[19] 中华人民共和国国家知识产权局

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

[12] 发明专利说明书

专利号 ZL 200510082378.1

[45] 授权公告日 2009 年 9 月 16 日

[11] 授权公告号 CN 100541489C

[22] 申请日 2005.6.15
[21] 申请号 200510082378.1
[30] 优先权
[32] 2004.7.15 [33] US [31] 10/891,609
[73] 专利权人 微软公司
地址 美国华盛顿州
[72] 发明人 J·K·豪威 J·R·本哈德特
T·T·李
[56] 参考文献
CN 1477558A 2004.2.25
US 5724575A 1998.3.3
US 6009428A 1999.12.28
US 6748388B1 2004.6.8
审查员 沈乐平

[74] 专利代理机构 上海专利商标事务所有限公司
代理人 李 玲

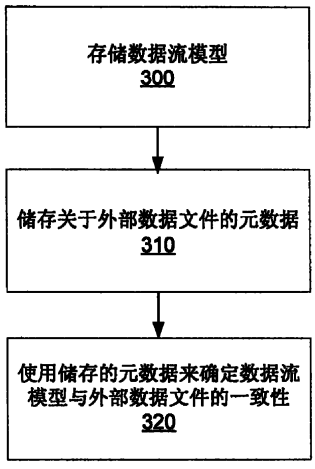
权利要求书 5 页 说明书 11 页 附图 4 页

[54] 发明名称

外部元数据处理

[57] 摘要

数据流的操作通过提供数据流与它所工作的外部数据文件的一致性的确认来改进，即使在面对那些外部数据文件的改变的情况下。 储存关于外部数据文件的元数据。 由于外部数据文件可能改变，因此当确定外部数据文件是否仍与数据流兼容时使用该元数据。 在执行这一确认时，元数据跟踪外部数据文件中出现了什么改变，并允许更灵活地向用户呈现选项或自动修补数据流以与外部数据文件的改变相对应。 当外部数据文件不可用时，可使用储存的元数据来将数据流确认至关于外部数据文件的最后信息。 可将外部元数据与数据流或外部数据文件重新同步，以提供更新的外部元数据供以后的确认。



1．一种使用至少一个外部数据文件准备关于数据流的数据流信息的方法，每一所述外部数据文件包括信息列，所述方法包括：

储存一数据流模型，所述数据流模型包括使用数据，所述使用数据对所述至少一个外部数据文件的每一个描述来自所述外部数据文件的至少一个列的使用；

储存元数据，所述元数据包括描述所述至少一个外部数据文件的每一个的至少一个列的数据；以及

使用所述储存的元数据来确定所述数据流模型是否与所述外部数据文件相一致。

2．如权利要求 1 所述的方法，其特征在于，所述描述至少一个列的数据包括涉及所述至少一个列的类型数据。

3．如权利要求 1 所述的方法，其特征在于，使用所述储存的元数据来确定所述数据流模型是否与所述外部数据文件相一致包括：

对于每一外部数据文件，验证所述储存的元数据与所述外部数据文件的当前状态相一致。

4．如权利要求 3 所述的方法，其特征在于，验证所述储存的元数据与所述外部数据文件的当前状态相一致包括：

确定所述储存的元数据中描述的每一列是否存在于所述外部数据文件中；以及

确定所述外部数据文件中的每一列是否在所述储存的元数据中描述。

5．如权利要求 4 所述的方法，其特征在于，所述描述至少一个列的数据包括描述储存在列中的数据的类型的类型数据，并且其中，验证所述储存的元数据与所述外部数据文件的当前状态相一致还包括：

确定所述储存的元数据中描述的每一列的所述类型数据是否与所述外部数据文件中的所述列的类型数据相一致。

6．如权利要求 3 所述的方法，其特征在于，使用所述储存的元数据来确定所述数据流模型和所述外部数据文件是否相一致还包括：

提供一指示所述储存的元数据是否与所述外部数据文件的所述当前状态相一致的确认状态。

7．如权利要求3所述的方法，其特征在于，使用所述储存的元数据来确定所述数据流模型与所述外部数据文件是否相一致还包括：

如果所述储存的元数据与所述外部数据文件的所述当前状态不一致，则解决所述不一致性。

8．如权利要求7所述的方法，其特征在于，所述不一致性的所述解决包括：

向用户查询以解决所述不一致性。

9．如权利要求7所述的方法，其特征在于，所述不一致性的所述解决包括：

使用所述数据流的接口来解决所述不一致性。

10．如权利要求1所述的方法，其特征在于，还包括：

重新同步所述储存的元数据以使其包括反映每一所述外部数据文件中储存的当前数据的数据。

11．如权利要求1所述的方法，其特征在于，还包括：

通过确定给定每一所述外部数据文件的当前状态，所述使用数据中描述的所述外部数据文件的至少一个列的每一所述使用是否可能，来确定所述数据流模型与所述外部数据文件是否相一致。

12．如权利要求11所述的方法，其特征在于，通过确定给定每一所述外部数据文件的当前状态，所述使用数据中描述的所述外部数据文件的至少一个列的每一所述使用是否可能，来确定所述数据流模型与所述外部数据文件是否相一致包括：

提供一指示所述数据流模型是否与所述外部数据文件的所述当前状态相一致的确认状态。

13．如权利要求11所述的方法，其特征在于，通过确定给定每一所述外部数据文件的当前状态，所述使用数据中描述的所述外部数据集文件的至少一个列的每一所述使用是否可能，来确定所述数据流模型与所述外部数据文件是否相一致包括：

