



[12] 发明专利申请公开说明书

[21] 申请号 200410043437. X

[43] 公开日 2004 年 12 月 1 日

[11] 公开号 CN 1551000A

[22] 申请日 2004. 4. 28

[21] 申请号 200410043437. X

[30] 优先权

[32] 2003. 5. 2 [33] US [31] 10/429, 116

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 O·罗森布鲁姆 J·D·特纳斯基

V·沙多夫斯基 B·D·曼德斯

[74] 专利代理机构 上海专利商标事务所

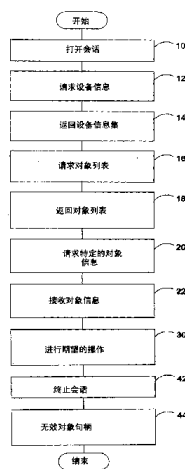
代理人 李家麟

权利要求书 6 页 说明书 26 页 附图 6 页

[54] 发明名称 方便计算设备和多种媒体设备之间的通信的方法和系统

[57] 摘要

提供一种用于控制多种类型的媒体设备的方法和系统。 该方法和系统同样方便了在计算设备和多种类型的媒体设备中之一之间的通信。 该系统包括由计算设备控制的设备恢复模块，用于从由媒体设备控制的一组设备数据集中恢复信息。 设备数据集包括一设备信息数据集，其包括一组对于该种类型的媒体设备特定的属性。 计算设备还包括用于从保存在媒体设备上的一组对象数据集中恢复对象信息的对象恢复工具和控制媒体设备上的对象的控制命令模块。



1. 一种方便在起始器和应答器之间的通信的方法，其中，所述应答器是多种类型的媒体设备中的一种，该方法包括：

向所述应答器提供多个数据集，所述数据集包括由所述应答器支持的一组属性，其中所述属性对于该种类型的媒体设备是特定的；以及

向起始器提供多个控制模块用于搜寻来自所述多个数据集的信息并用于控制所述应答器的对象。

2. 如权利要求 1 所述的方法，还包括向所述应答器提供一设备信息数据集，其中所述设备信息数据集包括对象属性和设备属性的属性码。

3. 如权利要求 2 所述的方法，还包括向所述起始器提供一设备信息恢复模块和一对象信息恢复模块，以便识别由所述属性码识别的属性类型。

4. 如权利要求 2 所述的方法，还包括在所述设备信息数据集中提供一供应商扩展字段，用于描述由所述媒体设备支持的扩展集。

5. 如权利要求 2 所述的方法，还包括在所述设备信息数据集的一功能模式字段中提供一媒体设备类别。

6. 如权利要求 2 所述的方法，还包括向所述应答器提供一设备属性描述数据集。

7. 如权利要求 6 所述的方法，还包括向所述应答器提供一对象信息数据集，其包括用于描述对象特征的多个字段。

8. 如权利要求 7 所述的方法，还包括提供一对象属性描述数据集，用于枚举所选对象的属性。

9. 如权利要求 1 所述的方法，还包括向所述起始器提供会话控制工具，用于打开与所述应答器的通信会话。

10. 如权利要求 2 所述的方法，还包括向所述起始器提供设备信息恢复工具，用于从所述设备信息数据集中恢复信息。

11. 如权利要求 10 所述的方法，还包括向所述起始器提供一对象信息恢复模块，用于恢复对象信息。

12. 如权利要求 1 所述的方法，还包括向所述起始器提供一对象修正模块，用于修正所述应答器上的对象。

13. 如权利要求 1 所述的方法，还包括向所述起始器提供一对象捕捉模块，用于指示所述应答器捕捉并增加新对象。

14. 如权利要求 1 所述的方法，还包括向所述起始器提供一关联机制，用于将一个所选的对象与至少另一个所选的对象相关联。

15. 一计算机可读媒体，其具有用于执行权利要求 1 所述方法的计算机可执行指令。

16. 一种用于控制媒体设备的方法，该方法包括：
获取一设备信息数据集，该设备信息数据集指定了对于该媒体设备特定的一组所支持的设备和对象属性码；
执行操作以获取与所述属性码相关的设备属性描述或对象属性描述；
根据对所述设备属性和对象属性的了解，对所选的对象执行期望的控制操作。

17. 如权利要求 16 所述的方法，还包括获取描述所选对象的特征的对象

信息数据集。

18. 如权利要求 16 所述的方法，其特征在于，执行操作以获取设备属性描述或对象属性描述包括：使用设备属性恢复模块和对象属性恢复模块来获取设备属性描述数据集或对象属性描述数据集。

19. 如权利要求 18 所述的方法，其特征在于，所述操作产生设备属性数据集和对象属性数据集之一。

20. 如权利要求 16 所述的方法，其特征在于，获取设备信息数据集包括：在所述设备信息数据集的功能模式字段中确定设备类别。

21. 如权利要求 16 所述的方法，其特征在于，执行期望的控制操作包括：使用对象捕捉模块来进行对象捕捉。

22. 如权利要求 16 所述的方法，其特征在于，执行期望的控制操作包括：使用对象修正模块来进行对象修正。

23. 如权利要求 16 所述的方法，其特征在于，执行期望的控制操作包括：使用关联机制以提供从一对象到至少另一对象的基准。

24. 如权利要求 16 所述的方法，还包括使用存储器位置模块来定位与所述媒体设备相关的各个存储器。

25. 如权利要求 16 所述的方法，还包括使用会话控制工具来打开与媒体设备的通信会话。

26. 如权利要求 16 所述的方法，还包括当媒体设备的属性改变时在所述媒体设备中产生一事件。

27. 一计算机可读媒体，其具有用于执行权利要求 1 所述方法的计算机可执行指令。

28. 一种用于方便控制多种类型的媒体设备的系统，该系统包括：
用于从一组设备数据集恢复信息的设备恢复模块，所述设备数据集包括一设备信息数据集，其包括一组对于一类媒体设备特定的属性；
用于从一组对象数据集恢复对象信息的对象恢复工具；以及
用于控制所述设备上的对象的控制命令模块。

29. 如权利要求 28 所述的系统，还包括用于打开和关闭与媒体设备的会话的会话控制工具。

30. 如权利要求 28 所述的系统，其特征在于，所述设备恢复模块包括一用于恢复设备信息数据集的设备信息恢复模块以及一用于恢复设备属性描述数据集的设备属性恢复模块。

31. 如权利要求 28 所述的系统，其特征在于，所述对象恢复工具包括：
用于恢复对象信息数据集的对象信息恢复模块、用于恢复对象属性描述数据集的对象属性恢复模块、以及用于恢复对象句柄列表的对象句柄恢复模块。

32. 如权利要求 28 所述的系统，其特征在于，所述控制命令模块包括用于修正所述媒体设备上的对象的对象修正模块。

33. 如权利要求 28 所述的系统，其特征在于，所述控制命令模块包括用于开始对象捕捉的对象捕捉模块。

34. 如权利要求 28 所述的系统，其特征在于，所述控制命令模块还包括用于将一对象与至少另一个对象相关的关联机制。

35. 如权利要求 28 所述的系统，其特征在于，所述设备数据集包括一设备信息数据集，其包括所支持的属性和设备类别的列表。

36. 如权利要求 35 所述的系统，其特征在于，所述设备数据集包括用于描述所述设备属性的设备属性描述数据集。

37. 如权利要求 28 所述的系统，其特征在于，所述对象数据集包括对象信息数据集和对象属性描述数据集。

38. 一种方便在起始器和应答器之间的通信的系统，该系统包括：

保存在所述应答器中的设备数据集，所述设备数据集包括至少一个设备类别和一组对于所述类别特定的属性；

保存在所述应答器的对象数据集，所述对象数据集包括对于各对象特定的信息；以及

所述起始器可访问的恢复模块，用于从所述设备数据集和对象数据集中收集信息。

39. 如权利要求 38 所述的系统，还包括对于所述起始器可访问的控制模块，用于控制保存在所述应答器中的对象。

40. 如权利要求 39 所述的系统，其特征在于，所述控制模块包括对象捕捉模块和对象修正模块。

41. 如权利要求 39 所述的系统，其特征在于，所述控制模块包括用于将一对象与至少另一个对象相关的关联机制。

42. 如权利要求 38 所述的系统，其特征在于，所述对象数据集被配置成保存多种类型的媒体的信息，包括图像对象、音频对象和视频对象。

43. 如权利要求 38 所述的系统，其特征在于，所述设备数据集包括设备信息数据集和设备属性描述数据集。

44. 如权利要求 38 所述的系统，其特征在于，所述对象数据集包括对象信息数据集和对象属性描述数据集。

45. 如权利要求 38 所述的系统，其特征在于，所述对象数据集包括对象属性值数据集。

46. 如权利要求 38 所述的系统，其特征在于，所述恢复模块包括设备信息恢复模块和对象信息恢复模块。

47. 如权利要求 38 所述的系统，其特征在于，所述恢复模块包括对象信息恢复模块、对象属性恢复模块以及对象属性值恢复模块。

方便计算设备和多种媒体设备之间的通信的方法和系统

发明领域

本发明涉及在一计算设备和其他设备间通信的领域，更具体地说，涉及在计算设备和多种类型的媒体设备之间的通信。

发明背景

近来，通信协议已被发展为允许一计算设备控制诸如数码相机的媒体设备并与之通信。现存的通信协议允许在一计算设备和一数字静止相机之间进行通信。一个现存的协议，ISO 157540 图片传输协议(PIP)，其被位于华盛顿州雷蒙德的微软公司结合于产品，该协议可被用于从诸如相机的图像设备向个人计算设备传送图像的连接中。该协议定义了如何使数字静止相机与使用 Microsoft Windows®和 Microsoft®类别的驱动器操作的个人计算设备进行通信，Microsoft Windows®和 Microsoft®类别的驱动器都是位于华盛顿州雷蒙德的微软公司的产品。

需要对现存的协议进行扩展以使其具有连接除数字静止相机以外的媒体设备的功能。这样的一种扩展可使现存的操作系统向数字媒体设备提供强健而稳定的集线器(hub)。此外，这种扩展可使硬件制造商关注于开发它们的硬件和有附加值的软件，因为与一个人计算设备进行通信的基础协议将不再是一个焦点。类似的示例应用于 DVD 播放机，DVD 制造商现在关注于它们的效果以及 DVD 播放机的能力，这一部分是由于其依靠用于与电视机进行通信的应用于电视的标准连接机制(S-Video, 合成, 等等), 以及依靠存储媒介的信息的标准设计。

