

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/14 (2006.01)



[12] 发明专利说明书

专利号 ZL 200510126788.1

[45] 授权公告日 2009 年 5 月 20 日

[11] 授权公告号 CN 100489799C

[22] 申请日 2005.11.18

[21] 申请号 200510126788.1

[30] 优先权

[32] 2004.12.20 [33] US [31] 11/018,914

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 B·M·琼斯 C·L·刘

C·B·罗斯切勒

R·R·麦克考格 S·A·威拉隆

S·-P·B·吴

[56] 参考文献

US6567826B1 2003.5.20

EP1093063A2 2001.4.18

WO02/19075A2 2002.3.7

US6243831B1 2001.6.5

审查员 冯婷霆

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 张政权

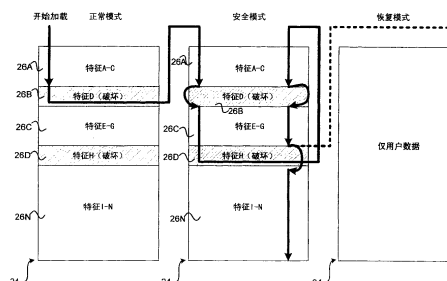
权利要求书 4 页 说明书 9 页 附图 4 页

[54] 发明名称

用于加载数据文件的内容的方法和计算机可读介质

[57] 摘要

提供一种用于加载电子数据文件的内容的方法和计算机可读介质。依照该方法、标识数据文件的破坏的部分，并试图修复这些部分。如果破坏的部分不能被修复，则对这些部分的加载被跳过。然后，数据文件中未破坏的和已修复的部分被加载到存储器中。如果数据文件的部分不能被修复或跳过，则试图仅加载包含在数据文件中的用户数据。以此方式，即使在严重破坏的情况下，数据文件中包含的用户数据也可被加载。



1. 一种用于加载包括一个或多个部分的数据文件的方法，所述方法包括：
试图在第一模式中加载所述数据文件的每一部分，其中，在所述第一模式中，对每一所述部分执行最小完整性检查；
确定在所述第一模式中所述数据文件的一部分是否是不可加载的；以及
响应于确定一部分是不可加载的，试图在第二模式中加载所述数据文件，其中，对每一所述部分执行更广泛的完整性检查，并且其中，对由一“跳过计数器”变量标识的每一不可加载的部分的加载被跳过，其中所述“跳过计数器”变量是用于单独地标识在加载期间应当被跳过的每一部分的数据结构；
其中用户能选择用于加载数据文件的模式，并且一“模式”变量被用于跟踪当前的加载模式。
2. 如权利要求1所述的方法，其特征在于，当在所述第二模式中试图加载所述数据文件时，确定不可加载部分是否可被修复，并且响应于确定不可加载部分可被修复，修复所述不可加载部分并加载所修复的部分。
3. 如权利要求2所述的方法，其特征在于，还包括响应于确定不可加载部分不可被修复，跳过对所述不可加载部分的加载。
4. 如权利要求3所述的方法，其特征在于，还包括：
确定对不可加载部分的加载是否可被跳过；以及
响应于确定对不可加载部分的加载不能被跳过，试图在第三模式中加载所述数据文件，其中，在所述第三模式中，仅加载所述数据文件中对应于用户数据的部分。
5. 如权利要求4所述的方法，其特征在于，所述用户数据包括用户输入的文本数据。
6. 如权利要求4所述的方法，其特征在于，所述用户数据包括用户输入的数字数据。
7. 一种用于加载包括一个或多个部分的数据文件的方法，所述方法包括：
提供用于加载具有一个或多个部分的数据文件的第一加载模式，其中，当在所述第一加载模式中加载时，对每一所述部分执行最小完整性检查；
提供用于加载所述数据文件的各部分的第二加载模式，其中，对每一所述部

分执行更广泛的完整性检查，并且其中，对由一“跳过计数器”变量标识的每一不可加载的部分的加载被跳过，其中所述“跳过计数器”变量是用于单独地标识在加载期间应当被跳过的每一部分的数据结构；

在所述第一加载模式中开始加载数据文件；以及

当在所述第一加载模式中操作时，确定所述数据文件的一部分是否是不可加载的，并且响应于确定一部分不可加载，切换到所述第二加载模式；

其中用户能选择用于加载数据文件的模式，并且一“模式”变量被用于跟踪当前的加载模式。

8. 如权利要求 7 所述的方法，其特征在于，在所述第二加载模式中，确定不可加载部分是否可被修复，并且响应于不可加载部分可被修复，修复所述不可加载部分并加载所修复的部分。

9. 如权利要求 8 所述的方法，其特征在于，在所述第二加载模式中，响应于确定不可加载部分不可被修复，跳过对所述不可加载部分的加载。

10. 如权利要求 9 所述的方法，其特征在于，还包括：

提供第三加载模式，其中，在所述第三加载模式中，仅加载所述数据文件中对应于用户数据的部分；

当在所述第二加载模式中操作时，确定对不可加载部分的加载是否可被跳过；以及

响应于确定对不可加载部分的加载不能被跳过，切换到所述第三加载模式。

11. 如权利要求 10 所述的方法，其特征在于，所述用户数据包括用户输入的文本数据。

12. 如权利要求 11 所述的方法，其特征在于，所述用户数据包括用户输入的数字数据。

13. 一种用于加载包括一个或多个部分的数据文件的方法，所述方法包括：提供第一加载模式、第二加载模式和第三加载模式；

在所述第一加载模式中开始加载具有一个或多个数据记录的数据文件，其中，对所述数据记录执行最小完整性检查；

当在所述第一加载模式中加载所述数据文件时，确定所述数据记录之一是否缺少或被破坏；

响应于在所述第一加载模式中确定所述数据记录之一是不可加载的，切换到所述第二加载模式；

在所述第二加载模式中试图从头开始加载所述数据文件，其中，在所述第二加载模式中，对所述数据记录执行与所述第一加载模式相对而言额外的完整性检查，并且其中，对任何由一“跳过计数器”变量标识的不可加载的数据记录的加载被跳过，其中所述“跳过计数器”变量是用于单独地标识在加载期间应当被跳过的每一数据记录的数据结构；

