于添加此处被称为原型的一个或多个新的列的机制。在一个实现中，一个或多个原型列由星号来指定，尽管具体机制可不同。当确认实际数据类型时以及当编译数据流时，实际列集可与原型列相对应，并插入实际类型。如箭头936所示，在图9B的示例中，原型列“输入5”被映射到列“分类”和“证明”。

[0090] 表916示出基于输入数据源的数据类型来生成、作为对表914的约束进行编译的结果的实际列类型的示例。在一个实现中，编译可由图3的约束编译器312来执行或控制。

[0091] 如图所示，表914的每一列的约束具有带有表916中的实际数据类型的一个或多个列。列“输入1”具有实际数据类型为长度(Long)；列“输入2”具有的实际数据类型为VarChar(40)；列“输入3”具有的实际数据类型为WVarChar(120)。这些数据类型的每一个与来自表912中所示的输入源的对应数据类型相匹配。

[0092] 所指定的实际列“输入4”具有的实际类型为值为空的日期时间，如表914的约束所指定的。由于输入数据源省略了对应的列，因此可以提供这个值。在输入数据源具有对应的列的环境中，该列的值将作为输入来接收。

[0093] 原型列的约束“输入5”被变换成两个列规范，以便匹配对应的数据列。由此，“输入5a”接收实际类型VarChar(20)，该实际类型VarChar(20)从表912的列“分类”中接收；“输入5b”接收实际类型Char(80)，该实际类型Char(80)从列“证明”中接收。

[0094] 注意，基于不同输入源的所编译的实际类型集合可有相当大的变化。如图3所示以及此处所讨论的，可使用所编译的列类型来生成执行计划，并且按有效方式执行每一流水线。

[0095] 图10是示出计算设备1000的一个实施方式的框图，示出可用于实现系统300或执行此处所描述的功能(包括过程400、500、600、700或800)的计算设备的所选组件。计算设备1000可包括比所示多得多的组件，或可包括比所示全部组件要少的组件。计算设备1000可以是独立计算设备或诸如具有一个或多个刀片的机箱中的某一刀片之类的集成系统的一部分。

[0096] 如所示，计算设备1000包括一个或多个处理器1002，处理器执行动作以执行各种计算机程序的指令。在一个配置中，每个处理器1002可包括一个或多个中央处理单元、一个或多个处理器核、一个或多个ASIC、高速缓存存储器或其他硬件处理组件和相关程序逻辑。如所示，计算设备1000包括操作系统1004。操作系统1004可以是通用或专用操作系统。华盛顿州雷蒙德市的微软公司的Windows®系列操作系统是可在计算设备1000上执行的操作系统的示例。

[0097] 存储器和存储1006可包括各种类型的非瞬态计算机存储介质中的一个或多个，包括易失性或非易失性存储器、RAM、ROM、固态存储器、盘驱动器、光学存储、或可用于存储数字信息的任何其他介质。

[0098] 存储器和存储1006可存储此处所述的一个或多个组件或其他组件。在一个实施方式中，存储器和存储1006存储系统300的软件组件或其一部分。存储器和存储1006可存储输入源322、元数据323或它们的一部分。这些组件中的任何一个或多个可通过操作系统1004或其他组件被移动到RAM、非易失性存储器中的不同位置，或在RAM和非易失性存储器之间移动。

[0099] 计算设备1000可包括便于将程序代码或其他信息显示给用户的视频显示适配器1012。尽管在图10中未示出，但是计算设备1000可包括基本输入/输出系统(BIOS)，以及相关联的组件。计算设备1000还可包括用于与网络通信的网络接口单元1010。系统200或300的软件组件可经瞬态介质和网络接口单元1010被接收。计算设备1000可包括一个或多个显示监视器1014。计算设备1000的实施方式可包括一个或多个输入设备1016，诸如键盘、定点设备、音频组件、话筒、语音识别组件、或其他输入/输出机制。

[0100] 将理解图4-8的流程图的每个框以及流程图中的框的组合可由软件指令来实现。这些程序指令可被提供给处理器以生成机器，使得在处理器上执行的指令创建用于实现某一流程框或多个框中指定的动作的手段。这些软件指令可由处理器执行来提供用于实现某一流程框或多个框中指定的动作的步骤。此外，流程图中的一个或多个框或框的组合也可与其他框或框的组合同时执行，或甚至以与所示不同的顺序执行，而不背离本发明的范围和精神。

[0101] 以上说明、示例和数据提供了对本发明的组成部分的制造和使用的全面描述。因为可以在不背离本发明的精神和范围的情况下做出本发明的许多实施方式，所以本发明落在所附权利要求的范围内。

