(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 101111895 B
(45)授权公告日 2012.06.13

(21)申请号 200580047676.5
(22)申请日 2005.11.10
(30)优先权数据
350193/2004 2004.12.02 JP
(85)PCT申请进入国家阶段日
2007.08.02
(86)PCT申请的申请数据
PCT/JP2005/021075 2005.11.10
(87)PCT申请的公布数据
WO2006/059483 JA 2006.06.08
(73)专利权人 索尼株式会社
地址 日本东京都
专利权人 索尼计算机娱乐公司
(72)发明人 浜田俊也 藤波靖 各务辰哉
内海秀介 井原宏二
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 郭定辉 黄小临
(51)Int.Cl.
G11B 27/10(2006.01)
(56)对比文件
JP 2004-328450 A,2004.11.18, 全文.
JP 10-271454 A,1998.10.09, 全文.
US 2003/161615 A1,2003.08.28,
审查员 孙燕

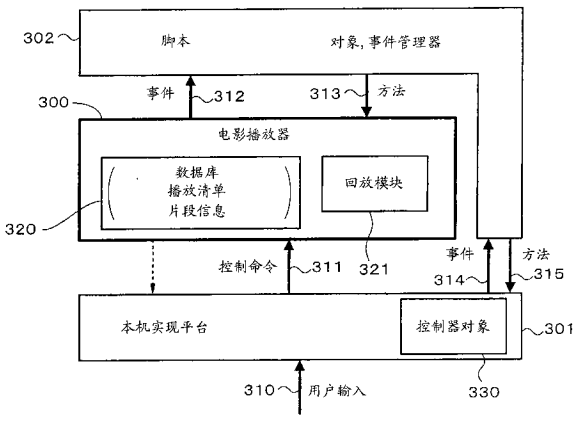
权利要求书 2 页 说明书 41 页 附图 47 页

(54)发明名称

再现装置和再现方法

(57)摘要

清楚地定义了播放器操作的状态转变,以便于交互式内容的生成。作为再现盘的播放器模型,考虑包含如下组成部分的模式:再现数据流的播放器(300);用作播放器(300)与硬件之间的接口的平台(301);以及实现内容制作者想要的情节的脚本层(302)。播放器(300)的状态被定义成从组合是否再现播放清单的两种状态和是否接受命令(311)的两种状态得出的四种状态。播放器(300)的四种状态之间的状态转变随来自脚本层(302)的方法(313)发生,而不是随播放器(300)本身或命令(311)发生。由于播放器(300)的状态数少和状态转变的条件清楚,因此容易生成交互式内容并将它们安装在设备上。



CN 101111895 B

1. 一种再现记录在记录媒体中的内容数据的再现装置，包含：

读取装置，用于从记录媒体中读取数据，该记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制该内容数据的再现的再现控制程序；

播放器装置，其根据四种状态被控制来按照再现控制程序再现该内容数据；

第一存储装置，用于存储在播放器装置再现期间指示播放器装置的状态的再现状态信息；

第二存储装置，用于备份存储在第一存储装置中的再现状态信息；以及

控制命令输出装置，用于将与用户操作相对应的控制命令给予播放器装置，其特征在于，与播放器装置的四种状态之间的状态转变一起，将存储在第一存储装置中的再现状态信息备份在第二存储装置中，并将备份在第二存储装置中的再现状态信息恢复到第一存储装置中，

这四种状态是通过组合根据是否再现内容数据分类的第一状态和第二状态和根据是否接受来自控制命令输出装置的控制命令分类的第三状态和第四状态定义的，其中在接受来自控制命令输出装置的控制命令的该第三状态中，不接受脚本层的控制命令，而在不接受来自控制命令输出装置的控制命令的该第四状态中，仅仅接受来自脚本层的控制命令，并且

由从脚本层而不是从控制命令输出装置发送的控制命令引起状态转变。

2. 根据权利要求 1 所述的再现装置，其特征在于，

播放器装置的四种状态之间的状态转变不是由播放器装置自发引起的。

3. 根据权利要求 1 所述的再现装置，其特征在于，

播放器装置在初始化之后处在不生成内容数据和不接受来自控制命令输出装置的控制命令的状态中。

4. 根据权利要求 1 所述的再现装置，其特征在于，

在播放器装置的状态从接受控制命令和再现内容数据的状态转变到另一个状态时进行备份操作。

5. 根据权利要求 1 所述的再现装置，其特征在于，

在播放器装置的状态从另一状态转变到接受控制命令和再现内容数据的状态时进行恢复操作。

6. 根据权利要求 1 所述的再现装置，其特征在于，

在根据再现控制程序的指令与状态转变一起进行恢复，利用备份在第二存储装置中的再现状态信息恢复内容数据的再现的情况下，在恢复之后放弃备份在第二存储装置中的再现状态信息。

7. 根据权利要求 1 所述的再现装置，其特征在于，

在根据再现控制程序再现内容数据的指令，播放器装置转变成接受控制命令和再现内容数据的状态的情况下，放弃备份在第二存储装置中的再现状态信息。

8. 根据权利要求 1 所述的再现装置，其特征在于，

将有关是否放弃备份在第二存储装置中的再现状态信息的信息加入再现控制程序让播放器装置停止内容再现的指令中。

9. 一种再现记录在记录媒体中的内容数据的再现方法，其特征在于，

根据播放器装置的四种状态，控制播放器装置按照从记录媒体中读取的再现控制程序进行内容数据的再现，其中，第一存储装置存储在播放器装置再现期间指示播放器装置的状态的再现状态信息，而第二存储装置备份存储在第一存储装置中的再现状态信息，并且与播放器装置的四种状态之间的状态转变一起，将存储在第一存储装置中的再现状态信息备份在第二存储装置中，并将备份在第二存储装置中的再现状态信息恢复到第一存储装置中；并且其中，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序，而这四种状态是通过组合根据是否再现内容数据分类的第一状态和第二状态和根据是否接受与用户操作相对应的控制命令分类的第三状态和第四状态定义的，其中在接受来自控制命令输出装置的控制命令的该第三状态中，不接受脚本层的脚本程序，而在不接受来自控制命令输出装置的控制命令的该第四状态中，仅仅接受来自脚本层的控制命令，并且

其特征在于，由从脚本层发送的控制命令而不是与用户操作对应的控制命令引起状态转变。

再现装置和再现方法

技术领域

[0001] 本发明涉及可以容易地控制记录在大容量记录媒体中的节目的再现的再现装置、再现方法、再现程序、记录媒体和数据结构。

[0002] 背景技术

[0003] 自 DVD（数字多功能盘）作为随机存取和可换式记录媒体出现在人们面前以来已经有很长时间了。近年来，容量比 DVD 大和携带起来比 DVD 方便的盘状记录媒体正在开发之中。

[0004] 人们已经将 DVD-Video 规定为将视频内容记录在 DVD 中的标准。根据 DVD-Video，处在盘再现操作中的 DVD 播放器可以呈现各种状态。在 DVD-Video 下，DVD 播放器可以呈现的状态分类成五种所谓的域，并且在 DVD 播放器中使用根据各种条件在域之间转变的模型。具体地说，可以将 DVD 域当作具有五种可能值的状态变量。DVD 播放器监视状态变量，以便掌握正在从盘中读取的内容的类型。

[0005] 在 DVD-Video 中定义了如下五种域：

[0006] (1) 第一播放域 (FP_DOM)；

[0007] (2) 视频管理器菜单域 (VMGM_DOM)；

[0008] (3) 视频标题设置菜单域 (VTSM_DOM)；

[0009] (4) 标题域 (TT_DOM)；以及

[0010] (5) 停止状态。

[0011] 顺便提一下，(5) 中的停止状态不是真正的域。

[0012] (1) 中的第一播放域被定义成盘的第一部分和在 DVD 播放器中再现的预备状态。非再现命令被认为是有效的。(2) 中的视频管理器菜单域被定义成整个盘或整个盘面上的主菜单和指示标题菜单的显示。与菜单有关的命令被认为是有效的。(3) 中的视频标题设置菜单域被定义成标题或标题组的菜单或子菜单（子画面、语言、音频或角度）和指示根菜单或子菜单。(4) 中的标题域被定义成标题中的视频内容，并且再现命令被认为是有效的。(5) 中的停止状态被定义成读写头离开盘再现的位置和返回到原始位置的状态，并且再现命令被认为是有效的。

[0013] 控制 DVD 播放器操作的导航命令受用于当前状态的域限制。只要显示菜单，例如，从菜单中选择预定项目的 SELECT BUTTON 命令是有意义的。具体地说，SELECT BUTTON 命令在 (2) 的视频管理器菜单域或 (3) 的视频标题设置菜单域的状态下是有意义的。只要播放器处在静止状态或正在显示由静止图像构成的菜单屏幕，例如，指定速度比正常单一速度高的视频再现的 FAST FEED 命令是无意义的。换句话说，FAST FEED 命令在标题域中是有意义的。

[0014] DVD-Video 的这些域描述在例如“Jim Taylor：New Book on DVD Anatomy，1st edition，Nexus Intercom Ltd.，June 7，2003，p.271”（非专利文件 1）中。

[0015] 上述的传统 DVD-Video 存在许多域类型和域间状态转变的详细条件，从而造成了不能轻易安装在 DVD 播放器上的问题。

[0016] 另一个问题是，除非完全理解域之间的转变，否则，盘制作是不可能的，以及内容制作者难以制作盘。具体地说，由于存在许多域类型和域间转变的详细条件，不能完全掌握域间的转变。这给制作配置复杂盘的制作者带来沉重负担。

[0017] 此外，域名和实际操作方法未必相互一致，因此，域必须存在的意义不大。这造成了许多域类型，联同域间转变的详细条件，构成了增加制作者制造盘的负担的因素。

[0018] 在例如如上所述的 DVD-Video 中，定义了标题域和两种菜单域（视频管理器菜单域和视频标题设置菜单域）。最初，应该在标题域中再现记录在 DVD 中的主内容（譬如，电影原始故事）。但是，实际上，在菜单域中再现电影原始故事也不会造成什么问题。尽管对于域间的转变，这些命令是不同的，但是，在状态转变到这些域之后，播放器操作在菜单域和标题域之间没有什么区别。这是由于菜单域中的菜单再现和标题域中的电影原始故事再现都是通过将内容数据和相关再现控制程序组合在一起的称为 PGC（程序链）的公用逻辑结构实现的。

[0019] 这就造成了其它自由内容再现受到限制的问题。具体地说，将给定内容确定为菜单还是标题有些取决于盘制作者的主观观点。假设在当前再现电影原始故事存在分支的情况下，例如，通过显示选择任何一个分支的选择按钮 构想应用交互性的内容。在这种情况下，使用传统 DVD-Video 的方法对于将菜单域还是将标题域用于再现选择按钮显示的内容是不明确的。在希望预备的情况下，在特定内容之后，难以预测具有交互性的内容，例如，状态转变，并且难以核实操作。

[0020] 发明内容

[0021] 于是，本发明的目的是提供可以肯定地确定播放器操作的状态转变和便于交互式内容再现的再现装置、再现方法、再现程序、记录媒体和数据结构。

[0022] 为了解决上述问题，如权利要求 1 所述的发明是再现记录在记录媒体中的内容数据的再现装置，它包含：读取装置，用于从记录媒体中读取数据，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序；播放器装置，用于按照再现控制程序再现内容数据；以及控制命令输出装置，用于将与用户操作相对应的控制命令给予播放器装置，其特征在于，播放器装置根据四种状态控制内容数据的再现，这四种状态是通过组合根据是否再现内容数据分类的第一状态和第二状态和根据是否接受来自控制命令输出装置的控制命令分类的第三状态和第四状态定义的，其中在接受来自控制命令输出装置的控制命令的该第三状态中，不接受脚本层的脚本程序，而在不接受来自控制命令输出装置的控制命令的该第四状态中，仅仅接受来自脚本层的控制命令，并且其特征在于，由从脚本层而不是从控制命令输出装置发送的控制命令引起状态转变。

[0023] 如权利要求 12 所述的发明是再现记录在记录媒体中的内容数据的再现方法，其特征在于，根据播放器装置的四种状态控制播放器装置按照从记录媒体中读取的再现控制程序的内容数据再现，其中，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序，而这四种状态是通过组合根据是否再现内容数据分类的第一状态和第二状态和根据是否接受与用户操作相对应的控制命令分类的第三状态和第四状态定义的，其中在接受来自控制命令输出装置的控制命令的该第三状态中，不接受脚本层的脚本程序，而在不接受来自控制命令输出装置的控制命令的该第四状态中，仅仅接受来自脚本层的控制命令，并且其特征在于，由从脚本层发送的控制命令而不是

与用户操作对应的控制命令引起状态转变。

[0024] 如权利要求 13 所述的发明是使计算机系统执行再现记录在记录媒体中 的内容数据的再现方法的再现程序，其特征在于，使再现方法根据播放器装置的四种状态控制播放器装置按照从记录媒体中读取的再现控制程序的内容数据再现，其中，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序，并且这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对应的控制命令分类的两种状态定义的。

[0025] 如权利要求 14 所述的发明是其中记录着使计算机系统执行再现记录在记录媒体中的内容数据的再现方法的再现程序的计算机可读记录媒体，其特征在于，使再现方法根据播放器装置的四种状态控制播放器装置按照从记录媒体中读取的再现控制程序的内容数据再现，其中，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序，并且这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对应的控制命令分类的两种状态定义的。

[0026] 如权利要求 15 所述的发明是其中记录着包括视频流和音频流中的至少一种的内容数据和使播放器装置控制内容数据再现的再现控制程序的计算机可读记录媒体，其特征在于，以指令控制内容数据再现的播放器装置根据四种状态控制内容数据再现的方式执行再现控制程序，这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对应的控制命令分类的两种状态定义的。

[0027] 如权利要求 18 所述的发明是包含包括视频流和音频流中的至少一种的内容数据和使播放器装置控制内容数据再现的再现控制程序的数据结构，其特征在于，以将再现控制指令给予根据四种状态控制内容数据再现的播放器装置的方式执行再现控制程序，这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对应的控制命令分类的两种状态定义的。

[0028] 如权利要求 19 所述的发明是再现记录在记录媒体中的内容数据的再现装置，它包含 ：读取单元，用于从记录媒体中读取数据，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序 ；播放器单元，用于按照再现控制程序播放内容数据 ；以及控制命令输出单元，用于将与用户操作相对应的控制命令给予播放器单元，其特征在于，播放器单元根据四种状态控制内容数据的再现，这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受来自控制命令输出单元的控制命令分类的两种状态定义的。

[0029] 如上所述，根据如权利要求 1、12、13、14 和 19 所述的发明，根据播放器装置的四种状态控制播放器装置按照从记录媒体中读取的再现控制程序进行的内容数据再现，其中，记录媒体中记录着包括视频流和音频流中的至少一种的内容数据和控制内容数据再现的再现控制程序，而这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对 应的控制命令分类的两种状态定义的。因此，使播放器装置的状态数减少了，并且使状态转变的操作便于理解，从而在减轻内容制作的负担的同时，使播放器便于安装。

[0030] 根据如权利要求 15 所述的发明，提供了其中记录着包括视频流和音频流中的至少一种的内容数据和使播放器装置控制内容数据再现的再现控制程序的计算机可读记录

媒体，其中，再现控制程序将再现控制指令给予根据四种状态控制内容数据再现的播放器装置，从而控制内容再现，其中，这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对应的控制命令分类的两种状态定义的。因此，甚至可以容易地制作需要复杂控制操作的内容。

[0031] 根据如权利要求 18 所述的发明，提供了包含包括视频流和音频流中的至少一种的内容数据和使播放器装置控制内容数据再现的再现控制程序的数据结构，其中，将再现控制指令给予根据四种状态控制内容数据再现的播放器装置，从而控制内容再现，其中，这四种状态是通过组合根据是否再现内容数据分类的两种状态和根据是否接受与用户操作相对应的控制命令分类的两种状态定义的。因此，甚至可以容易地制作需要复杂控制操作的内容，并且作为数据结构提供。

[0032] 根据本发明，将再现播放清单的电影播放器的状态定义成包括从播放清单再现的观点来看的两种状态，即，停止状态和播放状态、和根据是否接受与用户操作相对应的控制命令的两种状态，即，正常模式和菜单模式，并且通过这四种状态之间的状态转变控制播放清单再现。其结果是，本发明具有可以根据脚本程序控制 AV 流再现的优点。

[0033] 另一个优点是电影播放器的状态数少并且状态定义清楚。因此，易于理解产生状态转变的条件和产生状态转变的操作，从而便于安装再现 AV 流的播放器。

[0034] 此外，电影播放器的状态数少并且状态定义清楚。因此，易于理解产生状态转变的条件和产生状态转变的操作，从而使内容制作者更易于制作具有交互性的内容。

附图说明

[0035] 图 1 是示出 UMD 视频标准的播放器结构的示意图；

[0036] 图 2 是示意性地示出根据本发明实施例的示范性播放器模型的图形；

[0037] 图 3 是示出电影播放器的内容配置的示意图；

[0038] 图 4 是说明电影播放器的播放状态和停止状态的图形；

[0039] 图 5 是说明根据本发明实施例的电影播放器的事件模型的示意图；

[0040] 图 6 是示出在播放清单再现期间生成的示范性事件的示意图；

[0041] 图 7A 和 7B 是示出电影播放器对象拥有的示范性特性的清单的示意图；

[0042] 图 8 是示出电影播放器对象的示范性方法的清单的示意图；

[0043] 图 9 是示出用户的示范性键输入操作的示意图；

[0044] 图 10 是示出用户的示范性键输入操作的示意图；

[0045] 图 11A、11B 和 11C 是与键输入相对应的示范性控制命令的示意图；

[0046] 图 12 是示出与键输入相对应的示范性事件的示意图；

[0047] 图 13 是示出示范性事件管理器的示意图；

[0048] 图 14 是示出示范性事件管理器的示意图；

[0049] 图 15 是示出将用户输入事件作为动机准备的程序执行的示范性处理的流程图；

[0050] 图 16 是说明示范性脚本程序的图形；

[0051] 图 17 是示出示范性脚本程序的示意图；

[0052] 图 18 是示出应用于 UMD 视频标准的文件的示范性管理结构的示意图；

[0053] 图 19 是示出指示文件“PLAYLIST.DAT”的整个结构的示范性语法的示意图；

[0054] 图 20 是示出块“PlayItem()”的示范性内部结构的示意图；

[0055] 图 21 是示出块“PlayListMark()”的示范性内部结构的示意图；

[0056] 图 22 是说明块“Mark()”中的字段“mark_type”的图形；

[0057] 图 23 是说明片段 AV 流文件中的标记时间的指定的图形；

[0058] 图 24 是示出指示片段 AV 流文件“XXXXX.CLP”的整个结构的示范性语法的示意图；

[0059] 图 25 是说明块“StreamInfo()”与基本流之间的关系的图形；

[0060] 图 26 是示出块“StaticInfo()”的示范性内部结构的示意图；

[0061] 图 27 是示出块“DynamicInfo()”的示范性内部结构的示意图；

[0062] 图 28 是示出块“EP_map()”的示范性内部结构的示意图；

[0063] 图 29 是示意性地示出本发明可以应用的盘再现装置的示范性配置的方 块图；

[0064] 图 30A 和 30B 是更详细说明盘再现装置的操作的功能方块图；

[0065] 图 31 是示出根据本发明的电影播放器的状态定义的示意图；

[0066] 图 32 是示出当前状态和通过组合成电影播放器的四种状态的每一种的方法状态转变之后的状态的示意图；

[0067] 图 33A、33B、33C、33D 和 33E 是说明执行方法“play()”时电影播放器的状态转变例子的示意图；

[0068] 图 34 是说明再现播放项的方法的示意图；

[0069] 图 35 是示出在播放清单再现期间到达播放清单的开始点和结束点时电影播放器的示范性操作的示意图；

[0070] 图 36 是说明播放清单之间的再现的示意图；

[0071] 图 37 是更详细示出在播放清单的末端脚本层中的处理流程和电影播放器的示范性操作的流程图；

[0072] 图 38 是说明 UMD 视频播放器的三种类型存储区的示意图；

[0073] 图 39 是说明播放器状态的备份的示意图；

[0074] 图 40 是说明播放器状态的备份的示意图；

[0075] 图 41 是说明重新开始信息的恢复和放弃的示意图；

[0076] 图 42 是说明重新开始信息的恢复和放弃的示意图；

[0077] 图 43 是说明重新开始信息的恢复和放弃的示意图；

[0078] 图 44 是说明重新开始信息的恢复和放弃的示意图；

[0079] 图 45 是示出 UMD 视频播放器使用方法“stop()”的变元“resumeInfo_ClearFlag”的示范性操作的示意图；

[0080] 图 46 是示出播放器状态的示范性寿命的示意图；

[0081] 图 47A 和 47B 是示出重新开始信息的示范性寿命的示意图；以及

[0082] 图 48 是示出用户数据的示范性寿命的示意图。

具体实施方式

[0083] 下面按如下顺序说明本发明的实施例：

[0084] 1. UMD 视频标准

[0085] 2.UMD 视频标准的播放器模型

[0086] 3. 电影播放器的事件模型

[0087] 4. 电影播放器对象

[0088] 5. 脚本程序的例子

[0089] 6. 文件管理结构

[0090] 7. 盘再现装置

[0091] 8. 电影播放器的状态转变模型

[0092] 8-1 电影播放器状态的定义

[0093] 8-2 产生电影播放器的状态转变的方法

[0094] 8-3 播放清单再现期间电影播放器的操作

[0095] 8-4 电影播放器的再现恢复功能

[0096] 8-5 每个数据的寿命

[0097] 1.UMD 视频标准

[0098] 首先，为了便于理解，简要说明可用作本发明实施例的系统。根据本发明的实施例，利用称为 ECMA 脚本的脚本语言描述播放器模型。ECMA 脚本是 ECMA（欧洲计算机制造商协会）创建的基于 JavaScript（注册商标）的跨平台的脚本语言。由于它与 HTML 文本的关系非常密切和可以定义它自己的对象，ECMA 脚本适用于根据本发明的播放器模型。

[0099] 具体地说，传统 DVD-Video 使用通过 DVD-Video 标准定义的非通用命令来描述实现交互式功能的控制程序。控制程序被嵌入多个文件中，嵌入数据文件的多个点上，或分布式地嵌入 AV 流文件中。执行如此嵌入的控制程序的条件和顺序通过 DVD 标准定义。

[0100] 在这样的 DVD-Video 系统中，难以构建通用内容制作系统，因此，根据预定脚本制作故事，即，利用模板制作内容。另一方面，在制作无法用模板处理的复杂内容时，第一步是预备作为定制系统的内容制作系统。根据本发明的实施例，这个问题是利用 ECMA 语言 - 可扩充性高的通用脚本语言控制 AV 内容解决的。

[0101] 在如下的描述中，将利用基于 ECMA 脚本的脚本语言的基于本发明实施例的标准称为 UMD（通用媒体盘（注册商标））视频标准。此外，将 UMD 视频标准与脚本有关的部分特别称为 UMD 视频脚本标准。

[0102] 下面简要说明 UMD 视频脚本。图 1 示出了 UMD 视频标准的层结构。在 UMD 视频标准中，定义了包括脚本层、播放清单层和片段层的三层结构，并 且根据这种结构进行流管理。

[0103] 在 UMD 视频标准中，将数字编码视频和音频数据和字幕当作作为 MPEG2（运动图像专家组 2）分组基本流的多路复用 MPEG2 流来管理。多路复用成 MPEG2 流的视频和音频数据和字幕的基本流被称为片段 AV 流。片段 AV 流存储在片段 AV 流文件中。在记录片段 AV 流文件的同时，与片段 AV 流文件一一对应地准备片段信息文件。将一组片段信息文件和相应片段 AV 流文件称为片段。

[0104] 片段是所谓的盘记录单位，并且通过比片段高的播放清单层管理再现片段的顺序。播放清单层用于指定片段再现路径，它包括一个或多个播放清单。播放清单是一堆播放项。播放项包括指示片段再现范围的一组“In”点和“Out”点。通过连接播放项，可以以任何顺序再现片段。播放项可以通过复制片段来指定。片段 AV 流文件中的“In”和“Out”

点通过时间印记（片段内时间）指定，时间印记通过片段信息文件中的信息转换成片段 AV 流上的字节位置。

[0105] 播放清单只具有依次再现指定整个或一部分片段的播放项的结构，只利用播放清单不能实现再现顺序的转移或与用户的交互性。根据本发明的实施例，将多个播放清单集中成单个文件“PLAYLIST.DAT”。

[0106] 脚本层由从作为语言规定的 ECMA 脚本延伸出来的 UMD 视频脚本构成。在 UMD 视频脚本中，根据 ECMA 脚本加入实现 UMD 视频特有的功能的扩展。

[0107] 脚本层高于播放清单层，并且具有播放清单再现指令和设置播放器的命令串。通过脚本层的命令可以实现带有条件转移的播放清单再现，譬如，选择为多种语言准备的任何一种流，或将再现流改变到根据某个条件选择的播放清单。利用带有这种条件转移的播放清单再现的应用例子是多故事。这种脚本层引入了与用户交互的功能。

[0108] 根据本发明的这个实施例，脚本层具有称为资源文件的文件。资源文件包括根据实际 ECMA 脚本描述的脚本数据（脚本程序）、在操作按钮时输出效果声音的声音数据、和含有用于菜单屏幕的背景图像的图像数据和显示像按钮图标那样的 GUI 部分的图像数据（位图数据）的屏幕设计。

[0109] 可以存在多个资源文件。此外，根据本发明的这个实施例，给予资源文件以根据预定命名规则的文件名。例如，文件名的扩展名“RCO”指示该文件是资源文件。

[0110] 2. UMD 视频标准的播放器模型

[0111] 接着说明按照 UMD 视频标准再现数据的再现装置（播放器）的模型，即，播放器模型。播放器首先从盘中读取资源文件、播放清单文件和片段信息文件，并且按照预定再现顺序，读取片段 AV 流文件和再现视频和音频数据和字幕。

[0112] 根据脚本程序的语言规定，将再现播放清单的功能块作为对象安装在脚本程序中。根据 UMD 视频标准，将用于播放清单再现的对象称为电影播放器对象。播放清单再现指令和设置播放器的命令构成与电影播放器对象相关联的方法。电影播放器对象通过来自脚本控层器的方法控制。在该处理中，需要由电影播放器对象将状态变化和再现位置通知脚本层的功能。这对应于电影播放器对象将事件发送到脚本程序，并且将与这个事件相对应的处理描述成事件管理器。

[0113] 如上所述，构建通过事件将信息从电影播放器对象发送到脚本程序，并通过一种方法根据脚本程序控制电影播放器对象，以便可以通过脚本程序控制片段 AV 流再现的模型。

[0114] 图 2 示意性地示出了如上所述的根据本发明这个实施例的示范性播放器模型。电影播放器 300 是负责根据 UMD 视频标准再现视频和音频数据和字幕的模块。上述的电影播放器对象是形成脚本程序中的对象的电影对象，以便操作来自脚本程序的电影对象。换句话说，电影播放器对象是实现抽象成可以根据脚本程序管理的形式的电影播放器的功能的实现模块。

[0115] 顺便提一下，电影播放器 300 和电影播放器对象被认为代表基本相同的对象，因此，下面用同一标号表示。

[0116] 在图 2 中，在电影播放器 300 中，读取片段 AV 流文件，并且按照来自通过用户输入 301 引发的较低层（在图 2 的情况中，本机实现平台 301）或构成较高层的脚本层 302 的方

法，根据播放清单和片段信息的数据库解码和显示如此读取的片段 AV 流。

[0117] 电影播放器对象 300 的内部依赖于播放 UMD 视频的 UMD 视频播放器的安装，并且由脚本层 302 以黑箱的形式提供像方法和特性那样的 API（应用程序编码接口）作为对象。UMD 视频播放器指示电影播放器安装在上面的实际设备。所有 UMD 视频播放器都含有按照 UMD 视频标准安装在上面的电影播放器，并且具有再现兼容性。

[0118] 如图 2 所示，电影播放器对象 300 含有三条输入 / 输出路径，包括从本机实现平台 301 接收控制命令 311 的路径、将事件 312 通知脚本层 302 的路径、和从脚本层 302 接收方法 313 的路径。

[0119] 来自本机实现平台 301 的控制命令 311 控制电影播放器对象 300 的操作。本机实现平台 301 是例如设备专用的部分与作为实际设备的 UMD 视频播放器中的电影播放器 300 之间的接口。来自电影播放器 300 的事件 312 是脚本层 302 的脚本事件。来自脚本层 302 的脚本程序的方法 313 由电影播放器 300 指定。

[0120] 电影播放器对象 300 包括其中包含 UMD 视频标准的播放清单和片段信息的数据库 320。电影播放器对象 300 使用户输入 310 失效（屏蔽用户输入 310），或执行利用数据库 320 将指定时间再现位置转换成片段 AC 流中的字节位置的处理。

[0121] 电影播放器对象 300 中的回放模块 321 解码构成多路复用成 MPEG2 PS（节目流）的视频和音频数据和字幕的片段 AV 流。回放模块 321 具有播放和停止两种状态，并且按照控制指令和方法在这两种状态之间转变（图 3）。顺便提一下，片段 AC 流不局限于 MPEG2 PS。例如，如果将 MPEG2 TS（传输流）用作片段 AV 流，也可以以与模型相似的方式管理 MPEG2 TS。