当在所述第二加载模式中加载所述数据文件时确定对所述数据记录之一的加载是否不可跳过；以及

响应于在所述第二加载模式中加载时确定对所述数据记录之一的加载不可跳过，切换到所述第三加载模式，其中，仅加载所述数据文件中储存的用户数据；

其中用户能选择用于加载数据文件的模式，并且一“模式”变量被用于跟踪当前的加载模式。

14. 如权利要求 13 所述的方法，其特征在于，如果数据记录包括非预期的数据值，则所述数据记录是不可加载的。

15. 如权利要求 14 所述的方法，其特征在于，如果数据记录是缺少的数据，则所述数据记录是不可加载的。

16. 如权利要求 15 所述的方法，其特征在于，如果数据记录包括无效的可扩展标记语言，则所述数据记录是不可加载的。

17. 如权利要求 16 所述的方法，其特征在于，如果应用程序对数据记录的加载导致所述应用程序崩溃，则所述数据记录是不可加载的。

18. 如权利要求 17 所述的方法，其特征在于，如果数据记录是被破坏的，则所述数据记录是不可加载的。

19. 如权利要求 18 所述的方法，其特征在于，如果数据记录是缺少的，则所述数据记录是不可加载的。

20. 一种用于加载包括一个或多个数据记录的数据文件的方法，所述方法包括：

提供第一加载模式、第二加载模式和第三加载模式；

试图在所述第一加载模式中加载所述数据文件，其中，当在所述第一加载模式中加载时，对所述数据记录执行最小完整性检查；

当在所述第一加载模式中加载所述数据文件时，确定所述数据记录之一是否是不可加载的；

响应于在所述第一加载模式中确定所述数据记录之一是不可加载的，切换到

所述第二加载模式；

在所述第二加载模式中试图从头开始加载所述数据文件，其中，在所述第二加载模式中，对所述数据记录执行与所述第一加载模式相对而言额外的完整性检查，并且其中，对任何由一“跳过计数器”变量标识的不可加载的数据记录的加载被跳过，其中所述“跳过计数器”变量是用于单独地标识在加载期间应当被跳过的每一数据记录的数据结构；

当在所述第二加载模式中加载所述数据文件时确定对所述数据记录之一的加载是否不可跳过；以及

响应于当在所述第二加载模式中加载时确定对所述数据记录之一的加载不可跳过，切换到所述第三加载模式，其中，仅加载所述数据文件中储存的用户数据；

其中用户能选择用于加载数据文件的模式，并且一“模式”变量被用于跟踪当前的加载模式。

21. 如权利要求 20 所述的方法，其特征在于，如果数据记录包括非预期的数据值、是缺少的数据、包括无效可扩展标记语言、当应用程序加载所述数据记录时导致应用程序崩溃、是被破坏的、或者如果所述数据记录是缺少的，则所述数据记录是不可加载的。

用于加载数据文件的内容的方法和计算机可读介质

相关申请的交叉引用

本申请涉及共同提交的美国专利申请第 11/018,916 号（代理案卷号 60001.0465US01），该申请名为“Method and Computer-Readable Medium For Verifying and Saving an Electronic Document（用于验证和保存电子文档的方法和计算机可读介质）”，通过引用明确结合于此。

背景技术

计算机在当今的社会得到普遍深入的使用，以执行各种各样的任务以及用于娱乐目的。例如，当今的计算机用于游戏、通信、研究和实际上无限的各种其它应用。商业和个人对计算机的最常见使用之一是对电子和印刷文档的创建。存在用于创建所有种类电子文档的计算机应用程序，这些电子文档包括电子表格、演示、文字处理文档、诸如图表和数字图像等图形文档、计算机辅助设计文档以及许多其它类型的电子文档。

电子文档通常包括非常重要的内容。此外，在许多情况下，电子文档的内容如果丢失，将很难或不可能重新创建。例如，通常会创建高度复杂的法律、商务、营销和技术文档，如果储存这些文档的数据文件被损坏或破坏，则这些文档不能被容易地重新创建。即使在文档的内容可以被容易地重新创建的情况下，用户丢失其数据的甚至一小部分也将是非常沮丧的，因此，保护电子文档中包含的数据不被破坏和损坏是非常重要的。

现代计算机系统包括检错和其它机制，以保护数据文件不受无意的破坏或损坏。不幸的是，即使适当地有了这些机制，其中储存了电子文档的数据文件被破坏也是相当常见的。破坏可在数据文件被保存到大容量存储设备时发生，或者可作为有故障的存储设备或控制器硬件的结果而发生。数据文件也可在网络传输期间或因许多其它类型的事件的发生而被破坏。由于任何数量数据的丢失对于用户都可能是受挫的，并且由于重新创建被破坏的文档所需的时间和努力通常是非常高的，因此重要的是从破坏的数据文件中恢复尽可能多的数据。

本发明的各实施例正是相对于这些和其它考虑事项而做出的。

发明内容

依照本发明，以上和其它问题由用于加载电子数据文件的内容的方法和计算机可读介质克服。通过使用本发明的各实施例，在加载数据文件期间，标识文件的被破坏部分或记录，并且试图修复这些部分。如果破坏的部分不能被修复，则跳过对这些部分的加载。数据文件的未破坏以及修复的部分然后被加载到存储器中。如果数据文件的各部分不能被修复或跳过，则还可试图仅加载包含在该数据文件中的用户数据。以此方式，即使在对数据文件的剩余部分有严重破坏的情况下，也可加载包含在数据文件中的用户数据。