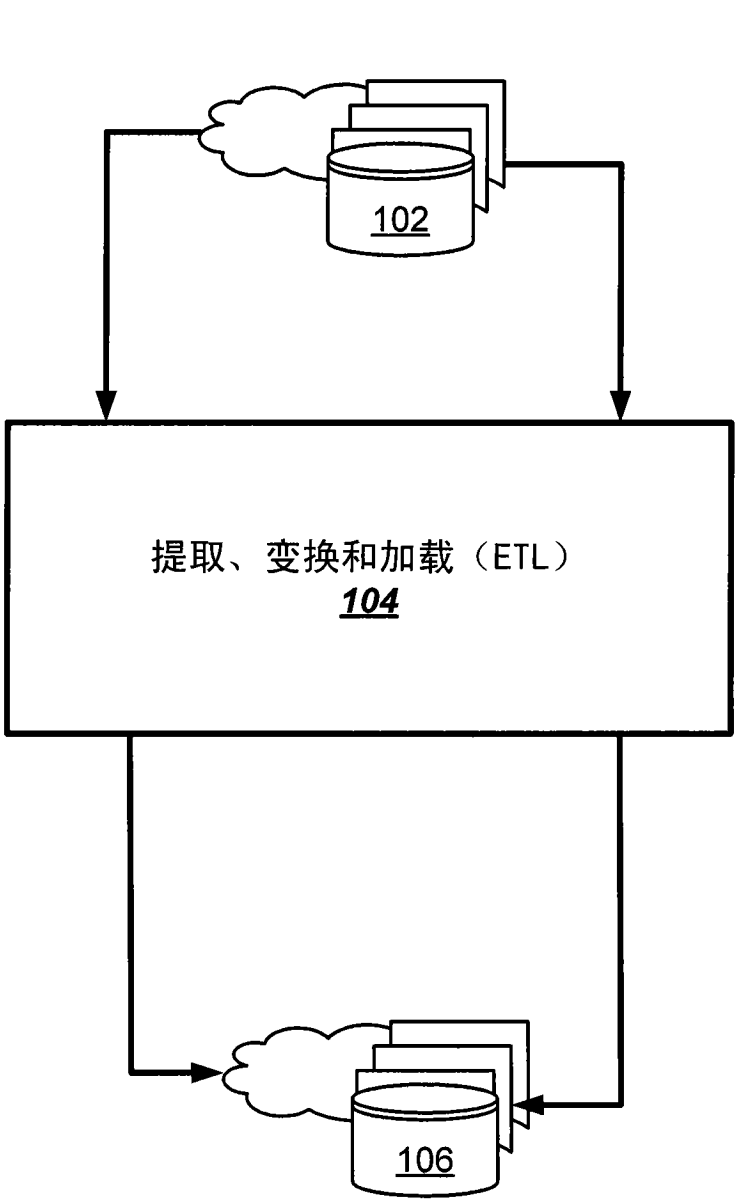


图1

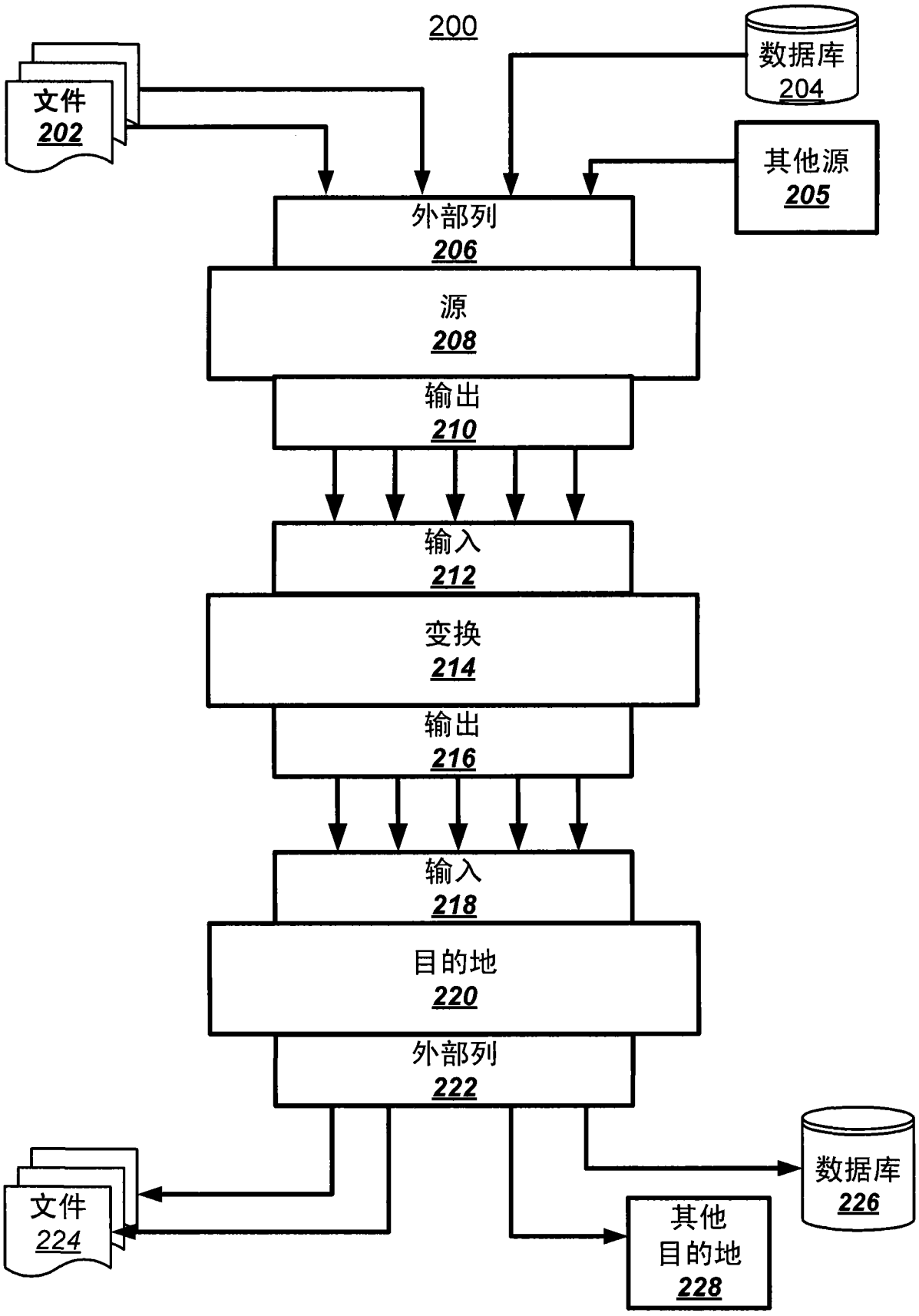

200
文件
202
数据库
204
其他源
205
外部列
206
源
208
输出
210
输入
212
变换
214
输出
216
输入
218
目的地
220
外部列
222
文件
224
数据库
226
其他
目的地
228

