(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 108701254 B
(45)授权公告日 2022.11.18

(21)申请号 201780012417.1
(22)申请日 2017.08.22
(65)同一申请的已公布的文献号
申请公布号 CN 108701254 A
(43)申请公布日 2018.10.23
(30)优先权数据
62/378,143 2016.08.22 US
62/378,146 2016.08.22 US
62/378,147 2016.08.22 US
62/378,150 2016.08.22 US
62/378,151 2016.08.22 US
62/378,152 2016.08.22 US
(85)PCT国际申请进入国家阶段日
2018.08.21
(86)PCT国际申请的申请数据
PCT/US2017/048065 2017.08.22
(87)PCT国际申请的公布数据
WO2018/039266 EN 2018.03.01
(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 G·西萨拉曼
A·S·斯托贾诺维克
H·H·纳玛瓦尔 D·阿兰
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
专利代理师 刘玉洁
(51)Int.Cl.
G06N 5/02(2006.01)
G06F 9/50(2006.01)
G06F 3/06(2006.01)
审查员 胡晓雨

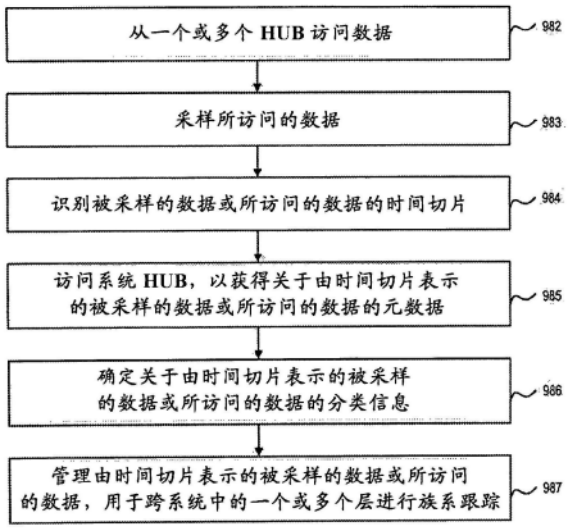
权利要求书3页 说明书45页 附图59页

(54)发明名称

用于动态族系跟踪、重建和生命周期管理的系统和方法

(57)摘要

根据各种实施例，本文描述的是用于与利用机器学习(ML、DataFlow机器学习、DFML)的数据集成或其它计算环境一起使用的、用于管理数据流(dataflow、DF)以及构建复杂的数据流软件应用(数据流应用，流水线)的系统(数据人工智能系统、数据AI系统)。根据实施例，该系统可以对于每个与特定时间快照相关的数据切片提供数据治理功能，诸如例如起源(特定数据来自哪里)、族系(如何获取/处理数据)、安全性(谁负责数据)、分类(数据关于什么)、影响(数据对业务有多大影响)、保留(数据应当存活多长时间)和有效性(是否应当排除/包括数据用于分析/处理)；然后可以将该数据治理功能用于制定生命周期决策和数据流推荐。



CN 108701254 B

1.一种用于与数据集成或其它计算环境一起使用的方法,包括:

在包括处理器的计算机处提供:

用于创建与软件应用相关联的数据流的图形用户界面,包括以下各项的规范:

一个或多个数据源,和

数据目标,

其中每个数据源包括具有属性、语义和与其他数据集的关系的一个或多个数据集,以及

其中事件协调器接收从所述一个或多个数据源接收的数据的通知以及与数据相关联的状态事务;

从存储简档信息和与数据源、数据集和实体相关联的其他元数据的知识源接收与处理与所述一个或多个数据源和数据目标相关联的数据流相关联的元数据;

经由边缘层从所述一个或多个数据源摄取数据并且将数据提供给可扩展输入/输出层,所述可扩展输入/输出层提供对结构化为主题的数据的访问;

将所摄取的数据写入到作为数据湖操作的数据储存库,以供执行数据流应用的计算层使用;

当数据从所述一个或多个数据源被接收并且由下游数据流应用使用时,识别所摄取的数据的时间切片;

访问所述知识源,以获得关于由所述时间切片表示的所摄取的数据的元数据;以及

管理由所述时间切片表示的数据,包括将所述时间切片写入所述数据湖,以及针对每个时间切片更新描述时间切片的族系的族系跟踪信息。

2.如权利要求1所述的方法,其中所述一个或多个数据源是HUB,并且所述知识源是系统HUB。

3.如权利要求2所述的方法,其中通过接口接收的元数据存储在所述系统HUB中,以供系统访问以用于处理数据流。

4.如权利要求1至3中任一项所述的方法,其中所述族系跟踪信息包括起源和族系信息。

5.如权利要求1至3中任一项所述的方法,其中系统提供图形用户界面,所述图形用户界面能够基于族系跟踪来指示数据流的生命周期,所述数据流的生命周期包括数据已在何处被处理。

6.如权利要求1至3中任一项所述的方法,其中所述方法在云或基于云的计算环境中执行。

7.一种与数据集成或其它计算环境一起使用的用于提供数据治理功能的系统,包括:

计算机,所述计算机包括一个或多个处理器,所述一个或多个处理器能够操作以:

提供用于创建与软件应用相关联的数据流的图形用户界面,包括一个或多个数据源和数据目标的规范,

其中每个数据源包括具有属性、语义和与其他数据集的关系的一个或多个数据集,以及

其中事件协调器接收从所述一个或多个数据源接收的数据的通知以及与数据相关联的状态事务;

知识源，所述知识源存储简档信息和与数据源、数据集和实体相关联的其他元数据，并且提供与处理与所述一个或多个数据源和数据目标相关联的数据流相关联的元数据；

其中所述系统操作以：

经由边缘层从所述一个或多个数据源摄取数据；

将数据提供给可扩展输入/输出层，所述可扩展输入/输出层提供对结构化为主题的数据的访问；

将所摄取的数据写入到作为数据湖操作的数据储存库，以供执行数据流应用的计算层使用；

当数据从所述一个或多个数据源被接收并且由下游数据流应用使用时，识别所摄取的数据的时间切片；

访问所述知识源，以获得关于由所述时间切片表示的所摄取的数据的元数据；以及

管理由所述时间切片表示的数据，包括将所述时间切片写入所述数据湖，以及针对每个时间切片更新描述时间切片的族系的族系跟踪信息。

8.如权利要求7所述的系统，其中所述一个或多个数据源是HUB，并且所述知识源是系统HUB。

9.如权利要求8所述的系统，其中通过接口接收的元数据存储在所述系统HUB中，以供系统访问以用于处理数据流。

10.如权利要求7至9中任一项所述的系统，其中所述族系跟踪信息包括起源和族系信息。

11.如权利要求7至9中任一项所述的系统，其中所述系统提供图形用户界面，所述图形用户界面能够基于族系跟踪指示数据流的生命周期，所述数据流的生命周期包括数据已在何处被处理。

12.如权利要求7至9中任一项所述的系统，其中所述系统在云或基于云的计算环境中提供。

13.一种非暂态计算机可读存储介质，包括存储在其上的指令，所述指令当由一个或多个计算机读取和执行时，使得所述一个或多个计算机执行包括以下操作的方法：

在包括处理器的计算机处提供：

用于创建与软件应用相关联的数据流的图形用户界面，包括一个或多个数据源和数据目标的规范：

其中每个数据源包括具有属性、语义和与其他数据集的关系的一个或多个数据集，以及

其中事件协调器接收从所述一个或多个数据源接收的数据的通知以及与数据相关联的状态事务；

从存储简档信息和与数据源、数据集和实体相关联的其他元数据的知识源接收与处理与所述一个或多个数据源和数据目标相关联的数据流相关联的元数据；

经由边缘层从所述一个或多个数据源摄取数据并且将数据提供给可扩展输入/输出层，所述可扩展输入/输出层提供对结构化为主题的数据的访问；

将所摄取的数据写入到作为数据湖操作的数据储存库，以供执行数据流应用的计算层使用；

当数据从所述一个或多个数据源被接收并且由下游数据流应用使用时，识别所摄取的数据的时间切片；

访问所述知识源，以获得关于由所述时间切片表示的所摄取的数据的元数据；以及

管理由所述时间切片表示的数据，包括将所述时间切片写入所述数据湖，以及针对每个时间切片更新描述时间切片的族系的族系跟踪信息。

14.如权利要求13所述的非暂态计算机可读存储介质，其中所述一个或多个数据源是HUB，并且所述知识源是系统HUB。

15.如权利要求14所述的非暂态计算机可读存储介质，其中通过接口接收的元数据存储在所述系统HUB中，以供系统访问以用于处理数据流。

16.如权利要求13至15中任一项所述的非暂态计算机可读存储介质，其中所述族系跟踪信息包括起源和族系信息。

17.如权利要求13至15中任一项所述的非暂态计算机可读存储介质，其中系统提供图形用户界面，所述图形用户界面能够基于族系跟踪来指示数据流的生命周期，所述数据流的生命周期包括数据已在何处被处理。

18.如权利要求13至15中任一项所述的非暂态计算机可读存储介质，其中所述方法在云或基于云的计算环境中执行。

19.一种包括用于执行如权利要求1至6中任一项所述的方法的部件的装置。

用于动态族系跟踪、重建和生命周期管理的系统和方法

[0001] 版权声明

[0002] 本专利文档的公开内容的一部分包含受版权保护的素材。版权拥有者不反对任何人对专利文档或专利公开内容按照在专利商标局的专利文件或记录中出现的那样进行传真复制，但是除此之外在任何情况下都保留所有版权。

[0003] 优先权要求：

[0004] 本申请要求于2016年8月22日提交的申请号为62/378,143，标题为“SYSTEM AND METHOD FOR AUTOMATED MAPPING OF DATA TYPES BETWEEN CLOUD AND DATABASE SERVICES”；于2016年8月22日提交的申请号为62/378,146，标题为“SYSTEM AND METHOD FOR DYNAMIC,INCREMENTAL RECOMMENDATIONS WITHIN REAL-TIME VISUAL SIMULATION”；于2016年8月22日提交的申请号为62/378,147，标题为“SYSTEM AND METHOD FOR INFERENCING OF DATA TRANSFORMATIONS THROUGH PATTERNDECOMPOSITION”；于2016年8月22日提交的申请号为62/378,150，标题为“SYSTEM AND METHOD FOR ONTOLOGY INDUCTION THROUGH STATISTICAL PROFILING AND REFERENCE SCHEMA MATCHING”；于2016年8月22日提交的申请号为62/378,151，标题为“SYSTEM AND METHOD FOR METADATA-DRIVEN EXTERNAL INTERFACE GENERATION OF APPLICATION PROGRAMMING INTERFACES”；以及于2016年8月22日提交的申请号为62/378,152，标题为“SYSTEM AND METHOD FOR DYNAMIC LINEAGE TRACKING AND RECONSTRUCTION OF COMPLEX BUSINESS ENTITIES WITH HIGH-LEVEL POLICIES”的美国临时专利申请的优先权权益；以上申请中的每一个均通过引用被并入本文。

技术领域

[0005] 本发明的实施例一般而言涉及集成从各个源获得的数据的方法，并且特别地涉及对于动态族系跟踪、重建和生命周期管理的支持。

背景技术

[0006] 当今的计算环境中的许多计算环境都需要能够在不同类型的软件应用之间共享大量数据。但是，由于例如分布式应用所支持的数据类型或者分布式应用的执行环境的差异，分布式应用在它们的配置方面可能明显不同。应用的配置可能取决于例如它的应用编程接口、运行时环境、部署方案、生命周期管理或安全管理。

[0007] 旨在用于开发此类分布式应用的软件设计工具往往是资源密集型的，通常需要人类域模型专家的服务来策划应用和数据集成。因此，面临构建将用于在不同类型的执行环境中集成不同类型的数据的复杂、可扩展、分布式应用的任务的应用开发人员通常必须花费大量的手动工作来设计、构建和配置这些应用。

发明内容

[0008] 根据各种实施例，本文描述的是用于与利用机器学习(ML、DataFlow机器学习、

DFML)的数据集成或其它计算环境一起使用、用于管理数据流(dataflow,DF)以及构建复杂的数据流软件应用(数据流应用、流水线)的系统(数据人工智能系统、数据AI系统)。根据实施例,该系统可以对于每个与特定时间快照相关的数据切片提供数据治理功能,诸如例如起源(特定数据来自哪里)、族系(如何获取/处理数据)、安全性(谁负责数据)、分类(数据关于什么)、影响(数据对业务有多大影响)、保留(数据应当存活多长时间)和有效性(是否应当排除/包括数据用于分析/处理);然后可以将该数据治理功能用于制定生命周期决策和数据流推荐。

附图说明

[0009] 图1示出了根据实施例的用于提供数据流人工智能的系统。

[0010] 图2示出了根据实施例的包括用于与系统一起使用的事件协调器的事件驱动的体系架构。

[0011] 图3示出了根据实施例的数据流中的步骤。

[0012] 图4示出了根据实施例的包括多个源的数据流的示例。

[0013] 图5示出了根据实施例的利用流水线的数据流的示例使用。

[0014] 图6示出了根据实施例的利用流水线的摄取/发布引擎和摄取/发布服务的示例使用。

[0015] 图7示出了根据实施例的从HUB摄取和训练的过程。

[0016] 图8示出了根据实施例的构建模型的过程。

[0017] 图9示出了根据实施例的对来自新添加的HUB的数据集或实体进行分类的过程。

[0018] 图10进一步示出了根据实施例的对来自新添加的HUB的数据集或实体进行分类的过程。

[0019] 图11进一步示出了根据实施例的对来自新添加的HUB的数据集或实体进行分类的过程。

[0020] 图12示出了根据实施例的用于功能类型分类的对象图。

[0021] 图13示出了根据实施例的维度功能类型分类的示例。

[0022] 图14示出了根据实施例的立方体功能类型分类的示例。

[0023] 图15示出了根据实施例的用于评估业务实体的功能类型的功能类型分类的示例用法。

[0024] 图16示出了根据实施例的用于功能变换的对象图。

[0025] 图17示出了根据实施例的推荐引擎的操作。

[0026] 图18示出了根据实施例的数据湖的使用。

[0027] 图19示出了根据实施例使用数据驱动策略来管理数据湖。

[0028] 图20示出了根据实施例使用过程驱动策略来管理数据湖。

[0029] 图21示出了根据实施例的流水线编译器的使用。

[0030] 图22示出了根据实施例的示例流水线图。

[0031] 图23示出了根据实施例的数据流水线的示例。

[0032] 图24示出了根据实施例的数据流水线的另一个示例。

[0033] 图25示出了根据实施例的编排流水线的示例。

[0034] 图26进一步示出了根据实施例的编排流水线的示例。

[0035] 图27示出了根据实施例的包括消息传送系统的协调架构的使用。

[0036] 图28进一步示出了根据实施例的包括消息传送系统的协调架构的使用。

[0037] 图29示出了根据实施例的与系统一起使用的内部部署(on-premise)代理。

[0038] 图30示出了根据实施例的数据流过程。

[0039] 图31示出了根据实施例的数据类型的自动映射。

[0040] 图32示出了根据实施例的用于生成映射的自动映射服务。

[0041] 图33示出了根据实施例的源模式和目标模式之间的映射的示例。

[0042] 图34示出了根据实施例的源模式和目标模式之间的映射的另一个示例。

[0043] 图35示出了根据实施例的用于提供数据类型的自动映射的过程。

[0044] 图36示出了根据实施例的显示针对所访问的数据启用的一个或多个语义动作的系统。

[0045] 图37示出了根据实施例的显示针对所访问的数据启用的一个或多个语义动作的图形用户界面。

[0046] 图38进一步示出了根据实施例的显示针对所访问的数据启用的一个或多个语义动作的图形用户界面。

[0047] 图39示出了根据实施例的用于显示针对所访问的数据启用的一个或多个语义动作的过程。

[0048] 图40示出了根据实施例的针对为一个或多个应用中的每个应用生成的一个或多个函数表达式识别数据流中的变换模式的部件。

[0049] 图41示出了根据实施例的针对一个或多个函数表达式识别数据流中的变换模式的示例。

[0050] 图42示出了根据实施例的用于针对为一个或多个应用中的每个应用生成的一个或多个函数表达式识别数据流中的变换模式的对象图。

[0051] 图43示出了根据实施例的针对为一个或多个应用中的每个应用生成的一个或多个函数表达式识别数据流中的变换模式的过程。

[0052] 图44示出了根据实施例的用于生成功能类型规则的系统。

[0053] 图45进一步示出了根据实施例的用于生成功能类型规则的系统。

[0054] 图46示出了根据实施例的用于生成功能类型规则的对象图。

[0055] 图47示出了根据实施例的用于基于生成的一个或多个规则来生成功能类型系统的过程。

[0056] 图48示出了根据实施例的用于基于经由外部功能接口提供的信息来识别用于在为数据流提供推荐中使用的模式的系统。

[0057] 图49进一步示出了根据实施例的基于经由外部功能接口提供的信息来识别用于在为数据流提供推荐中使用的模式。

[0058] 图50进一步示出了根据实施例的基于经由外部功能接口提供的信息来识别用于在为数据流提供推荐中使用的模式。

[0059] 图51示出了根据实施例的用于基于经由外部功能接口提供的信息来识别用于在为数据流提供推荐中使用的模式的过程。

[0060] 图52示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0061] 图53进一步示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0062] 图54进一步示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0063] 图55进一步示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0064] 图56进一步示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0065] 图57进一步示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0066] 图58进一步示出了根据实施例管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪。

[0067] 图59示出了根据实施例的用于管理采样数据或所访问的数据以用于跨一个或多个层进行族系跟踪的过程。

具体实施方式

[0068] 通过参考包括说明书、权利要求和附图的以下描述，前述内容以及其附加实施例和特征将变得明显。在以下描述中，出于解释的目的，阐述了具体细节以便提供对本发明的各种实施例的透彻理解。但是，将明显的是，可以在没有这些具体细节的情况下实践各种实施例。包括说明书、权利要求和附图的以下描述不旨在是限制性的。

[0069] 介绍

[0070] 根据各种实施例，本文描述的是用于与利用机器学习(ML、DataFlow机器学习、DFML)的数据集成或其它计算环境一起使用、用于在管理数据流(dataflow、DF)以及构建复杂的数据流软件应用(数据流应用、流水线)中使用的系统(数据人工智能系统、数据AI系统)。

[0071] 根据实施例，该系统可以提供对一个或多个数据源或数据目标(在本文中在一些实施例中称为HUB)之间的复杂数据结构、数据集或实体的自动映射的支持。自动映射可以由数据集的元数据、模式和统计剖析驱动；并且用于将与输入HUB相关联的源数据集或实体映射到目标数据集或实体，或反之亦然，以产生以一种格式或组织(投影)准备的、用于与一个或多个输出HUB一起使用的输出数据。

[0072] 根据实施例，该系统可以包括软件开发组件和图形用户界面，在本文中在一些实施例中称为流水线编辑器或Lambda Studio IDE，其提供与系统一起使用的可视环境，包括基于对与数据相关联的含义或语义的理解来提供用于对于针对从输入HUB访问的数据执行语义动作的实时建议。

[0073] 根据实施例，系统可以基于从用于软件应用的数据流的功能分解识别的模式来提供用于推荐对输入数据的动作和变换的服务，包括在后续应用中确定数据流的可能变换。可以将数据流分解为描述数据的变换、谓词和应用于数据的业务规则以及数据流中使用的

属性的模型。

[0074] 根据实施例，系统可以执行对模式定义的本体分析，以确定与该模式相关联的数据类型和数据集或实体；以及从参考模式生成或更新模型，该参考模式包括基于数据集或实体之间的关系及它们的属性而定义的本体。包括一个或多个模式的参考HUB可以被用于分析数据流，并进一步分类或提出推荐，诸如例如输入数据的变换的丰富、过滤或跨实体数据融合。

[0075] 根据实施例，系统提供程序接口，该程序接口在本文中在一些实施例中称为外部功能接口，通过该接口，用户或第三方可以以声明的方式基于功能和业务类型来定义服务、功能和业务类型、语义动作和模式或者预定义的复杂数据流，以扩展系统的功能。

[0076] 根据实施例，系统可以提供对于与特定时间快照有关的每个数据切片的数据治理(governance)功能，诸如例如起源(特定数据来自哪里)、族系(如何获取/处理数据)、安全性(谁负责数据)、分类(数据所关于的内容)、影响(数据对业务的影响有多少)、保留(数据应当存活多长时间)和有效性(数据是否应当被排除/包括在分析/处理中)；然后可以将其用于制定生命周期决策和数据流推荐。

[0077] 根据实施例，该系统可以被实现为服务，例如作为在基于云的计算环境内提供的云服务；并且可以用作与软件应用一起使用的数据的分析、设计、模拟、部署、开发和操作的单一控制点；包括启用来自一个或多个数据源(例如，在实施例中为输入HUB)的数据输入；提供图形用户界面，其使用户能够指定用于数据的应用；以及，取决于数据的预期目的地、使用或目标(例如，在实施例中为输出HUB)来缩放数据。

[0078] 根据实施例，如本文所使用的，当参考特定HUB使用时，术语“输入”和“输出”仅作为标记被提供，以反映特定用例或示例中的明显数据流，并且不旨在关于特定HUB的类型或功能是限制性的。

[0079] 例如，根据实施例，作为数据源操作的输入HUB还可以在相同的时间或在另一个时间作为输出HUB或目标操作，以接收相同的或另一个数据，并且反之亦然。

[0080] 此外，虽然为了说明的目的在本文描述的示例中的若干示例说明了输入HUB和输出HUB的使用；但是根据实施例，在实际实现中，数据集成或其它计算环境可以包括多个这种HUB，这些HUB中的至少一些HUB充当输入HUB和/或输出HUB两者。

[0081] 根据实施例，系统使得能够在大型(例如，基于云的)计算环境中快速开发软件应用，其中数据模型可以快速演化，并且其中诸如例如搜索、推荐或建议之类的特征是有价值的业务需求。在这种环境中，人工智能(AI)和语义搜索的组合使用户能够利用他们的现有系统完成更多任务。例如，可以基于对元数据、数据和用户与系统的交互的理解来推荐集成交互，诸如属性级映射。

[0082] 根据实施例，该系统还可以被用于建议复杂的情况，例如有趣的维度边缘，其可以被用于分析信息，并且使用户能够发现其数据内的迄今未知的事实。

[0083] 在一些实施例中，系统提供图形用户界面，该图形用户界面实现手动任务的自动化(例如，推荐或建议)，并利用机器学习和概率知识联合，以便为用户提供有用的上下文，并允许发现和语义驱动的解决方案，例如，数据仓库的创建、服务的扩展、数据准备和丰富以及软件应用的设计和监视。

[0084] 根据各种实施例，该系统可以包括或利用以下特征中的一些或全部：

[0085] 设计时(Design-Time)系统:根据实施例,一种计算环境,其实现软件应用(例如,数据流应用、流水线或Lambda应用)的设计、创建、监视和管理,包括对例如提供机器学习能力的数据AI子系统的使用。

[0086] 运行时系统:根据实施例,一种计算环境,其实现软件应用(例如,数据流应用、流水线或Lambda应用)的执行,并且从设计时系统接收输入以及向设计时系统提供推荐。

[0087] 流水线:根据实施例,一种定义处理流水线的声明性手段,具有多个阶段或语义动作,每个阶段或语义动作对应于功能,诸如例如输入数据的过滤、联接、丰富、变换或融合中的一个或多个,以用于准备作为输出数据。数据流软件应用或数据流应用表示例如DFML中的数据流。根据实施例,系统支持声明性流水线设计,该声明性流水线设计可以使用相同的代码库(例如,与Spark运行时平台一起)用于批量(历史)和实时(流式)数据处理两者;并且还支持构建可以对实时日期流运行的流水线或应用,以用于进行实时数据分析。由于流水线设计改变而导致的数据重新处理可以通过部署的流水线的滚动升级来处理。根据实施例,可以提供流水线作为Lambda应用,其可以适应不同批处理和实时层内的实时数据和批处理数据的处理。

[0088] HUB:根据实施例,包括数据集或实体的数据源或目标(云或内部部署的)。一种数据源,其可以被内省、可以从其消费数据或向其发布数据,并且包括具有属性、语义或者与其它数据集或实体的关系的数据集或实体。HUB的示例包括流传输数据、遥测、基于批处理的、结构化或非结构化的或其它类型的数据源。数据可以从一个HUB接收、与源数据集或实体相关联、并且映射到在相同HUB或另一个HUB处的目标数据集或实体。

[0089] 系统HUB:根据实施例,系统HUB可以作为知识源来操作,以存储简档信息和可以与其它HUB和这些HUB中的数据集或实体相关联的其它元数据,并且还可以以常规HUB的方式作为要处理的数据的源或接收者来操作。例如DFML中的中央储存库,在其中管理系统的状态和元数据。

[0090] 数据集(实体):根据实施例,可以由一个或多个HUB拥有或以其它方式与一个或多个HUB相关联的包括属性(例如,列)的数据结构,例如数据库表、视图、文件或API。根据实施例,一个或多个业务实体(例如客户记录),其可以用作语义业务类型,并且作为数据组件(例如,表)存储在HUB内。数据集或实体可以具有与其它数据集或实体;连同属性,诸如例如表中的列;以及数据类型,诸如例如字符串或整数的关系。根据实施例,系统支持在例如丰富、准备、变换、模型训练或评分操作期间对所有类型的数据(包括例如结构化、半结构化或非结构化数据)的模式无关的(agnostic)处理。

[0091] 数据AI子系统:根据实施例,系统(诸如例如数据AI系统)的组件,负责机器学习和语义相关功能,包括搜索、剖析、提供推荐引擎或支持自动映射中的一个或多个。数据AI子系统可以通过事件协调器支持设计时系统(例如软件开发组件,例如Lambda Studio)的操作,并且可以基于数据流应用(例如,流水线、Lambda应用)对数据的持续处理来提供推荐。例如,以推荐对现有的例如流水线的修改,以便更好地利用正在被处理的数据。数据AI子系统可以分析输入数据的量,并不断地更新域知识模型。在数据流应用(例如,流水线)的处理期间,例如流水线的每个阶段可以基于由数据AI子系统提供的推荐替代方案或选项、更新后的域模型以及来自用户的输入来继续,以便例如接受或拒绝推荐的语义动作。

[0092] 事件协调器:根据实施例,在设计时系统和运行时系统之间操作的事件驱动体系

架构(EDA)组件,以协调与数据流应用(例如,流水线、Lambda应用)的设计、创建、监视和管理相关的事件。例如,事件协调器可以接收来自HUB的数据(例如,符合已知数据类型的新数据)的已发布的通知、规范化来自该HUB的数据,并将规范化的数据提供给一组订户,以供例如流水线或其它下游消费者使用。事件协调器还可以接收系统内的状态事务的通知,以用于日志记录或族系跟踪,包括时间切片的创建;以及模式演进。

[0093] 剖析:根据实施例,提取来自HUB的数据的样本的操作,以便剖析由该HUB提供的数据,以及该HUB内的数据集或实体和属性;以及确定与采样HUB相关联的度量,以及更新与HUB相关联的元数据以反映该HUB中的数据的简档。

[0094] 软件开发组件(Lambda Studio):根据实施例,一种设计时系统工具,其提供图形用户界面,以使用户能够创建、监视和管理作为语义动作的流水线的Lambda应用或流水线的生命周期。允许用户设计例如流水线、Lambda应用的图形用户界面(UI、GUI)或工作室。

[0095] 语义动作:根据实施例,数据变换函数(例如关系代数运算)。可以由数据流应用(例如,流水线、Lambda应用)对HUB内的数据集或实体执行以用于投影到另一个实体上的动作。语义动作作为更高阶函数操作,该更高阶函数可以跨不同的模型或HUB使用,可以接收数据集输入并产生数据集输出。语义动作可以包括映射,并且可以响应于作为例如流水线或Lambda应用的一部分来处理数据而由例如数据AI子系统连续更新。

[0096] 映射:根据实施例,由例如数据AI子系统提供的第一(例如,源)数据集或实体与另一个(例如,目标)数据集或实体之间的语义动作的推荐映射,并且通过设计时系统(例如,经由软件开发组件、Lambda Studio)可访问。例如,数据AI子系统可以提供自动映射作为服务,其中自动映射可以通过数据集的元数据、模式和统计剖析基于与HUB或数据输入相关联的元数据的机器学习分析以及数据本身的剖析来驱动。

[0097] 模式:根据实施例,可以由数据流应用(例如,流水线、Lambda应用)执行的语义动作的模式。模板可以被用于提供可以由其它应用重用的模式的定义。表示通常与业务语义和流程相关联的数据的逻辑流和相关联的变换的数据流。

[0098] 策略:根据实施例,一组策略,其控制数据流应用(例如,流水线、Lambda应用)如何被调度、哪些用户或组件可以访问哪些HUB和语义动作、以及数据应当如何老化或者其它考虑因素。一种配置设置,定义例如流水线如何例如被调度、执行或访问。

[0099] 应用设计服务:根据实施例,向其它(例如,DFML)服务(例如,UI、系统外观)提供数据流,例如,特定于流水线、Lambda应用的服务,诸如例如验证、编译、打包、部署。设计时系统组件,其验证软件开发组件(例如,Lambda Studio)中的例如流水线或Lambda应用的流水线(例如,其输入和输出)、持久化流水线并控制流水线、Lambda应用到系统(例如,到Spark集群)的部署以供执行,并且此后可以用于管理应用的生命周期或状态。

[0100] 边缘层:根据实施例,收集数据并将数据转发到可扩展I/O层的层,例如,作为存储和转发层。一种运行时系统组件,其包括一个或多个节点,该一个或多个节点可以例如经由互联网可访问的网关接收数据,并且该运行时系统组件包括支持对例如数据AI系统的安全访问的安全性和其它特征。

[0101] 计算层:根据实施例,应用执行和数据处理层(例如,Spark)。运行时系统组件,作为分布式处理组件(例如,Spark云服务、计算节点的集群、虚拟机的集合或其它组件或节点)操作,以用于在例如流水线、Lambda应用的执行中使用。在多租户环境中,计算层内的节

点可以分配给租户,以供那些租户在执行流水线或Lambda应用中使用。

[0102] 可扩展输入/输出(I/O)层:根据实施例,提供被结构化为主题和分区的可扩展数据持久化和访问层(例如,Kafka)。运行时系统组件,其提供允许数据在系统内移动的、并在系统的各个组件之间共享的队列或其它逻辑存储(例如,Kafka环境)。在多租户环境中,可扩展的I/O层可以在多个租户之间共享。

[0103] 数据湖:根据实施例,用于持久化来自系统HUB或其它组件的信息的储存库。例如DFML中的数据的储存库,该数据通常由例如流水线、Lambda应用规范化或处理,并由其它流水线、Lambda应用或发布层消费。

[0104] 注册表:根据实施例,一个或多个信息储存库,例如用于存储功能和业务类型,用于将例如流水线、Lambda应用分解成它们的功能组件。

[0105] 数据流机器学习(DFML):根据实施例,数据集成、数据流管理系统,其利用机器学习(ML)来帮助构建复杂的数据流应用。

[0106] 元数据:根据实施例,底层定义、对数据集或实体以及属性及其关系的描述。它还可以是关于例如DFML中的工件的描述性数据。

[0107] 数据:根据实施例,由数据集或实体表示的应用数据。它们可以是批或流。例如,客户、订单或产品。

[0108] 系统外观:根据实施例,统一的API层,用于访问例如DFML事件驱动体系架构的功能能力。

[0109] 数据AI子系统:根据实施例,提供人工智能(AI)服务,包括但不限于例如搜索、自动映射、推荐或剖析。

[0110] 流传输实体:根据实施例,数据的连续输入以及近实时处理和输出要求,其可以支持对数据速度的强调。

[0111] 批处理实体:根据实施例,对数据的被调度的或应请求的摄取,其可以通过对量的强调来表征。

[0112] 数据切片:根据实施例,通常用时间标记的数据分区。

[0113] 规则:根据实施例,表示治理例如DFML中的工件的指令,例如数据规则、关系规则、元数据规则以及复杂或混合规则。

[0114] 推荐(数据AI):根据实施例,提议的动作过程,通常由一个或多个语义动作或细粒度指令表示,以帮助设计例如流水线、Lambda应用。

[0115] 搜索(数据AI):根据实施例,例如DFML中的语义搜索,其特征在于用户的上下文和意图以返回相关工件。

[0116] 自动映射(数据AI):根据实施例,一种将要在数据流中使用的候选源或目标数据集或实体列入候选的推荐的类型。

[0117] 数据剖析(数据AI):根据实施例,在属于数据集或实体的属性中表征数据的若干度量的集合,其中度量例如最小值、最大值、四分位数间距或稀疏度。

[0118] 动作参数:根据实施例,对于对其执行语义动作的数据集的引用。例如,在例如流水线、Lambda应用中等同联接(equi-join)的参数。

[0119] 外部功能接口:根据实施例,一种用于注册和调用服务(和语义动作)的机制,作为例如DFML Lambda应用框架的一部分。它可以用于扩展例如DFML中的能力或变换词汇表。

[0120] 服务:根据实施例,可以通过数据集成阶段(例如,准备、发现、变换或可视化)来表征的语义动作集合中的例如DFML中的拥有的工件。

[0121] 服务注册表:根据实施例,服务、它们的语义动作和其它实例信息的储存库。

[0122] 数据生命周期:根据实施例,在例如DFML内的数据使用中从摄取开始并在发布时结束的阶段。

[0123] 元数据收获:根据实施例,用于通常在HUB的注册之后进行剖析的元数据和样本数据的集合。

[0124] 规格化流水线:根据实施例,以某种格式对数据的标准化,以便于例如流水线、Lambda应用程序的消费。

[0125] 监视:根据实施例,识别、测量和评估例如流水线、Lambda应用的性能。

[0126] 摄取:根据实施例,在例如DFML中通过边缘层摄入数据。

[0127] 发布:根据实施例,从例如DFML将数据写入目标端点。

[0128] 数据AI系统

[0129] 图1示出了根据实施例的用于提供数据流人工智能的系统。

[0130] 如图1所示,根据实施例,系统(例如,数据AI系统150)可以提供用于处理和变换数据(诸如例如业务数据、消费者数据和企业数据)的一个或多个服务,包括使用机器学习处理,以用于各种计算资产,诸如例如数据库、云数据仓库、存储系统或存储服务。

[0131] 根据实施例,计算资产可以是基于云的、基于企业的或内部部署的(on-premise)或基于代理的。系统的各种元件可以通过一个或多个网络130连接。

[0132] 根据实施例,系统可以包括一个或多个输入HUB 110(例如,数据的源、数据源)和输出HUB 180(例如,数据的目标、数据目标)。

[0133] 根据实施例,每个输入HUB(例如,HUB 111)可以包括多个(源)数据集或实体192。

[0134] 根据实施例,输入HUB的示例可以包括数据库管理系统(DB、DBMS)112(例如,在线事务处理系统(OLTP)、业务智能系统或在线分析处理系统(OLAP))。在这种示例中,由诸如例如数据库管理系统之类的源提供的数据可以是结构化的或半结构化的。

[0135] 根据实施例,输入HUB的其它示例可以包括云存储库/对象存储库114(例如,AWS S3或另一对象存储库),其可以是具有非结构化数据的对象桶或点击流源;数据云116(例如,第三方云);流传输数据源118(例如,AWS Kinesis或另一流传输数据源)或其它输入源119。

[0136] 根据实施例,输入HUB可以包括数据源,从例如Oracle大数据准备(BDP)服务向该数据源中接收数据。

[0137] 根据实施例,系统可以包括一个或多个输出HUB 180(例如,输出目的地)。每个输出HUB(例如,HUB 181)可以包括多个(目标)数据集或实体194。

[0138] 根据实施例,输出HUB的示例可以包括公共云182、数据云184(例如,AWS和Azure)、内部部署云186或其它输出目标187。由系统提供的数据输出可以为在输出HUB处可访问的数据流应用(例如,流水线、Lambda应用)产生。

[0139] 根据实施例,公共云的示例可以是例如Oracle公共云,其包括诸如例如大数据准备云服务、Exadata云服务、大数据发现云服务和业务智能云服务之类的服务。

[0140] 根据实施例,系统可以被实现为用于流传输和按需(批量)数据处理的统一平台,

其中流传输和按需(批量)数据处理作为服务(例如,作为软件即服务)交付给用户,从而提供用于多个输入HUB的可扩展、多租户数据处理。作为服务的一部分,使用机器学习技术以及由图形用户界面提供的可视见解和监视来实时地分析数据。数据集可以从多个输入HUB融合,以输出到输出HUB。例如,通过由系统提供的数据处理服务,可以为一个或多个输出HUB中的数据仓库和群体生成数据。

[0141] 根据实施例,该系统提供用于数据的变换、丰富、路由、分类和混合的声明性和编程拓扑;并且可以包括设计时系统160和运行时系统170。用户可以创建被设计为执行数据处理的应用,诸如例如数据流应用(例如,流水线、Lambda应用)190。

[0142] 根据实施例,设计时系统可以使用户能够设计数据流应用、定义数据流、以及定义用于数据流处理的数据。例如,设计时系统可以提供软件开发组件162(在本文中在实施例中被称为Lambda Studio),软件开发组件162提供用于创建数据流应用的图形用户界面。

[0143] 例如,根据实施例,通过使用软件开发组件,用户可以指定输入HUB和输出HUB,以用于为应用创建数据流。图形用户界面可以呈现用于数据集成的服务的接口,其使得用户能够创建、操纵和管理用于应用的数据流,包括动态地监视和管理数据流流水线的能力,诸如例如查看数据族系(lineage)和执行取证分析。

[0144] 根据实施例,设计时系统还可以包括用于将数据流应用部署到运行时系统中的应用设计服务164。

[0145] 根据实施例,设计时系统还可以包括一个或多个系统HUB 166(例如,元数据储存库),用于存储用于处理数据流的元数据。一个或多个系统HUB可以存储数据的样本,诸如例如包括功能数据类型和业务数据类型的数据类型。系统HUB中的信息可以被用于执行本文公开的技术中的一种或多种技术。数据湖167组件可以作为用于持久化来自系统HUB的信息的储存库。

[0146] 根据实施例,设计时系统还可以包括数据人工智能(AI)子系统168,以执行用于数据人工智能处理的操作。操作可以包括使用ML技术,例如搜索和检索。数据AI子系统可以对数据进行采样以生成系统HUB的元数据。

[0147] 根据实施例,对于每个输入HUB,数据AI子系统可以执行模式对象分析、元数据分析、采样数据、相关性分析和分类分析。数据AI子系统可以通过连续地对作为输入的数据运行来向数据流应用提供丰富的数据;并且可以向例如流水线、Lambda应用提供推荐、见解和类型归纳。

[0148] 根据实施例,设计时系统使用户能够创建定义用例的功能需求的策略、工件(artifact)和流程。

[0149] 例如,根据实施例,设计时系统可以提供图形用户界面以创建用于摄取数据的HUB并定义摄取策略,该摄取策略可以基于时间或者根据相关数据流的需要。在选择输入HUB时,可以从输入HUB对数据进行采样以对源进行剖析,诸如例如执行元数据查询、获得样本以及获得用户定义的输入。简档可以存储在系统HUB中。图形用户界面使得能够联接(join)多个源以定义数据流流水线。这可以通过创建脚本或通过使用引导式编辑器来完成,通过该编辑器在每个步骤中数据可以被可视化。图形用户界面可以提供对推荐服务的访问,该推荐服务建议可以如何对数据进行例如修正、丰富或联接。

[0150] 根据实施例,在设计时期间,应用设计服务可以建议合适的结构来分析得到的内

容。应用设计服务可以使用知识服务(功能类型分类)来建议量度和相关的维度层次结构。一旦这已完成,设计时系统就可以推荐从早期流水线获取混合数据所需的数据流并填充维度目标结构。基于依赖性分析,它还可以导出并生成编排流,以加载/刷新目标模式。对于正向工程用例,设计时系统还可以生成用于托管目标结构的HUB并创建目标模式。

[0151] 根据实施例,运行时系统可以在用于数据处理的服务的运行时期间执行处理。

[0152] 根据实施例,在运行时或操作模式下,应用和/或执行由用户创建的策略和流定义。例如,这种处理可以包括调用摄取、变换、建模和发布服务,以处理流水线中的数据。

[0153] 根据实施例,运行时系统可以包括边缘层172、可扩展输入/输出(I/O)层174、以及分布式处理系统或计算层176。在运行时期间(例如,当从一个或多个输入HUB 110摄取数据时),对于导致数据被生成的事件,数据可以由边缘层接收。

[0154] 根据实施例,事件协调器165在设计时系统和运行时系统之间操作,以协调与数据流应用(例如,流水线、Lambda应用)的设计、创建、监视和管理相关的事件。

[0155] 根据实施例,边缘层将数据发送到可扩展I/O层,以用于将数据路由到分布式处理系统或计算层。

[0156] 根据实施例,分布式处理系统或计算层可以实现(以每租户为基础的)流水线处理以处理数据以供输出。分布式处理系统可以使用例如Apache Spark和Alluxio来实现;并且可以在将数据输出到输出HUB之前将数据采样到数据湖中。分布式处理系统可以与可扩展I/O层通信,以激活用于处理的数据。

[0157] 根据实施例,包括上述组件中的一些或全部的数据AI系统可以在一个或多个计算机处提供或由一个或多个计算机执行,该一个或多个计算机包括例如一个或多个处理器(CPU)、存储器和持久存储设备(198)。

[0158] 事件驱动的体系架构

[0159] 如前所述,根据实施例,系统可以包括在设计时系统和运行时系统之间操作的事件驱动的体系架构(EDA)组件或事件协调器,以协调与数据流应用(例如,流水线、Lambda应用)的设计、创建、监视和管理相关的事件。

[0160] 图2示出了根据实施例的包括用于与系统一起使用的事件协调器的事件驱动的体系架构。

[0161] 如图2中所示,根据实施例,事件协调器可以包括事件队列202(例如,Kafka)、事件引导程序(bootstrapper)服务204(例如,ExecutorService)、以及事件配置发布者/事件消费者206(例如,

[0162] DBCS)。

[0163] 根据实施例,在系统外观(facade)208(例如,事件API扩展)处接收的事件可以由一个或多个事件中介210(例如,(一个或多个)Kafka消费者)传送到系统的各个组件。例如,可以经由事件协调器将数据和/或事件(诸如例如外部数据212(例如,S3、OSCS或OGG数据)或来自图形用户界面214(例如,浏览器或DFML UI)的输入)传送给其它部件(诸如例如,如前所述的应用运行时216、数据湖、系统HUB、数据AI子系统、应用设计服务);和/或传送给摄取220、发布230、调度240或其它组件。

[0164] 根据实施例,事件中介可以被配置为流事件的消费者。事件引导程序可以起动多个已配置的事件中介,以代表已注册的订户处理事件;其中每个事件中介将事件的处理委

托给已注册的回调端点,以处理给定事件。事件协调器使得能够注册事件类型;注册事件实体;注册事件;以及注册订户。表1提供了包括发布事件和订阅事件的各种事件对象的示例。

[0165]

发布事件	2 元组（事件类型，事件实体）对象，它注册通过事件实体对事件的发布。
订阅事件	2 元组（事件，事件实体）对象，它注册对由另一个事件实体（发布者）发布的事件的订阅。

[0166] 表1

[0167] 事件类型

[0168] 根据实施例,事件类型定义对于系统重要的事件的状态改变,诸如例如HUB的创建、数据流应用(例如,流水线、Lambda应用)的修改、摄取用于数据集或实体的数据、或将数据发布到目标HUB。在下文和表2中描述示例数据格式和各种事件类型的示例:

[0169] {

[0170] "Id":"",//在创建时生成

[0171] "Name":"Hub-Creation"

[0172] "Type":"User Event"

[0173] }

[0174]

