(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 101916580 B
(45)授权公告日 2012.03.07

(21)申请号 201010274460.5

(22)申请日 2005.07.21

(30)优先权数据
2004-214916 2004.07.22 JP
2004-268512 2004.09.15 JP
2004-293043 2004.10.05 JP
2004-369701 2004.12.21 JP
2005-099410 2005.03.30 JP

(62)分案原申请数据
200580024823.7 2005.07.21

(73)专利权人 松下电器产业株式会社
地址 日本大阪府

(72)发明人 田中敬一 大户英隆 大芦雅弘

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 李光颖 王英

(51)Int.Cl.
G11B 27/10(2006.01)
G11B 27/32(2006.01)

(56)对比文件
WO 2004/030356 A1,2004.04.08, 全文.
WO 2004/049710 A1,2004.06.10, 全文.
US 2002/0199205 A1,2002.12.26, 全文.

审查员 树奇

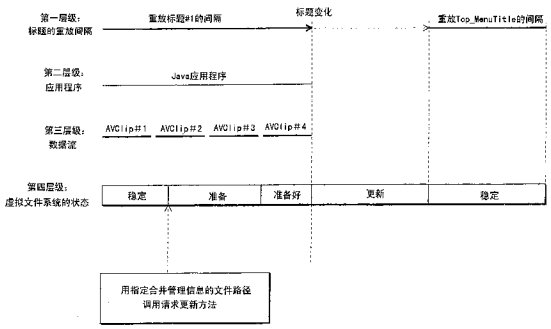
权利要求书 2 页 说明书 24 页 附图 44 页

(54)发明名称

用于执行应用程序同步重放的重放装置

(57)摘要

一种结合应用程序用于播放数字流的重放装置包括包管理单元,用于根据合并管理信息通过合并记录在第一记录介质和第二记录介质上的文件来生成包信息,和选择单元,用于检测来自包信息的多个可播放标题,并且选择所检测到的标题中的一个作为当前标题。在应用程序通过指定新合并管理信息请求包管理单元更新合并管理信息之后,在更新包信息之前,包管理单元将从新指定合并管理信息引用的文件变为只读,并且在数字流重放由于选择单元选择的当前标题变化而停止时,包管理单元根据新指定的合并管理信息通过组合记录在第一和第二记录介质上的文件生成新包信息。



CN 101916580 B

1. 一种边与执行中的应用程序相联动、边播放数字流的重放装置，该重放装置的特征在于，包括：

包管理单元，用于根据记录在可重写记录介质中的合并管理信息，来合并记录于只读记录介质中的文件和记录于上述可重写记录介质中的文件，从而生成包信息，并将上述包信息的生成所用的合并管理信息所指示的文件变更为只读属性，上述合并管理信息指示上述可重写记录介质中的用于与记录于上述只读记录介质中的文件进行合并的文件；

选择单元，用于从上述包信息中检测出多个可播放标题，并将所检测出的多个可播放标题其中的一个选择为当前标题；

Java 平台，用于执行显示在与上述当前标题相关的应用程序管理表中的应用程序；以及

重放控制引擎，用于根据包含于上述包信息中的、且在上述当前标题中指示的播放列表信息，控制数字流的重放，其中

当被执行的上述应用程序通过指定与上述包信息的生成所用的合并管理信息不同的、且存储于上述可重写记录介质中的新的合并管理信息，来请求上述包管理单元对上述生成的包信息进行更新时，

上述包管理单元，

在更新上述包信息之前，把上述指定的新的合并管理信息所指示的、且存储于上述可重写记录介质中的文件变更为只读属性，以及

在由于上述选择单元进行的当前标题改变而使得数字流重放和应用程序执行都停止的时刻，上述包管理单元根据上述指定的新的合并管理信息，将记录于上述只读记录介质中的文件和记录于上述可重写记录介质中的文件进行组合，来生成新的包信息。

2. 根据权利要求 1 所述的重放装置，在上述 Java 平台执行与上述应用程序不同的、显示在与上述当前标题相关的应用程序管理表中且显示在与新的当前标题相关的应用程序管理表中的标题未绑定应 用程序，并且请求了对上述包信息进行更新的情况下，由于上述当前标题改变，通过上述 Java 平台终止上述应用程序及上述标题未绑定应用程序，

在上述新的包信息被生成之后，上述 Java 平台启动上述终止的标题未绑定应用程序以及显示在与上述新的当前标题相关的应用程序管理表中的应用程序。

3. 一种边与执行中的应用程序相联动、边播放数字流的重放方法，其特征在于，包括：

包信息生成步骤，用于根据记录在可重写记录介质中的合并管理信息，来合并记录于只读记录介质中的文件和记录于上述可重写记录介质中的文件，从而生成包信息，上述合并管理信息指示上述可重写记录介质中的用来与记录于上述只读记录介质中的文件进行合并的文件；

属性变更步骤，用来将上述包信息的生成所使用的合并管理信息所指示的文件变更为只读属性；

选择步骤，用于从上述包信息中检测出多个可播放标题，并将所检测出的多个可播放标题其中的一个选择为当前标题；

处理步骤，用于执行显示在与上述当前标题相关应用程序管理表中的应用程序；以及

重放控制步骤，用于根据包含于上述包信息中的、且在上述当前标题中指定的播放列表信息，来控制数字流的重放，其中

当被执行的上述应用程序通过指定与上述包信息的生成所用的合并管理信息不同的、且存储于上述可重写记录介质中的新的合并管理信息，来请求更新上述生成的包信息时，

在更新上述包信息之前，通过上述属性变更步骤把上述指定的新的合并管理信息所指示的、且存储于上述可重写记录介质中的文件变更为只读属性，以及

在由于上述选择步骤中进行的当前标题改变而使得数据流重放和应用程序执行都停止的时刻，上述包信息生成步骤根据上述指定的 新的合并管理信息，将记录于上述只读记录介质中的文件和记录于上述可重写记录介质中的文件进行组合，来生成新的包信息。