图2

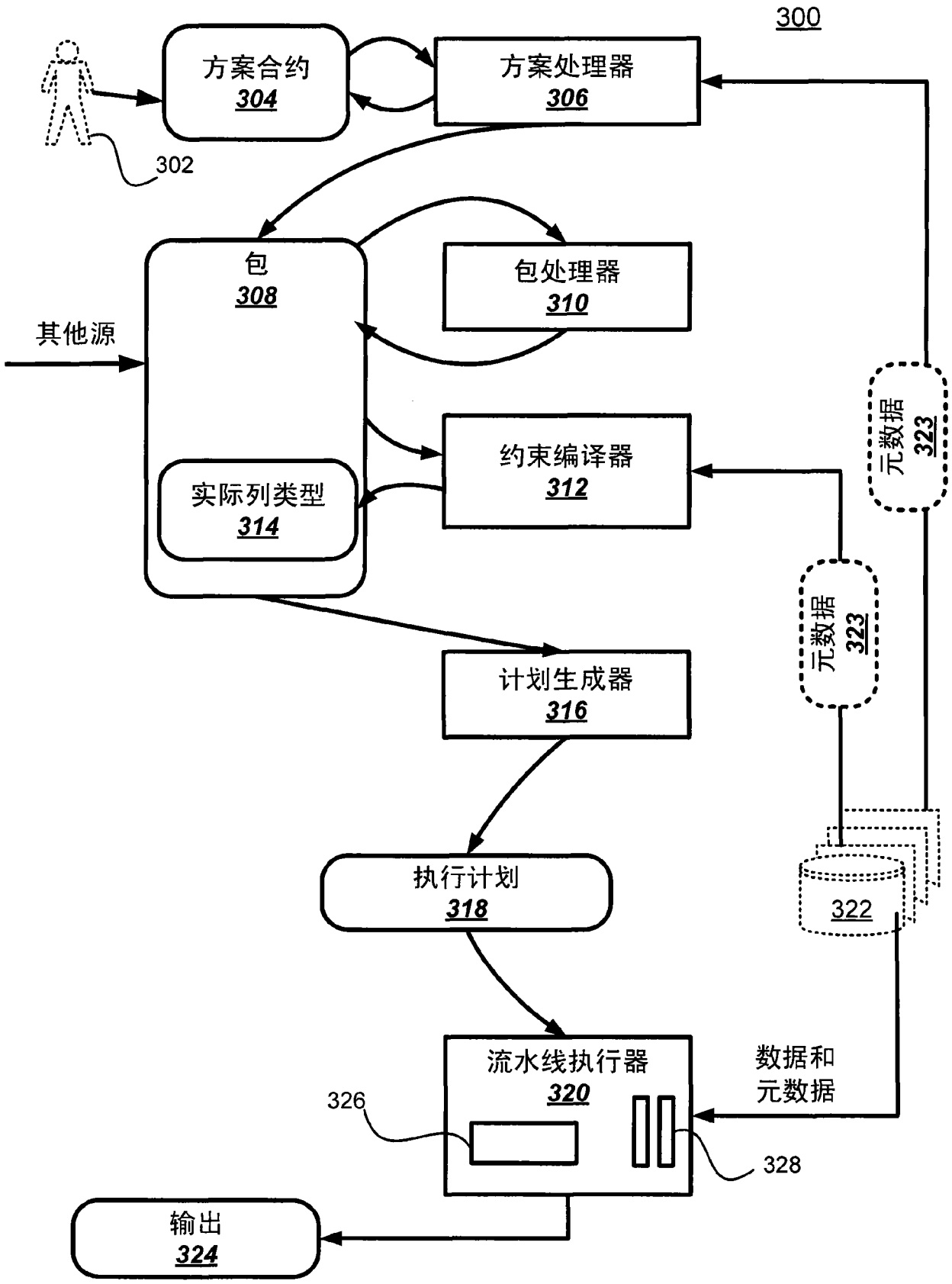

300
方案合约
304
方案处理器
306
302
包
308
包处理器
310
其他源
约束编译器
312
实际列类型
314
元数据
323
元数据
323
计划生成器
316
322
执行计划
318
流水线执行器
320
数据和
元数据
326
328
输出
324