如果所述储存的元数据与所述外部数据文件的所述当前状态不一致，则解决所述不一致性。

14．如权利要求13所述的方法，其特征在于，所述不一致性的所述解决包括：

查询用户以解决所述不一致性。

15．如权利要求13所述的方法，其特征在于，所述不一致性的所述解决包括：

使用所述数据流的接口来解决所述不一致性。

16．如权利要求11所述的方法，还包括：

重新同步所述数据流以反映所述外部数据文件的所述当前状态。

17．如权利要求 1 所述的方法，其特征在于，所述使用所述储存的元数据来确定所述数据流模型是否与所述外部数据文件相一致包括：

验证所述储存的元数据与所述数据流相一致。

18．如权利要求 17 所述的方法，其特征在于，所述验证所述储存的元数据与所述数据流相一致包括：

确定所述使用数据中描述的所述外部数据文件的至少一个列的每一所述使用是否与所述储存的元数据相一致。

19．如权利要求 18 所述的方法，其特征在于，还包括：

如果所述使用数据中描述的所述外部数据文件的至少一个列的每一所述使用与所述储存的元数据不一致，则解决所述不一致性。

20．一种评估数据流模型的方法，所述数据流模型包括关于数据流的信息，所述信息包括使用数据，所述使用数据对至少一个外部数据文件描述来自所述外部数据文件的至少一个列的使用，所述方法包括：

检索元数据，所述元数据包括描述所述至少一个外部数据文件的每一个的至少一个列的数据；以及

使用所述元数据来确定所述数据流模型与所述元数据是否相一致。

21．如权利要求 20 所述的方法，其特征在于，使用所述元数据来确定所述数据流模型与所述元数据是否相一致包括：

提供一指示所述储存的元数据是否与所述元数据相一致的确认状态。

22．如权利要求 20 所述的方法，其特征在于，使用所述元数据来确定所述数据流模型与所述元数据是否相一致还包括：

如果所述储存的元数据与所述外部数据文件的所述当前状态不一致，则解决所述不一致性。

23．如权利要求 22 所述的方法，其特征在于，所述不一致性的所述解决包括：

查询所述用户以解决所述不一致性。

24．如权利要求 22 所述的方法，其特征在于，所述不一致性的所述解决包括：

使用所述数据流的接口来解决所述不一致性。

25．如权利要求 20 所述的方法，其特征在于，还包括：

重新同步所述储存的元数据以使其包括反映储存在每一所述外部数据文件中的当前数据的数据。

26．一种用于使用至少一个外部数据文件准备关于数据流的数据流信息的数据流建模系统，每一所述外部数据文件包括信息列，所述数据流建模系统包括：

储存数据流模型的数据流模型存储，所述数据流模型包括使用数据，所述使用数据对所述至少一个外部数据文件的每一个描述来自所述外部数据文件的至少一个列的使用；

储存元数据的元数据存储，所述元数据包括描述所述至少一个外部数据文件的每一个的至少一个列的数据；以及

一致性核查器，它使用所述储存的元数据来确定所述数据流模型与所述外部数据文件是否相一致。

27．如权利要求26所述的数据流建模系统，其特征在于，所述描述至少一个列的数据包括涉及所述至少一个列的类型数据。

28．如权利要求26所述的数据流建模系统，其特征在于，所述一致性核查器对每一外部数据文件验证所述储存的元数据是否与所述外部数据文件的当前状态相一致。

29．如权利要求28所述的数据流建模系统，其特征在于，所述一致性核查器包括：

确认状态确定器，用于提供一指示所述储存的元数据是否与所述外部数据文件的所述当前状态相一致的确认状态。

30．如权利要求26所述的数据流建模系统，其特征在于，还包括：

元数据重新同步器，它修改所述储存的元数据以使其包括反映储存在每一所述外部数据文件中的当前数据的数据。

31．如权利要求26所述的数据流建模系统，其特征在于，所述一致性核查器包括：

一致性验证器，用于验证所述储存的元数据与所述数据流相一致。

32．如权利要求31所述的数据流建模系统，其特征在于，所述一致性验证器确定所述使用数据中描述的所述外部数据文件的至少一个列的每一所述使用是否与所述储存的元数据相一致。

33．如权利要求31所述的数据流建模系统，其特征在于，还包括：

解决器，用于如果所述使用数据中描述的所述外部数据文件的至少一个列的每一所述使用与所述储存的元数据不一致，则解决所述不一致性。

34．一种用于评估数据流模型的数据流建模系统，所述数据流模型包括关于

数据流的信息，所述信息包括使用数据，所述使用数据对至少一个外部数据文件的每一个描述来自所述外部数据文件的至少一个列的使用，所述数据流建模系统包括：

储存元数据的元数据存储，所述元数据包括描述所述至少一个外部数据文件的每一个的至少一个列的数据；以及

一致性核查器，它使用所述元数据来确定所述数据流模型与所述元数据是否相一致。

35．如权利要求 34 所述的数据流建模系统，其特征在于，所述一致性核查器包括：

确认状态指示器，它指示所述储存的元数据是否与所述元数据相一致。

36．如权利要求 33 所述的数据流建模系统，其特征在于，还包括：

不一致性解决器，用于如果所述储存的元数据与所述外部数据文件的所述当前状态不一致，则解决所述不一致性。

37．如权利要求 34 所述的数据流建模系统，其特征在于，还包括：

重新同步器，用于重新同步所述储存的元数据以包括反映储存在每一所述外部数据文件中的当前数据的数据。

外部元数据处理

技术领域

本发明一般涉及信息技术领域，尤其涉及关于数据流的元数据的创建和更新，以元数据的使用以允许确认数据流。

背景技术

数据的大集合可以用复杂的方式来使用。例如，数据集合，如文件、数据库和其它数据存储手段可以被打开、查询或用作长活动串的一部分，其中对数据出现不同的变换活动，并然后储存所得的数据。