[0122] 脚本层 302 根据 UMD 视频脚本标准执行脚本程序，以便控制和在屏幕上显示电影播放器对象 300。脚本层 302 起实现内容制作者这一方想要的情节的作用。脚本层 302 将方法 313 发送到电影播放器对象 300，并且接收来自电影播放器对象 300 的事件 312。脚本层 302 与本机实现平台 301 交换与用户输入 310 相对应的键事件 314 或指令本机实现平台 301 绘制屏幕的方法 315。

[0123] 除了在 UMD 视频标准中规定的之外，本机实现平台 301 还具有其它各种功能。根据本发明的这个实施例，存在使脚本层 302 起本机实现平台 301 作用的方法 315。因此，也为本机实现平台 301 定义了抽象其功能的对象，并且认为该方法 315 与脚本程序上的特定对象相关联。这是因为该方法属于该对象。因此，在本机实现平台 301 中定义控制器对象 330，并且将方法 315 定义成控制器对象 330 的方法。

[0124] 例如，排列在菜单屏幕上的按钮由本机实现平台 301 根据从脚本层 302 的脚本程序传递到本机实现平台 301 的方法 315 绘制。每当用户操作按钮加 以选择或确定（输入）时，与用户输入 310 相对应的键事件 314 由本机实现平台 301 通知脚本层 302，并且脚本层 302 中的脚本程序根据键事件 314 执行与键输入 310 相对应的处理。

[0125] 如上所述，这些功能以由电影播放器 330 负责解码视频、音频和字幕和控制视频、音频和字幕的显示的操作的方式分配，并且由脚本层 302 执行与构成像按钮那样的 GUI（图形用户界面）的部分图像（下文称为 GUI 部分）的安排或显示和选择或确定 GUI 部分的操作有关的处理。

[0126] 本机实现平台 301 形成操作电影播放器对象 300 和脚本程序的基础。在实际 UMD

视频播放器是例如硬件的情况下，以硬件特有的方式安装本机实现平台 301，以便起硬件和播放器模型之间的中介作用。

[0127] 例如，本机实现平台 301 接收来自用户的用户输入 310，并且确定如此接收的用户输入 310 是到电影播放器 300 的指令还是发送给绘制或显示在脚本层 302 中的按钮的指令。在确定用户输入 310 是到电影播放器 300 的指令的情况下，本机实现平台 301 将用户输入 310 转换成构成到电影播放器 300 的内部控制指令的控制命令 311，并且将控制指令发送到电影播放器 300。

[0128] 另一方面，在确定用户输入 310 是到绘制或显示在脚本层中的 GUI 部分的指令的情况下，本机实现平台 301 将与用户输入 310 相对应的键事件 314 通知脚本层 302。然后，例如，可以根据脚本层 302 按照特定键事件 314 指定的方法 315，在屏幕上显示按钮图标 3。具体地说，本机实现平台 301 和脚本层 302 可以不用电影播放器 300 的中介，直接接收和传送事件和方法。

[0129] 此外，本机实现平台 301 可以像后面所述那样存取电影播放器 300 的特性和观看电影播放器 300 的状态。

[0130] 接着，更详细地说明电影播放器 300。图 3 示出了电影播放器 300 的示范性内部配置。如上所述，电影播放器 300 具有数据库 320 和回放模块 321。数据库 320 是存储从盘中读取的有关播放清单的信息和有关片段的信息，即，片段信息的区域。

[0131] 回放模块 321 包括解码器引擎 322 和代表指示回放模块 321 的状态的值的特性 323。像例如语言代码那样的特性 323 具有包括特性 323A（只读参数）和特性 323B（播放器状态）的两种类型，特性 323A 的值通过初始化电影播放器 300 确定，并且特性 323B 的值随回放模块 321 的状态而改变。

[0132] 通过初始化确定的特性 323A 的值由像实际设备那样的本机系统设置，而不能通过播放清单、片段信息或脚本程序改变。特性 323A 的值被认为只能从脚本程序中读取。另一方面，指示回放模块 321 的状态的特性 323B 的值一方面可以从脚本程序中读取和另一方面可以写入一些脚本程序中。

[0133] 顺便提一下，这种操作模型假设在再现片段 AV 流之前从盘中预装播放清单和片段信息。但是，作为一种替代，可以为其它安装（插件）实现通过电影播放器模型确定的操作。

[0134] 电影播放器对象 300 按照来自脚本层 302 或本机实现平台 301 的指定再现播放清单。例如，电影播放器 300 参考数据库 320 和获取与指定播放清单相对应的片段 AV 流的再现位置，作为文件中的字节位置。在回放模块 321 中，解码器引擎 322 根据再现位置信息控制片段 AV 流的解码。

[0135] 如图 4 所示的电影播放器 300 按照播放清单再现条件存在播放和停止两种状态。播放状态是指定和再现播放清单的状态。除了正常再现之外，播放状态还包括各种状态，即，像倍速、半速、快进或快倒和暂停那样的各种速度再现。按帧向前或向后再现的所谓逐帧前进是暂停状态和播放状态交替的状态。停止状态是不再现播放清单的状态。在停止状态下，不选择播放清单，并且不确定指示“当前再现播放清单号”的播放器状态值。

[0136] 例如，电影播放器 300 的状态伴随着电影播放器 300 中的解码器引擎 322 中的播放状态和停止状态的状态转变，并且按照解码器引擎 322 的状态转变更新特性 323B 的值。

[0137] 重新开始信息 324 存储正好在停止状态之前的状态。在电影播放器 300 在播放状态下解码给定播放清单的情况下，例如，到停止状态的转变使正好在停止状态之前的状态被存储下来。可以以可通过每个盘标题识别的方式将重新开始信息 324 的多个项目存储在作为硬件的播放器的非易失性存储器中。例如，盘含有与每个盘标题有关的唯一标识信息（下文称为标题 ID），并且与标题 ID 相关联地存储重新开始信息 324。这样，根据重新开始信息 324 的信息，可以从正好在具有与标题 ID 相对应的标题的盘可能从其转变到播放状态的停止状态之前的位置开始再现盘。

[0138] 3. 电影播放器的事件模型

[0139] 现在说明电影播放器 300 的事件模型。电影播放器 300 在播放播放清单的播放状态下生成各种事件。这些事件引发描述在脚本中的称为事件管理器的处理程序的执行。事件管理器是通过事件的生成存取的方法。根据事件的 生成开始执行处理程序的程序执行模型被称为事件驱动模型。在事件驱动模型中，生成不规则事件，并且执行将事件生成作为动机准备的程序。根据本发明的这个实施例，脚本程序利用事件管理器组控制电影播放器对象 300 的操作。

[0140] 图 5 示意性地显示了根据本发明实施例的电影播放器 300 的事件模型。在图 5 中，事件管理器“onEventA()”、“onEventB()”和“onEventC()”是接口，并且每个事件管理器的内容描述在脚本中。事件管理器的内容由例如内容制作者准备和安装。根据 UMD 视频脚本标准，为电影播放器对象 300 通知脚本程序的每个事件准备事件管理器。在图 5 的情况中，例如，在生成事件 A 时执行的处理程序被确定为事件管理器“onEventA()”，对于事件 B 和事件 C，情况也是这样。具体地说，在生成事件 B 时，执行相应事件管理器“onEventB()”，而在生成事件 C 时，执行相应事件管理器“onEventC()”。

[0141] 按照事件生成存取的事件管理器由系统选择，因此，不需要内容制作者在脚本程序中描述确定可能生成的特定事件的处理。

[0142] 图 6 示出了在播放清单再现期间生成的示范性事件。在播放清单“PlayList”的首端，设置章节标记“ChapterMark”。因此，在从播放清单的首端开始再现时，生成与章节标记相对应的事件“Chapter”。此外，每当章节改变时，将事件“Chapter”通知脚本层 302 和执行相应事件管理器“onChapter”。此外，随着再现到事件标记“EventMark”设置的时间，生成相应标记事件。当再现到播放清单的末端时，在播放清单的末端暂时停止再现，并且由电影播放器 300 将事件“PlayListEnd”通知脚本层 302。在脚本层 302 中，在相应事件管理器“onPlayListEnd()”内指定再现另一个播放清单的首端。这样，以内容制作者想要的顺序连续再现一系列播放清单。

[0143] 如上所述，假设在操作播放器期间发生各种事件，并且通过将事件生成通知上层程序，上层程序可以掌握播放器状态。在上层程序中，准备为生成各种事件提供通知每个事件生成时执行的程序（事件管理器）。事件和事件管理器将在后面作更详细说明。

[0144] 在内容制作者未描述事件管理器的情况下，执行按标准规则的内置在播放器中的操作（默认事件管理器），或忽略事件和什么也执行。在要求什么也执行的情况下，不描述与事件相对应的事件管理器，从而断然忽略事件。

[0145] 其它可能事件模型包括由播放器对象中的对象登记与给定事件相对应的 收听器，并且在在播放器对象中生成的事件是登记事件的情况下，将事件从播放器对象发送到

其中记录了特定事件的对象和由该对象执行相应方法的事件收听器模型、和与可能生成的事件无关地存取单种方法的单方法模型。

[0146] 根据本实施例的事件模型比需要处理事件登记和取消事件登记的事件收听器模型简单。另一方面，单方法模型需要在方法中描述了解生成的特定事件和切换为每个事件预备的处理例程的预处理。该方法由内容制作者安装，因此，尽管像模型那样简单，但给内容制作者带来沉重负担。此外，每当生成一个事件，就存取一个大处理程序（方法），因此，在执行速度下降的同时，占据了大量存储区。从这个观点来看，根据本发明的这个实施例为每个事件准备处理程序（事件管理器）的模型被认为是有利的。

[0147] 4. 电影播放器对象

[0148] 接着，说明电影播放器对象 300 的外部规定。一般说来，通过基于 ECMA 脚本语言规定的语言定义的对象含有特性和方法。如已经参照图 2 和 3 所述、根据本发明这个实施例的电影播放器对象 300 类似地含有特性和方法。可以通过指定所涉及的对象名和特性名直接从外部对象中读取和写入特性。此外，通过定义设置特性值的方法“setXXX()”（“XXX”是所涉及的特性名）和读取特性值的方法“getXXX()”，可以通过该方法读取和写入其它对象的特性。

[0149] 图 7A 和 7B 示出了电影播放器对象 300 拥有的示范性特性的清单。这对应于图 3 中的特性 323。图 7A 示出了与如图 3 所示的只读参数 323A 相关联的示范性特性。特性“ScriptVersion”指示 UMD 视频脚本的版本。特性“audio_ChannelCapability”指示 UMD 视频播放器可以再现的音频声道的数量。特性“languageCode”指示设置在 UMD 视频播放器中的菜单显示语言的语言代码。特性“audiolanguageCode”指示设置在 UMD 视频播放器中的音频语言的语言代码。另一方面，特性“subtitlelanguageCode”指示设置在 UMD 视频播放器中的字幕语言的语言代码。

[0150] 一旦将盘装入，就根据设置在只读参数 323A 中的特性“languageCode”所指的语言代码确定从盘中读取的脚本文件。在装入盘缺乏与该语言相对应的脚本文件的情况下，读取默认脚本文件。例如，在所有多个脚本文件当中，读取安装在盘首端的文件作为默认脚本文件。

[0151] 图 7B 示出了与图 3 中的播放器状态 323B 相关联的示范性特性。特性“PlayListNumber”指示当前再现播放清单的号码。特性“chapterNumber”指示当 前再现章节的号码。特性“videoNumber”指示当前再现视频流的号码。特性“audioNumber”指示当前再现音频流的号码。特性“subtitleNumber”指示当前再现字幕流的号码。特性“palyListTime”指示假设播放清单首端是 0 的时间。特性“AudioFlag”指示音频再现的 on/off 和 dual mono LR 的指定。特性“subtitleFlag”指示字幕显示的 on/off。

[0152] “dual mono”是将左右 (L,R) 立体声音频声道的每一个用作各自独立的单声音频声道的模式。

[0153] 在电影播放器 300 正在再现操作或暂停的情况下，存在有关与这个播放器状态 323B 相关联的每个特性的信息。一旦操作转变到停止状态，将与特定时刻的播放器状态 323B 相关联的每个特性备份成重新开始信息（重新开始信息）324。同时，可以清除播放器状态 323B 的内容。

[0154] 图 8 示出了保存在电影播放器对象中的示范性方法的清单。这对应于如图 2 所示

的方法313。方法“play()”再现视频。方法“playChapter()”通过指定章节再现视频。方法“resume()”利用重新开始信息324开始再现。方法“stop()”停止视频再现。方法“pause()”暂时停止视频再现。方法“play_Step()”逐帧前进地再现视频。方法“changeStream()”改变视频流、音频流和/或字幕流。方法“getPlayerStatus()”获取电影播放器300中像再现、停止或暂停那样的状态。方法“changeResumeInfo()”改变重新开始信息324的内容。方法“reset()”停止视频再现和清除重新开始信息324的内容。

[0155] 根据UMD视频标准,可以将视频显示在显示屏的一部分中。如下四种方法与这种视频显示有关。方法“setPos()”设置视频显示位置。方法“getPos()”获取视频显示位置。方法“setSize()”设置视频显示尺寸。方法“getSize()”获取视频显示尺寸。

[0156] 实际上,电影播放器300和本机实现平台301构成一个整体。具体地说,按照作为实际将盘装入加以再现的硬件的UMD播放器与控制UMD播放器的软件之间的关系,硬件处理的特定部分和软件处理的特定部分依赖于安装时的配置。在UMD播放器由例如个人计算机构成的情况下,可以将除盘驱动器之外的其它部分配置成软件。另一方面,在UMD播放器被配置成简单单元的情况下,可以将除了盘驱动器之外的视频解码器和音频解码器配置成硬件。因此,在电影播放器300和本机实现平台301之间处理的方法、命令和事件不局限于像其例子显示在图2中那样的显式交换。

[0157] 另一方面,关于来自用户的键输入,如已经参照图2所述的那样,用户输入310首先被本机实现平台301接收。具体地说,本机实现平台301接收来自用户的键输入,作为用户输入310,并且确定用户输入310是到电影播放器300的命令还是有关电影播放器302的脚本程序的事件。按照确定结果,本机实现平台301生成控制命令311或键事件314,并且将它们通知相应上层(电影播放器300或脚本层302)。

[0158] 图9和10示出了通过用户输入310的示范性键输入。顺便提一下,图9和10中从“VK”开始的每个键都指示虚拟键。

[0159] 图9示出了操作电影播放器301的示范性键输入。键“VK_PLAY”提供与指定再现的再现键相对应的功能。键“VK_STOP”提供与指定停止再现的停止键相对应的功能。键“VK_PAUSE”提供与指定暂时停止再现的暂停键相对应的功能。键“VK_FAST_FORWARD”提供与指定快进再现的快进键相对应的功能。键“VK_FAST_REVERSE”提供与指定快倒再现的快倒键相对应的功能。键“VK_SLOW_FORWARD”提供与指定慢进再现的慢(进)键相对应的功能。键“VK_SLOW_REVERSE”提供与指定慢倒再现的慢(倒)键相对应的功能。键“VK_STEP_FORWARD”提供与指定逐帧向前再现的逐帧(向前)键相对应的功能。键“VK_STEP_REVERSE”提供与指定逐帧向后再现的逐帧(向后)键相对应的功能。

[0160] 键“VK_NEXT”提供与输入指示“下一个”的值的下一个指定键相对应的功能。键“VK_PREVIOUS”提供与输入指示“前一个”的值的前一个指定键相对应的功能。通过使用键“VK_NEXT”和“VK_PREVIOUS”,例如,可以指定到前后章节的转换。

[0161] 键“VK_ANGLE”提供与指定多角度视频的角度切换的视频角度切换键相对应的功能。键“VK_SUBTITLE”提供与切换英语字幕、日语字幕或显示/不显示字幕的字幕切换键相对应的功能。键“VK_AUDIO”提供与切换像环绕或双语那样的音频设置的音频切换键相对应的功能。键“VK_VIDEO_ASPECT”提供与指定视频宽高比切换的宽高比切换键相对应的功能。

[0162] 图 10 示出了菜单操作的的示范性键输入。键“VK_UP”提供与输入指示“向上”的值的向上指定键相对应的功能。键“VK_DOWN”提供与输入指示“向下”的值的向下指定键相对应的功能。键“VK_RIGHT”提供与输入指示“向右”的值的向右指定键相对应的功能。键“VK_LEFT”提供与输入指示“向左”的值的向左指定键相对应的功能。键“VK_UP_RIGHT”提供与输入指示“右上方”的值的右上方指定键相对应的功能。键“VK_UP_LEFT”提供与输入指示“左上方”的值的左上方指定键相对应的功能。键“VK_DOWN_RIGHT”提供与输入指示“右下方”的值的右下方指定键相对应的功能。键“VK_DOWN_LEFT”提供与输入指示“左下方”的值的左下方指定键相对应的功能。这些键的使用使在屏幕上指定光标移动成为可能。

[0163] 键“VK_MENU”提供与显示菜单的菜单键相对应的功能。键“VK_ENTER”提供与指定“确定”的 ENTER 键相对应的功能。键“VK_RETURN”提供与单步指定处理返回的键相对应的功能。

[0164] 键“VK_COLORED_KEY_1”、“VK_COLORED_KEY_2”、“VK_COLORED_KEY_3”、“VK_COLORED_KEY_4”、“VK_COLORED_KEY_5”和“VK_COLORED_KEY_6”分别提供与着色功能键 1、着色功能键 2、着色功能键 3、着色功能键 4、着色功能键 5 和着色功能键 6 相对应的功能。

[0165] 如图 9 所示的键输入和如图 10 所示的键输入在功能上是不同的，因此，需要通过本机实现平台 301 指定通知目的地。如上所述，与视频、音频和字幕的再现有关的指定通过如图 9 所示的键输入发出。一旦接收到作为用户输入 310 的如图 9 所示的键输入，本机实现平台 301 就将接收的键输入转换成如图 11A、11B 和 11C 所示的命令，并且将它们通知电影播放器 300。

[0166] 另一方面，如图 10 所示的键输入是用于 GUI 的用户输入 310，因此，需要将这个用户输入通知构成屏幕和安排按钮的脚本层 302 和由它处理。一旦接收到作为用户输入 310 的如图 10 所示的键输入，本机实现平台 301 就将它转换成如图 2 所示的事件 314，并且将它通知脚本层 302。图 12 示出了与这个键输入相对应的事件 314 的例子。

[0167] 图 9 和 10 还示出了像键 VK_ANGLE、键 VK_SUBTITLE 和键 VK_AUDIO 那样用于流切换的键输入。首先通过用户输入 310 将这些键通知电影播放器 300，据此将指示存在来自电影播放器 300 的流切换请求的事件通知脚本。然后，通过从脚本程序到电影播放器 300 的流切换方法切换音频和字幕。因此，需要将这个键输入从本机实现平台 301 发送到电影播放器 300。

[0168] 现在更详细地描述如图 11A、11B 和 11C 所示的命令。命令“uo_time_Search(playListTime)”指定从指定时间开始再现当前再现的播放清单。变元“playListTime”指示播放清单的首端被设置成 0 的时间。这个命令不能指定播放清单号，因此，变元“playListTime”所指的时间是包括在当前再现播放清单的范围内的指定时间。命令“uo_play()”指定以单一（正常）速度开始再现。开始位置根据重新开始信息 324 确定。在缺乏与重新开始信息 324 相对应的信息的情况下，认为用户操作是无效的。这个命令对应于执行未指定播放清单号的方法“play()”。此外，不能通过用户操作按这个命令指定播放清单号。

[0169] 命令“uo_playChapter(chapterNumber)”指定从通过当前再现播放清单的变量“chapterNumber”指定的章节开始再现。在缺乏章节指定的情况下，指定从当前再现

章节的首端开始再现。这对应于未指定章节号的方法“play_Chapter()”。命令“uo_playPrevChapter()”指定从正好在当前章节之前的章节开始再现。另一方面，命令“uo_playNextChapter()”指定从下一个章节开始再现。

[0170] 命令“uo_jumpToEnd()”指定跳转到播放清单的末端。这个命令对应于指令电影播放器300中止再现和生成事件“playListEnd”的用户操作。按照这个命令，脚本层302执行事件管理器“onPlayListEnd”。命令“uo_for_wardScan(speed)”指定以通过变元“speed”指定的再现速度向前再现。另一方面，命令“uo_backwardScan(speed)”指定以通过变元“speed”指定的再现速度向后再现。命令“uo_forwardScan(speed)”和命令“uo_backward_Scan(speed)”中的变元“speed”依赖于UMD视频播放器的安装。

[0171] 命令“uo_playStep(forward)”指定向前逐帧前进。命令“uo_playStep(backward)”指定向后逐帧前进。命令“uo_pauseOn()”指定根据用户操作暂时停止再现。命令“uo_pauseOff()”根据用户操作取消暂时停止再现。

[0172] 命令“uo_setAudioEnabled(boolean)”指定音频流的on/off操作。在执行这个命令时，标志“audioFlag”的值也改变成相应内容。命令“uo_set_SubtitleEnabled(boolean)”指定字幕流的on/off状态。在执行这个命令时，标志“subtitleFlag”的值也改变成相应内容。命令“uo_angleChange()”指定改变显示角度。一旦将根据这个命令的用户操作发送到电影播放器300，电影播放器300就将事件“angleChange”通知脚本层302。命令“uo_audio_Change(audioStreamNumber)”指定改变要再现的音频流。命令“uo_change_AudioChannel(value)”指定dual mono再现时切换音频信道或切换单个信道。在执行这个命令时，标志“audioFlag”的值也改变成相应内容。命令“uo_ subtitleChange(subtitleStreamNumber)”指定改变要再现的字幕流。

[0173] 现在更详细说明如图12所示的事件与有关这些事件的电影播放器300的方法之间的关系。事件“menu”跳转到菜单。本机实现平台301不将这个事件通知电影播放器300，而是通知脚本层302。一旦接收到这个事件“menu”，脚本层302就执行事件管理器“onMenu”。当本机实现平台301结束UMD视频应用程序时，本机实现平台301发送事件“exit”。一旦脚本层302接收到这个事件“exit”，脚本层302就执行事件管理器“onExit”。

[0174] 一旦产生资源文件切换，就从本机实现平台301发送事件“resource_Changed”。一旦脚本层302接收到这个事件“resourceChanged”，脚本层302就执行事件管理器“onResourceChanged”。

[0175] 在聚焦构成显示在屏幕上的GUI部分的按钮图标的情况下，生成事件“up”、事件“down”、事件“left”、事件“right”、事件“focusIn”、事件“focusOut”、事件“push”和事件“cancel”。本机实现平台301不将这些事件通知电影播放器300，而是通知脚本层302。顺便提一下，措词“聚焦按钮图标的情况”指示在屏幕上指定位置的光标指示按钮图标的显示坐标和特定按钮图标处在可选状态下的状态。当按钮图标的焦点向上、向下、向左和向右移动时，分别生成事件“up”、事件“down”、事件“left”、和事件“right”。在聚焦给定按钮图标的情况下，生成事件“focusIn”，并且在受到聚焦的按钮图标被移到焦点外的情况下，生成事件“focusOut”。此外，在按下受到聚焦的按钮图标的情况下，生成事件“push”。在取消按钮图标的按下的情况下，生成事件“cancel”。

[0176] 事件“autoPlay”和事件“continuePlay”指定在脚本层302中开始执行脚本。事

件“autoPlay”指定装入盘时脚本自动开始。事件“continuePlay”指定根据重新开始信息324，从以前中止的时刻，例如，在装入盘的时候重新开始执行脚本。

[0177] 对于如图12所示的事件，存在一旦生成事件就执行的程序。与这个事件相对应的程序被称为事件管理器。事件与事件管理器之间的对应关系可以通过附上例如名称来建立。举例来说，将“on”附在事件名的首端获得事件管理器名。图13和14示出了示范性事件管理器。事件管理器的内容由内容制作者描述。

[0178] 图13示出了电影播放器对象300和相应事件管理器拥有的部分事件例子。如图13所示的事件对应于如图2所示的上述事件312，并且由电影播放 器300通知脚本层302。事件管理器是一种接口，它的内容由内容制作者利用例如脚本语言安装。通过以这种方式配置事件管理器，可以在发生事件的时候实现内容制作者想要的操作。

[0179] 一旦检测到事件标记(Event_mark)，就执行事件“mark”和事件管理器“onMark()”。事件标记被嵌入例如播放清单中，由电影播放器300在再现播放清单时检测。电影播放器300一旦检测到事件标记，电影播放器300就将事件“mark”通知脚本层302。脚本层302执行与这个事件“mark”相对应的事件管理器“onMark()”。类似地，在结束播放清单时，执行事件“playListEnd”和事件管理器“onplayListEnd()”。一旦检测到章节标记(Chapter_mark)，就执行事件“chapter”和事件管理器“onchapter()”。章节标记被嵌入例如播放清单中，由电影播放器300在再现播放清单时检测。

[0180] 一旦通过用户操作指定了角度改变，就执行事件“angleChange”和事件管理器“onAngleChange()”。在按照用户操作将键输入“VK_ANGLE”输入本机实现平台301中作为用户输入310的情况下，例如，本机实现平台310将特定用户输入310转换成命令“uo_angleChange()”，并且将它传送到电影播放器300。电影播放器300按照命令“uo_angleChange()”生成事件“angleChange”，并且将它传送到脚本层302。脚本层302执行与事件“angle_Change”相对应的事件管理器“onAngleChange()”。类似地，一旦通过用户操作指定了音频改变，就执行事件“audioChange”和事件管理器“on_AudionChange()”。一旦通过用户操作指定了字幕改变，就执行事件“sub_titleChange”和事件管理器“onSubtitleChange()”。

[0181] 图14示出了控制器对象330拥有的部分示范性事件管理器。如图14所示的事件管理器与本机实现平台301的控制器对象330相关联，并且通过本机实现平台301发给脚本层302的通知执行。

[0182] 事件“menu”和事件管理器“onMenu()”跳转到菜单。例如，一旦通过用户操作按下菜单键，本机实现平台301就将事件“menu”通知脚本层302。一旦接收到这个事件，脚本层302就执行相应事件管理器“onMenu()”，并且在事件管理器“onMenu()”下安排和显示构成菜单屏幕的GUI部分。事件“exit”和事件管理器“onExit()”分别是本机实现平台301结束UMD视频应用程序时从本机实现平台301发出的事件和相应事件管理器。

[0183] 一旦通过用户操作等指定了结束UMD视频播放器操作时，就由本机实 现平台301将事件“exit”通知脚本层302。一旦接收到如此通知的事件“exit”，脚本层302的脚本就可以执行事件管理器“onExit()”中的结束处理。

[0184] 事件“resourceChanged”和事件管理器“onResourceChanged”分别是本机实现平台301切换了资源文件之后从本机实现平台301发出的事件和相应事件管理器。

[0185] 事件“autoPlay”和事件管理器“onAutoPlay()”、和事件“continue_Play”和事件管理器“onContinuePlay()”分别开始执行脚本。

[0186] 顺便提一下，除了控制器对象 330 的事件管理器之外，还存在按钮的事件管理器。按钮的事件管理器与本发明没有密切联系，因此，不作描述。

[0187] 现在参照图 15 的流程图，简要说明执行将用户输入事件作为动机准备的程序的示范性处理。图 15 示出了在用户在 UMD 视频播放器正常再现盘期间按下指定再现下一个章节的键（例如，“next”键）的情况下，在屏幕上显示准备消息的同时，按照这个键输入跳转到下一个章节开始再现的例子。

[0188] 在用户在 UMD 视频播放器正常再现盘期间利用 UMD 视频播放器的遥控命令器按下键“next”（步骤 S10）的情况下，例如，将键 VK_NEXT 作用户输入 310 传送到本机实现平台 301。本机实现平台 301 生成与这个用户输入 310 相对应的用户命令“uo_playNextChapter()”（步骤 S11）。将这个用户命令“uo_playNextChapter()”通知电影播放器 300。

[0189] 接收到这个命令“uo_playNextChapter()”的电影播放器 300 搜索数据库 320，并且根据当前再现的位置从播放清单信息中获取下一个章节标记的位置（步骤 S12）。步骤 S13 确定下一个章节标记是否存在，并且一旦确定不存在这样的章节标记，不作章节跳转地继续进行当前再现。

[0190] 另一方面，一旦在步骤 S13 中确定存在下一个章节标记，处置转到步骤 S14。在步骤 S14 中，电影播放器 300 中止当前再现，并且从数据库 320 的片段信息文件的特征点信息中获取片段 AV 流文件中指示下一个章节标记的字节位置。在步骤 S15 中，存取文件中的获得字节位置，并且通过开始从特定位置中读取数据流开始再现。

[0191] 步骤 S16 和随后的步骤是在屏幕上显示通知章节切换的消息的一系列处理。一旦章节被切换和从该章节的首端开始再现，就生成章节事件（步骤 S16）。例如，由电影播放器 300 检测安排在章节首端的章节标记，并且生成事件“chapter”。由电影播放器 300 将这个章节事件通知脚本层 302。在通知这个事 件的时候，电影播放器 300 还将要跳转的章节的章节号通知脚本层 302。脚本层 302 开始执行像事件管理器“onChapter()”那样与所通知事件相对应的事件管理器（步骤 S17）。

[0192] 在受到考虑的情况下，假设在事件管理器中描述了在屏幕上指示通知章节切换（如果有的话）的消息的操作。脚本层 302 的脚本执行这个事件管理器，获取事件生成时电影播放器 300 通知的跳转目的地的章节号（步骤 S18），并且将指令发送给本机实现平台 301，以便在屏幕上显示指示例如获得的章节号的章节的首端的预定消息。对这个指令作出响应，本机实现平台 301 将消息显示在屏幕上（步骤 S19），并且结束事件管理器的处理（步骤 S20）。