POST /eventTypes	创建新 eventType。
PUT /eventType/{eventide}	修改具有给定 Id 的 eventType
GET /eventTypes	检索系统中的所有 eventType
GET /eventTypes/{eventTypeId}	检索具有给定 Id 的 eventType 的表示
GET /eventTypes/ {eventide}/publishers	检索用于这个 eventType 的所有发布者。可以存在被注册以发布这种类型的事件的多于一个事件实体，在这种情况下，将返回发布这种类型的事件的所有不同事件实体。
GET /eventTypes/{eventide}/subscribers	检索用于这种 eventType 的所有订户。可以存在注册为这种类型的事件的订户的多于一个事件实体，在这种情况下，将返回订阅这种类型的事件的所有不同的事件实体。
DELETE /eventTypes/{eventide}	删除具有给定 ID 的事件。

[0175] 表2

[0176] 事件(eventing)实体

[0177] 根据实施例,事件实体可以是事件的发布者和/或订户。例如,事件实体可以注册以发布一个或多个事件和/或是一个或多个事件的消费者,包括注册端点或回调URL,该端

点或回调URL将被用于在发布时通知或发送确认以及委托针对订阅事件的处理。事件实体的示例可以包括元数据服务、摄取服务、系统HUB工件和流水线、Lambda应用。在下文和表3中描述示例数据格式和各种事件实体的示例：

[0178] {

[0179] "Id":"",//在创建时生成

[0180] "Name":"Metadata Service",

[0181] "endPointURL":"localhost:9010/processEvent",

[0182] "entityType":"DFMLService"

[0183] }

[0184]

POST /eventingEntities	创建新的事件实体
PUT /eventingEntities/{entityId}	修改由 Id 识别的现有事件实体
GET /eventingEntities	检索所有已注册的事件实体
GET /eventingEntities/{entityId}	检索具有给定 Id 的事件实体的表示
GET /eventingEntities/{entityId}/eventsPublished	检索由这个事件实体针对发布注册的所有事件
GET /eventingEntities/{entityId}/eventsSubscribed	检索由这个事件实体针对订阅注册的所有事件
DELETE /eventingEntities/{entityId}	从系统中删除这个事件实体

[0185] 表3

[0186] 事件

[0187] 根据实施例，事件是与注册为发布者的事件实体相关联的事件类型的实例；并且可以拥有订户(事件实体)。例如，元数据服务可以注册HUB创建事件以供发布；并且可以为这个事件发布一个或多个事件实例(一旦用于HUB的每个事件实例被创建)。表4中描述了各种事件的示例：

[0188]

POST /events	创建要由事件实体发布的新事件

[0189]

	{ "Id":"" , "acknowledgeURL": "locahost:9010/eventAcknowledge", "onProcessingOf":"/eventType/{eventId} ", "eventType":"/eventType/{eventId}", "eventingEntity":"/eventingEntity/{entitId} " }
PUT /events/{eventid}	修改由事件实体针对发布注册的现有事件
POST /events/{eventid}/publish	发布将事件排入队列以供消费的事件实例 { "event_type": "data", "subtype": "publication", "state": "ready", "context": {“eventContextId”:”{eventId}”,”accessToken”:””} "message": { _actual event data goes here_ } }
GET /events	检索系统中注册的所有事件
GET /events/{eventide}	检索具有给定 Id 的事件
GET /events/{eventide}/publisher	检索这个事件的发布者
GET /events/{eventide}/subscribers	检索这个事件的订户。
POST /events/{eventid}/subscribers	注册针对事件的订户。

[0190]

	{ "Id":"" ，//在创建时生成 "processingURL": "locahost:9010/eventProcess", "subscribingEvent":"/events/{eventId}", "callbackMethod":"REST", "subscriberentity":"/eventingEntity/{entitId}" }
PUT /events/{eventid}/subscribers/{subscriberId}	修改对于这个事件的订户的特性
DELETE /events/{eventid}/subscribers/{subscriberId}	删除对于这个事件的订户
DELETE /events/{eventide}	删除事件

[0191] 表4

[0192] 示例

[0193] 根据实施例,以下示例示出了创建事件类型;注册发布事件;注册订户;发布事件;获得事件类型;获得事件类型的发布者;以及获得事件类型的订户。

[0194] POST http://den00tnk:9021/dfml/service/eventType

[0195] {"name":"DATA_INGESTED",

[0196] "type":"SystemEvent"

[0197] }

[0198] 其返回全局唯一的ID(UUID),例如“8e87039b-a8b7-4512-862c-fdb05b9b8888”。事件对象可以发布或订阅系统中的事件。服务端点(诸如例如摄取服务、元数据服务和应用设计服务)可以使用静态端点发布和订阅事件,以进行确认、通知、错误或处理。DFML工件(例如,DFMLEntity、DFMLLambdaApp、DFMLHub)也可以被注册为事件对象;这些类型的实例可以发布和订阅事件,而无需注册为事件对象。

[0199] POST http://den00tnk:9021/dfml/service/eventEntity

[0200] {

[0201] "name":"DFMLEntity",

[0202] "endPointURL":"localhost:9010/<publisherURL>",

[0203] "notificationEndPointURL":"http://den00tnk:9021/<publisherURL>/notification",

[0204] "exceptionEndPointURL":"http://den00tnk:9021/<publisherURL>/

exception",

[0205] "errorEndPointURL":"http://den00tnk:9021/<publisherURL>/error",

[0206] "entityType":"DFMLEntity"

[0207] }

[0208] 以下示例将DFMLLambdaApps(类型)注册为事件对象。

[0209] {

[0210] "name":"DFMLLambdaApps",

[0211] "endPointURL":"localhost:9010/<publisherURL>",

[0212] "notificationEndPointURL":"http://den00tnk:9021/<publisherURL>

[0213] /notification",

[0214] "exceptionEndPointURL":"http://den00tnk:9021/<publisherURL>/exception",

[0215] "errorEndPointURL":"http://den00tnk:9021/<publisherURL>/error",

[0216] "entityType":"DFMLLambdaApps"

[0217] }

[0218] 对于事件类型为HUB、Entity和LambdaApp的实体,<publisherURL>可以在REST端点URL中注释,并且事件驱动的体系架构将通过替换DFML工件实例URL来导出实际的URL。例如,如果notificationEndpointURL被注册为http://den00tnk:9021/<publisherURL>/notification,并且被指定为消息的一部分的发布者URL是hubs/1234/entities/3456,那么为通知而调用的URL将是http://den00tnk:9021/hubs/1234/entities/3456/notification。POST返回UUID;例如“185cb819-7599-475b-99a7-65e0bd2ab947”。

[0219] 注册发布事件

[0220] 根据实施例,发布事件可以被注册为:

[0221] POST http://den00tnk:9021/dfml/service/event

[0222] {

[0223] "acknowledgeURL":"http://den00tnk:9021/<publisherURL>/acknowledge",

[0224] "onProcessingOf":"/eventType/{eventId}",

[0225] "eventType":"7ea92c868e87039b-a8b7-4512-862c-fdb05b9b8888",

[0226] "publishingEntity":"185cb819-7599-475b-99a7-65e0bd2ab947"

[0227] }

[0228] eventType指的是针对注册事件类型DATA_INGESTED而返回的UUID,并且publishEntity指的是被注册为事件对象的DFMLEntity类型。注册返回UUID,例如“2c7a4b6f-73ba-4247-a07a-806ef659def5”。

[0229] 注册订户

[0230] 根据实施例,订户可以被注册为:

[0231] POST http://den00tnk:9021/dfml/service/event/2c7a4b6f-73ba-4247-a07a-806ef659def5/subscribers

[0232] 从发布事件注册返回的UUID在用于注册订户的路径片段中使用。

[0233] {

[0234] "processingURL":

[0235] "http://den00tnk:9021/dfml/service/eventType/process3",

[0236] "callbackMethod":"SYNC_POST",

[0237] "subscriberEntity":"7599916b-baab-409c-bfe0-5334f111ef41",

[0238] "publisherURL":"/hubs/1234/entities/3456",

[0239] "publishingObjectType":"DFMLEntity",

[0240] "subscribingObjectType":"DFMLLambdaApps",

[0241] "subscriberURL":"/lambdaApps/123456"

[0242] }

[0243] publisherURL和publishingObjectType指的是发布者对象的实例和类型。在这里,指定URI/lambdaApps/123456的数据流(例如,Lambda)应用对于订阅来自实体/hubs/1234/entities/3456的DATA_INGESTED事件有兴趣。该注册返回UUID,例如“1d542da1-e18e-4590-82c0-7fe1c55c5bc8”。

[0244] 发布事件

[0245] 根据实施例,事件可以被发布为:

```
POST http://den00tnk:9021/dfml/service/event/publish

{
"event_type":"DATA_AVAILABLE",
"subtype":"publication",
"state":"ready",
```

[0246]

```
"eventTypeId":"2c7a4b6f-73ba-4247-a07a-806ef659def5",
"publisherURL":"dfml/service/eventType",
"message": {"id":"1234",
    "descr":"something happened here testing this and this
  again"
}
}
```

[0247] 如果发布对象是DFMLEntity、DFMLHub或DFMLLambdaApps之一,那么使用publisherURL,并且publisherURL被用于检查发布订户所登记的消息的事件对象的实例。发布者URL还被用于在订户成功处理消息时导出通知URL。该发布返回作为已发布事件的一部分的消息主体。

[0248] 获得事件类型

[0249] 根据实施例,事件类型可以被确定为:

[0250]
```
GET  http://den00tnk:9021/dfml/service/eventType
```

[0251]


```
{
"eventTypes": [
    {
"Id": "8e87039b-a8b7-4512-862c-fdb05b9b8888",
"name": "DATA_INGESTED",
"type": "SystemEvent",
"createdBy": "",
"updatedBy": "",
"description": "",
"typeQualifier": "",
"resourceType": "",
"verb": "",
"operationType": "",
"status": "",
"annotation": ""
    },
    {
"Id": "7ea92c86-8db5-42d6-992a-2578a6d025ce",
"name": "DATA_AVAILABLE",
"type": "SystemEvent",
"createdBy": "",
"updatedBy": "",
"description": "",
"typeQualifier": "",
"resourceType": "",
"verb": "",
"operationType": "",
"status": "",
"annotation": ""
    }
```

```
  ]
}
```

[0252] 获得事件类型的发布者

```
GET     http://den00tnk:9021/dfml/service/eventType/7ea92c86-
8db5-42d6-992a-2578a6d025ce/publishers

{
"eventingObjects": [
   {
"Id": "185cb819-7599-475b-99a7-65e0bd2ab947",
"name": "DFMLEntity",
"entityType": "DFMLEntity",
"endPointURL": "localhost:9010/<publisherURL>",
"notificationEndpointURL":
"http://den00tnk:9021/<publisherURL>/notification",
"exceptionEndpointURL":
"http://den00tnk:9021/<publisherURL>/exception",
"errorEndpointURL":
"http://den00tnk:9021/<publisherURL>/error",
"acknowledgeEndpointURL": "",
"description": "",
"entityQualifier": "",
"status": "",
"annotation": ""
   }
  ]
}
```

[0253]

[0254] 获得事件类型的订户

[0255] 根据实施例,可以将用于事件类型的订户确定为:

```
GET      http://den00tnk:9021/dfml/service/eventType/7ea92c86-
8db5-42d6-992a-2578a6d025ce/subscribers


{
"eventingObjects": [
   {
"Id": "7599916b-baab-409c-bfe0-5334f111ef41",
"name": "DFMLLambdaApps",
"entityType": "DFMLLambdaApps",
"endPointURL": "localhost:9010/<publisherURL>",
"notificationEndpointURL":
"http://den00tnk:9021/<publisherURL>/notification",
"exceptionEndpointURL":
"http://den00tnk:9021/<publisherURL>/exception",
"errorEndpointURL":
"http://den00tnk:9021/<publisherURL>/error",
"acknowledgeEndpointURL": "",
"description": "",
"entityQualifier": "",
"status": "",
"annotation": ""
   }
  ]
}
```

[0256]

[0257] 通过示例的方式提供上述说明，以示出事件协调器、事件类型、事件实体和事件的特定实施例。根据其它实施例，其它类型的EDA可以被用于提供在设计时系统和运行时系统之间操作的系统内的通信，以协调与数据流应用的设计、创建、监视和管理相关的事件，并且可以支持其它类型的事件类型、事件实体和事件。

[0258] 数据流机器学习(DFML)流程

[0259] 如前所述，根据各种实施例，系统可以与利用机器学习(ML、数据流机器学习、DFML)的数据集成或其它计算环境一起使用，以用于管理数据流(dataflow、DF)和构建复杂的数据流软件应用(例如，数据流应用、流水线、Lambda应用)。

[0260] 图3示出了根据实施例的数据流中的步骤。

[0261] 如图3中所示，根据实施例，DFML数据流260的处理可以包括多个步骤，包括摄取步骤262，在摄取步骤262期间，从各种源(例如，Salesforce(SFDC)、S3或DBaaS)摄取数据。

[0262] 在数据准备步骤264期间，可以例如通过去重复、标准化或丰富化来准备被摄取的数据。

[0263] 在变换步骤266期间，系统可以在数据集处执行一个或多个合并、过滤或查找，以变换数据。

[0264] 在模型步骤268期间，生成一个或多个模型，连同到模型的映射。

[0265] 在发布步骤270期间，系统可以发布模型、指定策略和调度，以及填充目标数据结构。

[0266] 根据实施例，系统支持贯穿其数据准备、变换和模型步骤中的每个步骤使用搜索/推荐功能272。用户可以通过一组良好定义的服务与系统交互，该一组良好定义的服务封装了数据集成框架中的功能能力的广度。这组服务定义了系统的逻辑视图。例如，在设计模式中，用户可以创建定义特定用例的功能需求的策略、工件和流程。

[0267] 图4示出了根据实施例的包括多个源的数据流的示例。

[0268] 如图4中所示的示例数据流280所示，根据实施例，要求是取得来自多个源282(这里指示为SFDC和FACS(融合应用云服务，Fusion Apps Cloud Service))的内容连同OSCS(Oracle存储云服务，Oracle Storage Cloud Service)中的一些文件；以可用于分析期望内容的方式将该信息混合在一起；导出目标立方体(cube)和维度；将混合的内容映射到目标结构；并且使这个内容连同维度模型一起可用于Oracle业务智能云服务(BICS)环境；包括使用摄取、变换266A/266B、模型、编排292、和部署294步骤。

[0269] 提供所示示例以示出本文描述的技术；本文描述的功能不限于与这些特定数据源一起使用。

[0270] 根据实施例，在摄取步骤期间，为了访问和摄取SFDC内容，在数据湖中创建HUB以接收该内容。这可以例如通过为相关访问模式(JDBC、REST、SOAP)选择SFDC适配器、创建HUB、提供名称以及定义摄取策略来执行，该摄取策略可以基于时间或者根据相关数据流的需要。

[0271] 根据实施例，可以对其它两个源执行类似的过程，不同之处在于，对于OSCS源，在开始时可能不知道模式，因此该模式可以代替地通过某种手段(例如，元数据查询、采样或用户定义)获得。

[0272] 根据实施例，可以可选地对数据源进行剖析以进一步研究源，这可以有助于稍后在集成流程中导出推荐。

[0273] 根据实施例，下一步是定义可以如何将单独的源围绕中心项联接在一起，这通常是分析的基础(事实)，并且这可以通过定义数据流流水线来实现。这可以通过创建流水线域特定语言(DSL)脚本来直接完成，或者使用引导式编辑器完成，在引导式编辑器中用户可以看到在每个步骤中对数据的影响并可以利用建议数据可以如何被例如校正、丰富、联接的推荐服务。

[0274] 此时，用户可以请求系统建议合适的结构来分析所得到的内容。例如，根据实施例，系统可以使用知识服务(功能类型分类)来建议量度(measure)和相关的维度层次结构。一旦这已完成，系统就可以推荐从早期流水线获取混合数据所需的数据流，并填充维度目

标结构。基于依赖性分析，它还将导出并生成用于加载/刷新目标模式的编排流程。

[0275] 根据实施例，系统现在可以生成HUB以托管目标结构并经由适配器将该HUB与DBCS相关联，该DBCS生成创建目标模式所需的数据定义语言(DDL)，并且例如部署XDML或BICS可用于生成访问新创建的模式所需的模型的任何形式。这可以通过执行编排流程并触发排出(exhaust)服务来填充。

[0276] 图5示出了根据实施例的利用流水线的数据流的示例使用。

[0277] 如图5所示，根据实施例，系统允许用户定义代表数据流的流水线302，在这个示例中流水线302包括流水线步骤S1至S5，以描述作为应用被构建/执行304时的数据处理。

[0278] 例如，根据实施例，用户可以调用摄取、变换、模型和发布服务，或诸如例如策略306、执行310或持久化服务312之类的其它服务，以处理流水线中的数据。用户还可以定义解决方案(即，控制流程)以指定可以将相关流水线集成在一起的统一流程。通常，解决方案对完整的用例进行建模，例如销售立方体和关联的维度的加载。

[0279] 数据AI系统组件

[0280] 根据实施例，适配器使得能够与各种端点连接并从各个端点摄取数据，并且适配器特定于应用或源类型。

[0281] 根据实施例，系统可以包括预定义的一组适配器，该一组适配器中的一些适配器可以利用其它SOA适配器，并允许将附加的适配器注册到框架。对于给定的连接类型，可以有多于一个适配器；在这种情况下，摄取引擎将基于HUB的连接类型配置来选择最适合的适配器。

[0282] 图6示出了根据实施例的利用流水线的摄取/发布引擎和摄取/发布服务的示例使用。

[0283] 如图6所示，根据实施例，流水线334可以经由摄取/发布服务332访问摄取/发布引擎330，在这个示例中，流水线334被设计为从输入HUB(例如，SFDC HUB1)摄取数据336(例如，销售数据)、变换摄取的数据338、并将数据发布到输出HUB 340(例如，Oracle HUB)。

[0284] 根据实施例，摄取/发布引擎支持多个连接类型331，该多个连接类型中的每个连接类型342与提供对HUB的访问的一个或多个适配器344相关联。

[0285] 例如，如图6的示例所示，根据实施例，SFDC连接类型352可以与提供对SFDC HUB 358、359的访问的SFDC-Adp1适配器354和SFDC-Adp2适配器356相关联；而ExDaaS连接类型362可以与提供对ExDaas HUB 366的访问的ExDaaS-Adp适配器364相关联；并且Oracle连接类型372可以与提供对Oracle HUB 376的访问的Oracle Adp适配器374相关联。

[0286] 推荐引擎

[0287] 根据实施例，系统可以包括作为专家过滤系统操作的推荐引擎或知识服务，该专家过滤系统从可以对数据执行的若干可能动作中预测/建议最相关的动作。

[0288] 根据实施例，推荐可以被链接以便于用户逐步通过它们，以实现规定的最终目标。例如，推荐引擎可以引导用户经过将数据集转换为数据立方体以发布到目标BI系统的一组步骤。

[0289] 根据实施例，推荐引擎利用三个方面：(A)业务类型分类，(B)功能类型分类，以及(C)知识库。数据集或实体上的本体管理和查询/搜索功能可以通过例如具有查询API、MRS和审计存储库的YAGO3播种的联合本体来提供。可以通过例如基于ML流水线的分类来提供

业务实体分类以识别业务类型。功能类型分类可以通过例如归纳和基于规则的功能类型分类来提供。动作推荐可以通过例如归纳和基于规则的数据准备、变换、模型、依赖性和相关推荐来提供。

[0290] 分类服务

[0291] 根据实施例，系统提供分类服务，该分类服务可以被分类为业务类型分类和功能类型分类，下面进一步描述每个分类服务。

[0292] 业务类型分类

[0293] 根据实施例，实体的业务类型是其表现型(phenotype)。在识别实体的业务类型时，实体中各个属性的可观察特点与定义同样重要。虽然分类算法使用数据集或实体的示意性定义，但它还可以利用使用数据构建的模型来对数据集或实体业务类型进行分类。

[0294] 例如，根据实施例，从HUB摄取的数据集可以被分类为系统已知的现有业务类型(从主HUB播种(seed))之一，或者如果它不能归类为现有类型，那么可以作为新类型被添加。

[0295] 根据实施例，业务类型分类用于基于(从流水线中的关于类似业务类型定义的变换)归纳推理或者基于从分类根实体导出的简单提议来做出推荐。

[0296] 一般而言，根据实施例，以下一组步骤描述了分类过程：从主(训练)hub(中枢)摄取和播种；建立模型并计算列统计数据并将它们注册用于分类；对来自新添加的hub的数据集或实体进行分类，包括创建简档/计算列统计数据；并对数据集或实体进行分类，以提供基于结构和列统计数据的实体模型的简短列表；以及对数据集或实体进行分类，包括多类分类以使用模型进行计算/预测。

[0297] 图7示出了根据实施例的从HUB摄取和训练的过程。

[0298] 如图7所示，根据实施例，来自HUB 382(例如，在这个示例中，RelatedIQ源)的数据可以由推荐引擎380读取，以作为数据集390(例如，作为弹性分布数据集，RDD)，数据集390在这个示例中包括帐号数据集391、事件数据集392、联系人数据集393、列表数据集394、用户数据集395。

[0299] 根据实施例，多个类型分类工具400可以与ML流水线402一起使用，例如GraphX 404、Wolfram/Yago 406和/或MLlib Statistics 408被用于在HUB首次注册时利用实体元数据(训练或播种数据)对知识图440进行播种。

[0300] 根据实施例，数据集或实体元数据和数据从源HUB摄取并存储在数据湖中。在模型生成410期间，例如通过FP-growth逻辑回归412，在生成模型420和表示所有数据集或实体(在这个示例中表示事件422、帐户424、联系人426和用户428)的知识图时，使用实体元数据(属性和与其它实体的关系)。作为播种的一部分，使用数据集或实体数据来构建回归模型，并计算属性统计数据(最小值、最大值、平均值或概率密度)。

[0301] 图8示出了根据实施例的构建模型的过程。

[0302] 如图8中所示，根据实施例，当在例如Spark环境430中运行时，Spark MLlib统计数据可以被用来计算作为知识图中的属性特性被添加的列统计数据。计算出的列统计数据以及其它数据集或实体元数据可以被用于列出(shortlist)如下实体：该实体的回归模型将用于测试新实体以进行分类。

[0303] 图9示出了根据实施例的对来自新添加的HUB的数据集或实体进行分类的过程。

[0304] 如图9中所示，根据实施例，当添加新HUB时(在这个示例中，Oracle HUB 442)，由该HUB提供的数据集或实体(例如，一方信息444，以及客户信息446)基于之前创建的训练数据或播种数据由模型分类为一方(party)448。

[0305] 例如，根据实施例，从新数据集或实体的数据计算列统计数据，并且使用这个信息以及作为摄取的一部分而可用的其它元数据来创建表示该实体的子图的一组谓词(predicate)。

[0306] 根据实施例，列统计数据计算在最大似然估计(MLE)方法、子图以及数据集的回归模型中是有用的。为新实体生成的图谓词集合将用于列出候选实体模型，以便对新实体进行测试和分类。

[0307] 图10进一步示出了根据实施例的对来自新添加的HUB的数据集或实体进行分类的过程。

[0308] 如图10中所示，根据实施例，将表示要被分类的新数据集或实体的子图的谓词与表示已经是知识图的一部分的数据集或实体的类似子图进行比较450。基于匹配概率的匹配实体的排名用于列出用于在分类新实体的测试中使用的实体模型。

[0309] 图11进一步示出了根据实施例的对来自新添加的HUB的数据集或实体进行分类的过程。

[0310] 如图11中所示，根据实施例，列出的匹配数据集或实体的回归模型用于测试来自新数据集或实体的数据。可以扩展ML流水线以包括附加的分类方法/模型，以提高过程的准确性。如果在可接受的阈值(例如，在这个示例中，大于0.8的概率)内存在匹配，那么分类服务将对新实体进行分类452。如果不存在匹配，那么可以将数据集或实体作为新业务类型添加到知识图。用户还可以通过接受或拒绝结果来验证分类。

[0311] 功能类型分类

[0312] 根据实施例，实体的功能类型(functional type)是它的基因型。功能类型还可以被描述为通过其定义变换动作的接口。例如，连接变换或过滤器是对功能类型(诸如在这种情况下的关系实体)定义的。总之，所有变换都是依据功能类型作为参数定义的。

[0313] 图12示出了根据实施例的用于功能类型分类的对象图。

[0314] 如图12中通过对象图460所示，根据实施例，系统可以通过一组规则来描述一般情况(在这个示例中，维度、级别(level)或立方体)，数据集或实体针对该一组规则被评估以识别它的功能类型。

[0315] 例如，根据实施例，可以依据其量度(measure)属性和维度来描述多维立方体，其中每个度量属性和维度本身可以依据其类型和其它特点来定义。规则引擎将评估业务类型实体并基于评估来对它的功能类型进行注释。

[0316] 图13示出了根据实施例的维度功能类型分类的示例。

[0317] 如图13中所示的示例功能类型分类470层次结构中所示，根据实施例，级别可以例如依据其维度和级别属性来定义。

[0318] 图14示出了根据实施例的立方体功能类型分类的示例。

[0319] 如图14中所示的示例功能类型分类480层次结构中所示，根据实施例，立方体可以例如依据其量度属性和维度来定义。

[0320] 图15示出了根据实施例的用于评估业务实体的功能类型的功能类型分类的示例

用法。

[0321] 如图15中所示，在这个示例490中，根据实施例，销售数据集应当由规则引擎评估为立方体功能类型。类似地，产品、客户和时间应当被评估为维度和级别(例如按年龄组、性别)。

[0322] 根据实施例，下面提供了识别该实体的功能类型和用于这个特定示例的数据集或实体元素的规则，包括可以被指定用于评估相同功能类型的若干规则。例如，无论是否存在对父级实体的引用，“Date(日期)”类型的列都可以被视为维度。类似地，邮政编码、性别和年龄可以只需要数据规则将它们识别为维度：

[0323] Customer

[0324] Id,Name→(Dimension isComposedOf DimensionAttrib)

[0325] AgeGroup,Gender→(Dimension isComposedOf IdAttrib,IdAttrib references Level)

[0326] Time

[0327] Day→(Dimension/Level isComposedOf DimensionAttrib/LevelAttrib)

[0328] Month→(Dimension isComposedOf IdAttrib,IdAttrib references Level)

[0329] Sales

[0330] Qty,Price,Amount→(Cube isComposedOf CubeAttr and Data rule on this columns,for example numeric min/max values,probability density)

[0331] Custid→(DimAttr references Dimension,Cube isComposedOf CubeAttr)

[0332] Date→(references Dimension,Cube isComposedOf CubeAttr)

[0333] 图16示出了根据实施例的用于功能变换的对象图。

[0334] 如图15中所示，根据这个示例500，根据实施例，可以针对功能类型定义变换函数。业务实体(业务类型)被注释为功能类型；缺省情况下包括复杂业务类型具有功能类型“实体”。

[0335] 图17示出了根据实施例的推荐引擎的操作。

[0336] 如图17中所示，根据实施例，推荐引擎生成推荐，推荐是针对业务类型定义的一组动作。每个动作都是用于对(一个或多个)数据集应用变换的指令。

[0337] 根据实施例，推荐上下文530抽象推荐的源，并包含元数据，该元数据用于识别生成推荐的提议集。上下文允许推荐引擎基于用户的响应来学习和优先化推荐。

[0338] 根据实施例，目标实体推导器/映射器512使用目标的定义(以及注释数据集或实体和属性业务类型的分类服务)来进行变换推荐，所述变换推荐促进当前数据集映射到目标。当用户以已知目标对象(比如销售立方体)起动并构建流水线来实例化立方体时，这是常见的。

[0339] 根据实施例，模板(流水线/解决方案)514定义可重用的一组流水线步骤和变换，以实现期望的最终结果。例如，模板可能包含丰富、变换和发布到数据集市的步骤。这种情况下的推荐集将反映模板设计。

[0340] 根据实施例，分类服务516识别从HUB摄取到数据湖中的数据集或实体的业务类型。可以基于应用于类似实体(业务类型)的变换或与目标实体推导器/映射器一起进行关于该实体的推荐。

[0341] 根据实施例，功能类型服务518基于所定义的规则来注释数据集或实体可以采用的(一个或多个)功能类型。例如，为了从给定数据集生成立方体或将其与维度表联接，评估数据集是否满足定义立方体功能类型的规则是重要的。

[0342] 根据实施例，来自流水线组件520的模式推断允许推荐引擎汇总在类似上下文中对现有流水线定义中的给定业务类型执行的变换，并建议类似的变换作为当前上下文中的推荐。

[0343] 根据实施例，推荐上下文可以用于处理推荐532，包括动作534、变换函数535、动作参数536、函数参数537和业务类型538。

[0344] 数据湖/数据管理策略

[0345] 如前所述，根据实施例，数据湖提供了用于持久化来自系统HUB或其它组件的信息的储存库。

[0346] 图18示出了根据实施例的数据湖的使用。

[0347] 如图18中所示，根据实施例，数据湖可以与一个或多个数据访问API 540、高速缓存542和持久性存储库544相关联，一个或多个数据访问API 540、高速缓存542和持久性存储库544一起操作以接收已经标准化的摄取数据，以与多个流水线552、554、556一起使用。

[0348] 根据实施例，可以使用各种不同的数据管理策略来管理数据湖中的数据(性能、可扩展性)及其生命周期，数据管理测量可以被广义地分类为数据驱动的或过程驱动的。

[0349] 图19示出了根据实施例使用数据驱动策略来管理数据湖。

[0350] 如图19中所示，根据实施例，在数据驱动方法中，基于HUB或数据服务器定义来导出管理单元。例如，在这种方法中，来自Oracle 1HUB的数据可以存储在与该HUB相关联的第一数据中心560中，而来自SFHUB1的数据可以存储在与该HUB相关联的第二数据中心562中。

[0351] 图20示出了根据实施例使用过程驱动策略来管理数据湖。

[0352] 如图20中所示，根据实施例，在过程驱动的方法中，基于访问数据的相关流水线来导出管理单元。例如，在这种方法中，与销售流水线相关联的数据可以存储在与该流水线相关联的第一数据中心564中，而来自其它流水线(例如，流水线1、2、3)的数据可以存储在与这些其它流水线相关联的第二数据中心566中。

[0353] 流水线

[0354] 根据实施例，流水线定义要对摄取的数据执行的变换或处理。处理后的数据可以存储在数据湖中，或者可以发布到另一个端点(例如，DBCS)。

[0355] 图21示出了根据实施例的流水线编译器的使用。

[0356] 如图21中所示，根据实施例，流水线编译器582在设计环境570和执行环境580之间操作，包括接受一个或多个流水线元数据572和DSL(例如，Java DSL 574、JSON DSL 576、Scala DSL 578)，并提供用于执行环境的输出，例如，作为Spark应用584和/或SQL语句586。

[0357] 图22示出了根据实施例的示例流水线图。

[0358] 如图22中所示，根据实施例，流水线588包括流水线步骤的列表。不同类型的流水线步骤表示可以在流水线中执行的不同类型的操作。每个流水线步骤可以具有一般由流水线步骤参数描述的多个输入数据集和多个输出数据集。通过将输出流水线步骤参数从先前流水线步骤绑定到后续流水线步骤来定义流水线中的操作的处理次序。以这种方式，流水线步骤和流水线步骤参数之间的关系形成有向无环图(DAG)。

[0359] 根据实施例，如果流水线包含表示流水线的输入和输出流水线步骤参数的一个或多个特殊流水线步骤(签名流水线)，那么该流水线可以在另一个流水线中被重用。封闭(enclosing)流水线是指通过(流水线使用)流水线步骤被重用的流水线。

[0360] 图23示出了根据实施例的数据流水线的示例。

[0361] 如图23中所示的示例数据流水线600中所示，根据实施例，数据流水线执行数据变换。流水线中的数据流被表示为流水线步骤参数的绑定。各种类型的流水线步骤被支持以用于不同的变换操作，包括例如：实体(从数据湖检索数据或将处理后的数据发布到数据湖/其它HUB)；以及联接(多个源的融合)。

[0362] 图24示出了根据实施例的数据流水线的另一个示例。

[0363] 如图24中所示的示例数据流水线610中所示，根据实施例，数据流水线P1可以在另一个数据流水线P2中被重用。

[0364] 图25示出了根据实施例的编排流水线的示例。

[0365] 如图25中所示的示例编排流水线620中所示，根据实施例，通过使用编排流水线，流水线步骤可以用于表示需要在整个编排流程中执行的任务或作业。假设编排流水线中的所有流水线步骤都具有一个输入流水线步骤参数和一个输出流水线步骤参数。任务之间的执行依赖性可以表示为流水线步骤参数之间的绑定。

[0366] 根据实施例，如果流水线步骤无条件地依赖于相同的在前流水线步骤(即，分叉)，那么可以调度任务的并行执行。如果流水线步骤依赖于多个在前路径，那么流水线步骤在其自己的执行(即，联接)之前将等待所有多条路径完成。但是，这并不总是意味着任务是并行执行的。编排引擎可以取决于可用资源来决定是以串行还是并行方式执行任务。

[0367] 在图25中所示的示例中，根据实施例，首先执行流水线步骤1。如果并行执行流水线步骤2和流水线步骤3，那么仅当流水线步骤2和流水线步骤3都完成时才执行流水线步骤4。编排引擎还可以将这个编排流水线串行执行为(流水线步骤1、流水线步骤2、流水线步骤3、流水线步骤4)，或(流水线步骤1、流水线步骤3、流水线步骤2、流水线步骤4)，只要它满足流水线步骤之间的依赖关系即可。

[0368] 图26进一步示出了根据实施例的编排流水线的示例。

[0369] 如图26中所示的示例流水线625中所示，根据实施例，每个流水线步骤可以返回状态630，诸如例如取决于其自身语义的成功或错误状态。基于流水线步骤的返回状态，两个流水线步骤之间的依赖关系可以是有条件的。在所示的示例中，首先执行流水线步骤1；如果它成功完成，那么将执行流水线步骤2，否则将执行流水线步骤3。在执行流水线步骤2或流水线步骤3之后，将执行流水线步骤4。

[0370] 根据实施例，可以对编排流水线进行嵌套，使得一个编排流水线可以通过流水线使用来引用另一个编排流水线。编排流水线还可以将数据流水线称为流水线使用。编排流水线和数据流水线之间的区别在于，编排流水线是指不包含签名流水线步骤的数据流水线，而数据流水线可以重用包含签名流水线步骤的另一个数据流水线。

[0371] 根据实施例，取决于流水线步骤的类型和代码优化，数据流水线可以被生成为用于在Spark集群中执行的单个Spark应用，被生成为用于在DBCS中执行的多个SQL语句，或者被生成为SQL和Spark代码的混合。对于编排流水线，它可以被生成以供在底层执行引擎中或在诸如例如Oozie之类的工作流调度组件中执行。

[0372] 协调架构

[0373] 根据实施例，协调架构(fabric)或架构控制器提供必要的工具来部署和管理框架组件(服务提供商)和(用户设计的)应用程序，管理应用执行和资源需求/分配，并提供集成框架(消息传送总线)以促进各个组件之间的通信。

[0374] 图27示出了根据实施例的包括消息传送系统的协调架构的使用。

[0375] 如图27中所示，根据实施例，消息传送系统(例如，Kafka)650协调资源管理器660(例如，Yarn/Mesos)、调度器662(例如，Chronos)、应用调度器664(例如，Spark)和多个节点(在这里表示为节点652、654、656、658)之间的交互。

[0376] 根据实施例，资源管理器被用于管理数据计算任务/应用的生命周期，包括调度、监视、应用执行、资源仲裁和分配、负载平衡；包括管理在消息驱动的组件集成框架中的组件(消息的生产者和消费者)的部署和配置；在无需停机的情况下的组件(服务)升级；以及在很少或没有服务中断的情况下的基础设施升级。

[0377] 图28进一步示出了根据实施例的包括消息传送系统的协调架构的使用。

[0378] 如图28中所示，根据实施例，通过简单的数据驱动的流水线执行用例来说明协调架构中的跨组件的依赖关系，其中(c)指示消费者，并且(p)指示生产者。

[0379] 根据图28中所示的实施例，调度器(p)通过发起向HUB的数据摄取来起动过程(1)。摄取引擎(c)处理将数据从HUB摄取到数据湖中的请求(2)。在摄取过程完成之后，摄取引擎(p)传送(3)完成状态以发起流水线处理。如果调度器支持数据驱动的执行，那么它可以自动发起(3a)要执行的流水线进程。流水线引擎(c)计算(4)等待数据以供执行的流水线。流水线引擎(p)传送(5)要调度的流水线应用的列表以供执行。调度器得到对于流水线的执行调度请求(6)；并发起流水线的执行(6a)。应用调度器(例如，Spark)与资源管理器仲裁(7)以进行资源分配以执行流水线。应用调度器将用于执行的流水线(8)发送给所分配的节点中的执行器。

[0380] 内部部署的代理

[0381] 根据实施例，内部部署的代理促进对本地数据的访问并且以有限的方式促进分布式流水线执行。内部部署的代理被提供并被配置为与例如Cloud DI服务通信，以处理数据访问和远程流水线执行请求。

[0382] 图29示出了根据实施例的与系统一起使用的内部部署的代理。

[0383] 如图29中所示，根据实施例，云代理适配器682提供(1)内部部署的代理680并配置代理适配器端点以进行通信。

[0384] 摄取服务通过消息传送系统发起(2)对HUB1的本地数据访问请求。通过提供对通过摄取服务发起的请求的访问以及将来自内部部署的代理的数据写入数据湖并通过消息传送系统通知任务的完成，云代理适配器作为内部部署的代理和消息系统之间的中间体(3)操作。

[0385] 前提代理针对数据访问请求来轮询(4)云代理适配器以处理数据或将数据上载到云中。云代理适配器将数据(5)写入数据湖，并通过消息传送系统通知流水线。

[0386] DFML流过程

[0387] 图30示出了根据实施例的数据流过程。

[0388] 如图30中所示，根据实施例，在摄取步骤692期间，从各种源(例如，SFDC、S3或

DBaaS)摄取数据。

[0389] 在数据准备步骤693期间,例如通过去重复、标准化或丰富来准备所摄取的数据。

[0390] 在变换步骤694期间,系统在数据集处执行合并、过滤或查找,以变换数据。

[0391] 在模型步骤695期间,生成一个或多个模型以及到模型的映射。

[0392] 在发布步骤696期间,系统可以发布模型,指定策略和调度,以及填充目标数据结构。

[0393] 元数据和数据驱动的自动映射

[0394] 根据实施例,系统可以提供对于在一个或多个数据源或数据目标(在本文中在一些实施例中称为HUB)之间自动映射复杂数据结构、数据集或实体的支持。自动映射可以由数据集的元数据、模式和统计剖析驱动;并且用于将与输入HUB相关联的源数据集或实体映射到目标数据集或实体,或反之亦然,以产生以一格式或组织(投影)准备的输出数据,以与一个或多个输出HUB一起使用。

[0395] 例如,根据实施例,对于实现(例如,构建)数据流、流水线或Lambda应用的用户,该用户可能期望选择要从输入HUB内的源或输入数据集或实体映射到输出HUB内的目标或输出数据集或实体的数据。

[0396] 根据实施例,由于针对非常大的HUB集合和数据集或实体手动生成输入HUB到输出HUB的数据映射可以是非常耗时且低效的任务,因此自动映射可以通过向用户提供对于映射数据的推荐而使用户能够专注于数据流应用(例如,流水线、Lambda应用)的简化。

[0397] 根据实施例,数据AI子系统可以经由图形用户界面(例如,Lambda Studio集成开发环境(IDE))接收针对自动映射服务的自动映射请求。

[0398] 根据实施例,请求可以包括为要对其执行自动映射服务的应用指定的文件,连同识别输入HUB、数据集或实体以及一个或多个属性的信息。应用文件可以包括关于用于应用的数据的信息。数据AI子系统可以处理应用文件以提取实体名称和实体的其它形状特点,包括属性名称和数据类型,自动映射服务可以在搜索中使用实体名称和实体的其它形状特点来查找用于映射的潜在候选集。

[0399] 根据实施例,系统可以访问用于变换到HUB中的数据,诸如例如数据仓库。所访问的数据可以包括各种类型的数据,包括半结构化数据和结构化数据。数据AI子系统可以对所访问的数据执行元数据分析,包括确定数据的一个或多个形状、特征或结构。例如,元数据分析可以确定数据的类型(例如,业务类型和功能类型)以及数据的柱状形状。

[0400] 根据实施例,基于数据的元数据分析,可以识别一个或多个数据样本,并且机器学习过程应用于采样的数据,以确定所访问的数据中的数据的类别,并更新模型。数据的类别例如可以指示数据的相关部分,诸如数据中的事实表。

[0401] 根据实施例,可以使用例如逻辑回归模型或者可以被实现以用于机器学习的其它类型的机器学习模型来实现机器学习。根据实施例,数据AI子系统可以基于数据的类别来分析数据中的一个或多个数据项的关系,该关系指示该数据类别的数据中的一个或多个字段。

[0402] 根据实施例,数据AI子系统可以执行用于特征提取的过程,包括针对所访问数据的属性确定随机采样数据的一个或多个元数据、数据类型和统计简档。

[0403] 例如,根据实施例,数据AI子系统可以基于所访问数据的数据类别来生成所访问

数据的简档。简档可以被生成以用于将数据变换到输出HUB中，并且例如在图形用户界面中被显示。

[0404] 根据实施例，作为创建这种简档的结果，模型可以支持具有一定的置信度的关于候选数据集或实体如何与输入数据集或实体相似的推荐。推荐可以被过滤和排序，然后经由图形用户界面提供给用户。

[0405] 根据实施例，自动映射服务可以基于用户正在构建数据流应用(例如，流水线、Lambda应用)的阶段来动态地建议推荐。

[0406] 根据实施例，在实体级别的推荐的示例可以包括对属性的推荐，例如，要自动映射到另一个属性的实体的列，或另一个实体。该服务可以基于用户以前的活动来持续提供推荐并指导用户。

[0407] 根据实施例，可以使用由自动映射服务提供的应用编程接口(API)(例如，REST API)将推荐从例如与输入HUB相关联的源数据集或实体映射到与输出HUB相关联的目标数据集或实体。推荐可以指示数据的投影，例如，属性、数据类型和表达式，其中表达式可以是数据类型的属性的映射。

[0408] 根据实施例，系统可以提供图形用户界面以基于推荐来选择输出HUB以用于变换所访问的数据。例如，图形用户界面可以使用户能够选择用于将数据变换到输出HUB的推荐。

[0409] 自动映射

[0410] 根据实施例，可以在数学上定义自动映射功能，其中实体集E被定义为：

[0411] $$\mathrm{E} = \{e_1, e_2, \ldots, e_n \mid \forall\, e_i \in S\}$$

[0412] Shape:S＝{MetaData×DataType×StatisticalProfile}

[0413] 其中形状集S包括元数据、数据类型和统计剖析(statistical profiling)维度。目标是找到j，使得e_i和e_j之间的相似性概率最高。

[0414] $$e_j^* = \arg\max_j p\{Sim(e_i, e_j) \mid entity = e_i\}$$

[0415] 在数据集或实体级别，问题是二元问题，即，数据集或实体是相似还是不相似。设f_s、f_t、$h(f_s, f_t)$表示源、目标的特征集合以及源和目标之间的交互特征。因此，目标是估计相似性概率：

[0416] $p＝g(f_s, f_t, h(f_s, f_t); \beta)$

[0417] $g(.):[0,1]^Q \rightarrow [0,1]$

[0418] 对数似然函数定义为：

[0419] $$\ell(\boldsymbol{\beta}) = \sum_{q=1}^{Q} c_q \log p_q(\boldsymbol{\beta}) + (1 - c_q) \log(1 - p_q(\boldsymbol{\beta}))$$

[0420] 因此，在逻辑回归模型中，可以如下估计未知系数：

[0421] $$g(x; \boldsymbol{\beta}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T x}}$$

[0422] $\beta^*＝\arg_\beta\max\ l(\beta)$

[0423] 根据实施例，可以通过例如从系统外观服务接收HTTP POST请求来触发自动映射服务。系统外观API将数据流应用(例如，流水线、Lambda应用)JSON文件从UI传递到自动映射REST API，并且解析器模块处理应用JSON文件并提取实体名称和数据集或实体的形状，包括属性名称和数据类型。