图3

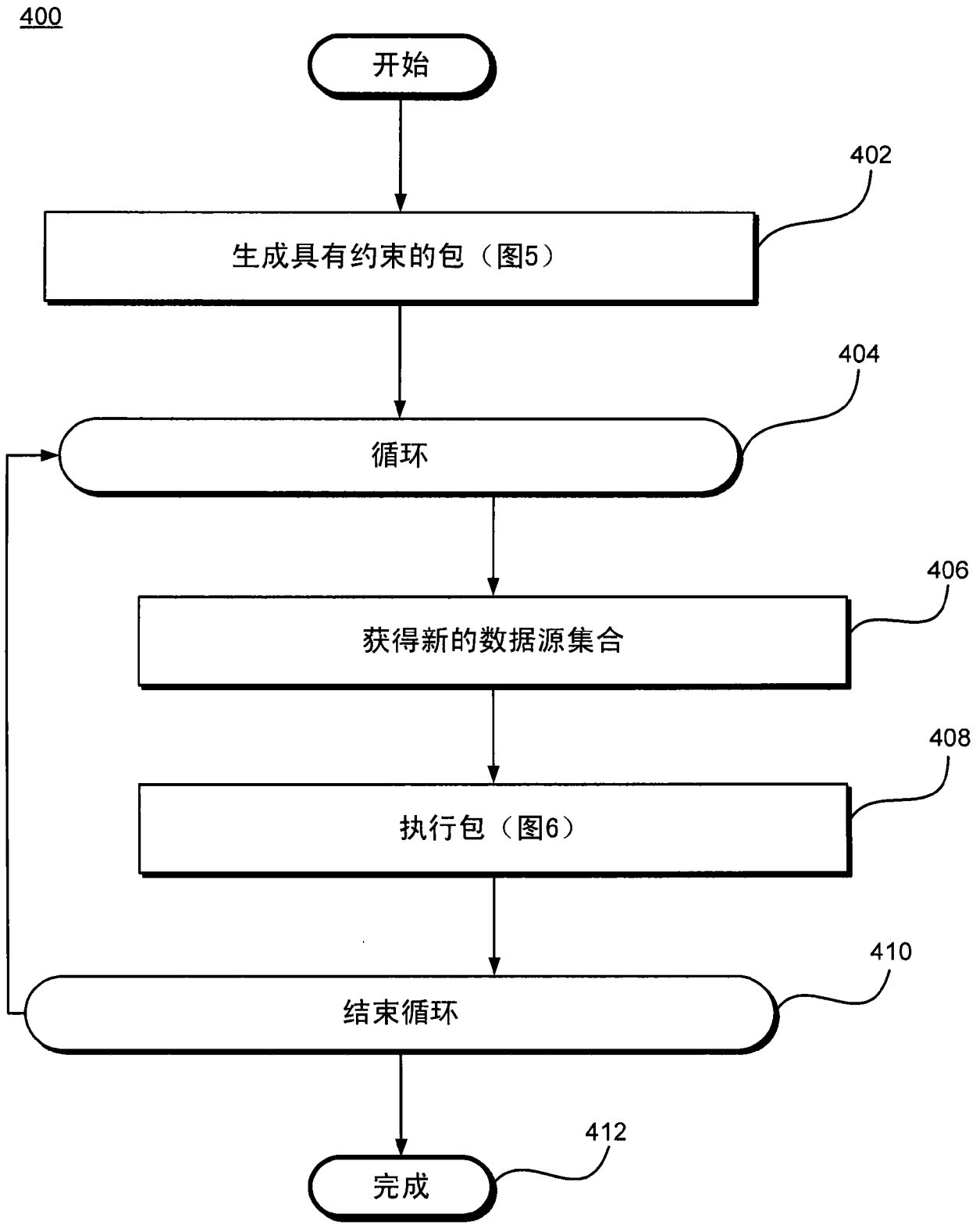

400
开始
402
生成具有约束的包（图5）
404
循环
406
获得新的数据源集合
408
执行包（图6）
410
结束循环
412
完成

图4

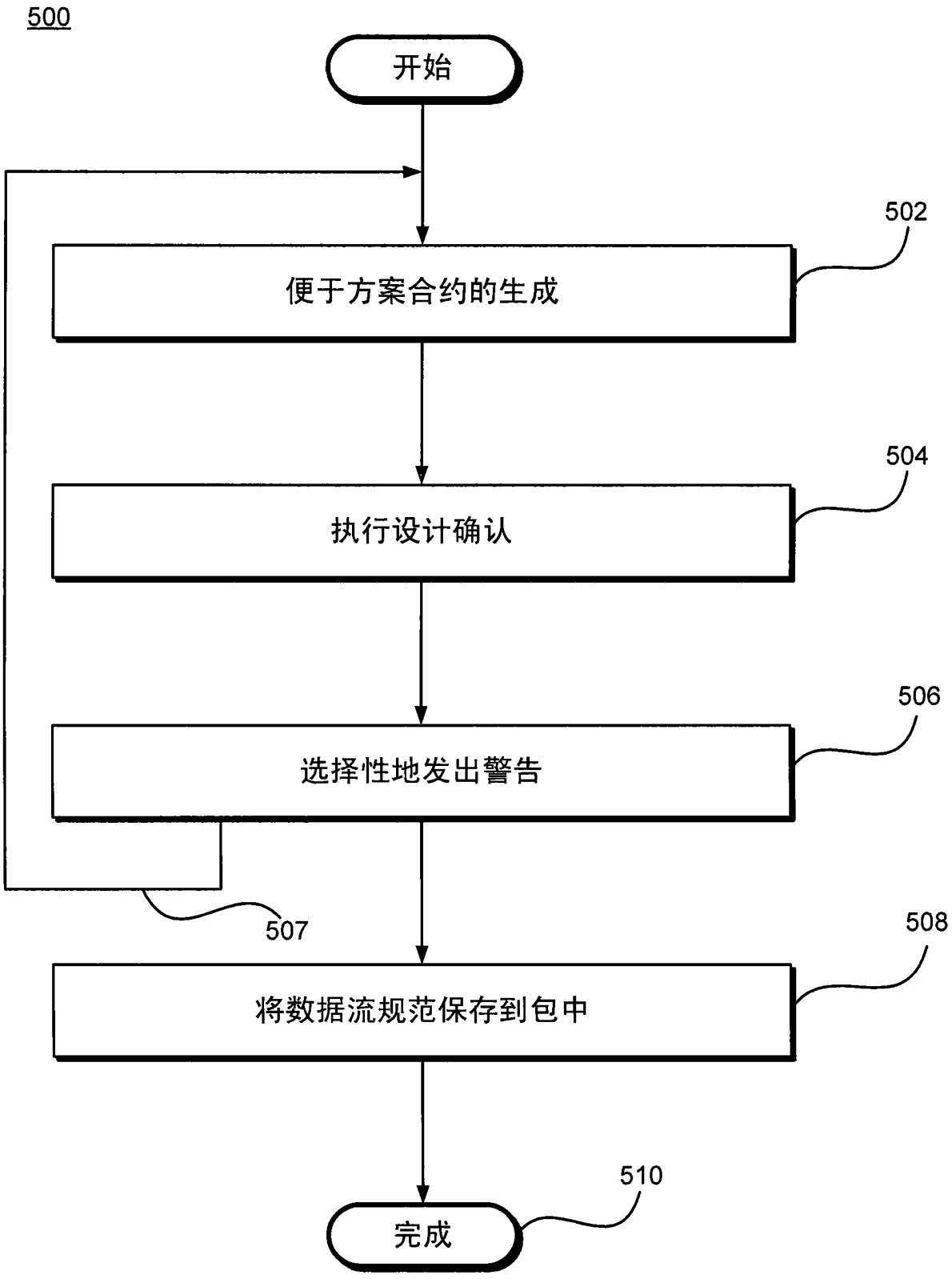

500
开始
502
便于方案合约的生成
504
执行设计确认
506
选择性地发出警告
507