[0193] 通过指定开始再现下一个章节的键“next”经过上述处理的用户操作，使章节跳转，并且在开始再现跳转到的下一个章节时，在屏幕上显示指示该章节首端的消息。

[0194] 如上所述，用户输入事件构成了改变电影播放器 300 的状态和生成新事件的动机，并且利用新生成的事件进行各种各样的处理。

[0195] 上述的播放器模型使再现视频、音频和字幕成为可能。通过在再现期间的给定时刻生成某个事件和执行像内容制作者预置的那样事先准备的事件管理器，可以实现内容制

作者想要的操作。此外，在用户在再现播放清单期间操作播放器的情况下，按照通过用户操作的用户输入310将控制命令从本机实现平台301发送到电影播放器300，并且可以像用户想要的那样改变播放器状态。此外，接收到通过播放器的用户操作的用户输入310的本机实现平台301将事件通知脚本层302的脚本，以便可以按照用户操作执行内容制作者准备的操作。

[0196] 通过以这种方式构建播放器模型，可以向用户提供视频、音频和字幕的再现和交互式操作。

[0197] 5. 脚本程序的例子

[0198] 接着，说明脚本层302的示范性脚本程序。首先，假设由内容制作者准备如图16所示的内容再现的流程。图16配有包括播放清单400，401、顶层菜单402和消息403的显示单元。播放清单400在装入盘时自动显示警告文件屏幕。播放清单401是构成这个内容的特征的电影的原始故事。顶层菜单屏幕402含有像安排成指定再现播放清单401的按钮那样的GUI部分。此外，消息403在再现播放清单401期间的任何时刻显示出来。

[0199] 此外，如图16所示的配置含有几个事件管理器。一旦将盘装入UMD播放器中，事件管理器“onAutoPlay()”就自动再现播放清单400和显示报警消息。在再现播放清单结束时，存取事件管理器“onPlayListEnd()”，并且在如图16所示的例子中，在播放清单400或401结束时存取事件管理器“onPlayListEnd()”。具体地说，事件管理器“onPlayListEnd()确定完成了哪个播放清单，并且一旦完成了播放清单400的再现，指定开始再现播放清单401。此外，在再现播放清单401结束时，事件管理器“onPlay_ListEnd()”存取顶层菜单屏幕402。

[0200] 一旦用户操作菜单键，就存取事件管理器“onMenu()”，并且由事件管理器“onMenu()”存取和在屏幕上显示顶层菜单402。在再现期间到达标记“Mark”指定的时间时执行事件管理器“onMark()”。在如图16所示的情况下，为播放清单401设置标记“Mark”，并且在播放清单401的再现已到标记“Mark”指定的时刻时，在屏幕上显示消息403。

[0201] 具体地说，在图16的情况下，一旦将盘装入UMD视频播放器中，就存取事件管理器“onAutoPlay”，并且在显示报警屏幕的同时再现播放清单400。在随着再现时间的流逝已到播放清单400的末端时，存取事件管理器“on_PlayListEnd()”，并且确定再现到播放清单400的末端。因此，再现下一个播放清单401。一旦用户在再现播放清单401期间操作菜单键，存取事件管理器“onMenu()”，并且显示顶层菜单屏幕402。此外，按照顶层菜单屏幕402上的预定操作，事件管理器“onMenu()”从播放清单401的首端开始再现。此外，在播放清单401的再现时间已到标记“Mark”指定的时刻时，存取事件管理器“onMark()”，并且在屏幕上显示消息403。一旦再现到播放清单401的末端，存取事件管理器“onPlayListEnd()”，并且确定再现到播放清单400的末端，从而显示顶层菜单屏幕402。

[0202] 图17示出了实现如图16所示的操作的脚本程序的例子。如上所述，脚本程序含有安排的事件管理器，并且按照事件的发生执行它们的任何一个。脚本程序存储在带有扩展名“RCO”的资源文件中。

[0203] 对电影播放器300指定播放清单再现的方法是“movieplayer.play()”。在括号中，将要再现的播放清单的号码描述成变元。一旦完成了播放清单的再现，就生成事件“playListEnd”。一旦生成事件“playListEnd”，就从脚本中检索“movieplayer.

onPlayListEnd()”。在处理过程中，将对象“event_info”与事件“playListEnd”一起传送给脚本。对象“event_info”中存储着指示完成的特定播放清单的播放清单号等。取决于对象“event_info”的内容，可以在这个脚本中改变下一个操作。

[0204] 6. 文件管理结构

[0205] 接着，参照图 18 说明根据 UMD 视频标准使用的文件管理结构。文件按目录结构分层管理和记录在盘上。盘文件系统可应用于通过 ISO（国际标准化组织）-9660 或 UDF（通用盘格式）定义的文件系统。

[0206] 在根目录下，存放着文件“TITLEID.DAT”和目录“VIDEO”。在目录“VIDEO”下，存放着目录“RESOURCE”、目录“CLIP”、目录“STREAM”和文件“PLAY_LIST.DAT”。

[0207] 文件“TITLEID.DAT”用于存储对于每个标题（内容类型）不同的标题标识符。每个盘保存一个文件“TITLEID.DAT”。

[0208] 在目录“RESOURCE”下，存放着资源文件（“JA000000.RCO”）。如上所述，资源文件中存储着构成脚本层 302 的脚本程序、和像包括图像数据和声音数据的部分数据那样用于构成菜单屏幕的数据。在目录“RESOURCE”下，通常存放一个资源文件。不过，在目录“RESOURCE”下也可以存放多个资源文件。在准备显示语言不同的多个菜单时，为每种语言准备多个资源文件。此外，在这种情况下，每次只使用一个资源文件。

[0209] 在资源文件的文件名中，接在构成定界符的黑点之后的扩展名被固定成“RCO”，从而指示特定文件是资源文件。此外，黑点之前的字符串简要指示特定资源文件的内容。例如，资源文件的整个文件名应用格式“CCdannnn.RCO”，前两个字符“CC”指示与资源文件相对应的语言代码，接着一个字符“d”指示用于指示语言代码是否是默认语言的标志，接着的“a”指示显示屏的宽高比，并且接着的四个字符“nnnn”指示标识号。标识号以在多个资源文件中不包括相同文件名的方式确定。

[0210] 通过制定以这种方式命名资源文件的规则，可以通过资源文件的文件名确定源数据的语言属性和显示屏的宽高比。在选择资源文件时，根据文件名确定适当资源文件。

[0211] 在目录“CLIP”下，存放着一个或多个片段信息文件。在片段信息文件中，文件名由构成定界符的黑点之前包括像“00001”那样的五个到几个字符的字符串（在这种情况下，数字字符）和黑点之后的扩展名“CLP”确定。扩展名“CLP”使标识特定文件是片段信息文件成为可能。

[0212] 在目录“STREAM”下，存放着一个或多个片段 AV 流文件。片段 AV 流文件具有由构成定界符的黑点之前像“00001”那样的五个到几个字符的字符串（在这种情况下，数字字符）和黑点之后的扩展名“PS”构成的文件名。扩展名“PS”使确定特定文件是片段 AV 流文件成为可能。根据本实施例，片段 AV 流文件被存储成包括视频流、音频流和字幕流的多路复用文件，并且通过扩展名“PS”标识成 MPEG2（运动图像专家组）的节目流。

[0213] 如上所述，通过压缩编码和时分多路复用视频数据和音频数据获取片段 AV 流文件，以便通过读取这个文件和执行解码处理，可以获得视频数据和音频数据。此外，片段信息文件用于描述片段 AV 流文件的特性和与片段 AV 流文件相对应。根据本实施例，在片段信息文件和相应片段 AV 流文件中，使文件名中扩展名之前包括五个到几个字符的字符串保持一致，从而可以容易地掌握两者之间的对应关系。

[0214] 资源文件包含包含如上所述的脚本程序的描述的脚本文件，并且其中存储着用于

使根据本实施例的盘再现具有交互性的程序。资源文件在存储在盘中的其它文件之前读取。

[0215] 文件“PLAYLIST.DAT”是包含指定片段 AV 流的再现顺序的播放清单的描述的播放清单文件。下面参照图 19 到 21，说明文件“PLAYLIST.DAT”的内部结构。图 19 示出了示出文件“PLAYLIST.DAT”的整个结构的示范性语法。在这种情况下，根据作为程序描述语言用于计算机系统等的 C 语言描述方法显示语法。这同样适用于如图所示的其它语法。

[0216] 字段“name_length”具有 8 个位的数据长度，指示附在播放清单文件上的名称的长度。字段“name_string”具有 255 个字节的长度，指示附在播放清单文件上的名称。在字段“name_string”中，从其首端开始具有字段“name_length”所指的字节长度的部分用作有效名。例如，在字段“name_length”具有值“10”的情况下，将字段“name_string”从其首端开始 10 个字节的部分解释成有效名。

[0217] 字段“number_of_PlayLists”具有 16 个位的数据长度，指示连续描述块“PlayList()”的数量。描述了数量等于下一行的“for”循环中的字段“number_of_PlayLists”所指的次数的块“PlayList()”。块“PlayList()”是播放清单本身。

[0218] 下面说明块“PlayList()”的示范性内部结构。在块“PlayList()”的首端，安 排了字段“PlayList_data_length”。字段“PlayList_data_length”具有 32 个位的数据长度，指示包括字段“PlayList_data_length”的块“PlayList()”的数据长度。然后，安排了具有 15 个位的数据长度的字段“reserved_for_word_alignment”和具有 1 个位的数据长度的字段“capture_enable_flag_PlayList”。字段“reserved_for_word_alignment”与具有 1 个位的数据长度的标志“capture_enable_flag_PlayList”组合用于保证块“PlayList()”中 16 个位的位置上的对齐。

[0219] 标志“capture_enable_flag_PlayList”指示是否允许二次使用与包括“capture_enable_flag_PlayList”的块“PlayList()”相关联的动态图像。例如，在标志“capture_enable_flag_PlayList”的值是“1”的情况下，指示允许在再现装置中二次使用与块 PlayList() 相关联的动态图像。

[0220] 尽管在上述情况中认为标志“capture_enable_flag_PlayList”是 1 位标记，但标志“capture_enable_flag_PlayList”不局限于此。例如，标志“capture_enable_flag_PlayList”可以由多个位构成，以便描述逐步允许二次使用。举例来说，标志“capture_enable_flag_PlayList”由 2 个位构成，并且在值是“0”的情况下，完全禁止二次使用，而在值是“1”的情况下，在将数据压缩编码成像 64 像素 ×64 行那样的预定分辨率或更低分辨率的条件下允许二次使用。此外，在值是“2”的情况下，可以没有任何限制地允许二次使用。作为其它选择，在两个位的位 0 是“1”的情况下，在内容再现的应用中允许二次使用，而另一方面，在位 1 的值是“1”的情况下，允许二次使用在同一容器内的其它应用中（譬如，壁纸图像或屏幕保护程序）。在这样的情况下，可以将位 0 和位 1 的值结合在一起使用。

[0221] 字段“PlayList_name_length”具有 8 个位的数据长度，指示附在块“PlayList()”的名称的长度。字段“PlayList_name_string”具有 25 5 个位的数据长度，指示附在块“PlayList()”的名称。字段“PlayList_name_string”从其首端开始具有字段“PlayList_name_length”所指的字节长度的部分用作有效名。

[0222] 字段“number_of_PlayItems”具有 16 位的数据长度，指示连续描述块

“PlayItem()”的数量。接着描述了数量等于下一行的“for”循环中的字段“number_of_PlayItems”所指的次数的块“PlayItem()”。块“PlayItem()”是播放项本身。

[0223] 将标识信息(ID)附在块“PlayList()”中的每个块“PlayItem()”上。例如，按“PlayItem()”出现的顺序，将号码0附在描述在块“PlayList()”中的第1块“PlayItem()”上，后面接着序号1,2,...。这些序号用作每个块“PlayItem()”的标识信息。重复了与块“PlayItem()”的数量相同的次数的“for”循环的变元“i”可以用作相应块“PlayItem()”的标识信息。块“PlayItem()”的后面接着块“PlayListMark()”。

[0224] 参照图20，说明块“PlayItem()”的示范性内部结构。在块“PlayItem()”的首端，安排了字段“length”。字段“length”具有16个位的数据长度，指示特定块“PlayItem()”的长度。然后，安排了字段“Clip_Information_file_name_length”。字段“Clip_Information_file_name_length”具有16个位的数据长度，指示与块“PlayItem()”相对应的片段信息文件的名称的长度。 字段“Clip_Information_file_name”含有按字节的数据长度变量，指示与块“PlayItem()”相对应的片段信息文件的名称。包括其首端具有字段“Clip_Information_file_name_length”所指的字节长度的字段“Clip_Information_file_name”用作有效名。一旦通过字段“Clip_Information_file_name”指定了片段信息文件，就可以根据上述文件名的对应关系规定与特定片段信息文件相对应的片段AV流文件。

[0225] 字段“IN_time”和字段“OUT_time”的每一个都具有33个位的数据长度，代表指定与块“PlayItem()”中的字段“Clip_Information_file_name”指定的片段信息文件相对应的片段AV流文件的再现开始和结束位置的时间信息。通过使用字段“IN_time”和字段“OUT_time”的信息，可以从除了片段AV流文件的首端之外的其它部分中指定再现开始位置。类似地，可以指定不在片段AV流文件的末端上的再现结束位置。字段“reserved_for_word_alignment”用于将数据结构的数据长度调整成16个位的整数倍和具有15个位的数据长度。

[0226] 参照图21，说明块“PlayListMark()”的示范性内部结构。在块“PlayListMark()”的首端，安排了字段“length”。字段“length”具有32个位的数据长度，指示块“PlayListMark()”的长度。然后，安排了字段“number_of_PlayList_marks”。字段“number_of_PlayList_marks”具有16个位的数据长度，指示连续块“Mark()”的数量。接着描述了数量等于下一行的“for”循环中的字段“number_of_PlayList_marks”所指的次数的块“Mark()”。

[0227] 下面描述块“Mark()”的示范性内部结构。在块“Mark()”的首端，安排了字段“mark_type”。字段“mark_type”具有8个位的数据长度，指示包括特定字段“mark_type”的块“Mark()”的类型。根据本实施例，像其例子显示在图22中那样，定义了包括章节标记和事件标记的两种标记。章节是划分播放清单(块“PlayList()”)的搜索单元，并且章节标记用时间信息指示章节位置。另一方面，事件标记生成标记事件。

[0228] 字段“mark_name_length”具有8个位的数据长度，指示附在块“Mark()”上的名称的长度。位于在块“Mark()”底行的字段“mark_name_string”指示附在块“Mark()”上的名称。字段“mark_name_string”包括其首端具有字段“mark_name_length”所指的字节长度的部分用作有效名。

[0229] 包括字段“ref_to_PlayItem_id”、字段“mark_time_stamp”、字段“entry_ES_stream_id”和字段“entry_ES_private_id”的四个元素建立定义在块“PlayList()”上的块“Mark()”与片段AV流文件之间的对应关系。具体地说，字段“ref_to_PlayItem_id”具有16个位的数据长度，指示块“Play_Item()”的标识信息。其结果是，规定了片段信息文件和片段AV流文件。

[0230] 具有33个位的数据长度的字段“mark_time_stamp”用于指定片段AV流文件中的标记的时间。参照图23，简要说明这个字段。在图23中，假设播放清单由分别指定成号码0、1和3的三个播放项（播放项(#0)、播放项(#1)、和播放项(#3)）构成，并且播放清单上的时间 t_0 包括在号码1的播放项（播放项(#1)）中。此外，假设号码0、1和2的播放项通过相应片段信息文件分别对应于片段AC流文件的节目流A、B和C。

[0231] 在这样的情况下，在在播放清单上的时刻 t_0 上指定标记的情况下，字段“ref_to_PlayItem_id”的值被设置成指示播放项包括时间 t_0 的“1”，并且在字段“mark_time_stamp”中描述与相应片段AC流文件B上的时间 t_0 相对应的时间。

[0232] 回到图21，字段“mark_time_stamp””的后面安排了字段“entry_ESstream_id”和字段“entry_ES_private_id”。字段“entry_ES_stream_id”和字段“entry_ES_private_id”分别具有8个位的数据长度，并且在块“Mark()”与规定基本流相关联的情况下，用于规定特定基本流。字段“entry_ES_stream_id”和字段“entry_ES_private_id”分别指示与相应基本流多路复用的分组“packet()”的流ID“stream_id”和专用分组首标“private_packet_header()”的专用流ID“stream_id”。

[0233] 顺便提一下，分组“packet()”的流ID“stream_id”和专用分组首标“private_packet_header()”的专用流ID“stream_id”都基于例如MPEG2系统的节 目流的规则。

[0234] 字段“entry_ES_stream_id”和字段“entry_ES_private_id”用在例如片段AV流#0和片段AV流#1具有不同章节配置的情况中。在相应块“Mark()”不与规定基本流相关联的情况下，认为这两个字段的值是“0”。

[0235] 接着，参照图24到28说明片段信息文件的内部结构。如上所述，片段信息文件“XXXXX.CLP”描述处在目录“STREAM”下的相应片段AV流文件“XXXXX.PS”的特性。

[0236] 图24示出了指示片段AV流文件“XXXXX.CLP”的整个结构的语法例子。片段AV流文件“XXXXX.CLP”包括安排在其首端的字段“presentation_start_time”和字段“presentation_end_time”。字段“presentation_start_time”和字段“presentation_end_time”的每一个都具有33个位的数据长度，指示相应片段AV流文件首端和末端的时间。时间信息可以使用MPEG2系统中的PTS(演播时间标记)。PTS具有90kHz的精度。

[0237] 接着，安排了具有7个位的数据长度的字段“reserved_for_word_align_ment”和具有1个的数据长度的标志“capture_enable_flag_Clip”。字段“reserved_for_word_alignment”与具有1个位的数据长度的标志“capture_enable_flag_Clip”组合用于将文件“XXXXX.CLP”中的排列与16个位的位置对齐。标志“capture_enable_flag_Clip”指示是否允许二次使用包括在与文件“XXXXX.CLP”相对应的片段AV流文件中的动态图像。例如，在标志“capture_enable_flag_Clip”的值是“1”的情况下，指示允许在再现装置中二次使用与文件“XXXXX.CLP”相对应的片段AV流文件的动态图像。

[0238] 字段“number_of_streams”具有8个位的数据长度，指示连续块“Stream_Info()”

结构的数量。在字段“number_of_streams”的后面，描述了数量等于“for”循环中的字段“number_of_streams”所指的次数的块“Stream_Info()”。在“for”循环之后，安排了块“EP_map()”。

[0239] 下面说明块“StreamInfo()”的示范性内部结构。在块“StreamInfo()”的首端，安排了字段“length”。字段“length”具有 16 个位的数据长度，指示块“StreamInfo()”的长度。然后，安排了每一个具有 8 个位的数据长度的字段“stream_id”和字段“private_stream_id”。像例子显示在图 25 中那样，块“StreamInfo()”与基本流相关联。在如图 25 所示的情况中，块“StreamInfo()”通过字段“stream_id”的值“0×E0”到“0×EF”与视频流相关联，并且通过值 “0×BD”与ATRAC（自适应变换声音编码）音频流、LPCM（线性脉冲码调制）音频流或字节流相关联。此外，块“StreamInfo()”分别通过字段“private_stream_id”的值“0×00”到“0×0F”、“0×10”到“0×0F”和“0×80”到“0×9F”与 ATRAC 音频流、LPCM 音频流和字幕流相关联。

[0240] 在图 25 中的值的表达中，“0×”指示用十六进制表示法表达随后的数值。这同样适用于下面的类似表面。

[0241] 块“StreamInfo()”大体上用包括在流中不改变的信息和在流中改变的信息的两种信息描述。在流中不改变的信息描述在块“StaticInfo()”中。另一方面，在流中改变的信息描述在改变点通过时间信息指定的块“Dynamic_Info()”中。

[0242] 在块“StreamInfo()”中，具有 8 个位的数据长度的字段“reserved_for_word_alignment”被安排成将字节位置与块“StaticInfo()”的后部对齐，字段“reserved_for_word_alignment”的后面接着字段“number_of_Dynamic_Info()”。字段“number_of_DynamicInfo()”具有 8 个位的数据长度，指示此后描述在块“StreamInfo()”中的块“DynamicInfo()”的数量。字段“pts_change_point”和块“DynamicInfo()”通过“for”循环中的字段 number_of_DynamicInfo()”所指的次数描述。

[0243] 字段“pts_change_point”具有 33 个位的数据长度，通过 PTS 指示确认相应块“DynamicInfo()”的信息的时间。对于每种数据流，也通过等于上述定义在文件“XXXXX.CLP”中的字段“presentation_start_time”的字段“pts_change_point”指示代表首端的时间。

[0244] 参照图 26，说明块“StaticInfo()”的示范性内部结构。块“StaticInfo()”的内容随相应基本流的类型而变。相应基本流的类型可以根据参照图 25 所述的字段“stream_id”和字段“private_stream_id”的值确定。在图 26 中，块“StaticInfo()”利用“if”语法描述相应基本流的类型是视频流、音频流还是字幕流。现在，针对每种基本流说明块“StaticInfo()”。

[0245] 在基本流是视频流的情况下，块“StaticInfo()”由每一个具有 4 个位的数据长度的字段“picture_size”和字段“frame_rate”和具有 1 个位的数据长度的标志“cc_flag”构成。字段“picture_size”和字段“frame_rate”分别指示视频流的图像尺寸和帧频率。标志“cc_flag”指示视频流是否包括封闭字幕。例如，在标志“cc_flag”的值是“1”的情况下，特定视频流包括封闭字幕。字段“reserved_for_word_alignment”用于将数据排列与 16 个位对齐。

[0246] 在基本流是音频流的情况下，块“StaticInfo()”包括具有 16 个位的数据长度的

字段“audio_language_code”、具有 8 个位的数据长度的字段“channel_configuraion”、具有 1 个位的数据长度的标志“lfe_existence”、和具有 4 个位的数据长度的字段“sampling_frequency”。字段“audio_language_code”指示包括在特定音频流中的语言的代码。字段“channel_configuraion”指示像单声、立体声或多信道那样音频数据的信道属性。标志“lfe_existence”指示是否包括低频强调信道，在值是，比如说，“1”的情况下，包括低频强调信道。字段“sampling_frequency”指示音频数据的取样频率。字段“reserved_for_word_alignment”用于将数据排列与 16 个位对齐。

[0247] 另一方面，在基本流是字幕流的情况下，块“StaticInfo()”包括具有 16 个位的数据长度的字段“subtitle_language_code”、和具有 1 个位的数据长度的标志“configurable_flag”。字段“subtitle_language_code”指示包括在字幕流中的语言的代码。标志“configurable_flag”指示对于字幕流显示，是否允许改变字符大小或位置，或者，例如，对于“1”的值，允许改变字符大小或位置。字段“reserved_for_word_alignment”用于将数据排列与 16 个位对齐。

[0248] 参照图 27，说明块“DynamicInfo()”的示范性内部结构。块“Dynamic_Info()”包括安排在其首端的具有 8 个位的数据长度的字段“reserved_for_word_alignment”。随后的内容随相应基本流的类型而变。相应基本流的类型可以根据参照图 25 所述的字段“stream_id”和字段“private_stream_id”的值确定。在图 27 中，块“DynamicInfo()”包含利用“if”语法确定相应基本流的类型是视频流、音频流还是字幕流的描述。现在，针对每种基本流说明块“DynamicInfo()”。

[0249] 在基本流是视频流的情况下，块“DynamicInfo()”包括具有 4 个位的数据长度的字段“display_aspect_ratio”。字段“display_aspect_ratio”指示视频显示输出的宽高比是 16 ： 9 还是 4 ： 3。字段“reserved_for_word_alignment”用于将数据排列与 16 个位对齐。

[0250] 在基本流是音频流的情况下，块“DynamicInfo()”包括具有 4 个位的数据长度的字段“channel_assignemt”。在音频流由两个信道构成的情况下，字段“channel_assignemt”指示输出是立体声还是单声。双单声状态用于，例如，两种语言的音频再现。字段“reserved_for_word_alignment”用于将数据排列与 16 个位对齐。

[0251] 在基本流是字幕流的情况下，块“DynamicInfo()”由用于将数据排列与 16 个位对齐的字段“reserved_for_word_alignment”构成。换句话说，对于字幕流，未定义动态改变属性。

[0252] 参照图 28，说明块“EP_map()”的示范性内部结构。对于每种基本流，块“EP_map()”利用时间信息和位置信息指示位流中的可开始解码位置（也称为入口点或随机存取点 (RAMP)）。位置信息可以将记录媒体中的最小存取单元用于记录例如基本流。假设能够从块“EP_map()”所指的位置中解码每种基本流。

[0253] 在固定速率的数据流中，可开始解码位置可以通过计算确定，因此，不需要像块“EP_map()”那样的信息。但是，对于可变速率的数据流或像数据大小随存取单元而变的 MPEG 系统的视频压缩编码方案那样的数据流，像块“EP_map()”那样的信息对于随机存取是重要的。

[0254] 块“EP_map()”在其首端包括具有 8 个位的数据长度将数据排列与 16 个位对齐

的字段“reserved_for_word_alignment”。然后，安排了字段“number_of_stream_id_entries”。字段“number_of_stream_id_entries”具有 8 个位的数据长度，指示描述在块“EP_map()”中的基本流的数量。接着描述了数量等于第 1 个“for”循环中的字段“number_of_stream_id_entries”所指的次数的字段“stream_id”、字段“private_stream_id”和字段“number_of_EP_entries”。此外，对于第 1 个“for”循环的每一次描述，都安排字段“PTS_EP_start”和字段“RPN_EP_start”达第 2 个“for”循环中的字段“number_of_EP_entries”所指的次数。

[0255] 在第 1 个“for”循环中，第 1 步是在第 1 个“for”循环中安排每一个具有 8 个位的数据长度的字段“stream_id”和字段“private_stream_id”，并且像例子显示在图 25 中的那样，规定基本流。接着安排的字段“number_of_EP_entries”具有 32 个位的数据长度，指示为基本流描述的入口点的数量。此后，以与第 2 个“for”循环中的字段“number_of_EP_entries”所指相同的数量安排字段“PTS_EP_start”和字段“RPN_EP_start”。

[0256] 字段“PTS_EP_start”和字段“RPN_EP_start”每一个都具有 33 个位的数据长度，指示入口点本身。字段“PTS_EP_start”用 PTS 指示片段 AV 流文件中的入口点的时间。另一方面，字段“RPN_EP_start”以例如 2048 个字节为单位指示片段 AV 流文件中的入口点的位置。

[0257] 根据本实施例，构成盘状存取单元的一个扇区含有 2048 个字节。因此，在字段“RPN_EP_start”中通过扇区指示片段 AV 流文件中的入口点的位置。

[0258] 分组“private_stream_2”总是正好安排在视频流的可开始再现位置之前。这个分组“private_stream_2”其中存储了可用于解码视频流的信息。因此，视频流的入口点被当作存储分组“private_stream_2”的分组“pack()”的位置。

[0259] 如上所述的块“EP_map()”建立起片段 AV 流上的时间与片段 AV 流文件中的位置之间的对应关系。因此，在给出对片段 AC 流的存取点的时间信息（时间印记）的情况下，可以容易地检索到开始读取片段 AC 流文件中的数据的数据地址，从而可以平稳地随机存取盘。

[0260] 根据本实施例，事件按升序（或降序）安排和登记每种基本流的一组时间信息和位置信息（第 2 个“for”循环中的一组字段“PTS_EP_start”和字段“RPN_EP_start”）和块“EP_map()”中的一组字段“PTS_EP_start”和字段“RPN_EP_start”两者。换句话说，事先沿着预定方向重新安排时间信息和位置。其结果是，可以按原样对数据进行二叉树搜索。

[0261] 根据本实施例，尽管上面根据 MPEG2-Video 标准来描述，但视频基本流不局限于此。视频基本流可以基于例如 MPEG4-Visual 或 MPEG4-AVC 标准。此外，尽管上面作为 ATRAC 音频基本流来说明，但音频基本流不局限于此，也可应用例如 MPEG 1/2/4 audio。

[0262] 7. 盘再现装置

[0263] 接着，说明根据本发明实施例的盘再现装置。图 29 简要示出了根据本发明的盘再现装置的配置例子。总线 111 与 CPU（中央处理单元）112、存储器 113、驱动器接口 114、输入接口 115、视频解码器 116、音频解码器 117、视频输出接口 118、和音频输出接口 119 连接。盘再现装置 100 的一些部分可以通过总线 111 相互交换视频流、音频流和各种命令和数据。

[0264] 驱动器接口 114 进一步与盘驱动器 102 连接。盘驱动器 102 通过驱动器接口 114

与总线 111 交换数据和命令。

[0265] CPU 112 包括 ROM(只读存储器)和 RAM(随机存取存储器)(未示出)。按照事先存储在 ROM 中的程序和数据,CPU 112 通过总线 111 与盘再现装置 100 的每个部分交换数据和命令,从而控制整个盘再现装置 100 的操作。RAM 用作 CPU 112 的工作存储器。

[0266] 尽管在图 29 中未示出,但盘再现装置 100 可以包括像闪速存储器那样可 重写数据和甚至在盘再现装置 100 的电源断开之后仍然可以保存存储数据的非易失性存储器。非易失性存储器与例如总线 111 连接,以便 CPU 112 可以将数据写入非易失性存储器中和从非易失性存储器中读取数据。