[0424] 根据实施例，自动映射服务使用搜索来快速找到用于映射的潜在候选集。候选集需要是高度相关的集合，并且因此可以使用特殊索引和查询来实现此目的。这个特殊索引结合特殊搜索字段，其中实体的所有属性都被存储并使用所有N-元(N-gram)组合被标记化。在查询时，搜索查询构建器模块利用基于例如Levenshtein距离的模糊搜索特征，以使用给定实体的实体名称和属性名称两者构造特殊查询，并利用搜索增强功能来通过字符串相似性意义上的结果的相关性对结果进行排序。

[0425] 根据实施例，推荐引擎向用户示出多个相关结果，例如在大多数情况下选择前N个结果。

[0426] 根据实施例，为了实现高精度，机器学习模型基于提取出的特征来比较源和目标对以及实体的得分相似性。特征提取包括用于每个属性的随机采样数据的元数据、数据类型和统计简档。

[0427] 虽然这里提供的描述根据实施例一般地描述了使用逻辑回归模型来学习从Oracle业务智能(OBI)族系映射数据获取的自动映射示例，但是可以代替地使用其它受监督的机器学习模型。

[0428] 根据实施例，逻辑回归模型的输出表示候选数据集或实体如何在统计意义上与输入数据集或实体相似的总体置信度。为了找到精确的映射，可以使用一个或多个其它模型来使用类似的特征计算源属性与目标属性的相似性。

[0429] 最后，根据实施例，对推荐进行过滤和排序，并将推荐发送回系统外观并传递给用户界面。自动映射服务基于用户在数据流应用(例如，流水线或Lambda应用)设计期间的哪个阶段来动态地建议推荐。该服务可以基于用户以前的活动来持续提供推荐并指导用户。自动映射可以在正向工程或逆向工程意义上执行。

[0430] 图31示出了根据实施例的数据类型的自动映射。

[0431] 如图31中所示，根据实施例，系统外观701和自动映射API 702允许从软件开发组件(例如，Lambda Studio)接收数据流应用(例如，流水线或Lambda应用)。解析器704处理应用的JSON文件，并提取实体名称和形状，包括属性名称和数据类型。

[0432] 根据实施例，搜索索引708被用于支持主搜索710，以找到用于映射的潜在候选数据集或实体集合。搜索查询构建器模块706使用给定实体的实体名称和属性名称来构造查询，以确定对数据集或实体的选择712。

[0433] 根据实施例，机器学习(ML)模型被用于基于提取出的特征比较源和目标对，并对数据集或实体的相似性进行评分。特征提取714包括用于每个属性的随机采样数据的元数据、数据类型和统计简档。

[0434] 根据实施例，逻辑回归模型716提供候选实体如何与输入实体相似的总置信度作为输出。为了找到更确切的映射，使用列映射模型718来进一步评估源属性与目标属性的相似性。

[0435] 根据实施例，然后将推荐排序为自动映射720，以返回到软件开发组件，例如

Lambda Studio。在数据流应用(例如,流水线或Lambda应用)设计期间,自动映射服务基于用户处于哪个阶段来动态地建议推荐。该服务可以基于用户以前的活动来持续提供推荐并指导用户。

[0436] 图32示出了根据实施例的用于生成映射的自动映射服务。

[0437] 如图32中所示,根据实施例,可以提供自动映射服务以用于生成映射,包括其中UI查询728被接收并被传递给查询理解引擎729,然后传递给查询分解730组件。

[0438] 根据实施例,使用数据HUB 722执行主搜索710,以确定候选数据集或实体731,以用于后续元数据和统计简档处理732。

[0439] 根据实施例,结果被传递到获取统计数据简档734组件,并且数据AI系统724提供特征提取735。结果被用于合成736、根据模型723的最终置信度合并和排名739,以及提供推荐和相关联的置信度740。

[0440] 自动映射示例

[0441] 图33示出了根据实施例的源模式和目标模式之间的映射的示例。

[0442] 如图33中所示,根据实施例,示例741示出了基于例如(a)上位词、(b)同义词、(c)相等、(d)同音(Soundex)和(e)模糊匹配的简单自动映射示例。

[0443] 图34示出了根据实施例的源模式和目标模式之间的映射的另一个示例。

[0444] 如图34中所示,如果这个信息不相关,那么仅基于元数据的方法将失败。根据实施例,图34示出了示例742,其中源和目标属性名称完全没有信息。当缺少元数据特征时,系统可以采用包括对特征的统计剖析的模型来实现找到类似的实体。

[0445] 自动映射过程

[0446] 图35示出了根据实施例的用于提供数据类型的自动映射的过程。

[0447] 如图35中所示,在步骤744,根据实施例,所访问的数据被处理,以执行所访问的数据的元数据分析。

[0448] 在步骤745,识别所访问的数据的一个或多个样本。

[0449] 在步骤746,应用机器学习过程来确定所访问数据内的数据的类别。

[0450] 在步骤748,基于所确定的数据的类别生成所访问的数据的简档,以用于自动映射所访问的数据。

[0451] 动态推荐和模拟

[0452] 根据实施例,系统可以包括软件开发组件(在本文中在一些实施例中称为Lambda Studio)和图形用户界面(在本文中在一些实施例中称为流水线编辑器或Lambda Studio IDE),其提供与系统一起使用的可视环境,包括基于对与数据相关联的含义或语义的理解来提供用于对从输入HUB访问的数据执行语义动作的实时推荐。

[0453] 例如,根据实施例,图形用户界面可以提供用于对从输入HUB访问的数据执行操作(也称为语义动作)的实时推荐,包括该数据的部分数据、形状或其它特点。可以基于与数据相关联的含义或语义来对数据执行语义动作。数据的含义可以用于选择可以对数据执行的语义动作。

[0454] 根据实施例,语义动作可以表示对于一个或多个数据集的操作符,并且可以引用在系统中声明性地定义的基本语义动作或功能。可以通过执行语义动作来生成一个或多个处理后的数据集。语义动作可以由与具体功能或业务类型相关联的参数定义。它们表示要

处理的具体上游数据集。图形用户界面可以是元数据驱动的，使得图形用户界面被动态生成，以基于数据中识别出的元数据提供推荐。

[0455] 图36示出了根据实施例的显示针对所访问的数据启用的一个或多个语义动作的系统。

[0456] 如图36中所示，根据实施例，通过使用具有用户输入区域752的图形用户界面750，对为所访问的数据启用的语义动作的查询被发送到系统的知识源，其中该查询指示所访问的数据的分类。

[0457] 根据实施例，从知识源接收对查询的响应，其中该响应指示为所访问的数据启用并基于数据的分类识别的一个或多个语义动作。

[0458] 根据实施例，显示为所访问的数据启用的语义动作中的所选择的语义动作，以供选择并与所访问的数据一起使用，包括在处理所访问的数据期间自动提供或更新为所访问的数据启用的所选择的语义动作756或推荐758的列表。

[0459] 根据实施例，可以动态地提供推荐而不是基于静态数据预先计算推荐。例如，系统可以基于实时访问的数据来实时提供推荐，考虑诸如例如用户简档或用户的体验级别之类的信息。系统提供的用于实时数据的推荐对于产生数据流应用(例如，流水线、Lambda应用)可以是显著的、相关的和精确的。可以基于关于与特定元数据相关联的数据的用户行为来提供推荐。系统可以推荐对信息的语义动作。

[0460] 例如，根据实施例，系统可以摄取数据、变换数据、集成数据并向任意系统发布数据。系统可以推荐实体被用来以有趣的分析方式分析实体的数值量度中的一些数值量度；在各个维度上枢转该数据，甚至指出哪些是有趣的维度，为这些维度层次结构汇总数据，并用更多的见解(insight)来丰富数据。

[0461] 根据实施例，可以基于使用诸如例如数据的元数据分析之类的技术的数据分析来提供推荐。

[0462] 根据实施例，元数据分析可以包括确定数据的分类，诸如例如数据的形状、特征和结构。元数据分析可以确定数据的类型(例如，业务类型和功能类型)。元数据分析还可以指示数据的列形状。根据实施例，可以将数据与元数据结构(例如，形状和特征)进行比较，以确定数据的类型和与数据相关联的属性。可以在系统的系统HUB(例如，知识源)中定义元数据结构。

[0463] 根据实施例，通过使用元数据分析，系统可以查询系统HUB以基于元数据识别语义动作。推荐可以是基于对从输入HUB访问的数据的元数据的分析而确定的语义动作。具体而言，语义动作可以映射到元数据。例如，语义动作可以映射到对于其这些动作被允许和/或适用的元数据。语义动作可以是用户定义的，和/或可以基于数据结构来定义。

[0464] 根据实施例，可以基于与元数据相关联的条件来定义语义动作。可以修改系统HUB，使得语义动作被修改、删除或扩充。

[0465] 根据实施例，语义动作的示例可以包括构建立方体、过滤数据、将数据分组、聚合数据或可以对数据执行的其它动作。通过基于元数据定义语义动作，可以不需要映射或方案来确定对于数据允许的语义动作。当发现新的和不同的元数据结构时，可以定义语义动作。照此，系统可以基于使用针对作为输入接收到的数据而分析的元数据对语义动作的识别来动态地确定推荐。

[0466] 根据实施例，语义动作可以由第三方定义，使得第三方可以供给数据(例如，定义与元数据相关联的一个或多个语义动作的数据)。系统可以动态地查询系统HUB，以确定可用于元数据的语义动作。照此，可以修改系统HUB，使得系统基于这种修改来确定当时允许的语义动作。系统可以执行操作(例如，过滤、检测和注册)以处理从第三方获得的数据，其中数据定义语义动作；并且可以基于由处理识别的语义动作使语义动作可用。

[0467] 图37和图38示出了根据实施例的显示为所访问的数据启用的一个或多个语义动作的图形用户界面。

[0468] 如图37中所示，根据实施例，软件开发组件(例如，Lambda Studio)可以提供可以显示所推荐的语义动作的图形用户界面(例如，流水线编辑器或Lambda studio IDE)750，所推荐的语义动作用于处理输入数据，或模拟输入数据的处理，以用于投影到输出HUB上。

[0469] 例如，根据实施例，图37中的界面允许用户显示与数据流应用(例如，流水线、Lambda应用)相关联的选项752，包括例如输入HUB定义754。

[0470] 根据实施例，在创建数据流应用(例如，流水线、Lambda应用)或利用输入数据对数据流应用(例如，流水线、Lambda应用)的模拟期间，一个或多个语义动作756或者其它推荐758可以显示在图形用户界面上，以供用户查看。

[0471] 根据实施例，在模拟模式下，软件开发组件(例如，Lambda Studio)提供沙箱环境，该沙箱环境允许用户立即看到对输出执行各种语义动作的结果，包括在处理所访问的数据期间自动更新适于所访问的数据的语义动作的列表。

[0472] 例如，如图38中所示，根据实施例，从用户搜索某种信息的起点，系统可以推荐关于信息的操作760，例如，实体用于以有趣的分析方式分析其数值量度中的一些数值量度；在不同维度上枢转数据，甚至指出哪些是有趣的维度，汇总这些维度层次结构的数据，并通过更多洞察丰富数据。

[0473] 根据实施例，在所示的示例中，已经为系统中的可分析实体推荐了源和维度，使得构建多维立方体的任务几乎就是指向和点击的任务。

[0474] 通常，此类活动需要大量经验和特定于领域的知识。使用机器学习来分析对于常见集成模式的用户行为模式和数据特点，连同语义搜索和来自机器学习的推荐的结合，允许如下方法，该方法为用于构建以业务为中心的应用的应用开发提供最先进的工具。

[0475] 图39示出了根据实施例的用于显示为所访问的数据启用的一个或多个语义动作的过程。

[0476] 如图39中所示，在步骤772，根据实施例，处理所访问的数据，以执行所访问的数据的元数据分析，其中元数据分析包括确定所访问的数据的分类。

[0477] 在步骤774，对于为所访问的数据启用的语义动作的查询被发送到系统的知识源，其中该查询指示所访问的数据的分类。

[0478] 在步骤775，从知识源接收对查询的响应，其中响应指示为所访问的数据启用并基于数据的分类识别的一个或多个语义动作。

[0479] 在步骤776，在图形用户界面，显示为所访问的数据启用的语义动作中的所选择的语义动作，以供选择和与所访问的数据一起使用，包括在处理所访问的数据期间自动提供或更新为所访问的数据启用的语义动作中的所选择的语义动作的列表。

[0480] 数据流的功能分解

[0481] 根据实施例，系统可以基于从软件应用的数据流的功能分解识别的模式来提供用于推荐对输入数据的动作和变换的服务，包括确定在后续应用中数据流的可能变换。可以将数据流分解为描述数据的变换、谓词和适用于数据的业务规则的数据、以及数据流中使用的属性的模型。

[0482] 图40示出了根据实施例的对于将流水线、Lambda应用评估成其组成部分以促进模式检测和归纳学习的支持。

[0483] 如图40中所示，根据实施例，功能分解逻辑800或软件组件可以作为可由计算机系统或其它处理设备执行的软件或程序代码提供，并且可以用于提供功能分解802和推荐804以供(例如，在流水线编辑器或Lambda Studio IDE内)显示805。例如，系统可以提供服务，该服务基于从用于数据流应用(例如，流水线、Lambda应用)的数据流的功能分解识别出的模式/模板来推荐对数据的动作和变换，即，通过数据流的功能分解来观察用于确定后续应用中数据流的可能变换的模式。

[0484] 根据实施例，服务可以由框架实现，框架可以将数据流分解或划分为描述应用于数据的变换、谓词和应用于数据的业务规则以及在数据流中使用的属性的模型。

[0485] 传统上，用于应用的数据流可以表示对数据的一系列变换，并且应用于数据的变换类型是高度与上下文有关的。在大多数数据集成框架中，过程族系通常在数据流如何被持久化、分析和生成方面是受限或不存在的。根据实施例，系统使得能够基于语义上丰富的实体类型从流或图中导出上下文相关的模式，并进一步学习数据流语法和模型，并使用它来生成给定类似上下文的复杂数据流图。

[0486] 根据实施例，系统可以基于数据流的设计规范生成定义模式和模板的一个或多个数据结构。可以将数据流分解为定义函数表达式的数据结构，以确定模式和模板。数据流可以用于预测并生成用于确定数据变换的推荐的模式的函数表达式，其中推荐基于从分解的数据流和固有模式的归纳学习导出的模型，并且可以细粒度化(例如，推荐对特定属性进行标量变换或在谓词中使用一个或多个属性进行过滤或联接)。

[0487] 根据实施例，数据流应用(例如，流水线、Lambda应用)可以使用户能够基于对数据的语义动作来生成复杂的数据变换。系统可以将数据变换存储为定义用于流水线、Lambda应用的数据流的一个或多个数据结构。

[0488] 根据实施例，分解用于数据流应用(例如，流水线、Lambda应用)的数据流可以被用于确定数据的模式分析并生成函数表达式。可以对语义动作以及变换和谓词或业务规则执行分解。可以通过分解来识别先前的应用语义动作中的每个应用语义动作。通过使用归纳过程，可以从包括其上下文元素(业务类型和功能类型)的数据流中提取业务逻辑。

[0489] 根据实施例，可以为该过程生成模型；并且，基于归纳，可以生成上下文丰富的规定性数据流设计推荐。推荐可以基于来自模型的模式推断，其中推荐中的每个推荐可以对应于可以对应用的数据执行的语义动作。

[0490] 根据实施例，系统可以执行用于基于功能分解来推断用于数据变换的模式的过程。系统可以访问一个或多个数据流应用(例如，流水线、Lambda应用)的数据流。可以处理数据流，以确定一个或多个函数表达式。可以基于数据流中识别出的动作、谓词或业务规则来生成函数表达式。动作、谓词或业务规则可以用于识别(例如，推断)关于数据流的变换模式。推断变换的模式可以是被动过程。

[0491] 根据实施例，可以基于针对不同应用的数据流的被动(passive)分析以众包方式确定变换的模式。可以使用机器学习(例如，深度强化学习)来确定模式。

[0492] 根据实施例，可以针对为数据流应用(例如，流水线、Lambda应用)生成的函数表达式来识别变换的模式。可以分解一个或多个数据流，以推断用于数据变换的模式。

[0493] 根据实施例，通过使用该模式，系统可以为新数据流应用(例如，流水线、Lambda应用)的数据流推荐一个或多个数据变换。在用于货币交易的数据处理的数据流的示例中，系统可以识别关于数据的变换模式。系统还可以为应用的新数据流推荐一个或多个变换，其中数据流涉及用于类似货币交易的数据。可以根据模式以类似的方式执行(一个或多个)变换，使得根据(一个或多个)变换修改新数据流，以产生类似的货币交易。

[0494] 图41示出了根据实施例的针对一个或多个应用中的每个应用生成的一个或多个函数表达式来识别数据流中的变换模式的部件。

[0495] 如前所述，根据实施例，流水线(例如，Lambda应用)允许用户基于与关系演算中的运算符对应的语义动作来定义复杂的数据变换。数据变换通常作为有向无环图或查询被持久化，或者在DFML的情况下作为嵌套功能被持久化。作为嵌套功能来分解和序列化数据流应用(例如，流水线、Lambda应用)使得能够对数据流进行模式分析并归纳数据流模型，该数据流模型然后可以用于生成抽象在类似上下文中关于数据集的复杂变换的函数表达式。

[0496] 根据实施例，嵌套功能分解不仅在语义动作(行或数据集操作符)级别执行，而且还在标量变换和谓词结构处执行，这允许复杂数据流的深度族系能力。基于归纳模型的推荐可以被细粒度化(例如，推荐对特定属性的标量变换或者在谓词中使用一个或多个属性进行过滤或联接)。

[0497] 根据实施例，功能分解的元素一般包括：

[0498] 应用表示顶级数据流变换。

[0499] 动作表示对一个或多个数据集(具体来说，数据帧)的操作符。

[0500] 动作引用在系统中声明性地定义的基本语义动作或功能。动作可以具有一个或多个动作参数，该一个或多个动作参数中的每个动作参数都可以具有具体的角色(输入、输出、输入/输出)和类型，返回一个或多个已处理的数据集，并且可以嵌入或嵌套若干级别深度。

[0501] 动作参数由动作拥有并且具有具体的功能或业务类型，并且表示要处理的具体上游数据集。绑定参数表示在变换中使用的HUB中的数据集或实体。值参数表示在当前变换的上下文中处理的中间或瞬态数据结构。

[0502] 范围解析器允许导出整个数据流中使用的数据集或数据集中的元素的过程族系。

[0503] 图42示出了根据实施例的用于针对为一个或多个应用中的每个应用生成的一个或多个函数表达式识别数据流中的变换模式的对象图。

[0504] 如图42中所示，根据实施例，功能分解逻辑可以被用于将数据流应用(例如，流水线、Lambda应用)的数据流分解或划分为描述将数据、谓词和应用于数据的业务规则以及数据流中使用的属性变换成对注册表的选择的模型，包括例如模式或模板812(如果模板与流水线、Lambda应用相关联)、服务814、函数816、函数参数818和函数类型820。

[0505] 根据实施例，这些功能组件中的每一个可以进一步被分解为例如任务822或动作824，从而反映数据流应用，例如，流水线、Lambda应用。

[0506] 根据实施例，范围解析器826可以被用于通过其范围来解析对特定属性或嵌入式对象的引用。例如，如图42中所示，范围解析器通过其直接范围来解析对属性或嵌入式对象的引用。例如，使用另一个表和过滤器的输出的联接函数将引用两者作为其范围解析器，并且可以与InScopeOf操作结合使用，以将叶子节点解析为其根节点路径。

[0507] 图43示出了根据实施例的针对为一个或多个应用中的每个应用生成的一个或多个函数表达式识别数据流中的变换模式的过程。

[0508] 如图43中所示，根据实施例，在步骤842，访问用于一个或多个软件应用中的每个应用的数据流。

[0509] 在步骤844，处理用于一个或多个软件应用的数据流，以生成表示数据流的一个或多个函数表达式，其中该一个或多个函数表达式是基于在数据流中识别的语义动作和业务规则生成的。

[0510] 在步骤845，针对为一个或多个软件应用中的每个软件应用生成的一个或多个函数表达式来识别数据流中的变换模式，其中语义动作和业务规则用于识别数据流中的变换模式。

[0511] 在步骤847，使用在数据流中识别出的变换模式，为另一个软件应用的数据流提供一个或多个数据变换的推荐。

[0512] 本体学习

[0513] 根据实施例，系统可以执行模式定义的本体分析，以确定与该模式相关联的数据和数据集或实体的类型；从参考模式生成或更新模型，该参考模式包括基于实体及其属性之间的关系定义的本体。包括一个或多个模式的参考HUB可以用于分析数据流，并进一步对用于例如输入数据的变换丰富、过滤或跨实体数据融合进行分类或做出推荐。

[0514] 根据实施例，系统可以执行模式定义的本体分析，以确定参考模式中的数据和实体的类型的本体。换句话说，系统可以从包括基于实体及其属性之间的关系定义的本体的模式生成模型。参考模式可以是系统提供的或缺省的参考模式，或者可替代地是用户提供的或第三方参考模式。

[0515] 虽然一些数据集成框架可以对来自已知源系统类型的元数据进行逆向工程，但是它们不提供元数据的分析以构建可以用于模式定义和实体分类的功能类型系统。收获元数据的范围也是有限的，并且不会扩展到用于提取出的数据集或实体的剖析数据。允许用户指定如下用于本体学习的参考模式的功能目前是不可用的，从该参考模式构建除了用于实体分类(在类似的拓扑空间中)之外还用于复杂过程(业务逻辑)和集成模式的功能类型系统。

[0516] 根据实施例，一个或多个模式可以存储在参考HUB中，参考HUB本身可以在系统HUB内或作为系统HUB的一部分提供。与参考模式一样，参考HUB还可以是用户提供的或第三方参考HUB，或者在多租户环境中可以与特定租户相关联，并且例如通过数据流API访问。

[0517] 根据实施例，参考HUB可以用于分析数据流并进一步对例如变换、丰富、过滤或跨实体数据融合进行分类或做出推荐。

[0518] 例如，根据实施例，系统可以接收将参考HUB定义为用于本体分析的模式的输入。可以导入参考HUB以获得实体定义(属性定义、数据类型以及业务规则或约束、数据集或实体之间的关系)。可以针对所有数据集或实体以及被剖析的数据提取参考HUB中的样本数据

(例如,属性向量,诸如例如列数据),以导出关于数据的若干度量。

[0519] 根据实施例,可以基于参考模式的命名来实例化类型系统。系统可以执行本体分析,以导出描述数据类型的本体(例如,规则集)。本体分析可以确定是被剖析数据(例如,属性或复合值)度量的定义项并描述业务类型元素(例如,UOM、ROI或货币类型)的性质连同其数据简档的数据规则;定义跨数据集或实体以及属性向量(从参考模式导入的约束或引用)的关联的关系规则;以及可以通过数据规则和关系规则的组合导出的复杂规则。然后可以基于通过元数据收获和数据采样导出的规则来定义类型系统。

[0520] 根据实施例,可以基于使用本体分析实例化的类型系统从系统HUB利用模式和模板。然后,系统可以使用类型系统执行数据流处理。

[0521] 例如,根据实施例,数据集或实体的分类和类型注释可以由注册的HUB的类型系统识别。类型系统可以被用于定义从参考模式导出的功能类型和业务类型的规则。通过使用类型系统,可以基于类型系统对数据流中识别出的实体执行诸如例如混合、丰富和变换推荐之类的动作。

[0522] 图44示出了根据实施例的用于生成功能类型规则的系统。

[0523] 如图44中所示,根据实施例,可以作为可由计算机系统或其它处理设备执行的软件或程序代码来提供的规则归纳逻辑850或软件组件使规则851能够与功能类型系统852相关联

[0524] 图44示出了根据实施例的用于生成功能类型规则的系统。

[0525] 如图45中所示,根据实施例,HUB 1可以充当参考本体,用于类型标记、比较、分类或以其它方式评估由其它(例如,新注册的)HUB(例如,HUB 2和HUB 3)提供的元数据模式或本体,并创建适当的规则,以供数据AI系统使用。

[0526] 图46示出了根据实施例的用于生成功能类型规则的对象图。

[0527] 根据实施例,如图46中所示,例如,规则归纳逻辑使规则能够与具有一组功能类型853的功能类型系统(例如,HUB、数据集或实体,以及属性)相关联,并且被存储在注册表中以用于创建数据流应用(例如,流水线、Lambda应用),包括每个功能类型854可以与功能类型规则856和规则858相关联。每个规则可以与规则参数860相关联。

[0528] 根据实施例,第一次处理参考模式时,可以准备包括适于该模式的规则集的本体。

[0529] 根据实施例,下次当新HUB或新模式被评估时,其数据集或实体可以与现有的本体和准备的规则进行比较,并且用于分析新HUB/模式及其实体、以及对系统的进一步学习。

[0530] 虽然在数据集成框架中的一些元数据收获仅限于逆向工程实体定义(属性及其数据类型以及在一些情况下的关系);但是根据实施例,由本文描述的系统提供的方法的不同之处在于允许将模式定义用作参考本体,从该参考本体可以导出业务类型和功能类型,连同参考模式中的数据集或实体的数据剖析度量。然后,这个参考HUB可以用于分析其它HUB(数据源)中的业务实体,以进一步分类或做出推荐(例如,混合或丰富)。

[0531] 根据实施例,该系统采用以下步骤集合来使用参考模式进行本体学习:

[0532] 用户指定使用新注册的HUB作为参考模式的选项。

[0533] 实体定义(例如,属性定义、数据类型、实体之间的关系、约束或业务规则被导入)。

[0534] 为所有数据集或实体提取样本数据,并且对数据进行剖析以导出关于数据的若干度量。

[0535] 基于参考模式的命名法,实例化类型系统(功能类型和业务类型)。

[0536] 导出描述业务类型的规则集。

[0537] 数据规则依据剖析的数据度量来定义并描述业务类型元素的性质(例如,UOM或ROI或货币类型可以被定义为业务类型元素连同其数据简档)。

[0538] 生成定义跨元素的关联(从参考模式导入的约束或引用)的关系规则。

[0539] 生成可以通过数据规则和关系规则的组合导出的复杂规则。

[0540] 基于通过元数据收集和数据采样导出的规则来定义类型系统(功能和业务)。

[0541] 然后,模式或模板可以使用基于参考模式实例化的类型来定义复杂的业务逻辑。

[0542] 然后可以在参考模式的上下文中分析向系统注册的HUB。

[0543] 可以基于从参考模式导出的功能类型和业务类型的规则来执行新注册的HUB中的数据集或实体的分类和类型注释。

[0544] 可以基于类型注释对数据集或实体执行混合、丰富、变换推荐。

[0545] 图47示出了根据实施例的用于基于生成的一个或多个规则来生成功能类型系统的过程。

[0546] 如图47中所示,根据实施例,在步骤862,接收定义参考HUB的输入。

[0547] 在步骤863,访问参考HUB,以获得与由参考HUB提供的数据集或实体相关联的一个或多个实体定义。

[0548] 在步骤864,从参考HUB为一个或多个数据集或实体生成样本数据。

[0549] 在步骤865,对样本数据进行剖析,以确定与样本数据相关联的一个或多个度量。

[0550] 在步骤866,基于实体定义生成一个或多个规则。

[0551] 在步骤867,基于所生成的一个或多个规则生成功能类型系统。

[0552] 在步骤868,持久化功能类型系统和样本数据的简档以用于处理数据输入。

[0553] 外部功能接口

[0554] 根据实施例,系统提供程序接口(在本文中在一些实施例中称为外部功能接口),通过该接口,用户或第三方可以以声明的方式定义服务、功能和业务类型、语义动作,基于功能和业务类型的模式或预定义的复杂数据流,以扩展系统的功能。

[0555] 如前所述,当前数据集成系统可以提供有限的接口,并且不支持类型,也没有用于对象合成和模式定义的良好定义的接口。由于存在这些缺点,因此复杂功能目前还不可用,如跨服务的推荐或统一的应用设计平台,以调用扩展框架的跨服务的语义动作。

[0556] 根据实施例,外部功能接口使用户能够以声明的方式提供定义或(例如,来自客户、其它第三方的)其它信息,以扩展系统的功能。

[0557] 根据实施例,系统是元数据驱动的,使得可以处理通过外部功能接口接收的定义以确定元数据,确定元数据的分类(诸如例如数据类型(例如,功能类型和业务类型)),并将数据类型(功能和业务)与现有元数据进行比较,以确定是否存在类型匹配。

[0558] 根据实施例,通过外部功能接口接收的元数据可以存储在系统HUB中,以供系统访问以用来处理数据流。例如,可以访问元数据以基于作为输入接收的数据集的类型来确定语义动作。系统可以确定针对通过接口提供的数据类型所允许的语义动作。

[0559] 根据实施例,通过提供公共声明性接口,系统可以使用户能够将服务原生(native)类型和动作映射到平台原生类型和动作。这允许通过类型和模式发现的统一的应

用设计体验。它还促进纯声明性数据流定义和设计，其涉及扩展平台的各种服务的组件以及用于各个语义动作的原生代码的生成。

[0560] 根据实施例，可以以自动方式处理通过外部功能接口接收的元数据，使得其中描述的对象或工件(例如，数据类型或语义动作)可以用于操作由系统处理的数据流。从一个或多个第三方系统接收的元数据信息还可以用于定义服务、指示一个或多个功能和业务类型、指示一个或多个语义动作、或指示一个或多个模式/模板。

[0561] 例如，根据实施例，可以确定所访问的数据的分类，诸如数据的功能和业务类型。可以基于关于与信息一起接收的数据的信息来识别分类。通过从一个或多个第三方系统接收数据，可以扩展系统的功能，以基于从第三方接收的信息(例如，服务、语义动作或模式)执行对数据流的数据集成。

[0562] 根据实施例，可以更新系统HUB中的元数据，以包括关于数据识别出的信息。例如，可以基于通过外部功能接口接收的元数据中识别出的信息(例如，语义动作)来更新服务和模式/模板。因此，可以通过外部功能接口增强系统功能而不中断数据流的处理。

[0563] 根据实施例，可以在元数据更新后使用系统HUB中的元数据来处理后续数据流。可以对数据流应用(例如，流水线、Lambda应用)的数据流执行元数据分析。然后，系统HUB可以用于考虑经由外部功能接口提供的定义来确定变换的推荐。可以基于模式/模板来确定变换，模式/模板被用于定义要为服务执行的语义动作，其中语义动作可以类似地考虑经由外部功能接口提供的定义。

[0564] 图48示出了根据实施例的用于基于经由外部功能接口提供的信息来识别用于提供用于数据流的推荐的模式的系统。

[0565] 如图48中所示，根据实施例，经由外部功能接口900接收的定义可以用于更新系统HUB内的服务注册表902、功能和业务类型注册表904或模式/模板906中的一个或多个。

[0566] 根据实施例，更新后的信息可以由包括规则引擎908的数据AI子系统使用，以确定例如系统HUB中的类型注释的HUB、数据集或实体或属性910，以及将这些数据集或实体提供给推荐引擎912，以用于经由软件开发组件(例如，Lambda Studio)为数据流应用(例如，流水线、Lambda应用)提供推荐。

[0567] 图49进一步示出了根据实施例的基于经由外部功能接口提供的信息来识别用于提供用于数据流的推荐的模式。

[0568] 如图49中所示，根据实施例，可以在外部功能接口处接收第三方元数据920。

[0569] 图50进一步示出了根据实施例的基于经由外部功能接口提供的信息来识别用于提供用于数据流的推荐的模式。

[0570] 如图50中所示，根据实施例，在外部功能接口处接收的第三方元数据可以用于扩展系统的功能。

[0571] 根据实施例，系统使得能够通过良好定义的接口允许框架可扩展性，该良好定义的接口允许服务、服务原生的类型、由服务实现的语义动作连同它们的类型化参数、抽象作为服务的一部分可用的预定义算法的模式或模板的注册等。

[0572] 根据实施例，通过提供公共声明性编程范式，可插拔服务体系架构允许将服务原生类型和动作映射到平台原生类型和动作。这允许通过类型和模式发现的统一的应用设计体验。它还促进纯声明性数据流定义和设计，其涉及扩展平台的各种服务的组件以及用于

相应语义动作的原生代码的生成。

[0573] 根据实施例，可插拔服务体系架构还定义了用于插件的编译、生成、部署和运行时执行框架(统一应用设计服务)的公共接口。推荐引擎可以机器学习和推理所有插入服务的语义动作和模式，并且可以为分布式复杂数据流设计和开发做出跨服务的语义动作推荐。

[0574] 图51示出了根据实施例的用于基于经由外部功能接口提供的信息来识别用于提供用于数据流的推荐的模式的过程。

[0575] 如图51中所示，根据实施例，在步骤932，经由外部功能接口接收用于处理数据的元数据的一个或多个定义。

[0576] 在步骤934，处理经由外部功能接口接收的元数据，以识别关于所接收的元数据的信息，包括分类、语义动作、定义模式的模板或由所接收的元数据定义的服务中的一个或多个。

[0577] 在步骤936，经由外部功能接口接收的元数据存储在系统HUB中，其中系统HUB被更新以包括关于所接收的元数据的信息并且以扩展系统的功能能力，包括系统支持的类型、语义动作、模板和服务。

[0578] 在步骤938，基于经由外部功能接口在系统HUB中更新的信息，识别用于为数据流提供推荐的模式。

[0579] 基于策略的生命周期管理

[0580] 根据实施例，系统可以提供对于与特定时间快照有关的每个数据切片的数据治理(governance)功能，诸如例如起源(特定数据来自哪里)、族系(数据如何被获取/处理)、安全性(谁负责数据)、分类(数据所关于的内容)、影响(数据对业务的影响有多少)、保留(数据应当存活多长时间)和有效性(数据是否应当被排除/包括在分析/处理中)；然后可以将其用于制定生命周期决策和数据流推荐。

[0581] 管理数据生命周期的当前方法不涉及基于跨时间分区的数据特点的改变来跟踪数据演化(数据简档的改变或漂移)或治理相关功能。系统观察到的或导出的数据的特点(分类、改变的频率、改变的类型或过程中的使用)不用于对数据做出生命周期决策或推荐(保留、安全性、有效性、获取间隔)。

[0582] 根据实施例，系统可以提供图形用户界面，该图形用户界面可以基于族系跟踪来指示数据流的生命周期。生命周期可以示出数据已在何处被处理以及数据的处理期间是否发生任何错误；并且可以被示为数据的时间线视图(例如，数据集的数量、数据集的体积和数据集的使用)。界面可以提供数据的时间点快照，并且可以在数据被处理时提供数据的可视指示符。照此，界面可以基于生命周期(例如，性能度量或资源使用)实现数据的完整审计或数据的系统快照。

[0583] 根据实施例，系统可以基于(从所摄取的数据周期性地采样的)样本数据和用于由用户定义的应用处理而获取的数据来确定数据的生命周期。数据生命周期管理的一些方面是跨所摄取的数据的类别(即，流传输数据和批量数据(参考和增量))相似的。对于增量数据，系统可以使用调度的方法、日志收集方法和事件驱动的方法来获取数据的时间切片，并跨应用实例管理切片的分配，从而覆盖以下功能。

[0584] 根据实施例，系统可以在丢失的情况下根据系统HUB中管理的元数据使用跨层的族系来重建数据。

[0585] 例如,根据实施例,可以识别增量数据属性列或用户配置的设置,以获取增量数据并跨摄取数据维持高水印和低水印。查询或API和对应的参数(时间戳或Id列)可以与所摄取的数据相关联。

[0586] 根据实施例,系统可以维护跨层的族系信息,诸如例如边缘层中的查询或日志元数据、可扩展的I/O层中的每个摄取的主题/分区偏移、数据湖中的切片(文件分区)、用于使用这个数据的后续下游处理过的数据集的对过程族系的引用(产生数据和与数据相关联的参数的应用的具体执行实例)、"标记"为要发布到数据湖中的目标端点及其对应的数据切片的数据集的主题/分区偏移、以及被处理并发布到目标端点的发布作业执行实例和分区中的偏移。

[0587] 根据实施例,在层(例如,边缘、可扩展I/O、数据湖或发布)失败的情况下,可以从上游层重建数据或从源获取数据。

[0588] 根据实施例,系统可以执行其它生命周期管理功能。

[0589] 例如,根据实施例,在这些层中的每一层处执行和审计针对数据切片的安全性。数据切片可以对于被处理或访问被排除或被包括(如果已经排除)。这允许排除虚假或损坏的数据切片被处理。可以通过滑动窗口对数据切片执行保留策略。分析数据切片的影响(例如,将给定窗口的切片标记为在为季度报告构建的数据集市的上下文中具有影响力的能力)。

[0590] 根据实施例,通过标记系统中定义的功能或业务类型来对数据进行分类(例如,用功能类型(作为立方体或者维度或分层数据)连同(一个或多个)业务类型(例如,订单、客户、产品或时间)标记数据集)。

[0591] 根据实施例,系统可以执行包括从一个或多个HUB访问数据的方法。可以对数据进行采样,并且系统确定数据的时间切片并管理切片,包括访问系统的系统HUB,以获得关于采样数据的元数据。可以管理采样数据,以便跨系统中的一个或多个层进行族系跟踪。

[0592] 根据实施例,可以针对所摄取的数据管理关于样本数据的增量数据和参数。可以通过标记与样本数据相关联的数据类型来对数据进行分类。

[0593] 图52示出了根据实施例管理采样数据或访问数据,以用于跨一个或多个层的族系跟踪。

[0594] 例如,如图52中所示,根据实施例,系统可以用于从HUB 952(在这个示例中为Oracle数据库)和HUB 954(在这个示例中为S3或其它环境)接收数据。在边缘层处,从输入HUB接收的数据作为一个或多个主题提供给可扩展I/O层,以供数据流应用(例如,流水线、Lambda应用)使用(其中每个主题可以作为分布式分区提供)。

[0595] 根据实施例,通常由分区中的偏移表示的所摄取的数据可以由计算层标准化964,并且作为跨越系统的层的一个或多个时间切片写入数据湖。

[0596] 根据实施例,数据然后可以由下游数据流应用(例如,流水线、Lambda应用966、968)使用并最终发布970到一个或多个附加的主题960、962,并在其后发布到在一个或多个输出HUB(诸如在这个示例中,DBCS环境)处的目标端点(例如,表)。

[0597] 如图52中所示,根据实施例,在第一时间,数据重建和族系跟踪信息可以包括诸如例如起源(Hub 1、S3)、族系(Hub 1中的源实体)、安全性(所使用的连接凭证)或关于数据摄取的其它信息之类的信息。

[0598] 图53进一步示出了根据实施例管理采样数据或访问数据，以用于跨一个或多个层的族系跟踪。如图53中所示，在随后的时间，可以更新数据重建和族系跟踪信息，以包括诸如例如更新后的起源(→T1)、族系(→T1(摄取过程))或其它信息之类的信息。

[0599] 图54进一步示出了根据实施例管理采样数据或访问数据，以用于跨一个或多个层的族系跟踪。如图54中所示，在随后的时间，可以进一步更新数据重建和族系跟踪信息，以包括诸如例如更新后的起源(→E1)、族系(→E1(标准化))或其它信息之类的信息。

[0600] 根据实施例，可以跨越系统的层创建由一个或多个数据流应用(例如，流水线、Lambda应用)使用的时间切片972。在失败的情况下，例如写入数据湖时失败，系统可以确定一个或多个未处理的数据切片，并整体上或递增地完成该数据切片的处理。

[0601] 图55进一步示出了根据实施例管理采样数据或访问数据，以用于跨一个或多个层的族系跟踪。如图55中所示，在随后的时间，可以进一步更新数据重建和族系跟踪信息，并且创建附加的时间切片，以包括诸如例如更新后的族系(→E11(应用1))、安全性(执行应用1的角色)或其它信息之类的信息。

[0602] 图56进一步示出了根据实施例管理采样数据或访问数据，以用于跨一个或多个层的族系跟踪。如图56中所示，在随后的时间，可以进一步更新数据重建和族系跟踪信息，并且创建附加的时间切片，以包括诸如例如更新后的族系(→E12(应用2))、安全性(执行应用2的角色)或其它信息之类的信息。

[0603] 图57进一步示出了根据实施例管理采样数据或访问数据，以用于跨一个或多个层的族系跟踪。如图57中所示，在随后的时间，可以进一步更新数据重建和族系跟踪信息，并且创建附加的时间切片，以包括诸如例如更新后的族系(→T2(发布))、安全性(执行发布到I/O层的角色)或其它信息之类的信息。

[0604] 图58进一步示出了根据实施例管理采样数据或访问数据，以用于跨一个或多个层的族系跟踪。如图58中所示，在随后的时间，可以进一步更新数据重建和族系跟踪信息，以将数据的输出反映到目标端点976。

[0605] 数据生命周期管理

[0606] 根据实施例，上文描述的基于族系跟踪的数据生命周期管理解决了若干功能区域，这些功能区域中的一些功能区域(访问控制、保留、有效性)可以由用户配置、一些(起源、族系)被导出以及其它(分类、影响)使用机器学习算法。例如，数据管理适用于(从摄取的数据中周期性地采样的)样本数据和用于由用户定义的应用处理而获取的数据。数据生命周期管理的一些方面跨所摄取的数据的类别(即，流传输数据和批量数据(参考和增量))是相似的。对于增量数据，DFML使用调度的方法、日志收集方法和事件驱动的方法，以获取数据的时间切片，并跨应用实例管理切片的分配，从而覆盖以下功能：

[0607] 在丢失的情况下根据系统HUB中管理的元数据使用跨层的族系来重建数据。

[0608] 识别增量数据属性列或用户配置的设置，以获取增量数据并跨摄取维持高水印和低水印。

[0609] 为每个摄取关联查询或API和对应的参数(时间戳或Id列)。

[0610] 维持跨层的族系信息。在边缘层中的查询或日志元数据。可扩展I/O层中的每个摄取的主题/分区偏移。数据湖中的切片(文件分区)。使用该数据的用于所有后续下游处理的数据集的对过程族系的引用(产生数据和与它相关联的参数的应用的具体执行实例)。被

“标记”为发布到数据湖中的目标端点及其对应的数据切片的数据集的主题/分区偏移。被处理并发布到目标端点的发布作业执行实例以及分区中的偏移。

[0611] 在层失败的情况下，可以从上游层重建数据或者从源获取数据。在这些层中的每一层处执行和审计数据切片的安全性。可以对于被处理或访问来排除或包括(如果已经排除)数据切片。这允许排除虚假或损坏的数据切片被处理。可以通过滑动窗口对数据切片执行保留策略。针对数据切片分析影响(例如，将给定窗口的切片标记为在为季度报告构建的数据集市的上下文中具有影响力的能力)。

[0612] 通过标记系统中定义的功能或业务类型来对数据进行分类(例如，用功能类型(作为立方体或者维度或分层数据)连同(一个或多个)业务类型(例如，订单、客户、产品或时间)标记数据集)。

[0613] 图59示出了根据实施例的用于管理采样数据或所访问的数据以用于跨一个或多个层的族系跟踪的过程。

[0614] 如图59中所示，根据实施例，在步骤982，从一个或多个HUB访问数据。

[0615] 在步骤983，对所访问的数据进行采样。

[0616] 在步骤984，针对采样数据或所访问的数据识别时间切片。

[0617] 在步骤985，访问系统HUB，以获得关于由时间切片表示的采样数据或所访问的数据的元数据。

[0618] 在步骤986，确定关于由时间切片表示的采样数据或所访问的数据的分类信息。

[0619] 在步骤987，管理由时间切片表示的采样数据或所访问的数据，以用于跨系统中的一个或多个层的族系跟踪。

[0620] 可以使用包括根据本公开的教导编程的一个或多个处理器、存储器和/或计算机可读存储介质的一个或多个常规的通用或专用数字计算机、计算设备、机器或微处理器来实现本发明的实施例。如对于软件领域的技术人员来说明显的，基于本公开的教导，熟练的程序员可以容易地准备适当的软件编码。