依照本发明的一方面，提供了一种用于加载包括一个或多个部分的数据文件的方法。依照该方法，提供了多个加载模式。在“正常”加载模式中，试图以正常的方式加载数据文件的每一部分。该正常加载模式包括对数据文件的每一部分的最小完整性检查，使得文件可被快速加载。如果在正常模式中遇到数据文件中缺少或破坏的部分，则使用被称为“安全”加载模式的第二模式来试图加载该数据文件的各部分。如果数据文件的一部分导致试图加载它的应用程序中的错误或崩溃，如果该部分包括非预期数据值，如果该部分是缺少的数据，如果该部分包括无效记录或无效可扩展标记语言（“XML”），以及对于其它原因，则数据文件的该部分可被认为是破坏的，因此是不可加载的。

在安全加载模式中，对数据文件的每一部分执行广泛的完整性检查。在安全加载模式中，也试图修复数据文件的破坏部分。然后加载可被修复的任何部分。如果在安全加载模式中遇到数据文件中缺少或破坏的部分，并且该部分也是不可修复的，则跳过对该不可修复部分的加载。如果遇到数据文件中不可修复且对其不可跳过加载的部分，则使用称为“恢复”加载模式的第三加载模式来试图加载该数据文件的各部分。

在恢复加载模式中，仅加载数据文件中包括用户数据的部分。例如，用户数据可包括用户输入的文本数据或数字数据。作为一个示例，如果数据文件包含电子表格，则在恢复加载模式中试图仅加载包含在电子表格单元格中的数据。在恢复加载模式中不尝试加载可能包含在该数据文件中的其它类型的数据，诸如数据透视表、列表对象、命名范围、自动过滤器、样式、格式以及应用程序或用户偏好。

依照本发明的其它实施例，也提供了一种其上储存计算机可执行指令的计算

机可读介质。当该计算机可执行指令由计算机执行时，它们使该计算机提供用于加载具有一个或多个部分的数据文件的第一加载模式。在第一加载模式中，当加载数据文件的各部分时对它们执行最小完整性检查。该计算机可执行指令还使计算机提供用于加载数据文件的第二加载模式，其中对数据文件的各部分执行比第一加载模式中更广泛的完整性检查。在第二模式中，也试图修复不可加载的部分。此外，在第二加载模式中，跳过对不可加载部分的加载。

该计算机可执行指令还使计算机在第一加载模式中开始加载数据文件。如果数据文件的一部分被确定为在第一加载模式中是不可加载的，则计算机切换到第二加载模式，并试图以该加载模式加载该数据文件。如果在第二加载模式中遇到可以被修复的不可加载部分，则修复并加载该不可加载部分。如果该不可加载部分不能被修复，则跳过对该不可加载部分的加载。

依照本发明的一个实施例，该计算机可执行指令也使计算机提供第三加载模式，其中，仅加载数据文件中包括用户数据的各部分。如果在第二加载模式中确定数据文件的一部分是不可加载的且该不可加载部分不能被修复或跳过，则试图以第三加载模式加载数据文件的内容。

本发明可以被实现为计算机进程、计算系统或诸如计算机程序产品或计算机可读介质等制品。计算机程序产品可以是计算机系统可读且编码了用于执行计算机进程的指令的计算机程序的计算机存储介质。计算机程序产品也可以是计算机系统可读且编码了用于执行计算机进程的指令的计算机程序的载波上的传播信号。

通过阅读以下详细描述并观察附图，可以清楚特性化本发明的这些和各种其它特征以及优点。

附图说明

图 1 是示出在本发明的各实施例中使用的且由本发明的各实施例提供的计算机系统的计算机系统体系结构图；

图 2 是示出由本发明的实施例提供的数据文件和各种加载模式的各方面的框图；以及

图 3A-3B 是示出依照本发明的各实施例用于加载数据文件的说明性过程的流程图。

具体实施方式

现在参考附图，将描述本发明的各方面，附图中，相同的标号表示相同的元素。具体地，图1和对应的讨论旨在提供对其中可实现本发明的各实施例的合适的计算环境的简要概括描述。尽管本发明将在个人计算机的操作系统上执行的程序模块的通用上下文中描述，然而本领域的技术人员可以认识到，本发明也可结合其它类型的计算机系统和程序模块来实现。

一般而言，程序模块包括例程、程序、组件、数据结构以及其它类型的结构，它们执行特定的任务或实现特定的抽象数据类型。此外，本领域的技术人员将理解，本发明可以用其它计算机系统配置来实施，包括手持式设备、多处理器系统、基于微处理器或可编程消费者电子产品、小型机、大型机等等。本发明也可在分布式计算环境中实施，其中任务由通过通信网络链接的远程处理设备来执行。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备中。

现在参考图1，将描述用于实施本发明的各实施例的计算机2的说明性计算机体系结构。图1所示的计算机体系结构示出了常规台式机或膝上型计算机，包括中央处理单元5（“CPU”）、系统存储器7（包括随机存取存储器9（“RAM”）和只读存储器（“ROM”）11）、以及将存储器耦合至CPU5的系统总线12。包含例如在启动时有助于在计算机内的元件之间传输信息的基本例程的基本输入/输出系统储存在ROM11中。计算机2还包括大容量存储设备14，用于储存操作系统16、应用程序以及其它程序模块，这些将在下文更详细描述。

大容量存储设备14通过连接至总线12的大容量存储控制器（未示出）连接到CPU5。大容量存储设备14及其相关联的计算机可读介质为计算机2提供了非易失性存储。尽管此处包含的计算机可读介质的描述指的是大容量存储设备，如硬盘或CD-ROM驱动器，然而本领域的技术人员应当理解，计算机可读介质可以是可由计算机2访问的任何可用介质。