508
将数据流规范保存到包中
510
完成

图5

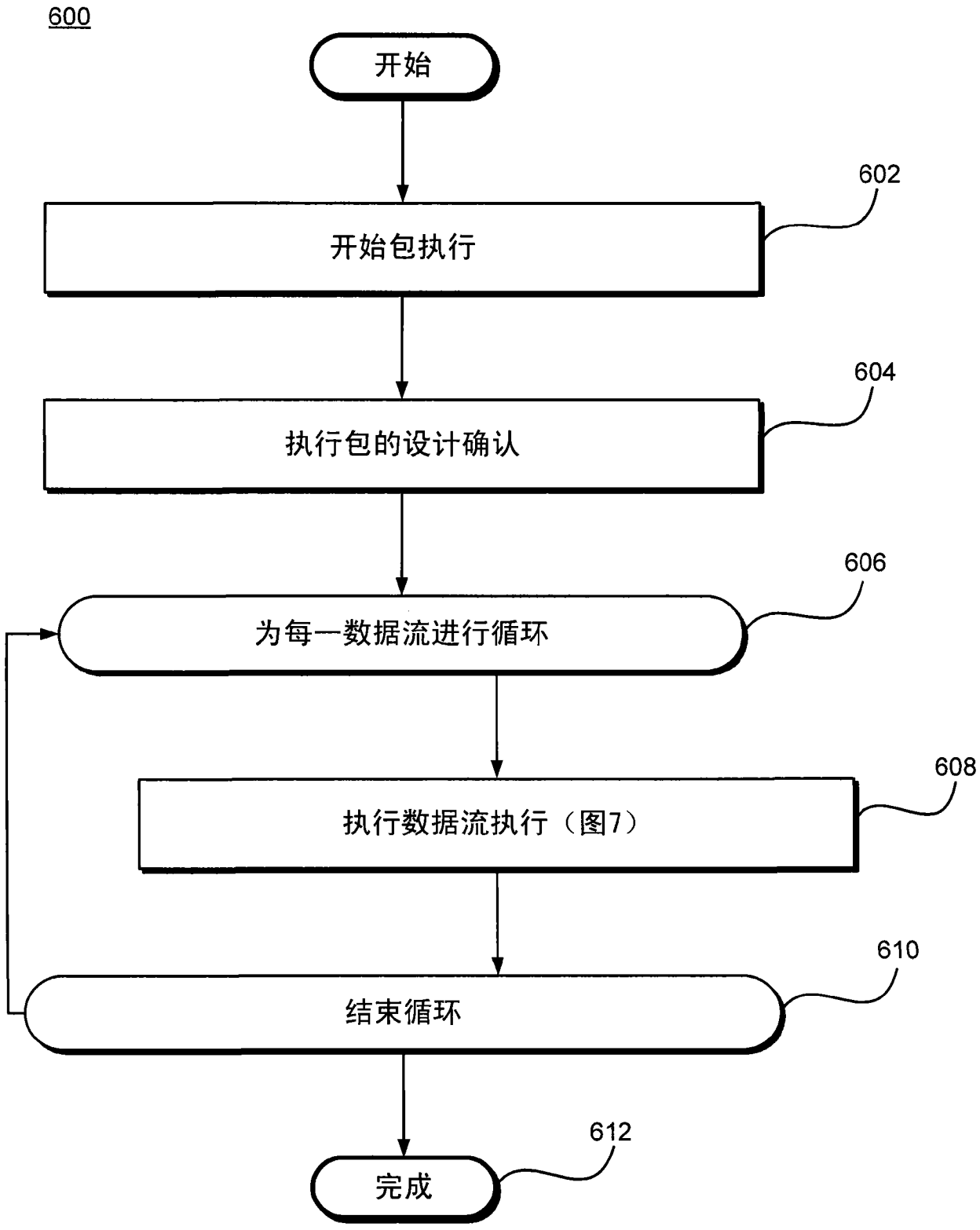

600
开始
602
开始包执行
604
执行包的设计确认
606
为每一数据流进行循环
608
执行数据流执行（图7）
610
结束循环
612
完成

图6

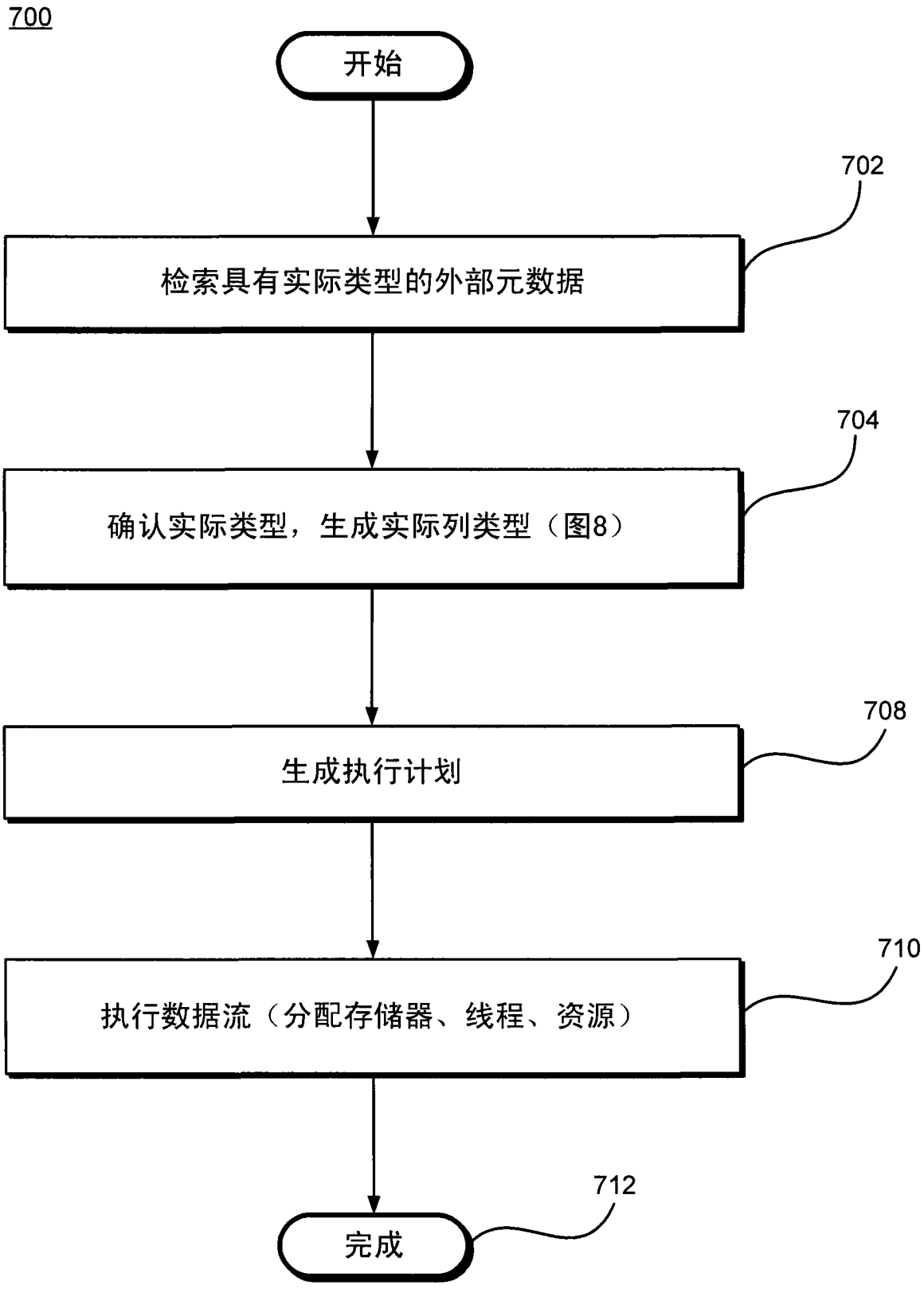

700
开始
702
检索具有实际类型的外部元数据