[0621] 在一些实施例中，本发明包括计算机程序产品，该计算机程序产品是具有存储在其上/其中的指令的非暂态计算机可读存储介质(媒介)，可以使用指令对计算机进行编程以执行本发明的过程中的任何过程。存储介质的示例可以包括但不限于软盘、光盘、DVD、CD-ROM、微驱动器和磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪存设备、磁卡或者光卡、纳米系统(包括分子存储器IC)、或者适用于指令和/或数据的非暂态存储的其它类型的存储介质或设备。

[0622] 已经出于说明和描述的目的提供了本发明的前述描述。它并非旨在是详尽的或将本发明限制到所公开的精确形式。许多修改和变型对于本领域技术人员来说是明显的。

[0623] 例如，虽然上述实施例中的若干实施例说明了诸如Wolfram、Yago、Chronos和Spark之类的产品的使用，以执行各种计算，以及诸如例如BDP、SFDC和S3之类的数据源，以充当数据的源或目标，但是本文描述的实施例还可以与提供类似类型的功能的其它类型的产品和数据源一起使用。

[0624] 此外，虽然上述实施例中的若干实施例示出了各种实施例的组件、层、对象、逻辑或其它特征，但是这些特征可以被提供为可以由计算机系统或其它处理设备执行的软件或程序代码。

[0625] 选择和描述实施例是为了最好地解释本发明的原理及其实际应用,从而使得本领域其他技术人员能够对于各种实施例以及利用适合于预期的特定用途的各种修改来理解本发明。修改和变型包括所公开特征的任何相关组合。本发明的范围旨在由以下权利要求及其等价物定义。

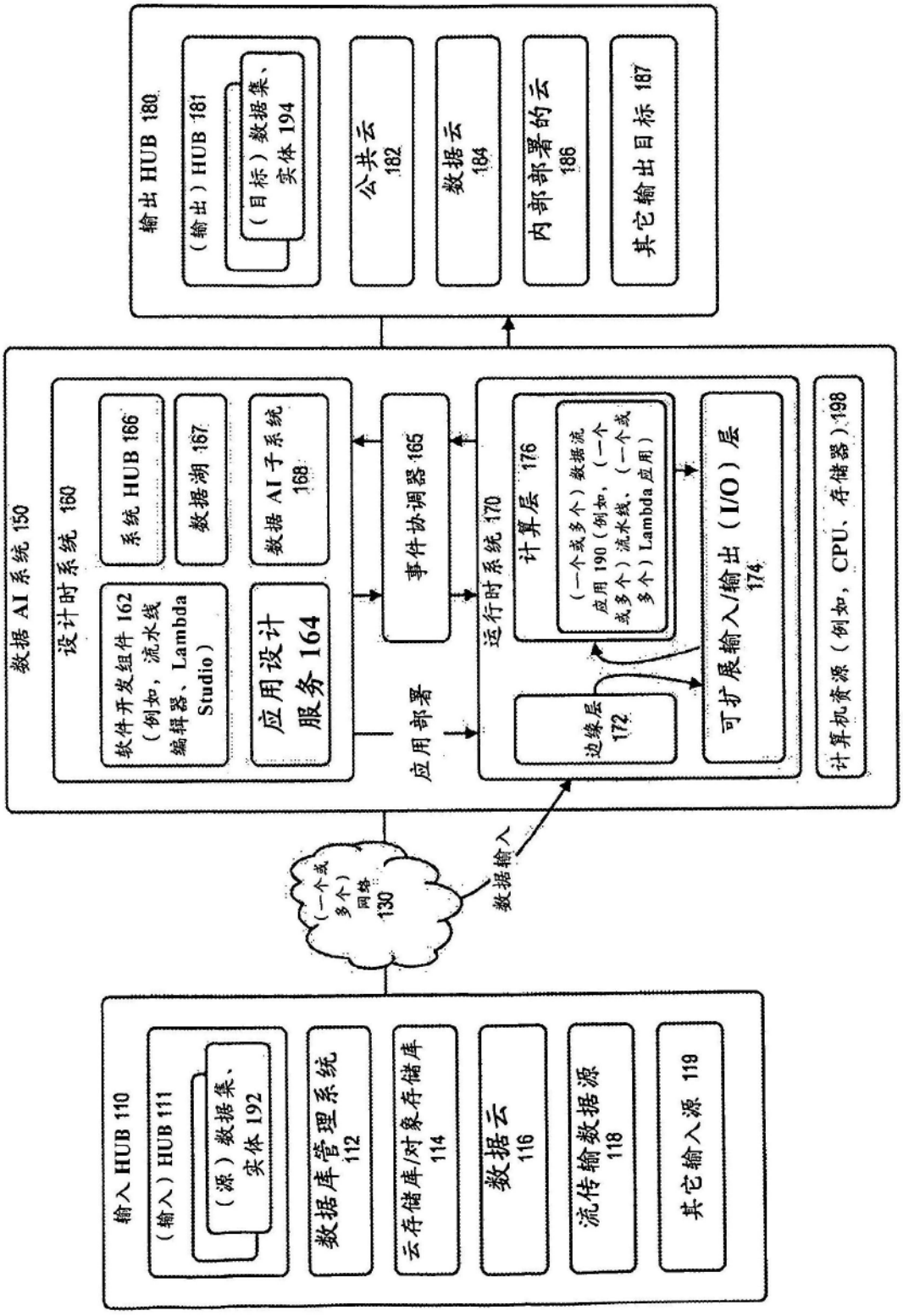

输出 HUB 180
（输出）HUB 181
（目标）数据集、实体 194
公共云 182
数据云 184
内部部署的云 186
其它输出目标 187
数据 AI 系统 150
设计时系统 160
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
应用设计服务 164
系统 HUB 166
数据湖 167
数据 AI 子系统 168
事件协调器 165
应用部署
运行时系统 170
计算层 176
（一个或多个）数据流应用 190（例如，（一个或多个）流水线、（一个或多个）Lambda 应用）
边缘层 172
可扩展输入/输出（I/O）层 174
计算机资源（例如，CPU、存储器）198
（一个或多个）网络 130
数据输入
输入 HUB 110
（输入）HUB 111
（源）数据集、实体 192
数据库管理系统 112
云存储库/对象存储库 114
数据云 116
流传输数据源 118
其它输入源 119

图1

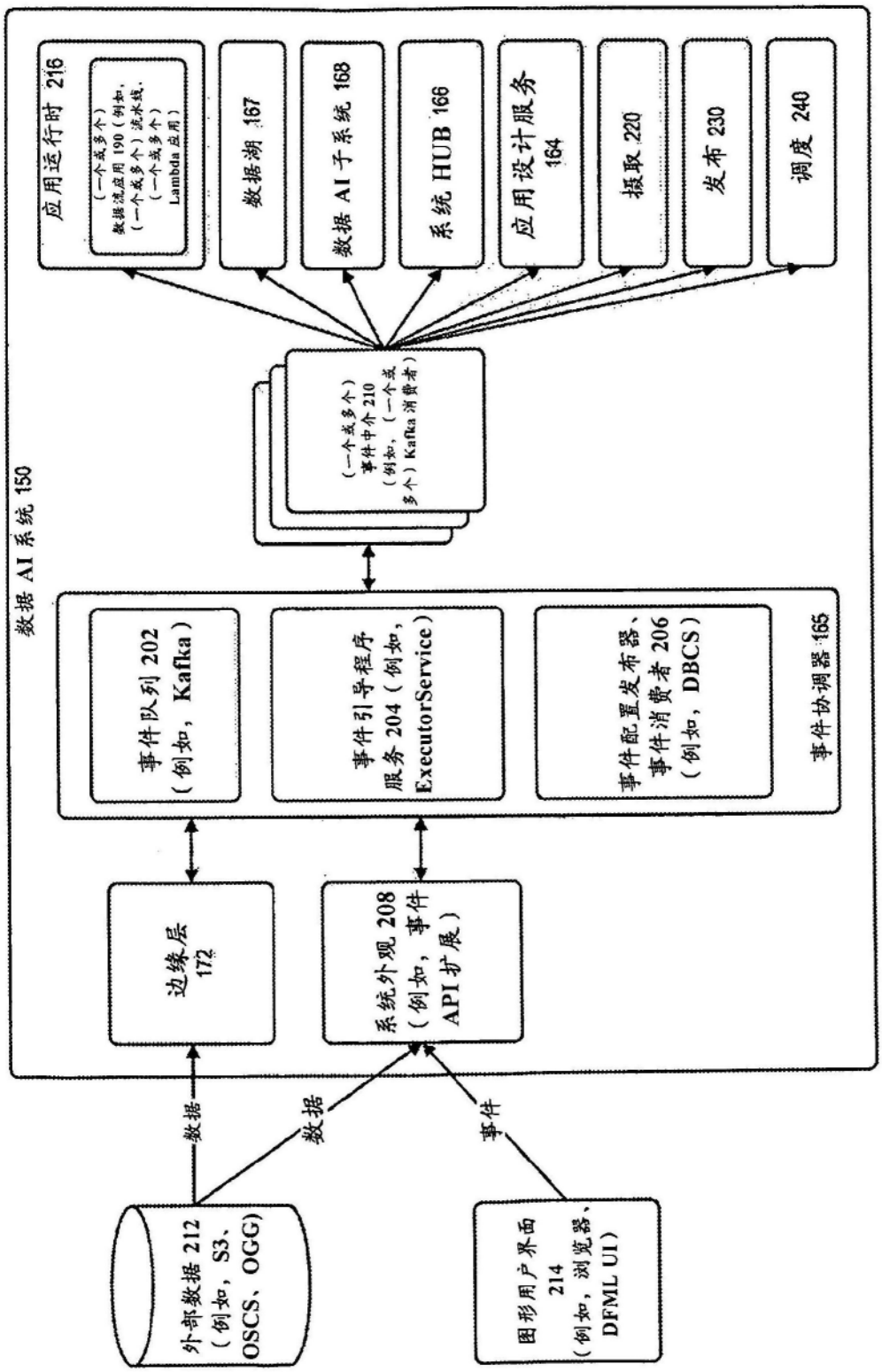

数据 AI 系统 150
应用运行时 216
（一个或多个）数据流应用 190（例如，（一个或多个）流水线，（一个或多个）Lambda 应用）
数据湖 167
数据 AI 子系统 168
系统 HUB 166
应用设计服务 164
摄取 220
发布 230
调度 240
（一个或多个）事件中介 210（例如，（一个或多个）Kafka 消费者）
事件协调器 165
事件队列 202（例如，Kafka）
事件引导程序服务 204（例如，ExecutorService）
事件配置发布器、事件消费者 206（例如，DBCS）
边缘层 172
系统外观 208（例如，事件 API 扩展）
数据
数据
事件
外部数据 212（例如，S3、OSCS、OGG）
图形用户界面 214（例如，浏览器、DFML UI）

图2

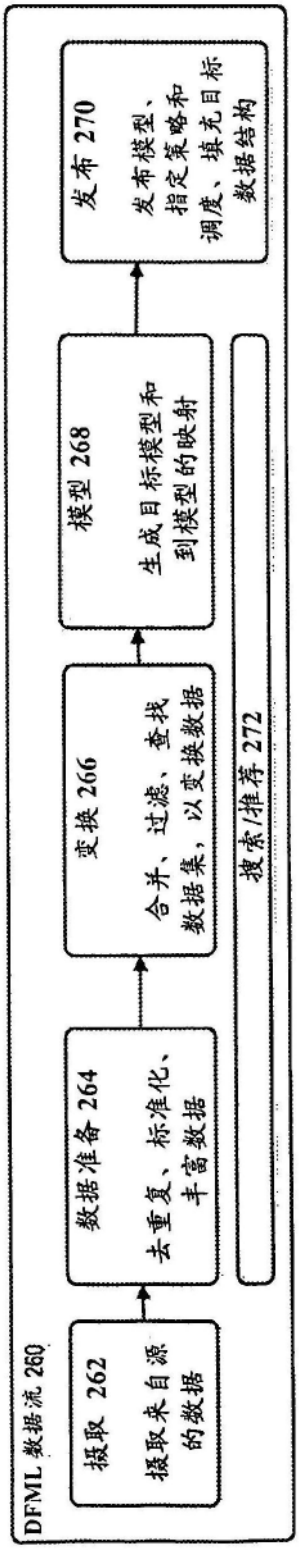

DFML 数据流 260
摄取 262
摄取来自源的数据
数据准备 264
去重复、标准化、丰富数据
变换 266
合并、过滤、查找数据集，以变换数据
模型 268
生成目标模型和到模型的映射
发布 270
发布模型、指定策略和调度、填充目标数据结构
搜索/推荐 272

图3

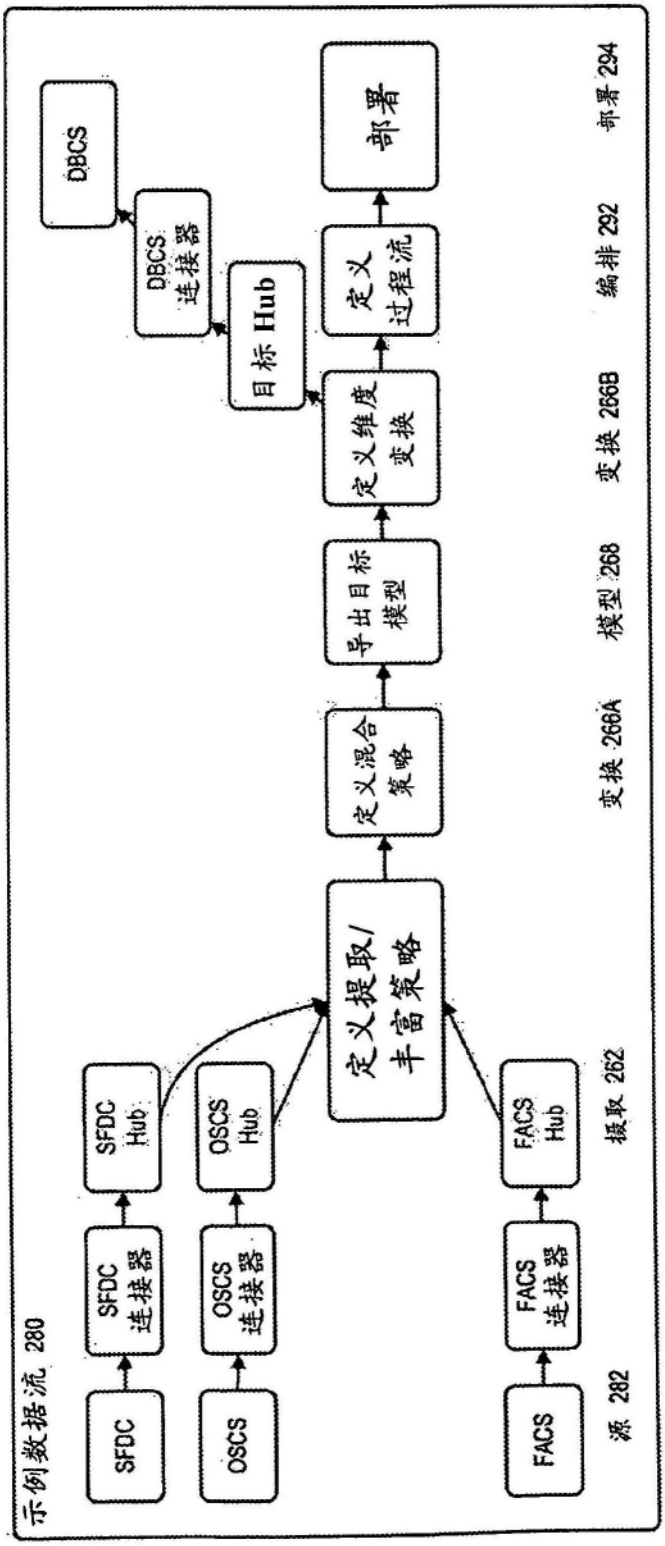

示例数据流 280
SFDC
OSCS
FACS
SFDC 连接器
OSCS 连接器
FACS 连接器
SFDC Hub
OSCS Hub
FACS Hub
定义提取/丰富策略
定义混合策略
导出目标模型
定义维度变换
目标 Hub
DBCS 连接器
DBCS
定义过程流
部署
源 282
摄取 262
变换 266A
模型 268
变换 266B
编排 292
部署 294

图4

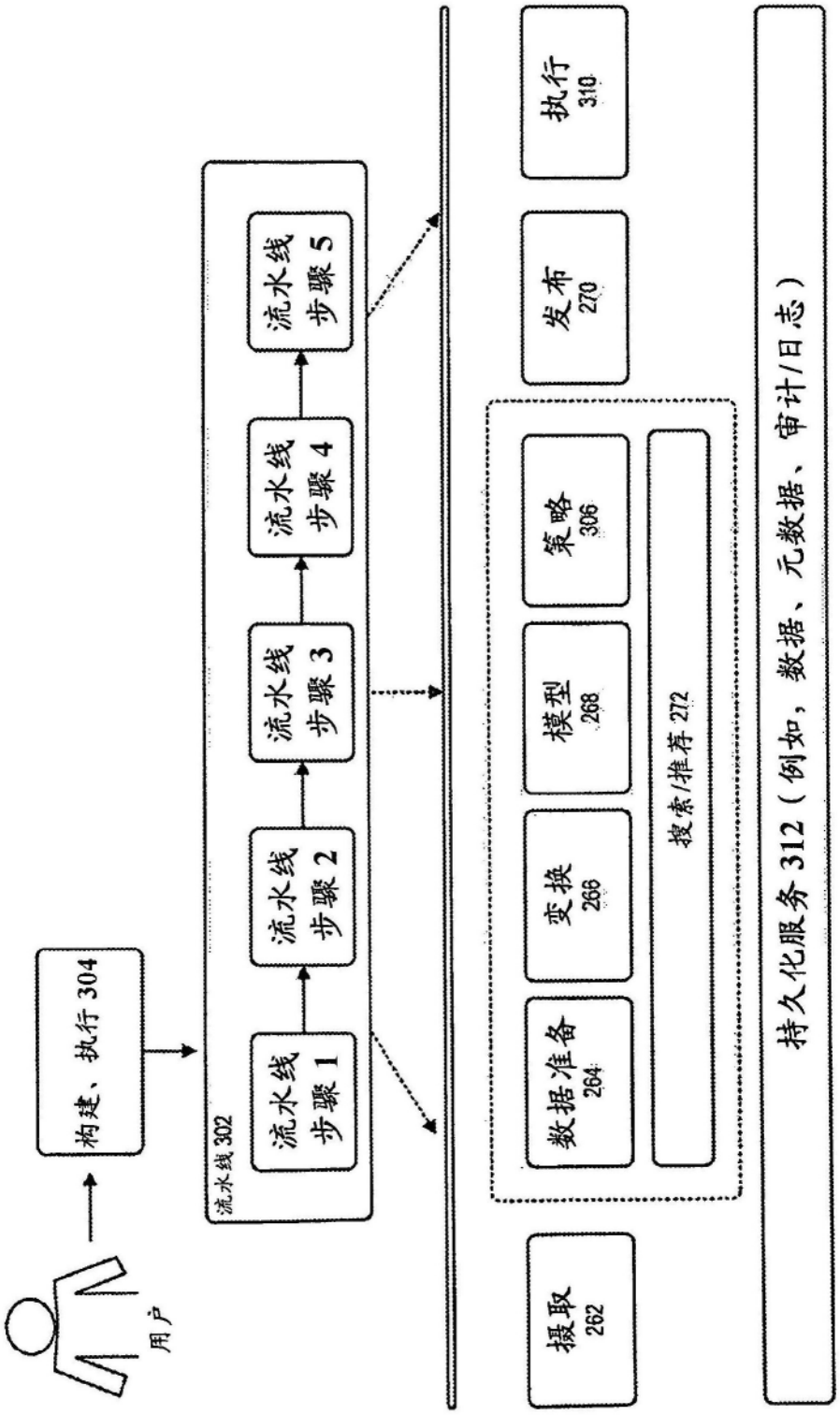

用户
构建、执行 304
流水线 302
流水线步骤 1
流水线步骤 2
流水线步骤 3
流水线步骤 4
流水线步骤 5
摄取 262
数据准备 264
变换 266
模型 268
策略 306
搜索/推荐 272
发布 270
执行 310
持久化服务 312（例如，数据、元数据、审计/日志）

图5

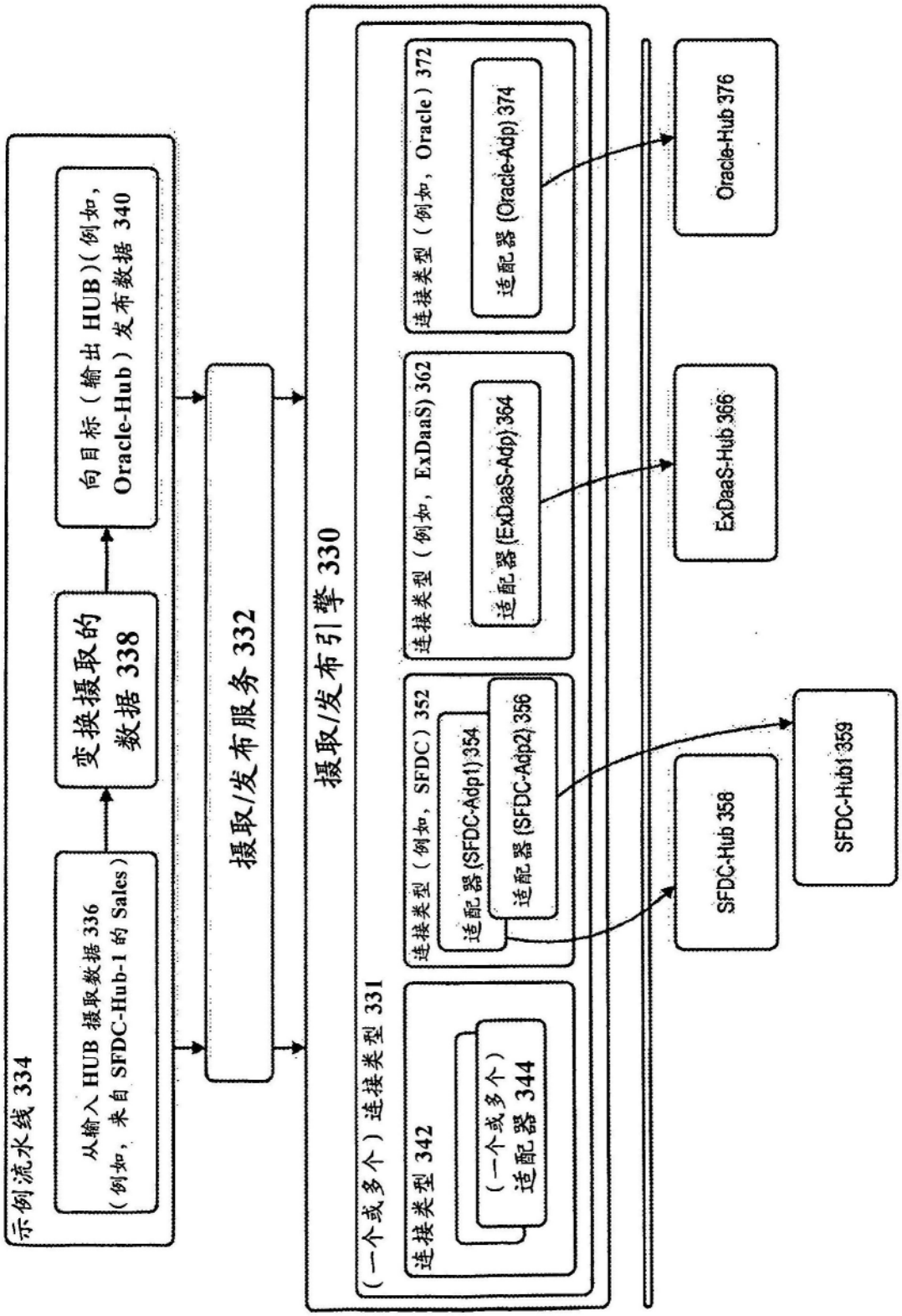

示例流水线 334
从输入 HUB 摄取数据 336
（例如，来自 SFDC-Hub-1 的 Sales）
变换摄取的数据 338
向目标（输出 HUB）（例如，Oracle-Hub）发布数据 340
摄取/发布服务 332
摄取/发布引擎 330
（一个或多个）连接类型 331
连接类型 342
（一个或多个）适配器 344
连接类型（例如，SFDC）352
适配器 (SFDC-Adp1) 354
适配器 (SFDC-Adp2) 356
连接类型（例如，ExDaaS）362
适配器 (ExDaaS-Adp) 364
连接类型（例如，Oracle）372
适配器 (Oracle-Adp) 374
SFDC-Hub 358
SFDC-Hub1 359
ExDaaS-Hub 366
Oracle-Hub 376

图6

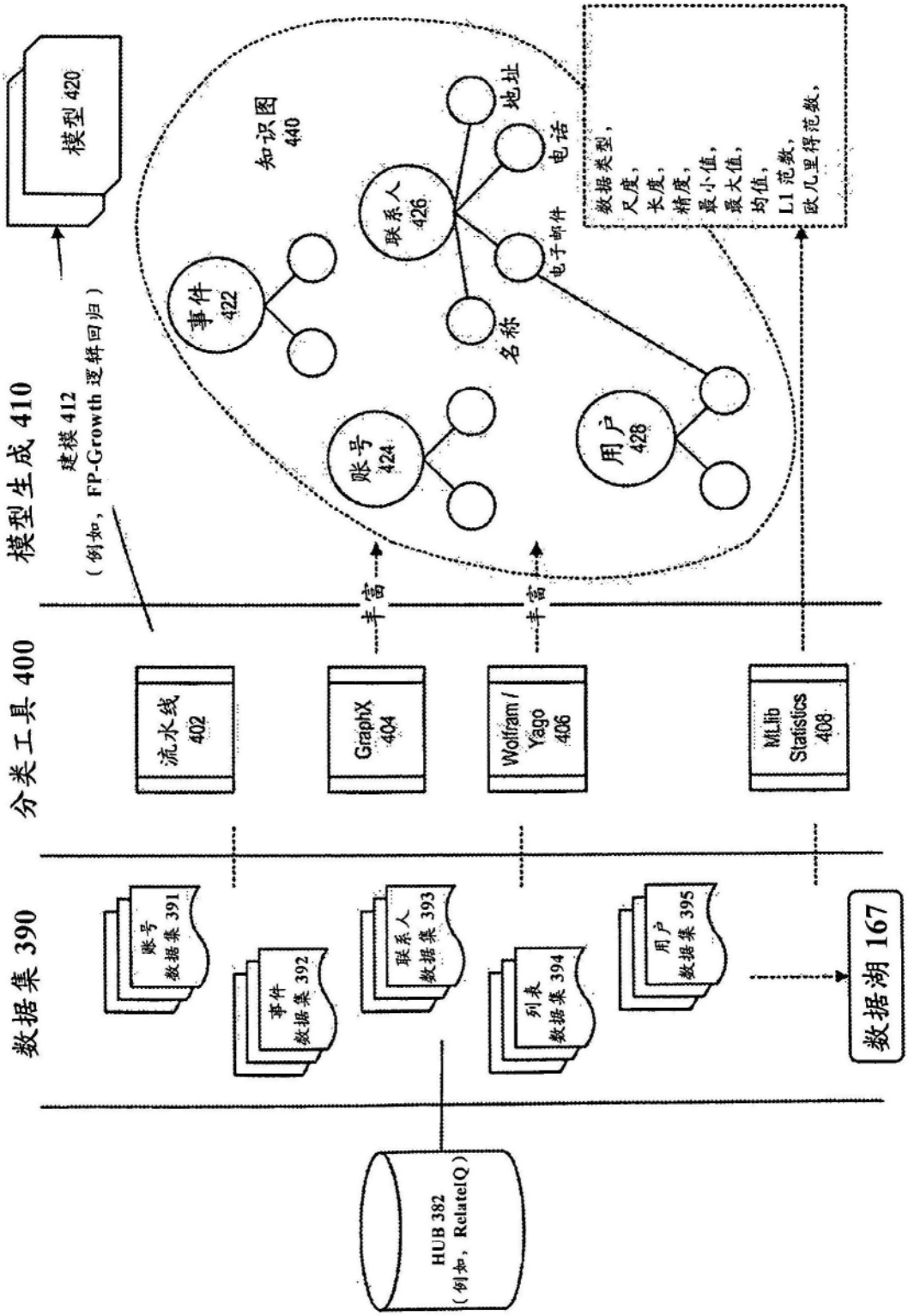

模型 420
模型生成 410
建模 412
（例如，FP-Growth 逻辑回归）
知识图
440
事件
422
账号
424
联系人
426
地址
电话
电子邮件
名称
用户
428
数据类型，
尺度，
长度，
精度，
最小值，
最大值，
均值，
L1 范数，
欧几里得范数，
丰富
丰富
分类工具 400
流水线
402
GraphX
404
Wolfram /
Yago
406
MLlib
Statistics
408
数据集 390
账号
数据集 391
事件
数据集 392
联系人
数据集 393
列表
数据集 394
用户
数据集 395
数据湖 167
HUB 382
（例如，RelateIQ）

图7

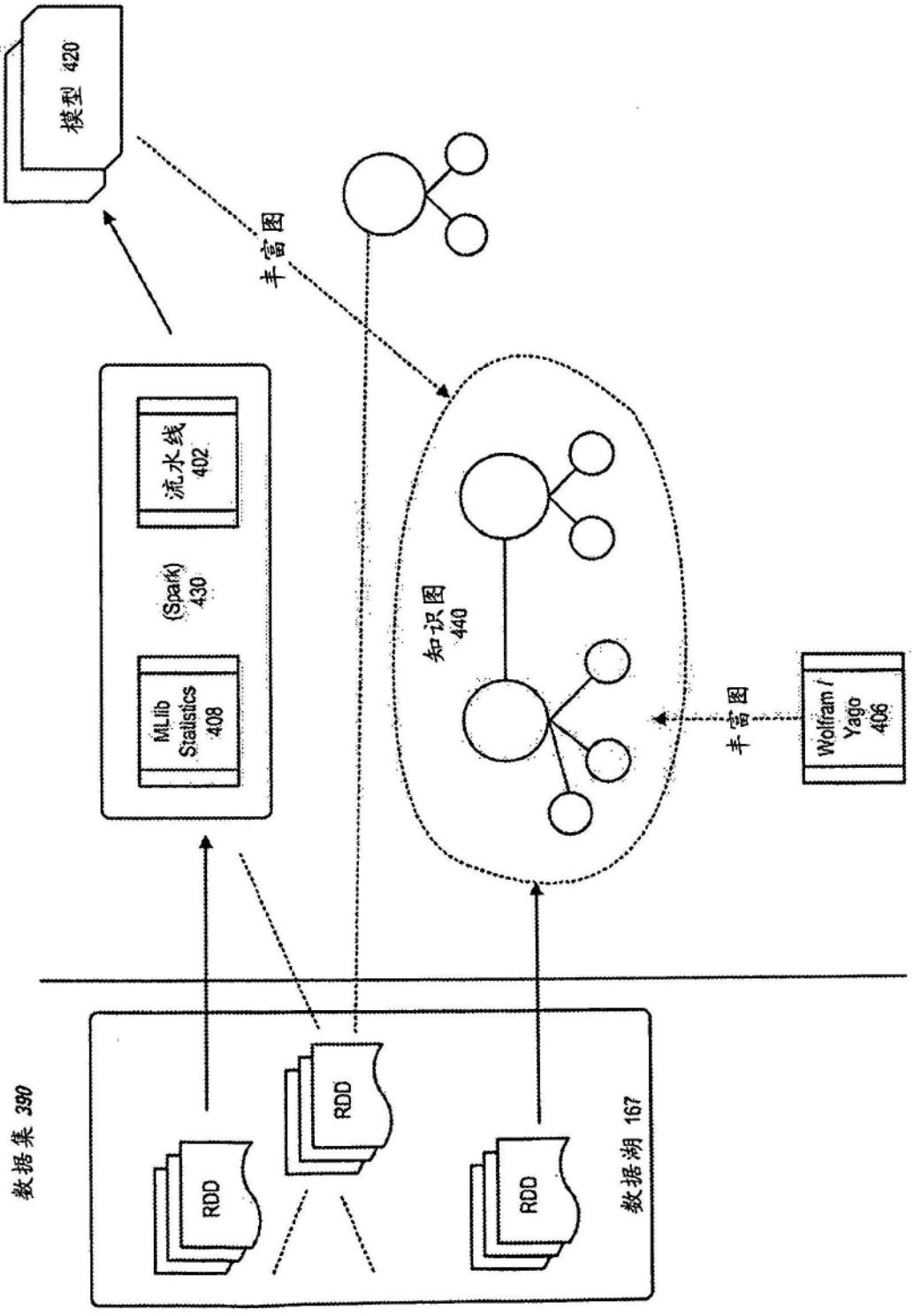

模型 420
丰富图
流水线 402
(Spark) 430
MLlib Statistics 408
知识图 440
丰富图
Wolfram / Yago 406
数据集 390
RDD
RDD
RDD
数据湖 167

图8

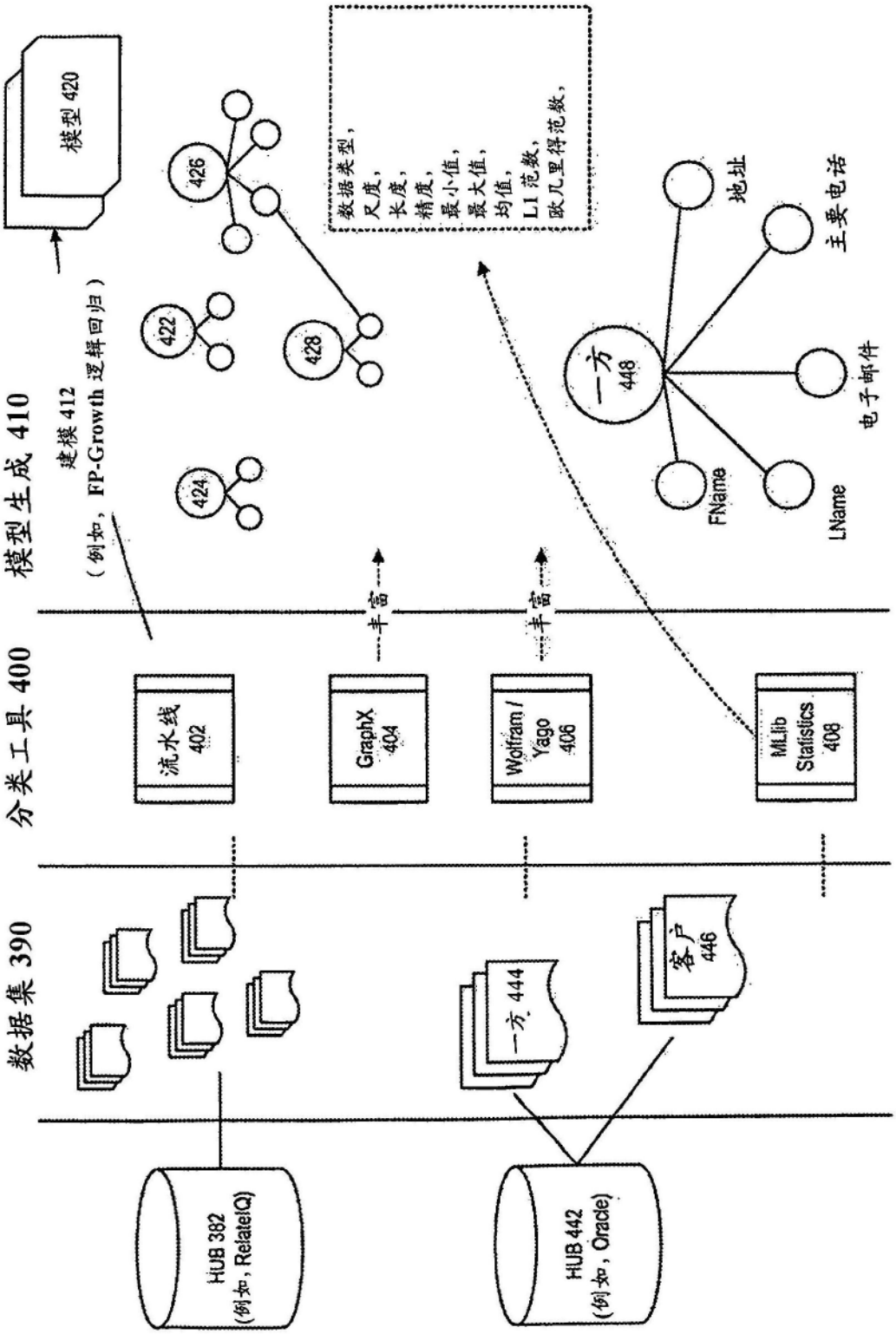

数据集 390
分类工具 400
模型生成 410
模型 420
建模 412
（例如，FP-Growth 逻辑回归）
426
422
424
428
数据类型，
尺度，
长度，
精度，
最小值，
最大值，
均值，
L1 范数，
欧几里得范数，
流水线 402
GraphX 404
Wolfram / Yago 406
MLlib Statistics 408
丰富
丰富
一方 448
地址
主要电话
电子邮件
FName
LName
一方 444
客户 446
HUB 382
（例如，RelateIQ）
HUB 442
（例如，Oracle）

图9

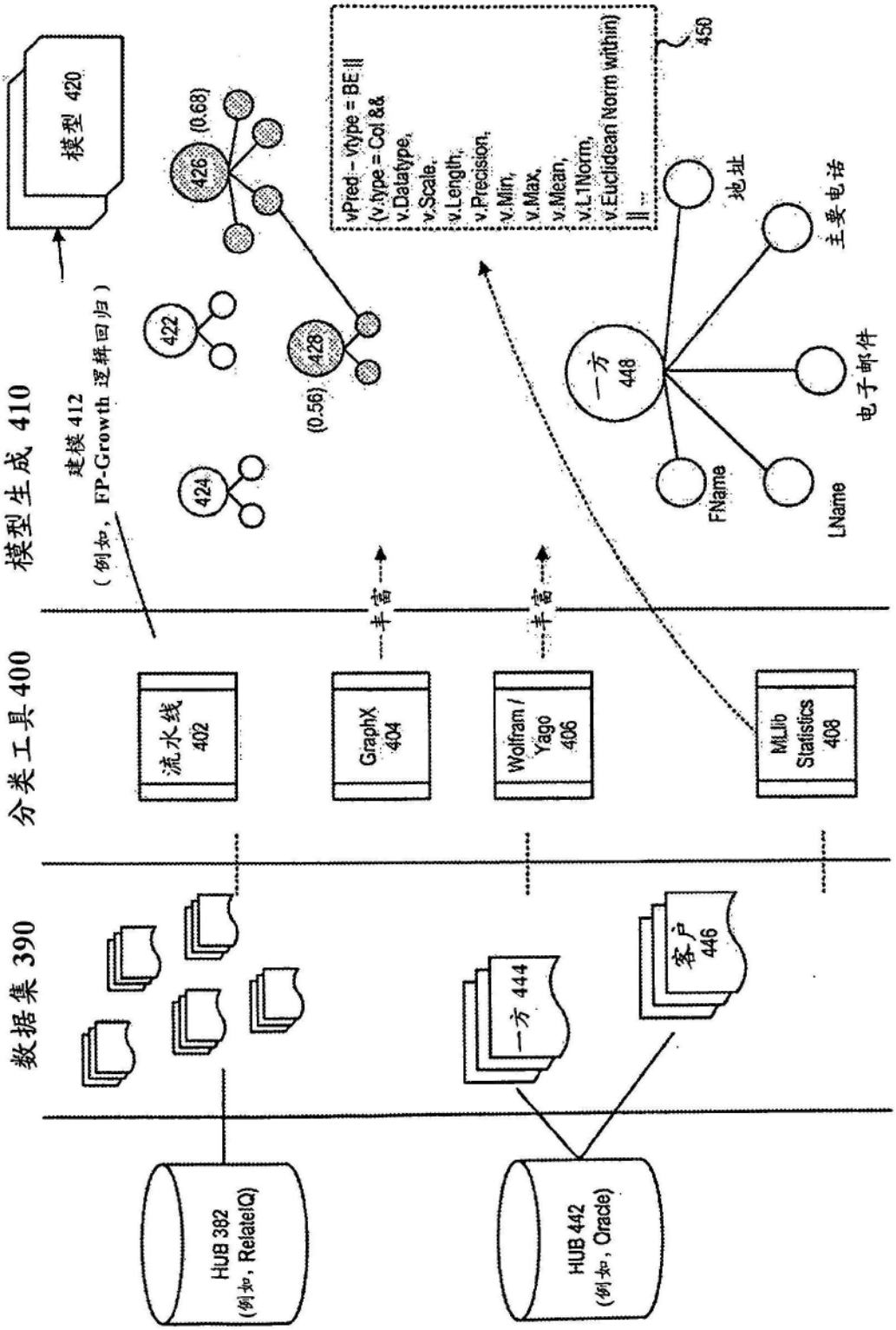

数据集 390
分类工具 400
模型生成 410
模型 420
建模 412
（例如，FP-Growth 逻辑回归）
流水线 402
GraphX 404
Wolfram / Yago 406
MLlib Statistics 408
丰富
丰富
HUB 382
（例如，RelateIQ）
HUB 442
（例如，Oracle）
一方 444
客户 446
一方 448
FName
LName
地址
主要电话
电子邮件
426
(0.68)
422
424
428
(0.56)
vPred – vtype = BE ||
(v.type = Col &&
v.Datatype,
v.Scale,
v.Length,
v.Precision,
v.Min,
v.Max,
v.Mean,
v.L1Norm,
v.Euclidean Norm within)
||...
450

图10

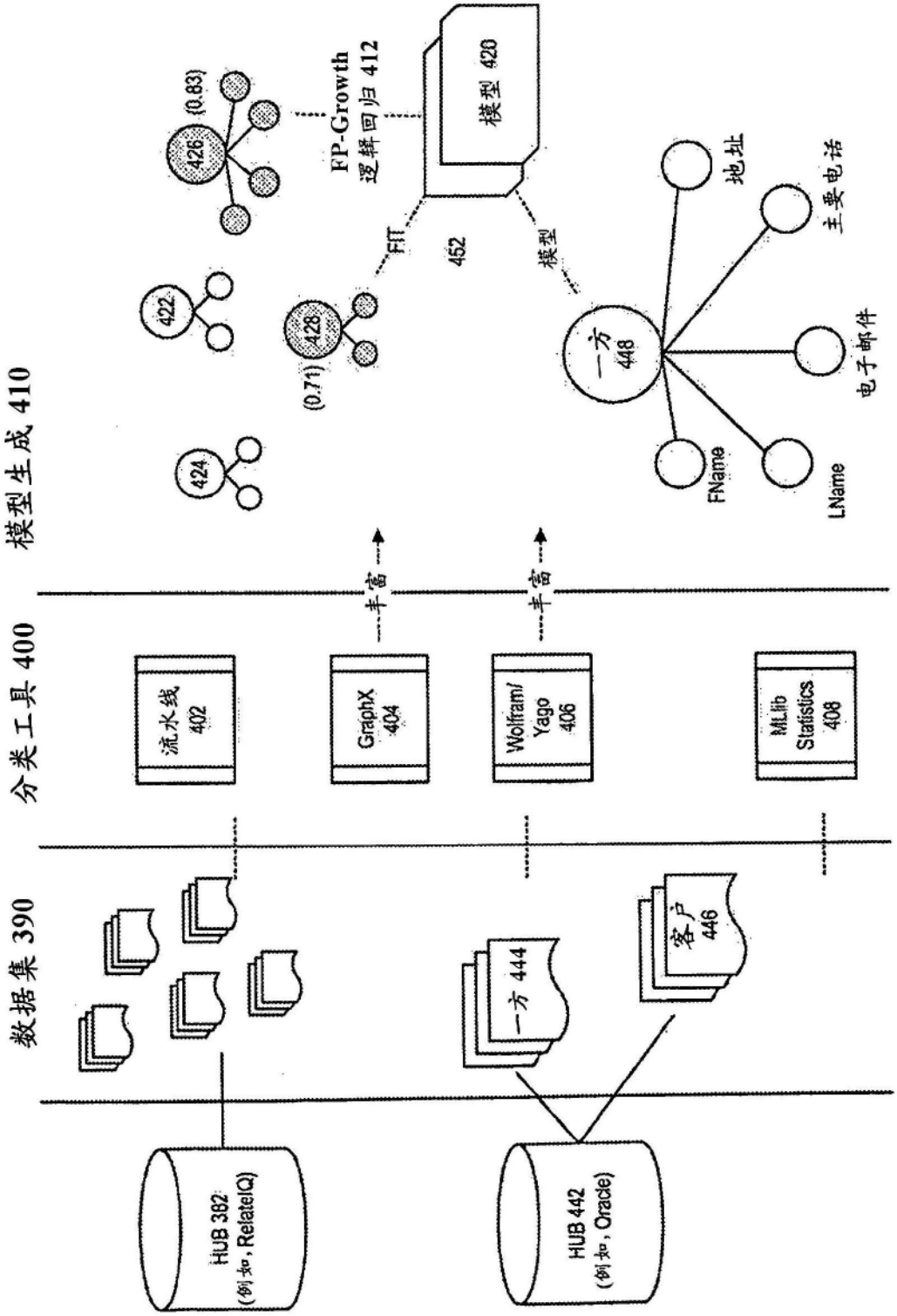

模型生成 410
分类工具 400
数据集 390
流水线 402
GraphX 404
Wolfram/ Yago 406
MLlib Statistics 408
丰富
丰富
HUB 382 (例如, RelateIQ)
HUB 442 (例如, Oracle)
一方 444
客户 446
一方 448
FP-Growth 逻辑回归 412
模型 420
FIT
452
模型
426
(0.83)
422
428
(0.71)
424
地址
主要电话
电子邮件
FName
LName