因此，就需要方便计算设备和其他类型的媒体设备之间的通信。这种设备可包括有集成相机或没有集成相机的蜂窝电话、数字音频播放器、数字视频播放器、具有静止成像能力的数字视频相机、个人数字助理(PDA)、数字语音记录器，以及其他类型的媒体设备或任何前述设备的组合。理想的，一协议将定义用于描述、配置以及控制该媒体设备的机制，就如 PTP 为数字静止相机所做的那样。需要一个协议来定义多种媒体，诸如照片、音频/音乐、视频以及其他与丰富的元数据(metadata)相关的媒体是如何向一设备发射并从该设备接收。

发明概述

按照一个方面，本发明针对一种方便在一起始器和一应答器之间的通信的方法，其中该应答器是多种类型的媒体设备之一。该方法包括向应答器提供多个数据集(data sets)，该数据集包括一组由应答器支持的属性，其中属性对于媒体设备的类型是特定的并向起始器提供多个控制模块，用于从多个数据集中搜寻信息并用于控制应答器的对象。

在另一个方面，本发明针对一种用于控制一媒体设备的方法。该方法包括：取得一设备信息数据集，该数据集指定一组被支持的设备和对于媒体设备特定的对象属性；以及执行操作以取得与属性编码相关的一设备属性描述或者一对象属性描述。所述方法还包括根据设备属性和对象属性的了解对所选择的对象执行期望的控制操作。

在又一个方面，本发明针对一种方便控制多种类型的媒体设备的系统。该系统包括用于从一组设备数据集中恢复信息的设备恢复模块，该设备数据集包括一设备信息数据集，其包括一组对于一类媒体设备特定的属性。该系统还包括用于从一组对象数据集中恢复对象信息的对象恢复工具以及用于控制媒体设备上的对象的控制命令模块。

在还有一方面，本发明针对一种用于方便一起始器和一应答器之间的通信的系统。该系统包括与应答器一同保存的设备数据集，该设备数据集包括至少一设备分类以及对于该分类特定的一组属性。该系统还包括与应答器一同保存的对象数据集，该对象数据集包括对于每一对象特定的信息以及起始器可访问的恢复模块，用于从设备数据集和对象数据集中收集信息。

附图简述

本发明在下面结合附图详细描述，其中：

图 1 是用于在其中实现本发明的适当计算系统的框图；

图 2 是显示本发明一系统的框图；

图 3 是显示按照本发明一实施例的媒体设备的框图；

图 4 是说明按照本发明一实施例的应答器内存的框图；

图 5 是说明按照本发明一实施例的起始器内存的框图；

图 6 是说明在一起始器和一应答器之间的会话所包括的步骤的流程图。

发明详述

图 1 说明了可实现本发明的合适的计算系统环境 100。该计算系统环境 100 仅是合适的计算环境的一个实例,其并不是希望对本发明的使用和功能范围做出任何限制。该计算环境 100 同样不应该被解释为需要依靠或要求在示范操作环境 100 中说明的任何一个组件或者它们的组合。

本发明可在由计算机执行的计算机可执行指令的一般环境(context)中描述,例如程序模块。通常,程序模块包括例程(routines)、程序、对象、组建、数据结构等等,它们执行特定的任务或者实现特定的抽象的数据类型。此外,熟悉本领域的技术人员应当了解本发明可与其他计算机系统配置一同实现,包括手持式设备、多处理器系统、基于微处理器或者可编程的消费者(consumer)电子器件、迷你计算机、大型计算机等等。本发明也可在分布式计算环境中实现,其中各任务是由通过一通信网连接的远程处理设备执行。在一分布式计算环境中,程序模块既可位于本地计算机存储媒体中也可以位于远程计算机存储媒体中,包括存储设备。

参照图 1,一用于实现本发明的示例系统 100 包括一以计算机 110 形式出现的通用计算设备,其包括一处理单元 120、以系统存储器 130 以及一系统总线 121,该总线耦合于多种系统组件,包括连接至处理单元 120 的系统存储器。

计算机 110 一般包括多种计算机可读媒体。为了示范而不是限制,计算机可读媒体可包括计算机存储器媒体和通信媒体。系统存储器 130 包括以易失性/非易失性存储器形式出现的计算机存储器媒体,诸如只读存储器(ROM)131 和随机存取存储器(RAM)132。一基本输入/输出系统 133(BIOS),包括帮助在计算机 110 内的元件之间传输信息的基本例程,例如在启动(start-up)期间的例程,一般被保存在 ROM 131 中。RAM 132 一般包括数据和/或程序模块,其对于处理单元 120 是可被即时存取的和/或当前正在被处理单元 120 操作。为了示范而不是限制,图 1 说明了操作系统 134、应用程序 135、其他程序模块 136 以及程序数据 137。

计算机 110 还可包括其他可移动/不可移动的、易失性/非易失性的计算机存储媒体。仅仅是为了示范,图 1 说明一硬盘驱动器 141,其从不可移动的、非易失性磁性媒体中读取数据或向其写入数据,一磁盘驱动器 151,其向可移动的、非易

失性的磁盘 152 中读取数据或向其写入数据, 以及一光盘驱动器 155, 其向可移动的、非易失性的光盘 156 中读取数据或向其写入数据, 所述光盘 156 是诸如 CD ROM 或其他光盘媒体。其他可移动/不可移动的、易失性/非易失性的计算机存储媒体可被用于一示范操作环境中, 包括但不限于卡式磁带、闪存存储器卡、数字通用光盘、数字视频带、固态 RAM、固态 ROM 等等。硬盘驱动器 141 一般通过一诸如接口 140 的不可移动的存储器接口而连接至系统总线 121, 而磁盘驱动器 151 和光盘驱动器 155 一般通过一诸如接口 150 的可移动存储器接口而连接至一系统总线 121。

上面讨论并在图 1 中说明的驱动器和与其相关的计算机存储媒体提供对于计算机可读指令、数据结构、程序模块和其他用于计算机 110 的数据的储存。例如, 在图 1 中, 硬盘驱动器 141 被说明为保存操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147。注意到这些组件可以与操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137 相同或不同。操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147 此处被给予了不同的数字以说明至少它们是不同的副本。用户可通过诸如键盘 162 和指示(pointing)设备 161 这样的输入设备来向计算机 110 输入命令和信息, 一般定点设备是指鼠标、轨迹球或触摸板。其他输入设备(没有图示)可包括麦克风、手柄、游戏板、圆盘卫星天线、扫描仪等等。这些以及其他的输入设备经常通过一耦合至该系统总线的用户输入接口 160 连接到处理单元 120, 但也可以通过其他的接口和总线结构连接, 例如并行接口、游戏接口或通用串行接口(USB)。一监视器 191 或其他类型的显示设备也通过一诸如视频接口 190 接口连接到该系统总线。除了监视器, 计算机还可以包括其他外围输出设备, 例如扬声器 197 以及打印机 196, 其可通过一输出外围接口 195 连接。

本发明中的计算机 110 可操作于一网络环境中, 其使用逻辑连接至一个或多个诸如远程计算机 180 的远程计算机。远程计算机 180 可以是一个人电脑, 且一般包括上述结合计算机 110 而描述的许多或全部元件, 尽管在图 1 中只说明了一个存储设备 181。图 1 中说明的逻辑连接包括一局域网(LAN)171 和一广域网(WAN)173, 但也可包括其他网络。

当用于 LAN 网络环境时, 计算机 110 通过一网络接口或适配器 170 连接至 LAN171。当用于 WAN 网络环境时, 计算机 110 一般包括一调制解调器 172 或其他用于在诸如 Internet 的 WAN 173 上建立通信的装置。调制解调器 172 可以是内置或

外置的，其可通过用户输入接口 160 或其他合适的机制连接至系统总线 121。在一网络环境中，结合计算机 110 而说明的程序模块或其部分可被保存在远程存储设备中。为了说明而不是限制，图 1 说明了远程应用程序 182、183、184 和 185 保存在存储谁 181 中。在示出的实现例中，包括设备程序 182、设备 183、媒体对象 184 和对象属性 185。需要理解示出的网络连接是示例性的，也可以使用其它用于在计算机之间建立通信链路的装置。

尽管计算机 110 的许多其他的内置组件没有示出，本领域内的一般技术人员应该理解这样的组建和互连是公知的。因此，关于计算机 110 的这些内在结构的其他细节不必要结合本发明而在此公开。

图 2 是示出本发明一系统的框图。该系统包括一起始器 200，其能够通过网络 20 与多个应答器 300、400 以及 500 通信。起始器 200 一般将和上面结合图 1 描述的计算机设备 110 类似。应答器 300、400 以及 500 表示属于不同种类的多个媒体设备。这些媒体设备包括数码静态相机设备、(具有或没有静态图像捕捉功能的)数码摄像机、诸如个人音乐播放器以及个人视频播放器这样的便携式媒体播放器、(具有或没有媒体捕捉/回放(playback)能力)的蜂窝电话，以及其他媒体设备。应答器 300、400 和 500 一般将被分成几类，每一类具有不同组的属性。网络 20 可以是诸如那些在上面结合图 1 描述的任何类型的网络。

在一个典型的情况中，起始器 200 打开一个与应答器 300、400 和 500 的通信会话。应答器 300、400 和 500 以允许通信会话继续的信息来应答。如此处说明的，个人计算设备 110 一般用作起始器 200 而媒体设备一般用作为应答器 300、400 和 500。然而，角色的互换也是可能发生的，比如一媒体设备打开了一个与个人计算设备 110 的通信会话。在这种情况下，该媒体设备可被增强以包括合适的用于打开并关闭一会话的工具，下面会详细描述。

图 3 是说明按照本发明的一实施例的一媒体设备/应答器 300 的框图。媒体设备 300 可归于多个分类中的一个且媒体设备 300 的特定特征取决于媒体设备 300 是数码静态相机设备、(具有或不具有静态图像捕捉功能的)数码摄像机、诸如个人音乐播放器以及个人视频播放器的便携式媒体播放器、(具有或不具有媒体捕捉/回放功能)的蜂窝电话、还是其他媒体设备。媒体设备包括媒体工具 302、信号处理设备 304、控制单元 306、通信接口 308、内存 310 以及存储器 350。通信接口

308 使能(enable)媒体设备或应答器 300 与计算系统或起始器 200 交互。媒体工具 302 对于所选择的媒体设备 300 是特定的。如果媒体设备 300 是视频或数码相机, 则媒体工具可包括一图像捕捉单元。如果媒体设备 300 是一音频设备, 则媒体工具 302 可以是音频记录和播放工具。存储器 350 对于所实现的媒体设备 300 的类型也是特定的。通信接口 308 可以是要求将相机直接插入计算机系统 200 或允许通过因特网连接到计算机系统 200 的接口。在一个实施例中, 媒体设备 300 通过一无线接口连接计算机系统 200。

