



(12) 发明专利

(10) 授权公告号 CN 102687142 B

(45) 授权公告日 2014. 11. 26

(21) 申请号 201080054527. 2

(22) 申请日 2010. 11. 29

(30) 优先权数据

12/631, 088 2009. 12. 04 US

(85) PCT国际申请进入国家阶段日

2012. 06. 01

(86) PCT国际申请的申请数据

PCT/EP2010/068443 2010. 11. 29

(87) PCT国际申请的公布数据

W02011/067216 EN 2011. 06. 09

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 M·Y·塞尔瓦格 P·A·科尔迪科特

陶晓峰

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 张亚非 于静

(51) Int. Cl.

G06F 17/30 (2006. 01)

(56) 对比文件

US 2007276883 A1, 2007. 11. 29,

CN 101547445 A, 2009. 09. 30,

CN 101154159 A, 2008. 04. 02,

审查员 姚楠

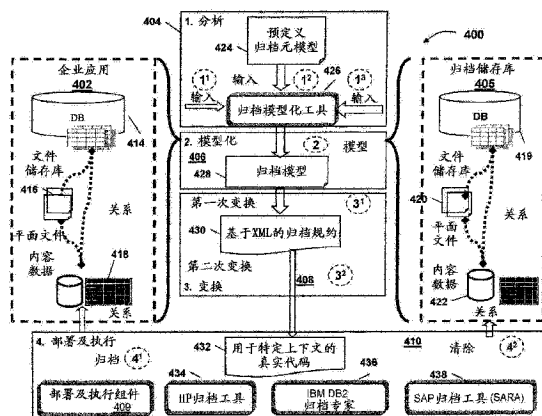
权利要求书2页 说明书9页 附图6页

(54) 发明名称

一种使用模型驱动方法的灵活数据归档的方法和装置

(57) 摘要

本发明提供一种用于使用模型驱动方法的灵活数据归档的系统及方法。本发明的该数据归档系统及方法提供将数据自原数据储存库移动至归档数据储存库中的操作。用户可指定描述将移动数据的哪些部分的归档条件。该方法可具有诸如以下各项的步骤：分析应用内容，定义归档模型，通过基于模型化结果产生归档规约来变换该模型且基于该归档模型构建应用，及部署新模型及归档应用。



1. 一种用于在系统中使用模型驱动方法的灵活数据归档的方法,该系统具有真实上下文且进一步具有一具有内容的应用,该方法包含:

分析该应用内容;

定义及模型化归档数据;

创建第一数据归档规约模型,其中包括使用图形数据归档元模型根据要求及所述分析该应用内容的结果来定义所述第一数据归档规约模型,其中所述元模型中包含归档组件,该归档组件包含归档属性及清除属性,其中当该清除属性设定为真时,该归档元素清除所归档的数据,当该清除属性不设定为真时,该归档元素归档数据;

基于该元模型变换该第一数据归档规约模型且基于图形模型化工具模型化该数据归档;

产生第二数据归档规约模型;

基于该第二数据归档规约模型及真实上下文来构建归档应用;及

部署该归档应用且创建数据归档。

2. 如权利要求 1 所述的方法,进一步包含识别数据归档要求且捕获与数据类型、关系及归档规则相关的信息。

3. 如权利要求 2 所述的方法,其进一步包含基于数据、数据类型、关系、数据筛选条件、归档规则及数据源与归档储存库来定义该第二数据归档规约模型。

4. 如权利要求 1 所述的方法,进一步包含测试该数据归档。

5. 如权利要求 4 所述的方法,其中存在归档结果,且该方法进一步包含观察该归档结果且反馈该归档结果以用于归档模型优化。

6. 如权利要求 1 所述的方法,其中所述元模型还包含:

归档储存库,其用于归档和清除数据;

排程组件,其用于指示开始执行归档过程的时间;

对象组件,其用于指示待归档的事项;

归档条件组件,其用于标识归档的事项及归档的量。

7. 如权利要求 6 所述的方法,其中该排程组件具有标识开始执行归档过程的开始时间分量以及标识归档过程的两次执行之间的间隔的分量。

8. 如权利要求 7 所述的方法,其中该对象组件具有唯一标识且被分类为文件或表。

9. 一种用于在系统中使用模型驱动装置的灵活数据归档的装置,该系统具有真实上下文且进一步具有一具有内容的应用,该装置包含:

用于分析该应用内容的装置;

用于定义及模型化归档数据的装置;

用于创建第一数据归档规约模型的装置,其中包括用于使用图形数据归档元模型根据要求及所述分析该应用内容的结果来定义所述第一数据归档规约模型的装置,其中所述元模型中包含归档组件,该归档组件包含归档属性及清除属性,其中当该清除属性设定为真时,该归档元素清除所归档的数据,当该清除属性不设定为真时,该归档元素归档数据;

用于基于该元模型变换该第一数据归档规约模型且基于图形模型化工具模型化该数据归档的装置;

用于产生第二数据归档规约模型的装置;

用于基于该第二数据归档规约模型及真实上下文来构建归档应用的装置；及
用于部署该归档应用且创建数据归档的装置。

10. 如权利要求 9 所述的装置，进一步包含用于识别数据归档要求且捕获与数据类型、关系及归档规则相关的信息的装置。

11. 如权利要求 10 所述的装置，其进一步包含用于基于数据、数据类型、关系、数据筛选条件、归档规则及数据源与归档储存库来定义该第二数据归档规约模型的装置。

12. 如权利要求 9 所述的装置，进一步包含用于测试该数据归档的装置。

13. 如权利要求 12 所述的装置，其中存在归档结果，且该装置进一步包含用于观察该归档结果且反馈该归档结果以用于归档模型优化的装置。

14. 如权利要求 9 所述的装置，其中所述元模型还包含：

归档储存库，其用于归档和清除数据；

排程组件，其用于指示开始执行归档过程的时间；

对象组件，其用于指示待归档的事项；

归档条件组件，其用于标识归档的事项及归档的量。

15. 如权利要求 14 所述的装置，其中该排程组件具有标识开始执行归档过程的开始时间分量以及标识归档过程的两执行之间的间隔的间隔分量。

16. 如权利要求 15 所述的装置，其中该对象组件具有唯一标识 (ID) 且被分类为文件或表。

一种使用模型驱动方法的灵活数据归档的方法和装置

技术领域

[0001] 本发明涉及一种使用模型驱动方法的数据归档的方法或系统。本发明提供一种使用模型驱动方法的灵活、通用的数据归档解决方案。

背景技术

[0002] 许多企业遭受数据爆炸问题。数据每年以指数速率增长,其很大程度上是由活动的数据、非活动的数据及不可用数据的混合而引起的。愈来愈多的政府法规强制企业将商业(business)上重要的数据保持一定时间范围。举例而言,萨班斯-奥克斯利法案(Sarbanes-Oxley Act)、健康保险可移植性和责任法案(Health Insurance Portability and Accountability Act, HIPAA)及巴塞尔 II(巴塞尔协议第二版)都强制企业将商业上重要的数据保持一定时间范围。统计数据显示在操作环境中非活动数据及不可用数据的百分比非线性地增加。此问题导致以下主要问题:

[0003] 1. 数据爆炸;