图11

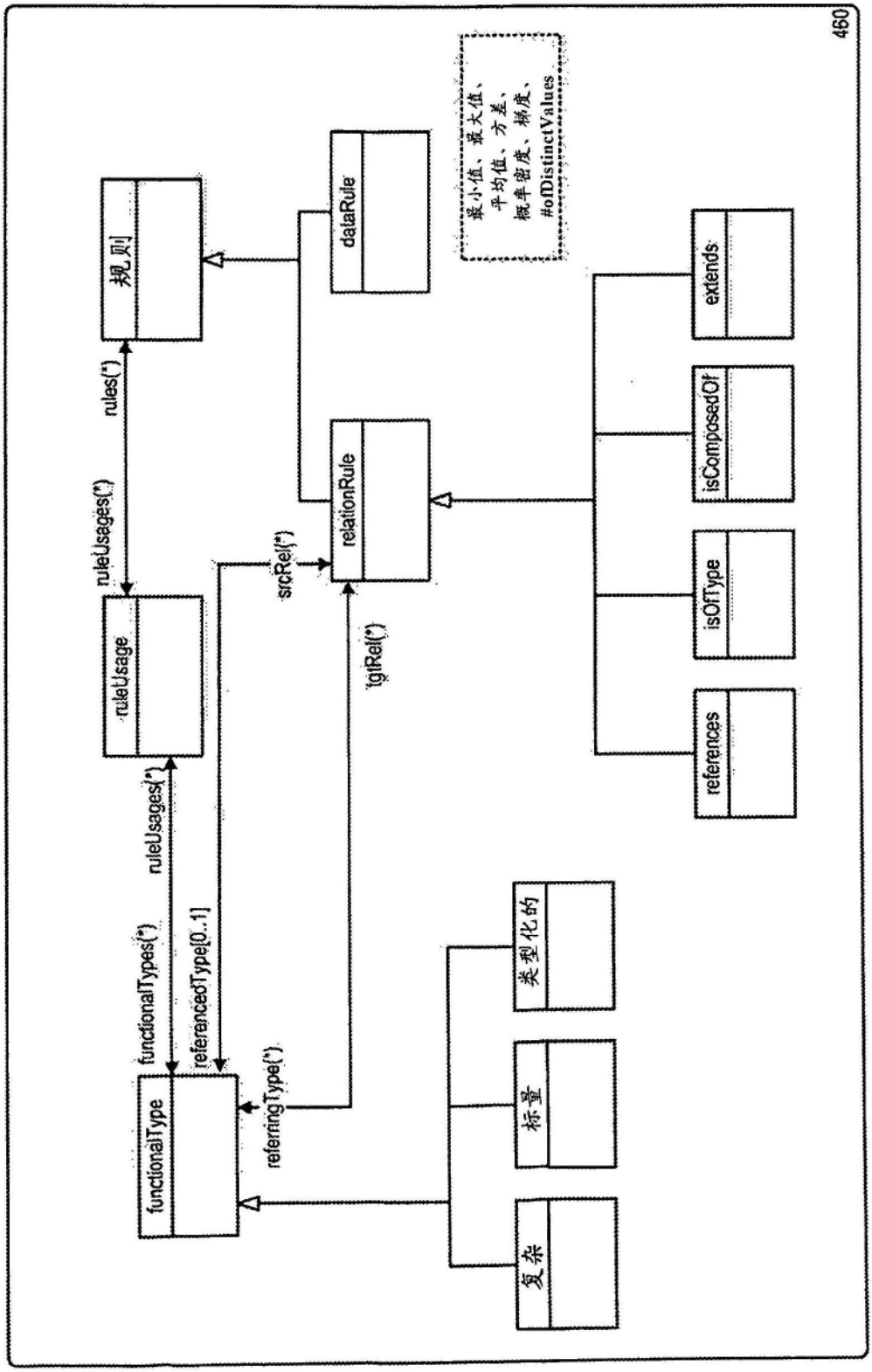

460
规则
dataRule
最小值、最大值、
平均值、方差、
概率密度、梯度、
#ofDistinctValues
rules(*)
ruleUsages(*)
relationRule
srcRel(*)
tgtRel(*)
ruleUsage
extends
isComposedOf
isOfType
references
ruleUsages(*)
functionalTypes(*)
referencedType[0..1]
referringType(*)
functionalType
类型化的
标量
复杂

图12

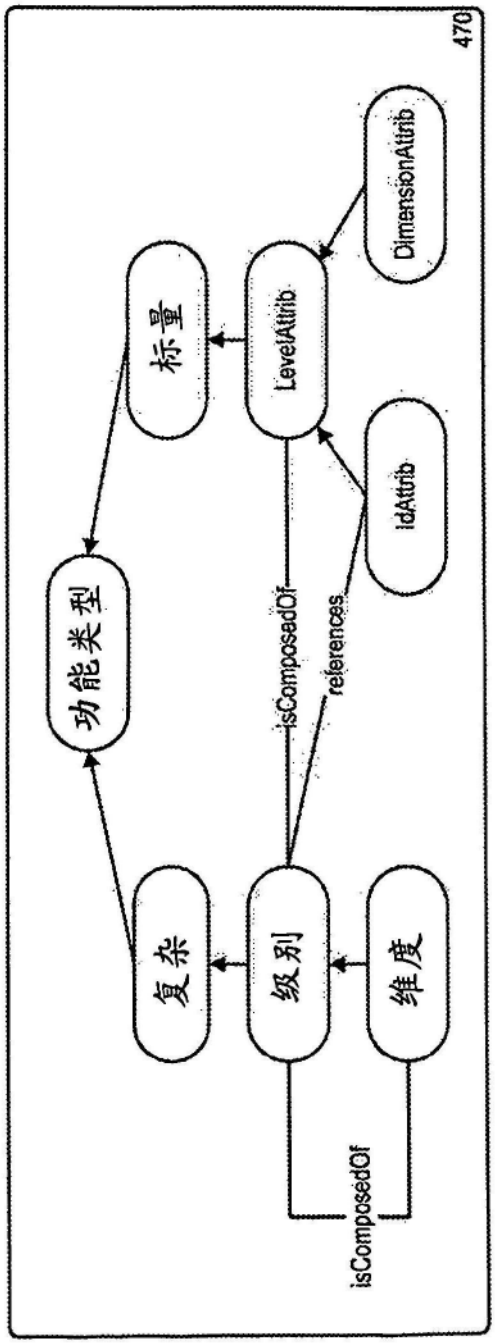

470
DimensionAttrib
标量
LevelAttrib
idAttrib
功能类型
isComposedOf
references
复杂
级别
维度
isComposedOf

图13

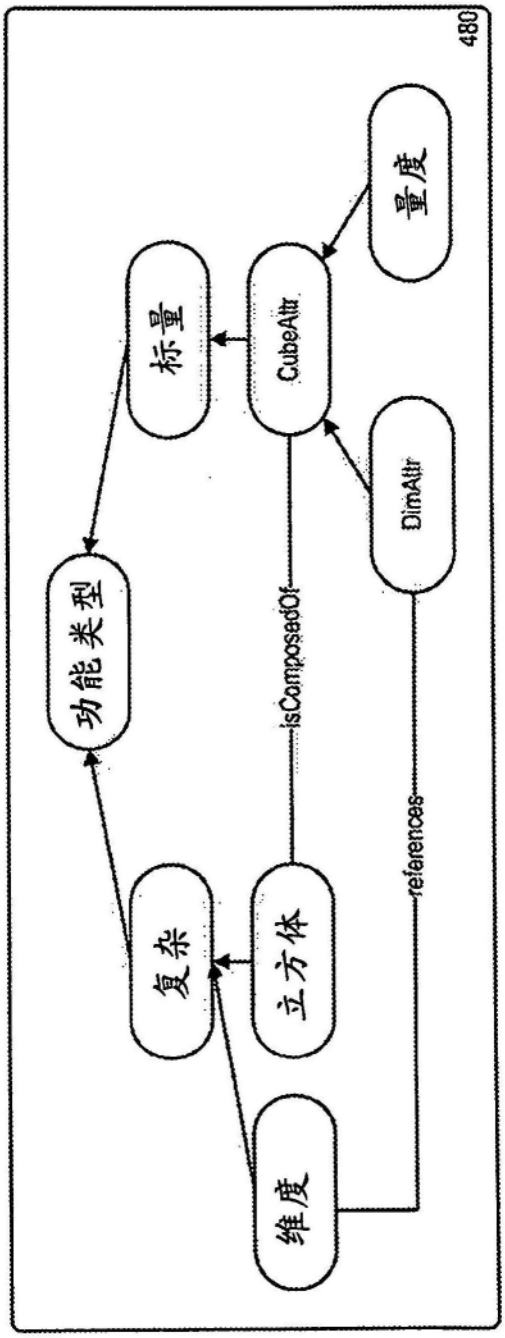

480
量度
标量
CubeAttr
DimAttr
功能类型
isComposedOf
references
复杂
立方体
维度

图14

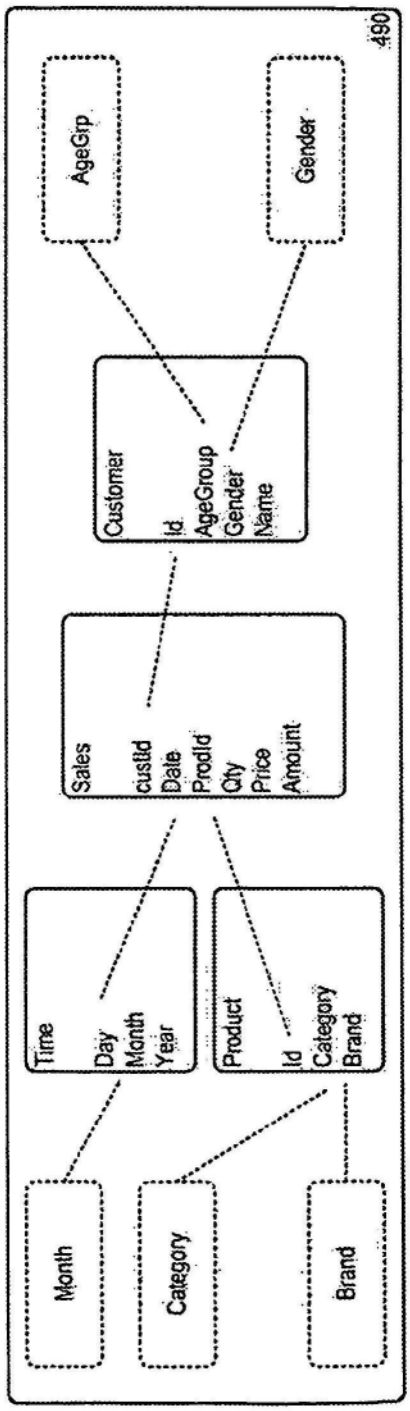

490
AgeGrp
Gender
Customer
Id
AgeGroup
Gender
Name
Sales
custId
Date
ProdId
Qty
Price
Amount
Time
Day
Month
Year
Product
Id
Category
Brand
Month
Category
Brand

图15

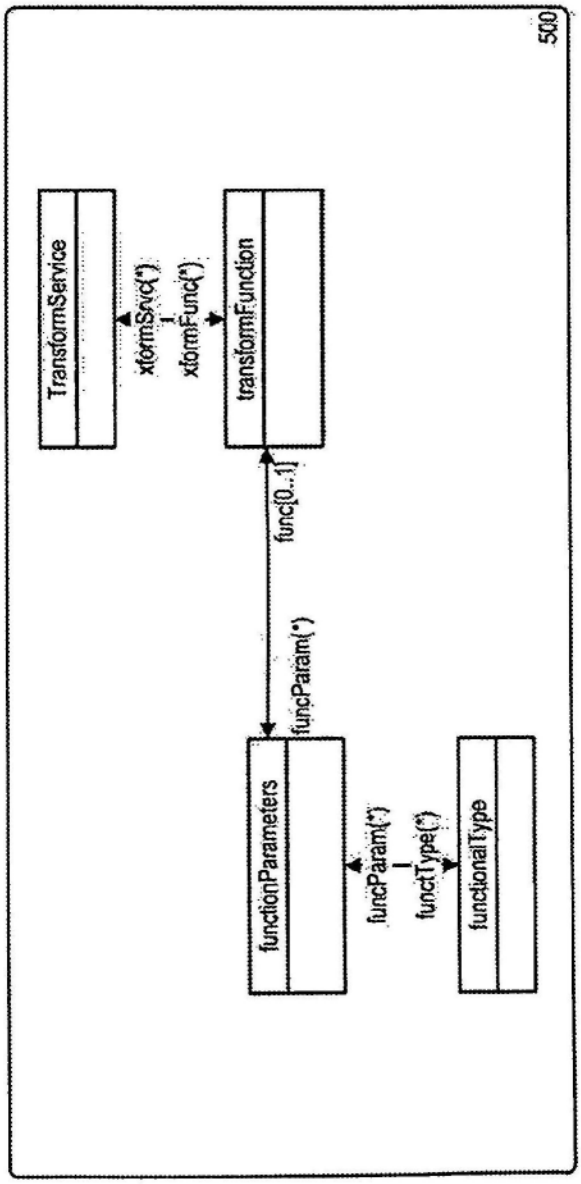

500
TransformService
xformSvc(*)
xformFunc(*)
transformFunction
func[0..1]
funcParam(*)
functionParameters
funcParam(*)
functType(*)
functionalType

图16

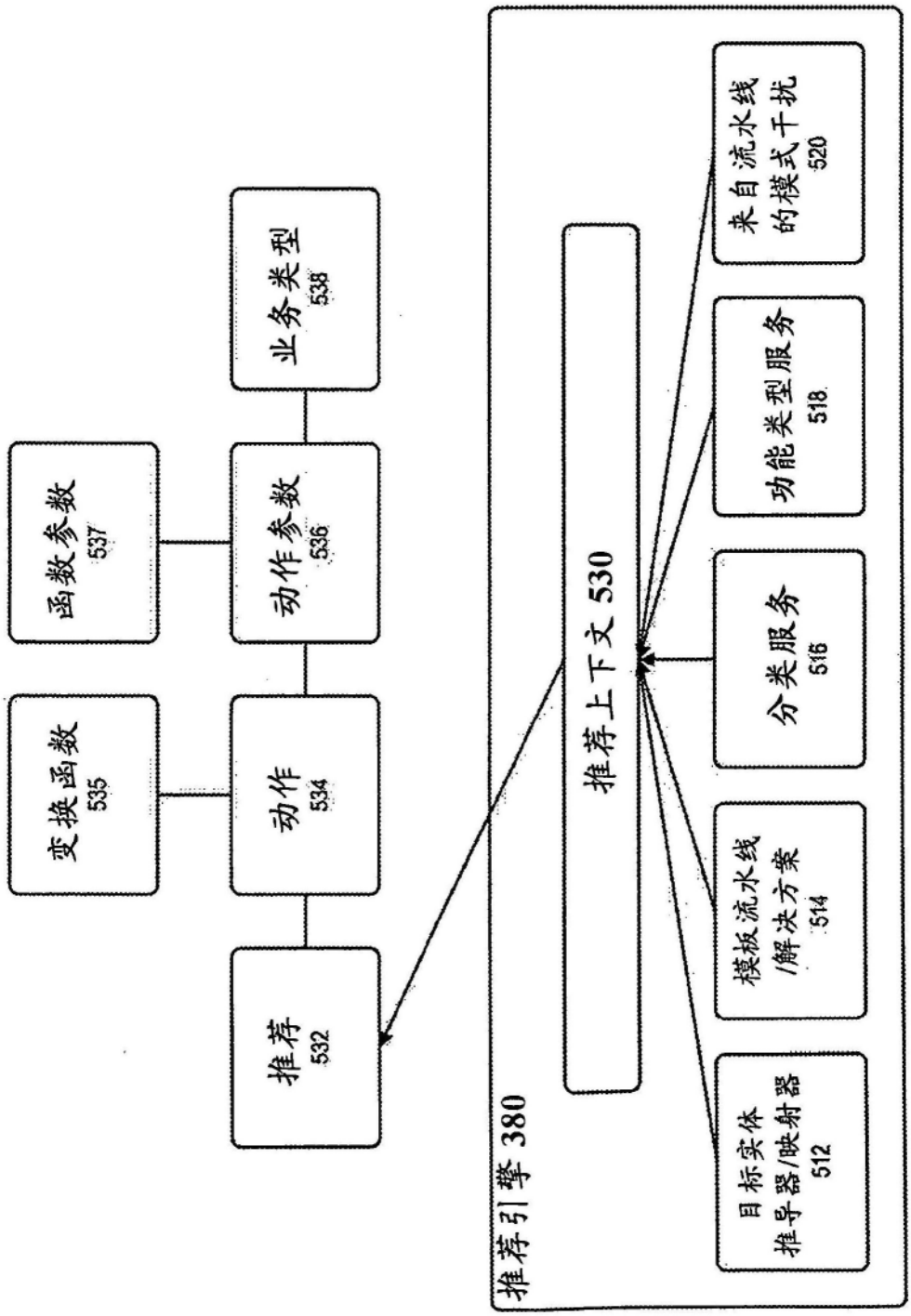

推荐引擎 380
推荐上下文 530
来自流水线的模式干扰 520
功能类型服务 518
分类服务 516
模板流水线/解决方案 514
目标实体推导器/映射器 512
推荐 532
动作 534
变换函数 535
动作参数 536
函数参数 537
业务类型 538

图17

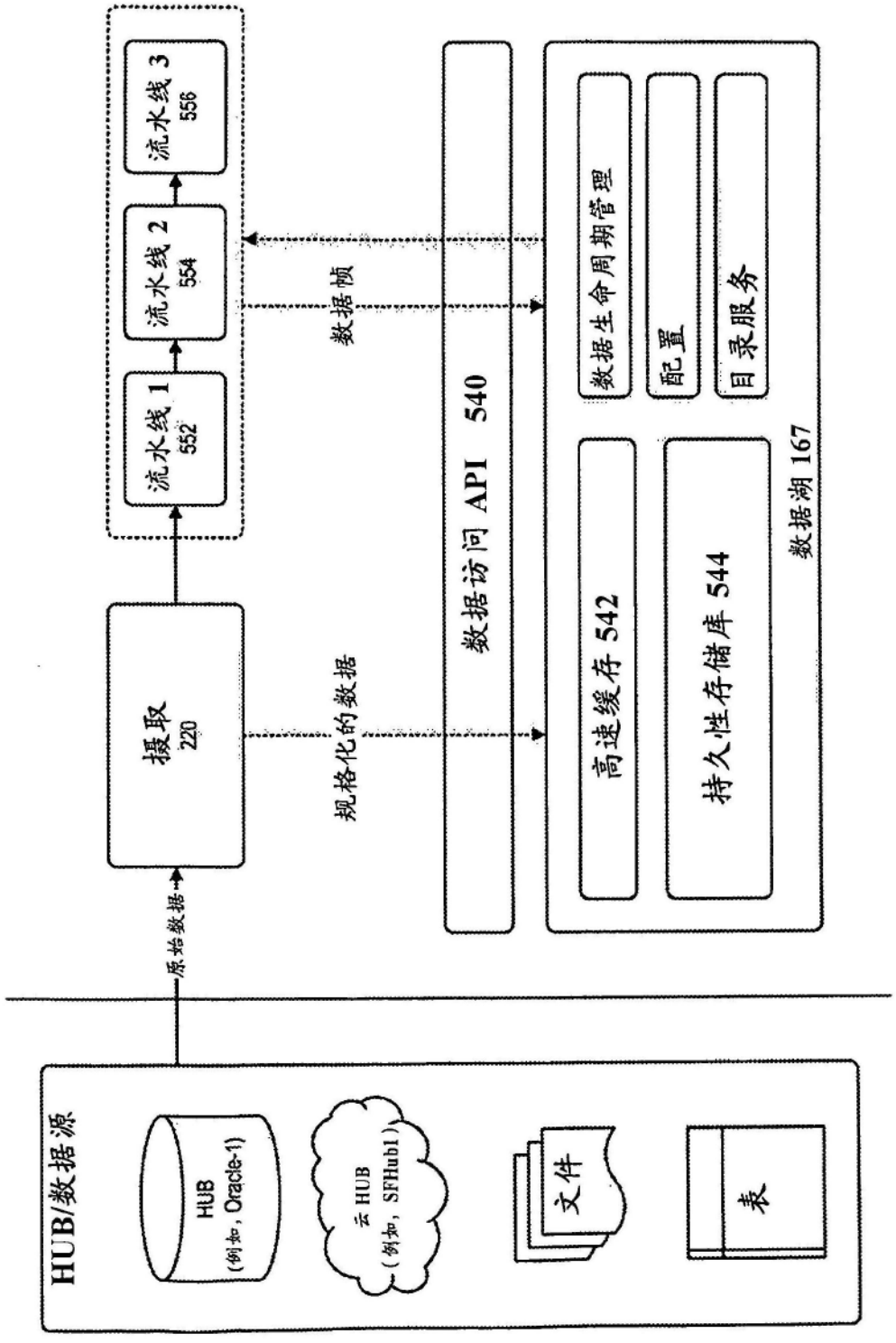

HUB/数据源
HUB
(例如, Oracle-1)
云 HUB
(例如, SFHub1)
文件
表
原始数据
摄取
220
流水线 1
552
流水线 2
554
流水线 3
556
规格化的数据
数据帧
数据访问 API 540
高速缓存 542
持久性存储库 544
数据生命周期管理
配置
目录服务
数据湖 167

图18

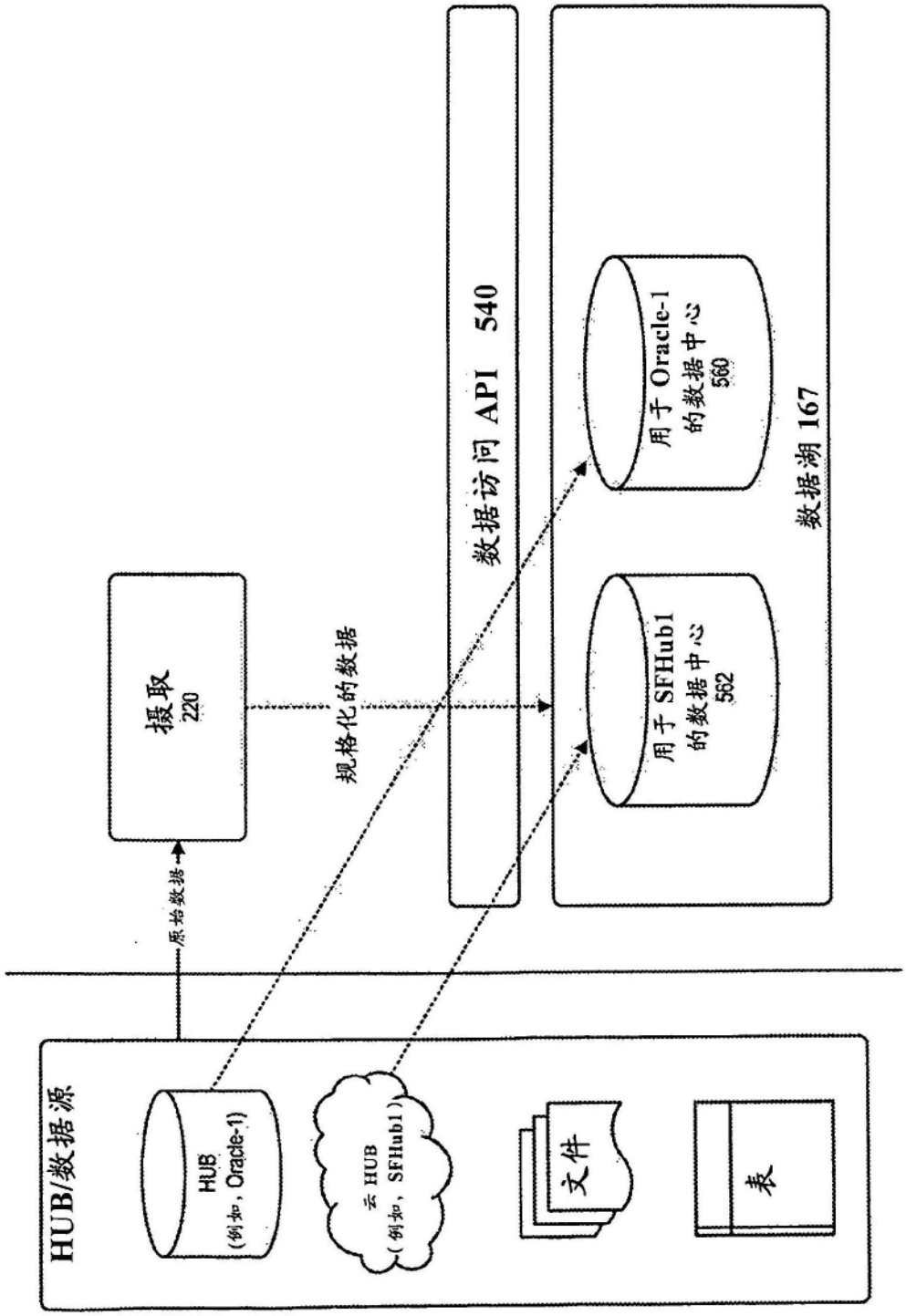

HUB/数据源
HUB
(例如，Oracle-1)
云 HUB
(例如，SFHub1)
文件
表
原始数据
摄取
220
规格化的数据
数据访问 API 540
用于 Oracle-1
的数据中心
560
用于 SFHub1
的数据中心
562
数据湖 167

图19

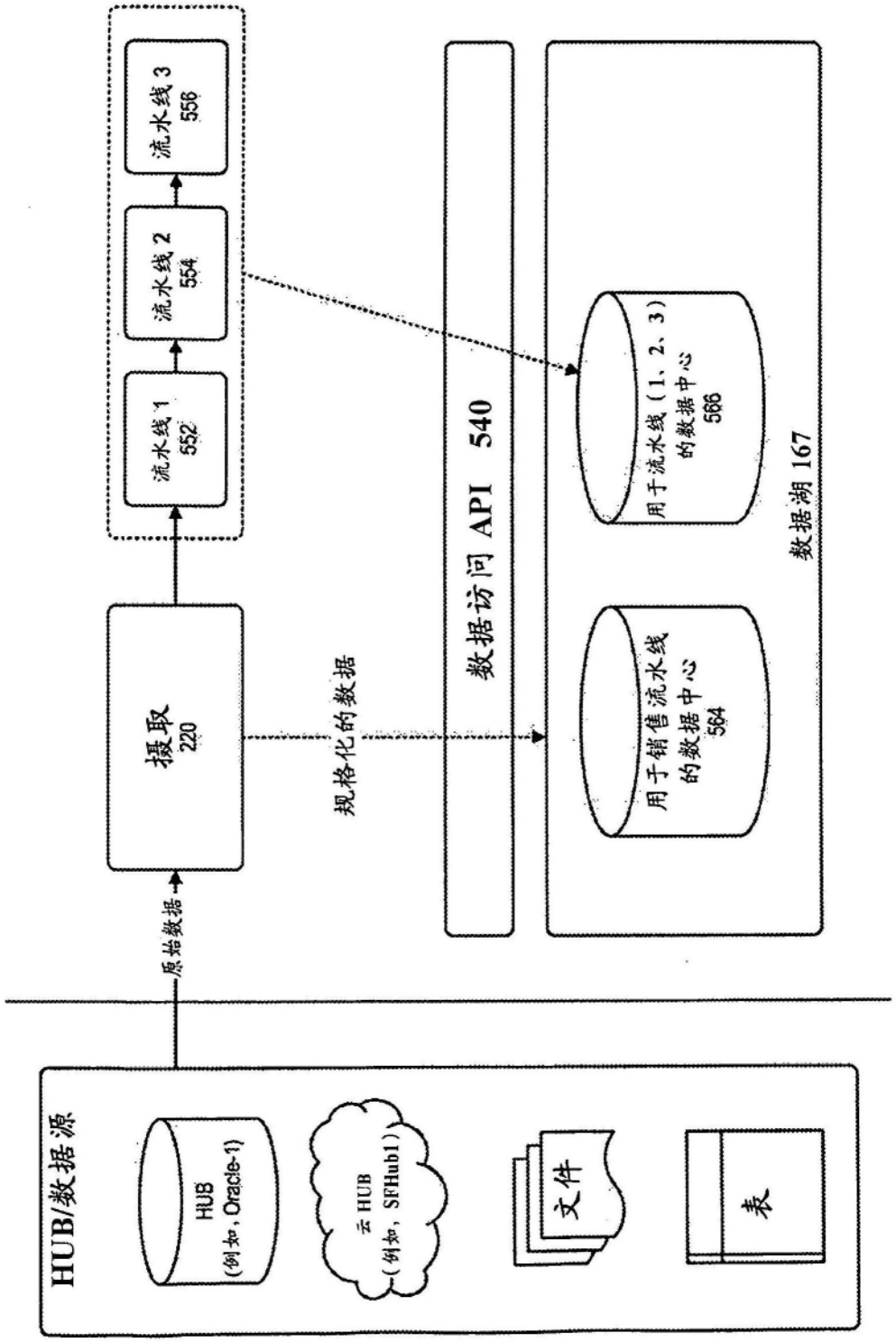

HUB/数据源
HUB
(例如，Oracle-1)
云 HUB
(例如，SFHub1)
文件
表
原始数据
摄取
220
流水线 1
552
流水线 2
554
流水线 3
556
规格化的数据
数据访问 API 540
用于销售流水线
的数据中心
564
用于流水线（1、2、3）
的数据中心
566
数据湖 167

图20

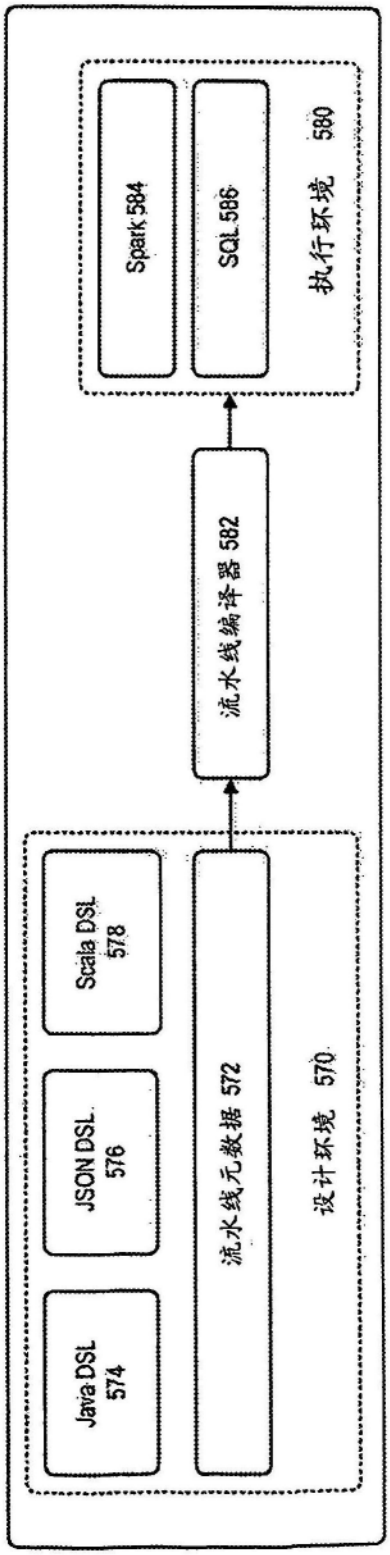

Spark 584
SQL 586
执行环境 580
流水线编译器 582
Scala DSL 578
JSON DSL 576
流水线元数据 572
Java DSL 574
设计环境 570

图21

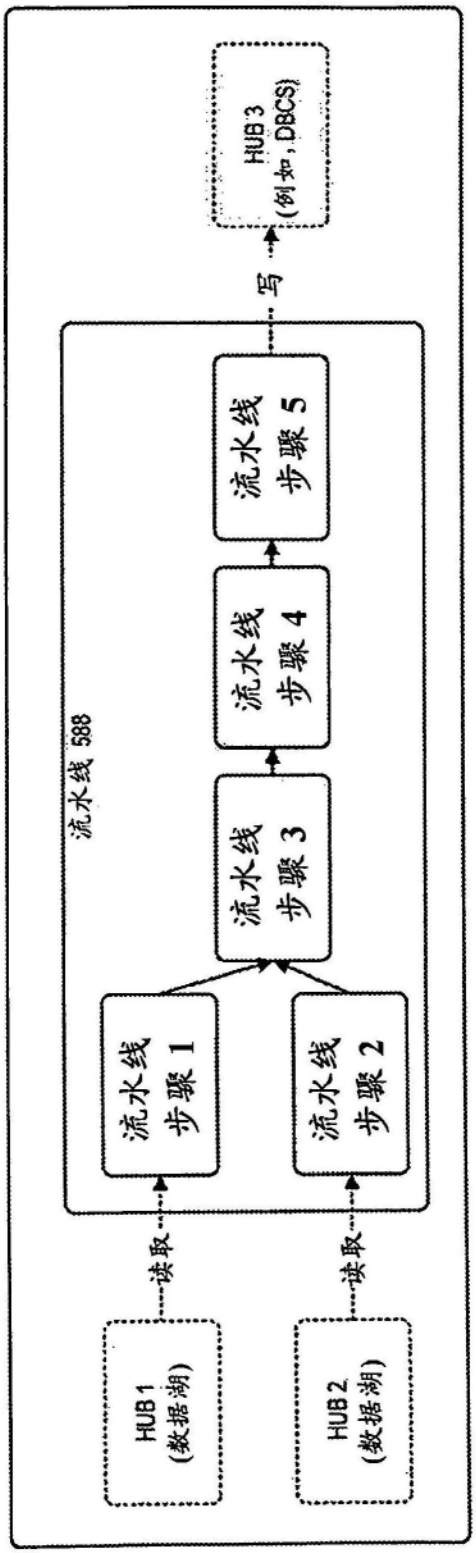

流水线 588
流水线 步骤 1
流水线 步骤 2
流水线 步骤 3
流水线 步骤 4
流水线 步骤 5
读取
读取
写
HUB 1 (数据湖)
HUB 2 (数据湖)
HUB 3 (例如, DBCS)

图22

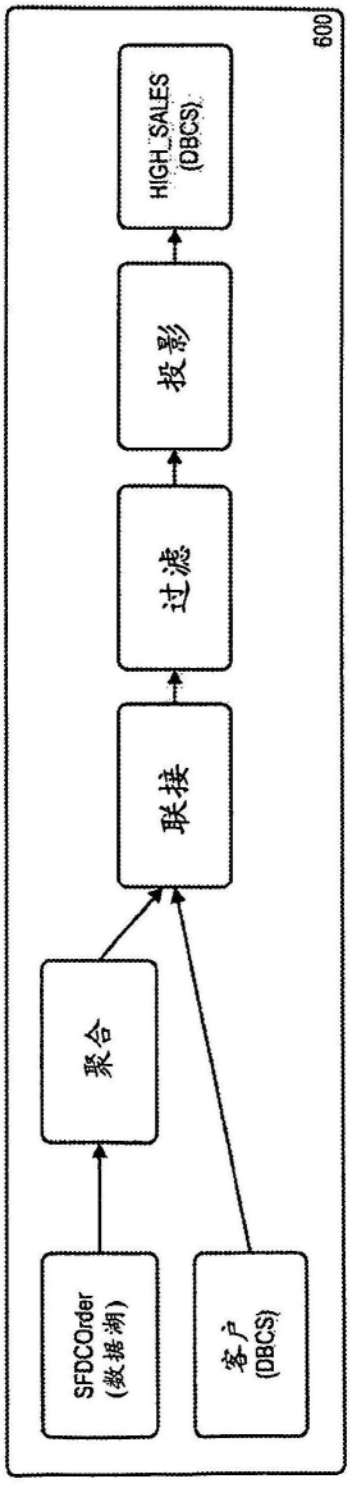


图23

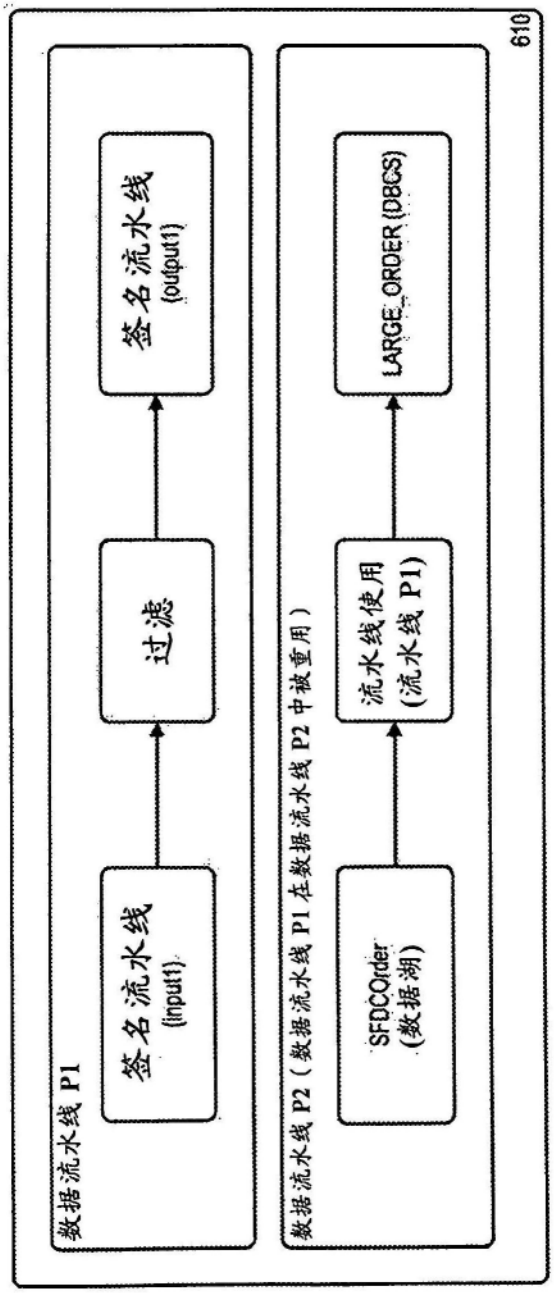

610
数据流水线 P1
签名流水线 (input1)
过滤
签名流水线 (output1)
数据流水线 P2（数据流水线 P1 在数据流水线 P2 中被重用）
SFDCOrder (数据湖)
流水线使用 (流水线 P1)
LARGE_ORDER (DBCS)

图24

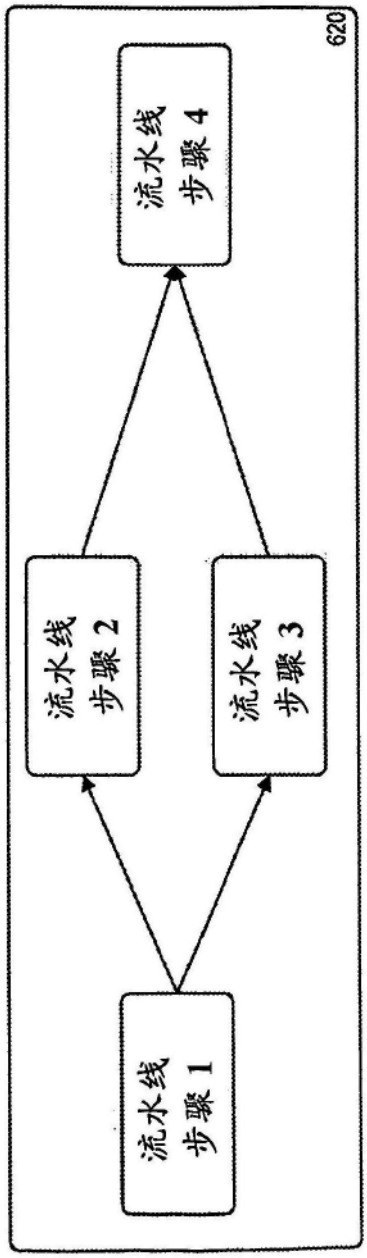


图25

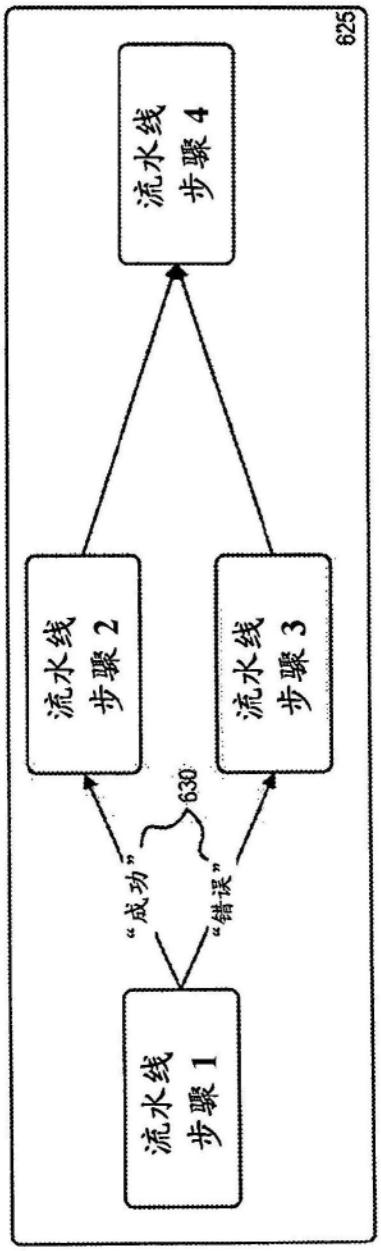


图26

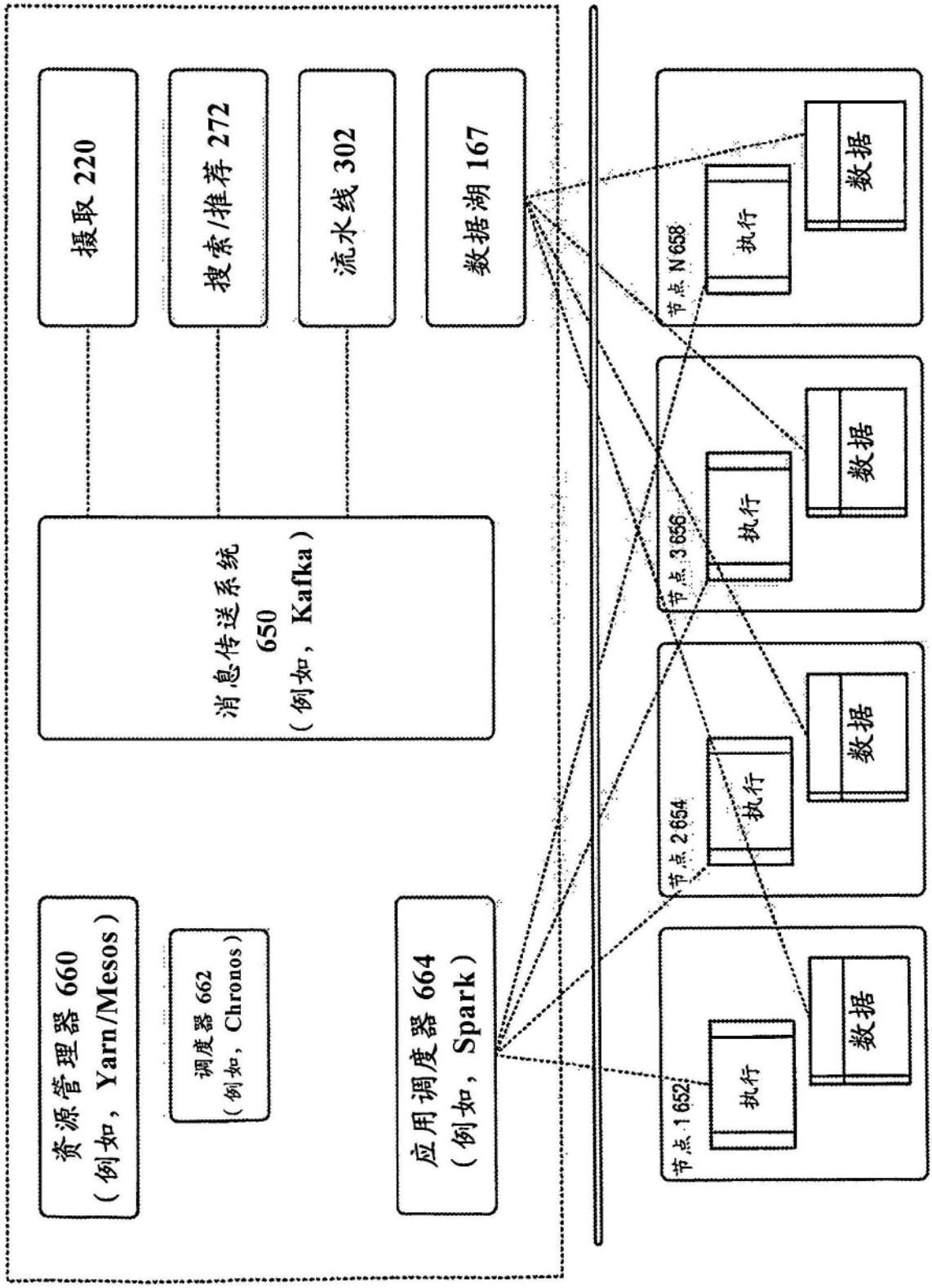

摄取 220
搜索/推荐 272
流水线 302
数据湖 167
消息传送系统
650
（例如，Kafka）
资源管理器 660
（例如，Yarn/Mesos）
调度器 662
（例如，Chronos）
应用调度器 664
（例如，Spark）
节点 1 652
执行
数据
节点 2 654
执行
数据
节点 3 656
执行
数据
节点 N 658
执行
数据