作为示例而非局限，计算机可读介质可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、EPROM、EEPROM、闪存或其它固态存储器技术、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2访问的任一其它介质。

依照本发明的各实施例，计算机2可以使用通过诸如因特网等网络18到远程

计算机的逻辑连接在网络化环境中操作。计算机 2 可以通过连接到总线 12 的网络接口单元 20 连接到网络 18。应当理解,网络接口单元 20 也可用于连接到其它类型的网络和远程计算机系统。个人计算机 2 也可包括用于接收和处理来自包括键盘、鼠标或输入笔(未示出)的多个其它设备的输入的输入/输出控制器 22。类似地,输入/输出控制器 22 可向显示屏幕、打印机或其它类型的输出设备提供输出。

如上所述,多个程序模块和数据文件可储存在计算机 2 的大容量存储设备 14 和 RAM 9 中,包括适用于控制网络化个人计算机的操作的操作系统 16,如来自华盛顿州雷蒙德市微软公司的 WINDOWS XP 操作系统。大容量存储设备 14 和 RAM 9 也可储存一个或多个应用程序。具体地,大容量存储设备 14 和 RAM 9 可储存电子表格应用程序 10。如本领域的技术人员已知的,电子表格应用程序 10 可用于提供用于创建和编辑电子表格的功能。

依照本发明的一个实施例,电子表格应用程序 10 包括来自微软公司的 EXCEL 电子表格应用程序。然而,应当理解,可使用来自其它制造商的其它电子表格应用程序来实施本发明的各方面。还应当理解,尽管此处所描述的本发明的实施例是在电子表格应用程序的环境中呈现的,然而本发明可用于从数据文件加载数据的任何其它类型的应用程序。例如,此处所描述的本发明的实施例可在文字处理应用程序、演示应用程序、绘图或计算机辅助设计应用程序或数据库应用程序中使用。

结合电子表格文档的创建和编辑,电子表格应用程序 10 提供了用于将数据文件 24 中的电子表格文档保存到大容量存储设备 14 的功能。数据文件 24 包含表示电子表格的各方面的数据,诸如包括电子表格单元格的内容的用户数据、应用程序偏好、格式信息、以及对应于由电子表格应用程序 10 提供的各种特征的其它数据。关于数据文件 24 的结构和其它细节将在下文参考图 2 提供。

依照本发明的各实施例,电子表格应用程序 10 提供了用于通过将数据文件 24 从大容量存储设备 14 加载到计算机 2 的随机存取存储器 9 来恢复电子表格文档的内容的功能。如下文将参考图 2-3B 更详细描述,电子表格应用程序 10 利用了一种用于加载数据文件 24 的方法,该方法解决了数据文件 24 中的破坏的可能性,并且试图即使在数据文件 24 被破坏的情况下也将从数据文件 24 加载的数据量最大化。

现在转向图 2,将提供关于电子表格应用程序 10 使用的数据文件 24 的结构和加载机制的操作的其它细节。如图 2 所示,数据文件 24 被细分成多个部分 26A-26N。每一部分 26A-26N 用于储存涉及电子表格应用程序 10 支持的一个或多个特征的信

息。此外，关于不同但相关特征的信息可被储存在单个部分 26A-26N 中。例如，如图 2 所示，关于特征 A-C 的数据被储存在部分 26A 中，关于特征 D 的数据被储存在部分 26B 中。关于特征 E-G 的数据被储存在部分 26C 中等等。用户数据可被储存在部分 26A-26N 的任一个中。

如上文简要描述的，如图 2 所示，包含在部分 26A-26N 中的数据可能被破坏。破坏可以在数据文件被保存到大容量存储时发生，或者可作为有故障的存储设备或控制器硬件的结果而发生。在网络传输期间或由于许多其它类型的事件的发生，数据文件也可能被破坏。用于特定部分的数据也可被确定为是缺少的。如果数据文件的一部分导致试图加载该部分的应用程序的错误或崩溃，如果该部分包括非预期的数据值，如果该部分是缺少的数据，如果该部分包括无效记录或无效可扩展标记语言（“XML”），以及由于其它原因，数据文件的该部分可被认为是破坏的，因此是不可加载的。在图 2 所示的说明性数据文件 24 中，部分 26B 和 26D 已被破坏。

如此处所描述的，尽管部分 26B 和 26D 被破坏，数据文件 24 的各部分仍可由电子表格应用程序 10 加载。图 2 也示出了利用说明性数据文件 24 的这一加载过程。具体地，电子表格应用程序 10 开始以正常加载方式加载数据文件 24。在正常加载模式中，对数据文件的各部分 26A-26N 执行最小完整性检查。如果在正常模式中加载时遇到数据文件 24 的被破坏的部分，则电子表格应用程序 10 切换到安全加载模式，并从头开始加载数据文件 24。例如，如图 2 所示，当在正常加载模式中遇到破坏的部分 26B 时，加载模式改为安全加载模式，且加载再一次从数据文件 24 的起始处开始。

在安全加载模式中，对数据文件 24 的各部分 26A-26N 执行与正常加载模式相对而言的其它完整性检查。另外，如果在安全模式中加载时遇到破坏的部分，则试图修复该破坏的部分。如果破坏的部分可以被修复，则加载该部分。如果破坏的部分不能被修复，则跳过对该破坏部分的加载。例如，如图 2 所示，部分 26B 被破坏且不能被修复。因此，对部分 26B 的加载被跳过，且加载部分 26C。

在部分 26C 被加载之后，试图加载部分 26D。然而，如图 2 所示，部分 26D 被破坏。因此，试图修复部分 26D。如果部分 26D 不能被加载，则跳过对部分 26D 的加载，且该过程继续，直到剩余的部分被加载或被跳过。依照本发明的一个实施例，在遇到不可加载的部分且确定该部分是不可修复的时候，对数据文件 24 的加载可以返回到数据文件 24 的开头。这在图 2 中示出。以此方式返回到数据文件 24 的开头允许即使在数据文件 24 中与不可加载部分有关的其它部分未被破坏的情况