[0004] 2. 应用性能降级;及

[0005] 3. 信息技术(IT)成本(例如,维护成本、硬件及存储成本等)的增加。

[0006] 数据归档为一种用于自操作环境选择非活动数据及不可用数据且将此数据移动至归档空间以供未来使用的实用方法。当不再需要非活动数据时,可将其自归档空间移除。这称为数据清除。因此,不仅可增强企业应用的性能,也可减少并节省成本。除引起性能降级及 IT 成本增加的“什么也不做”的方法的外,存在三种现今所采用的主要数据归档解决方案。

[0007] 第一种解决方案为手动地进行数据归档。在此情形下,使用者手动地选择、移动及移除数据。举例而言,数据库管理员可使用例如结构化查询语言(SQL)或使用通用数据库实用程序发出查询以对关系数据库进行查询且将查询结果保存为文件,且接着使用(例如)文件传送协议(FTP)将文件传送至另一位置,该 FTP 为用以经由基于 TCP/IP 的网络(诸如,因特网)交换及操纵文件的标准网络协议。此解决方案可能看似简单且不产生大的预付成本,但缺点在于其产生高风险且经常导致数据完整性问题,因此,此方法对企业造成潜在损害且甚至可能导致灾难。

[0008] 第二种解决方案为在具备充分计划及资金的情况下使应用具有其自身的特定“数据归档”功能。此解决方法具有以下缺陷。

[0009] 第一个缺陷在于,当应用需要重写时,数据归档功能或组件(component)由于紧密耦合架构也需要重写。

[0010] 第二个缺陷在于,其延长项目开发生命周期,因为使用者必须进行诸如分析、设计、实施及测试的额外归档功能开发。

[0011] 第三个缺陷在于,存在额外成本;每个项目必须在归档功能的实施上添加资源。

[0012] 另一缺陷在于,数据归档功能高度专用于特定类型的数据且足够通用以满足不同应用的变化数据归档要求。

[0013] 第三种主要数据归档解决方案为使用现成的数据归档产品。许多软件公司已提供其自身的数据归档产品。举例而言,国际商业机器公司 (IBM)、惠普公司 (HP)、Oracle 公司、SAP AG 及其他公司已开发数据归档产品。这些数据归档产品一般可提供用于数据归档的可配置控制台及可编程工具。

[0014] 然而,这些当前产品也具有限制。一个这样的限制在于,存在有限的数据源及位置支持。这些工具中的大部分仅支持特定关系数据库 (诸如, **IBM® DB2®**、Oracle 数据库 (通常称为 Oracle RDBMS 或简称为 Oracle) 等), 且仅归档至表或平面文件 (flat file)。

[0015] 另一限制在于,存在有限的数据类型支持。大部分的目前数据归档产品仅支持关系数据库中的常见数据类型。

[0016] 另外,一些归档工具仅像备份那样简单地复制文档 (文件), 而从不考虑商业逻辑。

[0017] 最后,目前归档工具并不具有足够的灵活性来改变归档规则。尽管一些归档工具集成使用者可选择的更大规模的归档规则,但这些规则硬编码于系统中且难以改变。

[0018] 以上问题及缺陷限制了这些解决方案的普遍使用。如所已知的,在企业环境中,不仅存在存储于数据库中的数据,而且存在存储于文件、文档、电子邮件及 XML (可扩展性标记语言) 中的数据。

[0019] 又,它仅具有少数基本的、粗粒度的且固定的归档规则,例如,什么存储池目标、如果文件处于使用中时则如何以及保持多长时间。因此,需要解决与如上所述相关联的问题。

发明内容

[0020] 本发明提供一种用于数据归档的系统及方法。本发明提供一种使用模型驱动方法的灵活、通用的数据归档解决方案。它定义了用于该系统的一元模型。

[0021] 本发明的数据归档为将数据自一原始数据储存库移动至一归档数据储存库中的操作。用户可指定描述将移动数据的哪些部分的归档条件。若该条件为空,即用户在归档数据时未指定任何条件,则数据将完全被移除至归档储存库。若该条件不为空,则不满足该归档条件的该数据将保留于原始数据储存库中。

[0022] 此外,本发明提供了一种用于在一系统中使用一模型驱动方法的灵活数据归档的方法,该系统具有真实上下文且进一步具有一具有内容的应用,该方法可具有以下步骤:分析该应用内容,定义及模型化 (modeling) 归档数据,创建第一数据归档规约模型,变换该第一数据归档规约模型,产生第二数据归档规约模型,基于该第二数据归档规约模型及该真实上下文构建一归档应用,及部署该第二数据归档规约模型且创建一数据归档。

[0023] 此外,本发明提供了一种用于使用一模型驱动方法的灵活数据归档的计算机系统,该系统具有真实上下文且进一步具有 CPU、计算机可读存储器及计算机可读存储媒体、一具有内容的应用,该计算机系统可具有:一分析器,其用于分析该应用内容,其可具有用以分析该应用内容的程序指令;一定义及模型化组件,其用于定义及模型化归档数据且用于创建第一数据归档规约模型,其可具有对于一定义及模型化组件以用于定义及模型化归档数据且用于创建第一数据归档规约模型的程序指令;一变换器,其用于变换该第一数据归档规约模型且用于产生第二数据归档规约模型,其可具有用以变换该第一数据归档规约模型且产生第二数据归档规约模型的程序指令;一数据归档规约组件,其具有一部署组件,

该部署组件用于部署该模型且创建数据归档,其可具有用以部署该模型且创建一数据归档的程序指令;及一归档应用组件,其用于基于该第二数据归档规约模型及该真实上下文来构建一归档应用,其可具有用以基于该第二数据归档规约模型及真实上下文来构建一归档应用的程序指令,其中这些程序指令存储于该计算机可读存储媒体上以供该 CPU 经由该计算机可读存储器执行。

[0024] 此外,本发明提供一种用于灵活数据归档的图形数据归档元模型,其可具有:一归档组件,其用于归档及清除数据;一排程组件,其用于指示开始执行一归档过程的时间;一对象组件,其用于指示待归档的事项;及一归档条件组件,其用于识别归档的事项及归档的量。

[0025] 此外,本发明提供一种计算机程序产品,其体现于一计算机可读媒体中以在一系统中操作,该系统包含处理单元、存储器、总线、输入/输出(I/O)接口、应用,该计算机程序产品用于在一系统中实施一种用于使用一模型驱动方法的灵活数据归档的方法,该系统具有真实上下文且进一步具有一具有内容的应用,该方法可具有以下步骤:分析该应用内容,定义及模型化归档数据,创建第一数据归档规约模型,变换该第一数据归档规约模型,产生第二数据归档规约模型,基于该第二数据归档规约模型及真实上下文来构建一归档应用,及部署该第二数据归档规约模型且创建一数据归档。

[0026] 此外,本发明提供了一种用于部署一计算基础结构的方法,其包含将计算机可读代码集成至一计算系统中,其中与该计算系统组合的该代码能够执行用于在一系统中使用一模型驱动方法的灵活数据归档的一过程,该系统具有真实上下文且进一步具有一具有内容的应用,该过程可具有诸如以下各项的步骤:分析该应用内容,定义及模型化归档数据,创建第一数据归档规约模型,变换该第一数据归档规约模型,产生第二数据归档规约模型,基于该第二数据归档规约模型及真实上下文来构建一归档应用,及部署该第二数据归档规约模型且创建一数据归档。