图27

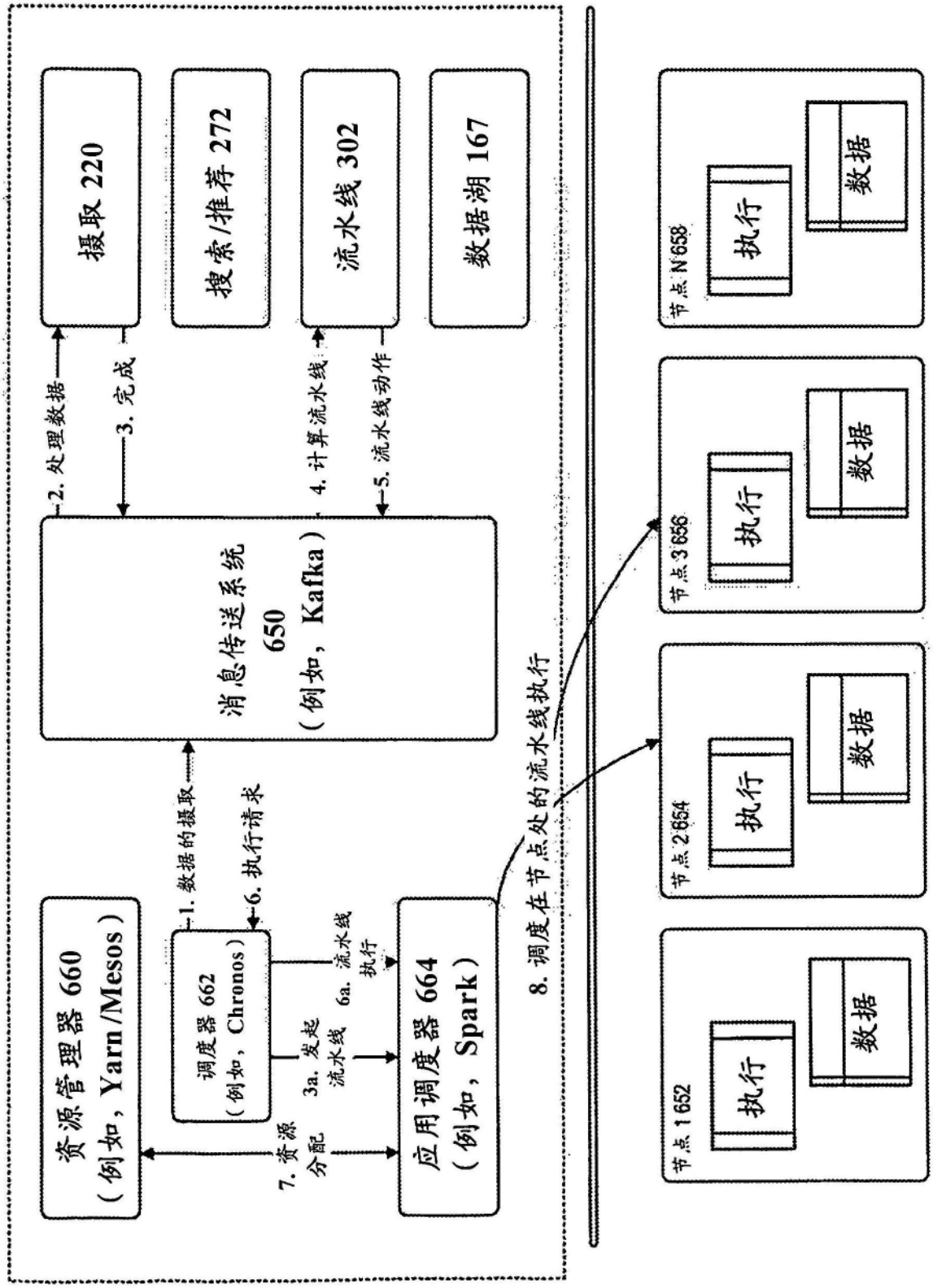

摄取 220
搜索/推荐 272
流水线 302
数据湖 167
2. 处理数据
3. 完成
4. 计算流水线
5. 流水线动作
消息传送系统
650
(例如，Kafka)
1. 数据的摄取
6. 执行请求
资源管理器 660
(例如，Yarn/Mesos)
调度器 662
(例如，Chronos)
3a. 发起流水线
6a. 流水线执行
7. 资源分配
应用调度器 664
(例如，Spark)
8. 调度在节点处的流水线执行
节点 1 652
执行
数据
节点 2 654
执行
数据
节点 3 656
执行
数据
节点 N 658
执行
数据

图28

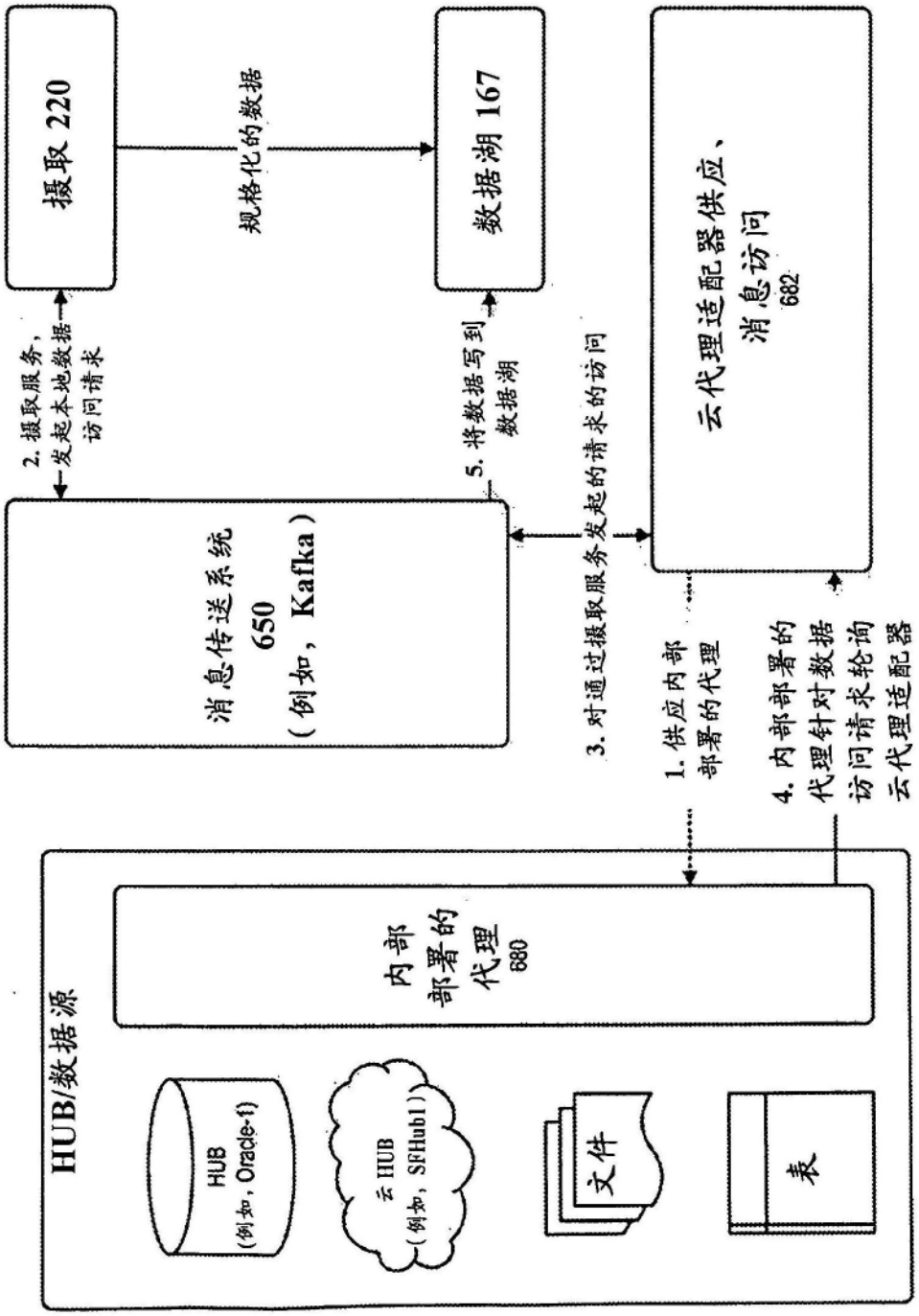

摄取 220
规格化的数据
数据湖 167
2. 摄取服务，发起本地数据访问请求
5. 将数据写到数据湖
消息传送系统 650 （例如，Kafka）
3. 对通过摄取服务发起的请求的访问
云代理适配器供应、消息访问 682
1. 供应内部部署的代理
4. 内部部署的代理针对数据访问请求轮询云代理适配器
HUB/数据源
内部部署的代理 680
HUB （例如，Oracle-f）
云 HUB （例如，SFHub1）
文件
表

图29

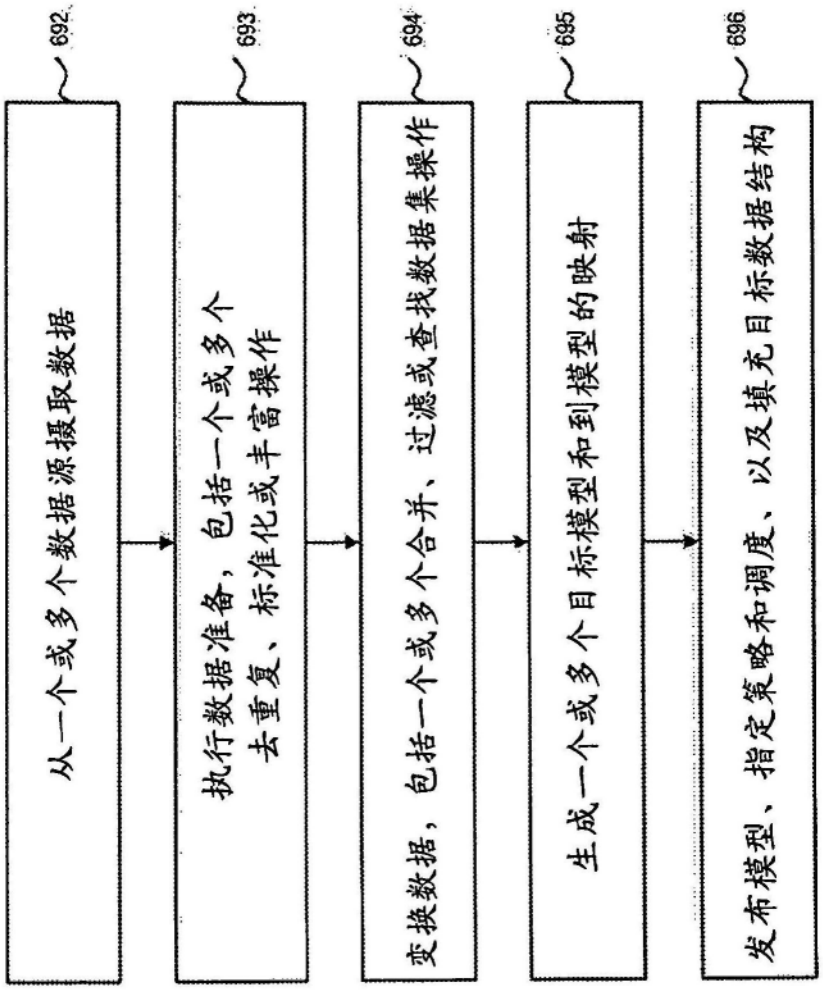

692
从一个或多个数据源摄取数据
693
执行数据准备，包括一个或多个去重复、标准化或丰富操作
694
变换数据，包括一个或多个合并、过滤或查找数据集操作
695
生成一个或多个目标模型和到模型的映射
696
发布模型、指定策略和调度、以及填充目标数据结构

图30

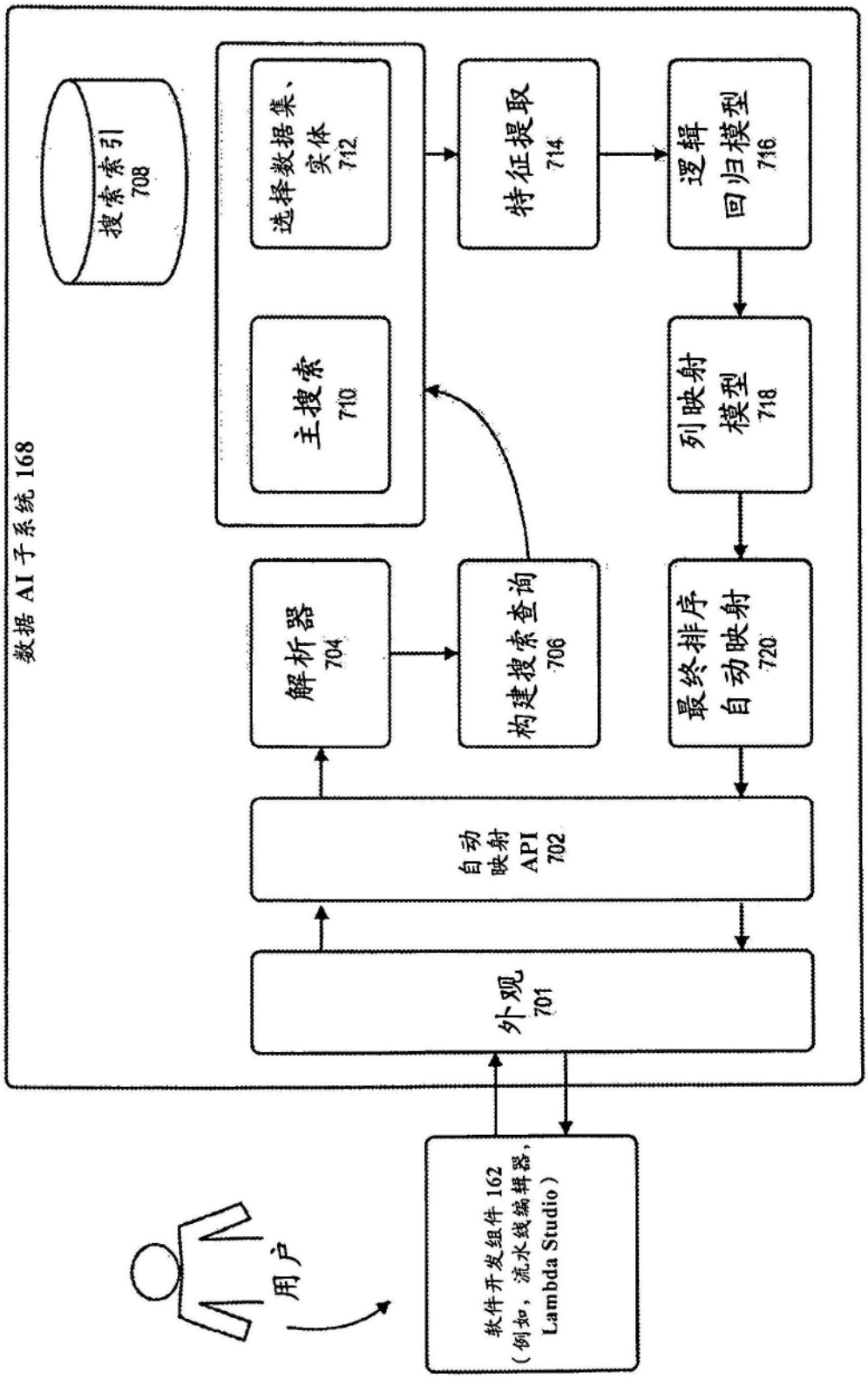

数据 AI 子系统 168
搜索索引 708
选择数据集、实体 712
特征提取 714
逻辑回归模型 716
主搜索 710
列映射模型 718
解析器 704
构建搜索查询 706
最终排序自动映射 720
自动映射 API 702
外观 701
软件开发组件 162（例如，流水线编辑器，Lambda Studio）
用户

图31

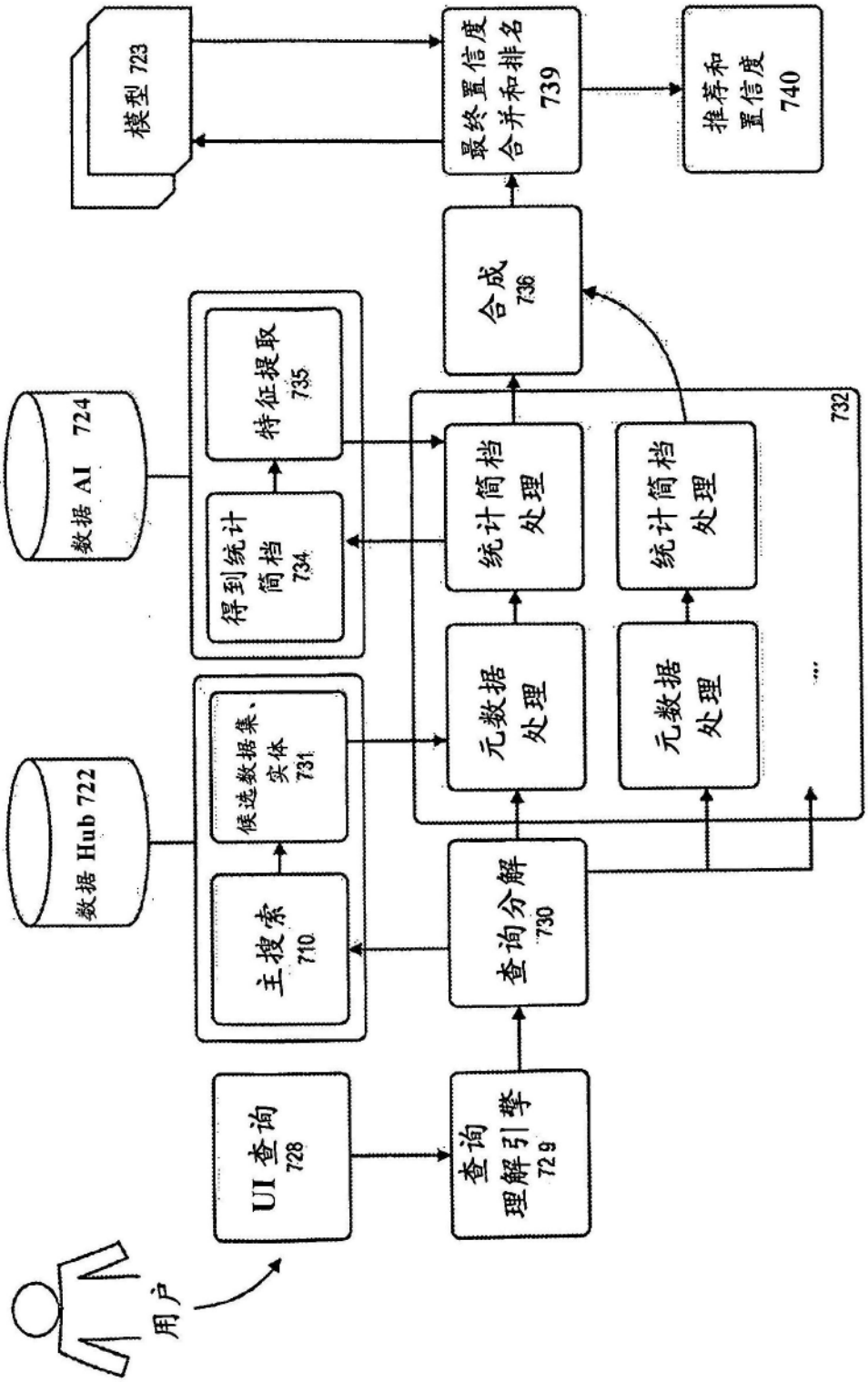

用户
UI 查询
728
查询理解引擎
729
查询分解
730
主搜索
710
候选数据集、实体
731
732
元数据处理
统计简档处理
元数据处理
统计简档处理
...
数据 Hub 722
得到统计简档
734
特征提取
735
数据 AI 724
合成
736
模型 723
最终置信度合并和排名
739
推荐和置信度
740

图32

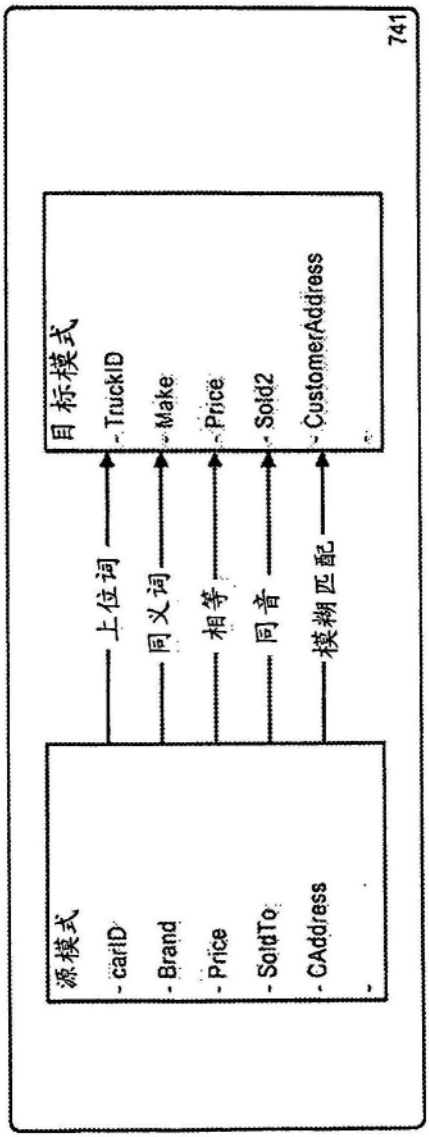

741
目标模式
- TruckID
- Make
- Price
- Sold2
- CustomerAddress
上位词
同义词
相等
同音
模糊匹配
源模式
- carID
- Brand
- Price
- SoldTo
- CAddress

图33

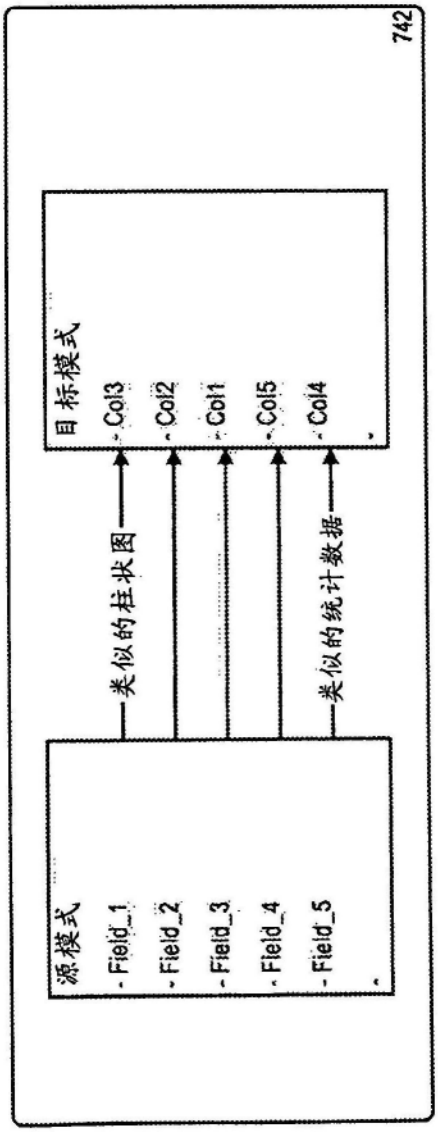

742
目标模式
- Col3
- Col2
- Col1
- Col5
- Col4
类似的柱状图
类似的统计数据
源模式
- Field_1
- Field_2
- Field_3
- Field_4
- Field_5

图34

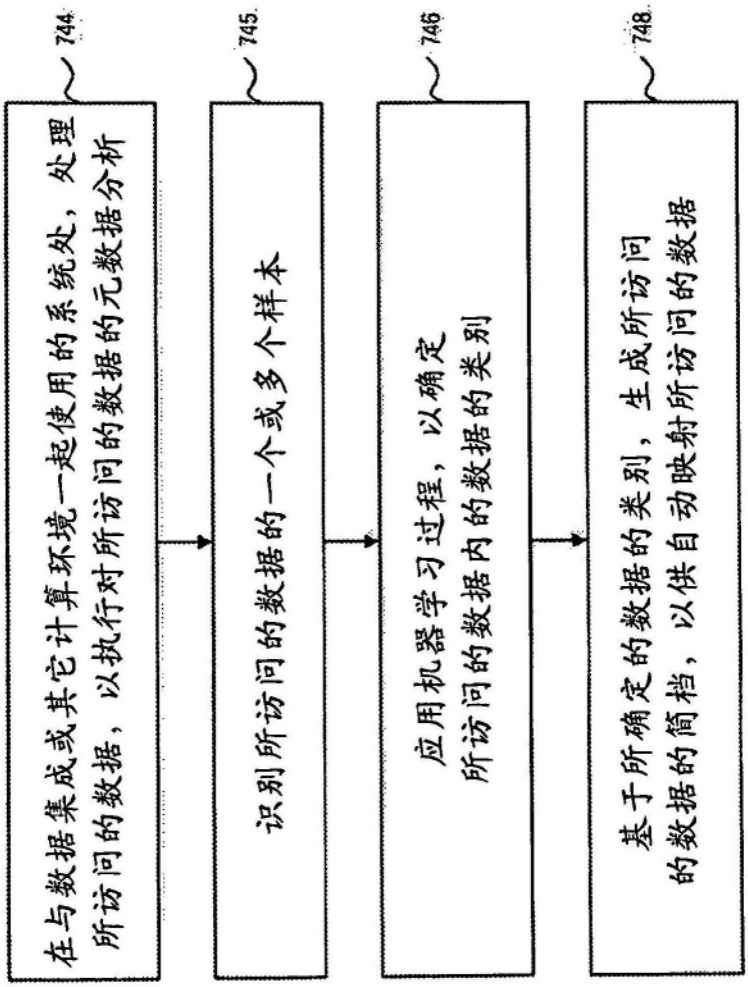

744
在与数据集成或其它计算环境一起使用的系统处，处理所访问的数据，以执行对所访问的数据的元数据分析
745
识别所访问的数据的一个或多个样本
746
应用机器学习过程，以确定所访问的数据内的数据的类别
748
基于所确定的数据的类别，生成所访问的数据的数据的简档，以供自动映射所访问的数据

图35

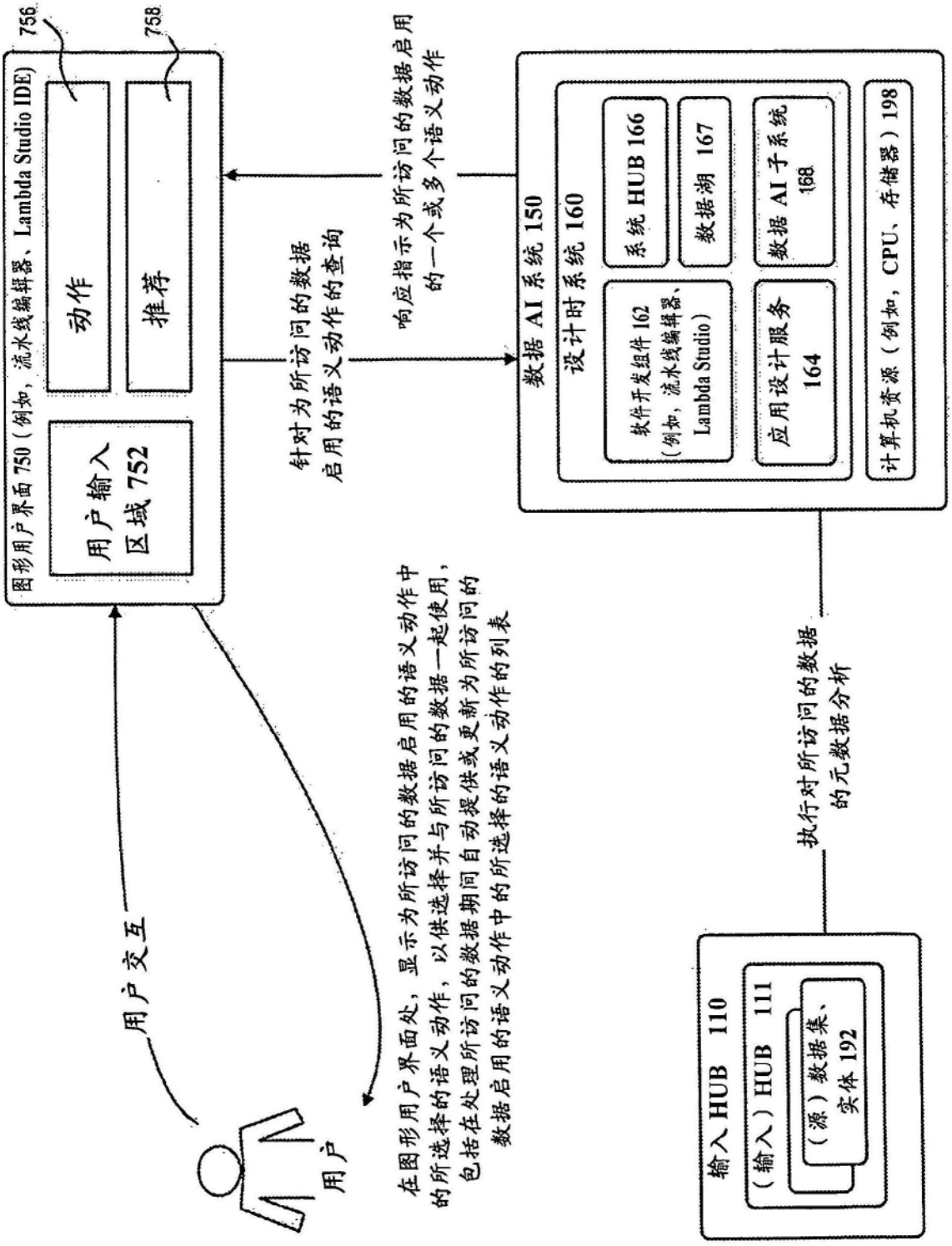

图形用户界面 750（例如，流水线编辑器、Lambda Studio IDE）
756
动作
758
推荐
用户输入区域 752
用户交互
用户
针对为所访问的数据启用的语义动作的查询
响应指示为所访问的数据启用的一个或多个语义动作
数据 AI 系统 150
设计时系统 160
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
系统 HUB 166
数据湖 167
应用设计服务 164
数据 AI 子系统 168
计算机资源（例如，CPU、存储器）198
在图形用户界面处，显示为所访问的数据启用的语义动作中的所选择的语义动作，以供选择并与所访问的数据一起使用，包括在处理所访问的数据期间自动提供或更新为所访问的数据启用的语义动作中的所选择的语义动作的列表
执行对所访问的数据的元数据分析
输入 HUB 110
（输入）HUB 111
（源）数据集、实体 192

图36

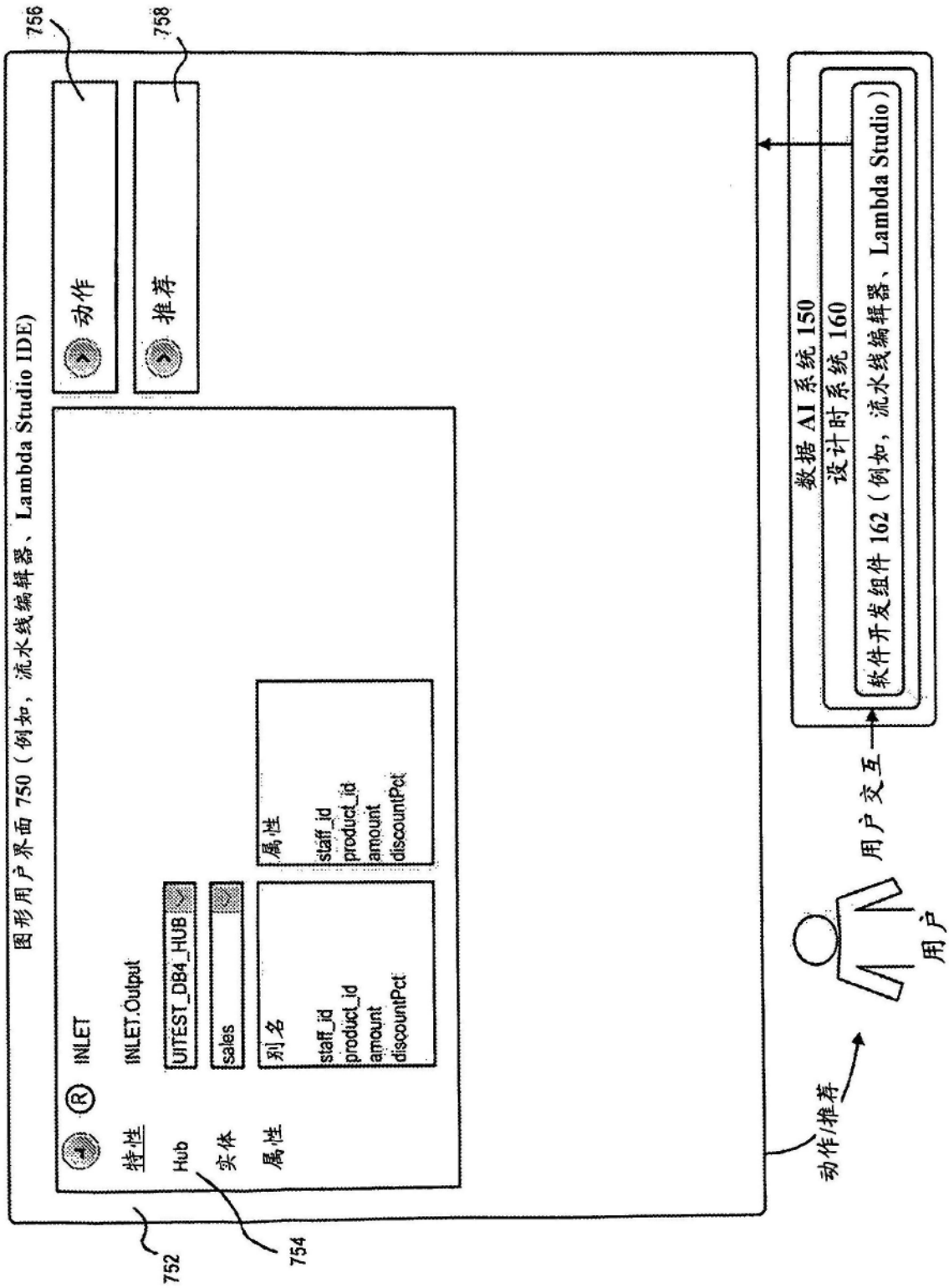

图形用户界面 750（例如，流水线编辑器、Lambda Studio IDE）
756
758
动作
推荐
INLET
INLET.Output
UITEST_DB4_HUB
sales
特性
Hub
实体
属性
列名
staff_id
product_id
amount
discountPct
属性
staff_id
product_id
amount
discountPct
752
754
数据 AI 系统 150
设计时系统 160
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
用户交互
用户
动作/推荐

图37

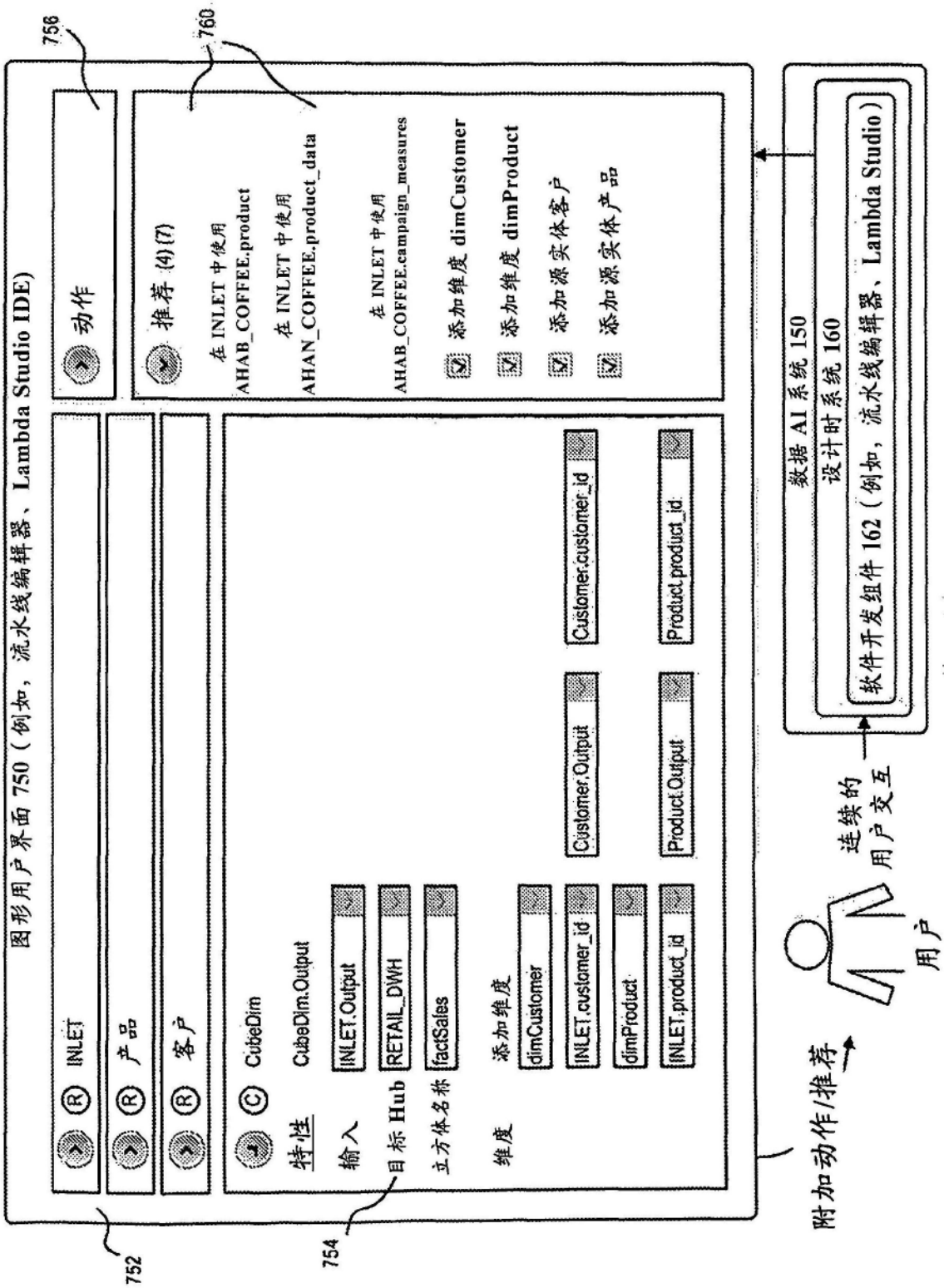

图形用户界面 750（例如，流水线编辑器、Lambda Studio IDE）
752
754
756
760
INLET
产品
客户
CubeDim
CubeDim.Output
特性
输入
目标 Hub
立方体名称
维度
添加维度
INLET.Output
RETAIL_DWH
factSales
dimCustomer
dimProduct
INLET.customer_id
INLET.product_id
Customer.Output
Product.Output
Customer.customer_id
Product.product_id
动作
推荐 (4) (7)
在 INLET 中使用
AHAB_COFFEE.product
在 INLET 中使用
AHAN_COFFEE.product_data
在 INLET 中使用
AHAB_COFFEE.campaign_measures
添加维度 dimCustomer
添加维度 dimProduct
添加源实体客户
添加源实体产品
数据 AI 系统 150
设计时系统 160
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
连续的
用户交互
用户
附加动作/推荐

