(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104685467 B
(45)授权公告日 2018.06.29

(21)申请号 201380050839.X
(22)申请日 2013.09.27
(65)同一申请的已公布的文献号
申请公布号 CN 104685467 A
(43)申请公布日 2015.06.03
(30)优先权数据
61/707,343 2012.09.28 US
(85)PCT国际申请进入国家阶段日
2015.03.27
(86)PCT国际申请的申请数据
PCT/US2013/062369 2013.09.27
(87)PCT国际申请的公布数据
WO2014/052873 EN 2014.04.03

(73)专利权人 起元技术有限责任公司
地址 美国马萨诸塞州
(72)发明人 猪饲太郎 A.安德森
(74)专利代理机构 北京林达刘知识产权代理事务所(普通合伙) 11277
代理人 刘新宇

(51)Int.Cl.
G06F 8/38(2018.01)
G06F 8/34(2018.01)
(56)对比文件
CN 101971165 A,2011.02.09,
CN 102239458 A,2011.11.09,
WO 01/82068 A1,2001.01.11,
审查员 祝子豪

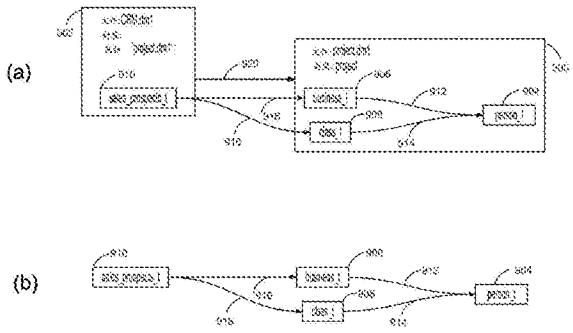
权利要求书4页 说明书13页 附图12页

(54)发明名称

图形地表示编程属性

(57)摘要

一种用于表示信息的计算系统，所述计算系统包括被配置为处理信息的至少一个处理器。所述处理包括：定义数据结构，所述数据结构表示用于开发应用的至少一个编程属性的层级。所述数据结构被存储在文件中，以允许所述数据结构被在其他文件中存储的其他数据结构使用。所述处理也包括：产生可视图，所述可视图包括所述数据结构的图形表示和用于存储所述数据结构的所述文件的图形表示。所述可视图也包括在所述数据结构和另一个数据结构之间的关系的图形表示和在用于存储所述数据结构的所述文件和用于存储所述另一个数据结构的另一个文件之间的关系的图形表示。所述计算机系统也包括输出装置，用于呈现所述可视图，所述可视图包括在所述数据结构和所述文件的图形表示与所述数据结构的关系和所述文件的关系的图形表示。



CN 104685467 B

1.一种用于表示信息的方法,包括:

定义第一数据结构,所述第一数据结构表示用于开发应用的至少一个编程属性的层级,其中,所述第一数据结构被存储在第一文件中,以允许所述第一数据结构被在其他文件中存储的其他数据结构使用,并且所述第一数据结构是基于针对第二文件中所存储的第二数据结构的定义而定义的;并且

产生可视图,所述可视图包括所述第一数据结构、所述第二数据结构、用于存储所述第一数据结构的所述第一文件以及用于存储所述第二数据结构的所述第二文件各自的图形表示,所述可视图还包括:

表示所述第一数据结构和所述第二数据结构之间的关系的第一图形特征,以及

表示用于存储所述第一数据结构的所述第一文件和用于存储所述第二数据结构的所述第二文件之间的关系的第二图形特征。

2.根据权利要求1所述的方法,其中,文件的图形表示是可移除的,以允许操纵数据结构的图形表示以定义一个或多个数据结构组,并且建立用于存储所述一个或多个数据结构组的一个或多个新的文件。

3.根据权利要求1所述的方法,进一步包括:

操纵所定义的数据结构以调整所述至少一个编程属性。

4.根据权利要求1所述的方法,进一步包括:

操纵所定义的数据结构以向文件内插入所述至少一个编程属性。

5.根据权利要求4所述的方法,其中,向所述文件内插入具有所述至少一个编程属性的语句。

6.根据权利要求4所述的方法,其中,操纵所定义的数据结构包括拖放操作。

7.根据权利要求3所述的方法,其中,操纵所定义的数据结构包括将所述编程属性的所述层级的内容增加、删除和编辑的至少一个。

8.根据权利要求1所述的方法,其中,所述编程属性是命名的数据类型。

9.根据权利要求1所述的方法,其中,所述编程属性是函数。

10.根据权利要求1所述的方法,其中,所述第一数据结构和所述第二数据结构之间的关系表示所述编程属性的族系。

11.根据权利要求1所述的方法,其中,表示所述第一数据结构和所述第二数据结构之间的关系的所述第一图形特征包括使所述第一数据结构的图形表示和所述第二数据结构的图形表示连接的第一图形特征,以及表示所述第一文件和所述第二文件之间的关系的所述第二图形特征包括使所述第一文件的图形表示和所述第二文件的图形表示连接的第二图形特征。

12.根据权利要求1所述的方法,其中,在所述可视图中,所述第一数据结构的图形表示在所述第一文件的图形表示内,并且所述第二数据结构的图形表示在所述第二文件的图形表示内。

13.一种用于表示信息的计算系统,所述计算系统包括:

被配置为处理信息的至少一个处理器,所述处理包括:

定义第一数据结构,所述第一数据结构表示用于开发应用的至少一个编程属性的层级,其中,所述第一数据结构被存储在第一文件中,以允许所述第一数据结构被在其他文件

中存储的其他数据结构使用，并且所述第一数据结构是基于针对第二文件中所存储的第二数据结构的定义而定义的；以及

产生可视图，所述可视图包括所述第一数据结构、所述第二数据结构、用于存储所述第一数据结构的所述第一文件以及用于存储所述第二数据结构的所述第二文件各自的图形表示，所述可视图还包括：

表示所述第一数据结构和所述第二数据结构之间的关系的第一图形特征，以及

表示用于存储所述第一数据结构的所述第一文件和用于存储所述第二数据结构的所述第二文件之间的关系的第二图形特征；以及

输出装置，用于呈现所述可视图，所述可视图包括所述第一数据结构、所述第二数据结构、所述第一文件和所述第二文件的图形表示以及表示所述第一数据结构和所述第二数据结构之间的关系的所述第一图形特征和表示用于存储所述第一数据结构的所述第一文件和用于存储所述第二数据结构的所述第二文件之间的关系的所述第二图形特征。

14.根据权利要求13所述的计算系统，其中，文件的图形表示是可移除的，以允许操纵数据结构的图形表示以定义一个或多个数据结构组，并且建立用于存储所述一个或多个数据结构组的一个或多个新的文件。

15.根据权利要求13所述的计算系统，所述处理还包括：

操纵所定义的数据结构以调整所述至少一个编程属性。

16.根据权利要求13所述的计算系统，所述处理还包括：

操纵所定义的数据结构以向文件内插入所述至少一个编程属性。

17.根据权利要求16所述的计算系统，其中，向所述文件内插入具有所述至少一个编程属性的语句。

18.根据权利要求16所述的计算系统，其中，操纵所定义的数据结构包括拖放操作。

19.根据权利要求15所述的计算系统，其中，操纵所定义的数据结构包括将所述编程属性的所述层级的内容增加、删除和编辑的至少一个。

20.根据权利要求13所述的计算系统，其中，所述编程属性是命名的数据类型。

21.根据权利要求13所述的计算系统，其中，所述编程属性是函数。

22.根据权利要求13所述的计算系统，其中，所述第一数据结构和所述第二数据结构之间的关系表示所述编程属性的族系。

23.一种用于表示信息的计算系统，所述计算系统包括：

用于定义第一数据结构的部件，所述第一数据结构表示用于开发应用的至少一个编程属性的层级，其中，所述第一数据结构被存储在第一文件中，以允许所述第一数据结构被在其他文件中存储的其他数据结构使用，并且所述第一数据结构是基于针对第二文件中所存储的第二数据结构的定义而定义的；以及

用于产生可视图的部件，所述可视图包括所述第一数据结构、所述第二数据结构、用于存储所述第一数据结构的所述第一文件以及用于存储所述第二数据结构的所述第二文件各自的图形表示，所述可视图还包括：

表示所述第一数据结构和所述第二数据结构之间的关系的第一图形特征，以及

表示用于存储所述第一数据结构的所述第一文件和用于存储所述第二数据结构的所述第二文件之间的关系的第二图形特征。

24.根据权利要求23所述的计算系统，其中，文件的图形表示是可移除的，以允许操纵数据结构的图形表示以定义一个或多个数据结构组，并且建立用于存储所述一个或多个数据结构组的一个或多个新的文件。

25.根据权利要求23所述的计算系统，还包括：

用于操纵所定义的数据结构以调整所述至少一个编程属性的部件。

26.根据权利要求23所述的计算系统，还包括：

用于操纵所定义的数据结构以向文件内插入所述至少一个编程属性的部件。

27.根据权利要求26所述的计算系统，其中，向所述文件内插入具有所述至少一个编程属性的语句。

28.根据权利要求26所述的计算系统，其中，操纵所定义的数据结构包括拖放操作。

29.根据权利要求25所述的计算系统，其中，操纵所定义的数据结构包括将所述编程属性的所述层级的内容增加、删除和编辑的至少一个。

30.根据权利要求23所述的计算系统，其中，所述编程属性是命名的数据类型。

31.根据权利要求23所述的计算系统，其中，所述编程属性是函数。

32.根据权利要求23所述的计算系统，其中，所述第一数据结构和所述第二数据结构之间的关系表示所述编程属性的族系。

33.根据权利要求23所述的计算系统，其中，表示所述第一数据结构和所述第二数据结构之间的关系的所述第一图形特征包括使所述第一数据结构的图形表示和所述第二数据结构的图形表示连接的第一图形特征，以及表示所述第一文件和所述第二文件之间的关系的所述第二图形特征包括使所述第一文件的图形表示和所述第二文件的图形表示连接的第二图形特征。

34.根据权利要求23所述的计算系统，其中，在所述可视图中，所述第一数据结构的图形表示在所述第一文件的图形表示内，并且所述第二数据结构的图形表示在所述第二文件的图形表示内。

35.一种计算机可读存储介质，用于存储用于表示信息的计算机程序，所述计算机程序包括用于使得计算系统进行下述行为的指令：

定义第一数据结构，所述第一数据结构表示用于开发应用的至少一个编程属性的层级，其中，所述第一数据结构被存储在第一文件中，以允许所述第一数据结构被在其他文件中存储的其他数据结构使用，并且所述第一数据结构是基于针对第二文件中所存储的第二数据结构的定义而定义的；并且

产生可视图，所述可视图包括所述第一数据结构、所述第二数据结构、用于存储所述第一数据结构的所述第一文件以及用于存储所述第二数据结构的所述第二文件各自的图形表示，所述可视图还包括：

表示所述第一数据结构和所述第二数据结构之间的关系的第一图形特征，以及

表示用于存储所述第一数据结构的所述第一文件和用于存储所述第二数据结构的所述第二文件之间的关系的第二图形特征。

36.根据权利要求35所述的计算机可读存储介质，其中，文件的图形表示是可移除的，以允许操纵数据结构的图形表示以定义一个或多个数据结构组，并且建立用于存储所述一个或多个数据结构组的一个或多个新的文件。