附图说明

[0027] 将参照以下附图仅作为示例描述本发明的优选实施例,所述附图中:

[0028] 图 1 展示适合于实施本发明的实施例的数据处理系统。

[0029] 图 2 展示用于实施本发明的实施例的网络;

[0030] 图 3 示出本发明的方法的实施例;

[0031] 图 4 示出本发明的系统及方法的实施例;

[0032] 图 5 示出本发明的元模型图;以及

[0033] 图 6 示出本发明的系统及方法的另一实施例。

[0034] 这些附图未必是按比例的。这些附图仅为示意性表示,其不旨在描绘本发明的特定参数。这些附图旨在仅描绘本发明的典型实施例,且因此不应被视为限制本发明的范围。

具体实施方式

[0035] 本发明包含一种用于归档数据的系统及方法。本发明的系统及方法在企业商业对象层级而非数据库记录、表及文件层级上考虑数据归档。数据归档为面向商业的且不同的企业应用具有不同的数据归档规则。具有为面向商业的、独立于个别应用、数据库及系统的

灵活、通用的端到端数据归档解决方案极为重要。

[0036] 图 1 展示可具有适合于实施本发明的实施例的数据处理系统 102 的系统 100。数据处理系统 102 可具有计算机系统 104, 计算机系统 104 连接至显示器 120、(一个或多个) 外部设备 116 及或其他外围设备, 所述设备用于向用户提供到经由 (一个或多个) I/O 接口 114 连接的计算机系统 104 的接口。计算机系统 104 可具有内部总线 112, 内部总线 112 用于提供诸如处理单元 106、(一个或多个) I/O 接口 114、网络配接器 138 及存储器 110 的模块之间的内部通信。存储器 110 可具有随机存取存储器 (RAM) 130 及高速缓存 132。存储系统 118 或其他形式的存储器可连接至计算机系统 104。RAM 可采用允许以任何次序 (即, 随机地) 存取所存储的数据的集成电路的形式。存储系统 118 可采取磁带、磁盘及光盘的形式, 且通常用于数据的长期存储。高速缓存 132 为用于存储数据集合 (复制存储于其他处或早先计算的原始值) 的存储器, 其中与读取高速缓存的成本相比, 取 (由于较长存取时间) 或计算原始数据是昂贵的。换言之, 高速缓存为可存储频繁存取的数据以用于快速存取的临时存储区域。一旦数据存储于高速缓存中, 便可在未来通过存取高速缓存的复本而非重新取或重新计算原始数据来使用该数据。已证明高速缓存在许多计算领域中极为有效, 因为典型的计算机应用中的存取模式具有访问局部性 (locality of reference)。

[0037] 图 2 展示用于实施本发明的实施例的网络系统 200。网络系统 200 可具有网络 210 或经由网络连接 206、208 互连的一组计算机 (诸如, 数据处理单元 202、204), 且可为例如局域网 (LAN) 或互连网络的类型。打印机 212 及存储器 214 可经由网络连接 216、218 连接至网络 210。基本网络组件可包括网络适配卡、中继器、集线器、桥接器、交换器及路由器。数据处理单元 202、204 可为诸如 web 服务器或个人计算机的计算机或其他用户代理。web 服务器通常具有负责自客户端 (诸如 web 浏览器的用户代理) 接受 HTTP 请求且将 HTTP 响应与可选数据内容一起提供给这些客户端的硬件及软件, 这些可选数据内容通常为诸如 HTML 文件及链接对象 (图像等) 的 web 网页。在本文档中, 使用术语“web 浏览器”, 但必须考虑用于在因特网上检索、呈现及传送信息资源的任何应用。

[0038] 存在使用本发明的实施例中的灵活、通用数据归档解决方案的四个主要步骤。

[0039] 第一步骤为分析应用内容。通过分析应用内容, 可识别数据归档要求且捕获与数据类型、关系及归档规则相关的信息。

[0040] 第二步骤为定义及模型化归档数据。通过定义及归档数据, 可基于在本发明中所提供的元模型来定义数据归档模型, 且基于图形模型化工具 (例如, 基于 Eclipse 的工具) 来模型化数据归档。Eclipse 为多语言软件开发环境, 其包含集成开发环境 (IDE) 及用以扩展它的插入 (plug-in) 系统。“元模型化”为构建一定域内的“概念” (事物、术语等) 的集合。模型是对真实世界中的现象的抽象化; 元模型为进一步的抽象化, 其突出了模型本身的属性。

[0041] 在第二步骤完成之后, 可创建以 UML 表示的数据归档规约模型。统一模型化语言 (UML) 为软件工程领域中的标准化通用模型化语言。使用本发明的实施例, 可指定及模型化企业应用中的归档数据集合的不同方面 (perspective), 诸如以下各项: 待归档的事项, 例如, 数据、数据类型、关系、数据筛选条件; 及归档的时间及方式, 即, 归档规则。

[0042] 第三步骤为执行模型变换且产生一归档规约 (诸如, 基于 XML 的规约), 且基于该归档规约及真实上下文来构建归档应用。

[0043] 第四步骤为部署这些模型且测试该归档应用。可选地,本发明的实施例中的方法及系统可通过观察归档结果及反馈至归档模型优化而优化归档模型。

[0044] 本发明的实施例的系统及方法具有许多优点。它通过改良操作应用的性能、节省管理成本、硬件及存储成本提供了商业优点。它提供了对归档要求的快速响应,且减轻了法规符合性的风险。它通过缩短项目生命周期减小了上市时间。它可通过利用相应数据归档产品来提高数据管理市场中的能力,且可通过利用相应数据归档产品来增加来自数据市场的收入。它可由于其为灵活的而提供技术优点,因为解决方案基于模型驱动开发方法,且提供良好的用户体验,因为图形模型化工具为可用的。

[0045] 本发明的实施例中的模型驱动方法及系统允许用户更为关注归档要求及规则,而非指定归档应用的构建。它可以是可支持较多现存归档工具的基于开放标准的(诸如,基于 XML 的归档规约)。基于 XML 的归档规约可视为用于归档的规范的(canonical)数据模型。

[0046] 本发明的实施例中的模型驱动方法及系统提供了一种通用数据归档解决方案,其不仅限于指定数据及指定数据存储/管理系统(其中数据归档解决方案可独立地工作),或与现存企业归档解决方案(诸如,IBM® DB2®归档专家或 SAP AG 的 SAP 归档工具)一起工作。

[0047] 本发明的实施例中的一种方法执行以下步骤:

[0048] 1. 分析企业应用的数据归档上下文;

[0049] 2. 定义数据归档规约模型;

[0050] 3. 执行从基于数据归档模型的规约至基于 XML 的规约的变换;及

[0051] 4. 部署数据归档模型且执行数据归档。

[0052] 一旦完成变换,本机(native)归档代码/规则/规约将被部署至相应归档引擎中。接着,新数据归档应用可开始执行。在一个实施例中,数据归档解决方案在于灵活的可扩展的元模型组件。用于数据归档定义的所有必要元数据信息包含于数据归档元模型中。