例如，图 1 所示是具有两个数据输入和两个数据输出的数据流。如图 1 所示，文件 A 和数据库表 A 都被打开。如图 1 所示，从文件 A 1000 读取数据，并且从数据库表 A 1010 读取数据。在这两个数据集上执行并操作 1020。检查并操作的结果，以确定（框 1030），对数据的并集中的每一记录，储存在相关联的年龄字段中的值是否小于 50。对于其中相关联的年龄字段小于 50 的记录，将这些记录按性别聚集（1040），并储存在数据库表 B 中（1050）。对于其中相关联的年龄字段小于 50 的记录，将这些记录储存在文件 B 中（1060）。

为允许使用数据的大的、复杂的集合，开发了 ETL（提取变换加载）工具。这些工具提供了使用数据集合执行操作的自动化方法。ETL 工具自动化了提取数据的任务－从数据源取出数据；变换数据的任务－使用提取的数据；以及加载数据的任务－储存变换的结果以供以后使用。例如，行动中所示的行动由 ETL 工具执行。

为允许对这一 ETL 功能的简易使用并扩展可用的功能，开发了允许使用文件或其它数据集合的过程的可视设计的设计工具。一种这样的设计工具被称为数据变换服务（DTS），它可从微软公司获得。DTS 允许用户可视地设计过程，可通过这些过程使用文件、数据库或其它数据集合中的数据。由 DTS 设计的过程中的操作可包括但不限于通过标准 ETL 工具可得的那些操作。例如，DTS 设计的数据流可允许用户指定某些文件被删除、获取其它文件（例如，通过文件传输协议（FTP）

从指定的源获取）、以及然后在如此获得的每一文件上执行特定的 ETL 过程。

由 ETL 工具或诸如 DTS 等设计工具设计的数据库在其使用前被设计。这可导致当使用数据流时的岐义。例如，一数据流被设计成打开数据源，并对数据源中的每一记录读取特定列 A 和列 C 中的信息。然而，在运行时，在打开数据源之后，数据源可能对每一记录包含列 A、列 B、列 C 和列 D 中的信息。

数据流的设计者可能知道列 B 将包括在数据源中。如果是，则可作出不从列 B 读取信息的设计选择，以最小化如此完成的时间和其它计算成本。，由此，在运行时询问用户列 B 是否应当被包括在内会导致不必要的混淆和延迟。

然而，数据流的设计者可能不知道列 D 被包括在数据源中，并且数据流的用户将发现列 D 以包括在数据流中是有用的。由此，在运行时询问用户列 D 是否应当包括将是有用的。

然而，没有方法在数据被有意地不包括在数据源中的情况，以及数据源改变的情况之间进行区分。由此，要么在运行时向用户发出不必要的问题，要么丢失有用的数据。

另外，可以对数据集合作出改变。例如，列 A 的类型可以已从所期望的改变。这可以与在数据流中为列 A 所设计的操作兼容或不兼容。数据类型的某些改变可允许操作成功地继续进行，但带有不期望的结果。然而，没有方法来说出改变是预期的还是非预期的。再一次，要么向用户咨询数据类型的不兼容性，即使是在改变为预期的情况下，要么不向用户咨询，这允许问题继续发展。

由此，需要一种克服本领域中这些缺陷的系统和方法。本发明着眼于上述需求并用此处所阐明的附加优点解决了这些需求。

发明内容

本发明允许储存关于外部数据文件（或数据集合）的外部元数据，以供数据流使用。由于这一信息的存储，稍后可检查数据集合的内容并使用外部元数据，以确定对数据集合是否发生了任何改变，以及如果是则可采取什么行动。检查描述数据流的数据流信息以及在数据流中使用的关于数据集合的所储存的外部元数据，以确定信息是否一致。如果不是，则可向用户咨询，或可发生自动的手段（如类型转换）。

当数据集合不可用时，仍可使用外部元数据，以确定数据流是否与外部元数据中反映的数据集合兼容。可改变数据流以确保兼容性。

外部元数据的重新同步可更新外部元数据，以确保得自外部元数据存储的连续益处。

本发明的其它特征将在下文描述。

附图说明

当结合附图阅读时，可以更好地理解以上概述以及以下较佳实施例的详细描述。为说明本发明的目的，附图中示出了本发明的示例性构造；然而，本发明不限于所解释的特定方法和手段。附图中：

图 1 是数据流的图示；

图 2 是其中可实现本发明的各方面的示例性计算环境的框图；

图 3 是依照本发明的一个实施例，使用至少一个外部数据文件准备关于数据流的数据流信息的方法的流程图；以及

图 4 是依照本发明的一个实施例评估数据流模型的方法的流程图。

具体实施方式

示例性计算环境

图 2 示出了可在其中实现本发明的各方面的示例性计算环境。计算系统环境 100 仅为合适的计算环境的一个示例，并非暗示对本发明的使用范围或功能的任何局限。也不应将计算环境 100 解释为对示例性操作环境 100 中示出的任一组件或其组合具有任何依赖或需求。

本发明可以使用众多其它通用或专用计算系统环境或配置来操作。适合使用本发明的众所周知的计算系统、环境和/或配置包括但不限于：个人计算机、服务器计算机、手持式或膝上设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费者电子设备、网络 PC、小型机、大型机、嵌入式系统、包括任一上述系统或设备的分布式计算环境等等。