图 4 是说明按照本发明一实施例的设备/应答器内存 310 的框图。不论媒体设备 300 的分类和特征, 在其内存 310 中, 媒体设备 300 保存数据集 320、对象数据集 330 和单个(per)会话数据集 340 以方便与个人计算设备或起始器 200 的通信。

数据集 320、330 和 340 由数据类型组成。在本发明的一个实施例中, 每个数据类型具有唯一的 16 比特数据码, 该数据码用于在起始器 200 和应答器 300 之间传递数据类型。数据类型包括多种大小的有符号和无符号的整数类型、相同的数组以及字符串。下面的归纳表提供了对于数据类型的解释。

数据类型归纳表		
类型码	类型名	描述
0x0000	UNDEF	未定义
0x0001	INT8	有符号 8 比特整数
0x0002	UINT8	无符号 8 比特整数数组
0x0003	INT16	有符号 16 比特整数
0x0004	UINT16	无符号 16 比特整数数组
0x0005	INT32	有符号 32 比特整数
0x0006	UINT32	无符号 32 比特整数数组
0x0007	INT64	有符号 64 比特整数
0x0008	UINT64	无符号 64 比特整数数组
0x0009	INT128	有符号 128 比特整数
0x000a	UINT128	无符号 128 比特整数数组
0x4001	AIN8	有符号 8 比特数组

0x4002	AUINT8	无符号 8 比特数组
0x4003	AINT16	有符号 16 比特数组
0x4004	AUINT16	无符号 16 比特数组
0x4005	AINT32	有符号 32 比特数组
0x4006	AUINT32	无符号 32 比特数组
0x4007	AINT64	有符号 64 比特数组
0x4008	AUINT64	无符号 64 比特数组
0x4009	AINT128	有符号 128 比特数组
0x400a	AUINT128	无符号 128 比特数组
0xffff	STR	统一码(unicode)字符串

数据集 320、330 和 340 用于描述在协议中使用的较高级的实体，包括设备、存储器、对象和属性。

数据码是 16 比特的无符号整数，其用于识别不同的操作、应答、事件、属性以及在协议中定义的格式。可能的 16 比特值的范围被分成几部分，以使这些实体中的每一个能位于该范围中的指定部分；这种分割允许实体类型的识别。而且，每一个子范围可被分成一组标准数据码和一组供应商(vendor)定义的数据码。标准数据码对所有使用该协议的设备是公共的，而供应商定义的数据码可被供应商用于定义对该供应商设备的扩展。一个供应商能共享多个设备之间的扩展，在这种情况下这些设备将通过该设备信息数据集的供应商扩展字段而表示其通用性。

设备数据集 320 包括一设备信息数据集 322 和一设备属性描述数据集 324。一示范设备信息数据集 322 在下面显示。

设备信息数据集			
数据集字段	字段 顺序	大小 (比特)	数据类型
标准版本(Standard Version)	1	2	UINT16
供应商扩展 ID(VendorExtensionID)	2	4	UINT32
供应商扩展版本(VendorExtensionVersion)	3	2	UINT16
供应商扩展描述 (VendorExtensionDescription)	4	可变	字符串

功能模式 (FunctionalMode)	5	2	UINT16
支持的操作 (OperationsSupported)	6	可变	AUINT16
支持的事件 (EventsSupported)	7	可变	AUINT16
支持的属性 (PropertiesSupported)	8	可变	AUINT16
捕捉格式 (CaptureFormats)	9	可变	AUINT16
对象格式 (ObjectFormats)	10	可变	AUINT16
制造商 (Manufacturer)	11	可变	字符串
型号 (Model)	12	可变	字符串
设备版本 (DeviceVersion)	13	可变	字符串
序列号 (SerialNumber)	14	可变	字符串

特别的，设备信息数据集 322 在字段顺序 1 中报告了一新版本号。在字段 2-4，设备数据集包括供应商扩展信息。在字段 5，功能模式字段可报告一设备分类。更进一步，由设备信息数据集 322 返回的设备属性支持的字段(字段顺序 8)将返回一组固定的属性数据码，其对于该设备所支持的分类是特定的。保存在属性支持的字段中的设备属性码能够描述设备的所有分类的属性。例如，对于视频播放器设备，设备属性码描述了设备上的视频显示的特征。这些属性包括但不限于，屏幕尺寸、屏幕亮度以及对比度。这使设备 300 能支持完整数据码集合的一个子集以及供应商扩展数据码。设备所实现的供应商扩展数据码集合在设备信息数据集内的供应商扩展字段中指出，位于字段顺序 2-4。设备所实现的特定的供应商扩展数据码在设备信息数据集的以下字段中指出：字段顺序 6-10 中的支持的操作、支持的事件、支持的属性、捕捉格式以及对象格式字段。设备信息数据集 322 是只读的并提供静态设备信息的基础等级。

在支持的属性字段中的属性由一描述定义。标准属性的描述对于所有的设备是固定的，而供应商扩展属性的描述对于实现相同供应商扩展集合的那些设备是公共的。该描述固定了一属性密钥，指定为一数据码。属性同样具有一值并可按照每个设备修改(对于设备属性)，或按照每个对象修改(对于对象属性)。

设备信息数据集 322 提供了关于设备所支持的扩展的额外信息，以处理对象属性和设备属性，并且除图像格式外还处理媒体格式。供应商扩展描述字段，字段顺序 4 描述了设备所支持的不同的扩展集。供应商扩展集的名称由该字段中的一个

字符串指定。该描述最好是人类和计算机都能读取的。对象格式字段(字段顺序 10)中返回数组可包括所有类型的媒体格式的格式码。捕捉格式字段(字段顺序 9)中返回的数组可包括图像格式和其他媒体格式。例如,格式码能定义图像、音频、视频、播放表、列表以及消息格式。

设备属性描述数据集 324 还定义了设备属性。给定的设备对于每一个设备属性将仅具有一个属性值。一示范设备属性描述数据集 324 在下面示出:

设备属性描述数据集				
字段	字段顺序	大小(字节)	数据类型	描述
属性码(PropertyCode)	1	2	UINT16	特定的属性码数据码
数据类型(Datatype)	2	2	UINT16	属性的数据类型
读写(GetSet)	3	1	UINT8	该字段指定其属性是只读(Get)还是读写(Get/Set)。 0x00 Get 0x01 Get/Set
默认值(DefaultValue)	4	DTS	-	指定属性的默认设置
当前值(CurrentValue)	5	DTS	-	属性的当前值
形式标志(FormFlag)	6	1	UINT8	该字段指定了下一字段的格式 0x00 无。 0x01 范围形式。 0x02 枚举形式。 0x03 日期时间形式。 0x04 事件列表形式。 0x05 关键值形式。 0x06 规则表示形式。 0x07 XML 模式形式。 0x08 XML 基准形式。
形式(FORM)	N/A	-	-	该数据集由形式标志所确定。

属性描述数据集 324 以字段顺序 1 中的属性码、字段顺序 2 中的数据类型、

字段顺序 3 中的读写以及字段顺序 4 中的默认值开始。字段顺序 1 中的属性码对应于属性的数据码。字段顺序 2 中的数据类型字段描述了属性值的数据类型。字段顺序 3 中的读写字段表示了属性值是否可由起始器 200 修改。字段顺序 4 中的默认值字段提供了属性的默认值。默认值具有由字段顺序 2 中的数据类型字段所提供的数据类型。默认值可用于通过下面描述的重置设备属性置操作把属性恢复至其默认值。在设备属性描述数据集 324 中，字段顺序 5 中的当前值字段提供设备属性的当前值。

在起始器 200 了解了在设备信息数据集 322 和设备属性描述数据集 324 中反映的上述设备能力后，起始器 200 能探测对象数据集 330。对象是大多数终端用户感兴趣的特征。对象对应于个别媒体，包括图片、歌曲或视频剪辑。大量协议操作包含在设备之间表征、管理以及传递对象。任何格式的对象可具有对象基准(object reference)。如果对象数据中的信息包括对设备上其他对象的静态基准，那么这些基准可与对象基准镜像。例如，一具有播放列表格式的对象可在其对象数据中保存一播放列表。该播放列表很可能是以特定顺序列出几首歌曲的静态格式。歌曲本身在设备 300 上表示为具有音频格式的对象。表示播放列表的对象可通过对象基准来索引表示歌曲的对象。这允许任何可访问该对象的设备 300，包括任何保存它们的设备，可以在播放列表中找到歌曲而不需要了解特定的播放列表格式。

对象数据集包括对象信息数据集 332、对象属性描述数据集 334 以及对象属性值数据集 336。对象信息数据集 332 的一范例在下面提供：

对象信息数据集			
数据集字段	字段顺序	大小(字节)	数据类型
存储 ID(Storage)	1	4	存储 ID
对象格式(ObjectFormat)	2	2	对象格式码
保护状态(ProtectionStatus)	3	2	UINT16
对象压缩大小(ObjectCompressedSize)	4	4	UINT32
缩略图格式(ThumbFormat)	5	2	对象格式码
缩略图压缩大小(ThumbCompressedSize)	6	4	UINT32
缩略图图片宽度(ThumbPixWidth)	7	4	UINT32

缩略图图片高度(ThumbPixHeight)	8	4	UINT32
图片宽度(ImagePixWidth)	9	4	UINT32
图片高度(ImagePixHeight)	10	4	UINT32
图片位长度(ImageBitDepth)	11	4	UINT32
父类对象(ParentObject)	12	4	对象句柄
相关类型(AssosiationType)	13	2	相关码
相关描述(AssosiationDesc)	14	4	相关码
序列号(SequenceNumber)	15	4	UINT32
文件名(Filename)	16	-	字符串
捕捉日期(CaptureDate)	17	-	数据时间字符串
修改日期(ModificationDate)	18	-	数据时间字符串
关键字(Keywords)	19	-	字符串

对象信息数据集 332 的字段描述了对对象的特征。在上面提供的对象信息数据集 332 中，字段主要是图像中心的元数据字段。对象由对象信息数据集 332、以及在对象属性描述数据集 334 中提出的任何数量的对象属性和在对象属性值数据集 336 中提出的对象基准来表征。

不同的设备 300 可以对扩展的对象属性提供不同级别的支持。如果设备不提供支持，对象信息数据集 332 仍可用于保存关于相应对象的信息。对象信息数据集 332 的字段顺序 19 的关键字字段可用于保存扩展的对象属性值。理想的，设备 300 为一组固定的对象属性提供支持并能够定义该设备的新对象属性。在这种情况下，新属性可用于保存相应对象上的扩展属性值。