37.根据权利要求35所述的计算机可读存储介质，所述计算机程序进一步包括用于使得计算系统进行下述行为的指令：

操纵所定义的数据结构以调整所述至少一个编程属性。

38.根据权利要求35所述的计算机可读存储介质，所述计算机程序进一步包括用于使得计算系统进行下述行为的指令：

操纵所定义的数据结构以向文件内插入所述至少一个编程属性。

39.根据权利要求38所述的计算机可读存储介质，其中，向所述文件内插入具有所述至少一个编程属性的语句。

40.根据权利要求38所述的计算机可读存储介质，其中，操纵所定义的数据结构包括拖放操作。

41.根据权利要求37所述的计算机可读存储介质，其中，操纵所定义的数据结构包括将所述编程属性的所述层级的内容增加、删除和编辑的至少一个。

42.根据权利要求35所述的计算机可读存储介质，其中，所述编程属性是命名的数据类型。

43.根据权利要求35所述的计算机可读存储介质，其中，所述编程属性是函数。

44.根据权利要求35所述的计算机可读存储介质，其中，所述第一数据结构和所述第二数据结构之间的关系表示所述编程属性的族系。

45.根据权利要求35所述的计算机可读存储介质，其中，表示所述第一数据结构和所述第二数据结构之间的关系的所述第一图形特征包括使所述第一数据结构的图形表示和所述第二数据结构的图形表示连接的第一图形特征，以及表示所述第一文件和所述第二文件之间的关系的所述第二图形特征包括使所述第一文件的图形表示和所述第二文件的图形表示连接的第二图形特征。

46.根据权利要求35所述的计算机可读存储介质，其中，在所述可视图中，所述第一数据结构的图形表示在所述第一文件的图形表示内，并且所述第二数据结构的图形表示在所述第二文件的图形表示内。

图形地表示编程属性

[0001] 优先权的要求

[0002] 本申请要求对于在2012年9月28日提交的美国专利申请No.61/707,343的在35 USC § 119(e)下的优先权,其整体内容通过引用被并入在此。

技术领域

[0003] 本说明书涉及一种用于表示编程属性的基于图形的手段。

背景技术

[0004] 复杂计算经常可以通过直接图形被表达为数据流(称为"数据流图形),计算的组件与图形的顶点和在与图形的链接(弧、边)对应的组件之间的数据流相关联。该组件可以包括:数据处理组件,其在一个或多个输入端口醋接收数据,处理数据,并且从一个或多个输出端口提供数据;以及,数据集组件,其作为数据流的源或汇。在美国专利5,966,072,EXECUTING COMPUTATIONS EXPRESSED AS GRAPHS(执行被表达为图形的计算)中描述了实现这样的基于图形的计算的系统。图形的组件可以接收、处理和输出各种类型的数据。因为类似的处理功能,等同类型的数据可以用于或再用于不同的应用。

发明内容

[0005] 在一个方面,一种用于表示信息的方法包括:定义数据结构,所述数据结构表示用于开发应用的至少一个编程属性的层级。所述数据结构被存储在文件中,以允许所述数据结构被在其他文件中存储的其他数据结构使用。所述方法也包括:产生可视图,所述可视图包括所述数据结构的图形表示和用于存储所述数据结构的所述文件的图形表示。所述可视图也包括在所述数据结构和另一个数据结构之间的关系的图形表示和在用于存储所述数据结构的所述文件和用于存储所述另一个数据结构的另一个文件之间的关系的图形表示。

[0006] 在另一个方面,一种计算机可读存储介质存储用于表示信息的计算机程序。所述计算机程序包括用于使得计算系统定义数据结构的指令,所述数据结构表示用于开发应用的至少一个编程属性的层级。所述数据结构被存储在文件中,以允许所述数据结构被在其他文件中存储的其他数据结构使用。所述指令也使得所述计算系统产生可视图,所述可视图包括所述数据结构的图形表示和用于存储所述数据结构的所述文件的图形表示。所述可视图也包括在所述数据结构和另一个数据结构之间的关系的图形表示和在用于存储所述数据结构的所述文件和用于存储所述另一个数据结构的另一个文件之间的关系的图形表示。

[0007] 在另一个方面,一种用于表示信息的计算系统包括被配置为处理信息的至少一个处理器。所述处理包括:定义数据结构,所述数据结构表示用于开发应用的至少一个编程属性的层级。所述数据结构被存储在文件中,以允许所述数据结构被在其他文件中存储的其他数据结构使用。所述处理也包括:产生可视图,所述可视图包括所述数据结构的图形表示和用于存储所述数据结构的所述文件的图形表示。所述可视图也包括在所述数据结构和另

一个数据结构之间的关系的图形表示和在用于存储所述数据结构的所述文件和用于存储所述另一个数据结构的另一个文件之间的关系的图形表示。所述计算机系统也包括输出装置，用于呈现所述可视图，所述可视图包括在所述数据结构和所述文件的图形表示与所述数据结构的关系和所述文件的关系的图形表示。

[0008] 在另一个方面，一种用于表示信息的计算系统包括用于处理的部件，所述处理包括：定义数据结构，所述数据结构表示用于开发应用的至少一个编程属性的层级。所述数据结构被存储在文件中，以允许所述数据结构被在其他文件中存储的其他数据结构使用。所述处理也包括：产生可视图，所述可视图包括所述数据结构的图形表示和用于存储所述数据结构的所述文件的图形表示。所述可视图也包括在所述数据结构和另一个数据结构之间的关系的图形表示和在用于存储所述数据结构的所述文件和用于存储所述另一个数据结构的另一个文件之间的关系的图形表示。所述计算机系统也包括用于呈现所述可视图的部件，所述可视图包括在所述数据结构和所述文件的图形表示与所述数据结构的关系和所述文件的关系的图形表示。

[0009] 实现方式可以包括下面的特征的任何一个或全部。所述文件的所述图形表示可以可移除，以允许操纵所述数据结构的所述图形表示以定义一个或多个数据结构组，并且建立用于存储所述一个或多个数据结构组的一个或多个新的文件。可以操纵所述定义的数据结构以调整所述至少一个编程属性。可以操纵所述定义的数据结构以向文件内插入所述至少一个编程属性。可以向所述文件内插入具有所述至少一个编程属性的语句。操纵所述数据结构可以包括拖放操作。操纵所述定义的数据结构可以包括将所述编程属性的所述层级的内容增加、删除和编辑的至少一个。所述编程属性可以是命名的数据类型、函数等。在所述数据结构和所述另一个数据结构之间的关系可以表示所述编程属性的族系。

[0010] 各方面可以包括下面的优点的一个或多个。

[0011] 图形地表示编程属性(例如，字段、命名的数据类型结构、函数等)允许开发者较快地确定属性细节(例如，在被使用的数据类型)和在属性(例如，使用先前定义的字段的命名的数据类型结构)和用于存储所述属性的文件之间的关系。可以有效地操纵以图形形式表示的属性、用于存储所述属性的文件等，以例如编辑命名的数据类型结构，并且由此根据需要允许改变传播。

[0012] 通过下面的说明书和所附的权利要求，本发明的其他特征和优点将变得清楚。

附图说明

[0013] 图1是用于执行基于图形的计算的系统的框图。

[0014] 图2-6是用于呈现数据类型信息的用户界面。

[0015] 图7-12是用于呈现数据类型信息的图形表示的用户界面。

[0016] 图13是示例性编程属性呈现过程的流程图。

具体实施方式

[0017] 图1示出了示例性数据处理系统100，其中，诸如数据类型、可执行函数等的编程属性可以被图形地呈现，以例如允许任意的观看者有效地确定属性的内容、层级和族系。通常，为了提供这样的功能，系统100包括数据源102，其可以包括诸如存储装置或到在线数据

流的连接的数据的一个或多个源，其中每一个可以以多种存储格式(例如，数据库表、电子表格文件、平面文本文件或由大型计算机使用的本机格式)的任何一种来存储数据。在这个示例中，执行环境104包括预处理模块106和执行模块112。执行环境104可以在诸如UNIX操作系统的适当的操作系统的可移除被容纳在一个或多个通用计算机上。例如，执行环境104可以包括多节点并行计算环境，其包括计算机系统的配置，该计算机系统使用本地的(例如，诸如SMP计算机的多处理器系统)或本地分布的(例如，作为簇或MPP耦合的多处理器)或远程的或远程分布的(例如，经由局域网(LAN)和/或广域网(WAN)耦合的多个处理器)或其任何组合的多个中央处理单元(CPU)。

[0018] 预处理模块106从数据源102读取数据，并且执行例如预期由其他模块的进一步进行的对应的处理操作。提供数据源102的存储装置可以是执行环境104本地的，例如，被存储在连接到运行执行环境104的计算机的存储介质(例如，硬盘驱动器108)上，或者可以远离执行环境104，例如，被容纳在通过远程连接与运行执行环境104的计算机进行通信的远程系统(例如，大型计算机110)上。

[0019] 执行模块112使用由预处理模块106产生的处理的数据来例如处理在可访问执行环境104的数据存储系统116中存储的数据114(例如，企业数据、公司记录等)。数据存储系统116也能够访问开发环境118，其中，开发者120能够将在数据存储系统116中存储的信息检查、编辑等。在一些布置中，开发环境118可以用于准备和调整执行环境104以执行期望的操作。例如，开发环境118可以是用于开发作为数据流图形的应用的系统，该数据流图形包括通过在顶点之间的直接链接(用于表示工作元件的流)连接的顶点(用于表示组件或数据集)。例如，在通过引用并入在此的、题目为“Managing Parameters for Graph-Based Applications(管理用于基于图形的应用的参数)”的美国公布No.2007/0011668中更详细地描述了这样的环境。在通过引用并入在此的美国专利5,566,072,EXECUTING COMPUTATIONS EXPRESSED AS GRAPHS(执行被表达为图形的计算)中描述了用于执行这样的基于图形的计算的系统。根据该系统作出的数据流图形提供了方法，用于向由图形组件表示的单独处理内输入信息和从其输出信息，用于在处理之间移动信息，并且用于定义处理的运行顺序。该系统包括算法，该算法选择进程之间的图形方法(例如，根据图形的链接的图形路径可以使用TCP/IP或UNIX域套接字或使用用于在进程之间传送数据的共享存储器)。

[0020] 预处理模块106可以从不同不同形式的数据库系统的多种类型的系统接收数据。该数据可以被组织为记录，该记录具有用于相应的字段(也称为“属性”或“列”)的值，包括可能的空值。当首先从数据源读取数据时，预处理模块106通常从关于在那个数据源中的记录的某种初始格式信息开始。在一些情况下，该数据源的记录结构可能初始未知，并且可以取代在数据源的分析后被确定。关于记录的初始信息可以包括用于表示离散值的比特的数量、在记录内的字段的顺序和由比特表示的数据的类型(例如，字符串、有符号/无符号的整数)。。

[0021] 与在记录中使用类似，不同的数据类型(例如，字符串、整数、流点值)可以用于处理由执行环境104、开发环境118等执行的数据(例如，记录)。例如，各种类型的数据可以被定义和用于处理记录(例如，雇员记录、学生记录等)。为了处理这样的记录，单独的原始数据类型(例如、字符串、日期、日期时间、整数、小数等)可以一致地用于定义被称为数据类型

结构的数据类型的组织。例如，下面的记录将四个原始类型编组，以表示人的属性：

```
record
   string(",") surname;
   string(",") given_name;
   date("YYYY-MM-DD") date_of_birth;
   integer(1) gender; // 0 for male, 1 for female
   decimal(9) SSN;
end
```

[0023] 记录包括多个字段（例如，个人的姓和名连同他们的生日、性别和社会安全号码被表示），并且每一个字段由名称和数据类型构成。参见图2，这样的记录可以被编辑者定义。在这个示例中，用户界面200呈现了用于定义在从输入文件读取的数据的格式的编辑器。连同定义被读取的数据（例如，姓、生日、性别），定义了用于每一个字段的独立的数据类型（例如，字符串、日期、整数）和限制。虽然这个示例呈现了包括信息的5个不同条目的记录类型，但是可以扩展或收缩该记录类型。例如，可以将更多的数据补充、删除、与当前数据组合等。从上面的指令继续，可以嵌套记录类型，以例如表达商业实体和对应的雇员的列表：

```
/* Business entity */
record
    string(",") business_name;
    decimal(9) tax_payer_id;
    decimal(11) main_TEL_no;
    record
        record
        string(",") last; // last name
        string(",") first; // first name
    end name; // subrecord
    record
        date("YYYY-MM-DD") date_of_birth;
        integer(1) gender; // 0 for male, 1 for female
    end bio; // subrecord
    decimal(9) SSN;
  end[integer(4)] employees; // vector of employees
end
```