4. 根据权利要求 3 所述的重放方法，在上述处理步骤执行与上述应用程序不同的、显示在与上述当前标题相关的应用程序管理表中且显示在与新的当前标题相关的应用程序管理表中的标题未绑定应用程序，并且请求了对上述包信息进行更新的情况下，由于上述当前标题改变，通过上述处理步骤终止上述应用程序及上述标题未绑定应用程序，

在上述新的包信息被生成之后，上述处理步骤启动上述终止的标题未绑定应用程序以及显示在与上述新的当前标题相关的应用程序管理表中的应用程序。

用于执行应用程序同步重放的重放装置

[0001] 本申请为 2005 年 7 月 21 日提交的申请号为 200580024823.7、名称为“用于执行应用程序同步重放的重放装置”的申请的分案申请。

技术领域

[0002] 本发明属于数字流和应用程序同步重放的技术领域。

背景技术

[0003] 应用程序同步重放是用于在重放装置或用于播放视频的类似装置中运行 Java（注册商标）应用程序的同时播放数字流的一项技术，并且属于在今后制造的消费品中将吸引大量关注的技术领域。用于同步的应用程序和数字流以被称为“标题”的重放为单位彼此相联系。下面描述传统的重放装置。传统的重放装置由虚拟文件系统单元、模块管理器、重放单元和平台单元构成。

[0004] 虚拟文件系统单元管理由重放装置访问的多个记录介质，例如光记录介质（例如 DVD-ROM，BD-ROM，在后面作为例子描述后者）和磁记录介质（例如硬盘），并且生成整体地显示这些记录介质所记录内容的包信息。

[0005] 每条包信息被称为虚拟包，其在记录于这些记录介质上的数据流和应用程序（在下文简称为“应用程序”）的重放 / 执行中被提交，就如同它们实际记录在单一包中。

[0006] 模块管理器选择多个标题中的一个作为当前标题。

[0007] 重放单元从包信息所示的数字流中播放构成当前标题的数字流（即记录在光或磁记录介质上）。

[0008] 平台单元从包信息所示的应用程序中运行构成当前标题的应用程序（即记录在光或磁记录介质上）。

[0009] 根据以上结构，记录在不同记录介质例如 BD-ROM 和硬盘上的数字流和应用程序可以作为单一（虚拟）包来对待，并且重放装置能够以标题为 单位实现重放控制。

[0010] 以上结构中的标题不仅用于记录在用于重放的 BD-ROM 上的数字流而且用于记录在硬盘上的数字流。由于硬盘是可重写的记录介质，因此标题的组成要素可能通过重放部分地被替换，例如当使用新采集的数字流新生成包信息时。

[0011] 在很大程度上认为这样的替换引起的情况难以恢复例如在重放期间由重放单元的异常操作产生的视频中断。

[0012] 如果重放装置中的重放单元顺序地检验例如用于重放的数据流和重放控制信息的要素的一致性，可以预期通过重放部分地替换重放对象。然而，假设需要检验的信息类型互异，这对重放单元造成了很大负担。

[0013] 为了绝对保证设备的稳定操作，理想的是当在硬盘上的记录内容中有变化时重启设备自身，就象当你安装软件时那样。然而，假设在任何地方重启设备花费几秒到几分钟的处理时间，这并不是用户能轻松承担的事情。

发明内容

[0014] 本发明的目标是提供一种重放装置，即使在重放对象中有变化时所述重放装置也能够实现稳定重放。

[0015] 为了解决以上问题，本发明是一种结合应用程序播放数字流的重放装置，其包括包管理单元，用于根据合并管理信息通过合并记录在第一记录介质和第二记录介质上的文件所述包管理单元生成包信息，和选择单元，用于检测来自包信息的多个可播放标题，并且选择检测到的标题中的一个作为当前标题。当应用程序通过指定新合并管理信息请求包管理单元更新合并管理信息之后，在更新包信息之前包管理单元将从新指定合并管理信息引用的文件变为只读，并且在数字流重放由于当前标题由选择单元改变而停止时，包管理单元根据新指定的合并管理信息通过组合记录在第一和第二记录介质上的文件生成新包信息。

[0016] 由于具有以上结构，本发明保证了在应用程序通过指定新合并管理信息请求包管理单元更新合并管理信息的情况下，从新指定的合并管理信息引用的文件的内容从提出更新请求的时间直到当前标题变 化都未被改变，这是因为在更新包信息之前包管理单元将从新指定的合并管理信息引用的文件设置为只读。而且，通过在数字流重放由于当前标题变化被停止时更新包信息，即使例如用于重放 / 执行的数字流或应用程序由于更新被替换也不会引起重放装置的异常操作。

[0017] 因而，由于可以通过动态地合并记录在第一和第二两个记录介质上的文件并且同时保证重放装置的稳定操作而实现重放控制，扩大了电影作品的表达宽度。

[0018] 在这里，包管理单元可以被配置成如果请求更新的应用程序的许可标记 (tag) 中的信息指示应用程序已被分配更新包信息的权利，则更新包信息，如果该信息指示未被分配更新的权利，则拒绝更新请求并且执行异常处理。