每个对象可具有来自设备所维持的一公共集合中的一个或多个属性。对象属性与设备属性共享相同的数据码范围，且具有一组平行的操作，这将在下面进一步描述。如果没有对给定的对象设置一特定的属性，该对象就将在被查询该属性时返回默认值。当一个新对象在设备上建立时，它自动采取设备所维持的所有显著对象属性的默认值。一示范对象属性描述数据集 334 在下面提供：

对象属性描述数据集				
字段	字段 顺序	大小 (字节)	数据类型	描述

属性码(PropertyCode)	1	2	UINT16	特定的属性码数据码
数据类型(Datatype)	2	2	UINT16	属性的数据类型
读写(GetSet)	3	1	UINT8	该字段指定该属性是只读(Get)还是读写(Get/Set) 0x00 Get 0x01 Get/Set
默认值(DefaultValue)	4	DTS	-	指定属性的默认设置
组码(GroupCode)	5	4	UINT32	该属性所属的组
形式标志(FormFlag)	6	1	UINT8	该字段指定下一字段的格式 0x00 无 0x01 范围形式 0x02 枚举形式 0x03 日期时间形式 0x04 事件列表形式 0x05 关键值形式 0x06 常规表示形式 0x07 XML 模式形式 0x08 XML 基准形式
形式(FORM)	N/A	-	-	该数据集有形式标志确定

字段顺序 5 中的组码字段提供了一特定属性组的组码。由于设备属性没有被置入组中，因此该字段对于对象属性描述数据集 334 是特定的。

对象属性描述数据集中大多数其余字段限制了属性会采用的可能值。这些值对于属性的数据类型是特定的，就如字段顺序 2 中的数据类型字段所提供的，且进一步限制了即使在该数据类型中属性可能采用的值。例如，具有字符串(0xFFFF)数据类型的属性可被限制为字符串值“红”、“绿”和“蓝”。

字段顺序 6 中的形式标志字段指定了描述数据集的属性中其余字段的结构。该字段的每一个可能的值都具有一相应的字段列表，其在形式标志字段之后附加在属性描述数据集 334 后。例如，形式标志 0x02 表示该属性值被限制在一个值的范

围中。因此，数据集将包括由范围形式描述的最小值、最大值和步长大小字段。形式标志的第一个值 0x00 表示超过该数据类型的属性值并没有以任何方式被限制，因此，对象属性描述数据集 334 中的形式标志后面没有附加的字段出现。

在给定的设备或应答器 300 上，区分设备属性和对象属性，这是由于仅有一个设备，因此每个设备属性将仅具有一个值。然而，由于可能有潜在的许多对象，因此对象属性对于每一个对象可采用唯一的值。这种区别反映在包括属性值的设备属性描述数据集和不包括属性值的对象属性描述数据集之间的区别中。对象属性描述数据集 332 描述了对象属性，但不能提供对保存在设备上的所有对象提供该属性的所有值。因此，一示范的对象属性值数据集 336 在下面提供：

对象属性值数据集				
字段名	字段顺序	大小 (字节)	数据类型	描述
元素数量 (NumberOfElements)	1	4	UINT32	元素的计数
元素 1 对象句柄 (Element1ObjectHandle)	2	4	对象句柄	该元数据 描述的对象的对象句柄
元素属性码 (Element1PropertyCode)	3	2	数据码	该元数据元素的描述
元素 1 值 (Element1Value)	4	DTS	-	该元数据的实际值
元素 2 对象句柄 (Element2ObjectHandle)	5	4	对象句柄	该元数据描述的对象的对象句柄
元素 2 属性码 (Element2PropertyCde)	6	2	数据码	该元数据元素的描述
元素 2 值 (Element2Value)	7	DTS	-	该元数据的实际值
元素 N 对象句柄 (ElementNObjectHandle)	5	4	对象句柄	该元数据描述的对象的对象句柄

元素 N 属性码 (ElementNPropertyCde)	6	2	数据码	该元数据元素的描述
元素 N 值 (ElementNValue)	7	DTS	-	该元数据的实际值

没有被设置成给定对象的特定值的属性将在对象属性值数据集 336 中返回默认值，其在所述属性的对象属性描述数据集 334 中被定义。

对象句柄是 32 位的无符号整数，用于识别设备存储器 350 上的对象。对象句柄是瞬时的且仅可在一单独会话中使用。相同的对象很可能被分配到不同会话中的一个完全不同的对象，即使这两个会话是并发的。对象句柄可用于与单个对象一起工作，包括读或写对象信息、属性或数据。

对象属性值和设备属性值具有稍许不同的要求，其反映在描述数据集和新的操作中。第一，与对于整个设备具有一个值相对，给定的对象属性对于该设备上的每一个对象具有一个值。第二，很容易在一次操作中从大量对象中恢复出全部或几个对象属性值，像数据的块传输。这些操作将在下面进一步描述。第三，对象属性的值类型和限制比设备属性的值类型和限制更加多。这些区别反映在对象属性数据集 334 的额外变体形式中。这些形式用于进一步限制对象属性在其数据类型中可采取的值。

如上所述，应答器 300 还包括存储器 350，其是在设备 300 上提供永久存储的物理媒体。存储信息数据集 346 定义了每一个存储器 350 的属性并在下面提供：

存储信息数据集			
数据集字段	字段顺序	大小(字节)	数据类型
存储类型(StorageType)	1	2	UINT16
文件系统类型(FilesystemType)	2	2	UINT16
访问能力(AccessCapability)	3	2	UINT16
最大容量(MaxCapability)	4	8	UINT64
以字节计的空闲空间(FreeSpaceInBytes)	5	8	UINT64
以对象计的空闲空间(FreeSpaceInObjects)	6	4	UINT32
存储描述(StorageDescription)	7	可变	字符串
卷标(VolumeLabel)	8	可变	字符串

不同于设备和对象，存储器 305 不具有单独的属性，仅具有收集在存储信息数据集 346 中收集的字段。设备 300 可具有一个以上的存储器。设备 300 在一个会话期间分配存储 ID 来区分不同的存储器 350。存储器保持对象以及与对象相关的信息。

一存储器由一存储 ID 识别。存储 ID 是 32 位的无符号整数，其识别物理存储媒体的一个单元，或者能够接受这种物理媒体的驱动器或连接器。在单独的会话期间，每一个存储器、或潜在的存储器都由设备 300 分配一个存储 ID。

如图 3 中所示，应答器 300 的几个数据集是对于每个会话唯一的每个会话 (per-session) 的数据集 340。这些数据集 340 包括一操作应答数据集 344、事件数据集 348 以及对象句柄列表 3422。事件发生机制 349 与事件数据集 348 相关联，而操作应答工具 345 与操作应答数据集 344 相关联。这些数据集将在下面结合本发明的方法而进一步描述。

图 5 说明了按照本发明的一个实施例的起始器内存 210。起始器内存 210 可包括会话控制工具 212，其用于在起始器 200 和应答器 300 之间打开、关闭并控制会话。设备信息恢复模块 214、设备属性恢复模块 216 以及设备属性修正模块 218 控制与设备信息数据集 322 和设备属性描述数据集 326 的相互作用。对象信息恢复模块 219、对象句柄恢复模块 220、对象捕捉模块 222、对象属性恢复模块 224、对象属性值恢复模块 225 以及对象修正模块 226 控制与对象数据集 330 和对象句柄列表 342 的相互作用。起始器内存一般还包括关联机制 228、操作请求数据集 230、存储器位置模块 232、存储器属性恢复模块 234 以及存储器修正模块 236。

通常，会话控制工具 212 进行一打开会话的操作并使设备 300 分配资源，根据需要向数据对象分配句柄，并进行任何连接特定的初始化。会话控制工具 212 分配一会话 ID。该会话 ID 在会话期间能被所有的其他操作使用。

如果起始器 200 试图以打开会话操作，如果该会话已经打开则该尝试可能会失败，设备 300 不支持多重会话。应该返回已打开的应答会话，并且以已经打开的会话的会话 ID 作为第一应答参数。如果设备支持多重会话，则已打开的应答会话也应该被使用，但是具有所选会话 ID 的会话已经被打开。如果设备支持多重会话，并且已经打开了最大数量的会话，则设备应该以设备忙进行应答。会话控制工具 212 还能够实现关闭会话操作以便关闭当前的会话。

起始器 200 可通过调用设备信息恢复模块 214 来恢复设备的基本特征。设备信息恢复模块 214 通过调用取得设备信息的操作来返回设备 300 的设备信息数据集 322。该操作可在会话之外被调用，但不开始一会话。在示出的实施例中，当在会话之外调用时，该操作请求数据集的会话 ID 和事务 ID 应该被设置为 0x00000000。

起始器 200 可通过检查在设备信息数据集 322 的属性支持的字段中返回的属性码数组，而使用它的设备信息恢复模块 214 来确定由设备 300 所支持的设备和对象属性。起始器 200 应该枚举属性码数组并调用设备属性恢复模块 216 和对象属性恢复模块 224，依次传送每一个属性码。设备信息数据集 322 上的属性码数组列出设备和对象属性两者的数据码。两类属性的数据码可以在该数组中混合，且起始器 200 负责识别哪个属性码定义哪种类型的属性。由设备属性恢复模块 216 和对象属性恢复模块 224 所实现的取得设备属性描述和取得对象属性描述操作确定了给定的属性码所定义的属性类型。

给定的属性码可以是设备属性或是对象属性，所以仅仅设备属性恢复模块 216 和对象属性恢复模块 224 中的一个将是成功的。如果两者都不成功的，则属性码对于设备 300 上的使用是无效的。

如果对象属性恢复模块 224 的操作对于给定的属性码是成功的，则起始器 200 可调用对象属性值恢复模块 225 以恢复对象属性值。因为对象属性值对于个别的对象是特定的，因此起始器 200 还将选择一个对象。如果设备属性恢复模块 216 是成功的，则一般可从设备属性描述数据集 324 中的数据类型中确定设备属性值。

设备属性修正模块 218 能通过调用取得设备属性描述操作并传送属性的属性码和新的属性值，从而修改设备上现有的设备属性值。起始器 200 能使用设备属性描述数据集 324 来确定属性的数据类型并确认该属性的新值。设备属性修正模块 218 返回一状态码来表示操作的成功。如果属性不可被设置，则返回应答访问被拒绝。如果该值不是该设备允许的，则返回无效设备属性值。如果属性值的格式或大小不正确，则返回无效设备属性格式。

对象句柄恢复模块 220 能快速定位设备上的所有对象。对象句柄恢复模块 220 返回由第一参数中的存储 ID 所表示的对象句柄的数组。如果期望一个贯穿所有存储器的集合的列表，该值应该被设置为 0xFFFFFFFF。对象句柄恢复模块 220 返回一对象句柄的列表，其中每一个保存在该设备上或所表示的存储器上的对象使用一