[0267] 将来自用户实际进行输入操作的输入单元的输入信号供应给输入接口 115。输入单元包括例如利用红外光信号远程操作盘再现装置 100 的遥控命令器或直接安装在盘再现装置 100 上的键。通过输入接口 115 将这些输入单元供应的输入信号转换成对 CPU 112 和输出到 CPU 112 的控制信号。

[0268] 盘 101 中以参照图 18 和随后的图所述的格式记录着播放清单、脚本程序、片段信息文件、片段 AV 流文件等。一旦将盘 101 装入盘驱动器 102 中,就自动或通过用户的输入操作再现盘 101。将从盘 101 中读取的脚本程序、播放清单文件和片段信息文件供应给 CPU 112,并且存储在例如 CPU 112 的 RAM 中。CPU 112 根据存储在 RAM 中的脚本程序和数据从盘 101 中读取片段 AV 流文件。

[0269] 将从盘 101 中读取的片段 AV 流文件暂时存储在存储器 113 中。视频解码器 116 根据 CPU 112 的指令解码存储在存储器 113 中的片段 AV 流文件的视频流和字幕流。如此解码的视频数据和字幕数据在被合成或加入单个视频数据中的同时被例如 CPU 112 放大,缩小,要不然进行图像处理。这些图像处理步骤也可以由视频解码器 116 或视频输出接口 118 执行。将这个视频数据缓存在存储器 113 中和供应给视频输出接口 118。视频输出接口 118 将供应的视频数据转换成例如引向视频输出端 120 的模拟视频信号。

[0270] 类似地,音频解码器 117 响应来自 CPU 112 的指令,解码存储在存储器 113 中的片段 AV 流文件的音频流。将如此解码的音频数据缓存在存储器 113 中和供应给音频输出接口 119。音频输出接口 119 将供应的音频数据转换成例如引向音频输出端 121 的模拟音频信号。

[0271] 如图 29 所示的部分尽管在上面被说明成独立硬件,但未必是这样。例如,可替代地,视频解码器 116 和 / 或音频解码器 117 可以以软件形式构成,以便在 CPU 112 上操作。

[0272] 此外,可以将上述包括 CPU 112 和存储器和按照程序操作的盘再现装置 100 当作一种计算机系统。

[0273] 图 30A 和 30B 是说明如图 29 所示的盘再现装置 100 的更详细操作的功能方块图。盘再现装置 100 大体包括操作系统 201 和视频内容再现单元 210。 视频内容再现单元 210 基本上是在操作系统 201 上操作的软件程序。不过,视频内容再现单元 210 也可以以软件和硬件组合的方式操作。如下的描述假设视频内容再现单元 210 是软件。在图 30A 和 30B 中,未示出盘驱动器 102。

[0274] 首先利用盘再现装置 100 的电源在 CPU 112 中启动操作系统 201,并且通过执行像初始化每个部分那样的所需处理,从 ROM 中检索应用程序(在这种情况下,视频内容再现单元 210)。在视频内容再现单元 210 操作期间,操作系统 201 将像从盘 101 中读取文件或解

释文件系统的处理那样的基本服务提供给视频内容再现单元 210。响应从视频内容再现单元 210 接收的文件读取请求，例如操作系统 201 通过驱动器接口 114 控制盘驱动器 102，并且读取记录在盘 101 中的数据。在操作系统 201 的控制下将如此读取的数据传送给视频内容再现单元 210。

[0275] 此外，操作系统 201 具有多任务处理功能，并且可以通过时分控制显然相互并行的多个软件模块。具体地说，像例子显示在图 30A，30B 中的那样，可以通过操作系统 201 的多任务处理功能并行地操作构成视频内容再现单元 210 的所有模块。

[0276] 下面将更具体地说明视频内容再现单元 210 的操作。视频内容再现单元 210 其中含有几个其它模块和执行如下功能：

[0277] (1) 确定装入的盘 101（下文称为 UMD 视频盘）是否遵从 UMD 视频标准。

[0278] (2) 一旦确定装入盘 101 是 UMD 视频盘，从盘 101 中读取资源文件，并且将它传送给脚本控制模块 211。

[0279] (3) 一旦确定装入盘 101 是 UMD 视频盘，进一步读取构成数据库的文件（播放清单文件、片段信息文件等），并且将它们传送给播放器控制模块 212。

[0280] 下面说明视频内容再现单元 210 每个模块的操作。

[0281] 脚本控制模块 211 将接收的资源文件存储在 CPU 112 未示出的 RAM 的预定区域中。CPU 112（脚本控制模块 211）读取、解释和执行存储在 RAM 中的脚本程序。作为一种选择，可以将资源文件存储在存储器 113 的预定区域中，并且由 CPU 112 未示出的 RAM 在需要时读取。

[0282] 正如已经结合播放器模型所述的那样，通过在脚本程序上控制图形处理模块 219，可以实现像按照用户输入制作和输出菜单屏幕、移动光标和改变 菜单屏幕那样的 GUI 操作。在该处理中，将存储在存储器 113 上的资源文件中的图像数据和声音数据用于制作菜单屏幕等。此外，脚本控制模块 211 可以通过执行脚本程序控制播放器控制模块 212。

[0283] 播放器控制模块 212 通过引用存储在像播放清单文件“PLAYLIST.DAT”和片段信息文件“XXXXX.CLP”那样从盘 101 中读取的文件中的数据库信息，进行再现记录在盘 101 中的视频内容的控制操作。

[0284] (1) 为了分析包括播放清单和片段信息的数据记录装置信息，

[0285] (2) 为了控制内容数据供应模块 213、解码控制模块 214 和缓冲器控制模块 215，

[0286] (3) 为了控制像再现（播放）、再现停止（停止）、再现的暂时停止（暂停）那样的播放器状态的转变，并且按照来自脚本控制模块 211 或输入接口 115 的指令控制像流切换那样的再现处理，和

[0287] (4) 为了从解码控制模块 214 获取有关当前再现视频流的时间信息，并且显示时间或生成标记事件，

[0288] 内容数据供应模块 213 响应播放器控制模块 212 的指令，从盘 101 中读取包括片段 AV 流文件的内容数据，并且将它传送到缓冲器控制模块 215。缓冲器控制模块 215 将如此传送给它的内容数据累积在作为缓冲实体 215A 的存储器 113 中。内容数据供应模块 213 以这样的方式控制缓冲器控制模块 215，那就是，响应来自视频解码器控制模块 216、音频解码器控制模块 217 和字幕解码器控制模块 218 的请求，以预定方式将累积在存储器 113 中的内容数据供应给模块 216、217 和 218。此外，内容数据供应模块 213 以这样的方式从盘

101 中读取控制数据，那就是，由缓冲器控制模块 215 以预定方式控制累积的内容数据量。

[0289] 解码控制模块 214 响应来自播放器控制模块 212 的指令，控制视频解码器控制模块 216、音频解码器控制模块 217 和字幕解码器控制模块 218 的操作。此外，其中具有时钟功能的解码控制模块 214 以这样的方式控制解码器控制模块 216、217 和 218 的操作，那就是，同步地输出视频数据和音频数据。

[0290] 构成缓冲器实体 215A 的缓冲器控制模块 215 排它地使用存储器 113 的一部分。此外，缓冲器控制模块 215 还存储数据首端指针和数据写入指针。缓冲器控制模块 215 进一步具有像视频读取功能、音频读取功能和字幕读取功能那样的内部模块。视频读取功能其中含有视频读取指针。此外，视频读取 功能其中含有累积构成存取单元信息的信息“au_information()”的寄存器。音频读取功能其中含有音频读取指针。字幕读取功能其中含有字幕读取指针和字幕读取功能标志。字幕读取功能标志按照写入的值控制字幕读取功能的有效性 / 无效性。例如，在将“1”写入字幕读取功能标志中的情况下，字幕读取功能被认为是有效的，而当写入“0”时，字幕读取功能被认为是无效的。

[0291] 提供缓冲器控制模块 215 的内部模块的视频读取功能、音频读取功能和字幕读取功能进一步具有从各种流多路复用的片段 AV 流中分离视频流、音频流和字幕流的去多路分用功能。根据本发明的实施例，片段 AV 流是以 MPEG2 系统的节目流的形式时分复用多种基本流形成的。因此，视频读取功能、音频读取功能和字幕读取功能具有多路分用 MPEG2 系统的节目流的功能。

[0292] 视频读取功能读取和保存以预定方式安排在数据流中的字段“stream_id”（图 25）的值。类似地，音频读取功能和字幕读取功能分别读取和保存字段“stream_id”和字段“private_stream_id”（图 25）的值。字段“stream_id”和字段“private_stream_id”的值用于分析供应的位流。

[0293] 视频解码器控制模块 216 指令缓冲器控制模块 215 中的视频读取功能从存储器 113 中读取视频流的单个视频存取单元，并且供应给视频解码器 116。视频解码器控制模块 216 控制视频解码器 116 按存取单元解码供应给视频解码器 116 的视频流。将通过解码视频流准备的视频数据供应给图形处理模块 219。

[0294] 类似地，音频解码器控制模块 217 指令缓冲器控制模块 215 中的音频读取功能从存储器 113 中读取音频流的单个音频存取单元，并且供应给音频解码器 117。根据本实施例，假设构成音频流的存取单元（音频帧）具有已知固定长度。音频解码器控制模块 217 控制音频解码器 117 按存取单元解码供应给音频解码器 117 的音频流。将通过解码音频流准备的音频数据供应给音频输出模块 242。

[0295] 此外，字幕解码器控制模块 218 指令缓冲器控制模块 215 中的字幕读取功能从存储器 113 中读取字幕流的单个字幕存取单元，并且供应给字幕解码器控制模块 218。根据本实施例，构成字幕流的字幕存取单元具有存储在单元首端的单元长度信息。字幕解码器控制模块 216 具有字幕解码功能和可以解码供应的字幕流。将字幕解码器控制模块 216 的字幕解码功能从字幕流中 解码的字幕图像数据供应给图形处理模块 219。

[0296] 如上所述，将在视频解码器控制模块 216 的控制下在视频解码器 116 中解码的视频数据和字幕解码器控制模块 218 解码的字幕图像数据供应给图形处理模块 219。图形处理模块 219 按预定的那样将字幕图像数据加入供应的视频数据中，生成供输出用的视频信

号。此外，在图形处理模块 219 中，按照脚本控制模块 211 和播放器控制模块 212 的指令生成和与输出视频信号合成（叠加）菜单图像和消息图像。

[0297] 在图形处理模块 219 中，例如，响应来自脚本控制模块 211 的指令，放大或缩小供应的字幕图像数据，并且按预定的那样加入视频数据中。

[0298] 此外，图形处理模块 219 根据预先指定输出视频设备的宽高比和在从盘 101 中再现的内容中指定的输出宽高比改变输出信号的宽高比。例如，在输出视频设备的宽高比是 16 ： 9 和输出宽高比是 16 ： 9 的情况下，按原样输出视频数据。另一方面，在输出宽高比是 4 ： 3 的情况下，挤压输出视频数据，以便使图像高度与输出视频设备的屏幕高度齐平，并且在图像的左右侧插入和输出黑色图像。在输出视频设备具有 4 ： 3 的宽高比和输出宽高比是 4 ： 3 的情况下，按原样输出视频数据，而另一方面，在输出宽高比是 16 ： 9 的情况下，挤压输出视频数据，以便使图像宽度与输出视频设备的屏幕宽度一致而在图像上下插入和输出黑色图像。

[0299] 响应来自播放器控制模块 212 的请求，图形处理模块 219 进一步捕获正在处理的视频信号，并且使它返回到播放器控制模块 212。

[0300] 视频输出模块 241 排它地占据存储器 113 的一部分，并且将它用作 FIFO（先进先出）缓冲器。在这个缓冲器中临时累积经过图形处理模块 219 处理的视频数据，从而控制预定时刻的读取操作。从视频输出接口 118 输出从缓冲器中读取的视频数据。

[0301] 音频输出模块 242 排它地占据存储器 113 的一部分，并且将它用作 FIFO 缓冲器。在这个缓冲器中累积从音频解码器 117 输出的视频数据，并且在预定时刻读取它。从音频输出接口 119 输出从缓冲器中读取的音频数据。

[0302] 在内容的音频模式是双单声（例如，双语）的情况下，音频输出模块 242 按照预定指定音频输出模式输出音频数据。在将音频输出模式指定成“主音频”的情况下，例如，在存储器 113 中将左信道的音频数据复制到右信道，以便按左信道的音频数据输出双信道输出。另一方面，在音频输出模式是“次音 频”的情况下，例如，在存储器 113 中将右信道的音频数据复制到左信道，以便按右信道的音频数据输出双信道输出。在音频输出模式是“主 / 次声道”或内容是立体声的情况下，按原样输出音频数据。

[0303] 这样，用户可以通过视频内容再现单元 210 生成的菜单屏幕交互式地设置音频输出模式。

[0304] 响应来自播放器控制模块 212 的指令，非易失性存储器控制模块 250 将数据写入视频内容再现单元 210 操作结束时未擦除的区域中，并且从特定区域中读取数据。非易失性存储器控制模块 250 具有将标题 ID(Title_ID) 作为关键词地将数组数据“Saved_Player_Status”和数据“Saved_User_Data”存储在相同区域中的功能。作为数据“Saved_Player_Status”，存储播放器控制模块 212 保存的数据“Backup_Player_Status”。这个数据“Back_up_Player_Status”对应于例如上述播放器状态 323B 正好在播放器控制模块 212 结束之前的数据，并且数据“Saved_Player_Status”对应于重新开始信息 324。此外，将保存在播放器控制模块 212 中的数据“User_Data”存储成数据“Saved_User_Data”。数据“User_Data”是用户为播放器控制模块 212 设置的预定数据。

[0305] 非易失性存储器控制模块 250 在保存在盘再现装置 100 中的非易失性存储器的预定区域中与盘 101 的标题 ID 相关联地存储一组数据“Saved_Player_Status”和数据

“Saved_User_Data”非易失性存储器控制模块 250 存储数据的存储媒体不局限于非易失性存储器，可替代地，它可以是硬盘 8。

[0306] 8. 电影播放器的状态转变模型

[0307] 8-1 电影播放器状态的定义

[0308] 接着，更详细地说明根据本发明实施例的电影播放器 300 的状态改变模型。根据本发明，只将电影播放器 300 的状态定义成电影播放器 300 的内部状态。具体地说，根据本发明，根据电影播放器 300 本身的操作和功能定义电影播放器 300 的状态。

[0309] 更具体地说，将电影播放器 300 的操作定义成从播放清单再现的观点来看包括播放状态和停止状态的两种状态。此外，将功能定义成基于电影播放器 300 是否接受来自本机实现平台 301 的控制命令的两种状态。

[0310] 图 31 示意性地示出了根据本发明的电影播放器 300 的状态定义。下面说明从操作的观点来看电影播放器 300 的状态。在图 3 中，从播放清单再现的观点来看，电影播放器 300 处在播放状态或停止状态。在播放状态下，电影 播放器 300 选择播放清单和再现所选播放清单。另一方面，在停止状态下，电影播放器 300 不再现播放清单。在停止状态下，不选择播放清单。换句话说，在播放状态下，电影播放器 300 的回放模块 321 解码片段 AV 流，而在停止状态下，不解码片段 AV 流。

[0311] 将播放状态细分成几种状态，包括像以单一速度沿着向前方向的正常再现、以非正常速度沿着向前和向后方向的变速再现、和暂停那样的各种再现状态。通过在正常再现和暂停之间交替改变可以实现逐帧向前再现和逐帧向后再现。因此，在再现播放清单期间，电影播放器 300 基本上处在播放状态。

[0312] 下面说明从其功能的观点来看电影播放器 300 的状态。从其功能的观点来看，电影播放器 300 存在两种操作模式，即，从本机实现平台 301 接受控制命令 311 的模式（称为正常模式）和忽略控制命令 311 的模式（称为菜单模式）。将电影播放器 300 的这两种操作模式的每一种定义成电影播放器 300 的状态。

[0313] 在正常模式下，无需脚本层 302 的脚本程序，通过用户输入 310 就可以控制电影播放器 300 的操作。

[0314] 另一方面，在菜单模式下，电影播放器 300 不接受控制命令 311。电影播放器 300 只接受来自脚本层 302 的方法 313。其结果是，可以通过脚本层 302 的脚本程序全面控制电影播放器 300 的操作。例如，将用户输入 310 作为事件 314 从本机实现平台 301 传送到脚本层 302。脚本层 302 的脚本程序利用与这个事件 314 相对应的方法 313 控制电影播放器 300 的操作。

[0315] 具体地说，在内容制作者这一方，可以利用菜单模式控制电影播放器 300 的操作。此外，使用菜单模式使利用少数几种键实现各种各样的控制操作成为可能。

[0316] 如上所述，电影播放器 300 一方面，就操作而言，具有包括播放状态和停止操作的两种状态，而另一方面，就功能而言，具有正常模式和菜单模式的两种模式。因此，对于电影播放器 300，可以定义包括两种操作状态和两种功能模式的四种状态。具体地说，电影播放器 300 在从其产生到消亡的间隔内与四种状态的某一种相关联。电影播放器 300 的产生和消亡将在后面说明。

[0317] 顺便提一下，在发送指令电影播放器 300 转变到与当前状态不同的状态的方法

313 的情况下，作为模型，电影播放器 300 马上按照发送的方法 313 转变状态。但是，在实际设备中，将给定方法 313 发送到电影播放器 300 的时间到电影播放器 300 按照方法 313 完成状态转变的时间的时间长度取决于特定设备的安装。

[0318] 此外，即使在指定状态转变到与电影播放器 300 的给定状态相同的状态的方法 313 被发送到电影播放器 300 的情况下，电影播放器 300 的状态也保持不变。例如，即使在使电影播放器 300 的状态转变成正常模式和停止状态的方法 313 被发送到已经处在正常模式和停止状态下的电影播放器的情况下，电影播放器 300 的状态也不改变。

[0319] 此外，还包括作为一种播放状态的暂时停止（暂停）状态。对于从停止状态到暂停的转变，使用具有将暂时停止指定成变元的值“pauseMode”的方法“play()”。

[0320] 接着，说明包括电影播放器 300 的两种状态和两种操作模式的组合的四种状态的每一种和四种状态之间的状态改变。在如下的描述中，假设包括在根据功能分类的电影播放器 300 的状态中的正常模式被称为“normal”，而菜单模式被称为“menu”。另一方面，在按操作分类的电影播放器 330 的状态当中，播放状态被称为“play”，而停止状态被称为“stop”。此外，为了方便起见，将电影播放器的模式和状态的组合表达成“mode，state”。此外，在如下的描述中，将电影播放器 300 中的状态改变和模式切换统称为状态改变或状态转变。

[0321] 从图 31 中可以看出，对于电影播放器 300，包括改变成它自己的状态在内，总共存在 16（＝4×4）种状态改变。这些状态改变由从脚本层 302 发送到电影播放器 300 的方法 313 引起。具体地说，电影播放器 300 中的状态改变由电影播放器 300 外部的因素引起，没有通过来自脚本层 302 的方法的指令，在电影播放器 300 中不会发生自动状态改变。此外，即使使用来自本机实现平台 301 的控制命令，在电影播放器 300 中也决不会发生状态改变。

[0322] 根据本实施例，方法 313 的变元的组合受到限制，因此，在电影播放器 300 中不能产生所有 16 种可能状态改变。

[0323] 下面逐个说明电影播放器 300 可以采用的四种状态或状态（“Menu，Stop”、“Normal，Stop”、“Menu，Play”、和“Normal，Play”）。

[0324] (1) 状态“Menu，Stop”

[0325] 这是电影播放器 300 不再现播放清单，并且不接受来自本机实现平台 301 的控制命令 311 的停止状态。这种状态用在例如没有在背景中再现动态图像的菜单屏幕上。

[0326] 为了绝对保证正好在生成电影播放器 300 之后来自脚本程序的控制操作，在特定时刻不接受来自本机实现平台 301 的控制命令 311 是有效的。由于这个原因，在生成电影播放器 300 之后马上设置状态“Menu，Stop”。

[0327] (2) 状态“Normal，Stop”

[0328] 这是电影播放器 300 不再现播放清单，但接受来自本机实现平台 301 的控制命令 311 的停止状态。这种状态用在例如不再现动态图像的状态中。在生成电影播放器 300 之后最好不马上使用这个接受控制命令 311 的状态。

[0329] (3) 状态“Menu，Play”

[0330] 这是电影播放器 300 再现播放清单，但不接受来自本机实现平台 301 的控制命令 311 的播放状态。这种状态用在例如在背景中再现动态图像的菜单屏幕上。

[0331] (4) 状态“Normal，Play”

[0332] 这是电影播放器 300 再现播放清单，并且接受来自本机实现平台 301 的控制命令 311 的播放状态。这种状态用在例如再现电影原始故事期间。

[0333] 下面简要说明生成电影播放器 300 的模型。如上所述，在生成电影播放器 300 时，例如，接通盘再现装置 100 的电源，并且在 CPU 112 中启动操作系统 201。然后，在从 ROM 中存取视频内容再现单元 210 的同时，为每个部分执行像初始化那样的所需处理。这个视频内容再现单元 210 由 CPU 112 执行。一旦断开盘再现装置 100 的电源，就毁灭电影播放器 300。

[0334] 在这种情况下，认为电影播放器 300 生成隐式对象，并且不要求在脚本程序中显式地生成电影播放器 300。

[0335] 如上所述，将正好在生成电影播放器 300 之后的状态当作菜单模式的停止状态（状态“Menu，Stop”）。正好在生成电影播放器 300 之后，保存在例如电影播放器 300 中的如下特性呈现不定值：

[0336] 特性“audioFlag”

[0337] 特性“audioNumber”

[0338] 特性“chapterNumber”

[0339] 特性“playListNumber”

[0340] 特性“playSpeed”

[0341] 特性“subtitleFlag”

[0342] 特性“subtitleNumber”

[0343] 特性“videoNumber”

[0344] 在具有从前一次中止再现的位置开始再现的“连续再现功能”的 UMD 视频播放器中，取代默认值，将每个特性的值设置成保存在非易失性存储器中的值可以初始化电影播放器 300。例如，可以使用重新开始信息 324。

[0345] 8-2 产生电影播放器中的状态改变的方法

[0346] 接着，说明产生电影播放器 300 中的状态改变或转变的方法。图 32 示出了当前状态“Mode，Status”和通过方法 313 使电影播放器 300 的四种状态的每一种发生状态改变之后的状态“Mode，Status”的组合。从图 32 可以看出，准备了方法“stop()”、方法“play()”和方法“resume()”，作为产生电影播放器 300 的状态改变的方法 313。顺便提一下，方法“resume()”的操作随重新开始信息 324 是否存在而变。

[0347] 下面说明方法“stop()”。方法“stop()”与其状态无关地将电影播放器 300 改变成停止状态。方法“stop()”可以为变元指定模式，并且在转变成停止状态的同时，可以改变电影播放器 300 的模式。如后所述，一旦执行了满足某种条件的方法“stop()”，就将播放器状态 323B 备份和保存成重新开始信息 324。

[0348] 下面说明方法“play()”。方法“play()”将电影播放器 300 改变成播放状态。方法“play()”可以为变元指定模式，并且在改变成播放状态的同时，可以改变电影播放器 300 的模式。如后所述，一旦执行了满足某种条件的方法“play()”，就在保存重新开始信息 324 的同时，备份播放器状态 323B。

[0349] 下面说明方法“resume()”。方法“resume()”通过将重新开始信息 324 恢复成播放器状态 32B 来恢复再现。具体地说，方法“resume()”使电影播放器 300 从重新开始信息 324

所指的位置开始再现。在缺乏重新开始信息 324 的情况下，即使执行了方法“resume()”，也不改变电影播放器 300 的状态。

[0350] 通过方法“resume()”恢复重新开始信息 324 的条件如下。具体地说，在在执行方法“resume()”时存在重新开始信息 324 和当前状态不是“Normal，Play”的情况下，电影播放器 300 恢复重新开始信息 324。换句话说，只要在执行方法“resume()”时存在重新开始信息 324 和当前状态是“Menu，Stop”、“Normal，Stop”和“Menu，Play”之一，电影播放器 300 在转变成状态“Normal，Play”的同时恢复重新开始信息 324。

[0351] 方法“play()”含有多个变元。在受到考虑的情况下，为了说明起见，假设方法“play()”含有包括“pauseMode”、“menuMode”、和“playListNumber”的三个变元。但是，实际上，可以定义更多的变元。

[0352] 变元“pauseMode”指定在播放状态下再现的条件，可以假设具有“×1”、“pause”或“-1”。值“×1”指定以正常速度向前再现。值“pause”指定暂时停止。值“-1”指定保持当前再现速度。这样，变元“pauseMode”设置执行了方法“play()”之后电影播放器 300 的播放状态的细节。顺便提一下，如果在缺乏通过变元指定的位置上的画面或通过变元的指定的情况下指定了暂停，显示通过分开制定的选择规则指定的位置上的画面，并且进入暂停状态。

[0353] 变元“menuMode”可以指定电影播放器 300 的模式（正常模式或菜单模式），并且可以呈现值“Normal”、“Menu”和“-1”的任何一个。值“Normal”指定切换到正常模式。值“Menu”指定切换到菜单模式。值“-1”指定保持当前模式。

[0354] 变元“playListNumber”指定要再现的播放清单的号码。可以省略变元“playListNumber”，指示当前所选播放清单保持不变。

[0355] 下面参照图 33A、33B、33C、33D 和 33E，说明执行方法“play()”时电影播放器 300 的状态改变的例子。在图 33A、33B、33C、33D 和 33E 的每一个中，左侧指示电影播放器 300 的当前状态 340A，右侧指示在当前状态 340A 下将方法 313 从脚本程序发送到电影播放器 300 改变的状态 340B。此外，在状态 340A 和 340B 下面，示出了为特定状态指定的播放清单号 (PL1，PL2)。

[0356] 图 33A 示出了在状态“Normal，Stop”下将方法“play(x1，Normal，PL2)”发送到电影播放器 300 的情况的例子。这表示按照方法“play(x1，Normal，PL2)”，将电影播放器 300 的状态改变成在正常模式下以正常速度再现播放清单号“PL2”的播放清单的状态。电影播放器 300 从状态“Normal，Stop”改变成状态“Normal，Play”。

[0357] 图 33B 示出了在暂时停止播放清单号“PL1”的播放清单的再现的状态“Normal，Play”下，将方法“play(x1，Normal，PL2)”发送到电影播放器 300 的情况的例子。这表示按照方法“play(x1，Normal，PL2)”，将电影播放器 300 的状态改变成开始在正常模式下以正常速度再现播放清单号“PL2”的播放清单的状态。在这种情况下，电影播放器 300 的再现操作按照方法“play(x1,Normal,PL2)”从暂停改变成正常速度的向前再现。在这种情况下，电影播放器 300 的再现操作从暂停改变成正常速度的向前再现。但是，状态在发出方法“play(x1， Normal,PL2)”之前和之后保持在“Normal,Play”上，没有发生状态改变。

[0358] 图 33C 示出了在以正常速度向前再现播放清单号“PL1”的播放清单期间，在状态“Normal,Play”下将方法“play(-1,-1,PL2)”发送到电影播放器 300 的情况的例子。这表

示按照方法“play(-1，-1，PL2)”，使电影播放器 300 处在在正常模式下以正常速度再现播放清单号“PL2”的播放清单的状态。此外，在这种情况下，尽管电影播放器 300 再现的播放清单发生了变化，但状态“Normal，Play”保持不变，没有发生状态改变。

[0359] 图 33D 示出了将方法“play(pause，-1，PL2)”发送到在状态“Normal，Play”下，处在以正常速度再现播放清单号“PL1”的播放清单的向前再现操作之中的的电影播放器 300 的情况的例子。这表示按照方法“play(pause，-1，PL2)”，使电影播放器 300 处在选择播放清单号“PL2”的播放清单和在播放清单号“PL2”的播放清单的首端暂时停止操作的正常模式下。此外，在这种情况下，电影播放器 300 的再现操作从正常速度的向前再现改变成暂停，而状态保持在“Normal，Play”上，没有发生状态改变。

[0360] 图 33E 示出了在暂停播放清单号“PL1”的播放清单的再现期间，在状态“Normal，Play”下将方法“play(-1，Menu)”发送到电影播放器 300 的情况的例子。在方法“play()”中，省略了变元“playListNumber”。这表示按照方法“play(-1，Menu)”，使电影播放器 300 处在在播放清单号“PL1”的播放清单的首端暂时停止的菜单模式下的同时，选择播放清单号“PL1”的播放清单。电影播放器 300 从状态“Normal，Play”改变成状态“Menu，Stop”。

[0361] 如上所述，电影播放器 300 在视条件而定产生状态转变的同时，通过接收来自脚本程序的方法“play()”，进行各种各样的操作。内容制作者通过在脚本程序中描述不同变元的方法“play()”，可以在电影播放器 300 中实现各种各样的操作。

[0362] 电影播放器 300 通过执行来自脚本程序的方法“play()”，开始只再现所选播放清单号的播放清单。开始再现播放清单表示从停止状态开始再现播放清单，或者通过中止播放清单再现和开始再现所选播放清单选择新播放清单。

[0363] 在脚本程序将带有变元的方法“play()”发送到电影播放器 300 的情况下，将变元的值设置在播放器状态 323B 中。在省略了变元的情况下，将特定变元的值设置成默认值，或通过按照为每个参数确定的规则的自动选择设置特定变元的值。