本发明可以在诸如由计算机执行的程序模块等计算机可执行指令的一般上下文环境中描述。一般而言，程序模块包括例程、程序、对象、组件、数据结构等等，执行特定的任务或实现特定的抽象数据类型。本发明也可以在分布式计算环境中实践，其中，任务由通过通信网络或其它数据传输介质连接的远程处理设备来执行。在分布式计算环境中，程序模块可以位于包括存储器存储设备的本地和远程计算机存储介质中。

参考图 2，用于实现本发明的示例性系统包括计算机 110 形式的通用计算装置。计算机 110 的组件可包括但不限于，处理单元 120、系统存储器 130 以及将包括系统存储器的各类系统组件耦合至处理单元 120 的系统总线 121。处理单元 120 可表示诸如多线程处理器上支持的多个逻辑处理单元。系统总线 121 可以是若干种总线结构类型的任一种，包括存储器总线或存储器控制器、外围总线以及使用各类总线体系结构的局部总线。作为示例而非局限，这类体系结构包括工业标准体系结构（ISA）总线、微通道体系结构（MCA）总线、增强型 ISA（EISA）总线、视频电子技术标准协会（VESA）局部总线以及外围部件互连（PCI）总线（也称为 Mezzanine 总线）。系统总线 121 也可被实现为点对点链接、交换光纤等其它通信设备。

计算机 110 通常包括各种计算机可读介质。计算机可读介质可以是可由计算机 110 访问的任一可用介质，包括易失性和非易失性介质、可移动和不可移动介质。作为示例而非局限，计算机可读介质包括计算机存储介质和通信介质。计算机存储介质包括以用于储存诸如计算机可读指令、数据结构、程序模块或其它数据等信息的任一方法或技术实现的易失性和非易失性，可移动和不可移动介质。计算机存储介质包括但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘（DVD）或其它光盘存储、磁盒、磁带、磁盘存储或其它磁存储设备、或可以用来储存所期望的信息并可由计算机 110 访问的任一其它介质。通信介质通常在诸如载波或其它传输机制的已调制数据信号中包含计算机可读指令、数据结构、程序模块或其它数据，并包括任一信息传送介质。术语“已调制数据信号”指以对信号中的信息进行编码的方式设置或改变其一个或多个特征的信号。作为示例而非局限，通信介质包括有线介质，如有线网络或直接连线连接，以及无线介质，如声学、RF、红外和其它无线介质。上述任一的组合也应当包括在计算机可读介质的范围之内。

系统存储器 130 包括以易失性和/或非易失性存储器形式的计算机存储介质，如只读存储器（ROM）131 和随机存取存储器（RAM）132。基本输入/输出系统 133（BIOS）包括如在启动时帮助在计算机 110 内的元件之间传输信息的基本例程，通常储存在 ROM 131 中。RAM 132 通常包含处理单元 120 立即可访问或者当前正在操作的数据和/或程序模块。作为示例而非局限，图 1 示出了操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137。

计算机 110 也可包括其它可移动/不可移动、易失性/非易失性计算机存储介质。

仅作示例，图 2 示出了对不可移动、非易失性磁介质进行读写的硬盘驱动器 141、对可移动、非易失性磁盘 152 进行读写的磁盘驱动器 151 以及对可移动、非易失性光盘 156，如 CD ROM 或其它光介质进行读写的光盘驱动器 155。可以在示例性操作环境中使用的其它可移动/不可移动、易失性/非易失性计算机存储介质包括但不限于，磁带盒、闪存卡、数字多功能盘、数字视频带、固态 RAM、固态 ROM 等等。硬盘驱动器 141 通常通过不可移动存储器接口，如接口 140 连接到系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常通过可移动存储器接口，如接口 150 连接到系统总线 121。

上文讨论并在图 2 示出的驱动器及其关联的计算机存储介质为计算机 110 提供了计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 2 中，示出硬盘驱动器 141 储存操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147。注意，这些组件可以与操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同，也可以与它们不同。这里对操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 给予不同的标号来说明至少它们是不同的副本。用户可以通过输入设备，如键盘 162 和定点设备 161（通常指鼠标、跟踪球或触摸板）向计算机 110 输入命令和信息。其它输入设备（未示出）可包括麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等。这些和其它输入设备通常通过耦合至系统总线的用户输入接口 160 连接至处理单元 120，但是也可以通过其它接口和总线结构连接，如并行端口、游戏端口或通用串行总线（USB）。监视器 191 或其它类型的显示设备也通过接口，如视频接口 190 连接至系统总线 121。除监视器之外，计算机也可包括其它外围输出设备，如扬声器 197 和打印机 196，它们通过输出外围接口 195 连接。

计算机 110 可以在使用到一个或多个远程计算机，如远程计算机 180 的逻辑连接的网络化环境中操作。远程计算机 180 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它公用网络节点，并通常包括许多或所有相对于计算机 110 所描述的元件，尽管在图 2 中仅示出了存储器存储设备 181。图 2 描述的逻辑连接包括局域网（LAN）171 和广域网（WAN）173，但也可包括其它网络。这类网络环境常见于办公室、企业范围计算机网络、内联网以及因特网。

当在 LAN 网络环境中使用时，计算机 110 通过网络接口或适配器 170 连接至 LAN 171。当在 WAN 网络环境中使用时，计算机 110 可包括调制解调器 172 或用于通过 WAN 173，如因特网建立通信的其它装置。调制解调器 172 可以是内置或

外置的，通过用户输入接口 160 或其它适当的机制连接至系统总线 121。在网络化环境中，相对于计算机 110 所描述的程序模块或其部分可储存在远程存储器存储设备中。作为示例而非局限，图 2 示出远程应用程序 185 驻留在存储器设备 181 上。可以理解，示出的网络连接是示例性的，也可以使用在计算机之间建立通信链路的其它装置。