[0019] 该结构能够阻止未授权应用程序请求的包信息的更新。

附图说明

[0020] 图 1 是系统图；

[0021] 图 2 显示了 BD-ROM 100 的内部结构；

[0022] 图 3 系统地显示了怎样构造扩展名为“m2ts”的文件；

[0023] 图 4 显示了 PL 信息的结构；

[0024] 图 5 显示了 AVClip 和 PL 之间的关系；

[0025] 图 6 显示了使用四个 Clip_Information_file_names 的分批指定；

[0026] 图 7 显示了 PLMark 信息的内部结构；

[0027] 图 8 显示了使用 PLMarks 的章节定义；

[0028] 图 9 显示了 Subpath 信息的内部结构；

[0029] 图 10 显示了同步指定和 SubClip 时间轴上的重放间隔定义；

[0030] 图 11A 显示了被包含在档案文件中的程序数据；

[0031] 图 11B 显示了类文件的内部结构；

[0032] 图 12 显示了 BD-J 对象的内部结构；

[0033] 图 13 显示了 INDEX.BDMV 的内部结构；

[0034] 图 14 显示了本地存储器中的目录结构；

[0035] 图 15 显示了由存储在本地存储器中的 PL 信息定义的播放列表 重放时间轴的类型；

[0036] 图 16A 显示了存储在 BD-ROM 上的 AVClip 和存储在本地存储器中的 Java 应用程序和 AVClip；

[0037] 图 16B 显示了作为单一标题来对待的 Java 应用程序和 AVClip#1-#4；

[0038] 图 17 显示了合并管理信息文件的典型内部结构；

[0039] 图 18 显示了重放装置的硬件配置；

[0040] 图 19 描绘了由硬件和存储在指令 ROM 21 上的软件组成的以层状结构重布置的元件；

[0041] 图 20 显示了 Java 虚拟机 30 的内部结构；

[0042] 图 21 显示了由标题变化产生的状态转变；

[0043] 图 22 显示了由虚拟文件系统单元 38 典型地产生虚拟包信息；

[0044] 图 23A 显示了整个光盘的时间轴；

[0045] 图 23B 显示了整个光盘的时间轴的结构；

[0046] 图 24 是由 Java 应用程序执行的下载处理的流程图；

[0047] 图 25 是更新“准备”过程的流程图；

[0048] 图 26 是虚拟包“更新”过程的流程图；

[0049] 图 27 是由模块管理器执行的处理的流程图；

[0050] 图 28 是 PL 重放过程的流程图；

[0051] 图 29 显示了在标题变化期间怎样更新虚拟包信息；

[0052] 图 30 显示了将当前合并管理信息文件发送到服务器的 Java 应用程序；

[0053] 图 31 显示了下载内容文件、新合并管理信息文件和新签名信息文件的 Java 应用程序；

[0054] 图 32 显示了向虚拟文件系统单元 38 提出更新请求的 Java 应用程序；

[0055] 图 33 显示了合并管理和签名信息文件的替换，和内容文件的映射；

[0056] 图 34A 是当标题调用之后暂停当前标题的重放时由重放控制引擎执行的处理的流程图；

[0057] 图 34B 是当所调用的标题的重放结束之后继续重放原始标题时由重放控制引擎执行的处理的流程图；

[0058] 图 35 显示了实施方式 3 中的合并管理信息文件；

[0059] 图 36 显示了关于实施方式 3 的 Java 应用程序，该应用程序请求更新虚拟包；

[0060] 图 37 显示了由使用合并管理信息文件的常驻应用程序向用户显示的典型附加内容列表；

[0061] 图 38 是关于实施方式 3 的从 BD-ROM 的装载直到重放的处理流程的流程图；

[0062] 图 39 显示了当请求虚拟包更新时被指定的有效间隔；

[0063] 图 40 是关于实施方式 4 的从 BD-ROM 的装载（或重放装置的重启）直到重放的处理流程的流程图；

[0064] 图 41 显示了用于筛选虚拟包更新请求的许可请求文件；

[0065] 图 42 显示了强加在用于合并的本地存储器中的目录上的访问限制；

[0066] 图 43 显示了其生命周期被限制到单一标题的 Java 应用程序和其生命周期持续多个标题的 Java 应用程序；

[0067] 图 44 显示了在标题变化期间当更新虚拟包时在未绑定标题 (title-unbound) 的应用程序上执行的处理；

[0068] 图 45 是考虑未绑定标题的应用程序的标题变化处理的流程图；和

[0069] 图 46 是在 INDEX.BDMV 文件变化之后的虚拟包更新的流程图。

具体实施方式

[0070] 实施例 1

[0071] 下面描述与本发明相关的记录介质的实施例。首先，描述关于本发明的重放装置的实现形式中的一种使用形式。图 1 显示了关于本发明的重放装置的典型使用形式。在图 1 中，关于本发明的重放装置是重放装置 200。重放装置 200 用于在家庭影院系统中提供电影作品，所述家庭影院系统例如由重放装置 200，遥控器 300，和电视 400 组成。

[0072] 关于本发明的重放装置的典型使用形式如上所述。接着描述用于由关于本发明的重放装置进行重放的记录介质。在给出的例子中，由关于本发明的重放装置播放的记录介质是 BD-ROM 100（光记录介质）。图 2 显示了 BD-ROM 100 的内部结构。

[0073] 在图 2 中的第四层级显示了 BD-ROM 100，而在第三层级显示了 BD-ROM 上的轨道。图 2 中描绘的轨道由从 BD-ROM 的内圆周到外圆周的轨道螺旋产生，其对于侧面已经被画出。该轨道由导入区、卷区和导出区组成。图 2 中的卷区具有由物理层、文件系统层和应用层组成的层状结构。使用目录结构表达 BD-ROM 的应用格式给出了图 2 中的第一层级。BDMV 目录放置在 BD-ROM 中的第一层级的根目录下。

[0074] INDEX.BDMV 文件布置在 BDMV 目录中，并且在 BDMV 目录下存在五个子目录，它们被称为 PLAYLIST 目录，CLIPINF 目录，STREAM 目录，BDJA 目录，和 BDBJ 目录。

[0075] STREAM 目录存储形成主数字流的文件，将扩展名“m2ts”分配给该文件 (00001.m2ts)。

[0076] 在 PLAYLIST 目录中存在扩展名为“mpls”的文件 (00001.mpls)。

[0077] 在 CLIPINF 目录中存在扩展名为“clpi”的文件 (00001.clpi)。

[0078] 在 BDJA 目录中存在扩展名为“jar”的文件 (00001.jar)。

[0079] 在 BDBJ 目录中存在扩展名为“bdbj”的文件 (00001.bdbj)。

[0080] 接着描述这些文件。

[0081] AVClip

[0082] 首先描述扩展名为“m2ts”的文件 (00001.m2ts)。图 3 系统地显示了怎样构造扩展名为“m2ts”的文件。该文件存储 AVClip。通过多路复用 TS 包构成 AVClip（中间层级），通过将由多个视频帧（图片 pj1，pj2，pj3）组成的视频流和由多个音频帧组成的音频流（上面第一层级）首先转换成 PES 包（上面第二层级），然后转换成 TS 包（上面第三层级），并且以相同的方式将字幕图形显示 (P 图形或 PG) 流（下面第一层级）和对话交互图形 (I 图形或 IG) 流（下面第一层级）转换成 TS 包（下面第三层级）来产生所述 TS 包。