[0053] 图 3 示出本发明的方法及计算机系统 300 在一计算机系统中的一个实施例,该计算机系统具有 CPU、计算机可读存储器及计算机可读存储媒体,其中在 302 处,分析应用上下文,其可具有用以分析应用内容的程序指令;在 304 处,定义归档模型,其可具有对于一定义及模型化组件的程序指令以用于定义及模型化归档数据且用于创建第一数据归档规约模型;在 306 处,由程序指令可能地变换该模型,以变换该第一数据归档规约模型且产生第二数据归档规约模型;且在 308 处,部署该模型,其可利用程序指令以部署该模型且创建数据归档。该计算机系统可利用程序指令以基于该第二数据归档规约模型及真实上下文来构建归档应用,其中这些程序指令存储于该计算机可读存储媒体上以供该 CPU 经由该计算机可读存储器执行。该计算机系统可进一步具有程序指令以测试数据归档。分析应用上下文步骤 302 在识别数据归档要求 318 处识别数据归档要求,且在辨识数据信息及归档规则步骤 320 处辨识数据信息(类型、关系等)及归档规则。定义归档模型步骤 304 在 322 处使用所提供的元模型根据要求及分析结果来定义模型,且在 324 处指定模型化元素(element)的属性(诸如,位置、条件、排程等)。模型变换步骤 306 在 326 处基于模型化结果来产生基于 XML 的归档规约,且在 328 处基于归档规约及真实上下文来构建一归档应用。部署模型步骤 308 在 330 处将所产生的结果部署至真实环境中,且在 332 处测试归档应用。

[0054] 图 4 展示本发明的示例性系统及方法 400, 其具有模型驱动方法以产生一数据归档应用, 其具有企业应用 402、分析组件 404、模型化组件 406、变换组件 408、归档储存库 405 以及部署及执行组件 409。企业应用 402 可使用关系数据库 414、文件储存库 416 及存储 XML 内容的内容储存库 418。分析组件 404 具有预定义的归档元模型 424 及归档模型化工具 426。模型化组件 406 具有归档模型组件 428。变换组件 408 具有基于 XML 的归档规约 430。归档储存库 405 可使用关系数据库 419、文件储存库 420 及存储 XML 的内容储存库 422。部署及执行组件 409 具有部署及执行归档基础结构 410, 部署及执行归档基础结构 410 具有归档特征 4¹、清除特征 4²、IBM DB2 归档专家 436、SAP 归档工具 (SARA) 438 及 HP 归档工具 434。

[0055] 图 5 示出本发明的数据归档解决方案元模型 500 的实施例。数据归档解决方案元模型 500 可具有排程(scheduling)元素 550, 排程元素 550 具有众多分量(component), 诸如, startTime:string 及 Interval:string。Scheduling 元素 550 可经由连接 552 连接至 Archive (归档) 元素 502 以用于传送一“排程器”组件。Archive 元素 502 可具有众多分量, 诸如, objectSet:Object, scheduler:Schedule, archiveCondition:ArchiveCondition 及 purge:Boolean。Archive 元素 502 可经由连接 506 连接至 ArchiveCondition (归档条件) 元素 504 以用于传送 archiveCondition, 且经由连接 516 连接至 Object (对象) 元素 514 以用于传送 objectSet。ArchiveCondition 元素 504 可具有多个分量, 诸如, ownerObject:Object 及 EvaluationSet:Evaluation。ArchiveCondition 元素 504 可连接至 Evaluation (评估) 元素 508 以用于传送 Evaluation 分量, 且经由连接 512 连接至 Object 元素 514 以用于传送 ownerObject 分量。Evaluation 元素 508 可具有多个分量, 诸如, evalOp:Evaluation、srcValSpeci:ValueSpecification 及 dstValSpeci:ValueSpecification。Evaluation 元素 508 可经由连接 520 连接至 Variability Point (可变点) 元素 518 以用于传送 evalOp 分量且用于存储为 Variability Point 元素 518 中的 EvaluOpEnum 分量。

[0056] 具有诸如 ID:Integer、srcLocation:Location、dstLocation:Location 及 ownerObject:Object 的 Object 元素 514 可连接至 Variability Point 元素 540 以用于传送 srcConveying 分量且存储为 Variability Point 元素 540 中的 Location。Variability Point 元素 540 可连接至 fileLocation(文件位置)元素 538 以用于自 fileLocation 元素 538 接收数据, fileLocation 元素 538 可具有如下分量:Path:String 分量及 FileName:String 分量。Variability Point 元素 540 可进一步连接至 tableLocation (表位置) 元素 536, tableLocation 元素 536 具有众多分量, 诸如, Server:String、user:String、password:String、tableSpace:string、dbname:String 及 tablename:String。

[0057] Object 元素 514 可经由连接 546 连接至 Attribute (属性) 元素 534 以用于传送 ownerObject 分量。Attribute 元素 534 可具有诸如 ownerObject:Object、attriName:String 及 attriType:Attribute 的分量, 且可经由连接 548 连接至 ValueSpecification (值规约) 元素 526。ValueSpecification 元素 526 也可经由连接 548 连接至 Constant (常数) 元素 528, Constant 元素 528 具有诸如 constVal:string 及 constType:data 的分量。ValueSpecification 元素 526 也可连接至 Expression (表达式) 元素 522, Expression 元素 522 可具有分量 Operator:OperEnum 及 ValueSequence。

Expression 元素 522 可进一步连接至可变点 524 以用于传送 operator 分量且存储为运算 OperEnum。另一可变点 530 包括 DataType。

[0058] 此元模型中的一些主要元素为 :Archive 元素 502,其代表整个归档处理过程。若将 purge 属性(分量)设定为真,则此元素代表清除操作。

[0059] 否则,它代表归档操作。以下为 Archive 元素的一些参数(分量)。objectSet :所有对象需要在此归档过程中归档 ;scheduler :此归档过程的排程 ;archiveCondition :此归档过程中的对象的所有条件 ;及 purge :指示归档元素是处于归档操作中或者清除操作中的一布尔值。排程元素 550 指示开始执行整个归档过程的时间。以下为排程元素的一些参数 :startTime 标识开始执行归档过程的时间 ;及 interval 描述两次执行之间的间隔。举例而言,在第一次执行之后,第二次执行将在等待由 interval 指示的时间之后开始执行。Object 元素 514 指示待归档的事项。Object 元素 514 可基本分类为两种类型中的任一种 :文件(包括目录)及表。对象在被归档时可受其他对象约束。每一对象在归档过程范围中具有唯一标识(ID)。在相应的归档过程中,对象可通过其 ID 访问。ID 的类型为统一资源识别符(URI)。URI 由用以识别或命名因特网上的资源的字符串组成。srcLocation :源数据储存库的规约。dstLocation :归档数据储存库的规约。

[0060] ArchiveCondition :ArchiveCondition 元素表明待归档的事项及归档的量。如果对象不满足在 ArchiveCondition 中所指示的条件,则将不归档来自 Object 元素的数据。有时,尽管对象满足条件,但并非其中的所有数据都可归档。最后仅可归档数据的满足 ArchiveCondition 中所指定条件的部分。Evaluation :一个评估对应于用于特定归档对象的一个独立归档条件。ValueSpecification Expression :运算符应用于顺序运算数。运算数可为任何 ValueSpecification 实体 :Attribute、Constant 或 Expression。“Attribute”用以描述所归档数据的属性名称及类型。“Location”用以模型化数据源与归档储存库,且模型化执行数据归档的地点。