图38

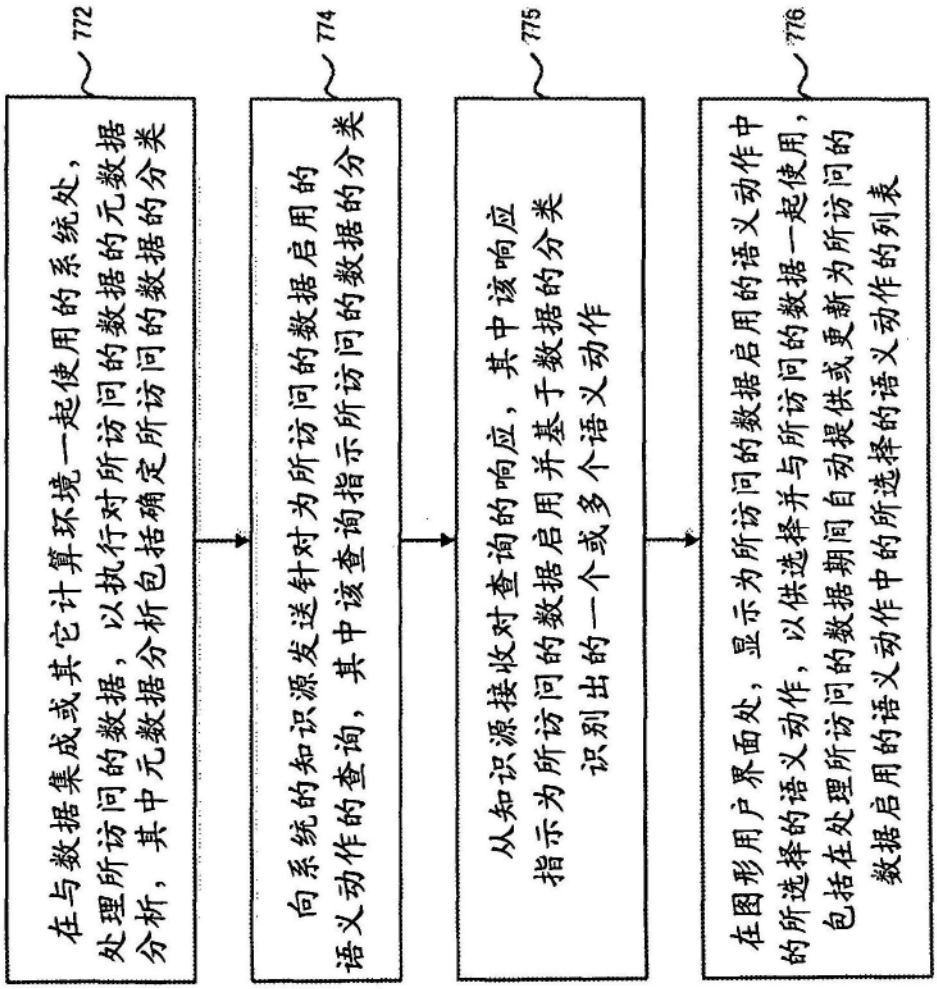

772
在与数据集成或其它计算环境一起使用的系统处，处理所访问的数据，以执行对所访问的数据的元数据分析，其中元数据分析包括确定所访问的数据的分类
774
向系统的知识源发送针对为所访问的数据启用的语义动作的查询，其中该查询指示所访问的数据的分类
775
从知识源接收对查询的响应，其中该响应指示为所访问的数据启用并基于数据的分类识别出的一个或多个语义动作
776
在图形用户界面处，显示为所访问的数据启用的语义动作中的所选择的语义动作，以供选择并与所访问的数据一起使用，包括在处理所访问的数据期间自动提供或更新为所访问的数据启用的语义动作中的所选择的语义动作的列表

图39

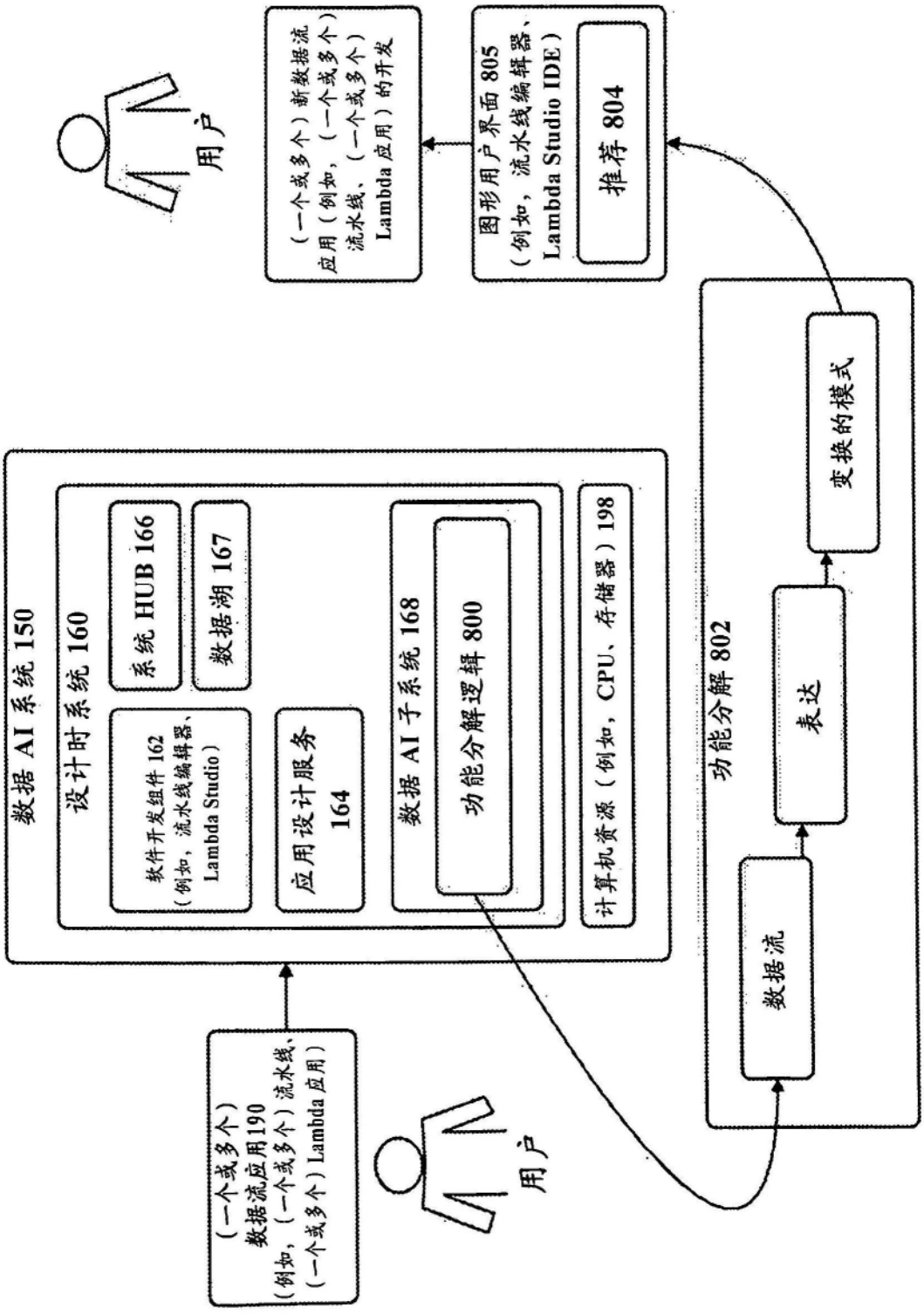

数据 AI 系统 150
设计时系统 160
软件开发组件 162
（例如，流水线编辑器、Lambda Studio）
系统 HUB 166
数据湖 167
应用设计服务 164
数据 AI 子系统 168
功能分解逻辑 800
计算机资源（例如，CPU、存储器）198
（一个或多个）数据流应用 190
（例如，（一个或多个）流水线、（一个或多个）Lambda 应用）
用户
功能分解 802
数据流
表达
变换的模式
图形用户界面 805
（例如，流水线编辑器、Lambda Studio IDE）
推荐 804
（一个或多个）新数据流应用（例如，（一个或多个）流水线、（一个或多个）Lambda 应用）的开发
用户

图40

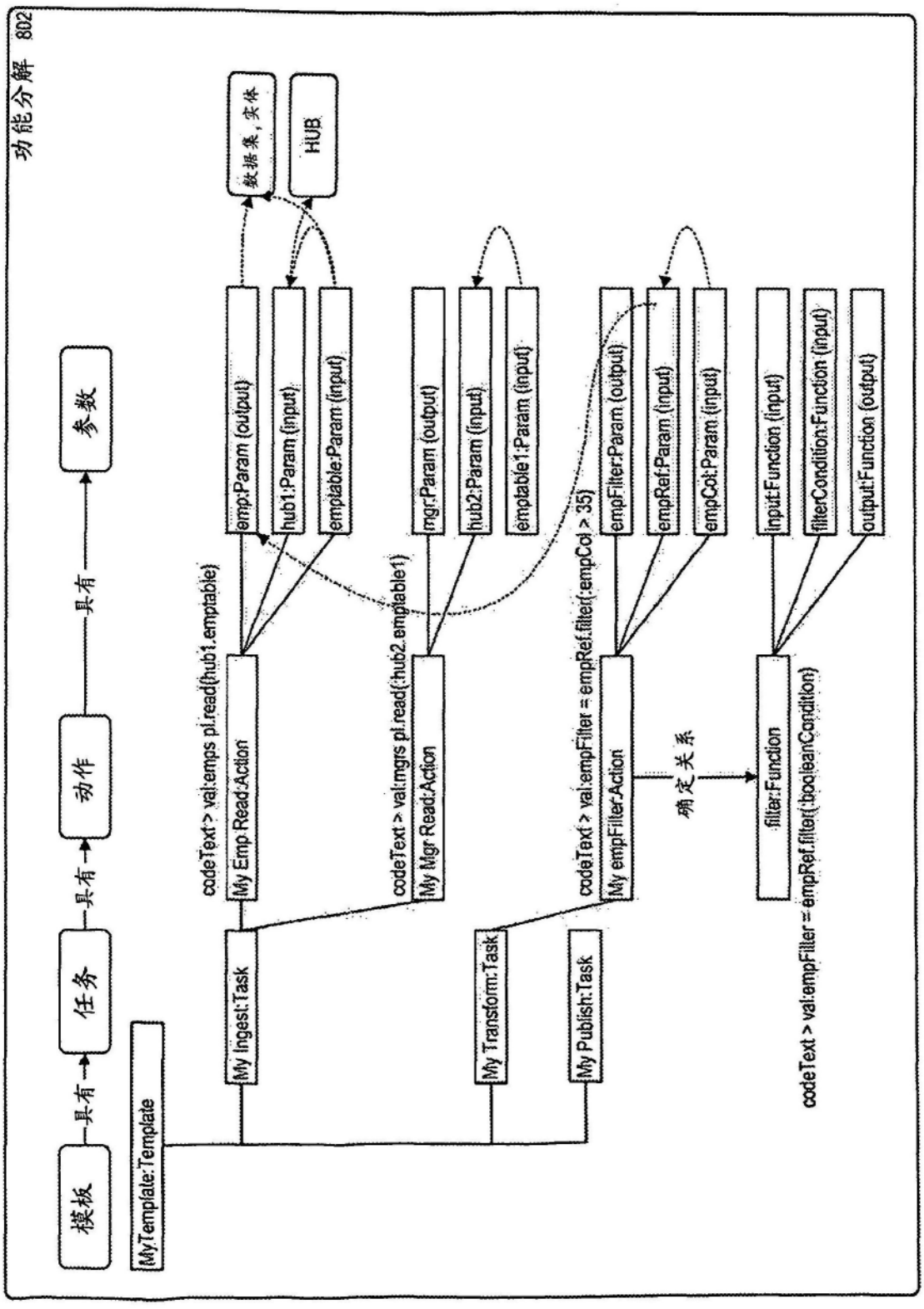

功能分解 802
模板
具有
任务
具有
动作
具有
参数
(MyTemplate:Template
My Ingest:Task
My Transform:Task
My Publish:Task
codeText > val:emps pi.read(hub1.emptable)
My Emp Read:Action
codeText > val:mgrs pi.read(:hub2.emptable1)
My Mgr Read:Action
codeText > val:empFilter = empRef.filter(:empCol > 35)
My empFilter:Action
确定关系
filter:Function
codeText > val:empFilter = empRef.filter(:booleanCondition)
emp:Param (output)
hub1:Param (input)
emptable:Param (input)
mgr:Param (output)
hub2:Param (input)
emptable1:Param (input)
empFilter:Param (output)
empRef:Param (input)
empCol:Param (input)
input:Function (input)
filterCondition:Function (input)
output:Function (output)
数据集，实体
HUB

图41

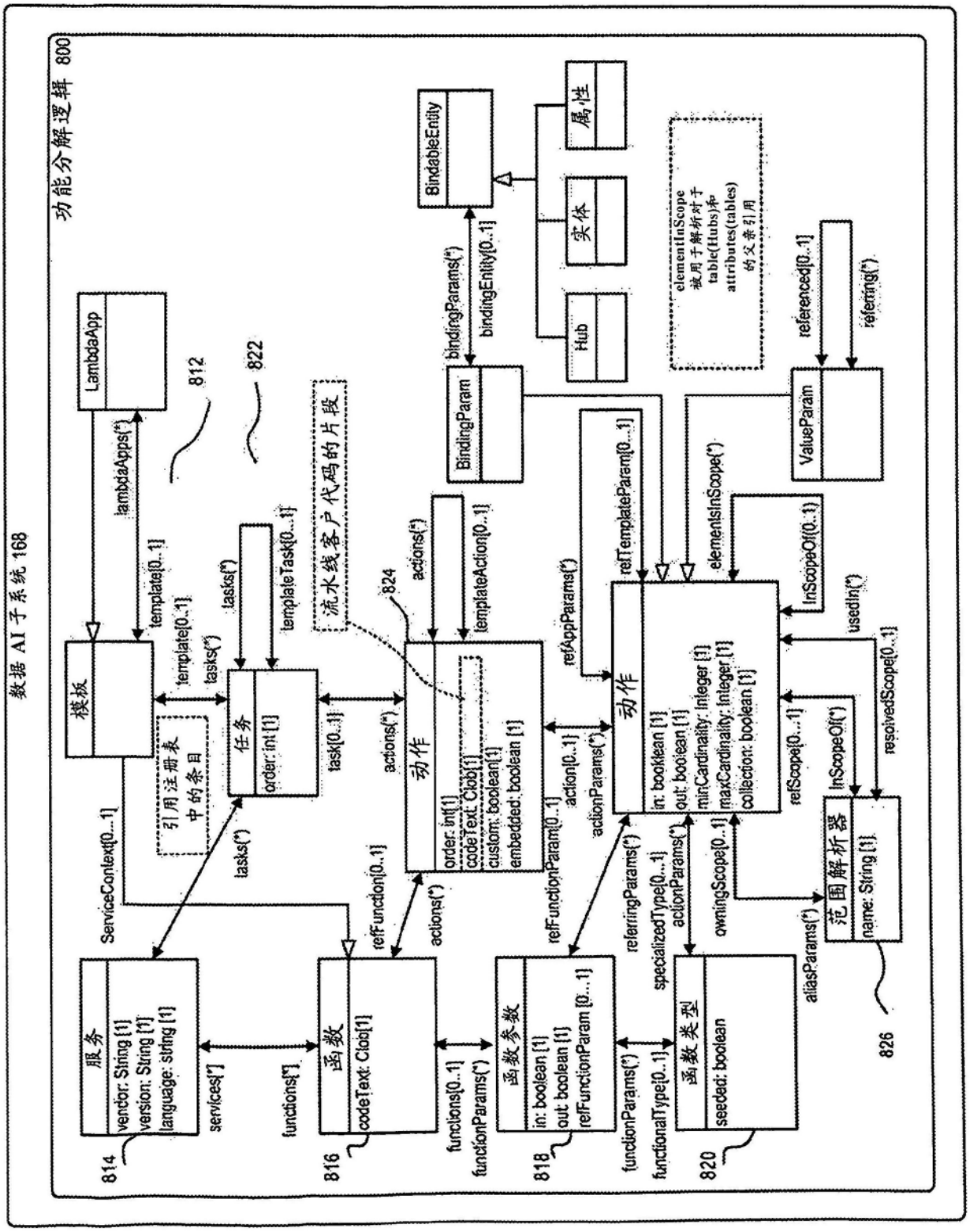

数据 AI 子系统 168
功能分解逻辑 800
LambdaApp
lambdaApps(*)
812
822
模板
template[0..1]
引用注册表中的条目
template[0..1]
tasks(*)
templateTask[0..1]
任务
order: int [1]
task[0..1]
流水线客户代码的片段
824
actions(*)
templateAction[0..1]
动作
order: int[1]
codeText: Clob[1]
custom: boolean[1]
embedded: boolean [1]
actions(*)
tasks(*)
ServiceContext[0..1]
refFunction[0..1]
actions(*)
action[0..1]
actionParams(*)
refFunctionParam[0..1]
refAppParams(*)
refTemplateParam[0..1]
动作
in: booklean [1]
out: boolean [1]
minCardinality: Integer [1]
maxCardinality: Integer [1]
collection: boolean [1]
elementsInScope(*)
InScopeOf[0..1]
usedIn(*)
refScope[0..1]
InScopeOf(*)
resolvedScope[0..1]
referencedParams(*)
referringParams(*)
specializedType[0..1]
actionParams(*)
owningScope[0..1]
aliasParams(*)
范围解析器
name: String [1]
826
BindableEntity
bindingParams(*)
bindingEntity[0..1]
BindingParam
Hub
实体
属性
elementInScope
被用于解析对于
table(Hubs)和
attributes(tables)
的父表引用
referenced[0..1]
referring(*)
ValueParam
服务
vendor: String [1]
version: String [1]
language: string [1]
814
services[*]
functions[*]
函数
codeText: Clob[1]
816
functions[0..1]
functionParams(*)
函数参数
in: boolean [1]
out: boolean [1]
refFunctionParam [0...1]
818
functionParams(*)
functionalType[0..1]
函数类型
seeded: boolean
820

图42

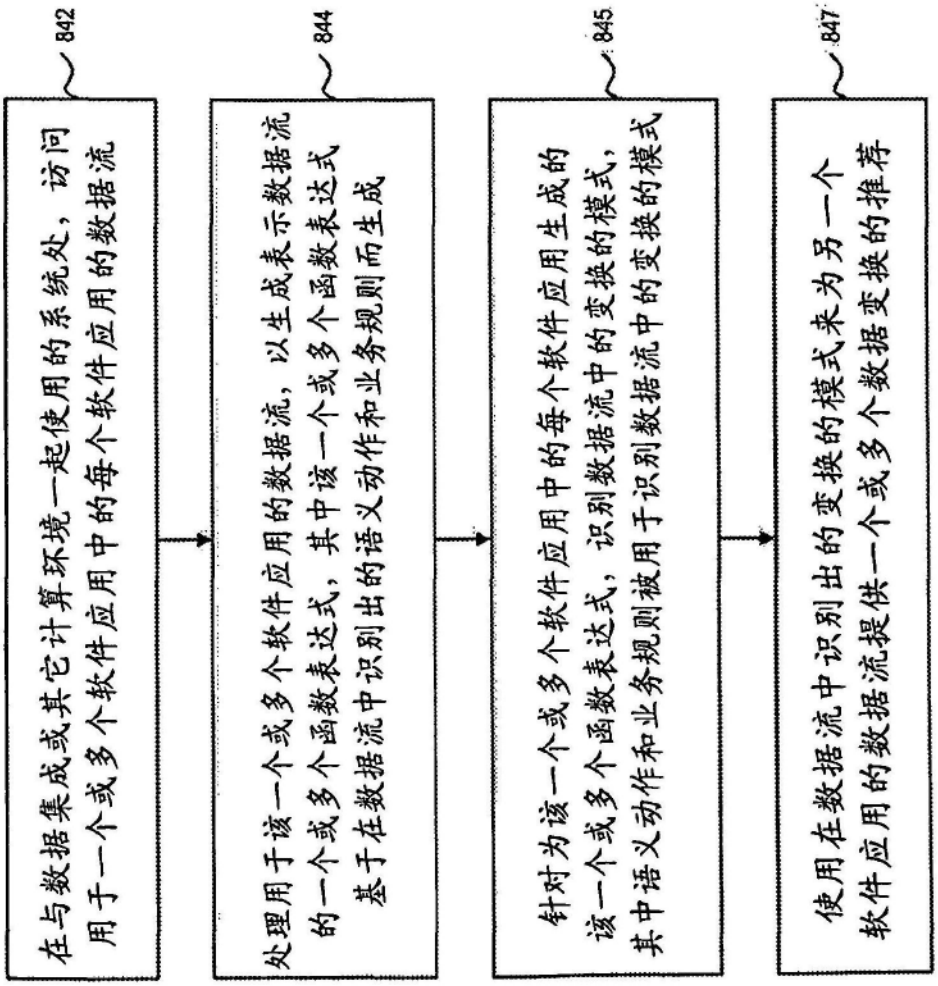

842
在与数据集成或其它计算环境一起使用的系统处，访问用于一个或多个软件应用中的每个软件应用的数据流
844
处理用于该一个或多个软件应用的数据流，以生成表示数据流的一个或多个函数表达式，其中该一个或多个函数表达式基于在数据流中识别出的语义动作和业务规则而生成
845
针对为该一个或多个软件应用中的每个软件应用生成的该一个或多个函数表达式，识别数据流中的变换的模式，其中语义动作和业务规则被用于识别数据流中的变换的模式
847
使用在数据流中识别出的变换的模式来为另一个软件应用的数据流提供一个或多个数据变换的推荐

图43

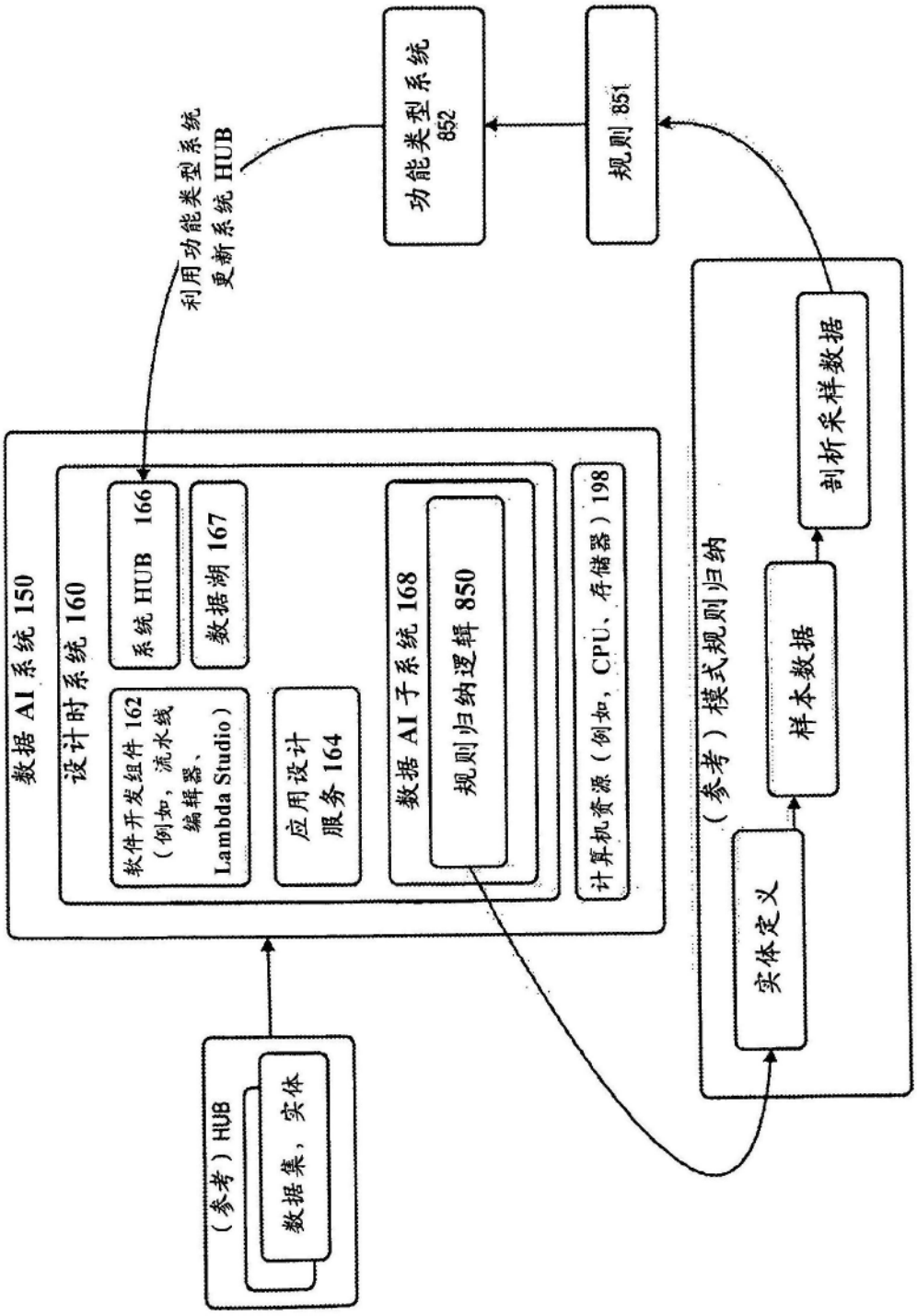

数据 AI 系统 150
设计时系统 160
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
系统 HUB 166
数据湖 167
应用设计服务 164
数据 AI 子系统 168
规则归纳逻辑 850
计算机资源（例如，CPU、存储器）198
利用功能类型系统更新系统 HUB
功能类型系统 852
规则 851
（参考）模式规则归纳
实体定义
样本数据
剖析采样数据
（参考）HUB
数据集，实体

图44

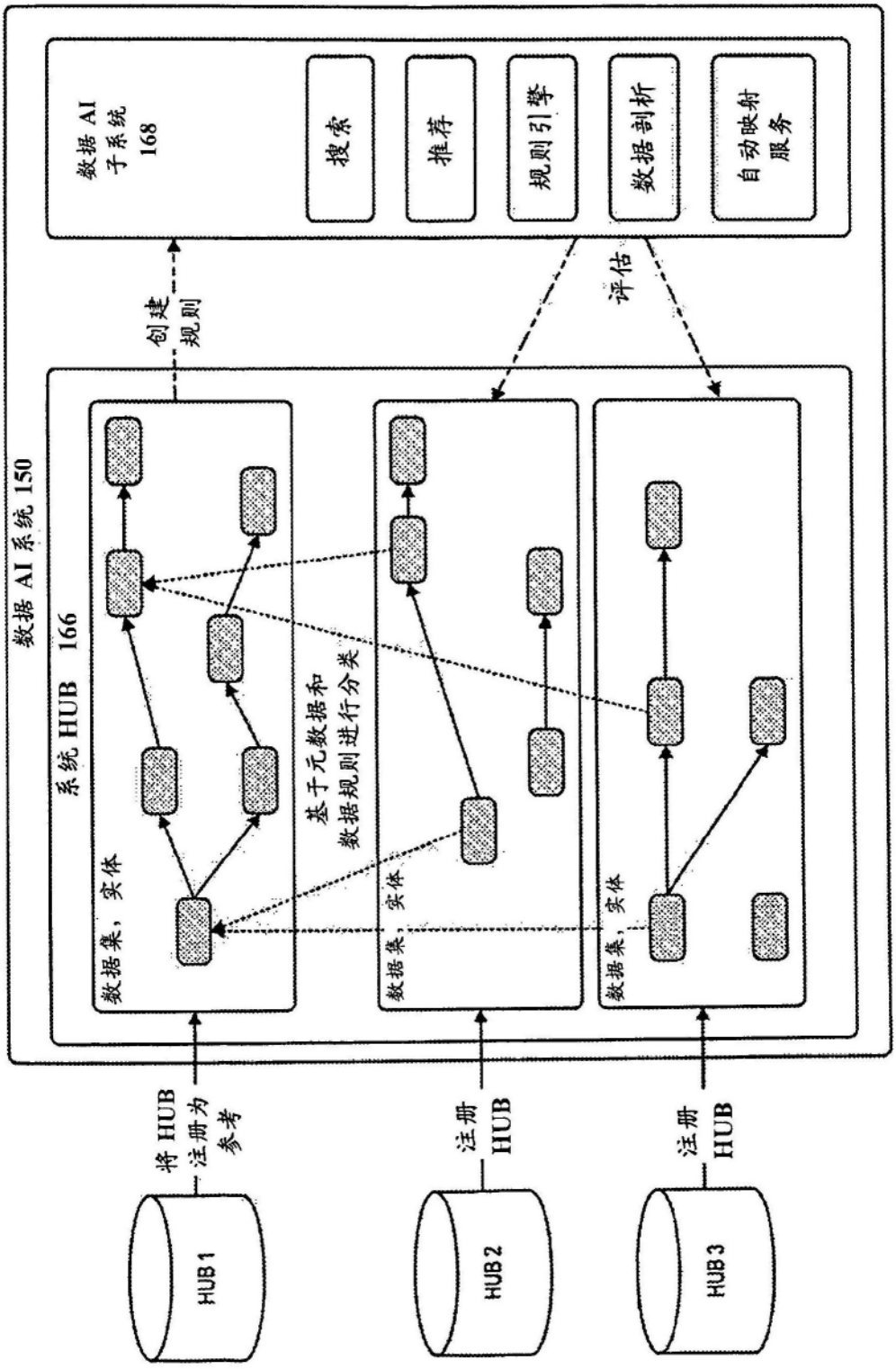

数据 AI 系统 150
系统 HUB 166
数据集，实体
基于元数据和
数据规则进行分类
数据集，实体
数据集，实体
数据 AI
子系统
168
搜索
推荐
规则引擎
数据剖析
自动映射
服务
创建
规则
评估
将 HUB
注册为
参考
注册
HUB
注册
HUB
HUB 1
HUB 2
HUB 3

图45

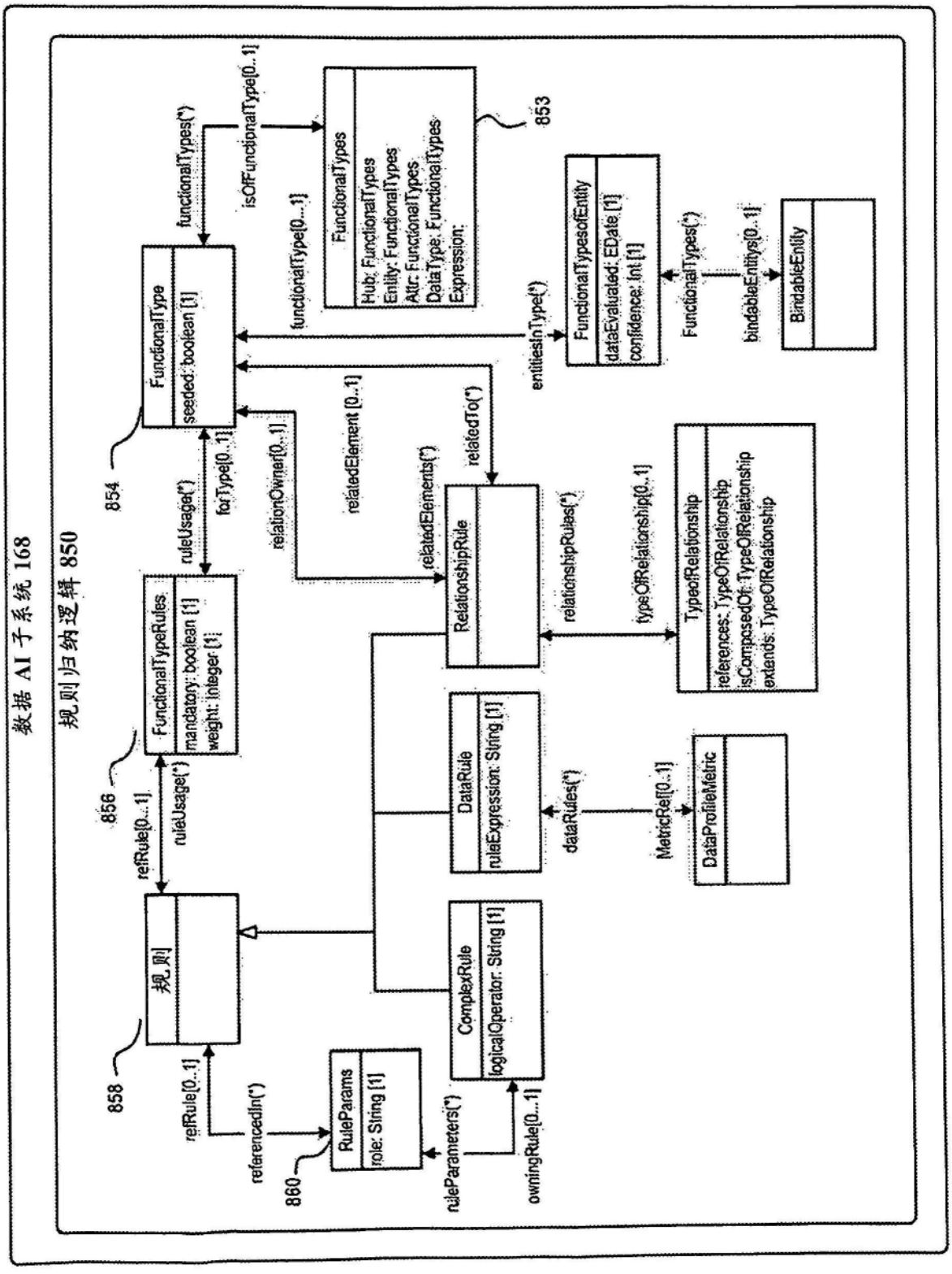

数据 AI 子系统 168
规则归纳逻辑 850
FunctionalType
seeded: boolean [1]
854
functionalTypes(*)
isOfFunctionalType[0..1]
functionalType[0..1]
FunctionalTypes
Hub: FunctionalTypes
Entity: FunctionalTypes
Attr: FunctionalTypes
DataType: FunctionalTypes
Expression:
853
entitiesInType(*)
FunctionalTypesofEntity
dataEvaluated: EDate [1]
confidence: Int [1]
FunctionalTypes(*)
bindableEntitys[0..1]
BindableEntity
ruleUsage(*)
forType[0..1]
relationOwner[0..1]
relatedElement [0..1]
relatedElements(*)
relatedTo(*)
FunctionalTypeRules
mandatory: boolean [1]
weight: Integer [1]
856
RelationshipRule
relationshipRules(*)
typeOfRelationship[0..1]
TypeofRelationship
references: TypeOfRelationship
isComposedOf: TypeOfRelationship
extends: TypeOfRelationship
refRule[0..1]
ruleUsage(*)
DataRule
ruleExpression: String [1]
dataRules(*)
MetricRel[0..1]
DataProfileMetric
规则
858
ComplexRule
logicalOperator: String [1]
refRule[0..1]
referencedIn(*)
RuleParams
role: String [1]
860
ruleParameters(*)
owningRule[0..1]

图46

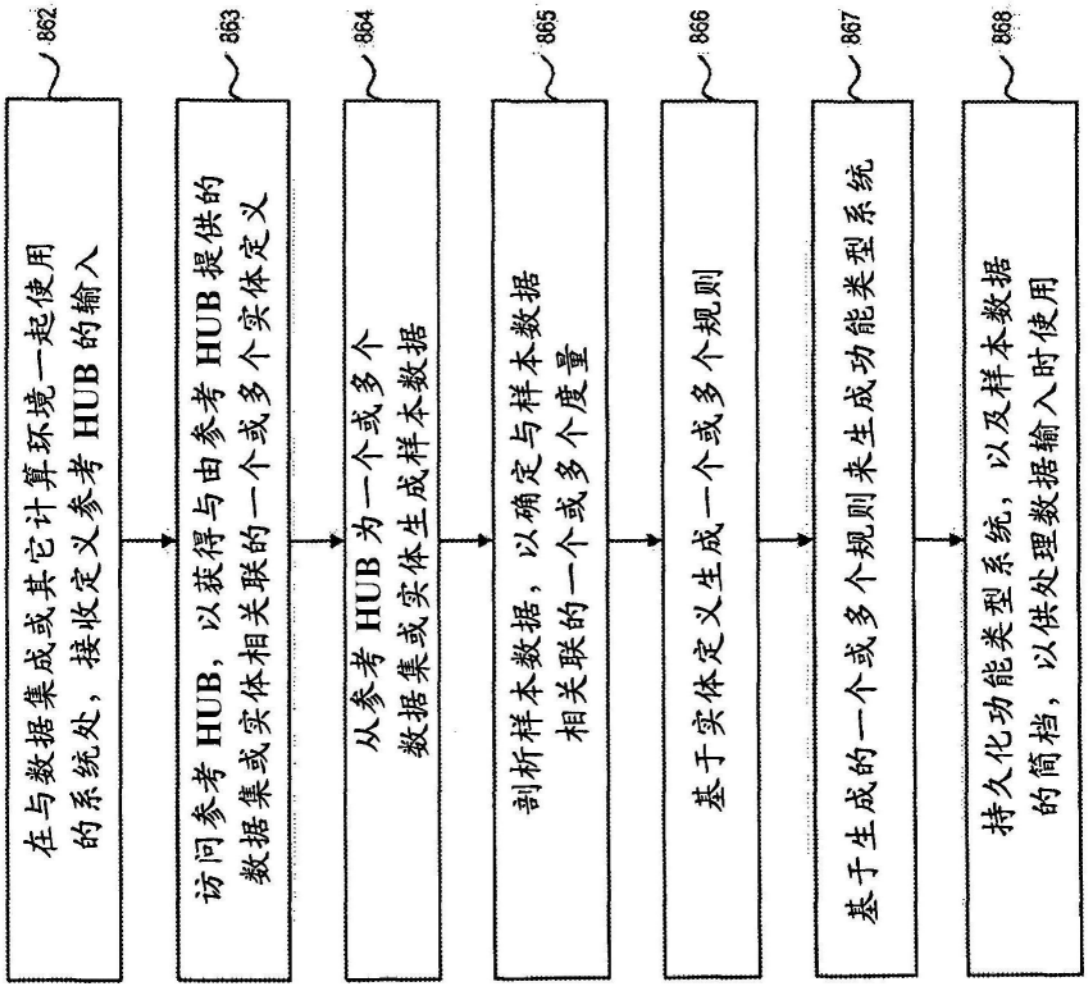

862
在与数据集成或其它计算环境一起使用的系统处，接收定义参考 HUB 的输入
863
访问参考 HUB，以获得与由参考 HUB 提供的数据集或实体相关联的一个或多个实体定义
864
从参考 HUB 为一个或多个数据集或实体生成样本数据
865
剖析样本数据，以确定与样本数据相关联的一个或多个度量
866
基于实体定义生成一个或多个规则
867
基于生成的一个或多个规则来生成功能类型系统
868
持久化功能类型系统，以及样本数据的简档，以供处理数据输入时使用

图47

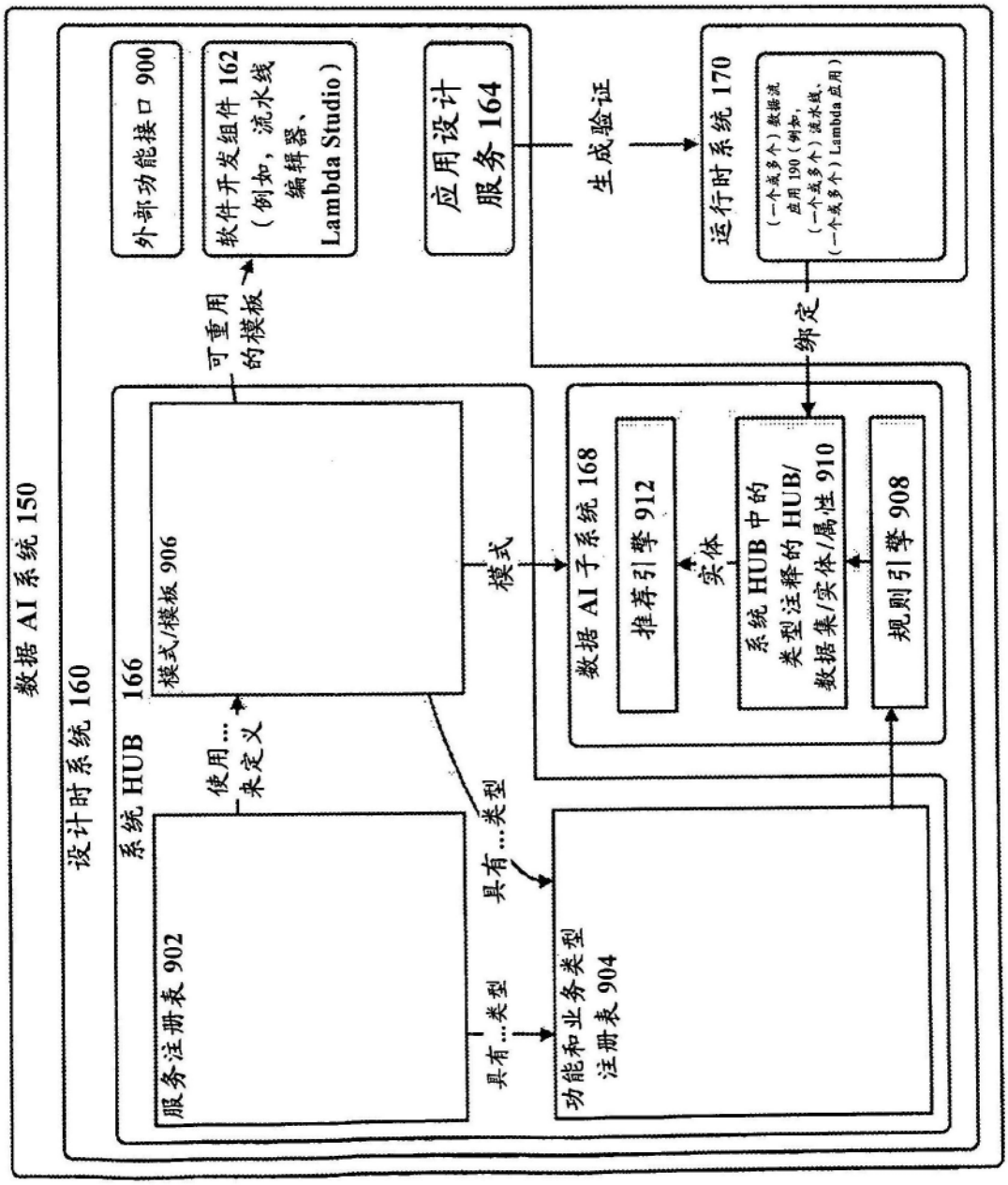

数据 AI 系统 150
设计时系统 160
外部功能接口 900
软件开发组件 162
（例如，流水线
编辑器、
Lambda Studio）
应用设计
服务 164
生成验证
运行时系统 170
（一个或多个）数据流
应用 190（例如，
（一个或多个）流水线、
（一个或多个）Lambda 应用）
可重用
的模板
绑定
系统 HUB 166
模式/模板 906
使用...
来定义
模式
数据 AI 子系统 168
推荐引擎 912
实体
系统 HUB 中的
类型注释的 HUB/
数据集/实体/属性 910
规则引擎 908
具有...类型
具有...类型
服务注册表 902
功能和业务类型
注册表 904