[0364] 按内容制作者不想要的顺序再现播放清单将会引起问题。因此，禁止通过利用基于用户操作的控制命令 311 指定播放清单号，开始再现播放清单。这是根据本发明实施例的电影播放器 300 的工作模型的特征之一。

[0365] 此外，在将无效播放清单或不存在时间指定成方法“play()”的变元的值的情况下，将无法执行特定方法“play()”。这表示脚本程序有错，并且脚本程序包含不合标准之处。对付这种情况的差错管理取决于电影播放器 300 的安装。

[0366] 现在，说明播放项之间的再现。开始再现播放清单的电影播放器 300 继续再现特定播放清单直到结束。从给定播放清单的首端再现到末端既不需要用户操作也不需要来自脚本程序的控制操作。在电影播放器 300 中，如图 34 简要显示的那样，按播放清单文件“PLAYLIST.DAT”(图 19) 指定的顺序依次再现构成播放清单的播放项。不受事件管理器控制地连续再现构成播放清单的播放项。

[0367] 从再现播放项结束到再现下一个播放项的间隔期间的操作取决于电影播放器 300 的安装，未按格式规定。例如，有关继续显示播放项的最后画面还是马上改变成黑屏的播放项之间的操作取决于安装。但是，通过像将播放项的 IN 点设置在随机存取点 (入口点，图 28) 上那样的适当编辑操作，可以尽可能地缩短播放项之间的时隙。

[0368] 8-3 播放清单再现期间的电影播放器操作

[0369] 接着，说明播放清单再现期间电影播放器 300 的操作。将像包括双倍速再现或三倍速再现的高速再现、和包括半速再现或反向再现的低速再现那样的用户变速再现指令输入本机实现平台 301 中作为用户输入 310，并且响应这个用户输入 310，将视安装而定的控制命令 311 从本机实现平台 301 发送到电影播放器 300。

[0370] 可变再现速度取决于电影播放器 300 的安装。在实现变速再现的例子中，将指令“较快”或“较慢”作为变元从能够指定速度的本机实现平台 301 发送到电影播放器 300，由电影播放器 300 转换成可实现速度。应该应用哪种方法取决于电影播放器 300 的安装。根据脚本程序，可以通过方法“get_PlayerStatus()”了解电影播放器 300 确定的速度。

[0371] 另一方面，对于从脚本程序到电影播放器 300 的方法“play()”，不能通过变元指定速度。在方法“play()”中，可以只指定包括“pause”（变换“pause”）和“normal speed reproduction”（变元“×1”）的两个变元。

[0372] 一旦在向前变速再现播放清单期间到达播放项的末端，电影播放器 300 就进行下一个播放项的再现。同时，电影播放器 300 继续沿着与前一个播放项相同的方向和以与前一个播放项相同的速度变速再现下一个播放项。

[0373] 图 35 示出了在在播放清单再现期间到达播放清单的开始点和结束点的情况下电影播放器 300 的操作例子。一旦在沿着向前方向的再现期间到达播放清单的结束点，电影播放器 300 在显示最后画面的同时进入暂停状态。为了擦除最后画面，需要在事件管理器“onPlayListEnd”等中为电影播放器 330 显式地指定停止（发送方法“stop()”）。

[0374] 顺便提一下，一旦在速度比正常模式高的高速再现期间到达播放清单的结束点，即使不对应于跳转点，也显示播放清单的最后画面。

[0375] 另一方面，一旦在沿着向后方向的播放清单再现期间到达播放项的首端，电影播放器 300 再现前一个播放项，即，沿着向前方向较早最后的播放项。此外，在这个前一个播放项的再现中，继续从末端到首端的反向再现。再现速度也保持不变。此外，一旦在反向再现期间到达播放清单的首端，取消变速再现，电影播放器 300 在播放清单的首端进入暂停状态。

[0376] 此外，还通过指定暂停的控制命令 311 将电影播放器 300 的状态改变成“pause”。通过控制命令 311 取消暂停时播放清单再现的方向和速度取决于 UMD 视频播放器的安装。

[0377] 接着，说明播放清单再现期间生成的事件。正如参照图 13 所述的那样，播放清单再现期间生成的事件包括与用户操作相对应的事件“angleChange”、事件“audioChange”和事件“substitleChange”、和与嵌在播放清单中的标记相对应的事件“chapter”和事件“mark”。此外，已经参照图 15 说明了生成事件时操作的详细例子。

[0378] 现在，说明播放清单的最后处理。正如已经说明的那样，电影播放器 300 再现通过方法“play()”指定的号码的播放清单。一旦开始，播放清单再现就不受脚本程序或控制命令 311 控制地一起持续到播放清单的末端。一旦到达播放清单再现的末端，电影播放器 300 就将事件“playListEnd”通知脚本程序。在到达播放清单的末端之前，可以使用任何方法。具体地说，与播放清单的再现是正常，快速还是由从其它播放清单的跳转引起的无关，电影播放器 300 在到达播放清单末端的时刻生成事件“playListEnd”。

[0379] 在再现到达播放清单的末端和生成事件“playListEnd”的情况下，电影播 放器 300 的状态进入暂停状态，并且电影播放器 300 内部保存的播放清单再现的时间与播放清

单的结束时间一致。顺便提一下，播放清单的结束时间被定义成结束播放清单的最后画面的再现的时间，并且与沿着再现时间轴的最后播放项的 OUT 点一致。

[0380] 事件“playListEnd”可以用于串行地再现播放清单或在多故事的分支点上显示菜单。

[0381] 一旦生成事件“playListEnd”，如果事件管理器“onPlayListEnd”被保存成要执行的程序，脚本程序就执行事件管理器“onPlayListEnd”。在在这个事件管理器“onPlayListEnd”中描述了开始再现其它播放清单的方法“play()”的情况下，电影播放器 300 开始再现其它播放清单。这样，播放清单再现就继续下去。

[0382] 下面参照图 36 给出具体说明。在播放清单号“PL1”的播放清单再现结束时，生成事件“playListEnd”。一旦生成这个事件“playListEnd”，就执行保存在脚本程序中的事件管理器“onPlayListEnd”。这个事件管理器“onPlayListEnd”指定再现播放清单号“PL2”的播放清单。响应这个事件管理器“onPlayListEnd”，电影播放器 300 再现指定播放清单号“PL2”的播放清单。

[0383] 再现路径暂时从播放清单号“PL1”的播放清单的末端转移到事件管理器“onPlayListEnd”，并且进一步转移到播放清单号“PL2”的播放清单的首端。

[0384] 例如，在要在多故事的分支点上显示菜单的情况下，可以将再现播放清单的指令描述成这样的意思，即，与将播放清单的末端作为分支点的事件“playListEnd”相对应的事件管理器“onPlayListEnd”显示菜单屏幕。

[0385] 图 37 更详细地示出了在播放清单的末端脚本层 302 的处理流程和电影播放器 300 的操作例子。在图 37 中，步骤 S30 到 S33 指示脚本层 302 这一方的处理，并且步骤 S40 到 S44 指示电影播放器 300 这一方的处理。

[0386] 在给定播放清单再现到末端之后，要求脚本程序发出再现下一个播放项的显式再现指令。再现播放清单的顺序由脚本程序决定，因此，在电影播放器 300 这一方不能自发确定下一个要再现的播放清单。

[0387] 一旦再现到播放清单的末端（步骤 S40），电影播放器 300 就将事件“play_ListEnd”通知脚本层 302（步骤 S41）。电影播放器 300 在继续显示在步骤 S40 中再现到末端的播放清单的最后画面的同时，转变到状态“pause”（步骤 S42）。

[0388] 在脚本层 302 中，一旦接收到事件“playListEnd”，就执行事件管理器“onPlayListEnd”（步骤 S30）。在这个事件管理器“onPlayListEnd”中通过脚本的描述确定电影播放器 300 的下一步操作。

[0389] 顺便提一下，在步骤 S40 之后，电影播放器 300 在播放清单末端的暂停期间忽略可能接收到取消暂停或开始向前再现的方法或控制命令 311。开始向前再现的方法是利用变元指定向前再现的方法“play()”和方法“play_Step()”。此外，开始向前再现的控制命令 311 包括命令“uo_play()”、命令“uo_playNextChapter()”、命令“uo_forwardScan()”、命令“uo_playStep()”、命令“uo_pauseOn()”或命令“uo_pauseOff()”。在操作处在播放清单末端的暂停状态的情况下，忽略这些命令。

[0390] 即使在操作处在播放清单末端的暂停状态的情况下，方法“stop()”和方法“resume()”也是有效的。此外，在播放清单末端的暂停期间，模式切换也是有效的。

[0391] 即使在生成事件“playListEnd”之后，处在正常模式下的电影播放器 300 除了开

始向前再现之外，也接受控制命令311。此外，在那种情况下，一旦执行了来自脚本程序的电影播放器300的方法313，就执行特定方法313指定的操作。

[0392] 在如图37所示的例子中，通过事件管理器“onPlayListEnd”指定方法“stop()”（步骤S31）。一旦执行了方法“stop()”，就中止控制命令311引起的电影播放器300的操作，并且转变到停止状态（步骤S43）。在停止状态下，例如，擦除至此再现的播放清单的最后画面，并且出现黑屏。

[0393] 此外，在脚本层302中，通过事件管理器“onPlayListEnd”指定再现下一个播放清单的方法313（步骤S32）。在方法“play()”中，例如，将值“x1”指定给变元“pauseMode”，将值“Menu”指定给变元“menuMode”，并将下一个要再现的播放清单号指定给变元“playListNumber”。然后，指令电影播放器300切换到菜单模式和以正常速度再现由变元“playListNumber”指定的号码的播放清单。在脚本层302中，结束事件管理器“onPlayList_End”（步骤S33）。在电影播放器300这一方，在以指定播放清单所指定的速度开始再现的同时，按照步骤S32指定的方法“play()”切换模式（步骤S44）。

[0394] 顺便提一下，为了改善用户可操作性，在结束播放清单的再现之后，内容制作者不应该未指定电影播放器300的下一步操作就离开。应该在事件管理器“onPlayListEnd”中描述下一步操作，并且应该使电影播放器300转变成 停止状态，指令它通过方法“play()”再现播放清单，或将它编辑成返回到菜单屏幕。

[0395] 8-4 恢复电影播放器的再现的功能

[0396] 接着，说明电影播放器300的状态转变和再现恢复功能。首先，参照图38，说明UMD视频播放器拥有三种类型的存储区。在UMD视频播放器的模型中，定义了包括播放器状态区501、重新开始信息区502和用户数据区503的大致三种类型的存储区。这三种类型的存储区501、502、503是在例如存储器113上形成的。可替代地，它们也可以在作为CPU 112的工作存储器的RAM上形成。

[0397] 播放器状态区501是保存指示电影播放器300的再现状态的信息的存储区。具体地说，将如图3所示的播放器状态323B保存在播放器状态区501中。播放器状态区501的内容可以通过来自脚本程序500的方法“getPlayer_Status()”读取。

[0398] 重新开始信息区502是临时保存（备份）保存在播放器状态区501中的部分信息的存储区。具体地说，以如图3所示的重新开始信息324的形式将播放器状态区501中的部分信息保存在重新开始信息区502中。在需要的时候，将保存在重新开始信息区502中的播放器状态区501中的部分信息恢复到播放器状态区501。备份操作和恢复操作由本机实现平台301执行。保存在重新开始信息区502中的信息由重新开始再现功能用于从前再现停止点开始再现。

[0399] 重新开始信息区502的内容可以通过来自脚本程序500的方法“get_ResumeInfo()”读取。在保存在重新开始信息区502中的重新开始信息324中，与数据流有关的参数的值可以通过来自脚本程序500的方法“change_ResumeInfo()”改变。

[0400] 如有需要，由本机实现平台301将保存在重新开始信息区502中的信息写入（保存）在非易失性存储器510中。类似地，如有需要，由本机实现平台301从非易失性存储器510中读取（装入）从重新开始信息区502写入非易失性存储器510中的信息，并且存储在重新开始信息区502中。

[0401] 顺便提一下，从播放器状态区 501 到重新开始信息区 502 的备件操作和从重新开始信息区 502 到播放器状态区 501 的恢复操作是通过方法执行与电影播放器 300 的特定状态转变一起产生的处理，并且由电影播放器 300 自动 执行。

[0402] 用户数据区 503 是保存取决于内容的信息的那一种，可以供内容制作者任意使用。这个区域可以按照像电影播放器 300 再现播放清单的路由历史或问题的正确或不正确答案那样的内容以任何方式使用。

[0403] 数据可以通过来自脚本程序 500 的方法“setUserData()”写入用户数据区 503 中。用户数据区 503 的内容可以通过来自脚本程序 500 的方法“get_UserData()”读取。如有需要，由本机实现平台 301 将保存在用户数据区 503 中的信息写入（保存）在非易失性存储器 510 中。类似地，如有需要，由本机实现平台 301 从非易失性存储器 510 中读取（装入）从用户数据区 503 写入非易失性存储器 510 中的信息，并且存储在用户数据区 503 中。

[0404] 为了实现再现恢复功能，说明根据本发明实施例构建的 UMD 视频播放器的模型。

[0405] 首先，简要说明重新开始操作。利用备份在重新开始信息区 502 中的信息恢复再现状态的操作被称为“重新开始”。“重新开始”通过方法“resume()”执行。

[0406] 更具体地说，将播放器状态区 501 中的播放器状态 232B 备份在重新开始信息区 502 中成为重新开始信息 324，并且按照方法“resume()”，利用备份在重新开始信息区 502 中的重新开始信息 324 恢复再现状态。播放器状态 232B 由电影播放器 300 的状态，即，电影播放器 300 当前再现的播放清单号、章节号或所选流号构成。

[0407] 电影播放器 300 利用发送给它的方法“resume()”的操作随在重新开始信息区 502 中是否存在重新开始信息 324 而变。在在重新开始信息区 502 中存在重新开始信息 324 的情况下，将特定重新开始信息 324 恢复成播放器状态区 501 中的播放器状态 323B。在该处理中，放弃重新开始信息区 502 中的重新开始信息 324。

[0408] 在在内容再现期间存取的菜单中再现流改变了的情况下，使用方法“changeResumeInfo()”。通过在通过方法“changeResumeInfo()”以预定方式改变了保存在重新开始信息区 502 中的重新开始信息 324 之后，利用方法“resume()”进行重新开始操作，可以通过改变再现流开始再现。

[0409] 通过执行方法“resume()”，使电影播放器 300 进行重新开始操作。作为一种选择，通过在通过方法“getResumeInfo()”获取重新开始信息 324 之后，执行 带有指定变元的方法“play()”，可以实现重新开始操作。

[0410] 现在参照图 39 和 40 说明重新开始信息区 502 中播放器状态 323B 的备份。图 39 示出了电影播放器 300 定义的四种状态之间，将保存在播放器状态区 501 中的播放器状态 323B 备份在重新开始信息区 502 中的状态转变。图 40 示出了将播放器状态 323B 备份在重新开始信息区 502 中的条件。

[0411] 在处在再现播放清单的正常模式（状态“Normal，play”）下播放的电影播放器 300 改变到停止状态的情况下，将保存在播放器状态区 501 中的播放器状态 323B 备份在重新开始信息区 502 中，并且以重新开始信息 324 的形式保存。顺便提一下，在停止状态下，部分播放器状态 323B 会呈现不定值。

[0412] 此外，随着电影播放器 300 从状态“Normal，play”转变到状态“Menu，play”，将保存在播放器状态区 501 中的播放器状态 323B 备份在重新开始信息区 502 中。

[0413] 但是，在菜单模式下再现播放清单的电影播放器 300 即使改变了状态，也不将保存在播放器状态区 501 中的播放器状态 323B 备份在重新开始信息区 502 中。

[0414] 具体地说，在如下情况下，以重新开始信息 324 的形式将播放器状态 323B 备份在重新开始信息区 502 中：

[0415] (1) 电影播放器 300 的当前状态是“Normal，Play”，并且通过执行方法“play()”，直接改变成状态“Menu，Play”；以及

[0416] (2) 电影播放器 300 的当前状态是“Normal，Play”，并且通过执行方法“stop()”，改变成状态“Normal，Stop”或“Menu，Stop”。在这种情况下，方法“stop()”的变元“resumeInfoClearFlag”的值是“false”。

[0417] 为了保存原始故事中的返回位置，将播放器状态 323B 保存（备份）在重新开始信息区 502 中。例如，根据将重新开始信息 324 用作备份在重新开始信息区 502 中的播放器状态 323B 的数据的假设，实现包括在原始故事再现期间跳转到动态图像菜单和再次返回到原始故事进行再现的一系列操作。

[0418] 在再现原始故事期间，即，在电影播放器 300 处在状态“Normal，Play”中的情况下，放弃重新开始信息区 502 中的重新开始信息 324。随着电影播放器 300 从状态“Normal，Play”转变成其它状态，将播放器状态 323B 备份在重新开始信息区 502 中成为重新开始信息 324。

[0419] 如上所述，为了实现重新开始再现，将播放器状态 323B 备份在重新开始 信息区 502 中，或按照电影播放器 300 的状态转变，适当地放弃重新开始信息区 502 中的重新开始信息 324。此外，一旦在脚本层 302 中指定了方法“resume()”，如果在重新开始信息区 502 中存在重新开始信息 324，将重新开始信息 324 恢复到播放器状态区 501 中成为播放器状态 323B。

[0420] 重新开始信息区 502 中的重新开始信息 324 可以利用来自脚本层 302 的方法“getResumeInfo()”读取。重新开始信息区 502 中的重新开始信息 324 中与数据流有关的参数可以通过方法“changeResumeInfo()”改变。此外，通过利用方法“stop()”的变元指定，可以放弃重新开始信息区 502 中的重新开始信息 324。

[0421] 下面参照图 41 到 44 说明保存在重新开始信息区 502 中的重新开始信息 324 到播放器状态区 501 的恢复及其放弃。在电影播放器 300 返回到再现原始故事，即，状态“Normal，Play”之后，放弃作为原始故事中的返回位置保存的重新开始信息 324。在该处理中，可以在恢复成播放器状态区 501 中的播放器状态 323B 之后或不作如此恢复地放弃重新开始信息。

[0422] 具体地说，在这个模型中，一旦电影播放器 300 返回到状态“Normal，Play”，就放弃重新开始信息区 502 中的重新开始信息 324。同时，在电影播放器 300 满足预定条件的情况下，在恢复到播放器状态区 501 中之后放弃重新开始信息区 502 中的重新开始信息 324。在重新开始信息 324 恢复到播放器状态区 501 中的情况下，从重新开始信息 324 指定的点开始再现。这对应于重新开始再现。

[0423] 图 41 示出了电影播放器 300 定义的四种状态之间的状态转变，其中，将重新开始信息 324 恢复到播放器状态区 501 中，然后放弃它。

[0424] 在描述在下面 (1) 到 (3) 中的三个条件得到满足的情况下，在恢复之后放弃重新

开始信息 324。

[0425] (1) 电影播放器 300 的当前状态是“Menu,Stop”、“Normal,Stop”或 Menu,Play”;

[0426] (2) 在重新开始信息区 502 内存在重新开始信息 324 ;以及

[0427] (3) 通过执行方法“resume()”,转变成状态“Normal,Play”。

[0428] 图 42 集中示出了这些条件。在电影播放器 300 处在状态“Normal,Play”中的情况下,不存在重新开始信息 324,因此,在图 42 中未定义。

[0429] 顺便提一下,一旦在重新开始信息 324 存在于重新开始信息区 502 之中 的时候执行方法“resume()”,电影播放器 300 的状态改变成状态“Normal,Play”。此外,在在重新开始信息区 502 中缺乏重新开始信息 324 的情况下,方法“resume()”不会改变电影播放器 300 的状态。此时,正好在执行方法“resume()”之前的状态“Mode,State”占优,播放器状态 323B 也未改变。

[0430] 另一方面,一旦描述在下面 (4) 到 (6) 中的三个条件都得到满足,不恢复地放弃重新开始信息 324。

[0431] (4) 电影播放器 300 的当前状态是“Menu,Stop”、“Normal,Stop”或 Menu,Play”;

[0432] (5) 在重新开始信息区 502 中存在重新开始信息 324 ;以及

[0433] (6) 通过执行方法“play()”将状态改变“Normal,Play”。

[0434] 图 43 集中示出了这些条件。在电影播放器 300 处在“Normal,Play”中的情况下,在重新开始信息区 502 中不存在重新开始信息 324,因此,在图 43 中未定义。

[0435] 在缺乏重新开始信息 324 的情况下,假设通过执行方法“play()”将电影播放器 300 的状态改变“Normal,Play”。其结果是,维持了缺乏重新开始信息 324 的状况。

[0436] 重新开始信息区 502 中的重新开始信息 324 也可以通过设置方法“stop()”的变元放弃。具体地说,根据本发明的实施例,将方法“stop()”的变元“resumeInfoClearFlag”定义成方法“stop()”确定是否放弃重新开始信息区 502 中的重新开始信息 324 的变元。如图 44 所示,在在执行方法“stop()”时为变元“resumeInfoClearFlag”指定了值“True”的情况下,放弃重新开始信息 324。

[0437] 在电影的原始故事再现到末端和电影播放器 300 已停止的情况下,例如,原始故事的最后位置被记录成重新开始信息 324。当用户随后再现(继续再现)时,处理跳转到电影原始故事的末端,进入暂停状态,这给操作带来不便。

[0438] 对这个问题的改进需要放弃在模型定义中自动记录的重新开始信息 324 的手段。电影原始故事的末端位置不为内容制作者以外的其它人所知。因此,应用可以通过来自脚本程序 500 的电影播放器 300 的方法“stop()”的变元“resumeInfoClearFlag”指定放弃重新开始信息 324 的手段。

[0439] 图 45 示出了 UMD 视频播放器利用方法“stop() 的变元“resumeInfo_ClearFlag”的操作例子。在图 45 中,步骤 S50 到 S54 指示脚本层 302 这一方 的处理,并且步骤 S60 到 S64 指示电影播放器 300 这一方的处理。

[0440] 当再现到播放清单的末端时(步骤 S60),电影播放器 300 将事件“play_ListEnd”通知脚本层 302(步骤 S61)。电影播放器 300 在继续显示在步骤 S60 中再现到末端的播放清单的最后画面的同时,将状态改变成暂停状态(步骤 S62)。

[0441] 响应事件“playListEnd”,脚本层 302 执行事件管理器“onPlayList_End”(步骤

S50)。下一个步骤S51确定通知了事件“playListEnd”的播放清单是否是最后一个创作情节。给定播放清单是否是最后一个情节可以根据例如脚本程序500确定。

[0442] 一旦确定播放清单不是最后一个，处理转到步骤S53，在步骤S53中，使方法“stop()”的变元“resumeInfoClearFlag”的值变成“false”，并且将不放弃重新开始信息324的方法“stop()”发送到电影播放器300。一旦接收到这个方法“stop()”，电影播放器300将播放器状态323B备份在重新开始信息区502中的同时，将状态改变成停止状态。

[0443] 另一方面，一旦在步骤S51中确定在脚本层302中已到情节的末端，处理转到步骤S52，并且将方法“stop()”的变元“resumeInfoClearFlag”的值变成“True”。然后，将放弃重新开始信息324的方法“stop()”通知电影播放器300。一旦接收到该方法“stop()”，电影播放器300放弃重新开始信息区502中的重新开始信息324的同时，将状态改变成停止状态。

[0444] 在脚本层302中，取决于脚本程序500的描述，在步骤S52之后执行方法“end()。

[0445] 8-5 每个数据的寿命

[0446] 接着，说明播放器状态323B、重新开始信息324和用户数据的寿命。

[0447] 图46示出了播放器状态323B的寿命例子。在生成电影播放器300时，也生成了播放器状态323B。一旦电影播放器300被毁灭，播放器状态323B也就毁灭了。在生成时初始化播放器状态323B。指示电影播放器300的状态的特性代表“停止状态”，而其它特性是不定的。播放器状态323B的值随电影播放器300的再现状态的改变而改变。此外，播放器状态323B的值随重新开始信息区502的内容的恢复而改变。此外，播放器状态323B可以通过来自脚本层302的方法“getPlayerStatus()”读取。

[0448] 存储播放器状态323B的具体形式取决于电影播放器300的安装。只要可 以通过来自脚本层302的方法“getPlayerStatus()”获取信息，可以以任何形式保存信息。

[0449] 图47A和47B示出了重新开始信息300的寿命例子。在生成电影播放器300时，保护和与重新开始信息324的生成一起初始化重新开始信息的存储区。一旦存储区被初始化，就放弃了重新开始信息324的内容。安装了非易失性存储器的UMD视频播放器在初始化电影播放器300时从非易失性存储器中装入重新开始信息324。同时，装入用户数据。

[0450] 当电影播放器300的状态从“Normal，Play”转变成其它状态时，将播放器状态323B备份在重新开始信息区502中。

[0451] 重新开始信息324中与数据流有关的参数“videoNumber”、“audio_Number”、“audioFlag”、“subtitleNumber”和“subtitleFlag”可以通过来自脚本层302的方法“changeResumeInfo()”改变。

[0452] 在电影播放器300中，当在正常模式下开始再现播放清单时，放弃重新开始信息324的内容。重新开始信息324在被放弃之前可以恢复或可以不恢复成播放器状态323B。此外，一旦执行了变元“resumInfoClearFlag”的值设置成“True”的方法“stop()”，放弃重新开始信息324的内容。

[0453] 一旦在缺乏重新开始信息324的情况下执行了方法“resume()”，将重新开始信息324恢复成播放器状态323B。

[0454] 重新开始信息324的值可以通过来自脚本层302的方法“getResumeInfo()”读取。在读取处在放弃状态下的重新开始信息324时，返回值“0”，作为返回值“playStatus”，因

此，可以确定重新开始信息 324 存在与否。

[0455] 在电影播放器 300 结束时（毁灭时），重新开始信息 324 也被毁灭。安装了非易失性存储器的 UMD 视频播放器在电影播放器 300 结束时（毁灭时），将重新开始信息 324 保存在非易失性存储器中。在该处理中，同时保存用户数据。

[0456] 图 48 示出了用户数据的寿命例子。在生成电影播放器 300 时，保护存储区，并且生成用户数据。在生成的同时初始化电影播放器 300。一旦被初始化，清除用户数据的内容（通过方法“getUserData()”恢复长度“0”的排列）。安装了非易失性存储器的 UMD 视频播放器在初始化电影播放器 300 时，从非易失性存储器中装入用户数据。在该处理中，同时装入重新开始信息。

[0457] 一旦执行了方法“setUserData()”，将用户数据写入用户数据区 503 中。通 过方法“setUserData()”将具有 64 个位的最大数据长度的排列保存在用户数据区 503 中。用户数据区 503 中的数据可以通过来自脚本层 302 的方法“getUserData()”读取。在未设置用户数据的情况下，返回长度 0 的排列。

[0458] 没有方法可用于从脚本层 302 清除用户数据区 503 的内容。可以通过盖写用户数据区 503 重写内容。

[0459] 在电影播放器 300 结束时（毁灭时），用户数据区 503 也被毁灭。安装了非易失性存储器的 UMD 视频播放器在电影播放器 300 结束时（毁灭时），将保存在用户数据区 503 中的数据保存在非易失性存储器中。在该处理中，同时保存重新开始信息 324。

[0460] 前面的描述涉及到将本发明应用于同时处理音频流和视频流的盘再现装置 100。但是，本发明不局限于这种配置。根据本发明，可以有选择地只再现音频流，或只再现视频流。

[0461] 此外，前面的描述论及了作为记录内容数据的记录媒体的盘记录媒体，但本发明不局限于这种配置。例如，可以将半导体存储器用作记录内容数据的记录媒体。

[0462] 此外，尽管前面描述了根据本发明的盘再现装置 100 含有专用硬件配置，但本发明不局限于这样的配置。具体地说，通过在计算机系统上运行的软件，也可以实现除了盘驱动器之外的盘再现装置 100 的配置。在这样的情况下，可以通过记录在像 CD-ROM（光盘只读存储器）或 DVD-ROM（数字多功能盘只读存储器）那样的记录媒体中供应实现盘再现装置 100 的软件。将记录了实现盘再现装置 100 的软件的记录媒体装入计算机系统的盘驱动器中，并且将记录在记录媒体中的特定软件安装在计算机系统中。通过将与 UMD 相对应的盘驱动单元与计算机系统连接，可以实现等效于盘再现装置 100 的配置。记录了 UMD 视频内容的记录媒体可以与记录在其中的特定软件一起提供。