[0061] 由于存在各种数据、数据类型及数据储存库及管理系统,数据归档元模型通过提供可变点来提供灵活性及可扩展性。图 5 中的虚线圆圈标识了这些可变点。

[0062] 可变点“Location”540 :对于异质数据归档储存库,可容易地引入自 Location 元素所继承的新的位置类型。图 5 中的示例展示了文件及关系数据库表位置,但用户可将位置扩展至 IMS 段(segment)或 XML 储存库位置。IBM 信息管理系统(IMS)为具有广泛的事务处理能力的联合层级式数据库及信息管理系统。XML(可扩展标记语言)为用于以电子形式编码文档的规则集合。“AttributeType”(Attribute534) :对于元数据类型,定义了称为 AttributeType 的枚举类型。可容易地将新的元数据类型作为新的枚举来添加。“DataType”(在可变点 530 处) :对于数据类型,定义了称为 DataType 的枚举类型。可将其其他新的数据类型作为新的枚举来添加。

[0063] 对于数据归档条件,定义了两种枚举类型元素(即,在可变点 518 处的“EvalOpEnum”及在表达式元素 522 处的“OpEnum”)以支持灵活性及可扩展性。

[0064] 下文为归档具有相应关系的多表及文件的简单但典型的场景 :

[0065] 1. “分析应用内容” :分析应用以理解需要归档哪些数据对象及目的地为何处。

[0066] 2. “定义归档模型” :存在需要归档的 3 个数据对象,即,诸如,名称为“table1”的表、称为“table2”的表及称为“file1”的文件。table1 及 table2 中的数据可以是关系数

据。file1 的一“绝对路径”属性将“file1 类型”文件与 table2 中的数据相关联。接着创建一数据归档规约模型。

[0067] 3. “变换”: 下一步骤为将以上数据归档模型变换为基于 XML 的规约, 且基于规约以真实代码 (real code) 产生一数据归档应用。

[0068] 4. “部署”: 在最后步骤中, 可将该数据归档应用部署至相应归档引擎 (工具) 中。接着, 该数据归档应用推出且服务于真实企业应用。

[0069] 图 6 展示本发明的优选实施例中的灵活、通用的数据归档系统 600 中的变换及部署流程。系统 600 具有源储存库 602 及归档储存库 604。源储存库 602 具有数据库 DB606 及文件归档 608, 且归档储存库 604 具有数据库 DB610 及文件归档 612。该过程包括数据归档建模器 614 及数据归档工具 616。数据归档建模器 614 通过自源储存库 602 接收分析信号 620 而辅助分析过程。它执行变换过程, 且将变换消息 622 发送至归档规约及归档代码 618。接着通过部署消息 624 将其部署至数据归档工具 616。数据归档工具 616 将选择或删除指令传送至源储存库 602。数据归档工具 616 也将保存 / 更新或清除指令传送至归档储存库 604 以指示是否应保存或者清除数据。

[0070] 尽管在本文中本发明的实施例展示并描述为归档数据的系统及方法, 但本发明的实施例可包含其他系统及方法。尽管本发明的实施例也提供了用于归档数据的商业应用, 但应理解, 本发明的实施例进一步提供各种替代实施例。举例而言, 在一个实施例中, 本发明提供计算机可读 / 可用媒体, 其包括用以使系统的计算机基础结构能够允许归档数据的计算机程序代码。就此而言, 计算机可读 / 可用媒体包括实施本发明的实施例的各过程步骤中的每一个的程序代码。

[0071] 应理解, 术语计算机可读媒体或计算机可用媒体包含程序代码的任何类型的实体实施例中的一个或多个。具体而言, 计算机可读 / 可用媒体可包含程序代码, 该程序代码体现于一个或多个便携式存储制品 (例如, 光盘、磁盘、磁带等) 上、体现于计算设备的一个或多个数据存储部分 (诸如, 存储器及 / 或存储系统 (例如, 固定式磁盘、只读存储器、随机存取存储器、高速缓存等)) 上, 及 / 或作为 (例如, 在程序代码的有线 / 无线电子分发期间) 经由网络传输的数据信号 (例如, 传播信号)。

[0072] 在另一实施例中, 本发明提供了一种用于归档数据的计算机实施的方法。在本发明的此实施例中, 可提供计算机化基础结构, 且可提供一或多个系统以用于执行可获得且部署至计算机化基础结构的本发明的过程步骤。

[0073] 用于本发明的实施例的系统的部署的过程可包含以下一个或多个过程步骤: 将程序代码从计算机可读媒体安装于计算设备 (诸如, 计算机系统) 上, 将一或多个计算设备添加至计算机基础结构, 及并入及 / 或修改该计算机基础结构的一个或多个现存系统以使计算机化基础结构能够执行本发明的实施例的过程步骤。

[0074] 如本文中所使用的, 应理解, 术语“程序代码”与“计算机程序代码”同义, 且意谓指令集的以任何语言、代码或符号的任何表达, 这些指令集旨在使具有信息处理能力的计算设备能够直接或在以下两项操作中的任一个或两个之后执行特定功能: (a) 转换成另一语言、代码或符号; 及 / 或 (b) 以不同材料形式再现。就此而言, 程序代码可体现为以下各项中的一个或多个: 应用 / 软件程序、组件软件 / 函数库、操作系统、用于特定计算及 / 或 I/O 设备的基本 I/O 系统 / 驱动程序等等。

[0075] 已出于说明及描述的目的而呈现本发明的各方面的前文描述。其不旨在为详尽的或将本发明限于所揭示的精确形式,且明显地,许多修改及变化是可能的。对于本领域的技术人员而言为显而易见的这些修改及变化旨在包括于由随附权利要求所界定的本发明的范围内。