704
确认实际类型，生成实际列类型（图8）
708
生成执行计划
710
执行数据流（分配存储器、线程、资源）
712
完成

图7

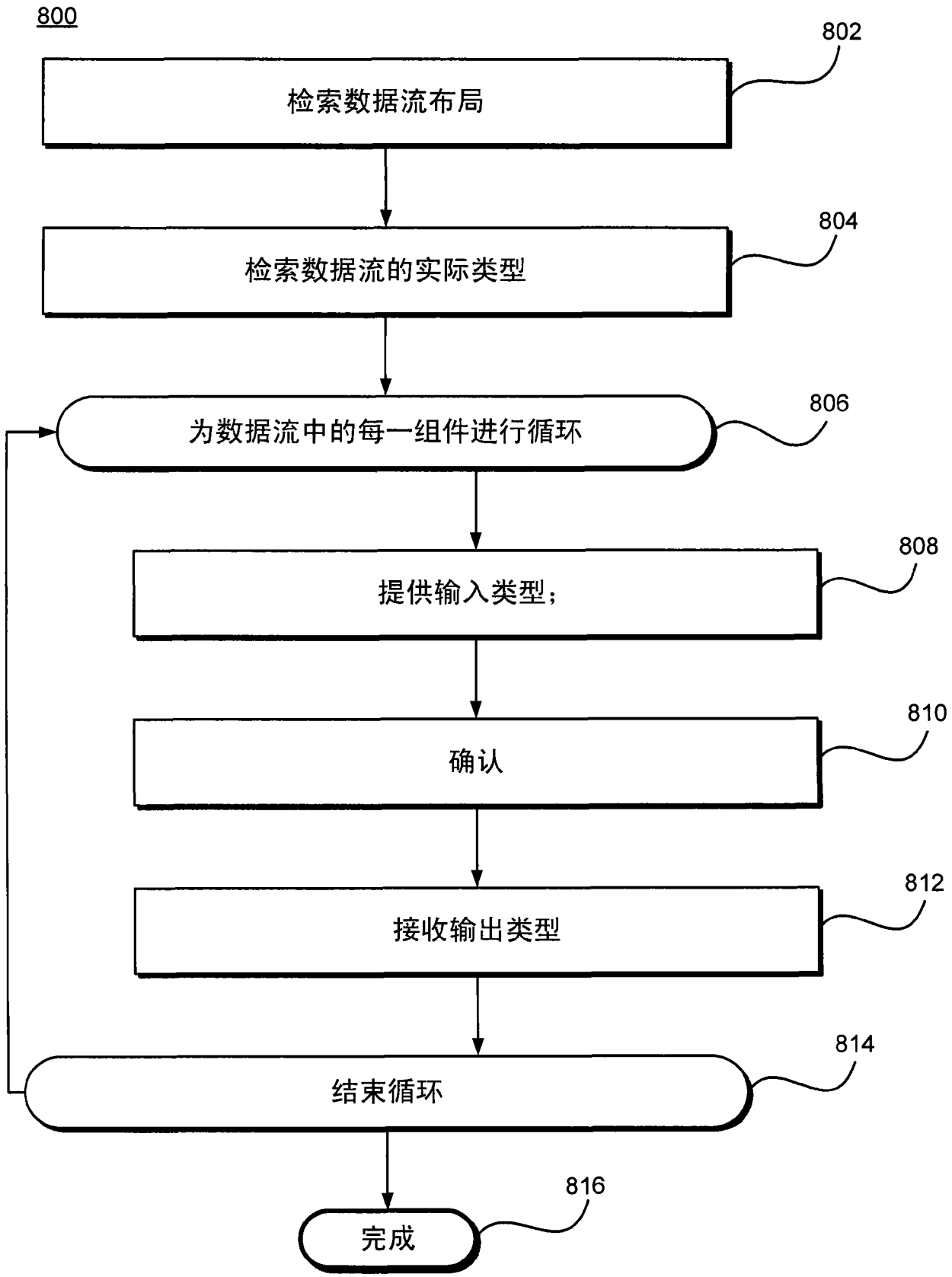

800
802
检索数据流布局
804
检索数据流的实际类型
806
为数据流中的每一组件进行循环
808
提供输入类型；
810
确认
812
接收输出类型
814
结束循环
816
完成