个条目(entry)。对于该列表中的每一个对象句柄,起始器 200 能调用对象信息恢复模块 228 并传递该对象句柄以恢复对象信息数据集 332。

对象信息恢复模块 219 返回一对象的基本特征。通过调用一取得对象信息操作并传递特定对象的对象句柄,对象信息恢复模块 219 返回该对象的对象信息数据集 332。

对象属性恢复模块 224 能通过调用取得对象属性列表操作并仅仅传递该对象的对象句柄,从而大量(in bulk)地恢复给定对象的所有属性。对象属性恢复模块 224 还可仅仅取得包括在一个组中的对象属性,该组由对象属性描述数据集 334 中的组值表示。对象属性恢复模块 224 还可通过传递对象句柄的“所有对象”值,从而大量(in bulk)恢复该设备上的所有对象。

对象属性值恢复模块 225 能恢复给定对象的属性值。通过调用一取得对象属性值操作并传递对象的对象句柄和属性的属性码,对象属性值恢复模块 225 将仅返回属性值。起始器 200 能使用相应的对象属性描述数据集 334 中的信息来解释属性值。对象属性数据恢复模块 225 也能返回一对象属性值数据集 336 的列表,其中每一个数据集包括相同的对象句柄、特定的属性码以及特定的属性值。

对象修正模块 226 能通过调用设置对象属性值操作并传递对象的对象句柄、属性的属性码以及属性的新值,从而修改对象的现有对象属性值。对象修正模块 226 能使用对象属性描述数据集 334 来确定属性的数据类型。对象修正模块 226 将返回一状态码以表示操作的成功。

对象修正模块 226 也能通过重置对象属性值操作并传递对象的对象句柄和属性的属性码,来恢复对象上的现有对象属性值。对象修正模块 226 将应用在属性的对象属性描述数据集 334 中定义的默认值。

对象修正模块 226 也能通过调用设置对象属性列表并传递仅仅指向给定的对象和属性的对象属性值数据集 336,从而大量(in bulk)地修改给定对象的所有属性。由该操作设置的对象属性的总数是由对象属性值数据集 336 中的第一字段确定的。该数组中的每一个条目(entry)能足以描述哪一个对象被修改、哪一个对象属性被设置以及什么值被设置在那个对象属性上。

对象修正模块 226 能通过调用一设置对象属性描述操作并传递一字段被设置为适当值的对象属性描述数据集 334,从而定义一新的对象属性。对象修正模块 226

返回一表示操作成功或是失败的状态码。如果操作成功，该新的对象属性被加到设备并可被应用到现有的或新的对象。对象属性没有被给定一值的现有对象将报告在对象属性描述数据集 334 中指定的默认值。如果操作失败，新的对象属性在该设备上维持未定义并不能被应用于任何现有的或新的对象。

对象修正模块 226 也能通过首先建立一新的对象，然后在该对象上设置基准来建立一对象的任意的集合。起始器 200 希望包括在该集合中的每一个对象应该具有对应的基准。

起始器 200 也能通过恢复该对象的对象句柄来修改该任意的集合，该对象句柄表示使用对象句柄恢复模块 220 的集合。对象句柄恢复模块 220 能使用集合对象的对象句柄来返回一对象句柄列表，该列表中的每一个对象句柄都对应于之前就属于该集合的对象。对象修正模块 226 能修改该列表，删除与不再属于该集合的对象相对应的对象句柄，或增加与新加入到集合中的对象相对应的对象句柄。

对象捕捉模块 222 能请求设备 300 通过一异步进程捕捉一新媒体对象。对象捕捉模块 222 能以一起始捕捉操作开始，而设备 300 将以捕捉完成或存储器满事件来通知该进程的结束。设备 300 还将使用对象增加的事件来通知设备 300 上新媒体对象的建立，下面将详细描述。当捕捉在进行时，设备可产生对象增加的事件来通知由捕捉进程产生了新的媒体对象。为了在设备上创建的每一个新媒体对象产生一个对象增加的事件。如果一次捕捉产生新的对象但是存储器 350 没有足够的空闲空间来容纳这些对象，设备 300 就可产生存储器满事件。当捕捉完成并且成功后，设备可产生一捕捉完成事件。事件的产生将在下面结合事件发生机制 349 进一步描述。

为了使用创建的任意的集合，起始器 200 以表示该集合的对象的对象句柄开始。起始器接下来通过调用取得对象基准列表操作而从该集合对象中请求基准。基准作为对象句柄列表被返回，每一个基准有一个对象。对象句柄恢复模块 220 能进一步取得一对象句柄的列表并使用该列表来恢复对象信息数据集 332，以及那些对象的属性、基准或数据。

存储器属性恢复模块 234 能通过调用取得存储器信息操作来恢复存储器 350 的基本特征，该取得存储器信息操作使用了特定存储器的存储 ID。存储器属性恢复模块 234 将返回存储器信息数据集 346。

存储器修正模块 236 能通过接收并处理一存储器信息变化的事件来跟踪存储器信息数据集 346 中的变化。对于与存储器信息数据集 346 中的变化有关的附加信息，存储器属性恢复模块 234 恢复存储器信息数据集 346，起始器 200 能检查存储器信息数据集 346 是否有变化。

存储器位置模块 232 能帮助起始器 200 找到设备 300 上的所有存储器。存储器位置模块 232 将调用一取得存储 ID 的操作以返回存储器 ID 的列表。该列表可包括有效逻辑存储器的存储器 ID。有效逻辑存储器的存储器 ID 具有非零的物理存储器 ID 部分和非零的逻辑存储器 ID 部分。该数组也可包括当前没有媒体插入到其中的可移动媒体驱动器的存储器 ID。没有媒体插入其中的驱动器的存储器 ID 具有非零的物理存储器 ID 部分和零值的逻辑存储器 ID 部分。存储器属性恢复模块 234 能使用由存储器位置模块 232 所恢复的存储器 ID 来恢复相应的存储器信息数据集 346。

起始器内存 210 的关联机制 228 处理一存储器上树型结构组织的对象。关联就是对象的组，它们自身被表示为对象。关联中的成员是专用的，使得没有对象能属于一个以上的关联。此外，关联被组织成分层的结构，通常与用于表示存储器 350 上对象的基础文件系统平行。关联类似于计算机文件系统中的目录或是桌面用户界面隐喻 (metaphor) 中的文件夹。

关联的排列将存储器 350 上的所有对象组织成父辈/子辈的分层结构。该组织通过对对象信息数据集 332 的父辈对象字段被传递至起始器。每一个对象都是某个父辈的子辈，而父辈由在父辈对象字段中返回的对象句柄来指定。父辈对象字段中的对象句柄总是指向一个关联。除了每个存储器上的最高父辈或分层结构的根之外，每个关联都具有一父辈对象，其也是一个关联。

基准是与每个对象一起保存的对象句柄。对象可具有任何数量的基准。每个基准是一个无类型的定向指针，指向相同设备或存储器上的另一对象。例如，从对象 A 至对象 B 的基准被置于对象 A 上，并且指向对象 B。因为基准被表示对象句柄，因此它们仅仅在一个会话期间有效。起始器 200 和应答器 300 都负责将基准转换成其他的静态形式，如果它们的含义需要在会话之间保留。

基准可用于建立任意的、可能是重复的对象集合。例如，假设设备 300 保存歌曲作为对象，在这些对象中的数据中表示采样的音乐。同样的设备能保存播放列

表作为对象,其具有从一播放列表至包括在该播放列表内的所有音乐对象或歌曲的基准。每一个音乐对象可属于一个或多个播放列表。由于播放列表使用对象句柄来指定所包括的音乐对象,因此音乐对象的位置或名称的改变不会影响播放列表的组成。

按照另一个例子,音乐对象可使用基准来指定应该由数字权利管理(DRM)系统检验的许可以允许访问采样的音乐内容。许可本身也可被表示为对象,且这些许可对象可使用基准来表示它们应用的音乐对象。音乐对象可使用基准来表示哪些许可控制对它们的内容的访问。两个方向的基准都将允许关联机制 228 快速地查找应用于单个音乐对象的所有许可对象,或查找由一许可对象覆盖的所有音乐对象。

每一个对象具有一个对象句柄的数组,其引用同一设备、最好是同一存储器上的其他对象。对象基准是有方向的指针,从一个源对象指向一个目标对象。对象基准由设备维持在一种静态格式,以允许从源对象到目标对象的快速访问,但该协议仅仅对对象句柄起作用。来自源对象的对象基准可由对象句柄恢复模块 220 恢复,该模块返回一对象句柄的数组。类似的,来自源对象的对象基准可由对象修正模块 226 修改。两个方向的对象基准都用于实现两个方向上的指针。

起始器 200 和应答器 300 之间的通信在被称为事务处理(transaction)的离散步骤中发生。传递可采取操作的形式,其由上述起始器 200 请求,也可以采用事件的形式。事件由应答器 300 发送并由起始器 200 接收。

在具有前述所有操作的情况下,起始器 200 通过请求操作采取行动,包括建立操作请求数据集 230,以及向应答器 300 发送操作请求数据集 230。应答器 300 然后建立操作应答数据集 334。起始器和应答器之间的单独相互作用是一次事务处理,在一会话中发生一系列的事务处理。一般而言,起始器 200 和应答器 300 之间的通信形成一通信会话。一示范操作请求数据集 230 在下面示出:

操作请求数据集			
数据集字段	字段顺序	大小(字节)	数据类型
操作码(OperationCode)	1	2	UINT16
会话 ID(SessionID)	2	4	UINT16
事务处理 ID(TransactionID)	3	4	UINT32
参数 1(Parameter 1)	4	4	任意

参数 2 (Parameter 2)	5	4	任意
参数 3 (Parameter 3)	6	4	任意
参数 4 (Parameter 4)	7	4	任意
参数 5 (Parameter 5)	8	4	任意

字段顺序 1 中的操作码指定了所请求的操作。操作码最好是一个 16 比特的无符号整数，其被限制于用于标准和扩展操作的特定范围。在这个例子中，对于所有实现公共的标准操作码被限制在范围 0x1000 至 0x1fff 之间。特定实现专有的扩展操作码可占据范围 0x9000 至 0x9fff。