下也跳过对该有关部分的加载。

如果在加载数据文件 24 期间遇到不可加载且不可修复的部分，则电子表格应用程序可切换到第三加载模式，称为恢复加载模式。在恢复加载模式中，试图仅从数据文件中加载用户数据。具体地，对于文本文档，试图仅加载文档的文本。对于电子表格文档，试图加载电子表格单元格的内容，包括用户输入的数据、公式、以及公式生成的数据。以此方式，即使数据文件的部分被破坏，某些或全部用户数据也可被恢复和加载。该过程由图 2 中的虚线示出，并且如果部分 26D 被确定为不可加载且不可修复，并且如果没有该部分文件就完全不应当被加载，则执行该过程。关于该过程的其它细节在下文参考图 3A-3B 提供。

现在参考图 3A-3B，将描述示出由电子表格应用程序 10 执行的用于加载数据文件 24 的内容的过程的例程 300。当阅读此处所提出的例程的讨论时，应当理解，本发明的各实施例的逻辑操作被实现为（1）运行在计算系统上的计算机实现的动作或程序模块的序列，和/或（2）计算系统内互连的机器逻辑电路或电路模块。实现是取决于实现本发明的计算系统的性能要求的选择问题。因此，图 3A-3B 中所示以及构成此处所描述的本发明的实施例的逻辑操作以不同的方式被称为操作、结构设备、动作或模块。本领域的技术人员可以认识到，这些操作、结构设备、动作和模块可以用软件、固件、专用数字逻辑以及其任一组合来实现，而不脱离所附权利要求书中所述的本发明的精神和范围。

应当理解，例程 300 在其操作中利用了若干变量。具体地，“模式”变量跟踪当前加载模式。该变量可以被设为“安全”、“正常”或“恢复”。“跳过计数器”变量是用于单独地标识在加载期间应当被跳过的每一部分的数据结构。“要跳过的记录数”变量描述了应当在当前加载试图中跳过的部分的当前数量。“当前记录”变量标识了所处理的数据文件中的当前部分。应当理解，可以使用更多或更少的变量来执行相同的任务。此外，应当理解，图 3 中所示的例程 300 仅表示本发明的一种可能的实现，并且本领域的技术人员可以明白许多其它实现。

例程 300 在操作 302、304 或 306 处开始。具体地，依照本发明的各实施例，可提供用户界面，允许用户选择是正常地加载数据文件（操作 304）、在安全加载模式中加载（操作 302）、还是在恢复加载模式中加载（306）。该用户界面可在用户请求加载文件时呈现给用户。基于用户在用户界面内的选择，例程 300 在操作 302、304 或 306 开始其操作。

如果加载要从安全加载模式开始，则例程 300 在操作 302 开始，其中模式变

量被设为“安全”。例程 300 然后继续到操作 308。如果加载是要从正常加载模式开始，则例程 300 在操作 304 开始，其中模式变量被设为“正常”。例程 300 然后从操作 304 继续到操作 308。如果加载是要在恢复加载模式中开始，则例程在操作 306 开始，其中模式变量被设为“恢复”。如下所述，例程 300 从操作 306 继续到操作 348。

在操作 308，初始化跳过计数器变量，以指示没有记录应当被跳过。例程 300 然后继续到操作 310，其中将当前记录设为数据文件中的第一个记录。也初始化要跳过的记录数变量。在第一遍时，这将要跳过的记录数设为等于零。例程 300 从操作 310 继续到操作 312。

在操作 312，试图在当前模式中加载当前记录。例如，如果模式变量等于“正常”，则对所加载的部分执行最小完整性检查。如果模式变量等于“安全”，则执行额外的完整性检查。例程 300 从操作 312 继续到操作 314，其中确定当前记录是否为不可加载（即，破坏或缺少）。如果当前记录是可加载的，则例程 300 分支到操作 316，其中确定是否还有剩余的记录要被加载。如果还有记录存在，则例程 300 从操作 316 分支到操作 318，其中将当前记录变量设为数据文件中的下一记录。例程 300 然后继续到操作 321，其中加载下一记录。如果在操作 316 确定没有剩下其它记录要加载，则例程 300 分支到操作 320 结束。以此方式，如果不存在破坏或缺少的记录，则在当前模式中加载所有的记录。

应当理解，在本发明的实施例中，某些完整性检查可以在与记录级相对的特征级执行。为执行这一特征级完整性检查，加载特定特征的所有记录。然后，确定该特征的数据是否有效。如果该数据无效，则用要跳过的特征的记录更新跳过数据结构，并且重新加载文件。也可以类似的方式做出文件级一致性检查。

如果在操作 314 确定当前记录是不可加载的，则例程 314 继续到操作 322，其中确定当前模式是否是正常模式。如果当前模式是正常模式，则例程 300 分支到操作 324，其中在跳过计数器变量中标记该记录，以指示标识了数据文件中可能需要被跳过的部分。例程 300 然后继续到操作 326，其中模式变量被设为“安全”。以此方式，当遇到数据文件的不可加载部分时，加载模式从正常切换到安全。例程 300 然后返回到操作 310，其中对数据文件的处理返回到开头。

如果在操作 322 确定当前加载模式不是正常模式，则例程 300 继续到操作 328，其中确定当前加载模式是否是安全模式。因为仅正常或安全加载模式应当是例程 300 的该部分的可能值，因此例程分支到操作 300，其中，如果当前加载模式不是

安全模式，则返回错误。例程 300 然后从操作 330 继续到操作 320 结束。然而，如果在操作 328 确定当前模式是安全模式，则例程 300 继续到操作 332。