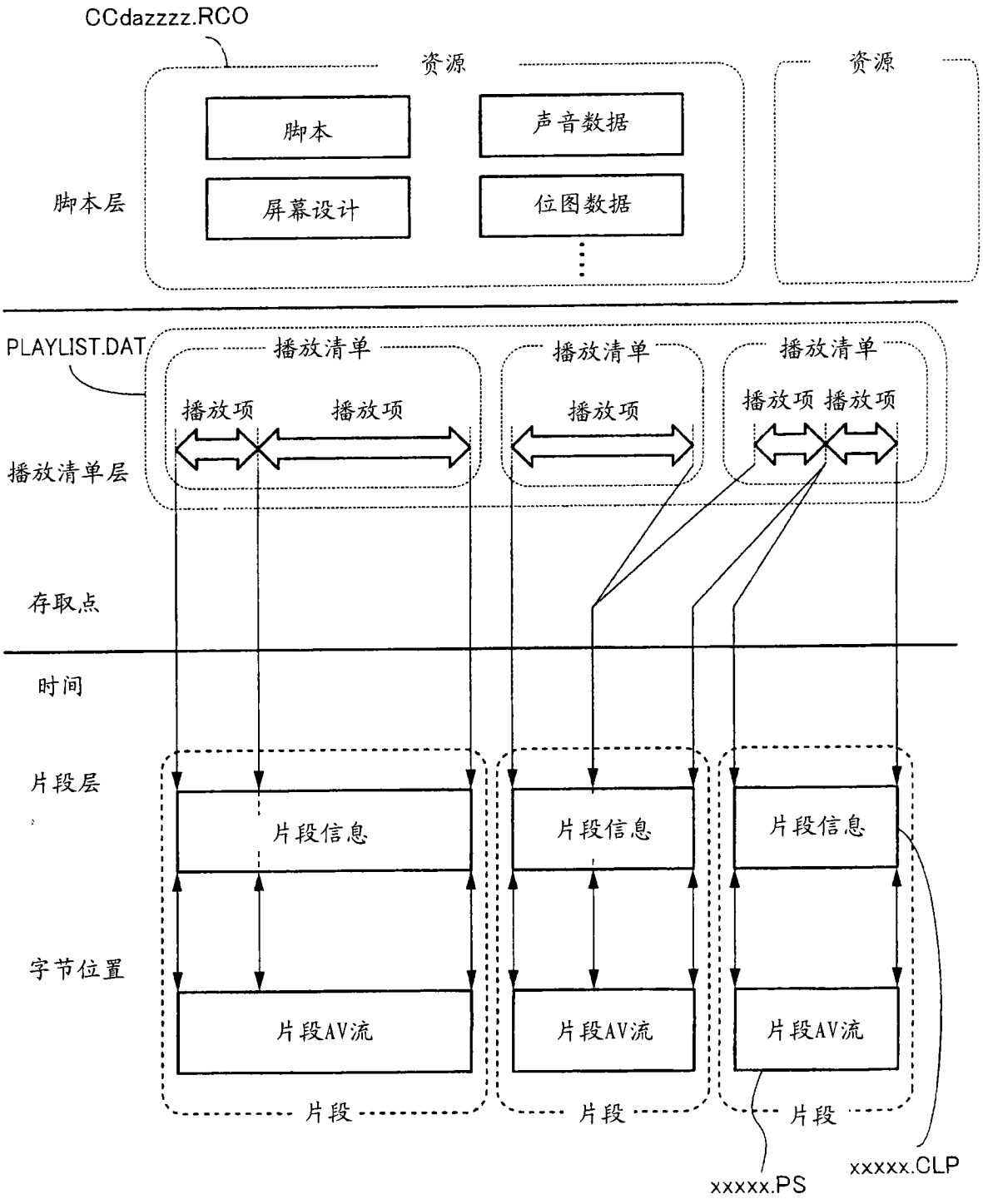

CCdazzzz.RCO
资源
资源
脚本
声音数据
脚本层
屏幕设计
位图数据
PLAYLIST.DAT
播放清单
播放清单
播放清单
播放项
播放项
播放项
播放项
播放项
播放清单层
存取点
时间
片段层
片段信息
片段信息
片段信息
字节位置
片段AV流
片段AV流
片段AV流
片段
片段
片段
xxxxx.PS
xxxxx.CLP

图 1

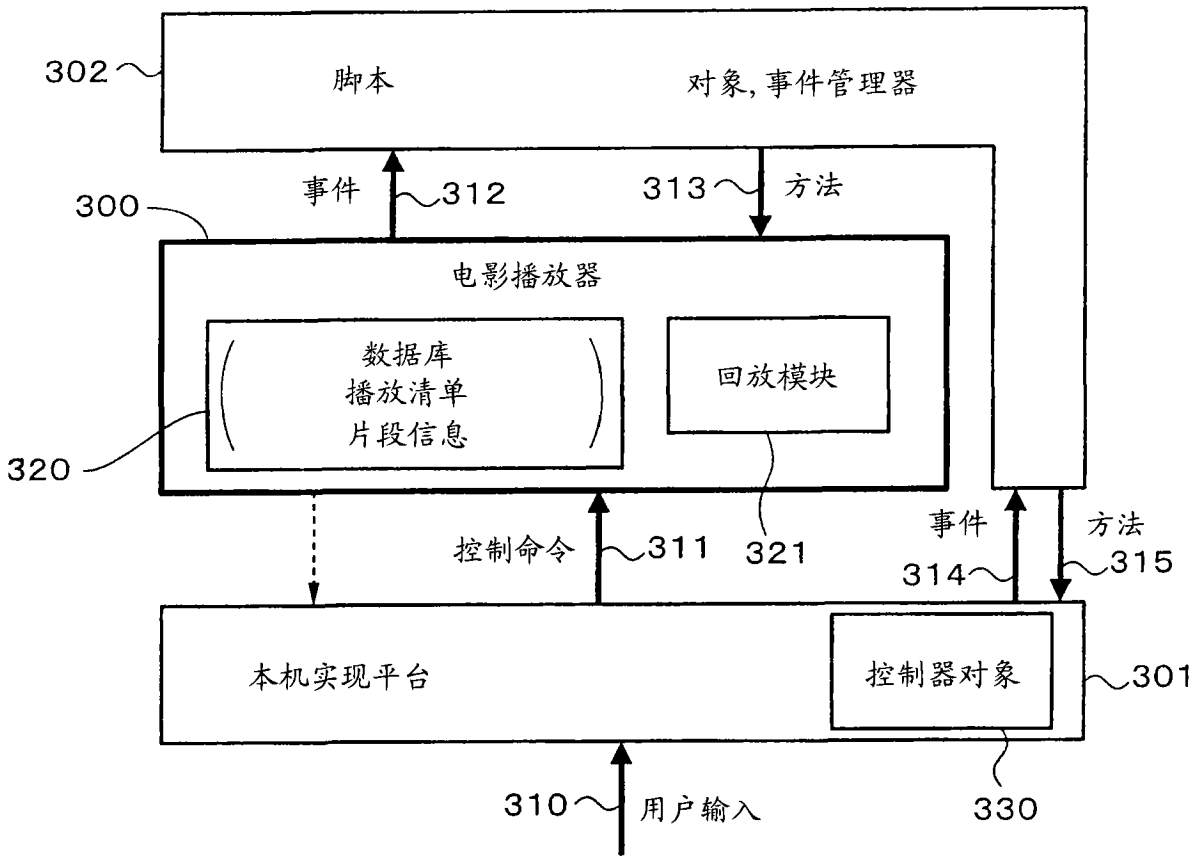

302
脚本
对象,事件管理器
事件
312
313
方法
300
电影播放器
数据库
播放清单
片段信息
回放模块
320
控制命令
311
321
事件
314
方法
315
本机实现平台
控制器对象
301
310
用户输入
330

图 2

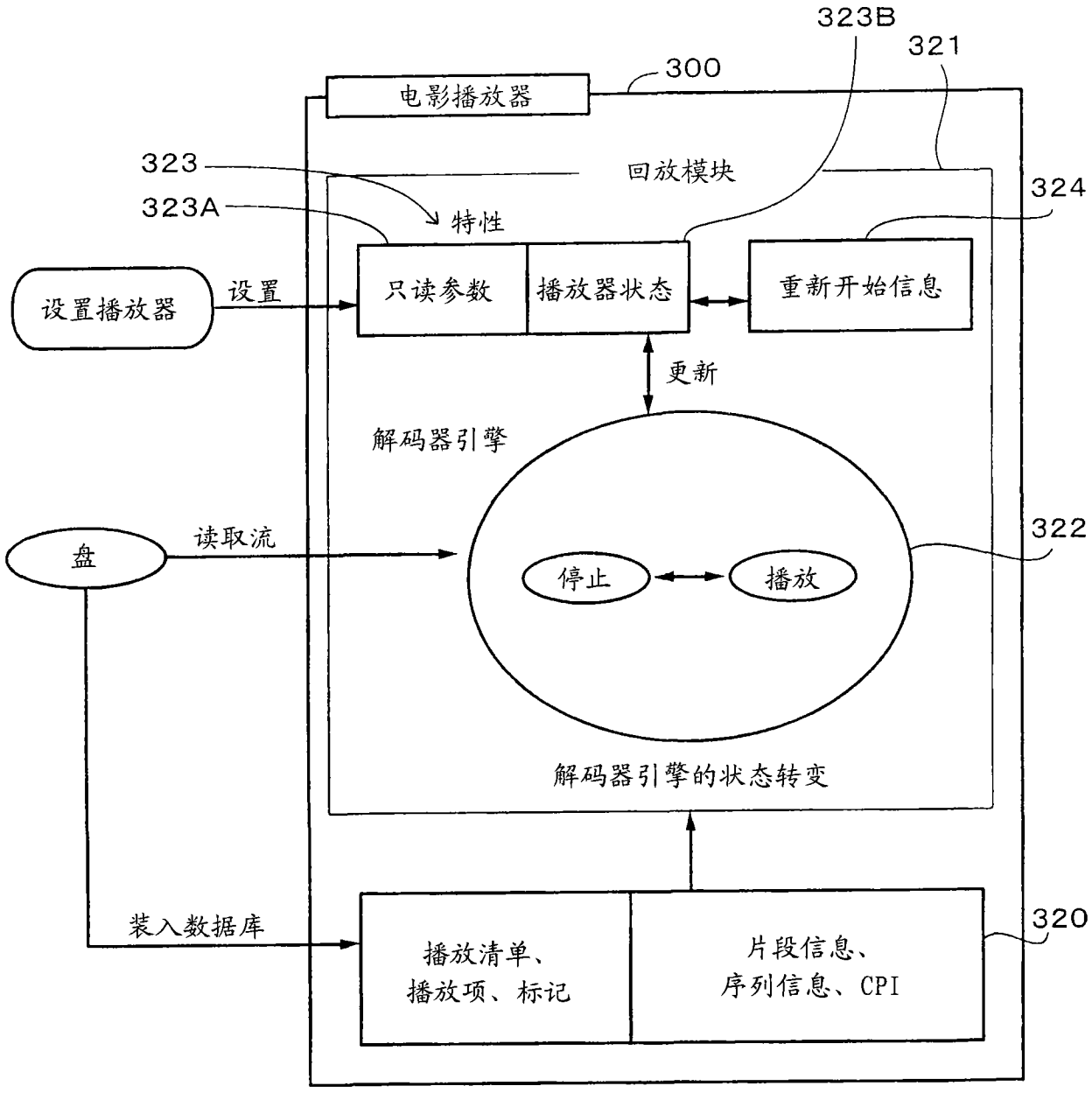

323B
321
300
电影播放器
323
回放模块
323A
特性
324
设置播放器
设置
只读参数
播放器状态
重新开始信息
更新
解码器引擎
读取流
盘
322
停止
播放
解码器引擎的状态转变
装入数据库
320
播放清单、
播放项、标记
片段信息、
序列信息、CPI

图 3

状态	说明
播放	指示选择和再现播放清单的状态. 播放状态被细分成分状态，和除了正常再现之外，一般包括各种分状态，即，像快进和快倒、和暂停那样的各种速度再现.
停止	指示不再现播放清单的状态. 不选择播放清单.

图 4

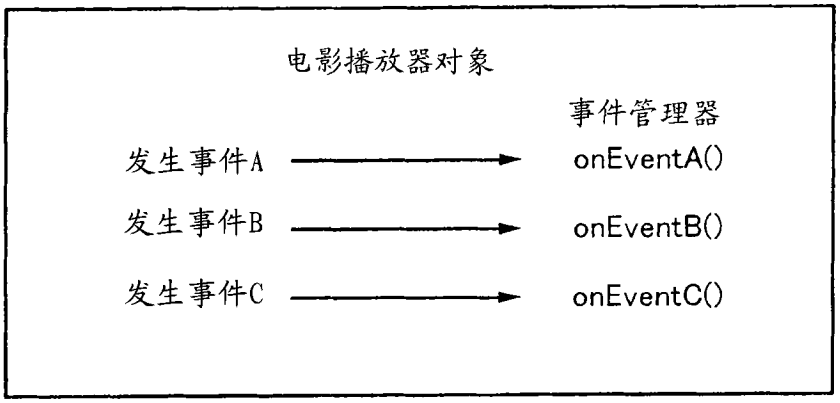


图 5

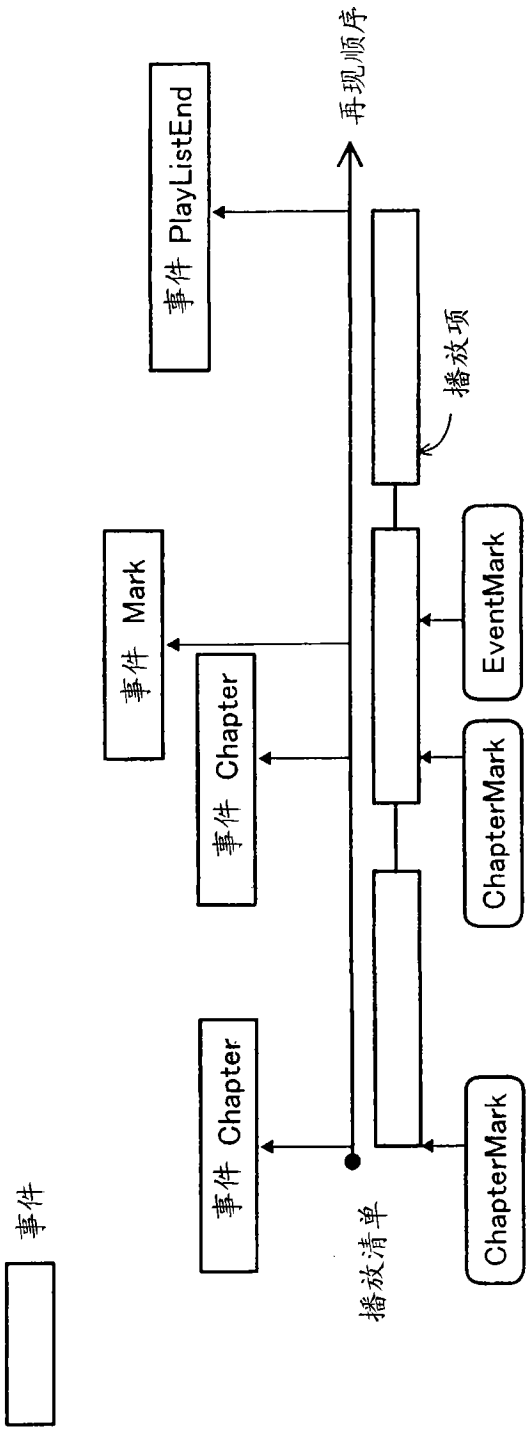

事件
播放清单
再现顺序
事件 Chapter
事件 Mark
事件 Chapter
事件 PlayListEnd
播放项
ChapterMark
EventMark
ChapterMark

图 6

只读参数

名称	说明
scriptVersion	UMD视频脚本的版本
audioChannelCapability	UMD视频播放器可以再现的音频信道的数量
languageCode	设置在UMD视频播放器中的菜单显示语言的语言代码
audioLanguageCode	设置在UMD视频播放器中的音频语言的语言代码
subtitleLanguageCode	设置在UMD视频播放器中的字幕语言语言代码

图 7A

播放器状态

名称	说明
playListNumber	当前再现播放清单的号码
chapterNumber	当前再现章节的号码
videoNumber	当前再现视频流的号码
audioNumber	当前再现音频流的号码
subtitleNumber	当前再现字幕流的号码
playListTime	假设播放清单首端是0的时间
audioFlag	音频再现的on/off和dual mono LR的指定
subtitleFlag	字幕显示on/off

图 7B

名称	说明
play()	再现
playChapter()	通过指定章节再现
resume()	利用重新开始信息开始再现
stop()	停止再现
pause()	暂时停止再现
playStep()	逐帧前进
changeStream()	改变视频、音频和字幕流
getPlayerStatus()	获取电影播放器中像再现、停止或暂停那样的状态
changeResumeInfo()	改变重新和清除重新开始信息
reset()	停止再现和清除重新开始信息
setPos()	设置视频显示位置
getPos()	获取视频显示位置
setSize()	设置视频显示尺寸
getSize()	获取视频显示尺寸

图 8

键名	说明
VK_PLAY	再现
VK_STOP	停止
VK_PAUSE	暂停
VK_FAST_FORWARD	快进
VK_FAST_REVERSE	快倒
VK_SLOW_FORWARD	慢(进)
VK_SLOW_REVERSE	慢(倒)
VK_STEP_FORWARD	逐帧(向前)
VK_STEP_REVERSE	逐帧(向后)
VK_NEXT	下一个
VK_PREVIOUS	前一个
VK_ANGLE	切换角度
VK_SUBTITLE	切换字幕
VK_AUDIO	切换音频
VK_VIDEO_ASPECT	切换宽高比

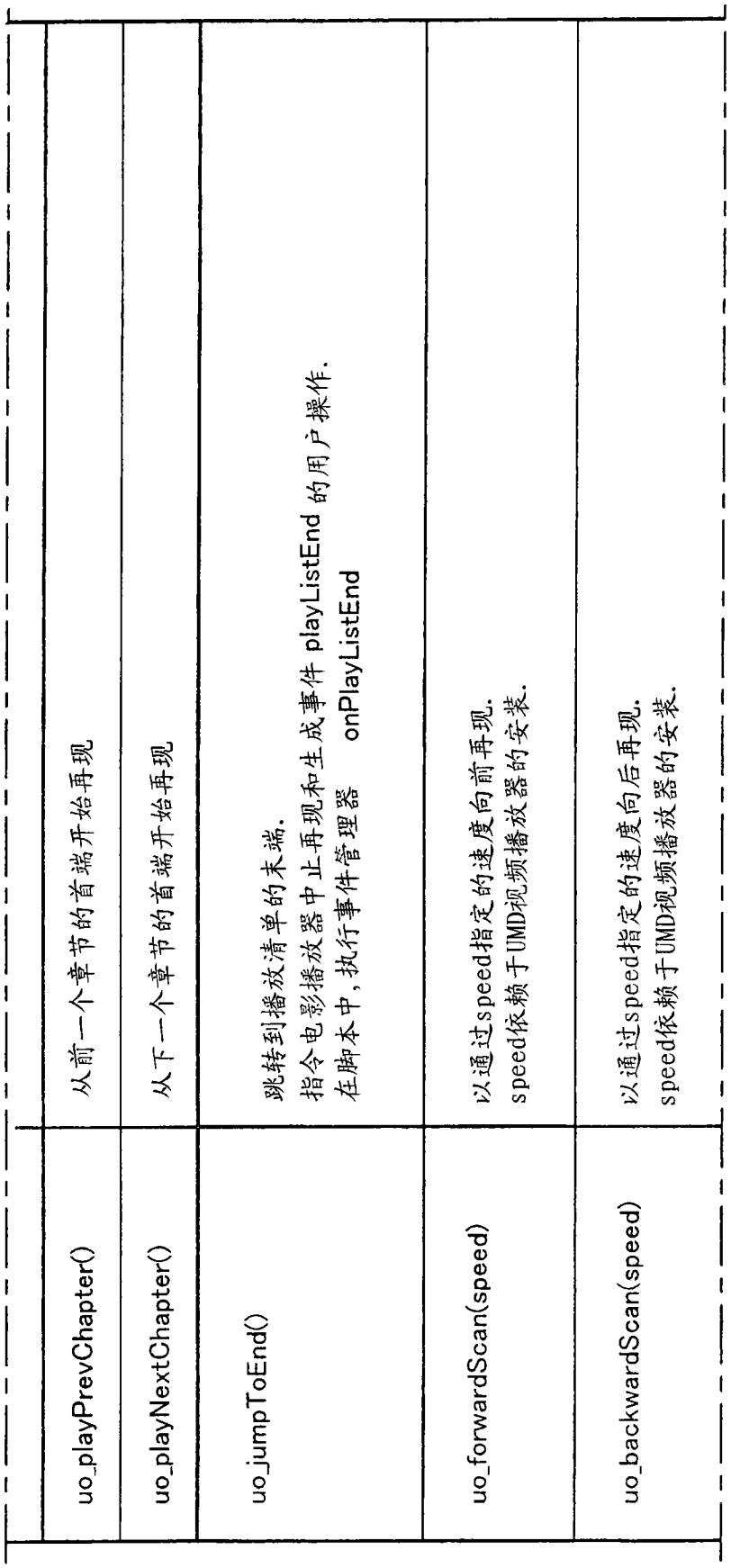
图 9

键名	说明
VK_UP	向上
VK_DOWN	向下
VK_RIGHT	向右
VK_LEFT	向左
VK_UP_RIGHT	右上方
VK_UP_LEFT	左上方
VK_DOWN_RIGHT	右下方
VK_DOWN_LEFT	左下方
VK_MENU	菜单
VK_ENTER	回车
VK_RETURN	返回
VK_COLORED_KEY_1	着色的功能键1
VK_COLORED_KEY_2	着色的功能键2
VK_COLORED_KEY_3	着色的功能键3
VK_COLORED_KEY_4	着色的功能键4
VK_COLORED_KEY_5	着色的功能键5
VK_COLORED_KEY_6	着色的功能键6

图 10

由用户操作引起的 控制指令	说明
uo_timeSearch(playListTime)	从指定时间开始再现当前再现播放清单. playListTime 指示播放清单的首端被设置成0的时间. 不能指定播放清单号. 因此, 指定包括在当前再现播放清单的范围内的时间.
uo_play()	以正常速度开始向前再现. 开始位置根据重新开始信息确定. 在缺乏 重新开始信息的情况下, 用户操作无效. 在执行未指定 playListNumber. 的方法 play() 时使用. 不能通过用户操作指定播放清单号.
uo_playChapter(chapterNumber)	从通过当前再现播放清单指定的章节开始再现. 在缺乏章节指定的情况下, 从当前再现章节的首端开始再现. 用于未指定 chapterNumber 的方法 playChapter()

图 11A



uo_playPrevChapter()	从前一个章节的首端开始再现
uo_playNextChapter()	从下一个章节的首端开始再现
uo_jumpToEnd()	跳转到播放清单的末端. 指令电影播放器中止再现和生成事件 playListEnd 的用户操作. 在脚本中，执行事件管理器 onPlayListEnd
uo_forwardScan(speed)	以通过speed指定的速度向前再现. speed依赖于UMD视频播放器的安装.
uo_backwardScan(speed)	以通过speed指定的速度向后再现. speed依赖于UMD视频播放器的安装.

图 11B

uo_playStep(forward)	向前逐帧前进地再现
uo_playStep(backward)	向后逐帧前进地再现
uo_pauseOn()	由用户暂进停止
uo_pauseOff()	取消暂时停止
uo_setAudioEnabled(boolean)	指定音频流的on/off，改变 audioFlag
uo_setSubtitleEnabled(boolean)	指定字幕流的on/off，改变 subtitleFlag
uo_angleChange()	改变显示角度. 一旦将用户操作发送到电影播放器，电影播放器就将事件 angleChange 通知脚本层.
uo_audioChange (audioStreamNumber)	改变要再现的音频.
uo_changeAudioChannel(value)	在dual mono再现时切换音频信道或单个信道的号码. 改变 audioFlag.
uo_subtitleChange (subtitleStreamNumber)	改变要再现的字幕.

图 11C

事件	说明
menu	跳转到菜单
exit	当本机实现平台结束UMD视频应用程序时，从本机实现平台发送事件.
resourceChanged	当切换资源文件时，将事件通知脚本层.
up,down,left,right focusIn, focusOut, push, cancel	在聚焦显示在屏幕上的按钮小部件的同时生成的事件.
autoPlay, continuePlay	指定开始执行脚本的事件.

图 12

事件名	相应事件管理器名	说明
mark	onMark()	检测到事件标记时执行
playListEnd	onPlayListEnd()	播放清单结束时执行
chapter	onChapter()	检测到章节标记时执行
angleChange	onAngleChange()	通过用户操作指定了角度改变时执行
audioChange	onAudioChange()	通过用户操作指定了音频改变时执行
subtitleChange	onSubtitleChange()	通过用户操作指定了字幕改变时执行

图 13

事件名	相应事件管理器名	内容
menu	onMenu()	跳转到菜单
exit	onExit()	本机实现平台结束UMD视频应用程序时从本机实现平台发出的事件
resourceChanged	onResourceChanged()	描述资源切换之后的处理
autoPlay, continuePlay	onAutoPlay(), onContinuePlay()	开始执行脚本

图 14

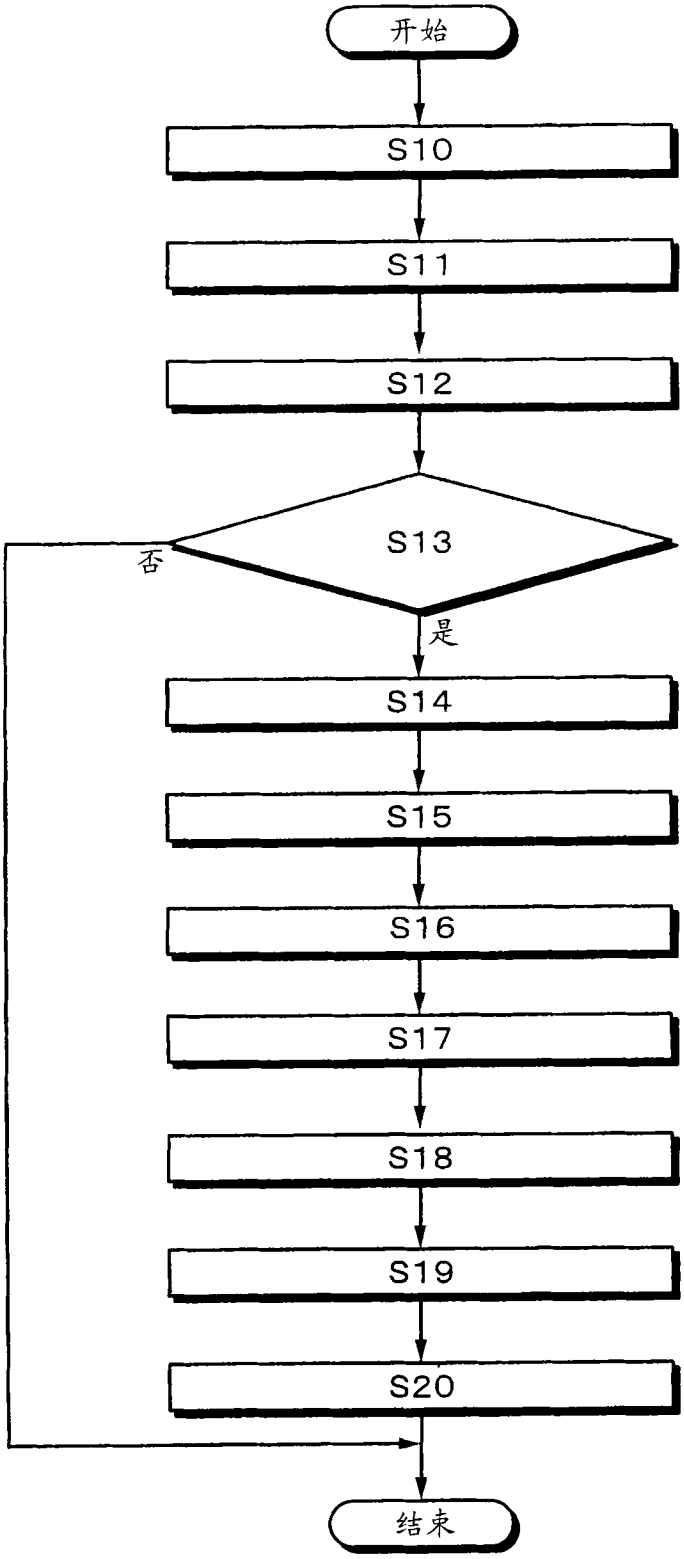


图 15

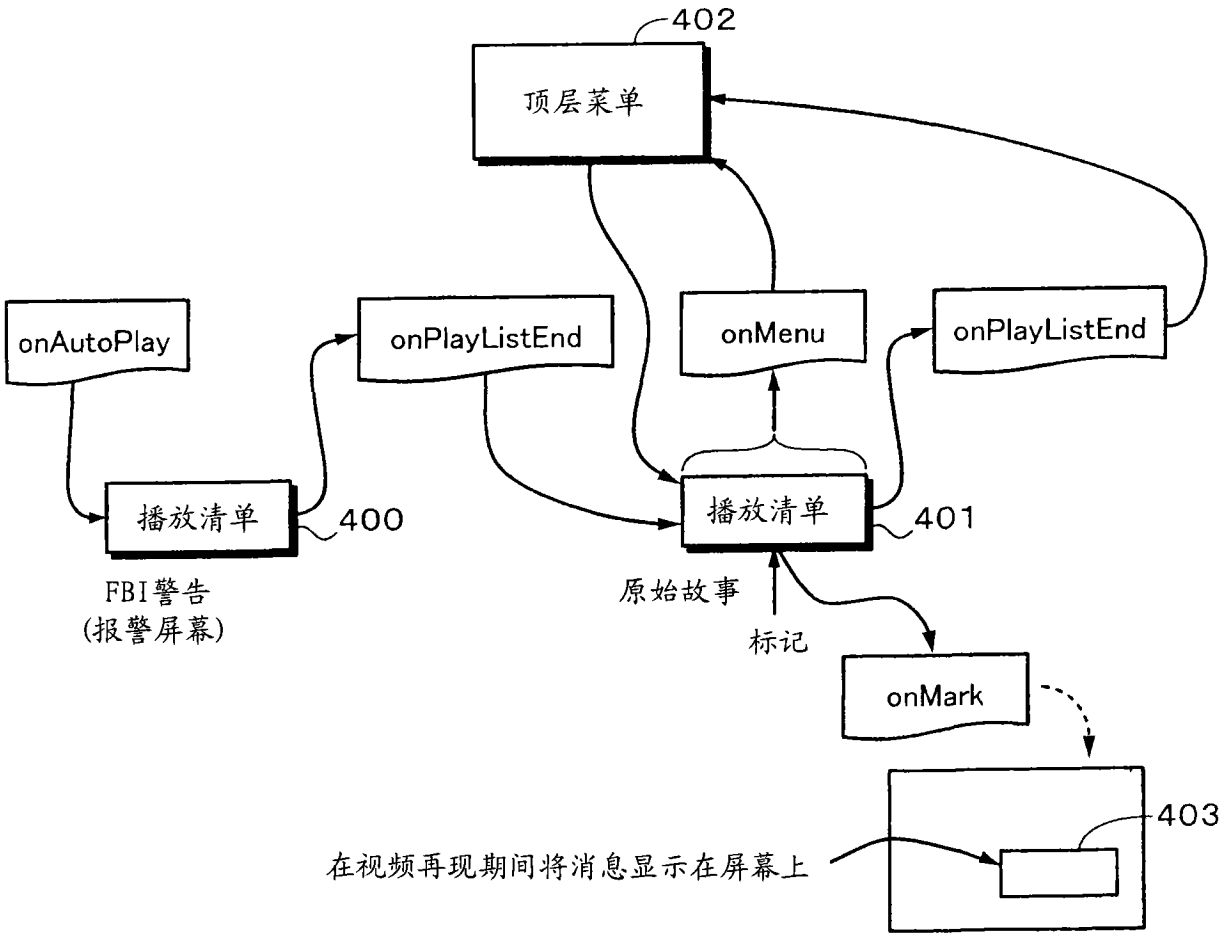

402
顶层菜单
onAutoPlay
播放清单
400
FBI警告
(报警屏幕)
onPlayListEnd
onPlayListEnd
onMenu
播放清单
401
原始故事
标记
onPlayListEnd
onMark
403
在视频再现期间将消息显示在屏幕上