外部元数据处理系统

当创建数据流时，为数据流标识输入和输出数据集合。查询数据流的这些输入和输出数据集合，并且检索外部元数据（关于用于数据流的外部数据源的元数据）并对每一数据集合储存。数据集合的元数据描述了数据集合中可用的信息。

在一个实施例中，输入或输出数据集合包括记录信息。对于每一记录，储存多个列。对数据集合检索的外部元数据将对数据集合中可用的每一列储存描述以下列的各种元数据：

数据类型：储存在列中的数据的类型

长度：储存在列中的数据的长度

精度：储存在列中的数据的精度

代码页：如果列的数据类型是基于字符的，则为储存在列中的数据的代码页。代码页是其中向代码页中的每一字符分配数字值的字符的有序集合。

标度：如果数据本质上是数字的，则为储存在列中的数据的标度。标度是数据的小数点右边的数字位数。

在其它实施例中，也可储存描述可用列的元数据的其它变化。

该外部元数据最初在设置或配置数据流时储存。外部元数据的第一版本描述了当设计数据流时存在的数据集合。然后可在运行时对数据流使用外部元数据，以确定是否对数据流发生了改变。如下文所描述的，可通过重新同步来更新外部元数据。

确认和重新同步

当要使用数据流时，可使用外部元数据来确认数据流对当前时刻存在的数据集合的使用。以此方式，可检测到对数据集合的任何相关改变。另外，可使用确认信息来确定如何可解决所检测到的任何改变。

图 3 是依照本发明的一个实施例，使用至少一个外部数据文件准备关于数据

流的数据流信息的方法的流程图。如图 3 所示，在步骤 300，储存数据流模型。该数据流模块包括描述数据流以及描述数据流中来自多个外部数据文件（数据集合）的每一个的至少一个列的使用的信息。在步骤 310，储存元数据。储存的外部元数据描述了外部数据文件/数据集合中存在的列。然后，在步骤 320，使用储存的元数据来确定数据流的模型和外部数据文件是否一致。这被称为确认。

确认可以是连接或断开的。连接的确认在数据集合可用（连接）时发生，并可被查询以确定关于数据集合的内容的信息，并将其与储存的外部元数据比较。连接的确认确定了当前数据集合的每一个的相关状态是否在外部元数据中令人满意地表示，或者是否由于数据集合中的某一改变或破坏而存在问题。在一个实施例中，确定每一数据集合的确认状态。如果任一数据集合被分配了指示存在问题的确认状态，则向用户查询，以确定应当采取什么行动。另外，连接的确认也将每一数据集合的相关状态与数据流所需的信息进行比较。再一次，这一状态信息用于检测数据集合中在数据流运行时导致问题的改变。

断开的确认将外部元数据与数据流所需的信息进行比较，而不参考数据集合中的任何改变。用户可例如在数据集合不可用时选择执行断开的确认。由此，例如，当从向用户提供对数据集合的访问的网络断开时，缺少到数据集合的当前连接的用户仍可执行断开的确认。这一确认将允许用户标识对数据流的潜在的有问题的改变，而无需到数据集合的连接。

重新同步改变数据流，以维持其与当前数据集合（在联机的重新同步的情况下）或与当前储存的外部元数据（断开的重新同步）的兼容性。作出修补，它最小化了对数据流中剩余对象的改变，并最小化了必要的用户交互。

连接的确认和重新同步

如所讨论的，当发生连接的确认时，再一次查询数据集合，以确定对于该数据连接外部元数据是什么。对该数据集合的储存的外部元数据作出比较。如果存在差异，则它们都因此被检测。

如表 1 中所示，在确认时可出现若干情形：

数据集合中存在列？	在储存的外部元数据中存在列？	数据集合中的类型	储存的外部元数据中的类型	推论
否	是		X	列不再存在于数据集合中

是	否	X		列被添加到数据集合中
是	是	X	X	没有在外部元数据中检测到改变
是	是	X	Y	列具有数据集合中改变的数据类型

表 1：连接的确认－在将数据集合与储存的外部元数据比较的确认时的可能情形

在第一情形中，列存在于数据集合中，但不存在于储存的外部元数据中。由此，推论是，当创建外部元数据时的确存在的列已从数据集合中删除。在第二情形中，在收集储存的外部元数据时不存在的列现在存在于数据集合中，由此，推论是，列被添加到数据集合。在第三和第四情形中，列存在于数据集合中，也存在于储存的外部元数据中。在第三情形中，在储存的外部元数据中收集的类型信息与数据集合中的类型信息一致。由此，没有检测到外部元数据中对列的改变。然而，在第四情形中，列具有数据集合中改变的数据类型。在这一情形中，在一个实施例中，将向用户通知该不一致性。在一个实施例中，可给予用户修补储存的外部元数据中的数据类型的机会。

另外，在一个实施例中，也参考由数据流对数据集合实际使用的列来执行确认。可出现的情形在表 2 中示出：

数据集合中存在列?	数据流中存在列?	数据集合中的类型	数据流中期望的类型	推论/确认结果
否	是		X	由数据流引用的列已从源中删除/请求用户输入
是	否	X		数据集合中的列未在数据流中引用
是	是	X	X	数据集合中的列在数据流中被引用，所有的类型是同步的
是	是	X	Y	数据集合中的列在数据流中被引用，所有的类型信息是不