图48

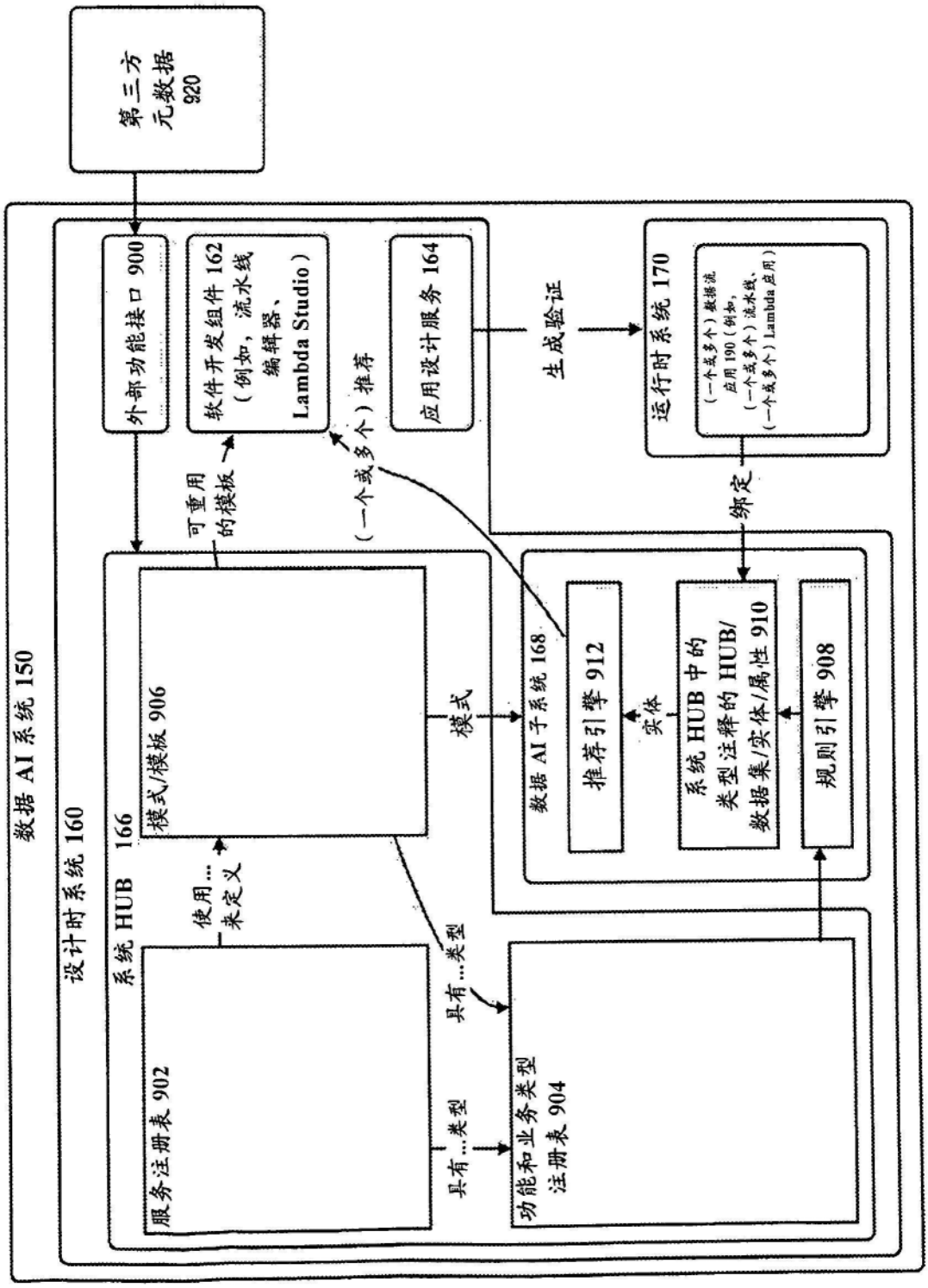

第三方元数据 920
数据 AI 系统 150
设计时系统 160
外部功能接口 900
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
应用设计服务 164
生成验证
运行时系统 170
（一个或多个）数据流应用 190（例如，（一个或多个）流水线、（一个或多个）Lambda 应用）
绑定
可重用的模板
（一个或多个）推荐
系统 HUB 166
模式/模板 906
使用...来定义
模式
数据 AI 子系统 168
推荐引擎 912
实体
系统 HUB 中的类型注释的 HUB/数据集/实体/属性 910
规则引擎 908
具有...类型
具有...类型
服务注册表 902
功能和业务类型注册表 904

图49

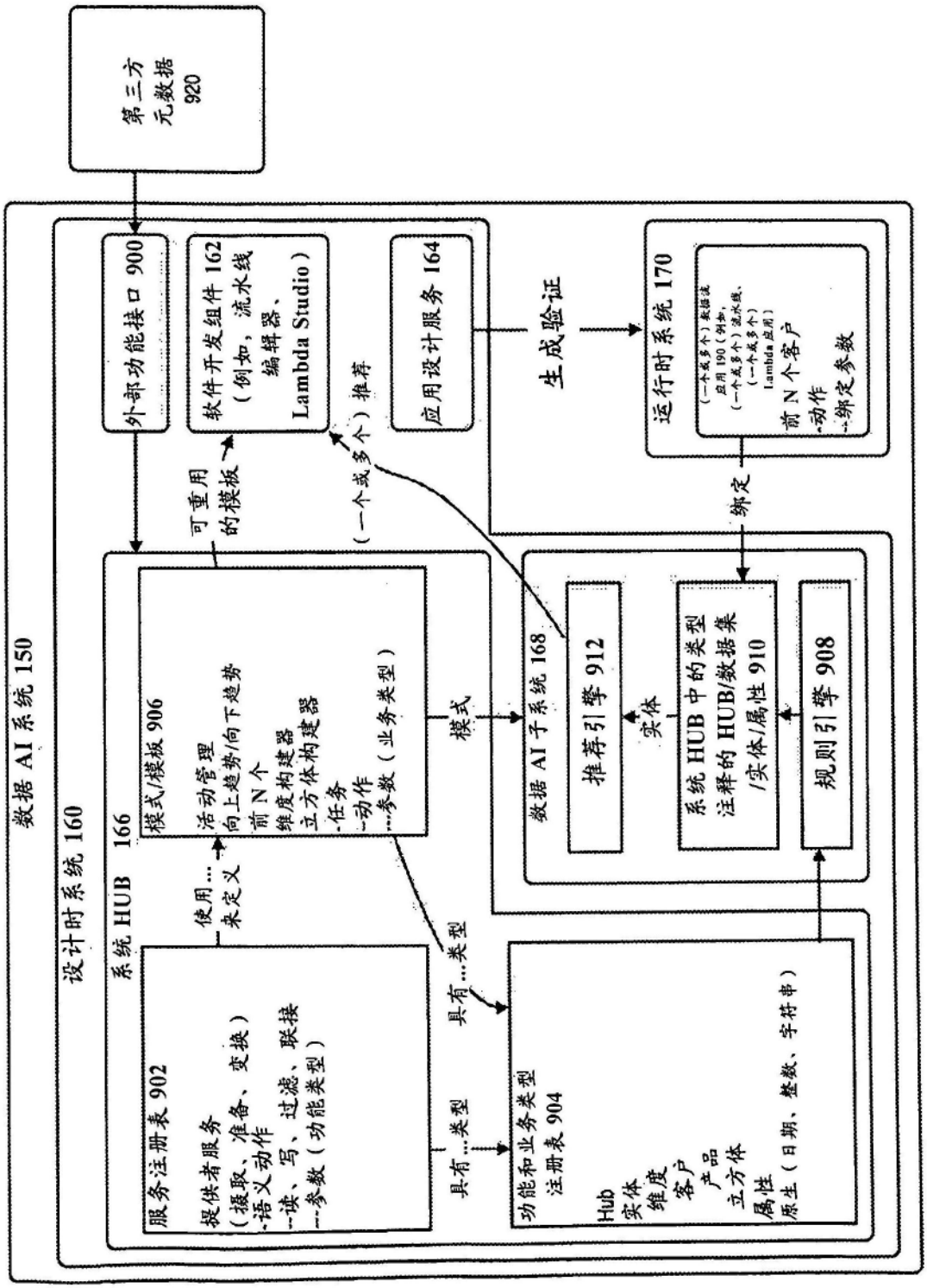

第三方元数据 920
数据 AI 系统 150
设计时系统 160
外部功能接口 900
软件开发组件 162（例如，流水线编辑器、Lambda Studio）
应用设计服务 164
生成验证
运行时系统 170
前 N 个客户
-动作
-绑定参数
绑定
可重用的模板
（一个或多个）推荐
系统 HUB 166
模式/模板 906
活动管理
向上趋势/向下趋势
前 N 个
维度构建器
立方体构建器
-任务
-动作
---参数（业务类型）
使用...来定义
模式
数据 AI 子系统 168
推荐引擎 912
实体
系统 HUB 中的类型注释的 HUB/数据集/实体/属性 910
规则引擎 908
服务注册表 902
提供者服务（摄取、准备、变换）
-语义动作
--读、写、过滤、联接
---参数（功能类型）
具有...类型
具有...类型
功能和业务类型注册表 904
Hub
实体
维度
客户
产品
立方体
属性
原生（日期、整数、字符串）

图50

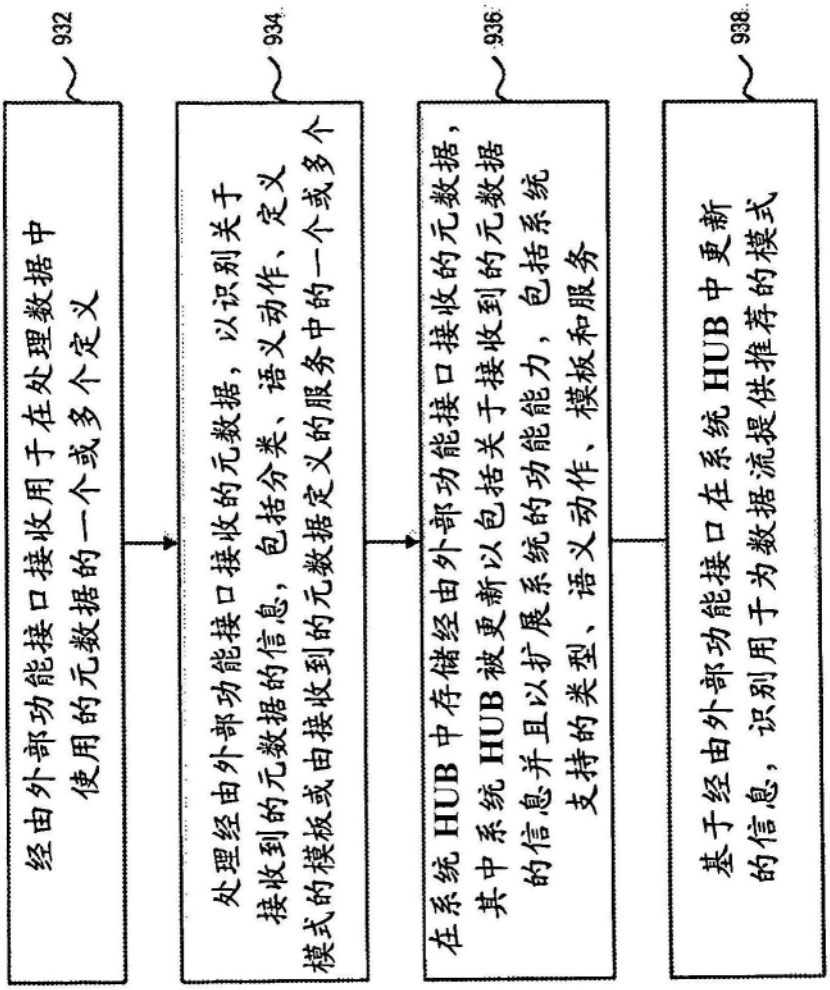

932
经由外部功能接口接收用于在处理数据中使用的元数据的一个或多个定义
934
处理经由外部功能接口接收的元数据，以识别关于接收到的元数据的信息，包括分类、语义动作、定义模式的模板或由接收到的元数据定义的服务中的一个或多个
936
在系统 HUB 中存储经由外部功能接口接收的元数据，其中系统 HUB 被更新以包括关于接收到的元数据的信息并且以扩展系统的功能能力，包括系统支持的类型、语义动作、模板和服务
938
基于经由外部功能接口在系统 HUB 中更新的信息，识别用于为数据流提供推荐的模式

图51

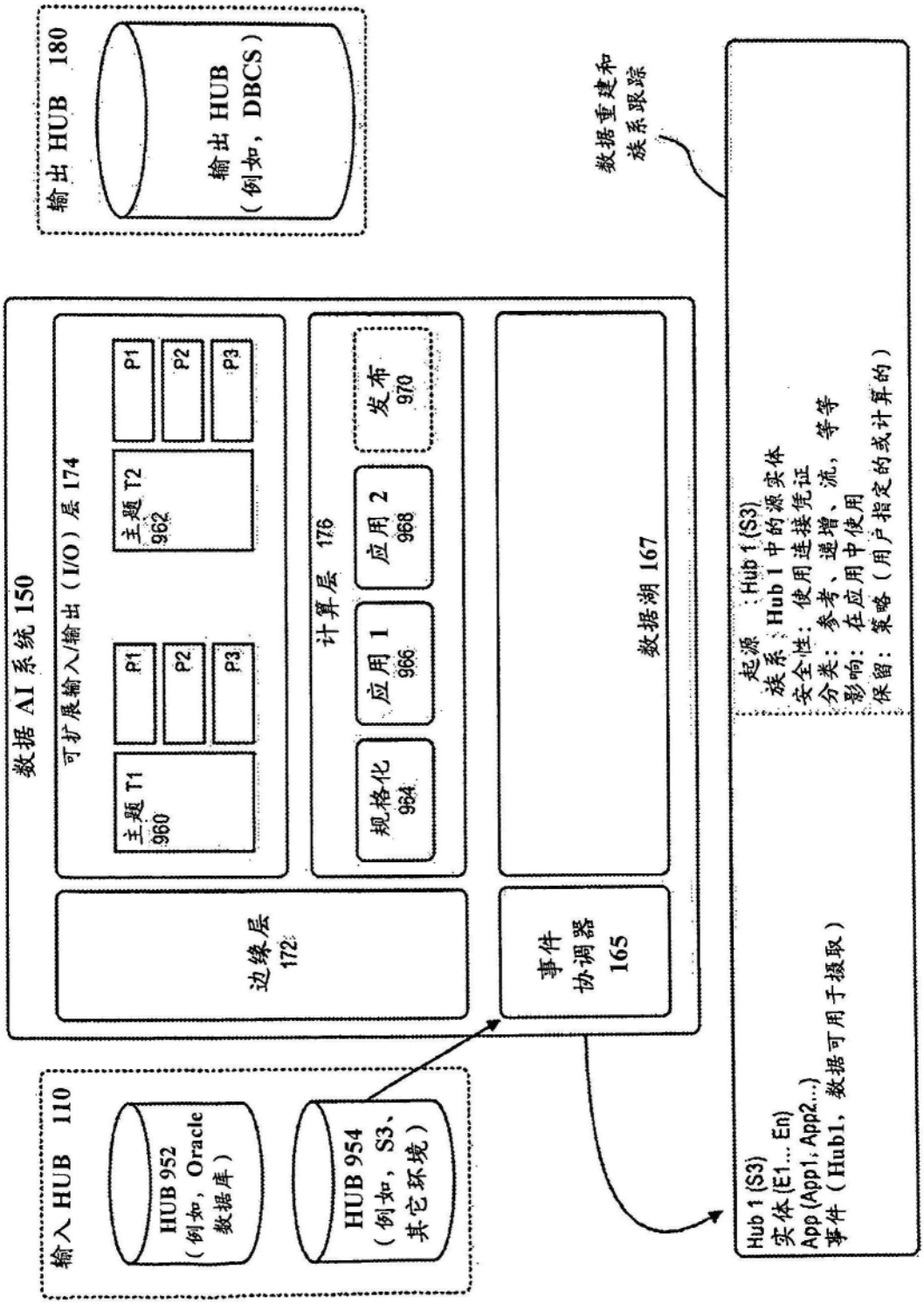

输出 HUB 180
输出 HUB
（例如，DBCS）
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
P1
P2
P3
主题 T2
962
P1
P2
P3
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
边缘层
172
事件
协调器
165
数据湖 167
起源：Hub 1 (S3)
族系：Hub 1 中的源实体
安全性：使用连接凭证
分类：参考、递增、流，等等
影响：在应用中使用
保留：策略（用户指定的或计算的）
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）
Hub 1 (S3)
实体 (E1... En)
App (App1, App2...)
事件（Hub1，数据可用于摄取）

图52

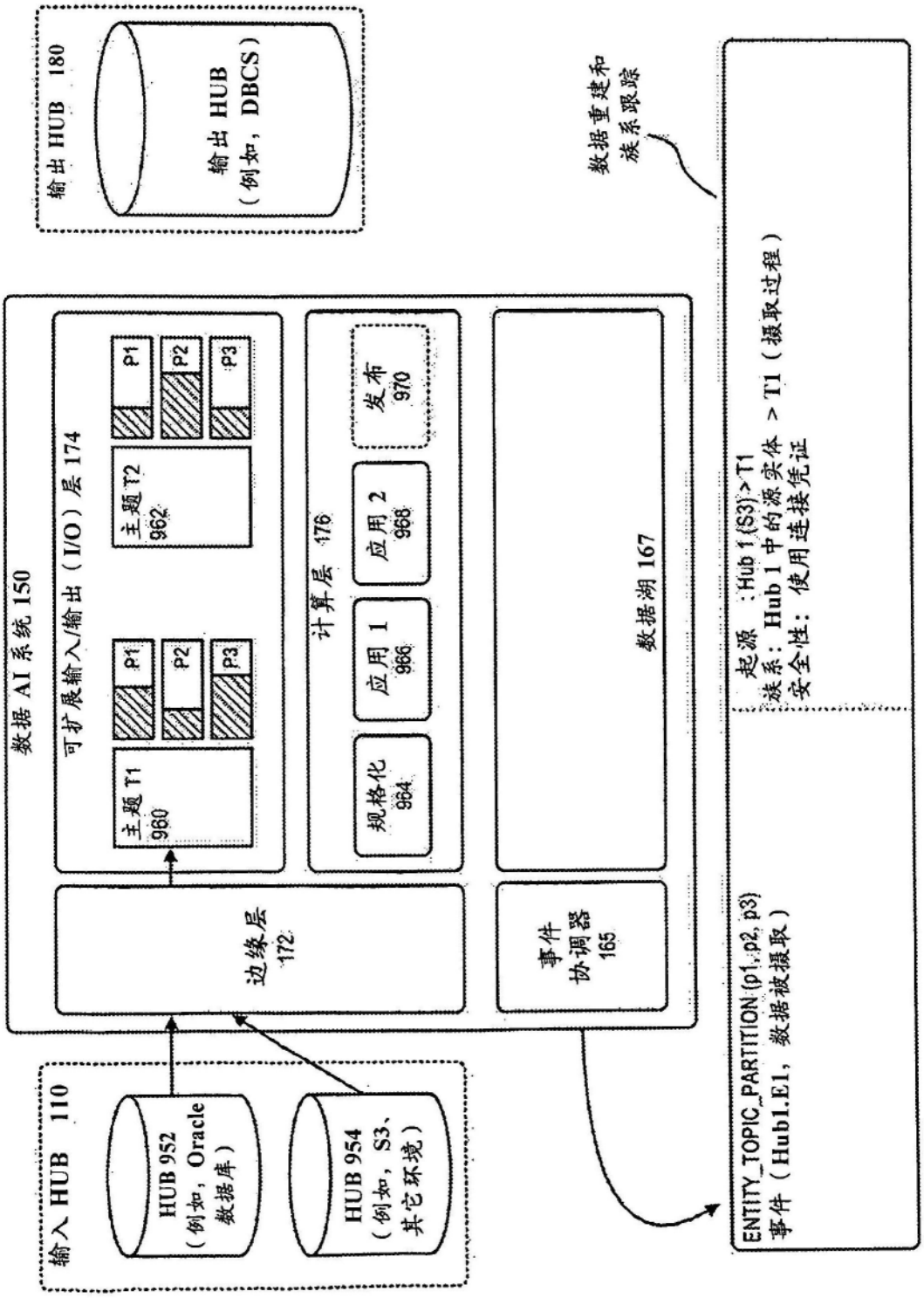

输出 HUB 180
输出 HUB
（例如，DBCS）
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
P1
P2
P3
主题 T2
962
P1
P2
P3
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
数据湖 167
边缘层
172
事件
协调器
165
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）
起源 :Hub 1(S3)>T1
族系：Hub 1 中的源实体 > T1（摄取过程）
安全性：使用连接凭证
ENTITY_TOPIC_PARTITION (p1, p2, p3)
事件（Hub1.E1，数据被摄取）

图53

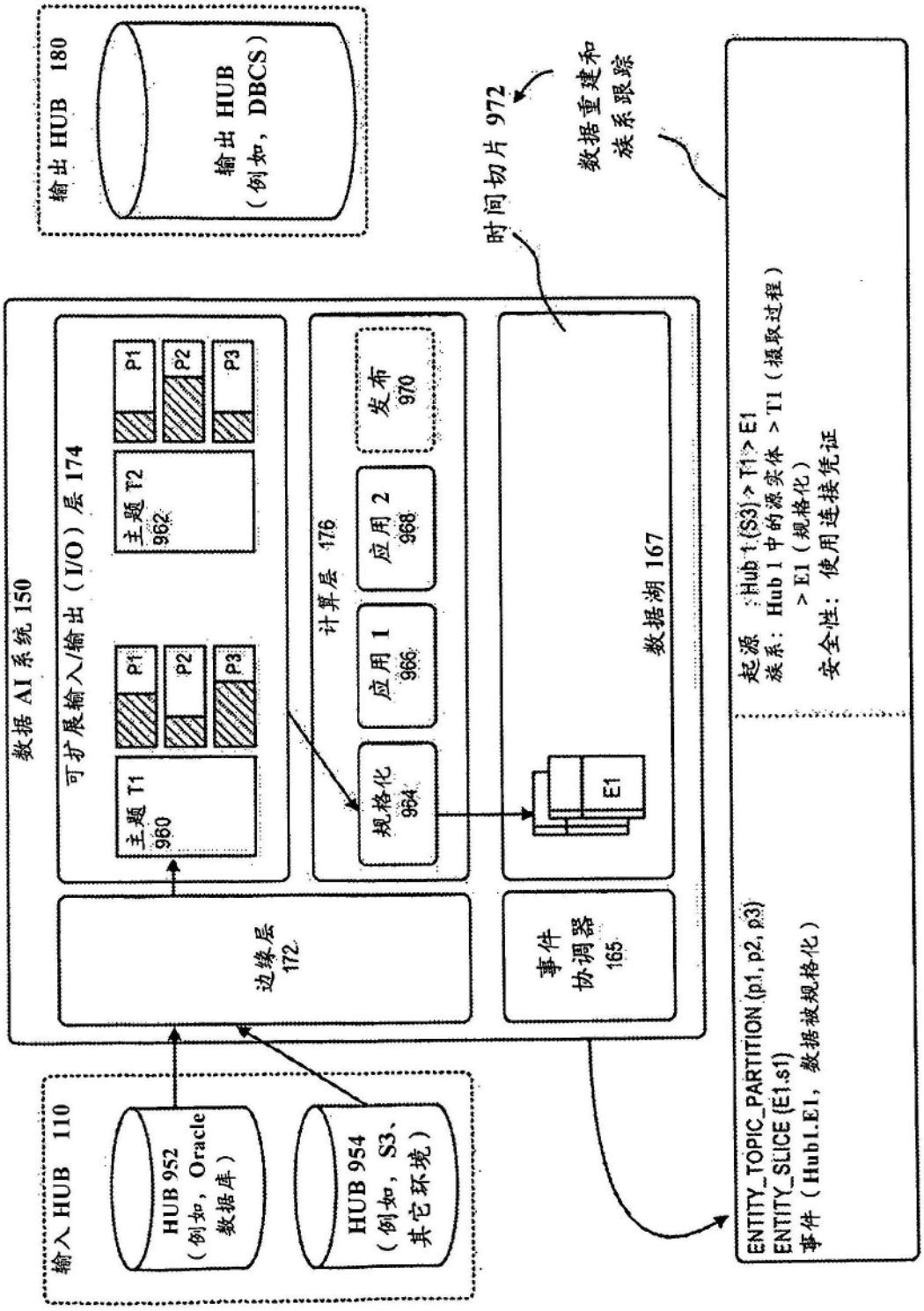

输出 HUB 180
输出 HUB
（例如，DBCS）
时间切片 972
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
P1
P2
P3
主题 T2
962
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
数据湖 167
E1
边缘层
172
事件
协调器
165
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）
起源 ：Hub 1 (S3) > T1 > E1
族系：Hub 1 中的源实体 > T1（摄取过程）
> E1（规格化）
安全性：使用连接凭证
ENTITY_TOPIC_PARTITION (p1, p2, p3)
ENTITY_SLICE (E1.s1)
事件（Hub1.E1，数据被规格化）

图54

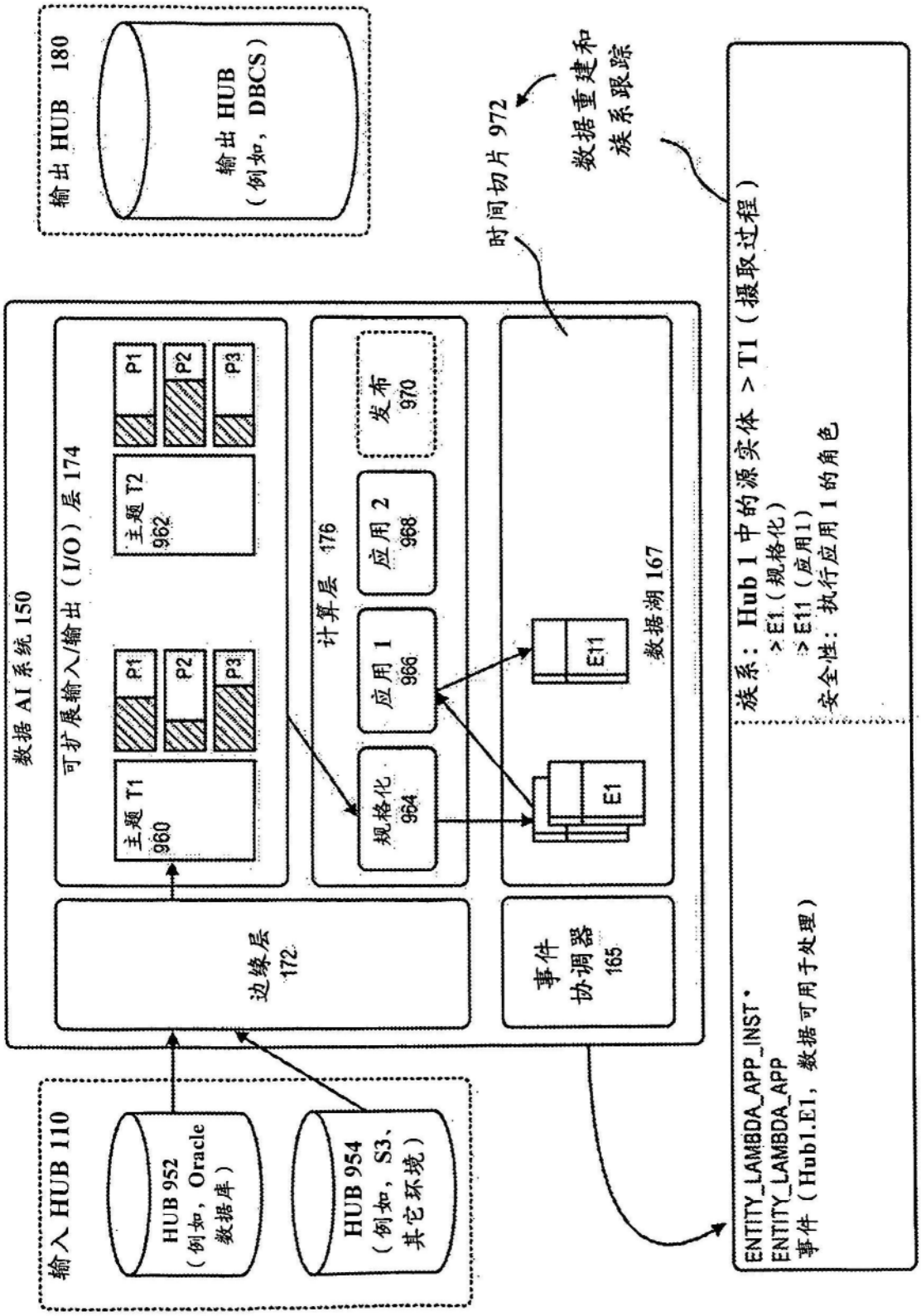

输出 HUB 180
输出 HUB
（例如，DBCS）
时间切片 972
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
主题 T2
962
P1
P2
P3
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
数据湖 167
E1
E11
边缘层
172
事件
协调器
165
族系：Hub 1 中的源实体 > T1（摄取过程）
> E1（规格化）
> E11（应用1）
安全性：执行应用 1 的角色
ENTITY_LAMBDA_APP_INST*
ENTITY_LAMBDA_APP
事件（Hub1.E1，数据可用于处理）
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）

图55

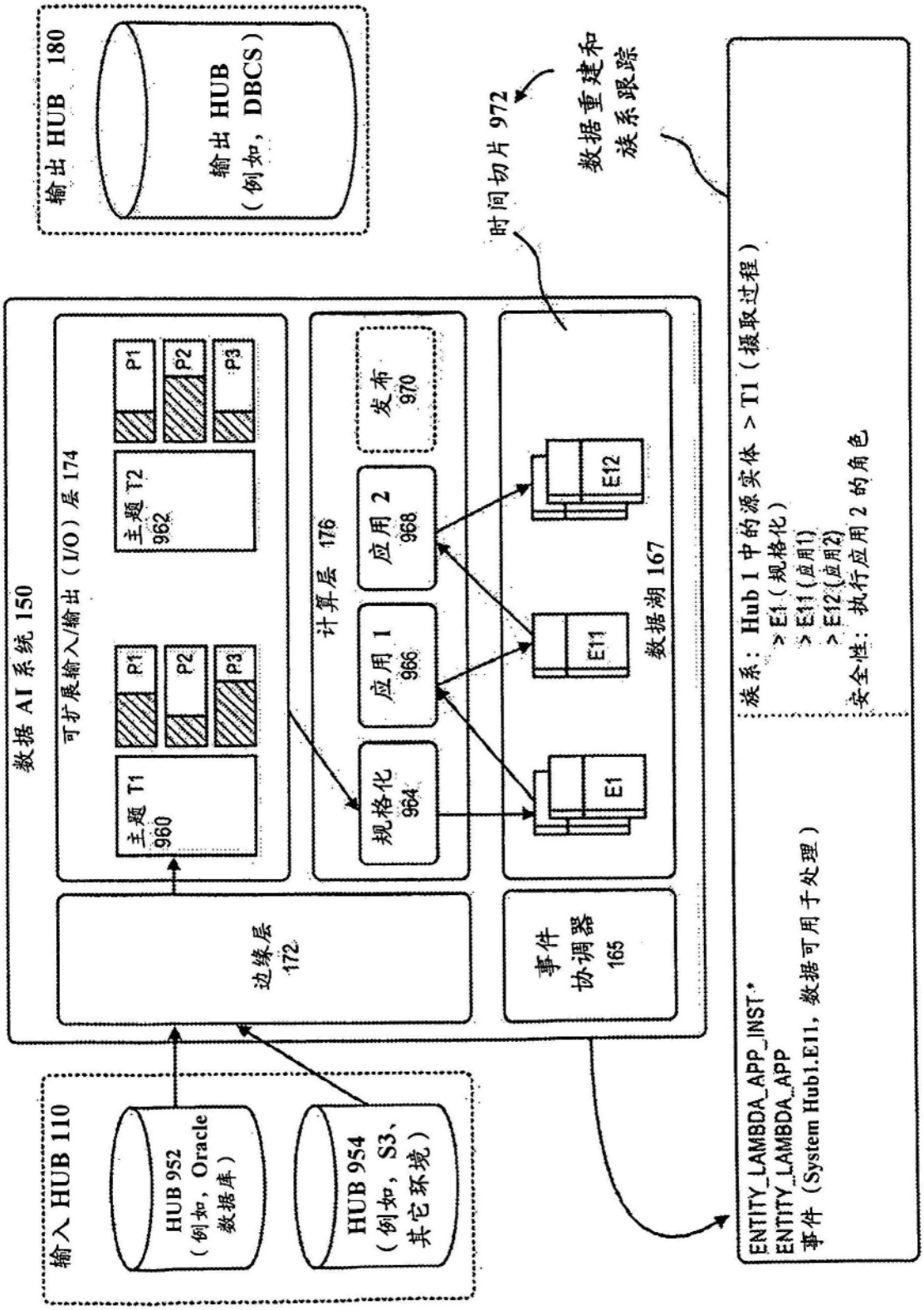

输出 HUB 180
输出 HUB
（例如，DBCS）
时间切片 972
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
主题 T2
962
P1
P2
P3
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
数据湖 167
E1
E11
E12
边缘层
172
事件
协调器
165
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）
族系：Hub 1 中的源实体 > T1（摄取过程）
> E1（规格化）
> E11（应用1）
> E12（应用2）
安全性：执行应用 2 的角色
ENTITY_LAMBDA_APP_INST'
ENTITY_LAMBDA_APP
事件（System Hub1.E11，数据可用于处理）

图56

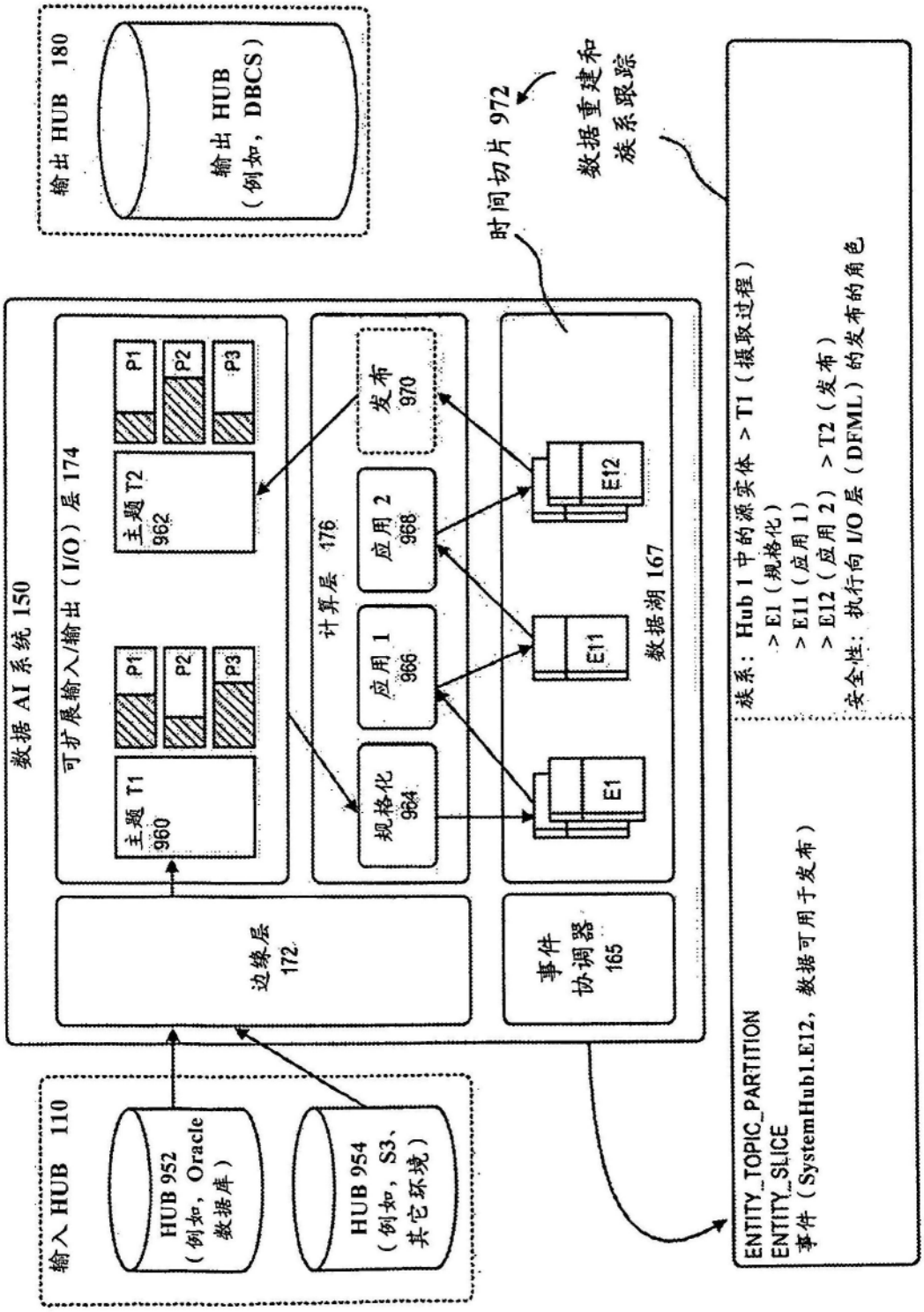

输出 HUB 180
输出 HUB
（例如，DBCS）
时间切片 972
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
主题 T2
962
P1
P2
P3
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
数据湖 167
E1
E11
E12
边缘层
172
事件
协调器
165
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）
族系：Hub 1 中的源实体 > T1（摄取过程）
> E1（规格化）
> E11（应用 1）
> E12（应用 2）> T2（发布）
安全性：执行向 I/O 层（DFML）的发布的角色
ENTITY_TOPIC_PARTITION
ENTITY_SLICE
事件（SystemHub1.E12，数据可用于发布）

图57

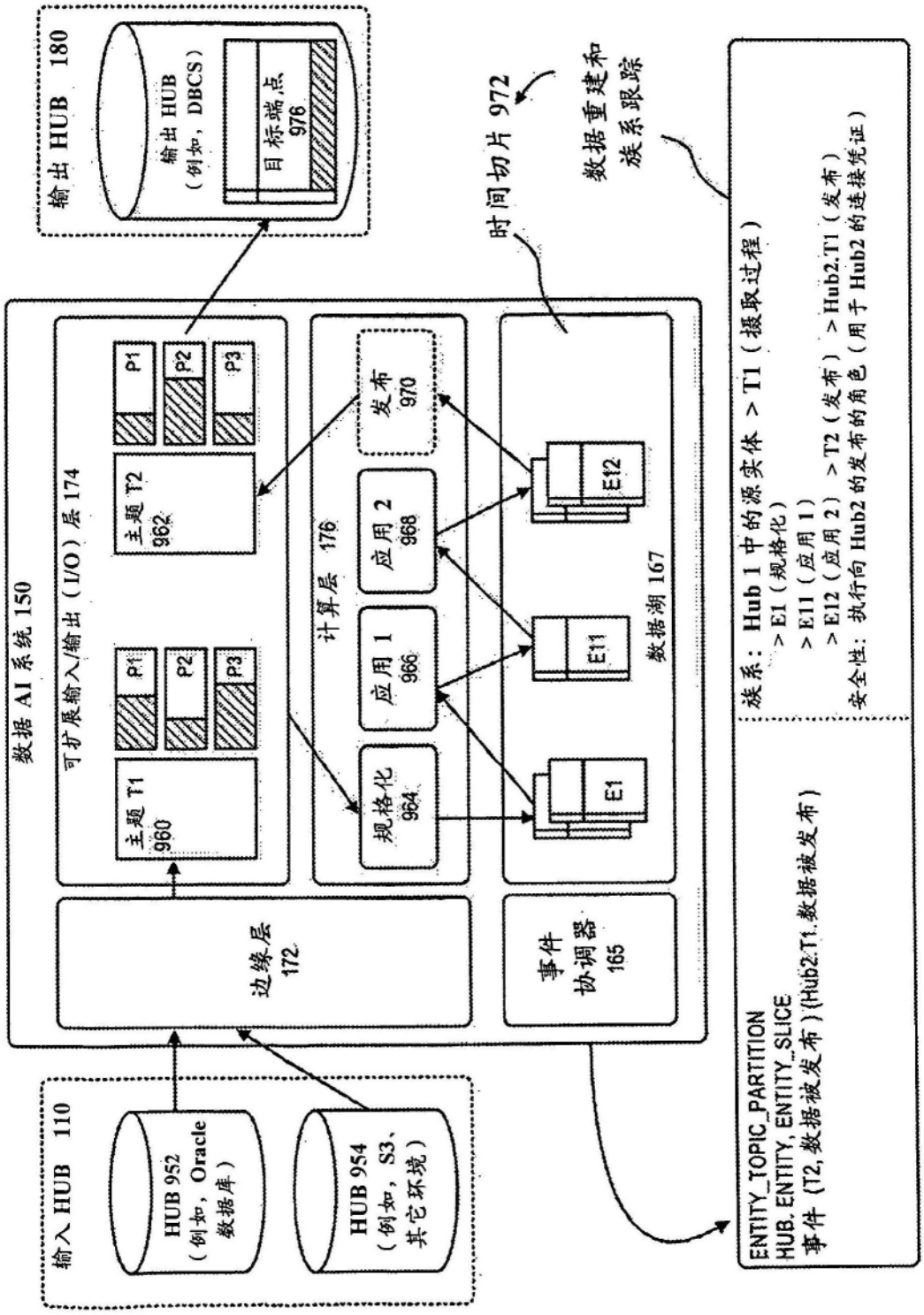

输出 HUB 180
输出 HUB
（例如，DBCS）
目标端点
976
时间切片 972
数据重建和
族系跟踪
数据 AI 系统 150
可扩展输入/输出（I/O）层 174
主题 T1
960
主题 T2
962
P1
P2
P3
计算层 176
规格化
964
应用 1
966
应用 2
968
发布
970
数据湖 167
E1
E11
E12
边缘层
172
事件
协调器
165
族系：Hub 1 中的源实体 > T1（摄取过程）
> E1（规格化）
> E11（应用 1）
> E12（应用 2）> T2（发布）> Hub2.T1（发布）
安全性：执行向 Hub2 的发布的角色（用于 Hub2 的连接凭证）
ENTITY_TOPIC_PARTITION
HUB.ENTITY,ENTITY_SLICE
事件（T2,数据被发布）(Hub2.T1.数据被发布）
输入 HUB 110
HUB 952
（例如，Oracle
数据库）
HUB 954
（例如，S3、
其它环境）

图58

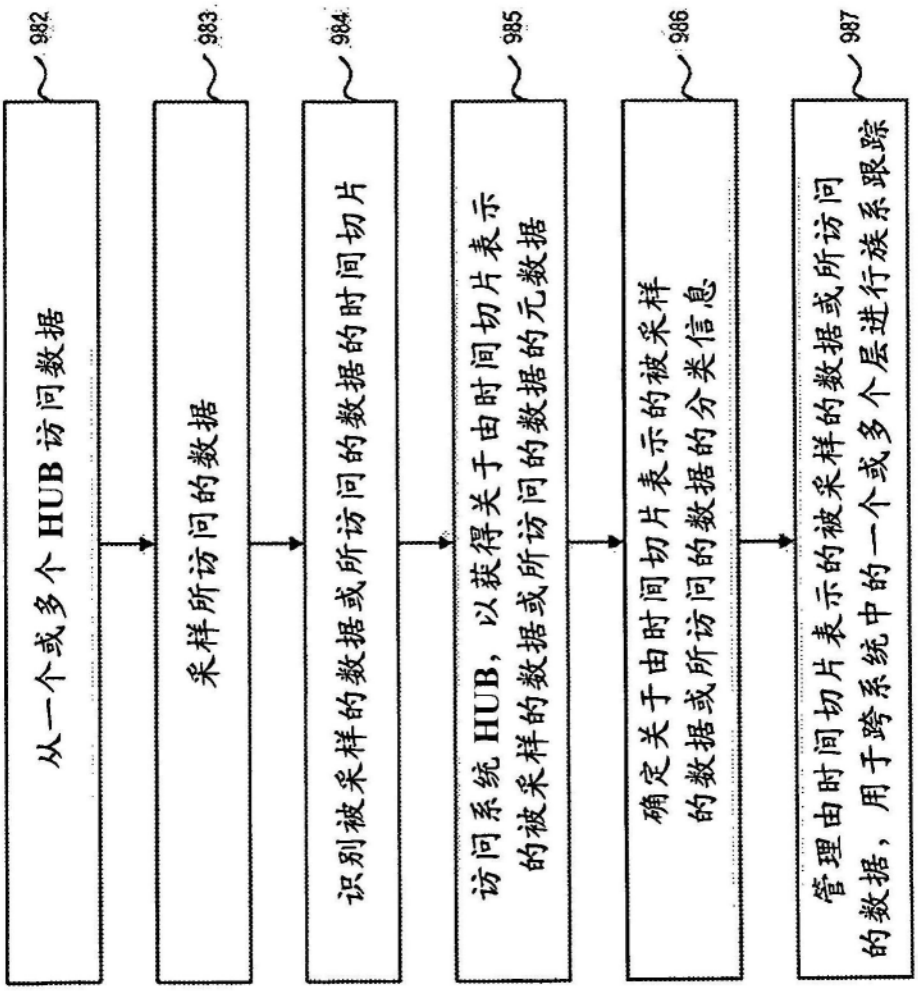

982
从一个或多个 HUB 访问数据
983
采样所访问的数据
984
识别被采样的数据或所访问的数据的时间切片
985
访问系统 HUB，以获得关于由时间切片表示
的被采样的数据或所访问的数据的元数据
986
确定关于由时间切片表示的被采样
的数据或所访问的数据的分类信息
987
管理由时间切片表示的被采样的数据或所访问
的数据，用于跨系统中的一个或多个层进行族系跟踪

图59