在操作 332，试图修复当前记录。在操作 334，确定当前记录是否是可修复的。如果记录是可修复的，则例程 300 分支到操作 336，其中加载当前记录。在操作 336，也更新跳过计数器变量，以指示当前记录的加载不应当被跳过，因为该记录是可修复的。例程 300 从操作 336 分支回到操作 316，其中以上述方式处理数据文件的剩余记录。

如果在操作 334 确定当前记录不能被修复，则例程 300 分支到操作 338。在操作 338，确定要跳过的记录数是否等于零。这将是加载以正常模式启动且遇到第一个破坏记录且该记录是不可修复的情况。在这一情况下，例程 300 分支到操作 340，其中更新跳过计数器变量以指示该记录应当被跳过。例程 300 然后返回到操作 310，其中以上述方式对数据文件的处理返回到开头。

如果在操作 338 确定要跳过的记录数不等于零，则例程 300 继续到操作 342，其中试图跳过对当前记录的加载。在操作 344，确定是否可跳过对当前记录的加载。如果对当前记录的加载可被跳过，则例程 300 分支到操作 346，其中在跳过记录变量中标记该记录。例程然后如上所述继续到操作 316。

如果在操作 344 确定当前记录不能被跳过，则例程 300 继续到操作 306，其中将模式变量设为“恢复”。例程 300 然后继续到操作 348，其中试图在恢复模式中加载数据文件。如上所述，在恢复模式中仅加载用户数据。此外，如果用户数据也被破坏，则试图加载尽可能多的用户数据。例程 300 然后继续到操作 320 结束。

基于以上内容，应当理解，本发明的各实施例包括用于加载数据文件内容的方法、系统、装置和计算机可读介质。以上说明书、示例和数据提供了对本发明的组成部分的制造和使用的完整描述。由于可以在不脱离本发明的精神和范围的情况下做出本发明的许多实施例，因此本发明归于所附权利要求书。