[0025] 通过建立和命名数据类型结构，等同字段等可以用于多个应用(例如，用于表示学生信息、多个公司的雇员的排序信息等)。参见图3，用户界面机300可以被实现来用于呈现包括嵌套的数据类型信息的数据类型结构。在该布置中，在上部(如括号302表示)定义与公司相关联的字段，并且在下部(如括号304表示)定义用于雇员信息的字段。可以想象，可以对于大量的应用定义数据类型结构，并且可以使用和重新使用许多类似字段(例如，以定义用于学生信息的记录、用于雇员信息的记录等)。

[0026] 参见图4，可以实现一种或多种技术以帮助开发者适当地定义字段，并且将该字段编组以形成数据类型结构以用在例如应用开发中。例如，通过与用户界面400交互(数据结构移动计算机单元，使用指示装置选择字段)，下拉菜单402可以出现和呈现可能(被开发者)选择来在应用中使用的命名的数据类型结构的列表。如在下拉菜单402中表示，当定义越来越多的命名数据类型结构时，所呈现的列表可能变得繁重地大，并且使得用户难以导航以识别感兴趣的命名的数据类型结构。例如，连同频繁选择的命名的数据类型结构，该列表可以随着时间而增长以包括对于特定应用独特的大量较不频繁使用的命名的数据类型结构。如在该图中所示，在菜单402中的条目的数量之多可以强烈地影响开发者的效率，并且可能甚至阻止用户界面400的使用。而且，仅被分配名称并且缺少任何其他信息(如在菜单402的条目中所示)的许多命名的数据类型结构可能冗余，并且未提供关于在其他命名的数据类型结构之间的关系(如果有)的信息。

[0027] 一种或多种技术和方法可以被实现来提供数据类型结构的更可管理的表示。在一个示例中，可以公共地定义在多个不同应用中使用的数据类型结构。从上面的示例，用于每一个学生的数据类型结构等同于每一个企业雇员的数据类型结构，并且可以被公共地定义为：

```
record
   record
      string(",") last; // last name
      string(",") first; // first name
   end name; // subrecord
```

[0028]

```
   record
      date("YYYY-MM-DD") date_of_birth;
      integer(1) gender; // 0 for male, 1 for female
   end bio; // subrecord
   decimal(9) SSN;
end
```

[0029] 通过命名该公共数据类型结构(例如，“person_t”)，命名的数据类型结构可以被调出，并且用在用于诸如收集学生和雇员应用的信息的类似目的的应用中。例如，命名的数据类型结构(“person_t”)可以被定义为：

```
type person_t =
record
   record
      string(",") last; // last name
      string(",") first; // first name
   end name; // subrecord
   record
      date("YYYY-MM-DD") date_of_birth;
      integer(1) gender; // 0 for male, 1 for female
   end bio; // subrecord
   decimal(9) SSN;
end;
```

[0030]

[0031] 一旦被定义，命名的数据类型结构(“person_t”)可以用于分别定义用于企业雇员和教室学生应用的数据类型结构。

```
/* Business entity */
record
   string(",") business_name;
   decimal(9)   tax_payer_id;
   decimal(11) main_TEL;
   person_t[integer(4)] employees; // here
end
```

[0032]

```
/* Classroom */
record
   string(",") home_room_instructor;
   decimal(2) grade;
   person_t[integer(4)] students; // and here
end;
```

[0033] 类似地，可以命名新的数据类型结构来用在多个应用中。例如，使用命名的数据类型结构“person_t”的命名的数据类型结构(题目为“business_t”)可以被定义来用于存储商业信息；

```
type business_t =
record
   string(",") business_name;
   decimal(9) tax_payer_id;
   decimal(11) main_TEL;
   person_t[integer(4)] employees; // vector of employees
end;
```

[0034]

[0035] 类似地，使用命名的数据类型结构“person_t”的命名的数据类型结构(命名为“class_t”)可以被定义来存储类别信息：

```
type class_t =
record
   string(",") home_room_instructor;
   decimal(2) grade;
   person_t[integer(4)] students; // vector of students
end;
```

[0036]

[0037] 可以通过使用公共的数据类型结构定义多个其他数据类型结构来提供各种优点。例如，不需要单独地调整该两个数据类型结构(以调整包括的数据类型结构)，可以再一次重新定义公共地使用的命名的数据类型结构。对应地，通过调整命名的数据类型结构，将在数据类型结构(和使用命名的数据类型结构的任何其他数据类型结构)两者中反映任何改变。由此，可以通过允许开发者调整在多个数据结构中使用的命名的数据类型结构的单个实例来改善效率。然而，开发者应当对于这样的传播的调整保持警惕(例如，当开发者仅感兴趣于在使用包括的数据类型结构的数据类型结构的不是全部中调整包括的数据类型结构时)。

[0038] 参见图5，一旦被定义，则可以实现用于存储命名的数据类型结构和检索该结构以使用的一种或多种技术或方法。例如，命名的数据类型结构可以以诸如数据操纵语言(DML)的语言被表达，并且被存储在文本文件(被称为DML文件)中。使用上面定义的命名的数据类型结构(例如，“person_t”)，题目为“project.dml”DML文件可以被存储在存储装置中，并且被检索以使用定义的数据类型结构。可以实现用于访问和使用在DML文件中存储的命名的数据类型结构的各种技术。例如，文件(例如，另一个DML文件)可以包括一个或多个指令(例如，“include”语句、“package”语句等)，该一个或多个指令允许一个或多个其他DML文件(和在该文件内定义的命名的数据类型结构)被访问和使用。例如，DML文件的内容可以包括下面的语言：

```
include "project.dml";

type sales_prospects_t =
record
   class_t[integer(4)] educational;
   business_t[integer(4)] commercial;
end;
```

[0040] 通过经由使用“include”语句访问DML文件(“project.dml”),在DMRL文件内定义的命名的数据类型结构(例如,“”business_t)可以被检索和用于定义在文件内定义的数据类型结构(例如,“sales_prospects_t”)。这样的能力减少了数据类型定义的冗余连同与数据类型结构(例如,具有相同的名称但是不同的类型的定义的数据类型结构)冲突的概率。

[0041] 如在图中所示,一旦被定义和存储,命名的数据类型结构可以被呈现来用于被开发者选择和使用。例如,用户界面500可以包括用于建立应用的编程属性(例如,全局变量502、函数504、用户定义的类型406等)。如在用户界面500中所呈现,三个命名的数据类型结构(例如,“person_t”、“business_t”和“class_t”)被包括在用户定义的类型506中,并且可以被选择来由开发者使用。然而,类似于在图3中呈现的下拉菜单402,当数据类型结构的列表对于不同应用增长时,在用户定义的类型506中包括的条目可能连同冗余条目变得猖獗而变得笨重。通过频繁使用和不太频繁使用的命名的数据类型结构的增长,可能出现用于识别可获得什么特定的数据类型结构的挑战。当识别先前定义的数据类型结构可能对于开发者变得困难并且过分耗时时,冗余也可能变为问题。在用户界面500的呈现提供了每一个命名的数据类型结构的列表的同时,提供了较少的另外的信息。例如,在数据类型结构之间的相关性的血统和如何可以将相关的数据类型编组等一般与用户界面500无关。而且,在呈现数据类型结构的名称的同时,除了可以从每一个数据类型结构的题目隐约地闪现的内容之外,不从该界面提供关于数据类型结构的内容的信息。

[0042] 参见图6,图示了图形表示600,其表示命名的数据类型结构和定义该结构的字段的集合。另外,图形表示600识别DML文件602(例如,“project.dml”),其内,定义和存储了数据类型结构。图形表示600的一部分提供了文件的选用的命名604(例如,DML文件),其被称为“封装”,其当在该文件之外被调用时提供了在文件中定义的属性的名称的前缀。在这个特定示例中,已经将封装名称留为空的。

[0043] 在这个示例中,图形表示600呈现了命名的数据类型结构“person_t”,其被定义为:

[0044]
```
type person_t =
record
    string(",") surname;
    string(",") given_name;
    date("YYYY-MM-DD") date_of_birth;
    integer(1) gender; // 0 for male, 1 for female
    decimal(9) SSN;
end;
```

[0045] 在这个示例中，图形表示600被定向为从观看者的左面向右面被读取，并且初始在观看者的左面呈现命名的数据类型结构名称606(例如，“person_t”)。从上面定义的结构，一系列矩形呈现被嵌套到数据类型结构“person_t”内的单独字段(例如，姓608、名610、生日612、性别614和社会安全号码616)。图形表示600传送命名的数据类型结构的分级布局和其内容。位于图形表示600的远左侧的名称606识别整体结果，而单独字段608、610、612、614、616位于更右面，并且指示它们驻留在数据类型结构的分级中的较低层中。从该呈现，向观看者提供了与命名的数据类型结构相关联的信息的容易读取的图形布局。虽然在该情况下使用矩形来呈现信息，但是其他形状和/或差形状的集合可以用于呈现。其他图形特征也可以被包含来用于帮助观看者有效地确定命名的数据类型结构信息。例如，可以实现不同的颜色以例如迅速地向观看者警告可能的问题(例如，在命名的数据类型结构内定义重复或冲突的字段)。在该情况下，统计图形用于呈现信息；然而，随着时间改变的图形(例如，动画、视频等)也可以用于例如迅速地抓住观看者的注意力。也可以使用其他图形表示，例如，不是使用左到右阅读方向，可以实现用于呈现一个或多个命名的数据类型结构的其他布局和方向。

[0046] 参见图7，图形表示700图示了DML文件(例如，命名的“project.dml”)，其内，定义了三个命名的数据类型结构。具体地说，连同先前定义的命名的数据类型结构“person_t”的图形表示600(在图6中所示)，在DML文件中图形地表示了两个另外的命名的数据类型结构(题目为“business_t”和“class_t”)：

[0047]
```
type business_t =
record
    string(",") business_name;
    decimal(9)    tax_payer_id;
    decimal(11) main_TEL;
    person_t[integer(4)] employees; // here
end
```

[0048] 以及

```
type class_t =
record
   string(",") home_room_instructor;
   decimal(2) grade;
      person_t[integer(4)] students; // and here
end;
```

[0049]

[0050]

[0051] 由对应的图形表示702、704图示的“business_t”命名的数据类型结构和“class_t”命名的数据类型结构两者包括由命名的数据类型结构“person_t”定义的字段(例如,在“business_t”命名的数据类型结构中包括的“雇员”字段和在“class_t”数据类型结构中包括的“学生”字段)。为了图形地图示由“business_t”和“class_t”数据类型结构对于“person_t”命名的数据类型结构的使用,对应的带箭头的线706、708示出在多对命名的数据类型结构之间的连接。从这些图形表示的关系,观看者(例如,开发者)可以较快地识别在命名的数据类型结构之间的关系,诸如在数据类型结构之间共享的信息、先前定义的命名的数据类型结构(诸如“person_t”)的使用的血统等。连同提供命名的数据类型结构的关系的布局,图形表示700可视地向观看者警告可能的调整问题。例如,如果对于“person_t”命名的数据类型结构作出了改变,如两个带箭头的线706、708所示,“business_t”和“class_t”命名的数据类型结构两者将被该改变影响(例如,“business_t”命名的数据类型结构的“employees”字段和“class_t”命名的数据类型结构的“students”字段将体验对于“person_t”命名的数据类型结构的任何改变)。类似于呈现在数据类型结构之间的关系,可以图形地呈现其他类型的关系,诸如在文件之间的关系。