图8

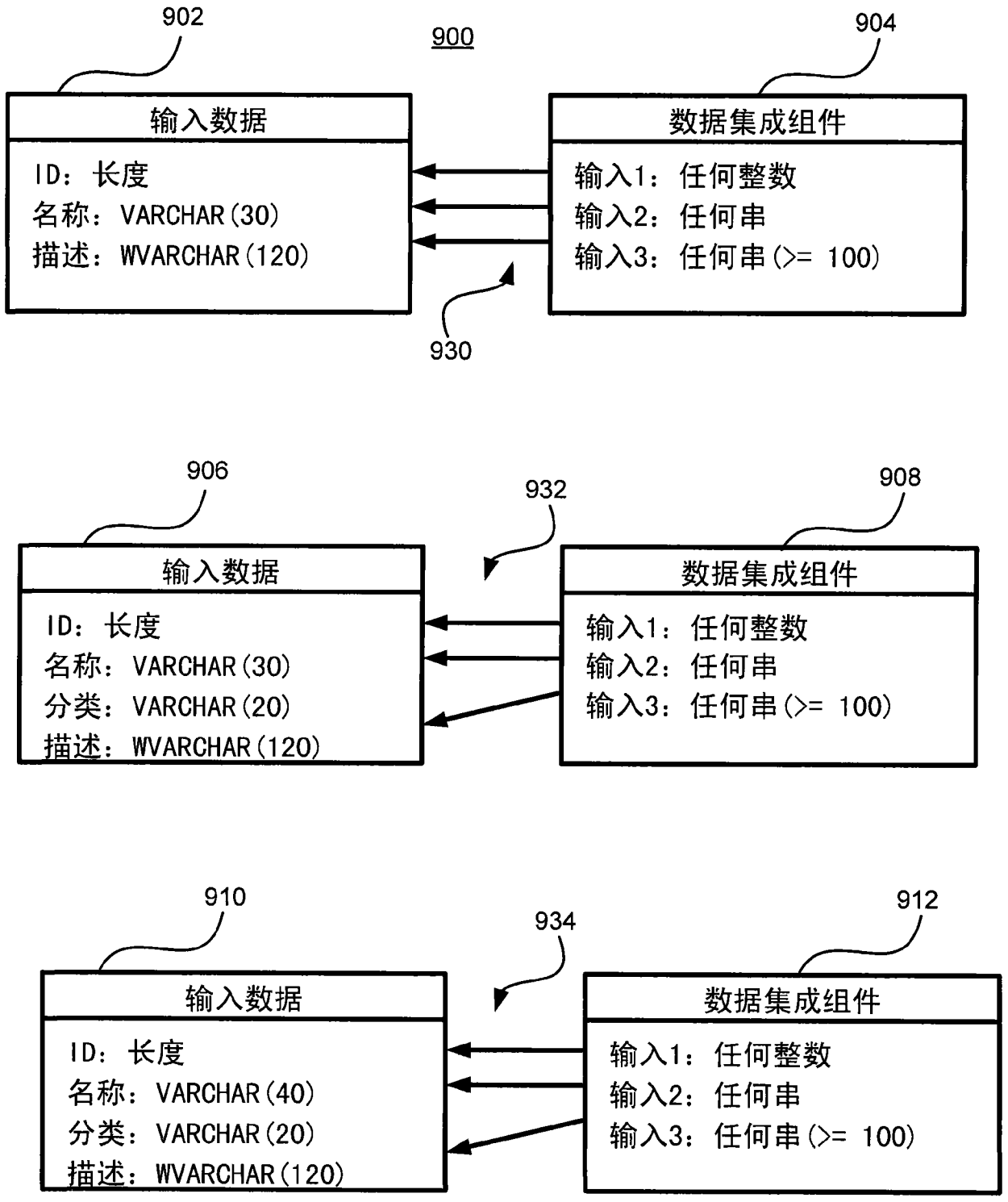

902
900
904
输入数据
ID：长度
名称：VARCHAR(30)
描述：WVARCHAR(120)
数据集成组件
输入1：任何整数
输入2：任何串
输入3：任何串(>= 100)
930
906
932
908
输入数据
ID：长度
名称：VARCHAR(30)
分类：VARCHAR(20)
描述：WVARCHAR(120)
数据集成组件
输入1：任何整数
输入2：任何串
输入3：任何串(>= 100)
910
934
912
输入数据
ID：长度
名称：VARCHAR(40)
分类：VARCHAR(20)
描述：WVARCHAR(120)
数据集成组件
输入1：任何整数
输入2：任何串
输入3：任何串(>= 100)