[0083] 除了如图 3 中所示通过多路复用获得的 AVClip 之外，也存在不是由多 路复用产

生的AVClip。这些被称为SubClip，并且包括构成音频流、图形流或文本字幕流(TextST流)等的AVClip。

[0084] 剪辑信息

[0085] 扩展名为“clpi”的文件(00001.clpi)是与AVClip对应的一条剪辑信息。剪辑信息是管理信息，其包含显示GOP的头位置的EP_map，和诸如AVClip中的流的编码格式、帧频、比特率和分辨率等的信息。

[0086] PL信息

[0087] 扩展名为“mpls”的文件(00001.mpls)存储播放列表(PL)信息。PL信息通过参考AVClip定义播放列表。图4显示了PL信息的结构，如图的左侧所示，PL信息由MainPath信息，PLMark信息和SubPath信息构成。

[0088] MainPath信息“MainPath()”由箭头mp1所指示的播放项目信息“PlayItem()”组成。播放项目是通过在一个或多个AVClip时间轴上指定In_time和Out_time来定义的重放间隔。多条播放项目信息的放置定义由多个重放间隔组成的列表。图4中的箭头mp2显示了播放项目信息的内部结构的特写。如图4中所示，播放项目信息由In_time，Out_time和显示相应的AVClip的Clip_Information_file_name组成。图5显示了AVClip和PL之间的关系。第一层级显示AVClip的时间轴，而第二层级显示PL的时间轴。PL信息包括三条播放项目信息(PlayItem#1-#3)，其中三个重放间隔由PlayItem#1、#2和#3的In_times和Out_times定义。当这些重放间隔以线布置时定义不同于AVClip的时间轴。这是在第二层级所示的PL时间轴。因而允许通过播放项目信息中的定义来定义不同于AVClip的时间轴。

[0089] 通常，在任何一个时间指定一个AVClip，尽管多个AVClip的分批指定也是可能的。使用播放项目信息中的Clip_Information_file_names执行AVClip的分批指定。图6显示了使用四个Clip_Information_file_names分批指定AVClip。图中的第一至第四层级显示了四个AVClip时间轴(时间轴AVClips#1-#4)，而第五层级显示了PL时间轴。四个时间轴由包括在播放项目信息中的这四个Clip_Information_file_names指定。这允许由In_times和Out_times定义四种备选的可播放的重放间隔。结果，在PL时间轴上定义由多条可切换角视频(所称的多角度间隔)组成的间隔。

[0090] PLMark信息“PLMark()”指定PL时间轴上的任意间隔作为章节。图7显示了PLMark信息的内部结构，该内部结构包括箭头pm1所指示的ref_to_playitem_id和Mark_time_stamp。图8显示了使用PLMark定义章节。图8中的第一层级显示AVClip时间轴，而第二层级显示PL时间轴。图8中的箭头pk1和pk2显示PLMark中的播放项目(ref_to_playitem_id)和时间点(Mark_time_stamp)的指定。由于这些指定在PL时间轴上定义三个章节(Chapters#1-#3)。这完成了对PLMark的描述。

[0091] 接着描述SubPath信息。

[0092] SubPath信息“SubPath()”通过在SubClip时间轴上指定In_time和Out_time定义一个或多个重放间隔，并且具有图9中所示的内部结构。SubPath信息由箭头sh1所指示的子播放项目信息“SubPlayItem()”组成。在箭头sh2所标记的特写中，子播放项目信息由Clip_Information_file_name，In_time，Out_time，Sync_PlayItem_Id，和Sync_Start_PTS_of_PlayItem组成。使用包括在子播放项目信息中的Clip_Information_file_name，

In_time 和 Out_time 指定 SubClip 时间轴上的 In_times 和 Out_times。Sync_PlayItem_Id,和 Sync_Start_PTS_of_PlayItem 用于使 SubClip 时间轴和 PL 时间轴上的重放间隔同步。这允许在 SubClip 时间轴和 PL 时间轴上的处理以彼此同步进行。

[0093] 图 10 显示了同步指定和 SubClip 时间轴上的重放间隔的定义。图 10 中的第一层级显示 PL 时间轴,而第二层级显示 SubClip 时间轴。图 10 中的 SubPlayItem.In_time 和 SubPlayItem.Out_time 分别显示重放间隔的开始和结束。因而显然重放间隔也在 SubClip 时间轴上被定义。箭头 Sn1 所标记的 Sync_PlayItem_Id 显示播放项目的同步指示,而箭头 Sn2 所标记的 Sync_Start_PTS_of_PlayItem 指定 PL 时间轴上的播放项目期间的一点。

[0094] BD-ROM 中的 PL 信息的特征在于它使得定义允许切换 AVClip 的多角度间隔和允许同步 AVClip 和 SubClip 的同步间隔成为可能。将剪辑信息和 PL 信息归类为“静态脚本”。

[0095] 下面描述“动态脚本”。在这里“动态”指的是重放控制的内容由于用户键事件和重放装置 200 中的状态变化等而变化。使用 BD-ROM,可以使用与 Java 应用程序相同的描述来描述重放控制。也就是说,使用 BD-ROM,Java 应用程序充当动态脚本。

[0096] Java 应用程序

[0097] 下面描述 Java 应用程序。Java 应用程序由装载在虚拟机的堆积区(也称为工作存储器)中的一个或多个 xlet 程序组成。应用程序由装载在工作存储器中的 xlet 程序以及数据构成。Java 应用程序结构如上所述。