[0052] 参见图8,将两个DML文件连同它们的关系图形地表示。将“project.dml”文件(在图7中所示)的图形表示700连同由DML文件定义的三个命名的数据类型结构(例如,“class_t”命名的数据类型结构702、“business_t”命名的数据类型结构704和“person_t”命名的数据类型结构600)呈现。另外,使用图形表示800来图示题目为“CRM.dml”的另一个DML文件。,在图形表示800内,表示命名的数据类型结构(题目为“sales_prosects_t”)。在这个示例中,通过“CRM.dml”限定了两个命名的数据类型结构(例如,命名的“educational(教育的)”和“commercial(商业的)”),并且该两个命名的数据类型结构的每一个通过由“project.dml”提供的命名的数据类型结构来被定义。具体地说,从“class_t”命名的数据类型结构定义“educational(教育的)”命名的数据类型结构,并且通过“business_t”命名的数据类型结构定义“commercial(商业的)”命名的数据类型结构。为了表示在“CRM.dml”文件和“roject.dml”文件之间的这两种关系,将两个带箭头的线802、804图示为链接文件的两个图像表示。类似于与文件链接的命名的数据类型结构的表示,可以通过在两个或更多文件之间的字段和命名的类型定义的链接等来形成类似的关系。例如,对于在“project.dml”文件中的“class_t”命名的数据类型结构702或“business_t”命名的数据类型结构704的定义的调整(例如,通过改变“person_t”命名的数据类型结构600)可以影响在链接的“CRM.dml”文件中的字段(例如,“educational(教育的)”和“commercial(商业的)”)和命名的数据类型结构(例如,“sales_prospect_t”命名的数据类型结构)。

[0053] 连同呈现在多个文件的字段和命名的数据类型结构之间的关系的图形表示，可以向观看者图形地表示在文件之间的关系。例如，可以表示文件级操作。在这个示例中，为了CRM.dml文件的字段(例如，“educational(教育的)”和“commercial(商业的)”)获得对于在“project.dml”文件中的命名的数据类型结构的访问，需要通过“CRM.dml”文件来识别“project.dml”文件。例如，可以在“CRM.dml”文件中输入“include”语句，以获得对于“project.dml”文件的访问。例如，为了图形地表示该识别，连同在图形表示800中列出其本身的文件名称806，也将一个或多个需要的文件(例如，“project.dml”文件)表示为所需的封装808。而且，在该示例中，带有箭头的虚线810图形地表示使用“include”语句来获得对于“project.dml”文件的内容的访问的、“CRM.dml”文件的文件级操作。

[0054] 参见图9a，可以使用各种类型的图形表示来图示在文件、字段和命名的数据类型结构之间的关系。例如，为了降低图形表示的可视复杂度，可以减少或去除各种数量的细节。在所示的示例中，从两个DML文件的图形表示900、902来去除单独的字段信息。通过去除该信息，通过简单地表示每一个命名的数据类型结构的名称(例如，“person_t”、“business_t”、“class_t”和“sales_prospect_t”)来将数据类型结构的图形表示904、906、908、910紧凑化。连同减少图形表示的可视业务的数量，对于诸如带箭头的线912、914和带箭头的线916、918的其他信息保留不动产，该带箭头的线912、914表示在文件内的命名的数据类型结构之间的关系，该带箭头的线916、918表示在不同文件中驻留的命名的数据类型结构之间的关系。类似地，保留的不动产可以帮助观看者迅速地识别其他关系，例如，用于使用带有箭头的虚线920识别其他文件，以指示使用“include”语句来提供文件访问。连同向观看者(例如，开发者)提供命名的数据类型结构和它们的关系的紧凑视图，可以通过该图形表示来提供其他功能。例如，可以通过使用该图形表示来更有效地执行操纵字段、命名的数据类型结构和相关的信息。

[0055] 参见图9b，图形表示可以被调整和操纵以有效地将文件数据类型结构、文件等有效地构造、重构等。例如，数据结构的表示可以被不同地编组以定义新的文件。如所示，已经去除了文件的图形表示(例如，DML文件表示900和902)，以允许开发者调整数据类型结构904、906、908、910的图形表示的编组。连同允许数据类型结构被重新组织以用于在相同或不同的一个或多个文件中存储，这样的操纵可以改善在文件之间的关系，并且减少不必要的“include”语句的出现。仅为了使用所示的示例来演示，可以将图形表示906(用于数据类型结构“business_t”)与图形表示910(用于数据类型结构“sales_prospect_t”)编组以用于更有效的操作。一旦被编组，则该图形表示可以启动用于存储新编组的数据类型结构的文件建立。连同使用这样的操作来用于编组和操纵数据类型结构，也可以执行文件级操作。例如，可以操纵文件的图形表示(例如，表示900、902)以用于将文件的内容(例如，数据类型结构)组合、去除、附加等。

[0056] 参见图10，图示了用户界面1000，其提供了编辑器1002，用于操纵字段、命名的数据类型结构和其他类型的编程属性的图形表示。编辑器1002包括窗口1004，其呈现字段、命名的数据类型结构和相关信息的图形表示(例如，用于表示接收类型关系的带箭头的线)。编辑器1002也包括调色板1006，其允许用户(例如，开发者)来从多个字段、命名的数据类型结构等选择以包括在在开发的应用中。例如，如在图中以粗体带箭头的线1008表示，指示装置可以用于选择命名的数据类型结构和将其插入(例如，拖放)到窗口1004内以用于项目开

发。类似地，可以将该选择和插入操作逆反，使得从窗口1004选择命名的数据类型结构(在被开发后)，并且将其插入调色板1006内。例如，可以在调色板1006中执行操作(例如，拖放操作)以将一个或多个字段、命名的数据类型结构等建立、编辑等。类似地，用户在窗口1004中启动操作(例如，选择、插入、删除、附加等)以调整字段、命名的数据类型结构等。也可以例如执行其他类型的操纵操作，例如，可以图形操纵在字段、命名的数据类型结构、文件(例如，DML文件)等之间的关系。

[0057] 参见图11，可以由用户(例如，开发者)图形地启动用于操纵字段、命名的数据类型结构、文件等和关系的操作。例如，可以操纵(例如，删除、增加、移动等)带箭头的线以调整在字段和命名的数据类型结构之间的关系。在该图示的示例中，删除两条线1100和1102(如分别位于通过用户的指示装置的每条线上的图形符号“x”表示)。基于因为删除该线导致的在“project.dml”文件和“CRM.dml”文件之间的断开的关系，“CRM.dml”不再需要对于“project.dml”文件的内容的访问。如此一来，从“CRM.dml”文件(如虚线方框1104表示)去除用于访问“project.dml”文件(例如，“include(包括) ‘project.dml’”)的、在“CRM.dml”文件内的指令。对应地，可以类似地从文件的图形表示去除表示在“CRM.dml”文件和“project.dml”文件之间的该关系的虚线1106。替代地，当建立或重新建立(例如，通过将该两个文件与线1100和1102连接)在文件之间的这样的关系时，可以向适当的文件(例如，“CRM.dml”)内插入或重新插入该指令(例如，“include(包括) ‘project.dml’”)，并且，可以再一次呈现图形表示(例如，虚线1106)以图示该关系。

[0058] 参见图12，当命名的字段、命名的数据类型结构、文件等的数量增大时，可以实现用于一种或多种技术，用于帮助用户(例如，开发者)在可以被选择来在应用开发期间使用的可能字段、命名的数据类型结构、文件等之间导航。例如，一个或多个图形表示可以呈现可选择的文件的列表，其包括可使用字段和命名的数据类型结构。在一些布置中，分层列表可以用于帮助用户导航文件、命名的数据类型结构等。如在该图中所示，在用户界面1202中包括窗格1200，其允许用户在不同封装(例如，XML处理数据类型、查找表数据类型、元编程数据类型、日期/上数据类型、元数据类型等)的列表之间导航。在一种布置中，一旦进行了选择，则诸如命名的数据类型结构和函数的在所选择的封装中定义的人为效应可以根据需要被显示在位于用户界面1202的右手侧上的图形表示中并且被操纵(例如，被选择、被导航、被拖放在窗格1200上)。

[0059] 连同将字段命名的数据类型结构、用于应用开发的文件组织、操纵等，可以类似地将其他类型的编程属性图形地表示以用于帮助开发者。例如，等一个或多个这样的编程属性被建立和存储在库中以用于以后的检索和再用时，由应用使用的函数、变量等可以类似地变得笨重。也可以实现其他技术，用于帮助开发者识别和选择适当的编程属性。例如，用于在文件之间逻辑地分布编程属性的算法(称为成簇算法)可以用于命名的数据类型结构图形表示组织字段、命名的数据类型结构、函数等。通过这样的组织技术，可以减少指令(例如，“include”语句)的指令连同命名的数据类型结构的重复使用。

[0060] 参见图13，流程图1300表示用于图形地表示诸如在应用开发中使用的命名的数据类型结构的编程属性的过程的操作。该操作通常被单个计算装置(例如，提供开发环境)执行；然而，可以通过多个计算装置来执行操作。连同在单个站点簇被执行，操作执行可以被分布在两个或更多位置之间。

[0061] 操作可以包括定义1302数据结构，其表示用于开发应用的一个或多个编程属性的分级。文件的图形表示可被去除以允许操纵数据结构的图形表示以定义一个或多个数据结构组，并且建立一个或多个新的文件以存储该一个或多个数据结构组。例如，子记录、记录、字段、命名的数据类型结构等的分级可以用于定义数据结构。该数据结构被定义以被包括在单个文件中(用于存储)，但是在一些布置中，该数据结构可以被包括在多个文件中(例如，用于存储和以后检索)。操作也包括产生1304可视图，该可视图包括数据结构的图形表示和用于存储该数据结构的文件的图形表示。该可视图也包括在数据结构和另一个数据结构之间的关系的图形表示和在用于存储该数据结构的文件和用于存储该另一个数据结构的文件之间的关系的图形表示。例如，如在图7中所示，在用于表示在文件“project.dml”中包括的命名的数据类型结构的可视图中图形表示两个命名的数据类型结构(例如，“business_t”和“class_t”)。也呈现第三数据类型结构(例如，“person_t”)，其包括被如在该可视图中表示的图形线706、808指示的其他命名的数据类型结构(例如，“business_t”和“class_t”)两者使用的字段的分级。连同图示每一个数据结构的内容，图形表示在数据结构之间的关系以例如允许观看者(例如，开发者)较快地确定字段、命名的数据类型结构等的血统和它们的关系。

[0062] 可以使用用于在计算机上执行的软件来实现如上所述的用于图形表示计算人为效应的手段。例如，该软件在一个或多个编程或可编程计算机系统(其可以是各种架构的，诸如分布的、客户机/服务器或网格)上执行的一个或多个计算机程序中形成过程，每个计算机系统包括至少一个处理器、至少一个数据存储系统(包括一小时和非易失性存储器和/或存储元件)、至少一个输入装置或端口和至少一个输出装置或端口。该软件可以形成例如较大程序的一个或多个模块，其提供与数据流图形的设计和配置相关的其他服务。该图形的节点和元素可以被实现为在计算机可读介质中存储的数据结构或符合在数据存储库中存储的数据模型的其他组织的数据。

[0063] 该软件可以被设置在由一般或专用可编程计算机可读的诸如CD-ROM的存储介质上，或者通过网络的通信介质被传递(在传播信号中被编码)到计算机的存储介质，其中，它被执行。可以在专用计算机上或使用诸如协处理器的专用硬件执行所有的功能。可以以分布的方式来实现该软件，其中，通过不同的计算机来执行由软件指定的计算的不同部分。每一个这样的计算机程序优选地被存储在或被下载到由通用或专用可编程计算机可读的存储介质或装置(例如，固态存储器或介质或磁或光介质)，以用于当该存储介质或装置被计算机系统读取以执行在此所述的过程时配置和操作该计算机。本发明的系统也可以被看作被实现为被配置计算机程序的计算机可读存储介质，其中，如此配置的存储介质使得计算机系统以特定和预定义方式运行，以执行在此所述的功能。