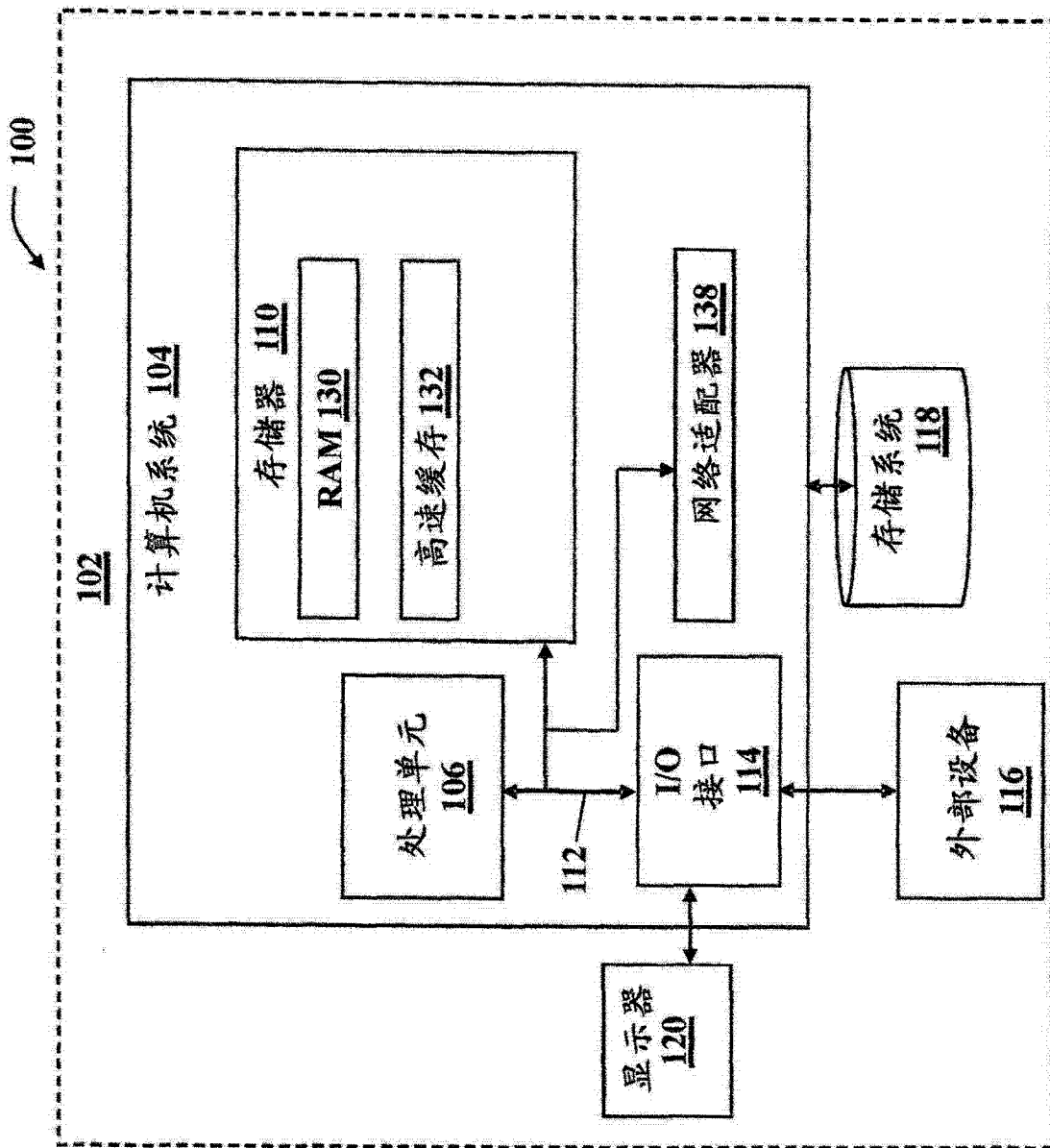


图 1

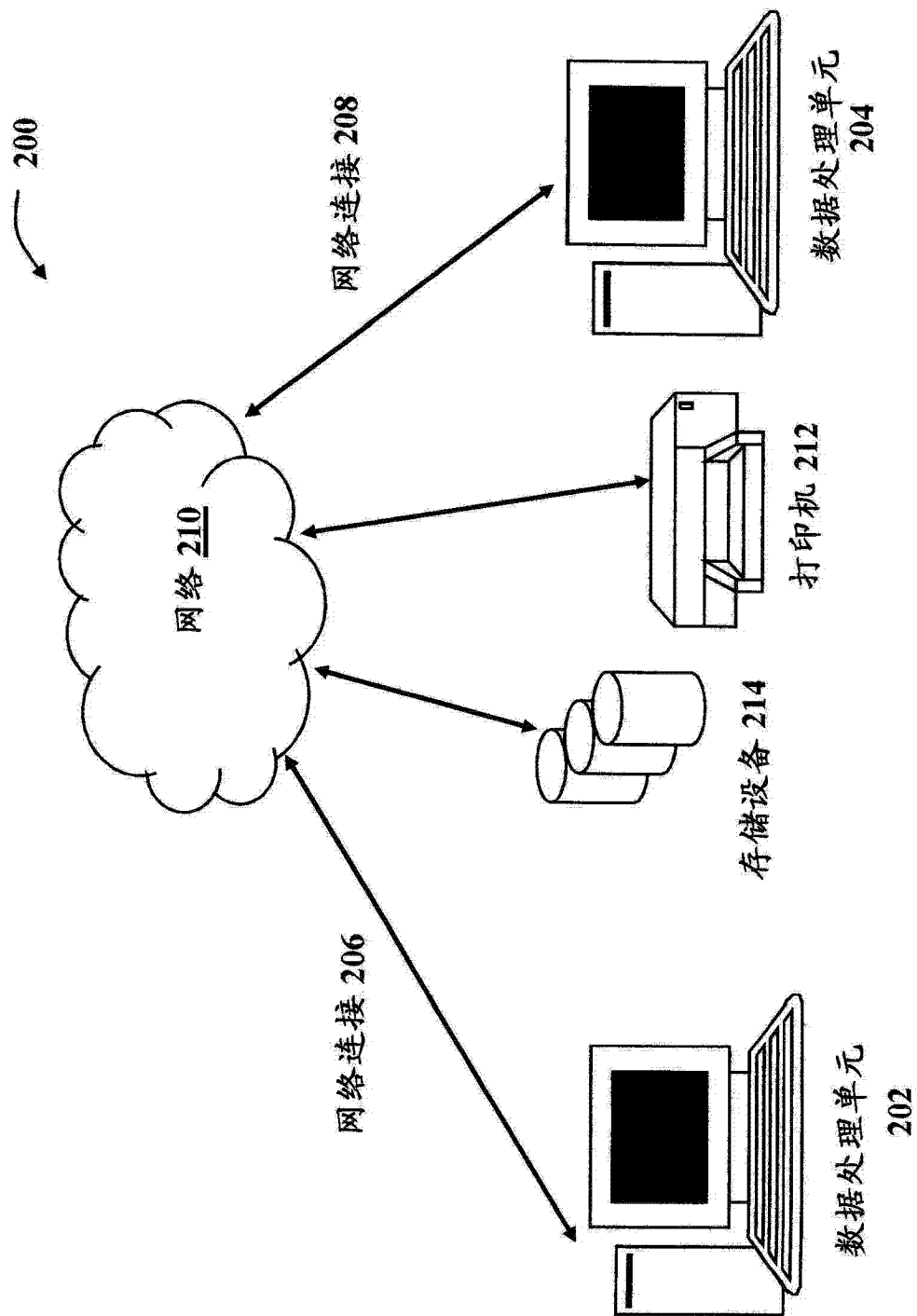


图 2