[0098] 实际 Java 应用程序是存储在图 2 中的 BDMV 目录下的 BDJA 目录中的 Java 档案文件 (00001.jar)。下面参考图 11 描述 Java 档案文件。

[0099] Java 档案文件

[0100] Java 档案文件(图 2 中的 00001.jar)是一个或多个类文件和数据文件等的集合。图 11A 显示了收集在档案文件中的程序和数据。图 11A 中的数据是由 Java 档案库存储器收集并且排列在框内所示的目录结构中的多个文件。该目录结构由根目录,Java 目录,和图像目录组成,其中 common.pkg 文件,类文件 (aaa.class,bbb.class),和 menu.jpg 文件放置在各自的目录内。Java 档案文件是 Java 档案库存储器将这些文件收集在一起的结果。当从 BD-ROM 读入高速缓存时类文件和数据被扩展,并且在高速缓存中作为存在于目录中的多个文件来对待。Java 档案文件的文件名中的五位数字数值“00001”显示 Java 档案文件的标识符 (ID),并且 BD-J 对象使用该值来参考 Java 档案文件。当将 Java 档案文件读入到高速缓存中时通过参考文件名中的该数值,有可能提取构成任意 Java 应用程序的数据以及程序。

[0101] 图 11A 中的类文件 (aaa.class, bbb.class) 对应于以上的 xlet 程序。使用这些类文件的 xlet 程序(即实例)定义由 Java 平台支持的工作模式 (BD-J) 中的重放过程。xlet 程序是能够使用 Java 多媒体框架 (JMF) 接口的 Java 程序,并且基于根据 JMF 等的键事件执行处理。

[0102] 此外,xlet 程序也可以执行访问网站和下载内容的过程。这允许重放通过混合下载内容和列表创建的原始作品。

[0103] 接着描述 xlet 程序的类文件。图 11B 显示了类文件的内部结构。如图 11B 中所示,该类文件与普通类文件类似,由常数池,接口,和方法 1,2,3…n 组成。类文件中的方法包括由预记载的键事件触发的那些方法 (EventListener) 和用于调用重放装置 200 中的应

用编程接口 (API) 函数的那些方法。通过利用分配给指定方法的局部变量和调用所述方法时出现的自变量描述这些方法中的计算过程等。Java 档案文件如上所述。

[0104] 接着描述扩展名为“bdbj”的文件。该文件 (00001.bdbj) 存储 BD-J 对象。BD-J 对象是信息，该信息通过使 PL 信息中被定义的 AVClip 与应用程序相联系来定义标题。图 12 显示了 BD-J 对象的内部结构。BD-J 对象显示应用程序管理表和 PL 信息参考值。应用程序管理表显示 Java 应用程序，所述 Java 程序的生命周期是由 BD-J 对象通过枚举单个 Java 应用程序的 ID(应用程序 ID) 和属于特殊应用程序的 Java 档案文件的 ID 来定义的标题。换句话说，每个应用程序由一个或多个 Java 档案文件构成。

[0105] PL 信息参考值显示当开始标题时待显示的 PL 信息。

[0106] 扩展名为“bdbj”的文件如上所述。

[0107] 接着描述 INDEX.BDMV 文件。

[0108] INDEX.BDMV 是与整个 BD-ROM 有关的管理信息。该文件包含以下信息，诸如标识电影作品供应商的组织 ID 和分配给由供应商提供的单个 BD-ROM 的光盘 ID。当装载光盘之后首先通过读取 INDEX.BDMV 在重放装置中唯一地识别光盘。INDEX.BDMV 可以附加地包括这样的表，该表将 BD-ROM 中的多个可播放标题映射到定义单个标题的 BD-J 对象。下面描述可记录到 BD-ROM 的标题的类型，所述类型包括 FirstPlayTitle，Top_menuTitle，和标题 #1-#3。

[0109] FirstPlayTitle 负责当装载 BD-ROM 时在做其他事情之前播放 BD-ROM 的动态商标。FirstPlayTitle 因而实现了当装载 BD-ROM 时播放代表电影作品的创作者和 / 或发行人的动态商标。

[0110] Top_menuTitle 由 AVClip 和播放位于 BD-ROM 中的菜单层次的最顶层的菜单的应用程序组成。

[0111] 标题 #1、#2 和 #3 对应于普通电影作品。

[0112] 换句话说，INDEX.BDMV 显示 FirstPlayTitle，Top_menuTitle 和标题 #1-#3 与单个 BD-J 对象的对应关系。

[0113] 图 13 显示了 INDEX.BDMV 的内部结构。该文件显示 FirstPlayTitle 信息，Top_menuTitle 信息，标题 #1 信息，标题 #2 信息，和标题 #3 信息与标题 ID 和定义这些标题的 BD-J 对象的对应关系。定义标题的 BD-J 对象可以使用标题信息进行标识，而用于同步的 PL 信息和应用程序可以从这些 BD-J 对象导出。这完成了对 BD-ROM 的描述。

[0114] BD-ROM 不是关于本发明的重放装置所用的唯一记录介质。

[0115] 在重放期间也使用与重放装置一体化的硬盘 (本地存储器)。下面描述记录在本地存储器中的数据。

[0116] 图 14 显示了本地存储器中的目录结构。在该目录结构中，子目录“organization#1”位于根目录下，在该目录下有子目录“disc#1”和“disc#2”。organization#1 目录被分配给电影作品的特定供应商。disc#1 和 disc#2 目录被分配给由每个供应商提供的不同的 BD-ROM。在这些目录名中利用在各自 BD-ROM 中的 INDEX.BDMV 文件中所示的组织 ID 和光盘 ID。

[0117] 提供与对应于特定供应商的目录中的 BD-ROMs 对应的目录允许独立地存储与单个 BD-ROMs 有关的下载数据。在这些子目录下存储了 PL 信息，剪辑信息，和 AVClip，与存储