图 16

```
Controller.onAutoPlay = function(){
      //Play PlayList # 1 FBI warning.
      movieplayer.play(1);
}

movieplayer.onPlayListEnd = function(event_info){
     if(event_info.playListNumber == 1){
          // play feature film after FBI warning ends.
          movieplayer.play(2);
     }else{
          // transit to top menu after feature film ends.
          resource.pagetable["top_menu"].open();
    }
}

Controller.onMenu = function(){
     // transfer to top menu with display menu user
operation.
      resource.pagetable["top_menu"].open();
 }

movieplayer.onMark = function(event_info){
    //display dialog when event mark encountered.
    if(event_info.mark_data == 1){
         resource.pagetable["dialog_window_1"].open();
    }
    ...
 }
```

图 17

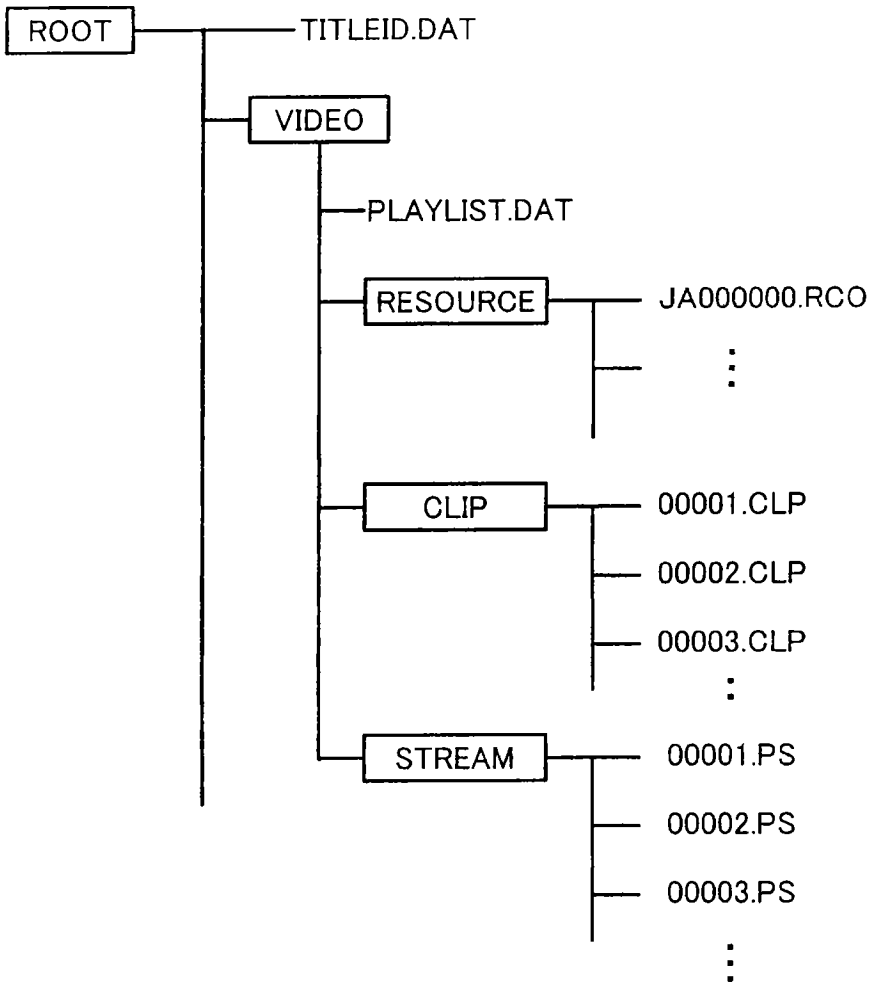

ROOT
TITLEID.DAT
VIDEO
PLAYLIST.DAT
RESOURCE
JA000000.RCO
CLIP
00001.CLP
00002.CLP
00003.CLP
STREAM
00001.PS
00002.PS
00003.PS

图 18

语法	位数	助记符
"PLAYLIST.DAT" {		
name_length	8	uimsbf
name_string	8*255	bslbf
number_of_PlayLists	16	uimsbf
for(i=0; i<number_of_PlayLists; i++){		
PlayList(){ // A PlayList()		
PlayList_data_length	32	uimsbf
// ATTRIBUTE INFORMATION		
reserved_for_word_alignment	15	bslbf
capture_enable_flag_PlayList	1	bslbf
PlayList_name_length	8	uimsbf
PlayList_name_string	8*255	bslbf
//		
number_of_PlayItems	16	uimsbf
for (i=0; i<number_of_PlayItems; i++) {		
PlayItem()		
}		
PlayListMark()		
}		
}		
}		

图 19

语法	位数	助记符
PlayItem() {		
length	16	uimsbf
Clip_Information_file_name_length	16	uimsbf
Clip_Information_file_name	8*Clip_Information_file_name_length	bslbf
reserved_for_word_alignment	15	bslbf
IN_time	33	uimsbf
reserved_for_word_alignment	15	bslbf
OUT_time	33	uimsbf
}		

图 20

语法	位数	助记符
PlayListMark() {		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for(i=0; i < number_of_PlayList_marks; i++) {		
Mark(){		
mark_type	8	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
reserved_for_word_alignment	15	bslbf
mark_time_stamp	33	uimsbf
entry_ES_stream_id	8	uimsbf
entry_ES_private_stream_id	8	uimsbf
mark_data	32	bslbf
mark_name_string	8*24	bslbf
}		
}		
}		

图 21

mark_type	流编码
0	保留
1	章节标记
2	事件标记
3-255	保留

图 22

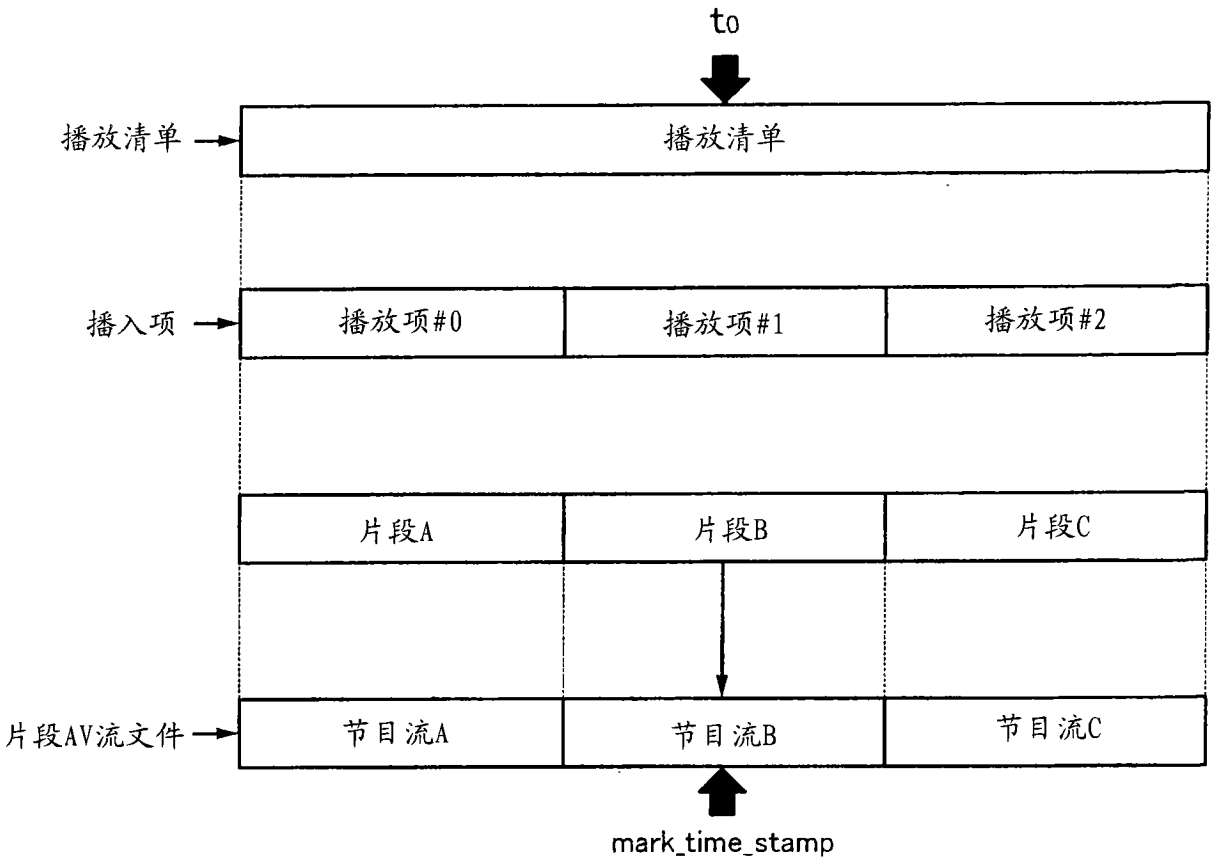


图 23

语法	位数	助记符
XXXXX.CLP{		
reserved_for_word_alignment	15	bslbf
presentation_start_time	33	uimsbf
reserved_for_word_alignment	15	bslbf
presentation_end_time	33	uimsbf
reserved_for_word_alignment	7	bslbf
capture_enable_flag_Clip	1	bslbf
number_of_streams	8	uimsbf
for (i = 0;i < number_of_streams;i++) {		
StreamInfo(){		
length	16	uimsbf
stream_id	8	uimsbf
private_stream_id	8	uimsbf
StaticInfo()		
reserved_for_word_alignment	8	bslbf
number_of_DynamicInfo	8	uimsbf
for (j = 0;j < number_of_DynamicInfo;j++) {		
reserved_for_word_alignment	15	bslbf
pts_change_point	33	uimsbf
DynamicInfo()		
}		
} //end of oneStreamInfo		
}		
EP_map()		
}		

图 24

基本流的类型	stream_id	private_stream_id
视频	0xE0-0xEF	(NONE)
ATRAC 音频	0xBD	0x00-0x0F
LPCM 音频	0xBD	0x10-0x1F
字幕	0xBD	0x80-0x9F

图 25

语法	位数	助记符
StaticInfo() {		
if (stream == VIDEO) {		
reserved_for_word_alignment	16	bslbf
picture_size	4	uimsbf
frame_rate	4	uimsbf
reserved_for_word_alignment	7	bslbf
cc_flag	1	bslbf
} else if (stream == AUDIO) {		
audio_language_code	16	bslbf
channel_configuration	8	uimsbf
reserved_for_word_alignment	3	bslbf
lfe_existence	1	bslbf
sampling_frequency	4	uimsbf
} else if (stream == SUBTITLE) {		
subtitle_language_code	16	bslbf
reserved_for_word_alignment	15	bslbf
configurable_flag	1	uimsbf
}		
}		

图 26

语法	位数	助记符
DynamicInfo(i,j) {		
reserved_for_word_alignment	8	bslbf
if (stream == VIDEO){		
reserved_for_word_alignment	4	bslbf
display_aspect_ratio	4	uimsbf
} else if (stream == AUDIO) {		
reserved_for_word_alignment	4	bslbf
channel_assignment	4	uimsbf
} else if (stream == SUBTITLE) {		
reserved_for_word_alignment	8	bslbf
}		
}		

图 27

语法	位数	助记符
EP_map(){		
reserved_for_word_alignment	8	bslbf
number_of_stream_id_entries	8	uimsbf
for (k=0; k<number_of_stream_id_entries; k++) {		
stream_id	8	bslbf
private_stream_id	8	bslbf
number_of_EP_entries	32	uimsbf
for (i=0; i<number_of_EP_entries; i++) {		
reserved_for_word_align	15	bslbf
PTS_EP_start	33	uimsbf
RPN_EP_start	32	uimsbf
}		
}		
}		

图 28

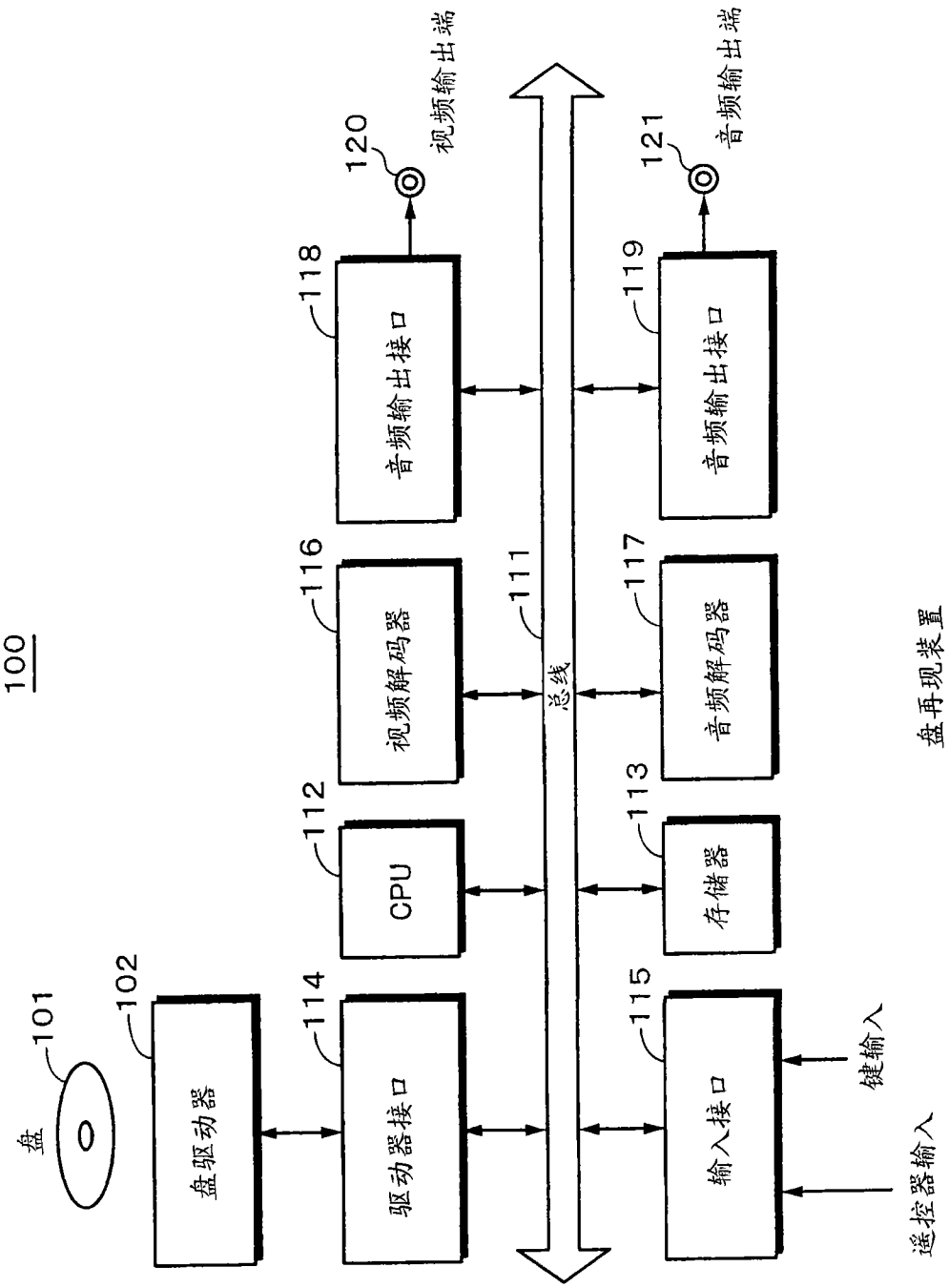

100
盘
101
盘驱动器
102
驱动器接口
114
CPU
112
视频解码器
116
音频输出接口
118
120
视频输出端
总线
111
音频解码器
117
存储器
113
输入接口
115
119
音频输出接口
121
音频输出端
盘再现装置
键输入
遥控器输入

图 29

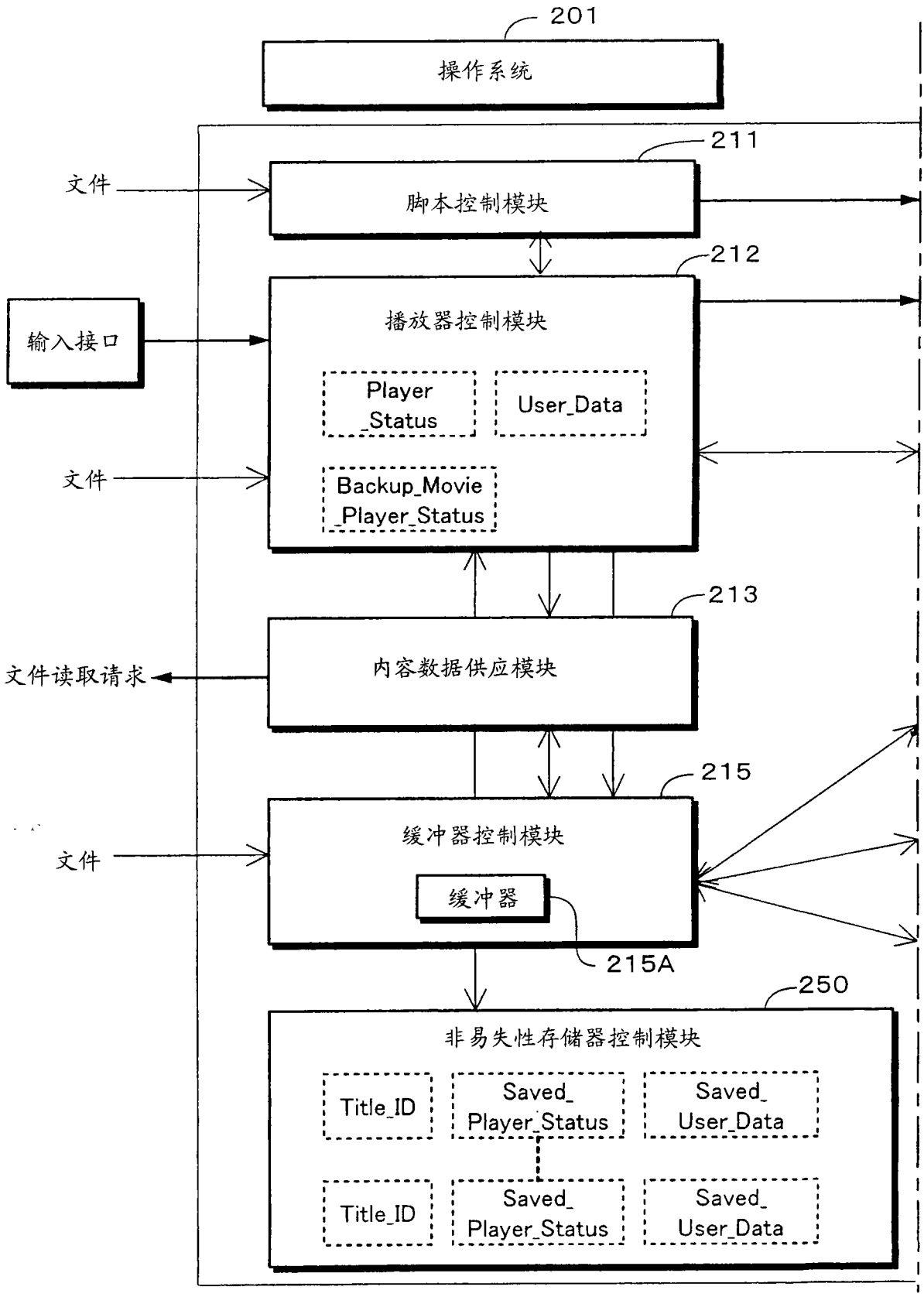

201
操作系统
211
文件
脚本控制模块
212
播放器控制模块
输入接口
Player
_Status
User_Data
文件
Backup_Movie
_Player_Status
213
内容数据供应模块
文件读取请求
215
缓冲器控制模块
文件
缓冲器
215A
250
非易失性存储器控制模块
Title_ID
Saved_
Player_Status
Saved_
User_Data
Title_ID
Saved_
Player_Status
Saved_
User_Data

图 30A

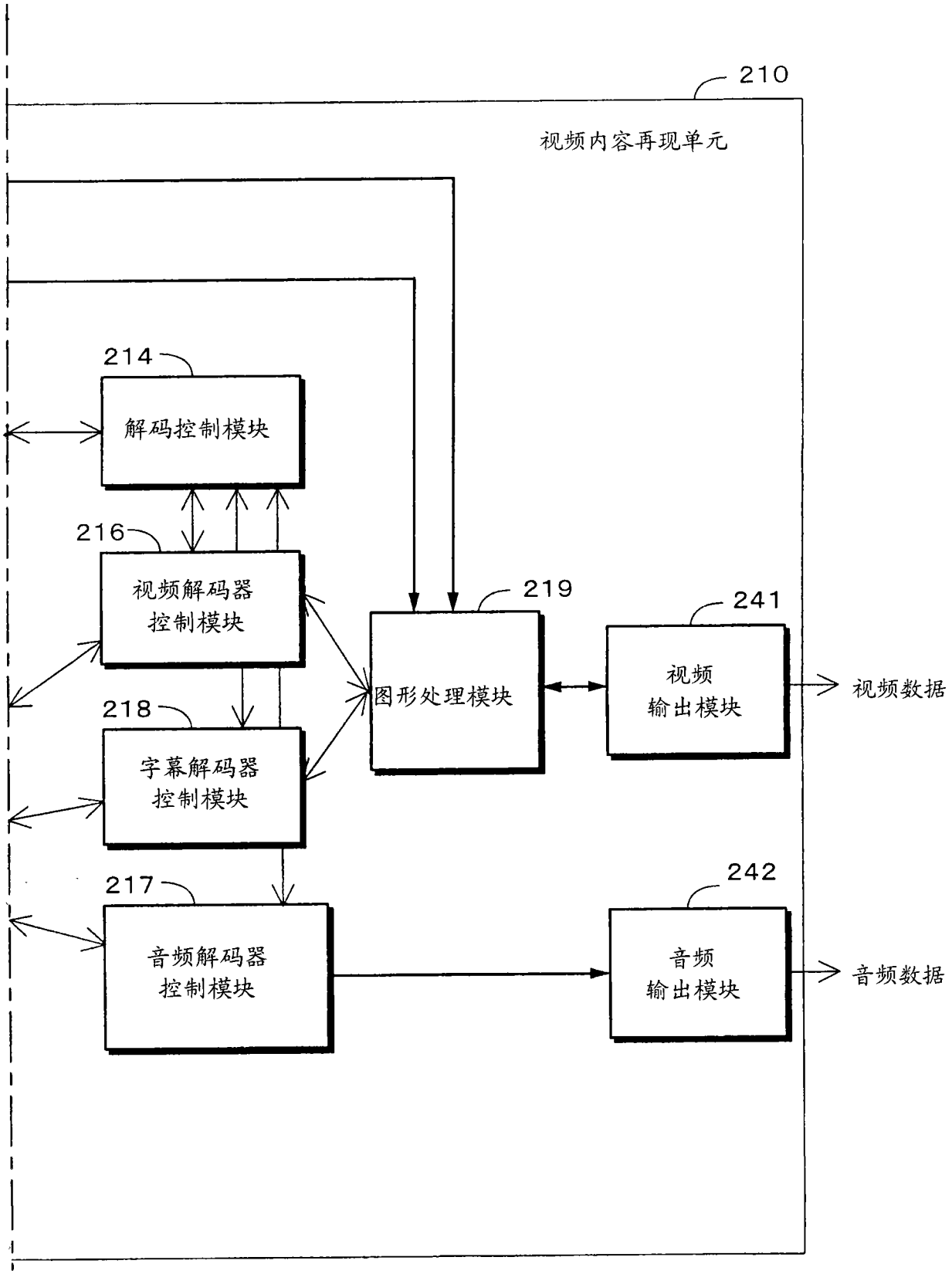

210
视频内容再现单元
214
解码控制模块
216
视频解码器
控制模块
218
字幕解码器
控制模块
217
音频解码器
控制模块
219
图形处理模块
241
视频
输出模块
视频数据
242
音频
输出模块
音频数据

图 30B

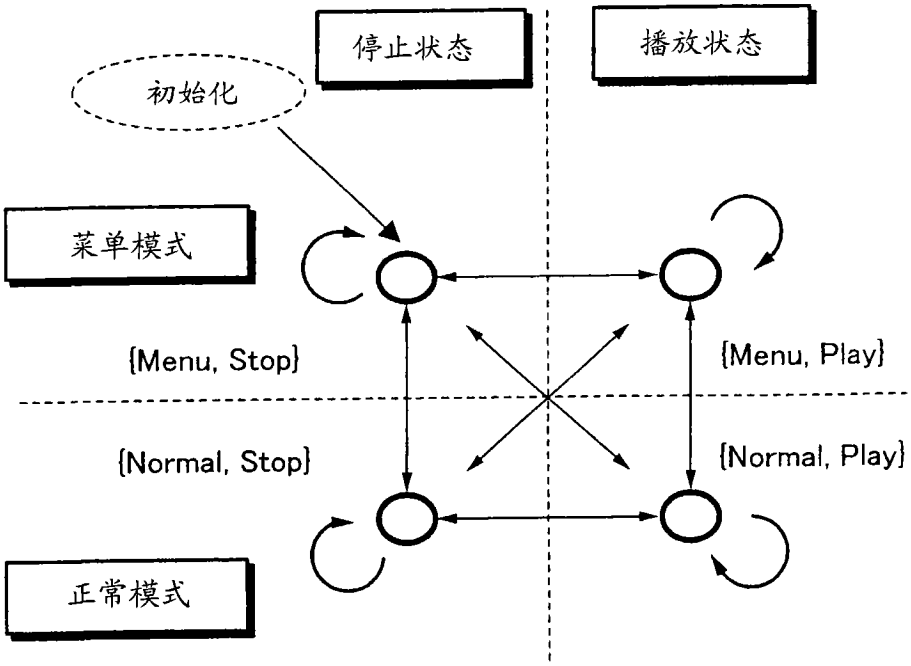

停止状态
播放状态
初始化
菜单模式
{Menu, Stop}
{Menu, Play}
{Normal, Stop}
{Normal, Play}
正常模式

图 31

当前状态		执行方法之后的状态			
		菜单	正常	菜单	正常
菜单	停止	stop(), resume()	stop()	play()	play(), resume()
正常	停止	stop()	stop(), resume()	play()	play(), resume()
菜单	播放	stop()	stop()	play(), resume()	play(), resume()
正常	播放	stop()	stop()	play()	play(), resume()