会话 ID 字段指定了其中开始该操作的会话。对于在任何会话之外开始的操作而言，或对于开始新会话的操作，会话 ID 字段被设为 0x00000000。操作应答数据集 344 的会话 ID 字段必须与相应的操作请求数据集 230 的会话 ID 字段相同。会话 ID 是无符号的 32 比特整数，其由起始器 200 使用会话控制工具 212 在会话开始时选择。会话 ID 最好对于起始器 200 和应答器 300 都是唯一的。会话 ID 被包括在所有的操作请求数据集 230 以及操作应答数据集 344 中，以将会话或事件与其发生的会话相关联。

会话和会话 ID 允许设备与相同的设备或与不同设备维持多重、并发的会话。例如，在同一宿主 起始器 200 上运行的两个不同的应用程序可打开与一目标设备 300 的单独的会话。第一应用程序可使用该会话来恢复并修改设备属性，而第二应用程序使用一不同的会话把新媒体对象下载至设备 300。

事务处理 ID 指定操作开始的事务处理。事务处理 ID 在开始一新会话的操作中必须从零开始并且应该根据每次额外的事务处理而增一。操作应答数据集 344 的事务处理 ID 字段必须与相应的操作请求数据集 230 的事务处理 ID 字段相同。事务处理 ID 是一个无符号的 32 比特整数，其在一会话中是唯一的。事务处理 ID 由起始器 200 在操作请求数据集 230 中发送并由应答器 300 在相应的操作应答数据集 344 中返回。

起始器 200 负责产生 事务处理 ID 的序列，应答器 300 负责检验到来的 事务处理 ID 没有已经在先前的事务处理中使用过。因此，起始器 200 记录在操作请求数据集 230 或事件数据集中使用的最近的事务处理 ID，使用操作请求数据集或事件数据集 348 中的值使其增一，并把最近的值保存为最近的事务处理 ID。应答器

300 可检验新的 事务处理 ID 大于任何之前接收的事务处理 ID 并将到来的 事务处理 ID 作为之前接收的最大 事务处理 ID 保存。

事务处理 ID 允许会话双方对包括在一次事务处理中的不同数据集进行相关，包括操作请求数据集 230、操作应答数据集 344 和事件数据集 348。这种能力在异步事件时尤其重要，诸如事务处理取消、通知显著交易处理中的状态改变。

参数字段 1-5 由各操作不同地解释。任何没有被给定操作使用的参数字段都应该被设置为 0x000000。应答器 300 在接收到操作请求数据集 230 后对操作应答，可能从起始器接收数据，并且可能向起始器发送数据。一给定事务处理可或者在请求中从起始器传送数据到接收器，或者在应答中从接收器传送数据到起始器。应答器 300 可建立操作应答数据集 344 并将它发送给起始器。一示范操作应答数据集 344 在下面示出：

操作请求数据集			
数据集字段	字段顺序	大小(字节)	数据类型
应答码(Response Code)	1	2	UINT16
会话 ID(SessionID)	2	4	UINT16
事务处理 ID(TransactionID)	3	4	UINT32
参数 1(Parameter 1)	4	4	任意
参数 2(Parameter 2)	5	4	任意
参数 3(Parameter 3)	6	4	任意
参数 4(Parameter 4)	7	4	任意
参数 5(Parameter 5)	8	4	任意

操作应答数据集 344 的应答码字段包含一指定给定应答的数据码。应答码是 16 比特的无符号整数，其被限制于用于标准以及扩展应答的特定范围。一般对于所有实现公共的标准应答码被限制于范围 0x2000 至 0x2fff 之间。属于一特定实现的扩展应答码被限制于范围 0xa000 至 0xffff 之间。额外的参数在上面结合操作请求数据集 230 实质性地描述。

在一事务处理期间，数据在一个或另一个方向上流动。起始器 200 可向应答器 300 发送数据或应答器 300 可向起始器 200 发送数据。起始器 200 也可在操作请求数据集 230 中向应答器 300 发送信息，而应答器 300 可在操作应答数据集 344

中向起始器 200 发送信息, 但该信息受到各自数据集的尺寸和参数字段的数量所限制。

当对象在一设备 300 上改变时, 或者当操作在进行时, 设备 300 可使用事件发生机制 349 产生一事件数据集 348。一示范事件数据集 348 在下面示出:

事件数据集			
数据集字段	字段顺序	大小(字节)	数据类型
事件码(Event Code)	1	2	UINT16
会话 ID(SessionID)	2	4	UINT16
事务处理 ID(Transaction ID)	3	4	UINT32
参数 1(Parameter 1)	4	4	任意
参数 2(Parameter 2)	5	4	任意
参数 3(Parameter 3)	6	4	任意

事件可由起始器 200 发送但是一般由应答器 300 发送。事件码字段指定了所产生的事件。字段顺序 1 中的事件码是 16 比特的无符号整数, 其被限制在用于标准和扩展事件的特定范围。标准事件码被限制于范围 0x4000 至 0x4fff。属于一特定实现的扩展事件码被限制于范围 0xc000 至 0xcfff。

字段顺序 2 中的会话 ID 字段指定了该事件所应用的会话。如果事件应用于一特定的操作, 在事件数据集的会话 ID 字段就与相应的操作请求数据集 230 的会话 ID 相同。如果事件应用于一特定的会话, 事件数据集 348 的会话 ID 字段就必须被设置为该会话的会话 ID。最后, 如果事件应用于所有显著的会话, 会话 ID 应该被设置成 0xffffffff。

字段顺序 3 中的事务处理 ID 字段指定了该事件所应用的事务处理。如果事件应用于一特定的操作, 则事件数据集的事务处理 ID 字段必须与相应的操作请求数据集 230 的事务处理 ID 相同。如果事件不应用于一特定的传递, 则事务处理 ID 字段应该被设置成 0xffffffff。参数字段 1 到 3 对于每个事件由不同的解释。任何没有被一给定操作使用的参数字段都应该被设置成 0x00000000。

可以产生多种类型的事件。当设备信息数据集 332 改变时, 事件发生机制 349 可产生一设备信息改变事件或一对象信息改变事件。两个事件都是从会话之外的改变或者不由起始器 200 引起的改变而产生。为了接收关于该改变的信息, 起始器

200 将调用设备信息恢复模块 214 或对象信息恢复模块 219 以恢复适当的信息集。类似的,当设备属性值或对象属性值被改变时,事件发生机制 349 可产生一设备属性改变事件或一对象属性改变事件。

当在存储器 350 上建立新的对象时,事件发生机制 349 可产生一对象增加事件。如果同时建立一个以上的对象,则将为每一个新对象产生分开的对象增加事件。该事件仅当对象由设备或由另一会话明确地建立时产生。对象增加事件包括新对象的对象句柄作为参数。为了搜寻信息,起始器 200 应该使用对象信息恢复模块 220 以调用取得对象信息操作并获取对象信息数据集 332。

当新的存储器被增加到设备 300 时,事件发生机制 349 可产生一存储器增加事件而不是一对象增加事件。存储器增加事件包括所增加的存储器的存储器 ID 作为参数。如果同时有一个以上的逻辑存储器通过插入一片物理存储媒体而被增加到该设备,则仅有一个存储器增加事件将由设备 300 产生。如果增加了一单独的逻辑存储器,则新存储器的存储器 ID 是有效的。如果一个以上的逻辑存储器被同时增加,可能是由一单独的物理存储器增加,则存储器 ID 将为 0x00000000。如果起始器 200 搜寻关于新存储器 350 的信息,它应该使用适当的操作以恢复存储器信息数据集 346。

事件发生机制 349 也可产生一对象删除事件。如果同时有一个以上的对象被删除,将为每一个被删除的对象产生分开的对象删除事件。该事件仅在对象被设备 300 或被其他会话明确地删除时产生。该事件在现有的存储器从设备上被删除时不产生,即使被删除的存储器上保存有对象。作为替代,当现有的存储器 350 从设备 300 被删除时,事件发生机制 349 可产生一存储器删除事件。

存储器删除事件包括被删除的存储器的存储器 ID 作为参数。起始器 200 不应该使用该存储器 ID 来表征存储器或与存储器上的对象一起工作。相反,起始器 200 应该使用该存储器 ID 来跟新它自己的关于存储器 300 以及设备 300 上可用的对象的信息。

图 6 是说明起始器 200 和应答器 300 之间的一示范会话的流程图。在步骤 10,起始器 200 使用其会话控制工具 212 以打开与应答器 300 的会话。尽管步骤 10 表示会话的开始,然而如果设备 300 不首先将其本身标识为支持该协议,则通信序列不会发生。在其上建立协议的传输层确定了标识的方式。设备 300 可同时被一个以

上的会话涉及，但并发的会话是相互独立的。会话各方之间的所有通信由对于该会话唯一的会话 ID 所限定(qualified)。当会话开始时由起始器 200 选择会话 ID，且会话 ID 必须对于起始器 200 和应答器 300 都是唯一的。

在步骤 12，起始器 200 通常使用设备信息恢复模块 214 来请求关于设备 300 的信息。在步骤 14，应答器 300 将返回设备信息数据集 322。如前面所说明的，除了其它数据，设备信息数据集 322 包含所支持的数据码的列表。设备信息数据集 322 是只读的且提供基本等级的静态设备信息。对于更加详细的信息，起始器 200 可使用设备属性恢复模块 216 来恢复设备属性描述数据集 324。

当起始器 200 了解应答器 300 的能力后，可进入步骤 16 以使用对象句柄恢复模块 220 来向设备查询用于该设备上的对象列表。对象句柄值是瞬时的并且在关闭会话操作或断开事件后不再维持。在步骤 18，设备 300 返回对象句柄的列表。

在步骤 20，起始器 200 使用对象属性恢复模块 224 向应答器 300 查询对于各对象特定的信息。每个对象属性由对象属性描述数据集 334 中的对象属性数据码唯一地标识。对象属性描述数据集 334 在步骤 22 中返回。对象属性描述数据集 334 描述了对对象属性值的大小(字节)和数据类型。

在设备的能力被了解且设备上的对象已被表征之后，起始器 200 可继续上载或恢复对象；增加、改变或删除对象元数据；或者进行任何其他由设备在步骤 30 中所支持的操作。

执行的操作包括任何上述结合图 5 描述的操作。这些命令能够控制媒体设备 300。对不从同类别的媒体设备来的对象进行操纵将导致不同的操作。例如，媒体回放设备要求不同于数字静态相机的控制命令。为了控制来自起始器 200 的回放，对媒体回放设备 300 上对象的操纵可导致诸如播放、暂停、下一轨、快进等命令。

当起始器 200 完成期望的操作时，会话控制工具 212 会在步骤 42 中终止该会话。会话控制工具 212 在步骤 44 中使现有的对象句柄无效。

上述的协议是传输层不可知的并能在 USB、USB2.0、IEEE1394、蓝牙、UPnP、TCP/IP 或满足传输要求的任何其他传输层上实施。另外，该协议提供广播媒体类型的支持、扩展和元数据支持。