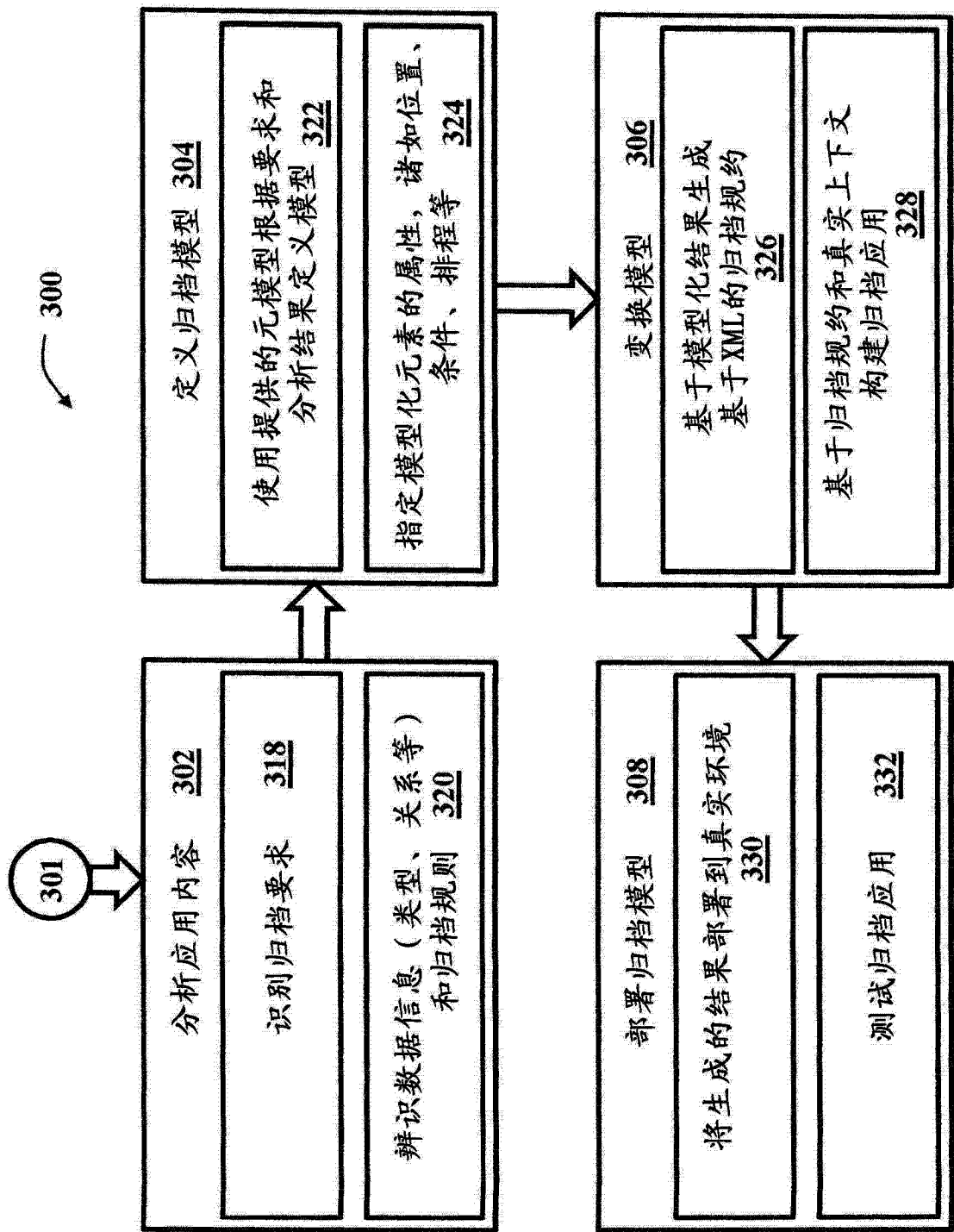


图 3

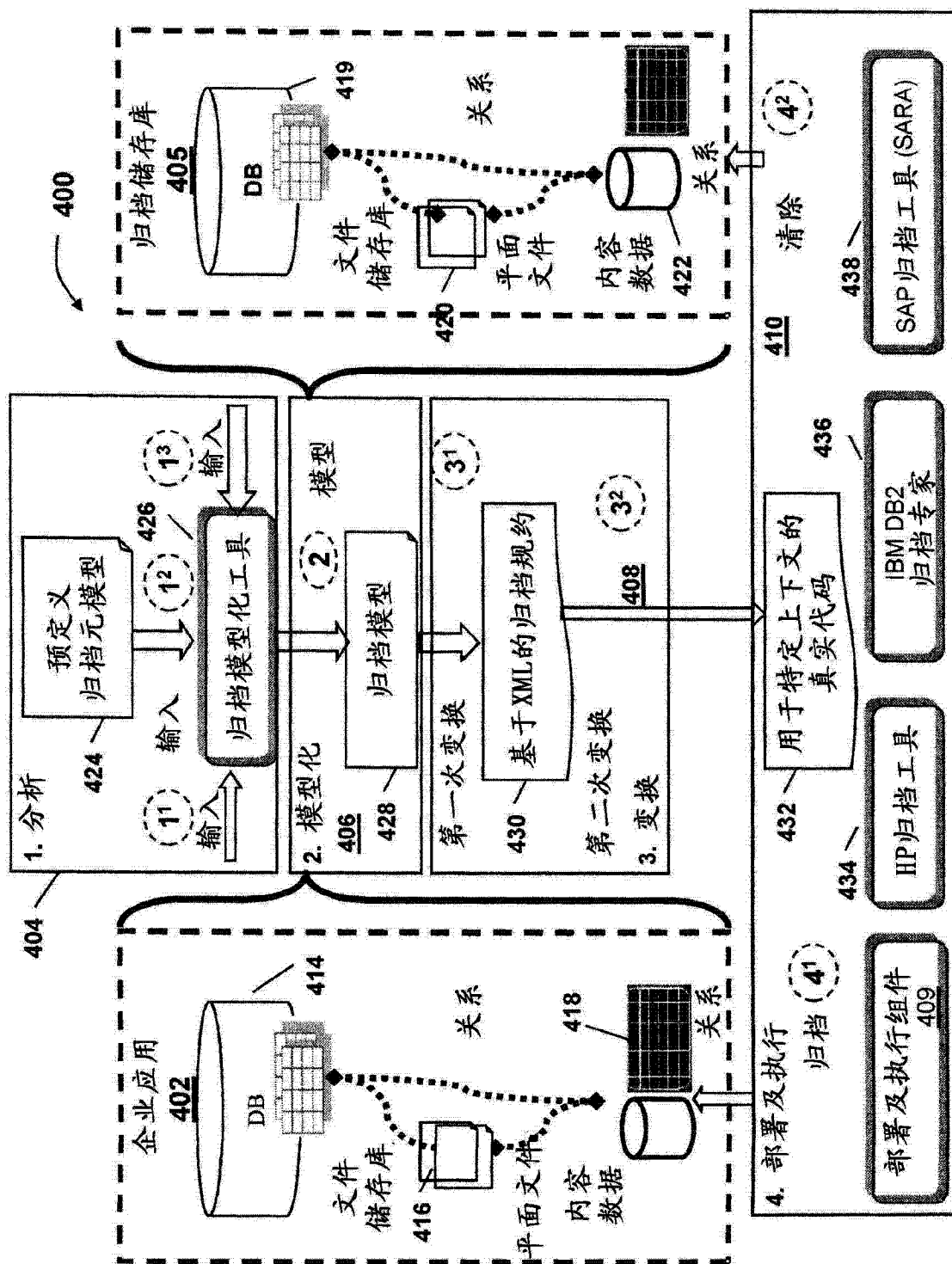


图 4