在 BD-ROM 上的内容类似。也附加地存在 Java 档案文件，BD-J 对象，合并管理信息文件，和签名信息文件。

[0118] 与仅仅涉及 BD-ROM 上的 AVClip 的 BD-ROM 上的 PL 信息相比，本地存储器中的 PL 信息包括涉及 BD-ROM 上的和本地存储器中的 AVClip 的信息；也就是，作为虚拟包被新添加的 PL 信息，其特定例子是图 14 中的 PL INFO#2。

[0119] 在这里，由四条播放项目信息构成本地存储器中的 PL 信息（播放项目信息 #1-#4）。在头条（播放项目信息 #1）涉及 BD-ROM 上的剪辑信息而其余三条（播放项目信息 #2-#4）涉及本地存储器中的剪辑信息的情况下，该 PL 信息可以从 BD-ROM 上的和本地存储器中的 AVClip 定义单一数据流序列，如图 15 中所示。

[0120] 图 15 显示了由存储在本地存储器中的 PL 信息定义的播放列表重放时间轴的类型。第一层级显示存储在 BD-ROM 上的 AVClip#1 的重放时间轴，而第二层级显示在存储在本地存储器中的 PL 信息中定义的播放列表的重放时间轴，第三、第四和第五层级分别显示存储在本地存储器中的 AVClip#2、#3 和 #4 的重放时间轴。

[0121] 在播放项目信息 #2、#3 和 #4 指定 AVClip#2、#3 和 #4 作为重放间隔的情况下，本地存储器中的 PL 信息能够将 BD-ROM 上的和本地存储器中的 AVClip 定义为单一数据流序列。

[0122] 如上所述，BD-ROM 上的和本地存储器中的 AVClip 可以被定义为单一数据流序列，并且通过合并该数据流序列和 BD-ROM 上的或本地存储器中 的应用程序，可以由 AVClip 和由记录在 BD-ROM 上的或本地存储器中的应用程序构成单一标题。如图 16A 中所示，在 AVClip#1 记录在 BD-ROM 上而 AVClip#2-#4 和 Java 应用程序记录在本地存储器中的情况下，这些 AVClip 和 Java 应用程序可以如图 16B 中所示作为单一标题来对待。

[0123] 接着描述合并管理信息。合并管理信息唯一地显示本地存储器中的 disc#1 和 #2 目录中的构成虚拟包的所有文件，并且被存储在指定文件（在下文被称为“合并管理信息文件”）中。图 17 显示了合并管理信息文件的典型内部结构。合并管理信息文件由本地存储器中构成虚拟包的每个文件的存储位置信息组成。存储位置信息由访问作为虚拟包的各个文件的标识符和指示本地存储器中各个文件的存储位置的文件路径组成。

[0124] 接着描述签名信息文件。签名信息文件在合并管理信息文件上显示供应商的电子签名。通常使用的电子签名通过计算需要防窜改的信息的散列值，并使用某种秘密密钥加密该散列值来获得。

[0125] 这完成了对本地存储器的描述。

[0126] 下面描述关于本发明的重放装置的实施例。图 18 显示了重放装置的硬件配置。重放装置由 BD-ROM 驱动器 1，读缓冲器 2，多路分解器 (Demux)3，视频解码器 4，视频平面 5，图形显示 (P 图形）解码器 6，图形显示 (P 图形）平面 7，合成单元 8，字体发生器 9，交互图形 (I 图形）解码器 10，开关 11，交互图形 (I 图形）平面 12，合成单元 13，颜色查找表 (CLUT) 单元 14，颜色查找表 (CLUT) 单元 15，音频解码器 16，网络设备 17，本地存储器 18，读缓冲器 19，多路分解器 (Demux)20，指令 ROM 21，用户事件 (UE) 处理单元 22，播放器状态寄存器 (PSR) 组 23，中央处理单元 (CPU)24，脚本存储器 25，本地存储器 26，和开关 27。

[0127] 首先描述关于记录在 BD-ROM 100 上的 AVClip 的重放的元件 (BD-ROM 驱动器 1- 音频解码器 16)。

[0128] BD-ROM 驱动器 1 装载 / 退出 BD-ROMs，并且访问 BD-ROM 100。

[0129] 读缓冲器 2 是先进先出 (FIFO) 存储器，其中从 BD-ROM 100 或本地存储器 18 读取的传输流 (TS) 包在先进先出的基础上被存储。

[0130] Demux 3 从读缓冲器 2 读取 TS 包，并且将这些 TS 包转换成打包基本流 (PES) 包。具有由 CPU 24 设置的包标识符 (PID) 的 PES 包，然后被 输出到视频解码器 4，P 图形解码器 6，I 图形解码器 10，和音频解码器 16 中的一个。

[0131] 视频解码器 4 解码从 Demux3 输出的 PES 包以获得未压缩格式的图片，并且将这些图片写入视频平面 5。

[0132] 视频平面 5 用于存储未压缩图片。平面是重放装置中用于存储一屏像素数据的存储区。视频平面 5 具有 1920×1080 的分辨率，其中存储的图片数据由 16 位 YUV 表达的像素数据构成。在视频平面 5 中，每帧视频流中的重放视频可以被缩放。缩放包括将每帧的重放视频变化为整个视频平面 5 的 1/4（四分之一）或 1/1（全比例尺）。在 BD-J 模式中根据来自 CPU 24 的指令执行缩放，从而允许屏幕产生，由此视频流的重放图像被转移到屏幕的角上或投影到整个屏幕上。

[0133] P 图形解码器 6 解码从 BD-ROM 读取的 P 图形流，并且将未压缩图形写入 P 图形平面 7。字幕由于图形流被解码出现在屏幕上。

[0134] P 图形平面 7 是带有用于一屏数据的存储空间的存储器，其能够存储一屏未压缩图形。该平面具有 1920×1080 的分辨率，其中 P 图形平面 7 中的未压缩图形的像素由 8 位索引颜色表达。存储在 P 图形平面 7 中的未压缩图形通过使用 CLUT 转换索引颜色被提交以供显示。