图9A

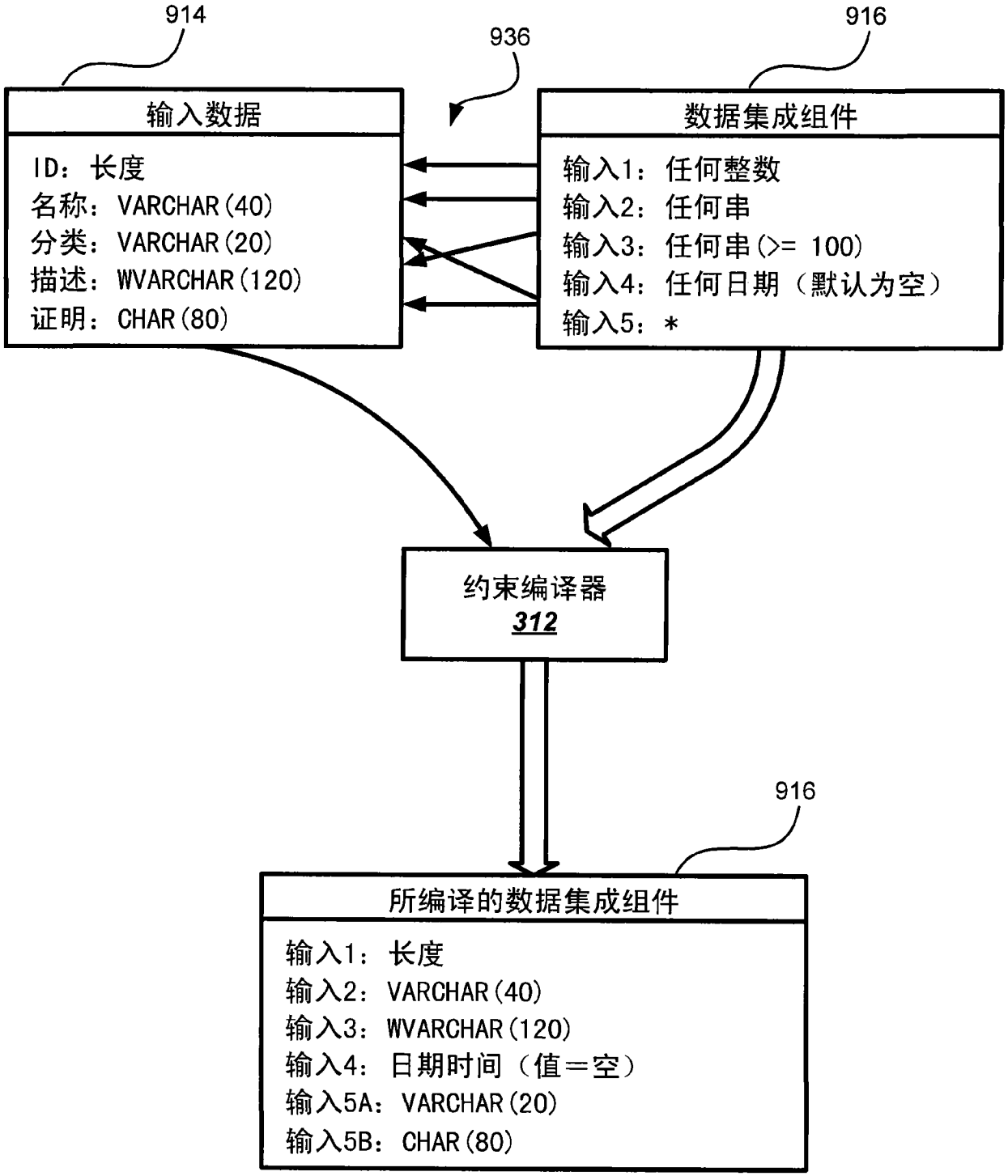

914
936
916
输入数据
ID：长度
名称：VARCHAR(40)
分类：VARCHAR(20)
描述：WVARCHAR(120)
证明：CHAR(80)
数据集成组件
输入1：任何整数
输入2：任何串
输入3：任何串(>= 100)
输入4：任何日期（默认为空）
输入5：*
约束编译器
312
916
所编译的数据集成组件
输入1：长度
输入2：VARCHAR(40)
输入3：WVARCHAR(120)
输入4：日期时间（值＝空）
输入5A：VARCHAR(20)
输入5B：CHAR(80)

图9B

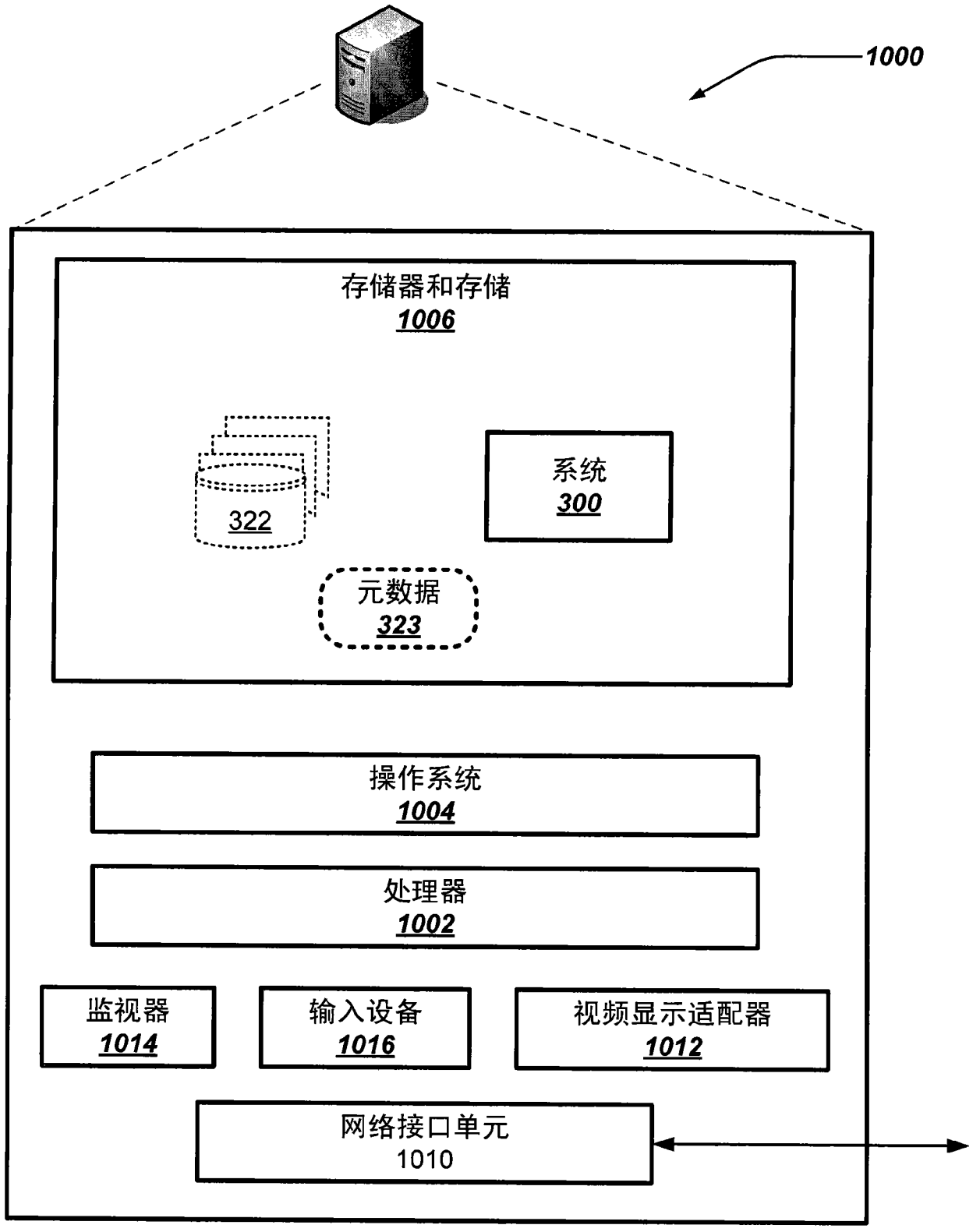

1000
存储器和存储
1006
322
系统
300
元数据
323
操作系统
1004
处理器
1002
监视器
1014
输入设备
1016
视频显示适配器
1012
网络接口单元
1010

图10