[0064] 已经描述了本发明的多个实施例。尽管如此，可以明白，可以在不偏离本发明的精神和范围的情况下作出各种修改。例如，如上所述的步骤的一些可以是独立于顺序的，并且因此可以以与所述的顺序不同的顺序被执行。

[0065] 应当明白，上述的说明意图说明而不是限由所附权利要求的范围限定的本发明的范围。例如，可以以不同的顺序执行如上所述的多个功能步骤，而不实质上影响整体处理。其他实施例在所附的权利要求的范围内。

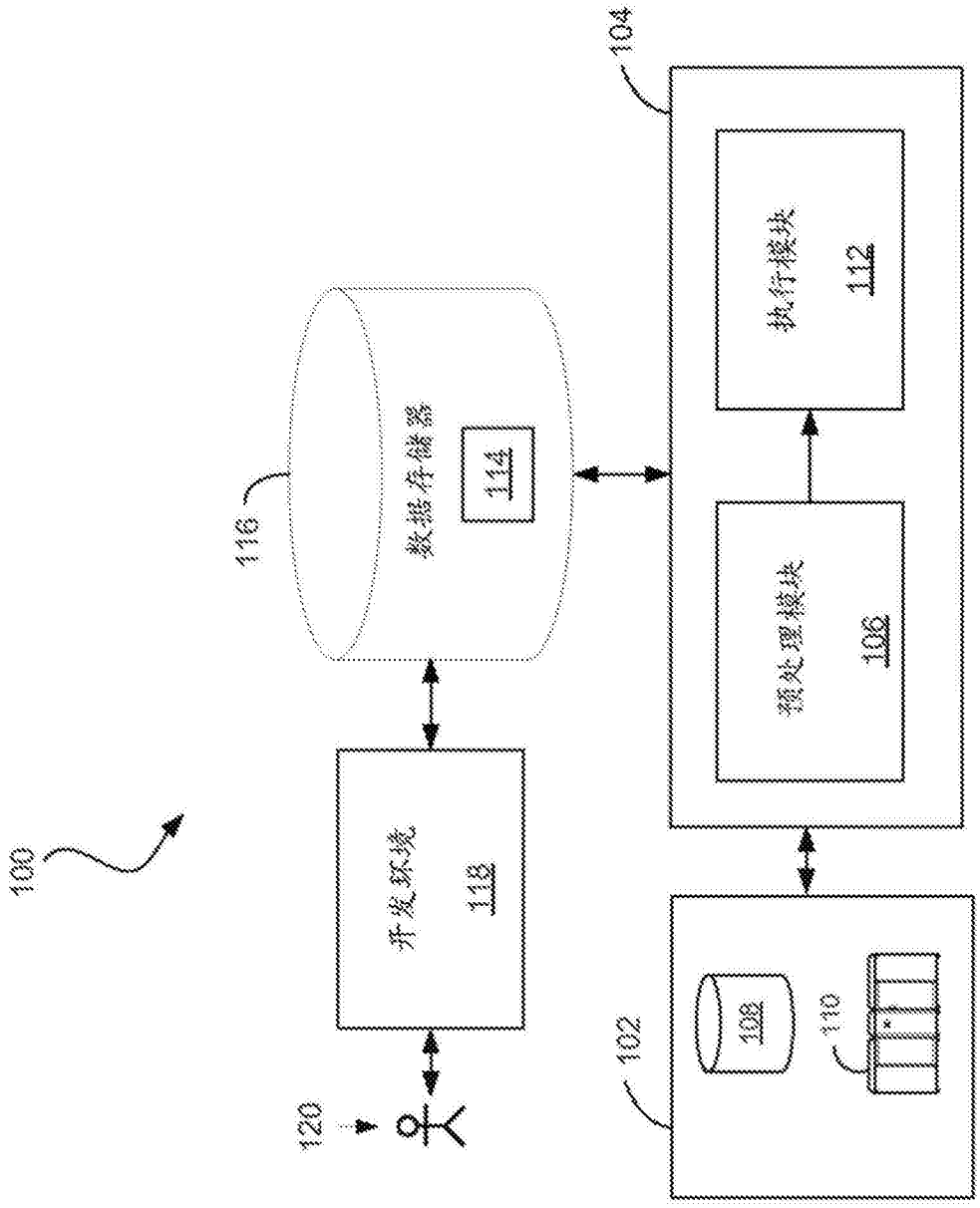

100
104
执行模块
112
预处理模块
106
116
数据存储器
114
开发环境
118
120
102
108
110

图1

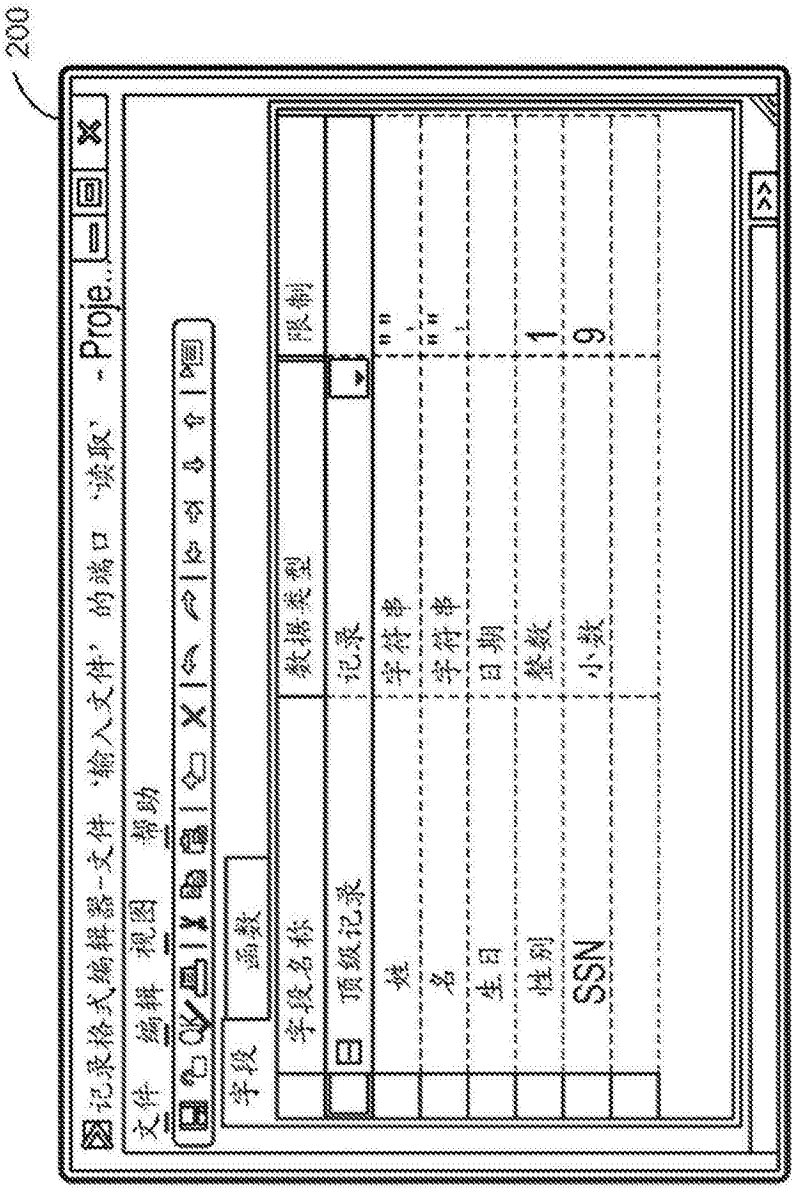

200
记录格式编辑器-文件 '输入文件' 的端口 '读取' - Proje...
文件 编辑 视图 帮助
字段 函数
字段名称
数据类型
限制
顶级记录
记录
姓
字符串
名
字符串
生日
日期
性别
整数
1
SSN
小数
9

图2

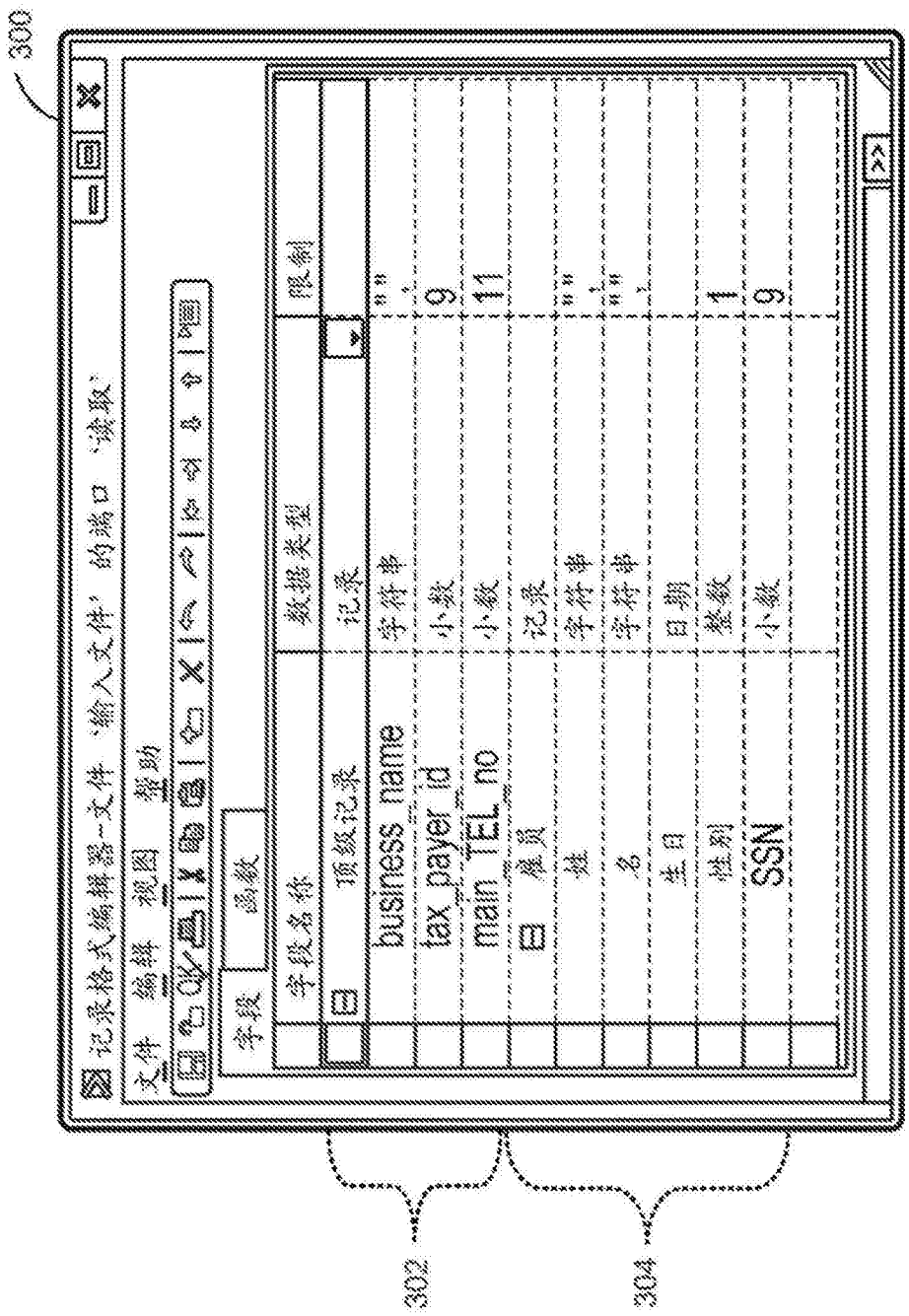

300
记录格式编辑器-文件'输入文件'的端口'读取'
文件 编辑 视图 帮助
字段
函数
字段名称
数据类型
限制
顶级记录
记录
business_name
字符串
""
tax_payer_id
小数
9
main_TEL_no
小数
11
雇员
记录
""
姓
字符串
""
名
字符串
""
生日
日期
1
性别
整数
1
SSN
小数
9
302
304