[0135] 合成单元 8 合成存储在视频平面 5 中的未压缩图片数据和 P 图形平面 7 的存储内容。

[0136] 字体发生器 9 使用字符字体来扩展包括在位图中的 TextST 流中的文本代码，并且将扩展代码写入 P 图形平面 7。

[0137] I 图形解码器 10 在 DVD-like 模式中解码从 BD-ROM 或本地存储器 18 读取的 I 图形流，并且将未压缩图形写入 I 图形平面 12。

[0138] 开关 11 选择性地将字体发生器 9 生成的字体序列和由 P 图形解码器 6 解码产生的图形中的一个写入到 P 图形平面 7。

[0139] 用由 I 图形解码器 10 解码产生的未压缩图形写入 I 图形平面 12。使用 α RGB 全彩色在 BD-J 模式中将应用程序绘制的字符和图形写入到 I 图形平面 12 中。

[0140] 合成单元 13 合成 I 图形平面 12 的存储内容和从合成单元 8 输出的合成图像（即合成未压缩图片数据和 P 图形平面 7 的存储内容）。该合成允许由 应用程序写入 I 图形解码器 10 的字符和图形重叠在未压缩图片数据上并且被显示。

[0141] CLUT 单元 14 将存储在视频平面 5 中的未压缩图形中的索引颜色转换成 Y/Cr/Cb。

[0142] 当在 DVD-like 模式（即用于播放象传统 DVD 这样的数字流的模式）中工作时 CLUT 单元 15 将存储在 I 图形平面 15 中的未压缩图形中的索引颜色转换成 Y/Cr/Cb。当在 BD-J 模式（即用于与 Java 应用程序同步地播放数字流的模式）中工作时，CLUT 单元 15 将 α RGB 全彩色转换成 Y/Cr/Cb。需要注意的是 Java 应用程序可以绑定或不绑定到标题，以及绑定或不绑定到光盘。

[0143] 音频解码器 16 解码从 Demux 3 输出的 PES 包并且输出未压缩音频数据。

[0144] 关于 AVClip 重放的元件如上所述。下面描述关于在 BD-J 模式中工作的元件（网络设备 17-Demux20）。

[0145] 网络设备 17 实现重放装置中的通信功能。在应用程序在 BD-J 模式中指定 URL 的情况下，网络设备 17 建立与 URL 所指示的网站的传输控制协议 (TCP) 或文件传送协议 (FTP) 连接等。由于连接被建立 Java 应用程序从网站被下载。

[0146] 本地存储器 18 是硬盘，其用于存储通过网络设备 17 建立的连接从网站下载的内容，从除 BD-ROM 之外的通信和记录介质提供的内容，以及元数据。元数据是用于绑定和管理本地存储器 18 中的下载内容的信息。通过访问本地存储器 18，BD-J 模式中的应用程序可以使用下载内容执行许多处理。本地存储器 18 也保存合并管理信息文件。

[0147] 读缓冲器 19 是 FIFO 存储器，其在 SubClip 被包括在存储于 BD-ROM100 上或本地存储器 18 中的下载内容中的情况下基于先入先存储构成 SubClip 的 TS 包。

[0148] Demux 20 从读缓冲器 19 读取 TS 包，并且将读取的 TS 包转换成 PES 包。具有特定 PID 的 PES 包然后输出到 P 图形解码器 6，字体发生器 9，和音频解码器 16。

[0149] 以上元件 17-20 允许由 Java 应用程序通过网络下载的内容以类似于记 录在 BD-ROM 上的内容的方式被播放。下面描述用于实现重放装置中的集中控制的元件（指令 ROM 21- 开关 27）。

[0150] 指令 ROM 21 存储软件，该软件定义与重放装置有关的控制。

[0151] 响应重放装置的遥控器或前面板的键操作，UE 处理单元 22 将用于执行这些操作的用户事件输出到 CPU 24。

[0152] PSR 组 23 是在重放装置内部的一组寄存器，由 64 个播放器状态寄存器 (PSR) 和 4096 个通用寄存器 (GPR) 组成。PSR 4-8 用于表示当前重放点。

[0153] PSR 4 由于被设置到 1-100 的值指示当前重放点的标题。将 PSR 4 设置为“0”表示当前重放点是菜单顶部 (top menu)。

[0154] PSR 5 由于被设置到 1-999 的值指示当前重放点的章节编号。将 PSR 5 设置为“0xFFFF”指示在重放装置中的章节编号为零。

[0155] PSR 6 由于被设置到 0-999 的值指示当前重放点所属的 PL（当前 PL）的编号。

[0156] PSR 7 由于被设置到 0-255 的值指示当前重放点所属的播放项目（当前播放项目）的编号。

[0157] PSR 8 由于被设置到 0-0xFFFFFFFF 的值指示使用 45KHz 的时间精度的当前重放点（当前显示时间或“PTM”）。PSR 4-8 允许在图 23A 中所示的整个 BD-ROM 的时间轴上标识当前重放点。

[0158] CPU 24 运行存储在指令 ROM 21 中的软件以执行与整个重放装置有关的控制。这些控制动态地变化，其取决于从 UE 处理单元 22 输出的用户事件和 PSR 组 23 中的 PSR 值。

[0159] 脚本存储器 25 用于存储当前 PL 信息和当前剪辑信息。当前 PL 信息是记录在 BD-ROM 上的当前被处理的一条 PL 信息。当前剪辑信息是记录在 BD-ROM 上的当前被处理的一条剪辑信息。

[0160] 假设从 BD-ROM 低速读取，本地存储器 26 是临时存储 BD-ROM 的记录内容的高速缓冲存储器。本地存储器 26 的提供允许 BD-J 模式中的应用程序高效地运行。

[0161] 开关 27 选择性地将从 BD-ROM 和本地存储器 18 读取的数据输送到读缓冲器 2，读缓冲器 19，脚本存储器 25 和本地存储器 26 中的一个。