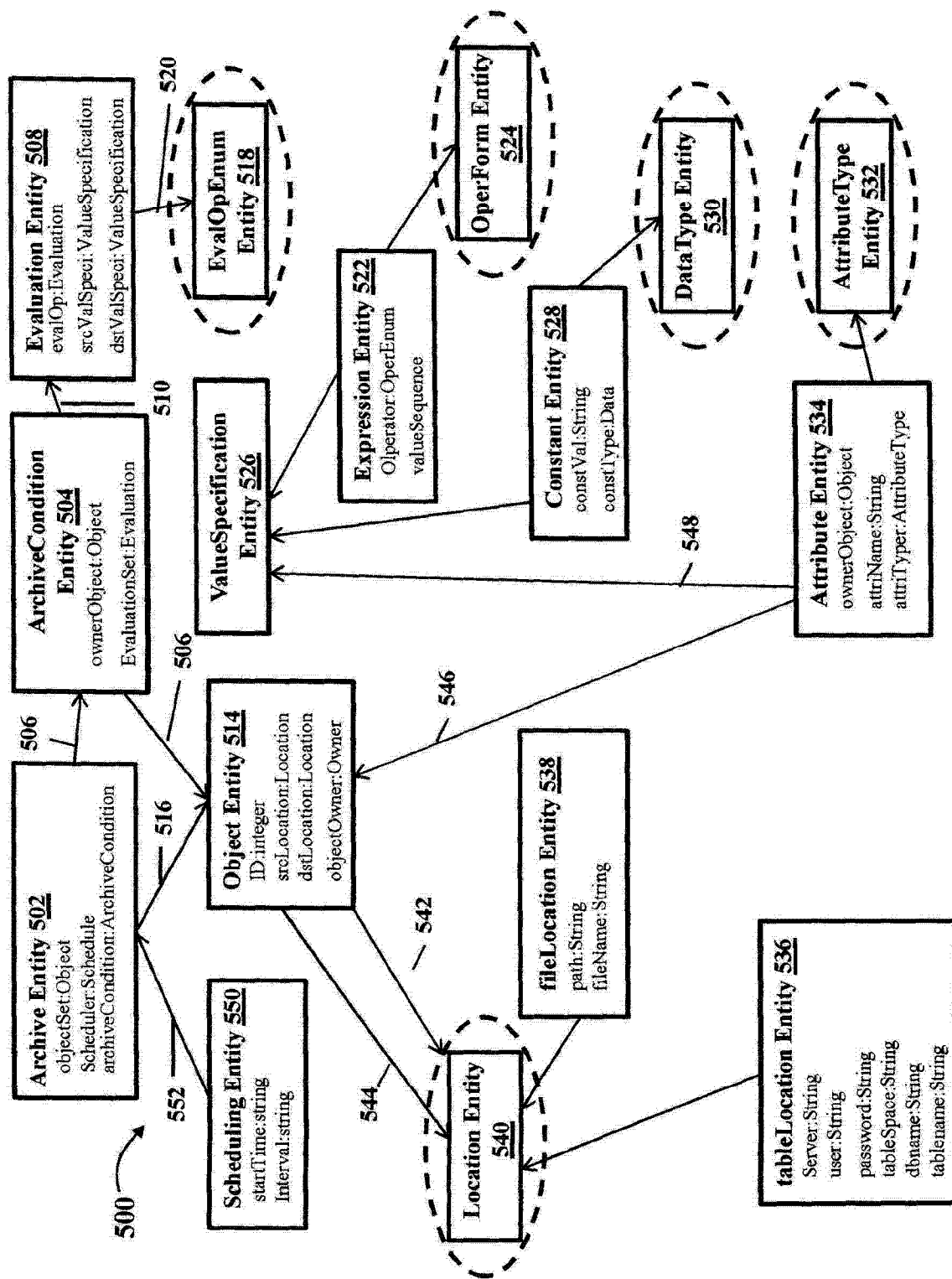


图 5

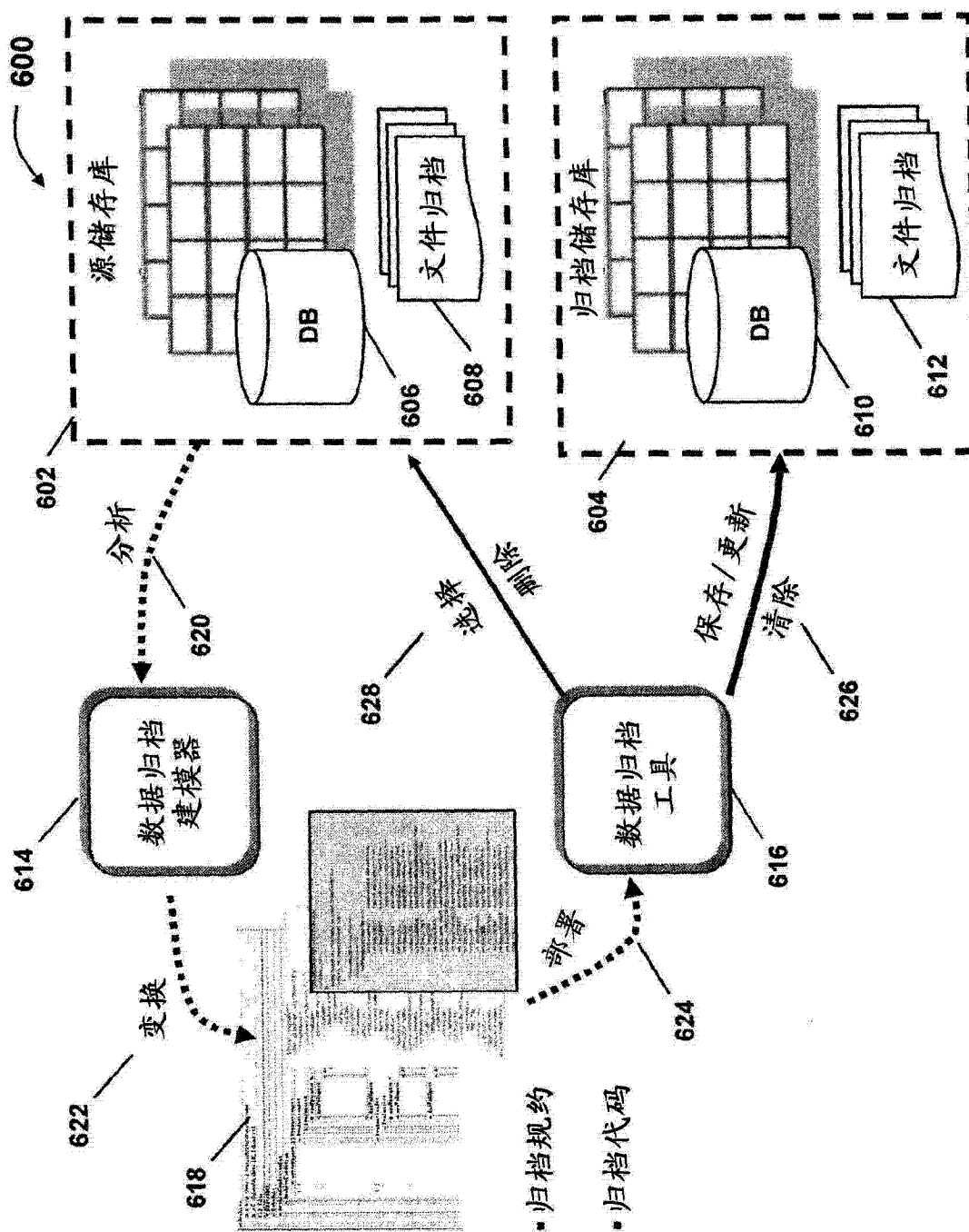


图 6