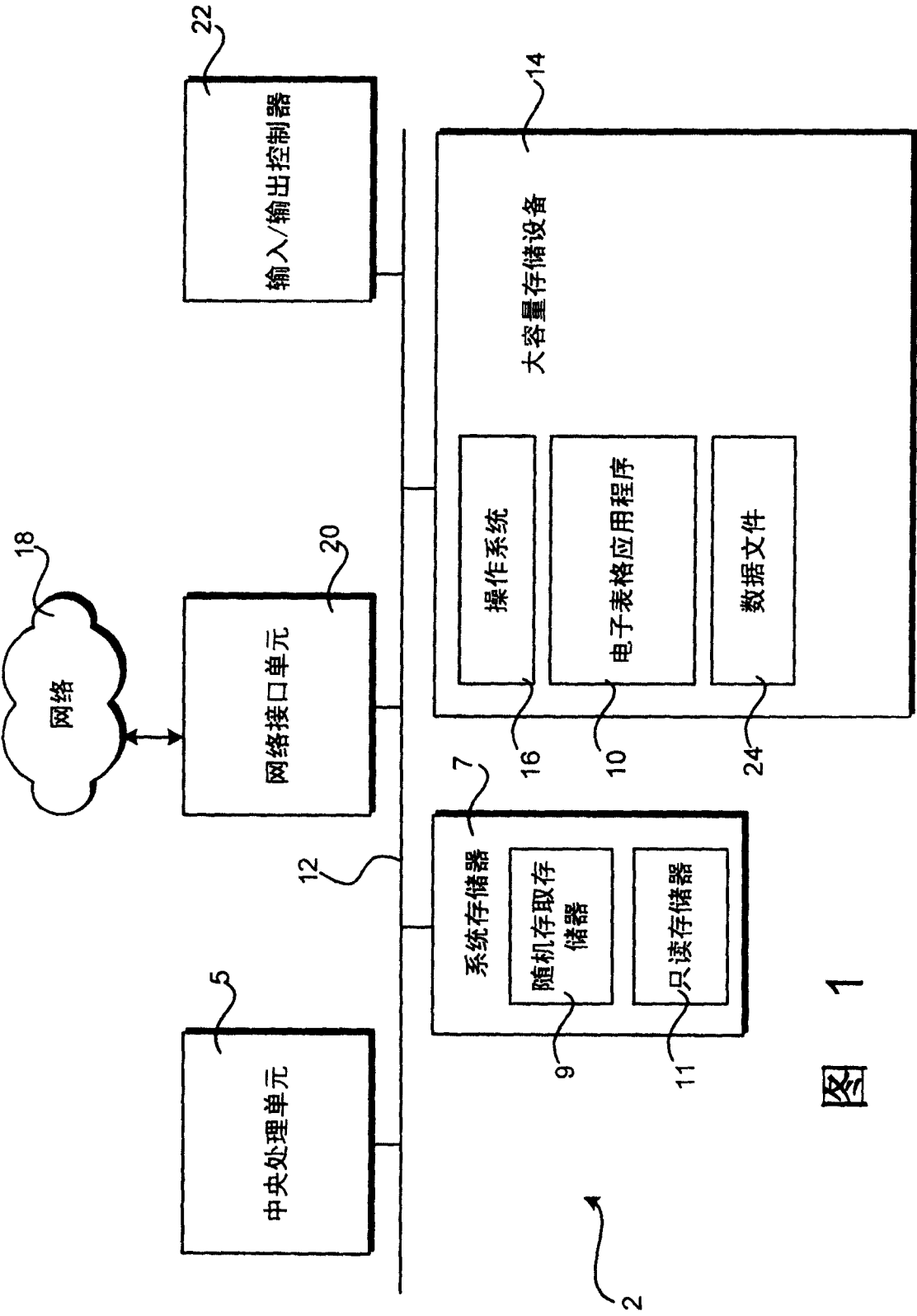


图 1

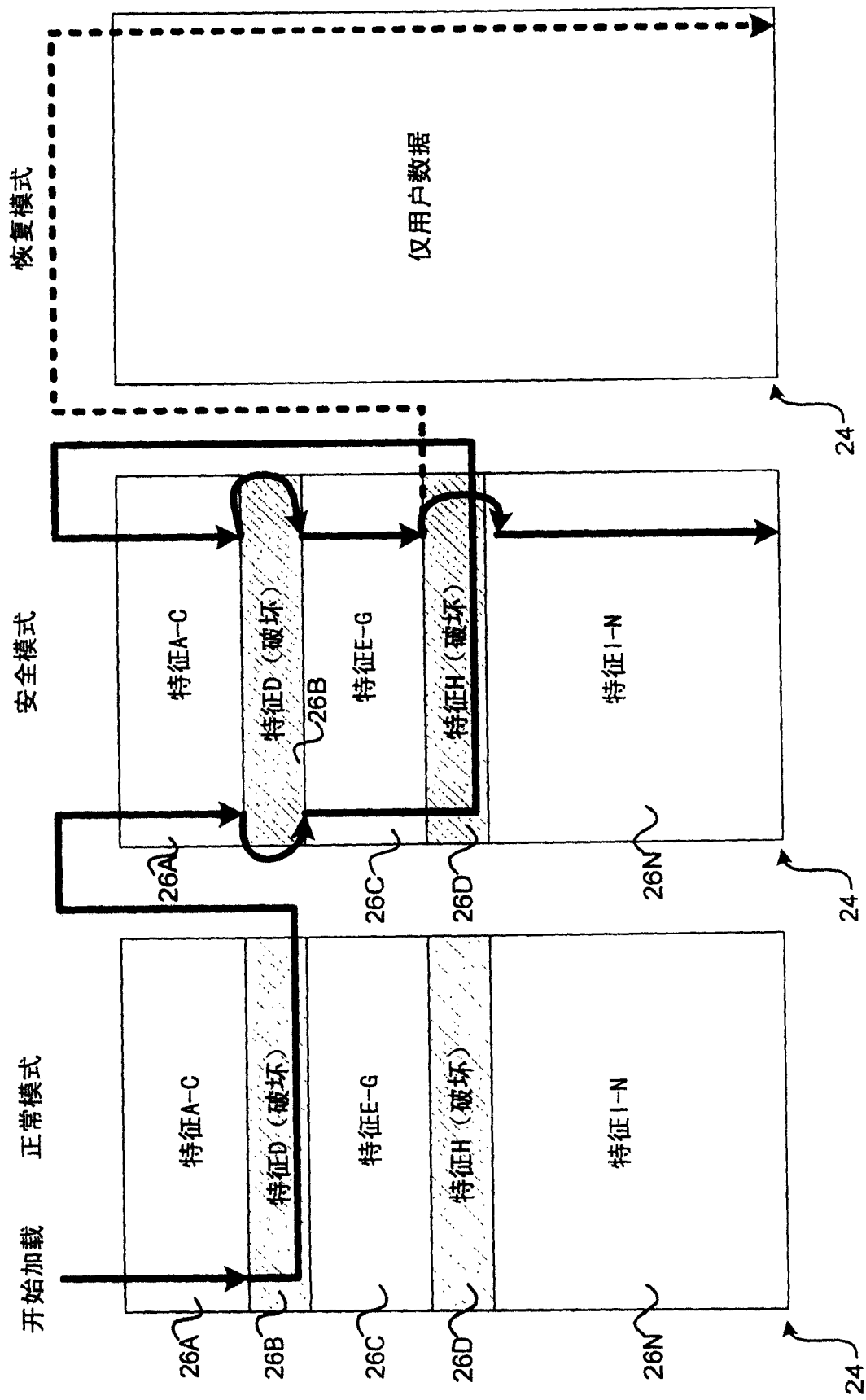


图 2