图 32

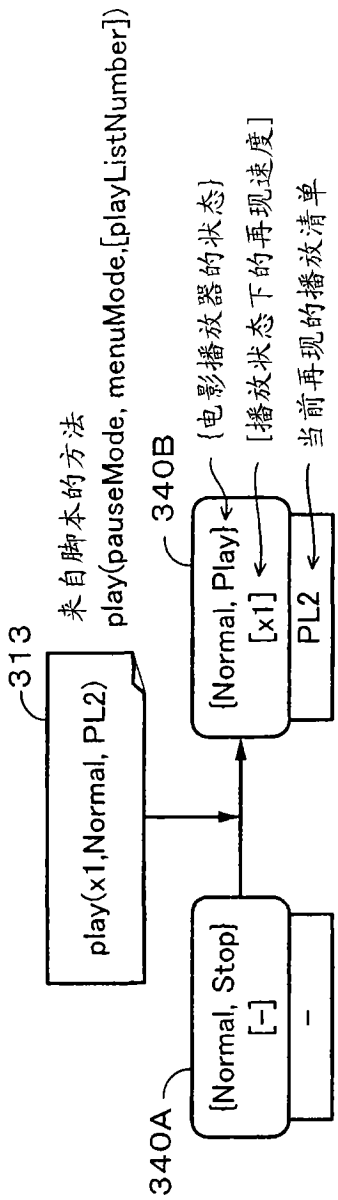


图 33A

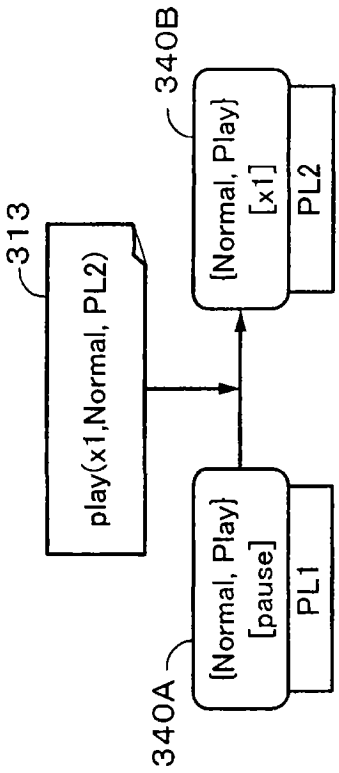


图 33B

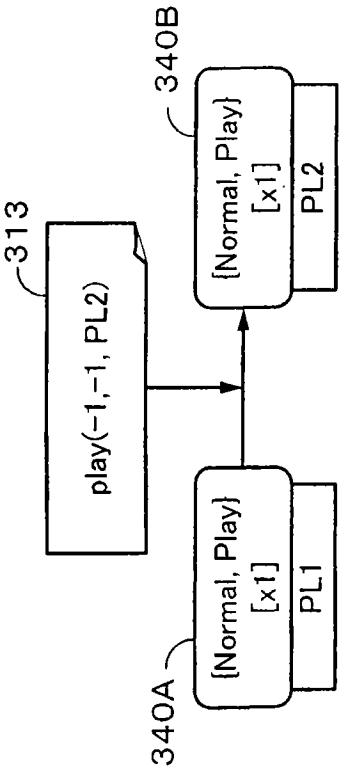


图 33C

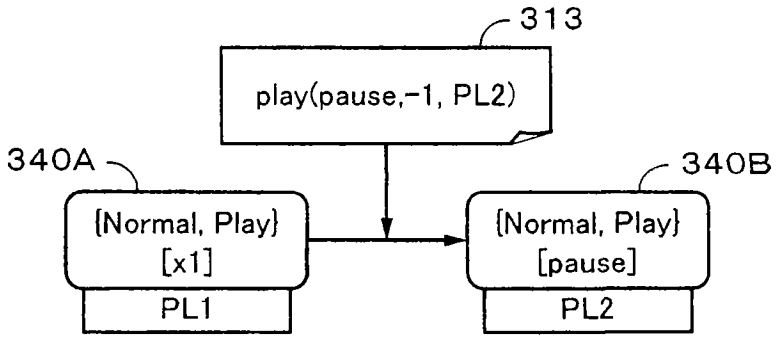


图 33D

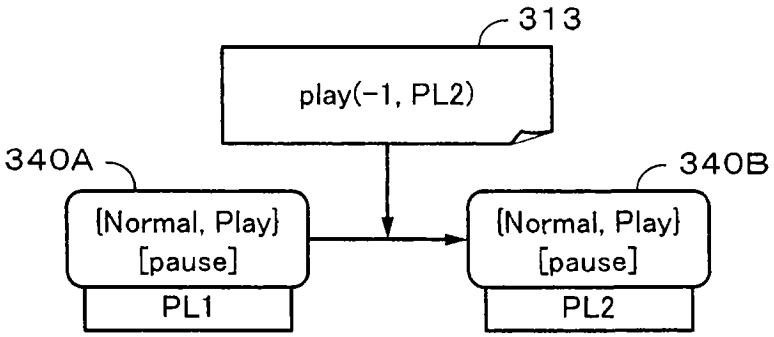


图 33E

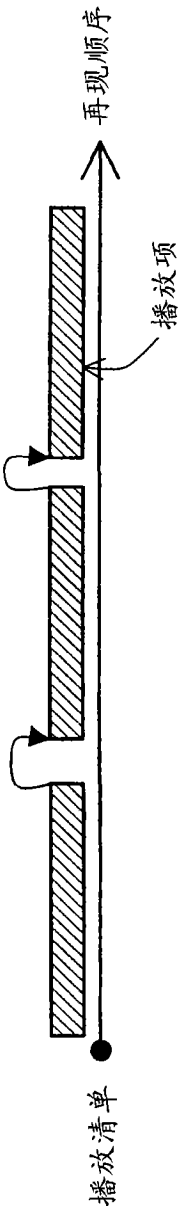


图 34

播放清单再现方法	播放清单开始点和结束点上的操作
向前	一旦到达播放清单的结束点，暂停和继续显示 播放清单的最后画面（在高速再现期间， 即使不对应于跳转点，也显示播放清单的最后画面）
向后	一旦到达播放清单的首端，暂停.

图 35

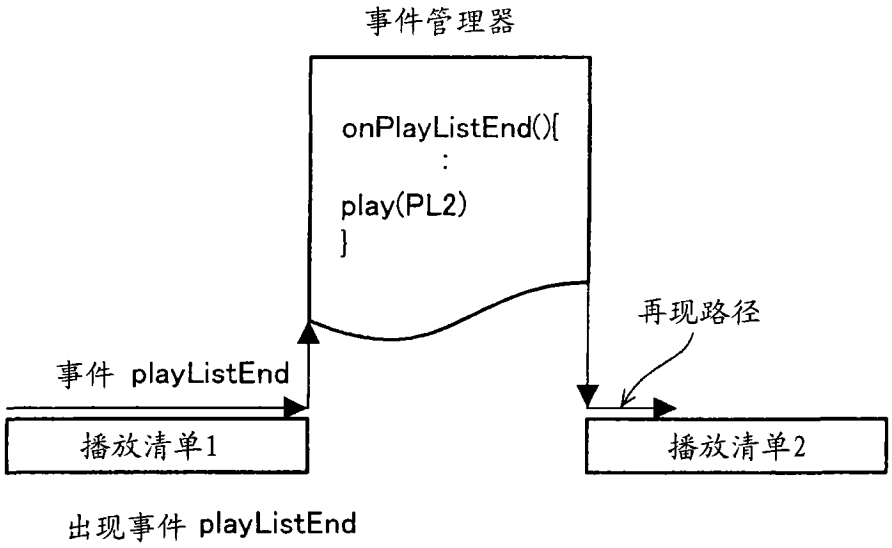


图 36

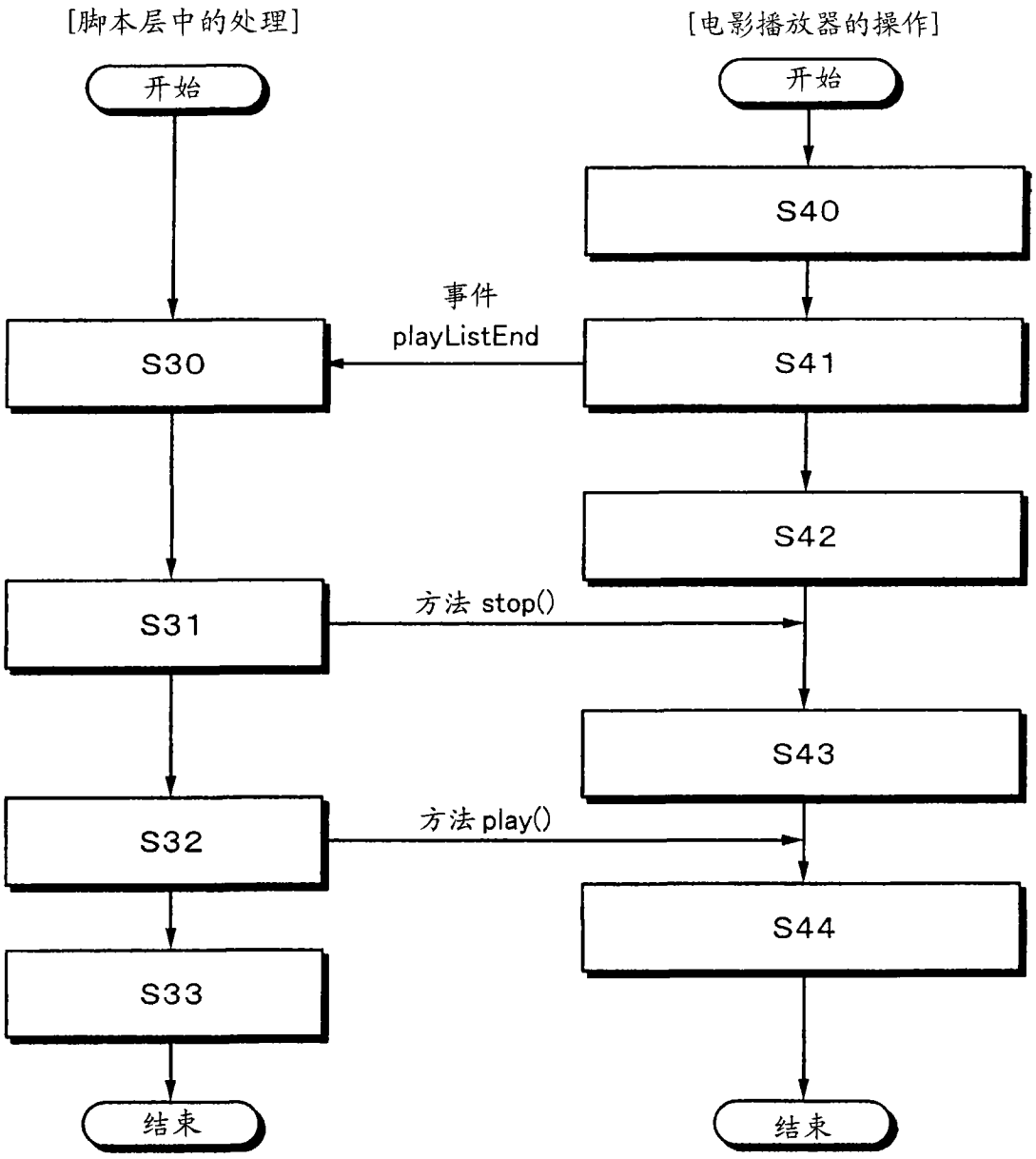

[脚本层中的处理]
[电影播放器的操作]
开始
开始
S40
事件
playListEnd
S30
S41
S42
S31
方法 stop()
S43
S32
方法 play()
S44
S33
结束
结束

图 37

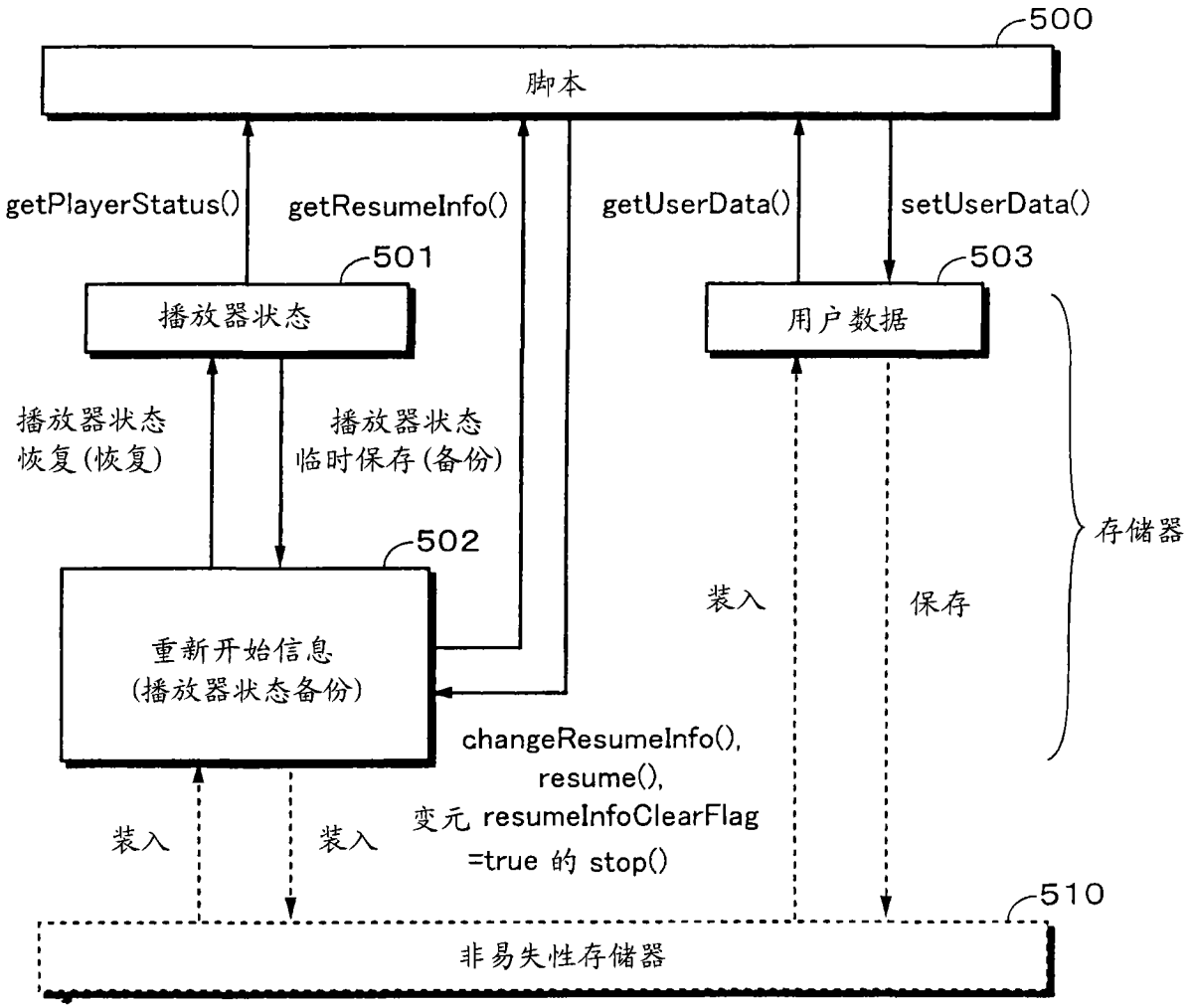

500
脚本
getPlayerStatus()
getResumeInfo()
getUserData()
setUserData()
501
播放器状态
503
用户数据
播放器状态
恢复(恢复)
播放器状态
临时保存(备份)
存储器
502
重新开始信息
(播放器状态备份)
装入
保存
changeResumeInfo(),
resume(),
变元 resumeInfoClearFlag
=true 的 stop()
装入
装入
510
非易失性存储器

图 38

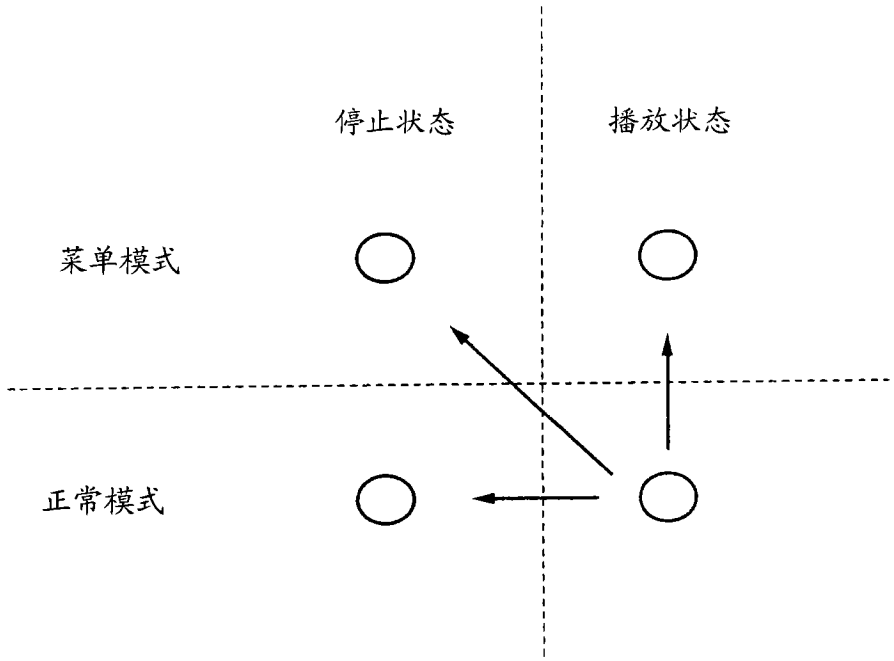


图 39

当前状态		执行方法之后的状态					
		菜单		正常		菜单	正常
		停止		停止		播放	播放
菜单	停止	-		-		-	-
正常	停止	-		-		-	-
菜单	播放	-		-		-	-
正常	播放	利用stop() 转变和备份		利用stop() 转变和备份		利用play() 转变和备份	-

图 40

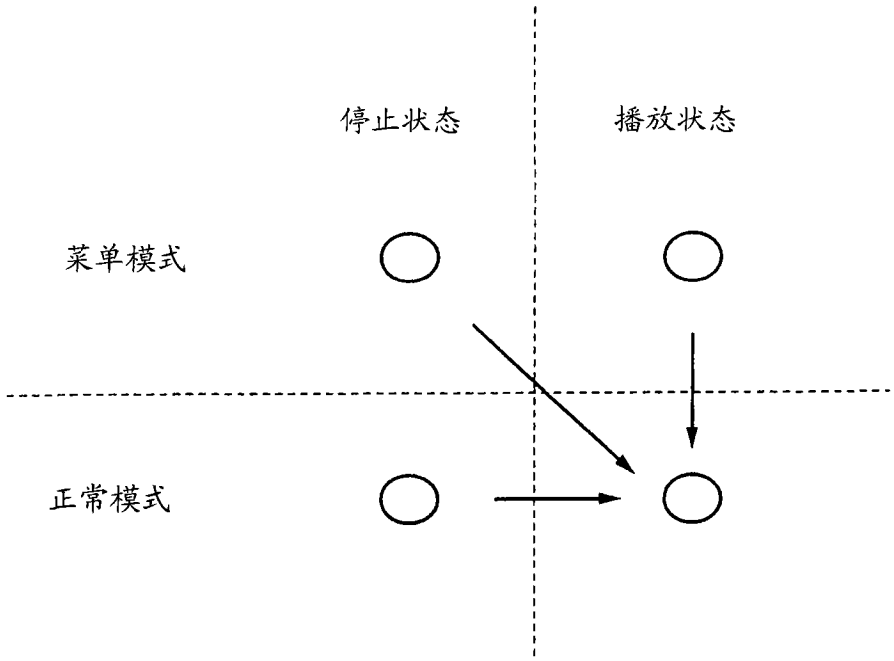


图 41

当前状态		执行方法之后的状态			
		菜单	正常	菜单	正常
		停止	停止	播放	播放
菜单	停止	-	-	-	利用 resume() 转变、恢复和放弃
正常	停止	-	-	-	利用 resume() 转变、恢复和放弃
菜单	播放	-	-	-	利用 resume() 转变、恢复和放弃
正常	播放	-	-	-	(未定义)

图 42

当前状态		执行方法之后的状态			
		菜单	正常	菜单	正常
		停止	停止	播放	播放
菜单	停止	-	-	-	利用play()转变和放弃
正常	停止	-	-	-	利用play()转变和放弃
菜单	播放	-	-	-	利用play()转变和放弃
正常	播放	-	-	-	(未定义)

图 43

当前状态		执行方法之后的状态					
		菜单	正常	菜单	正常		
		停止	停止	播放	播放		
菜单	停止	如果 resumeInfoClearFlag＝True 放弃	如果 resumeInfoClearFlag＝True 放弃	-	-		
正常	停止	如果 resumeInfoClearFlag＝True 放弃	如果 resumeInfoClearFlag＝True 放弃	-	-		
菜单	播放	如果 resumeInfoClearFlag＝True 放弃	如果 resumeInfoClearFlag＝True 放弃	-	-		
正常	播放	如果 resumeInfoClearFlag＝True 放弃	如果 resumeInfoClearFlag＝True 放弃	-	-		

图 44

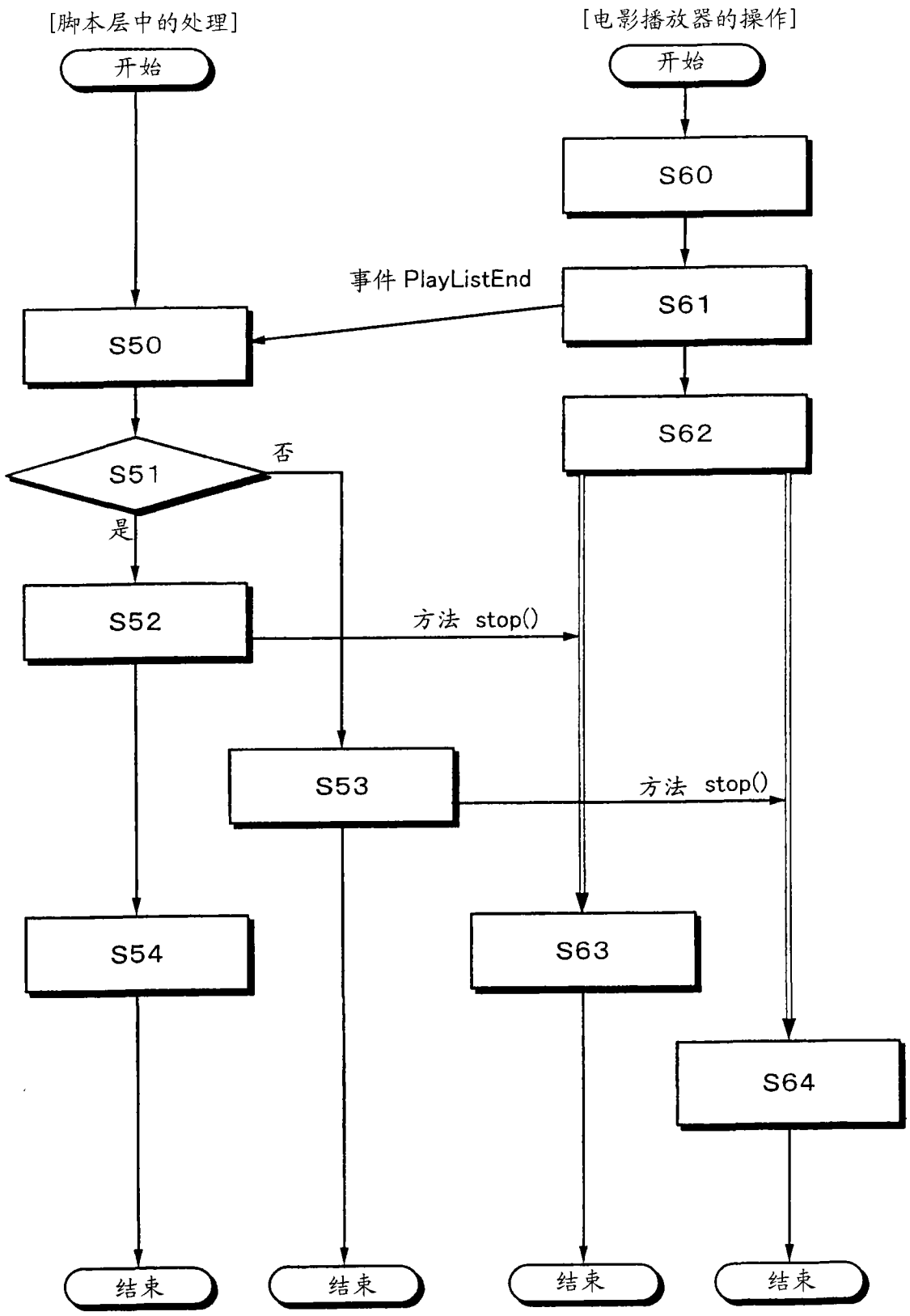

[脚本层中的处理]
[电影播放器的操作]
开始
开始
S60
事件 PlayListEnd
S61
S50
S62
S51
否
是
S52
方法 stop()
S53
方法 stop()
S54
S63
S64
结束
结束
结束
结束

图 45

播放器状态的状态	说明
生成	在生成电影播放器时，也生成了播放器状态. 在生成时，初始化播放器状态，和指示电影播放器的状态的特性指示停止状态，而其它特性不变
值改变	(1) 播放器状态的内容随的再现状态而改变 (2) 当恢复重新开始信息的内容时
通过方法读取值	可以通过方法 getPlayerStatus() 读取
毁灭	当电影播放器结束或被毁灭时

图 46

重新开始信息的状态	说明
生成	当生成电影播放器对象时，保护和在生成的同时初始化重新开始信息的存储区. 一旦初始化，就放弃重新开始信息的内容. 含有安装在其中的非易失性存储器的UMD视频播放器在初始化电影播放器时从非易失性存储器中装入重新开始信息. 同时，装入用户数据.
写入 (播放器状态备份)	当电影播放器从 {Normal,Play}转变成另一个状态时， 将播放器状态备份在重新开始信息中
改变	像 videoNumber, audioNumber, audioFlag, subtitleNumber 和 subtitleFlag 那样 与数据流有关的参数可以通过 changeResumeInfo() 改变.
放弃 (1)	当在正常模式下开始再现播放清单时，放弃重新开始信息的内容. 在放弃之前可以恢复或可以不恢复重新开始信息.

图 47A

放弃(2)	一旦执行了 resumeInfoClearFlag＝True 的方法 stop() 放弃重新开始信息的内容.
恢复播放器状态	一旦在缺乏重新开始信息的情况下执行了方法 resume()，恢复.
通过方法读取值	值可以通过来自脚本的方法 getResumeInfo() 读取. 一旦读取放弃重新开始信息，恢复 playStatus＝0 ，因此， 可以确定重新开始信息存在与否.
毁灭	在电影播放器结束时(毁灭时)，重新开始信息也被毁灭. 安装了非易失性存储器的UMD视频播放器在电影播放器结束时(毁灭时)，将重新开始信息保存在非易失性存储器中，同时保护用户数据.

图 47B

用户数据的状态	说明
生成	当生成电影播放器对象时，保护和在生成的同时初始化重新开始信息的存储区. 一旦初始化，清除用户数据的内容(通过getUserData()恢复长度0的排列). 安装了非易失性存储器的UMD视频播放器在初始化电影播放器时， 从非易失性存储器中装入用户数据.同时，装入重新开始信息.
写入	一旦执行了方法 setUserData(). 写入. 通过方法 setUserData() 将具有64个位的最大长度的整数排列保存在用户数据区中.
读取	可以通过方法 getUserData() 读取. 在未设置用户数据时，恢复长度0的排列.
清除内容	没有从脚本清除用户数据的内容的方法. 可以通过盖写重写内容.
毁灭	在电影播放器结束时(毁灭时)，用户数据也被毁灭.安装了非易失性存储器的 UMD视频播放器在电影播放器结束时(毁灭时)，将用户数据保存在非易失性存储器中. 同时，保存重新开始信息.

图 48

标号说明

101	盘
112	CPU
113	存储器
115	输入接口
116	视频解码器
117	音频解码器
118	视频输出接口
119	音频输出接口
201	操作系统
210	视频内容再现单元
211	脚本控制模块
212	播放器控制模块
214	解码控制模块
215	缓冲器控制模块
216	视频解码器控制模块
217	音频解码器控制模块
218	字幕解码器控制模块
219	图形控制模块
241	视频输出模块
242	音频输出模块
250	非易失性存储器控制模块
300	电影播放器
301	本机实现平台
302	脚本层
310	用户输入
311	控制命令
312	事件
313	方法
320	数据库
321	回放模块
322	解码器引擎
323	特性
323B	播放器状态
324	重新开始信息
500	脚本程序

501 播放器状态区
502 重新开始信息区
503 用户数据区
510 非易失性存储器
S10 用户在再现期间按下" next" 键
S11 生成"uo_playNextChapter()"
S12 从播放清单数据库中了解下一个章节标记的位置
S13 存在下一个章节标记?
S14 中止当前再现
S15 跳转到下一个章节标记所指的位置和开始视频再现
S16 生成标记事件
S17 开始执行与标记事件相对应的事件管理器
S18 从生成事件时通知的信息中了解章节号
S19 显示指示章节首端的信息
S20 结束事件管理器的执行
S30 开始执行事件管理器"onPlayListEnd".
S31 通过事件管理器"onPlayListEnd"指定方法"stop()"
S32 通过事件管理器"onPlayListEnd"指定方法
"play(pauseMode,menuMode,PlayListNumber)"
S33 结束事件管理器"onPlayListEnd"的执行
S40 使播放清单再现到末端
S41 电影播放器将"PlayListEnd"事件发送给脚本
S42 电影播放器在显示播放清单的最后画面的同时转变成暂停状态
S43 电影播放器转变成停止状态. 停止最后画面的显示(出现黑屏)
S44 电影播放器开始再现指定的播放清单
S50 开始执行事件管理器"onPlayListEnd".
S51 结束创作情节?
S52 将放弃重新开始信息的"stop()"发送到电影播放器
S53 将不放弃重新开始信息的"stop()"发送到电影播放器
S54 (取决于脚本描述,这里执行方法"end()")
S60 使播放清单再现到末端
S61 将事件"PlayListEnd"事件通知脚本
S62 在显示播放清单的最后画面的同时转暂停
S63 电影播放器转变成停止状态. 清除重新开始信息
S64 电影播放器转变成停止状态. 将播放器状态备份成重新开始信息