本发明已结合特定的实施例进行描述，其在所有的方面都是用于说明而不是用于限制。适合本发明而不脱离其范围的其他实施例对于熟悉本领域的技术人员来

说是很明显的。

通过前面的描述，可见本发明能很好地适应所有上述的终端和对象，还具有其他的优势，其对于该系统和方法来说固有和明显的。需要理解的是，某一个特征以及子组合可以被使用而不需要参考其他的特征和子组合。这是可预期的并在权利要求的范围之内。

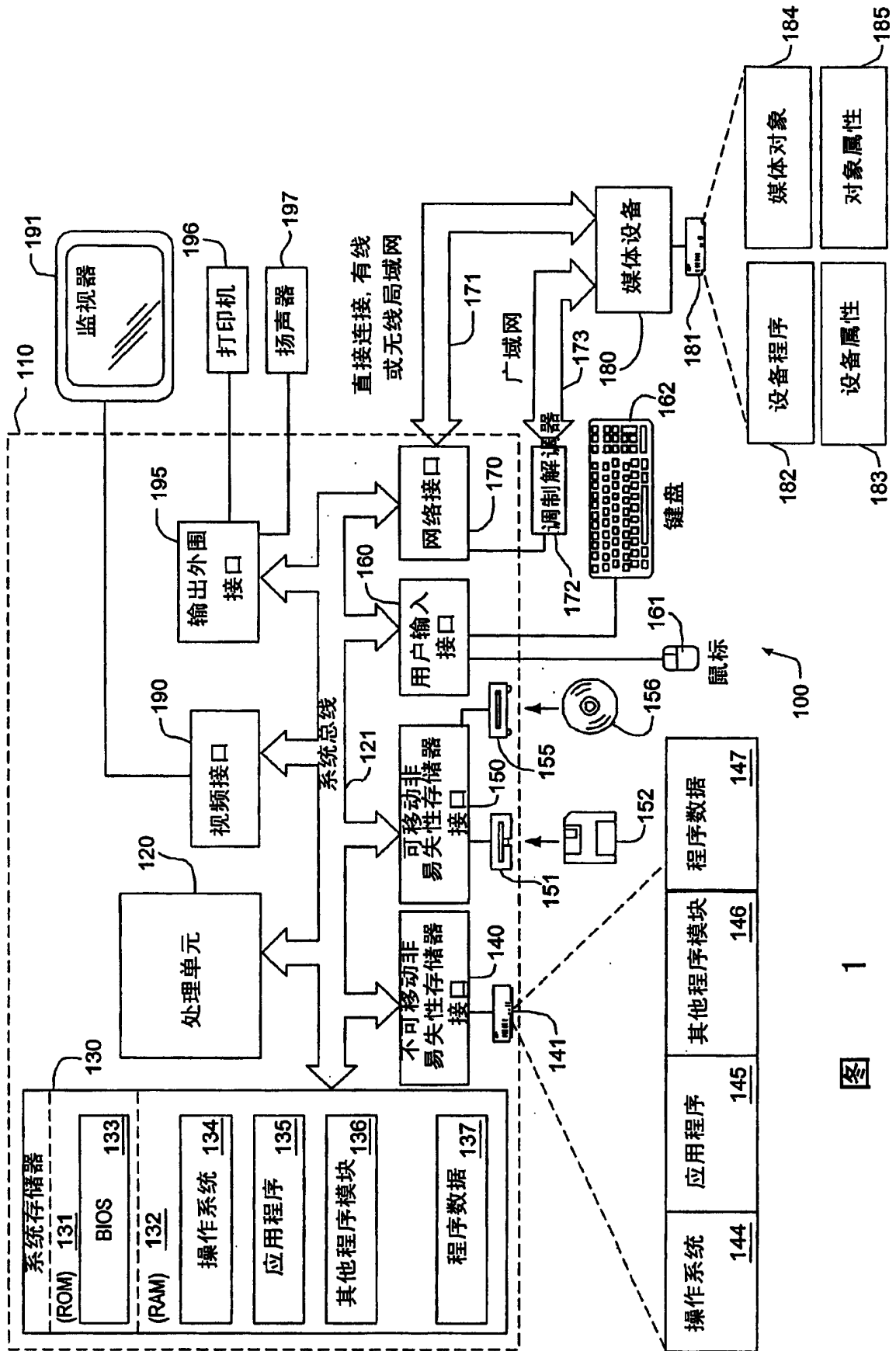