图3

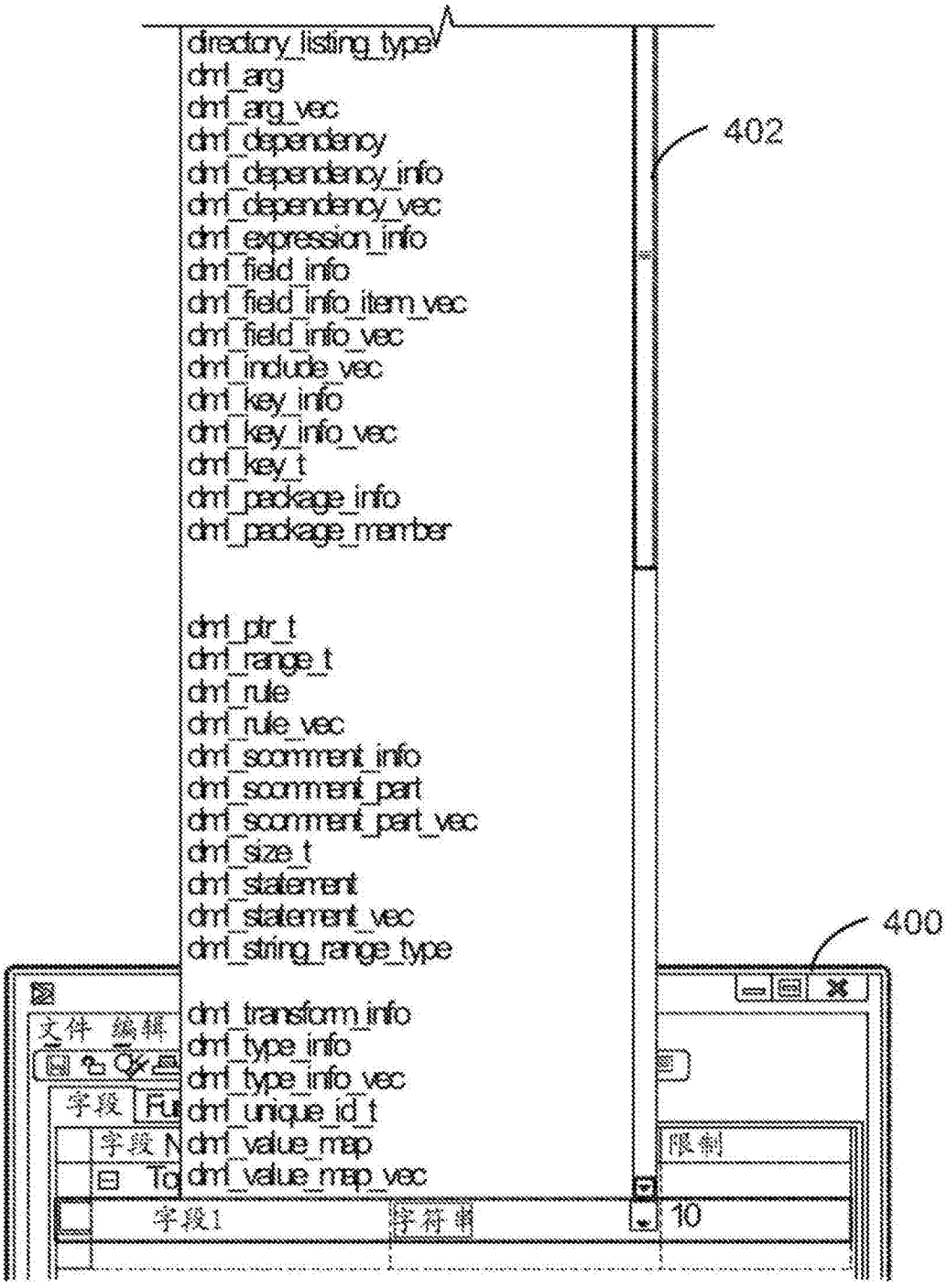

directory_listing_type
dmf_arg
dmf_arg_vec
dmf_dependency
dmf_dependency_info
dmf_dependency_vec
dmf_expression_info
dmf_field_info
dmf_field_info_item_vec
dmf_field_info_vec
dmf_include_vec
dmf_key_info
dmf_key_info_vec
dmf_key_t
dmf_package_info
dmf_package_member
dmf_ptr_t
dmf_range_t
dmf_rule
dmf_rule_vec
dmf_scomment_info
dmf_scomment_part
dmf_scomment_part_vec
dmf_size_t
dmf_statement
dmf_statement_vec
dmf_string_range_type
dmf_transform_info
dmf_type_info
dmf_type_info_vec
dmf_unique_id_t
dmf_value_map
dmf_value_map_vec
402
400
文件 编辑
字段
字段1
限制
10

图4

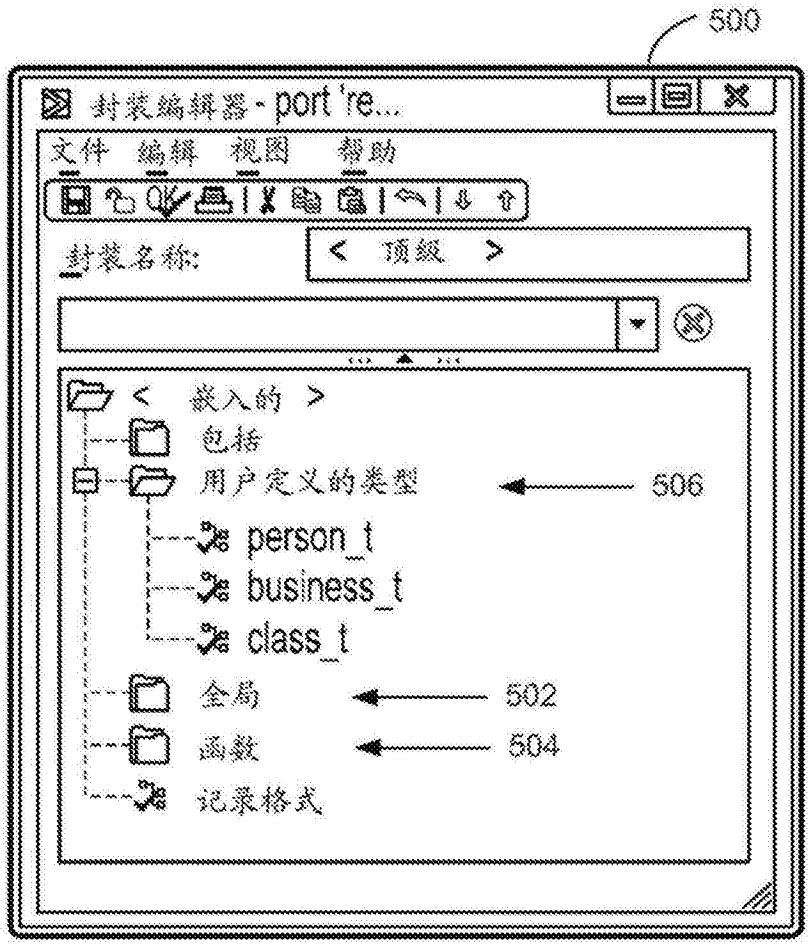

500
封装编辑器 - port 're...
文件 编辑 视图 帮助
封装名称:
< 顶级 >
< 嵌入的 >
包括
用户定义的类型
506
person_t
business_t
class_t
全局
502
函数
504
记录格式

图5

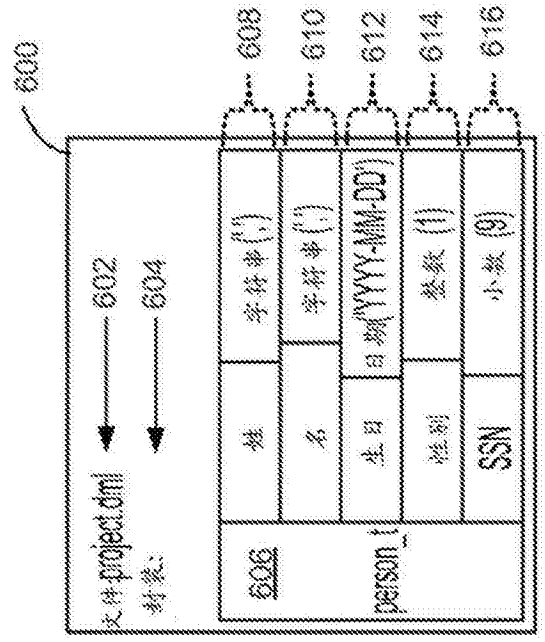

600
608
610
612
614
616
602
604
person_t
606

图6

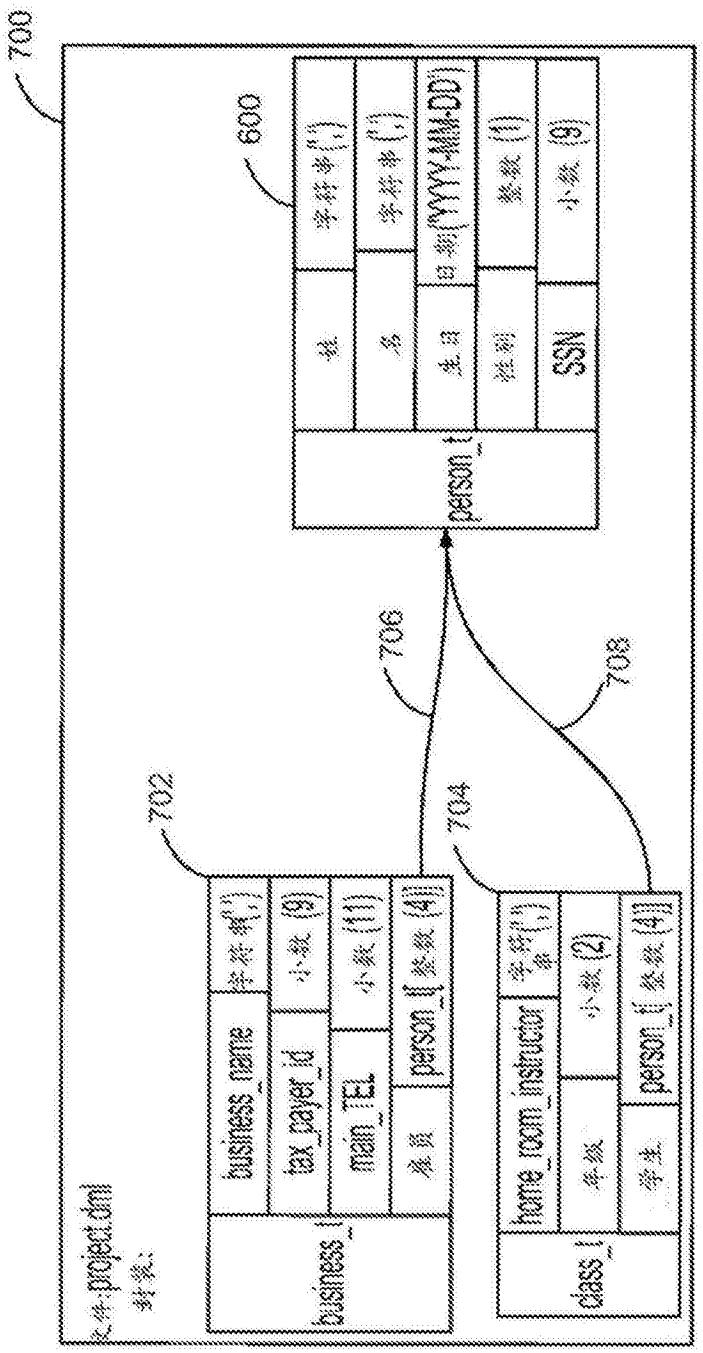

700
600
person_t
SSN
702
704
706
708
business_t
tax_payer_id
main_TEL
person_t[ ]
class_t
home_room_instructor
person_t[ ]