				同步的/试图解决

表 2：连接的确认－在将数据集合与数据流信息比较的确认时的可能情形

由此，如可在表 2 的第一情形中看到的，当列存在于数据流中而不存在于数据集合中时，出现确认问题。数据流中使用的列不存在于数据集合中。在这一情形中，在一个实施例中，请求用户输入以解决确认问题。

在第二和第三情形中，未遇到任何确认问题。数据集合中的列未在数据流中被引用，或者数据集合中的列在数据流中被引用，且所有的类型信息都对应。在这些情形中，未遇到确认问题。或者列不在数据流中使用，或列被使用且类型信息如所期望的。

在第四情形中，列存在于数据集合中，并在数据流中被引用，然而，类型信息改变。在这一情况下，在一个实施例中，作出解决不一致性的尝试。如果类型兼容，使得存在可解决任何类型不一致性的转换，则可使用该转化，并因此解决了不一致性。如果类型兼容，但是情况可以由数据流以某一其它方式（例如，通过数据流中实现的方法）来修补，则该不一致性也是可解决的。然而，如果类型不兼容，且情况不能被修补，则由于不兼容性确认不能完成。在一个实施例中，提示用户解决该不兼容性。

作为确认的结果，可返回确认状态。例如，在一个实施例中，确认状态 ISVALID 反映了确认没有问题。确认状态 ISBROKEN 指示存在问题，但是该问题可由用户或通过数据流中的方法来解决。例如，如上所述，可解决类型不一致性的类型不一致性转换可以是可用的。确认状态 NEEDSNEWMETADATA 指示应当执行重新同步（下文描述）。确认状态 ISCORRUPT 指示不能被简单地解决的问题。

当数据集合改变时，储存的元数据反映了在先前的时刻数据集合的状态。重新同步将储存的外部元数据与数据集合信息（连接的重新同步）或数据流信息（断开的重新同步）重新同步。

对于连接的重新同步，改变储存的外部元数据以与找到的数据集合信息协调。在一个实施例中，当发现数据集合中存在储存的外部元数据中未找到的列时，将关于该列的信息添加到储存的外部元数据中。当列存在于储存的外部元数据和数据集合中，但是类型信息已改变，则用来自数据集合的新类型信息更新储存的外部元数据。

类似地，在连接的重新同步中，更新数据流以反映数据集合信息的改变。当

数据流中引用的列在数据集合中不再可用时，从数据流中删除该列。另外，如上文参考确认所讨论的，当对于在储存的外部元数据中找到信息的列改变了类型时，可以有解决差异的方法，并且重新同步中的差异可被解决（如果可能的话）。在一个实施例中，可以调用数据流的所展示的方法，以解决不一致性，并且数据流被改变。在另一实施例中，向用户作出解决不一致性的请求。

断开的确认和重新同步

如所讨论的，当发生断开的确认时，不查询数据集合。将储存的外部元数据与数据流比较，以确保它们兼容。可能的情形在表 3 中示出：

储存的外部元数据中存在列?	数据流中存在列?	储存的外部元数据中的类型	数据流中期望的类型	推论/确认结果
否	是		X	由数据流引用的列已从源删除/请求用户输入
是	否	X		数据集合中的列未在数据流中被引用
是	是	X	X	数据集合中的列在数据流中被引用，所有的类型信息是同步的
是	是	X	Y	数据集合中的列在数据流中被引用，所有的类型信息是不同步的/试图解决

表 3：连接的确认－在将储存的外部元数据与数据流信息进行比较的确认时的可能情形

由此，如可以在表 3 的第一情形中见到的，当列存在于数据流中但不存在于储存的外部元数据中，出现确认问题。数据流中使用的列不被储存的外部元数据指示为存在于数据集合中。在这一情形中，在一个实施例中，请求用户输入，以解决

确认问题。

在第二和第三情形中，未遇到确认问题。被指示为存在于储存的外部元数据中的列不在数据流中被引用，或者被指示为存在于储存的外部元数据中的列在数据流中被引用，且所有的类型信息都对应。在这些情形中，未遇到确认问题。或者列不在数据流中使用，或者使用了列且类型信息是所期望的。

在第四情形中，列存在于储存的外部元数据中且在数据流中被引用，然而类型信息已改变。在这一情况下，在一个实施例中，作出解决不一致性的尝试。如果类型兼容，使得存在可解决任何类型不一致性的转换，则使用该转换，并因此解决不一致性。如果类型不兼容，但情况可由数据流以某一其它方式（例如，通过数据流组件的接口）来修补，则该不一致性也是可解决的。然而，如果类型不兼容，且情况不能被修补，则由于不兼容性确认不能完成。在一个实施例中，提示用户解决不兼容性。

图 4 是依照本发明的一个实施例评估数据流模型的流程图。在步骤 400，检索描述外部数据文件的元数据。在步骤 410，使用该数据来确定数据流模型和元数据是否一致。

当执行断开的重新同步而非断开的确认时，改变数据流以与储存的外部元数据中反映的数据集合的理解协调。由此，在一个实施例中，如果列存在于数据流中而不存在于储存的外部元数据中，则删除数据流中的列以反映对该列在数据集合中不可用的理解，如由储存的外部元数据中的列的缺乏所指示的。另外，如果数据流和储存的外部元数据包含列的不同类型信息，则解决差异（如果可能的话）。在一个实施例中，可调用数据流的所展示的方法来解决不一致性，并且改变数据流。在另一实施例中，向用户作出解决不一致性的请求。