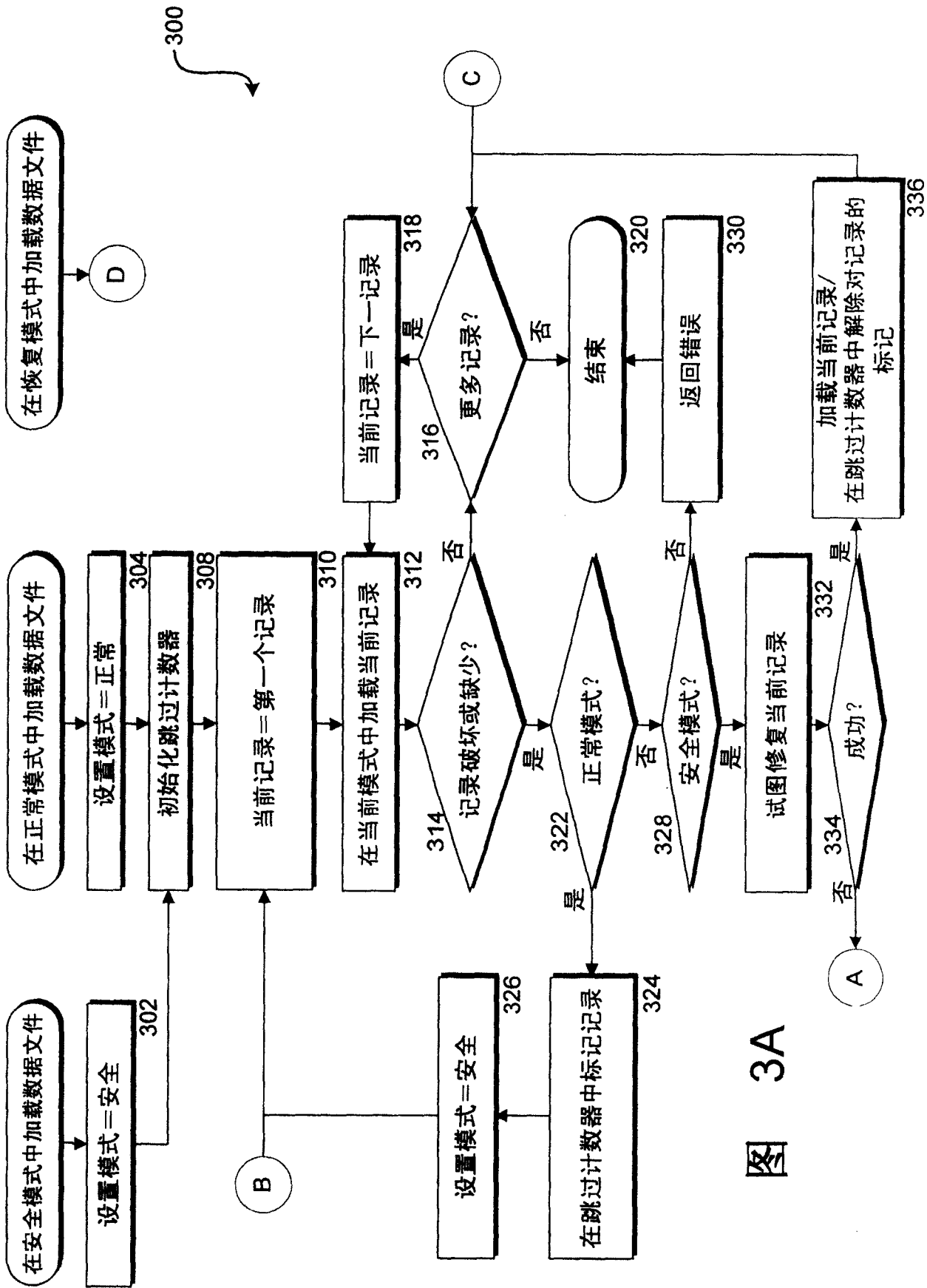


图 3A

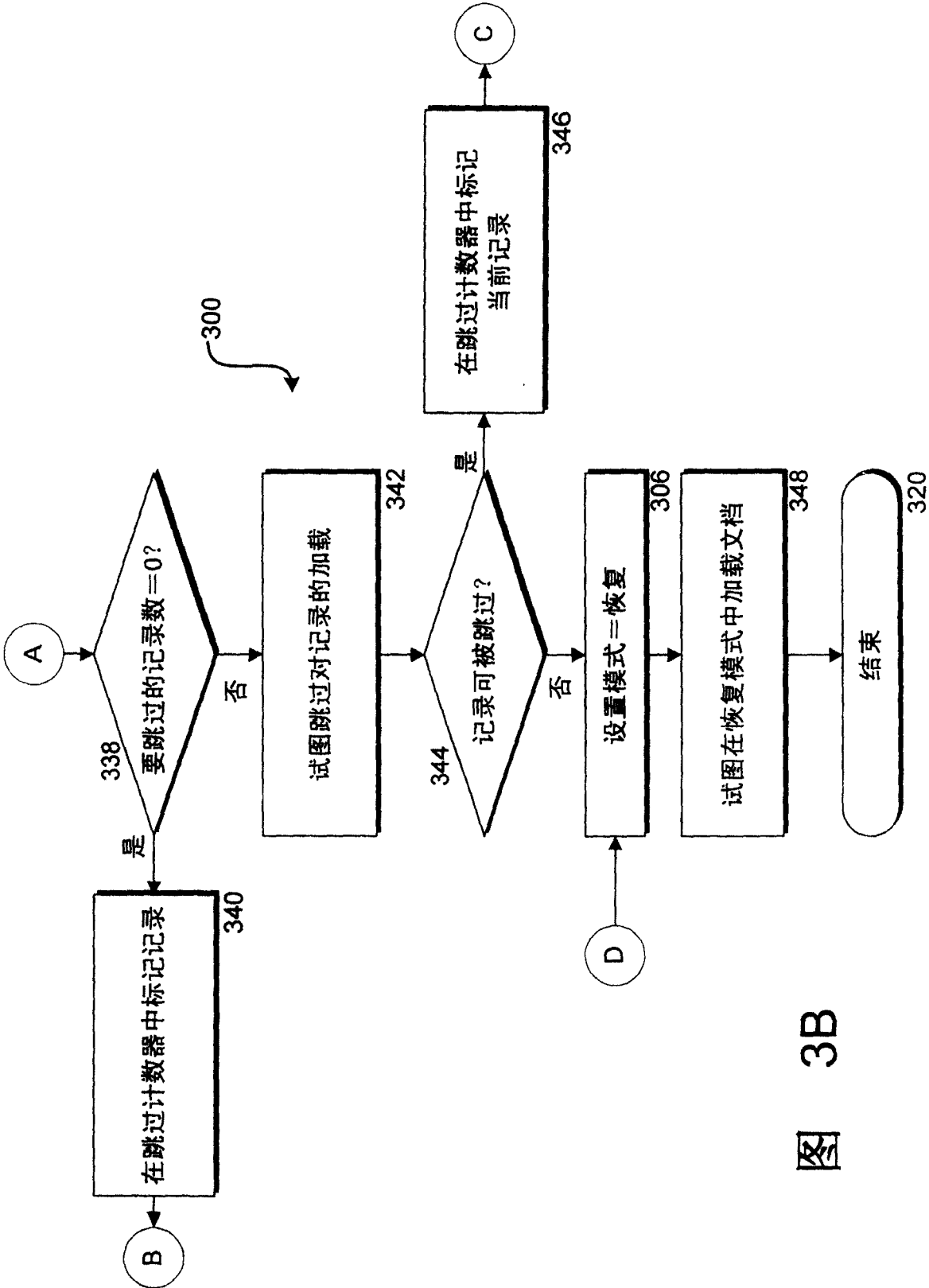


图 3B