图7

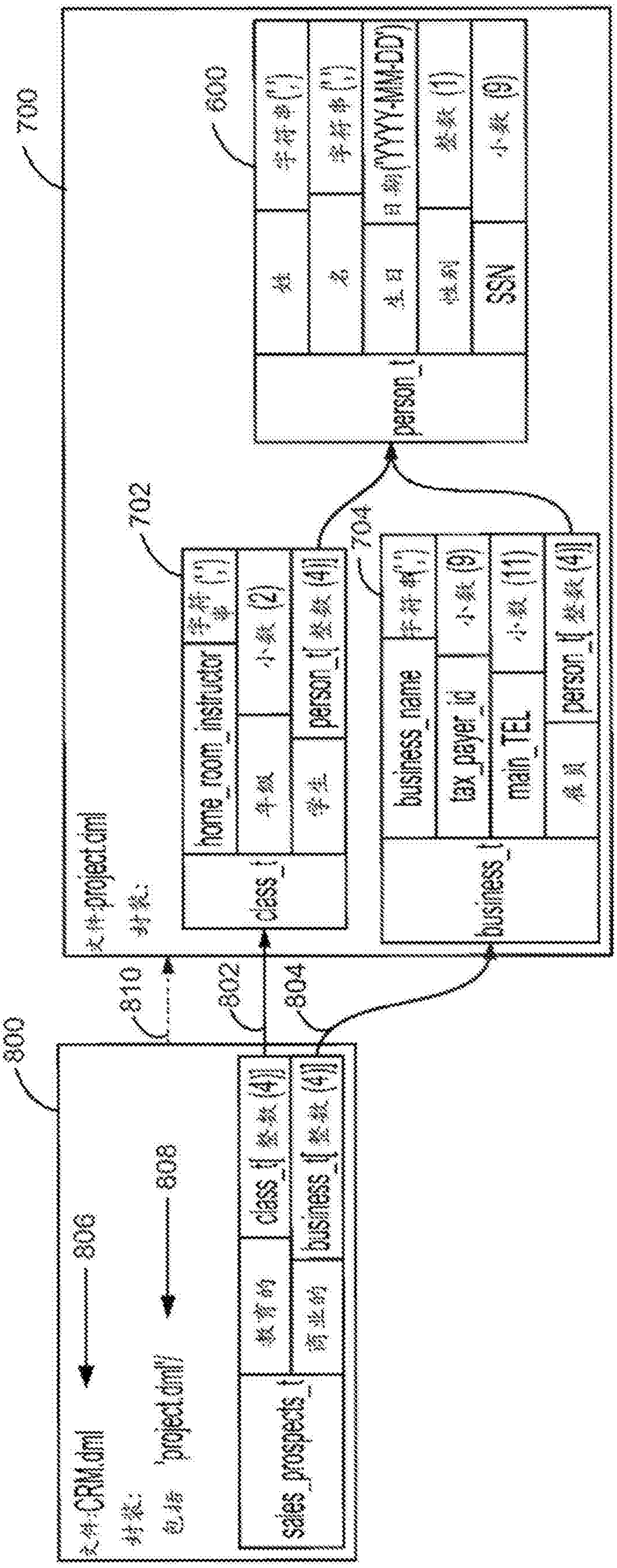

700
600
702
704
文件:project.dml
封装:
person_t
姓
字符串(',')
名
字符串(',')
生日
日期('YYYY-MM-DD')
性别
整数(1)
SSN
小数(9)
class_t
home_room_instructor
字符串(',')
年级
小数(2)
学生
person_t[整数(4)]
business_t
business_name
字符串(',')
tax_payer_id
小数(9)
main_TEL
小数(11)
雇员
person_t[整数(4)]
810
802
804
800
文件:CRM.dml
封装:
包括 'project.dml'
806
808
sales_prospects_t
教育的
class_t[整数(4)]
商业的
business_t[整数(4)]

图8

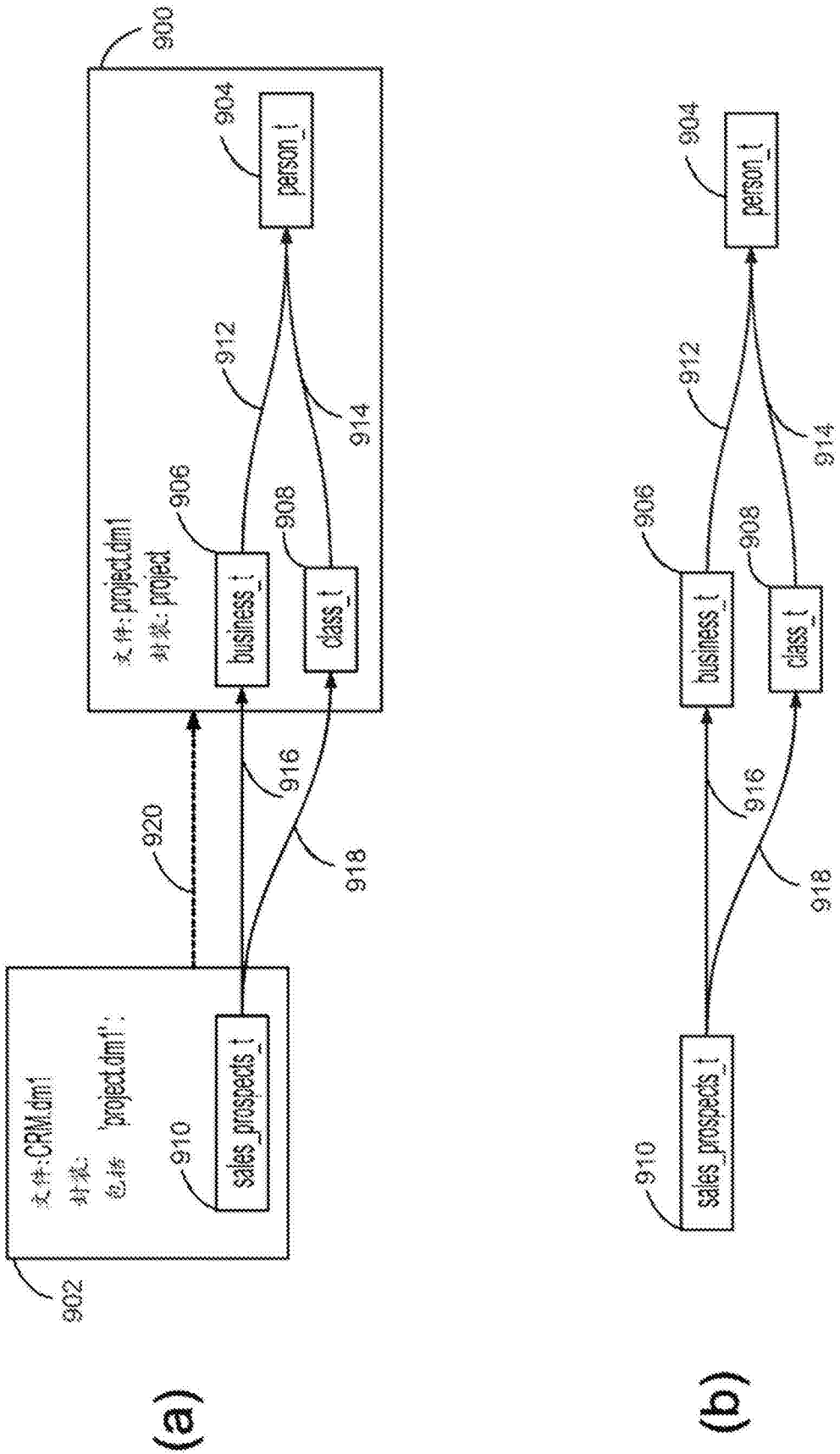

(a)
902
文件:CRM.dm1
对象:
包括 'project.dm1';
910
sales_prospects_t
920
900
文件:project.dm1
对象: project
906
business_t
908
class_t
904
person_t
912
914
916
918
(b)
910
sales_prospects_t
906
business_t
908
class_t
904
person_t
912
914
916
918

图9

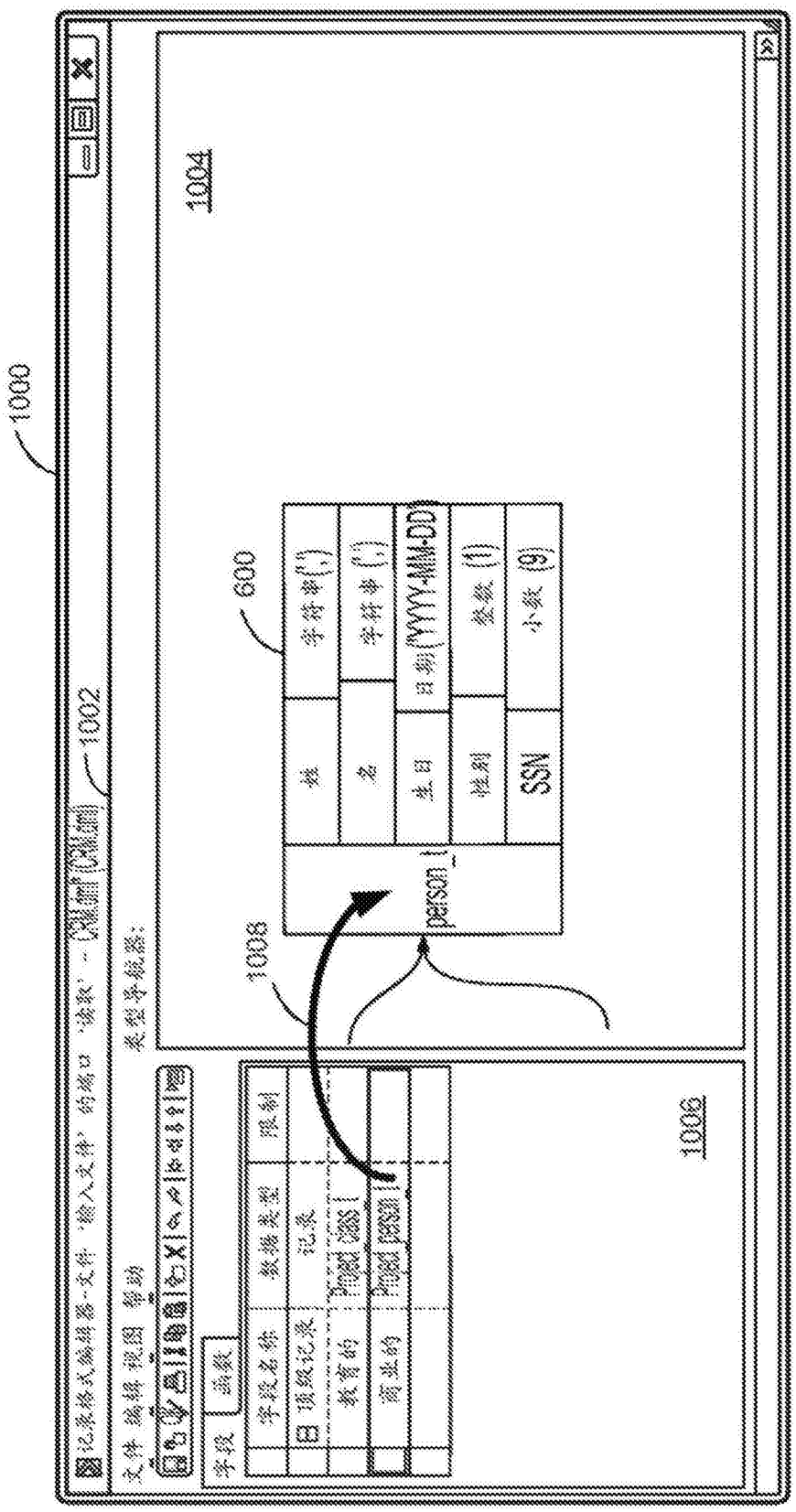

1000
1002
1004
1006
1008
600
person_1
姓
名
生日
地址
SSN
字符串(*)
字符串(*)
日期(YYYY-MM-DD)
整数 (I)
小数 (9)