结论

注意，上述示例仅为了解释的目的而提供，并且决不被解释为限制本发明。尽管参考各种实施例描述了本发明，可以理解，此处所使用的词语是描述和说明的词语，而非限制的词语。此外，尽管参考特定的装置、材料和实施例描述了本发明，然而本发明并不打算限于此处所解释的细节；相反，本发明延及如在所附权利要求书范围内的所有功能上等效的结构、方法和使用。从本说明书的教导中获益的本领域的技术人员可以在不脱离本发明的各方面的范围和精神的情况下对其实现各种修改并可作出改变。

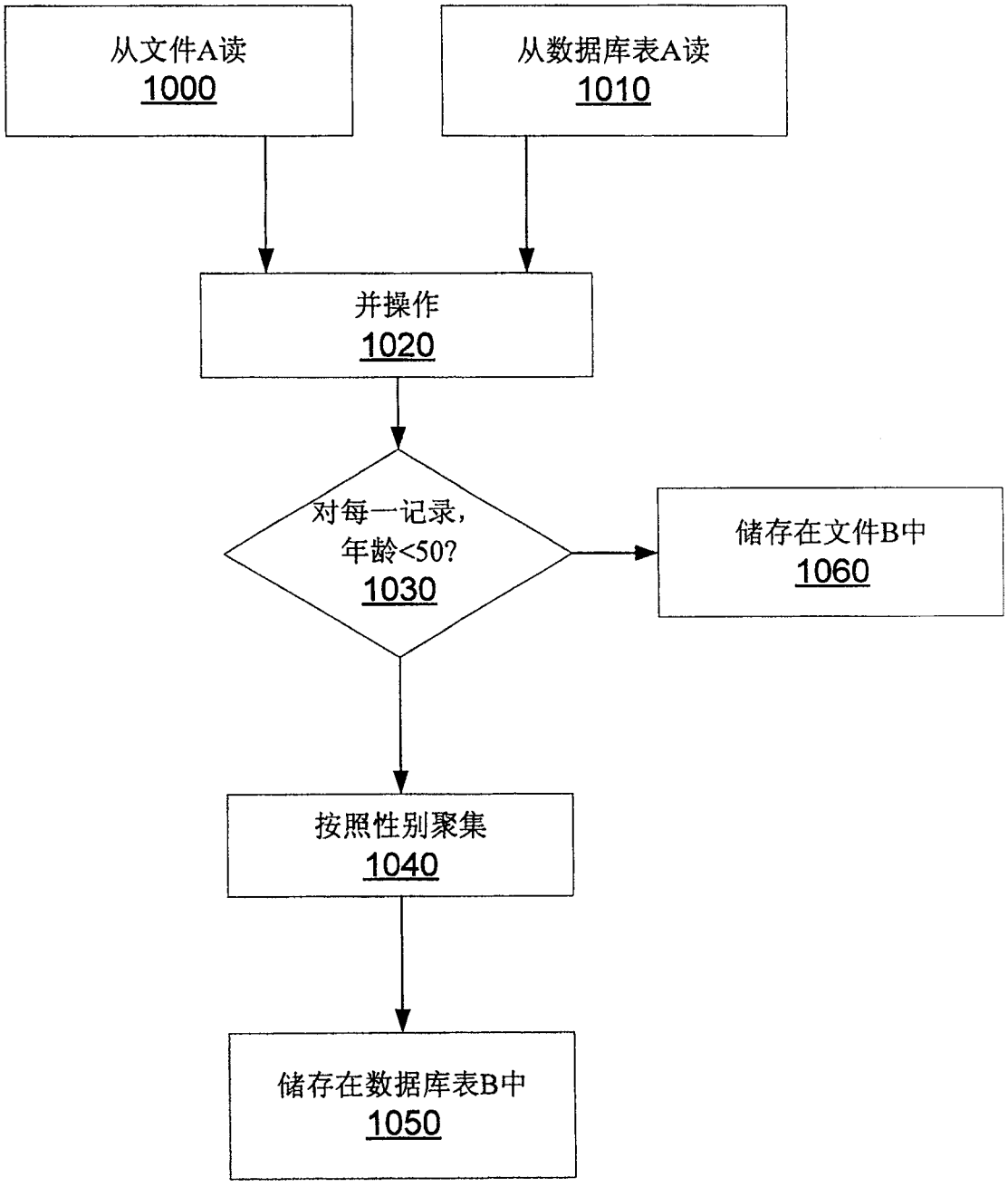

从文件A读
1000
从数据库表A读
1010
并操作
1020
对每一记录，
年龄<50?
1030
储存在文件B中
1060
按照性别聚集
1040
储存在数据库表B中
1050