图 1

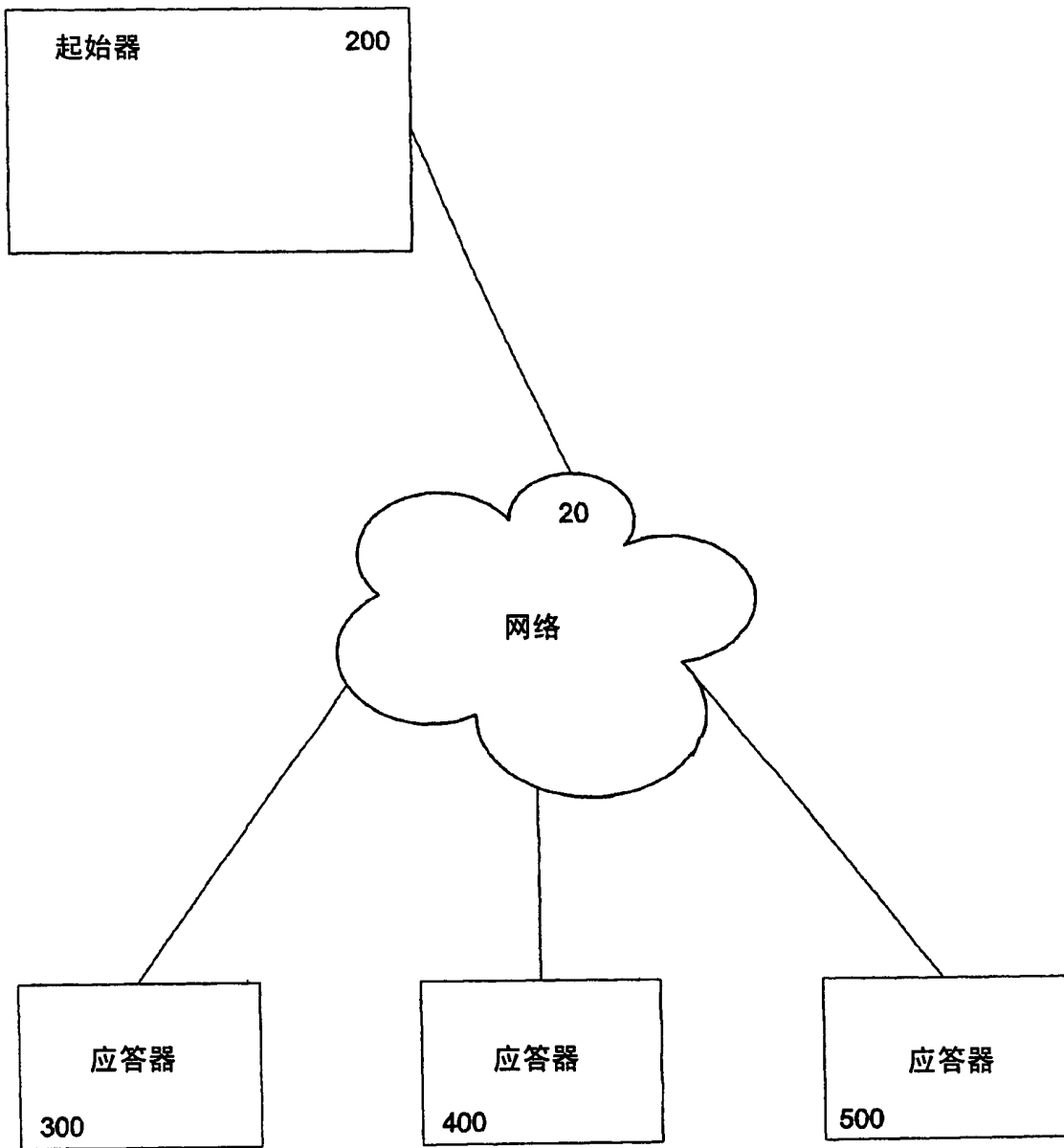


图 2

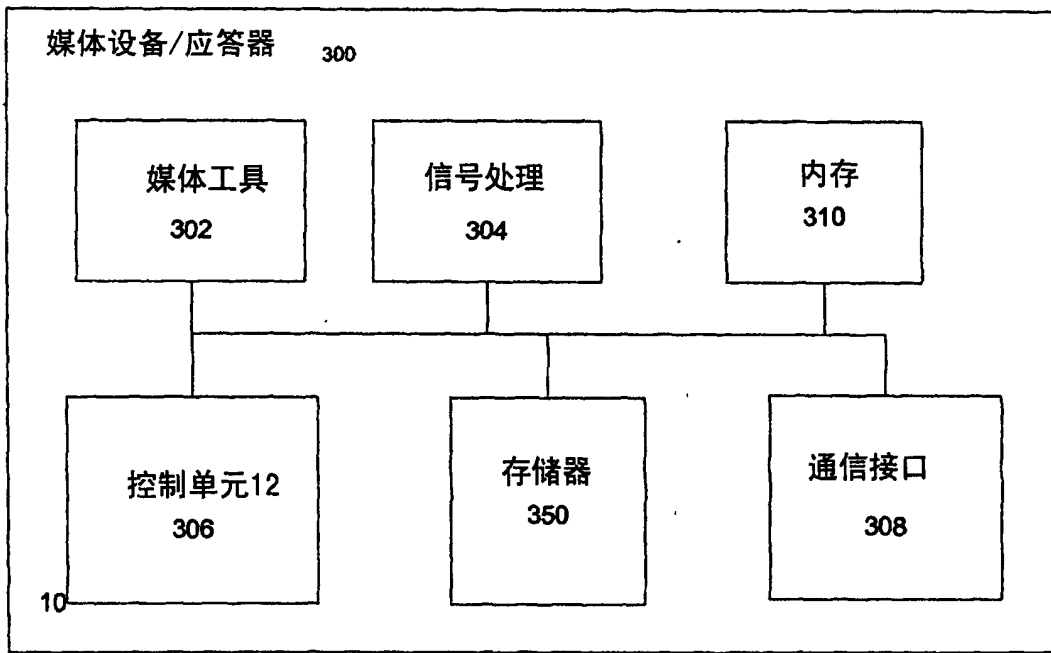


图 3

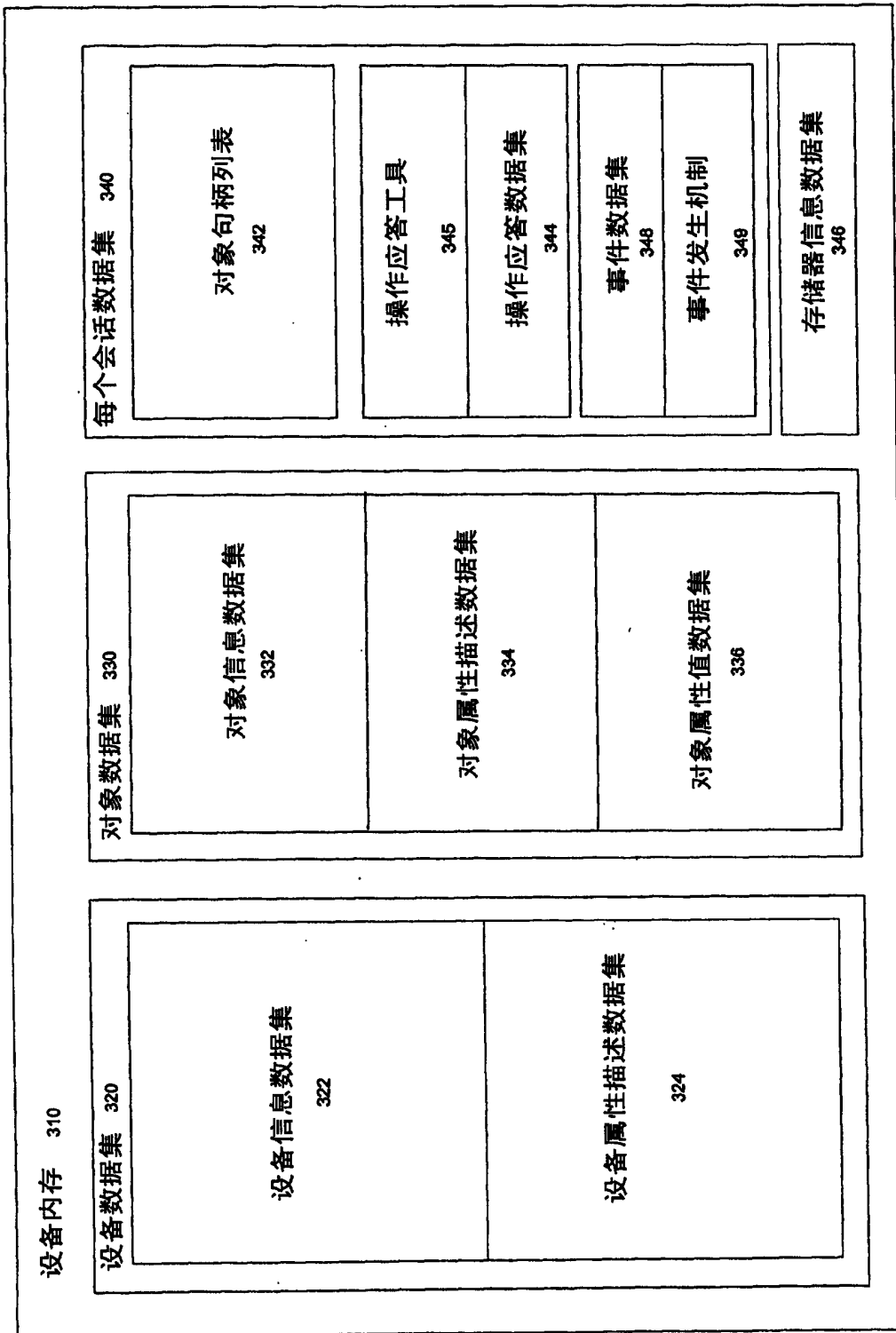


图 4

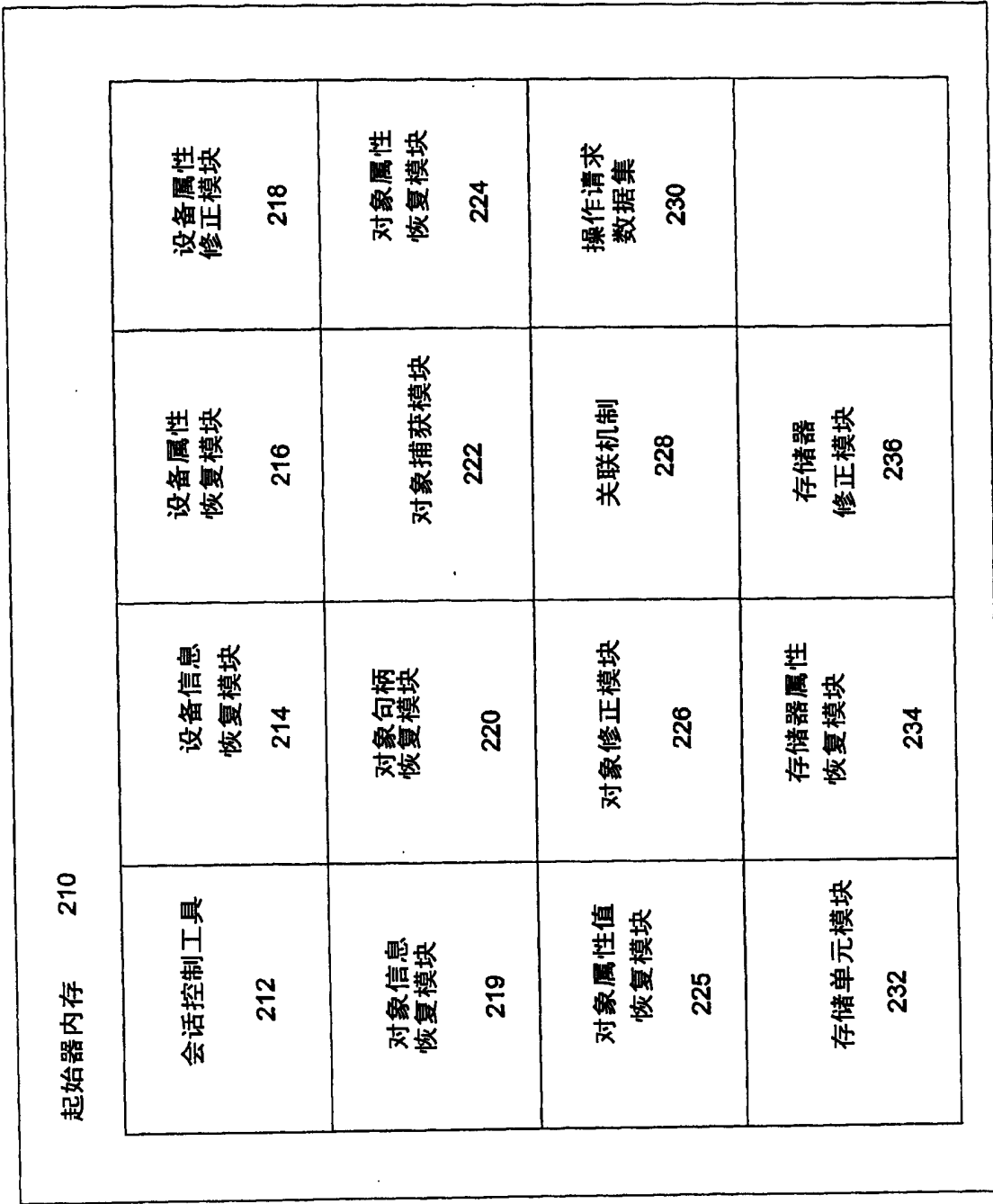


图 5

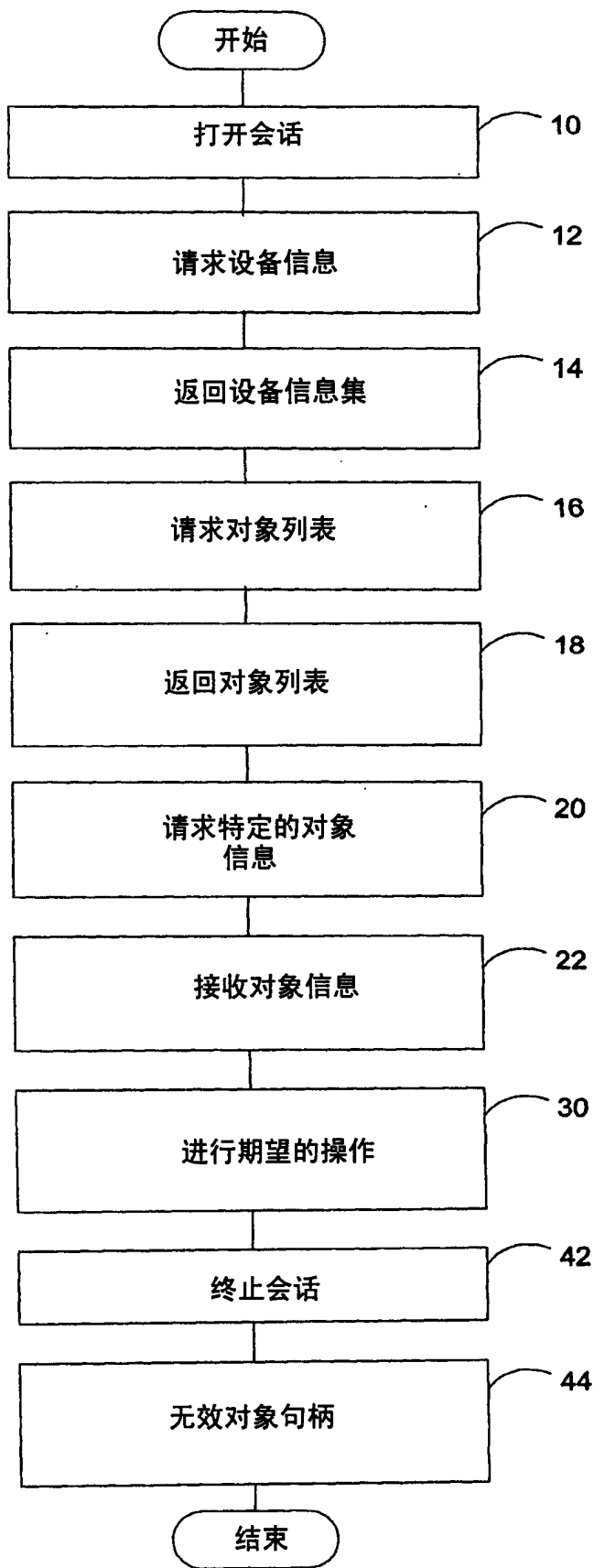


图 6