图10

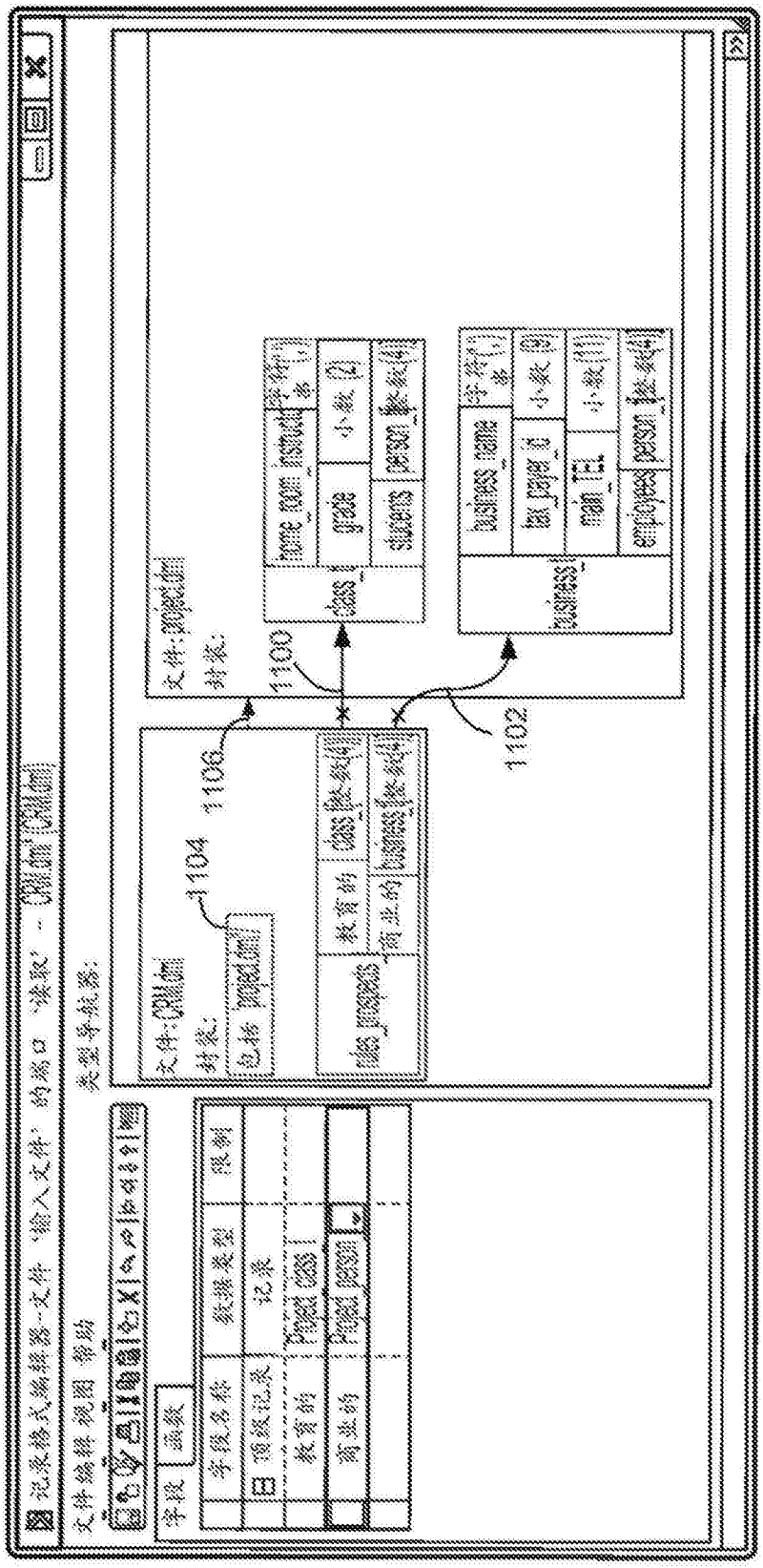


图11

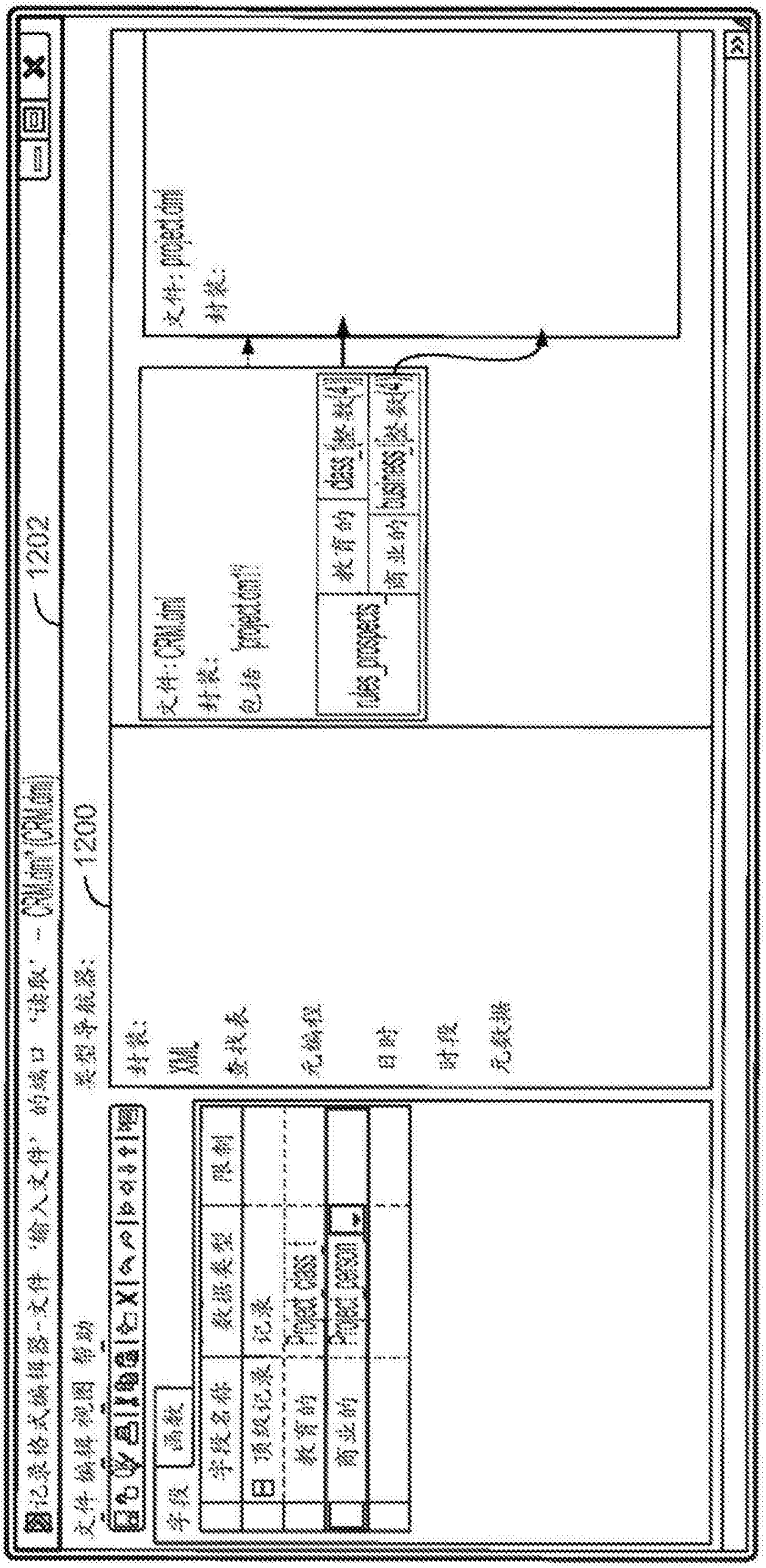

1200
1202
文件 编辑 视图 帮助
类型导航器:
字段
函数
字段名称
数据类型
限制
顶级记录
记录
教育的
商业的
文件:project.dml
封装:

图12

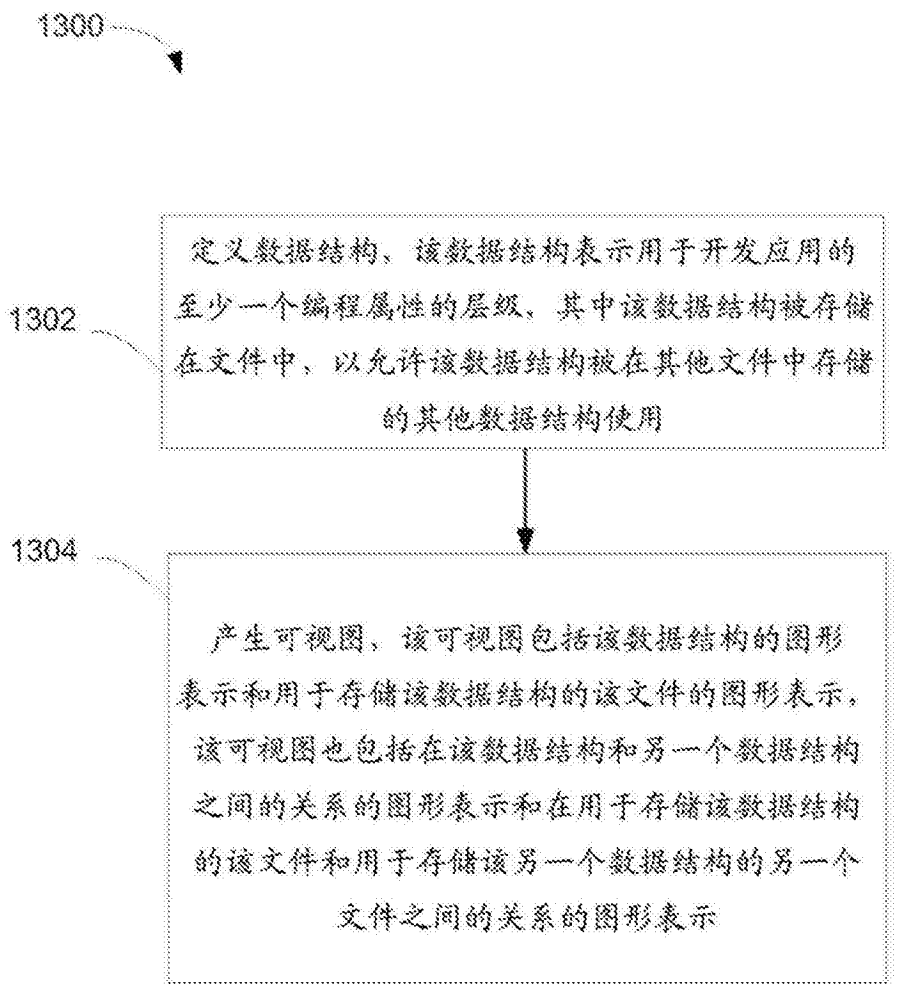

1300
1302
定义数据结构，该数据结构表示用于开发应用的至少一个编程属性的层级，其中该数据结构被存储在文件中，以允许该数据结构被在其他文件中存储的其他数据结构使用
1304
产生可视图，该可视图包括该数据结构的图形表示和用于存储该数据结构的该文件的图形表示，该可视图也包括在该数据结构和另一个数据结构之间的关系的图形表示和在用于存储该数据结构的该文件和用于存储该另一个数据结构的另一个文件之间的关系的图形表示

图13