图 1

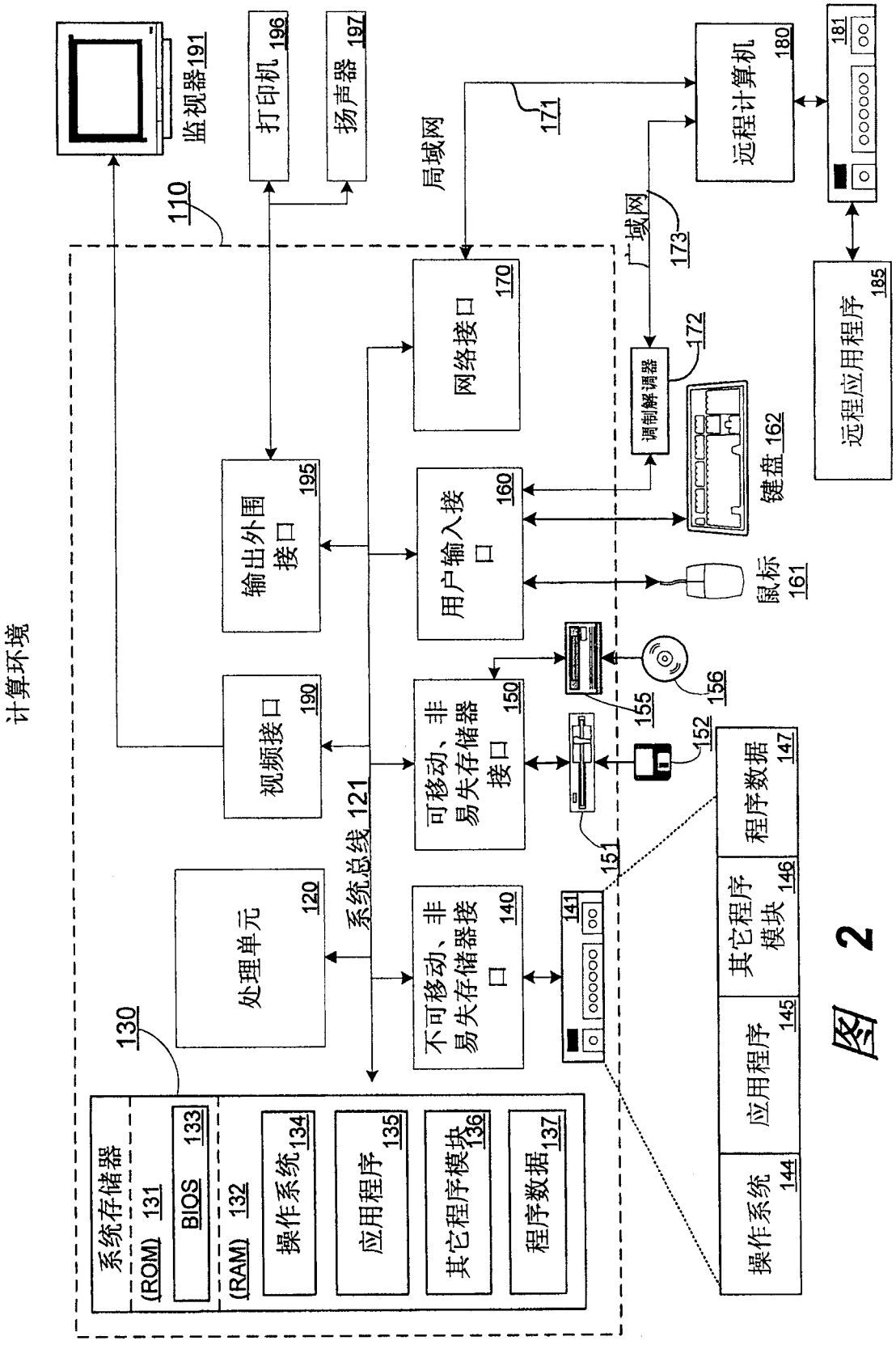

计算环境
110
监视器191
打印机 196
扬声器 197
局域网
171
远程计算机 180
181
广域网
173
远程应用程序 185
网络接口 170
调制解调器 172
输出外围接口 195
用户输入接口 160
键盘 162
鼠标 161
视频接口 190
可移动、非易失存储器接口 150
155
156
152
151
处理单元 120
系统总线 121
不可移动、非易失存储器接口 140
141
130
系统存储器
(ROM) 131
BIOS 133
(RAM) 132
操作系统134
应用程序 135
其它程序模块136
程序数据137
操作系统 144
应用程序 145
其它程序模块 146
程序数据 147

图 2

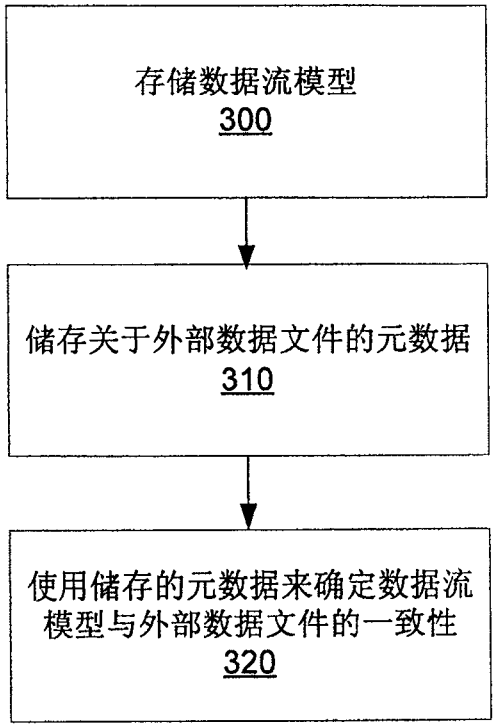

存储数据流模型
300
储存关于外部数据文件的元数据
310
使用储存的元数据来确定数据流模型与外部数据文件的一致性
320

图 3

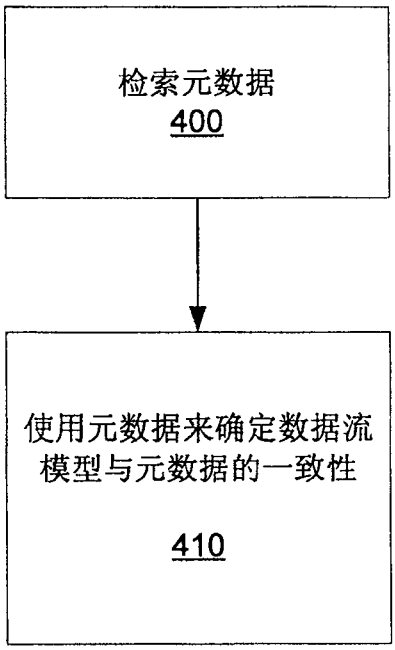


图 4