[0162] 关于本实施例的重放装置的硬件配置如上所述。下面描述关于本实施例的重放装置中的软件结构。

[0163] 图 19 描绘了由硬件和存储在指令 ROM 21 上的软件组成的以层状结构重布置的元件。如图 19 中所示，重放装置的层状结构由第一层 (BD 层)，第二层 (BD 播放器模型)，和第三层 (应用程序运行时间环境) 组成。

[0164] 图 18 中所示的重放装置的硬件配置属于第一层。在该硬件配置中，在图 19 中的第一层的“BD 播放器”包括由视频解码器 4，P 图形解码器 6，I 图形解码器 10 和音频解码器 16 组成的“解码器”，由视频平面 5，P 图形平面 7 和 I 图形平面 12 组成的“平面”，BD-ROM 100 和相关文件系统，本地存储器 18 和相关文件系统，以及网络设备 17。

[0165] 在第二层的“BD 播放器模型”由用于显示引擎 31 和虚拟文件系统单元 38 的下层和用于重放控制引擎 32 的上层组成，并且提供与更高级别有关的 API 函数。

[0166] 图 18 中所示的 PSR 组 23 和脚本存储器 25 存在于重放控制引擎 32 内部。

[0167] 在第三层的“应用程序运行时间环境”由包括模块管理器 33 的层组成，所述模块管理器 33 堆叠在包括 DVD-like 模块 29a 和 Java 平台 29b 的层上。

[0168] 下面描述该软件结构中的元件。

[0169] DVD-like 模块 29a，Java 平台 29b

[0170] DVD-like 模块 29a 解码导航命令，并且基于解码结果执行与重放控制引擎 32 有关的函数调用。

[0171] Java 平台 29b 是所谓的具有层次结构的 Java 平台，所述层次结构由 Java 虚拟机 30 和 Java 虚拟机运行的中间软件 (未示出) 组成。

[0172] Java 虚拟机 30

[0173] Java 虚拟机 30 将构成应用程序的 xlet 程序装载到工作存储器中，解码 xlet 程序，并且根据解码结果在下层上执行控制。为了执行这些控制，Java 虚拟机 30 向中间软件发布方法，使中间软件用对应于 BD 重放装置的函数调用替换所述方法，并且向重放控制引擎 32 发布所述函数调用。

[0174] Java 虚拟机 30 的内部结构

[0175] 下面描述 Java 虚拟机 30 的内部结构。图 20 显示了 Java 虚拟机 30 的 内部结构。如图 20 中所示，Java 虚拟机 30 由 CPU 24，用户类装载器 52，方法区 53，工作存储器 54，线程 55a，55b，…55n，和 Java 栈 56a，56b，…56n 构成。

[0176] 用户类装载器 52 从本地存储器 26 等读取 BDJA 目录中的 Java 档案文件中的类文件，并且将读取的类文件存储在方法区 53 中。由用户类装载器 52 读取类文件由于模块管理器 33 将指定文件路径的读指令发送到用户类装载器 52 被执行。如果文件路径指示本地存储器 26，用户类装载器 52 将构成应用程序的 Java 档案文件中的类文件从本地存储器 26 读入工作存储器 54。如果文件路径指示文件系统中的目录，用户类装载器 52 将构成应用程序的 Java 档案文件中的类文件从 BD-ROM 或本地存储器 18 读入工作存储器 54。

[0177] 方法区 53 存储由用户类装载器 52 从本地存储器 26 读取的类文件。

[0178] 工作存储器 54 是所谓的存储各种类文件的实例的堆积区。工作存储器 54 存储对

应于常驻应用程序的实例和读入到方法区 53 的类文件。一个实例是构成应用程序的 xlet 程序，通过将 xlet 程序放置到工作存储器 54 中使所述应用程序可执行。

[0179] 线程 55a，55b，…55n 是用于执行存储在工作存储器 54 中的方法的逻辑执行实体。它们使用局部变量和存储在操作数栈中作为操作数的自变量执行计算，并且将计算结果存储在局部变量或操作数栈中。图 20 中的箭头 ky1，ky2，和 kyn 象征性地指示方法从工作存储器 54 提供给线程 55a，55b，…55n。尽管 CPU 是唯一的物理执行实体，在 Java 虚拟机 30 中可以有高达 64 个逻辑执行实体或线程。线程可以被新创建并且现有线程可以在该数值限度内被删除，而且当 Java 虚拟机 30 工作时工作线程的数目可以变化。能够适当地增加线程的数目也使得有可能使用每个实例的多个线程并行地运行实例，并且由此加速实例的执行。

[0180] Java 栈 56a，56b，…56n 与线程 55a，55b，…55n 一比一地存在，并且每个具有程序计数器（图 20 中的“PC”）和一个或多个帧。程序计数器显示当前正在执行实例的哪个部分。帧是分配给方法的每次调用的栈型区域，并且由用于存储与调用同时出现的自变量的操作数栈和被所调用的方法使用的局部变量栈（图 20 中的“局部变量”）组成。由于无论何时进行调用 帧都堆叠在 Java 栈 56a，56b，…56n 上，因此递归地调用自身的方法的帧也一个堆叠在另一个上面。

[0181] Java 虚拟机的内部结构如上所述。具有以上结构的 Java 虚拟机充当事件驱动执行实体。这完成了对 Java 虚拟机的描述。

[0182] 显示引擎 31

[0183] 显示引擎 31 执行 AV 重放功能。重放装置的 AV 重放功能是从 DVD 播放器和 CD 播放器继承的传统功能组，包括播放，停止，暂停开始 (PAUSE ON)，暂停结束 (PAUSE OFF)，静音 (STILL OFF)，快进播放 (x2，x4 等)，快退播放 (x2，x4 等)，音频变化，字幕变化，和角度变化。为了实现这些 AV 重放功能，显示引擎 31 控制视频解码器 4，P 图形解码器 6，I 图形解码器 10 和音频解码器 16 以解码读入到读缓冲器 2 的 AVClip 的一部分，所述部分对应于预期时间。通过将 PSR 8 所指示的地方（当前 PTM）解码为预期时间，可以致使 AVClip 中的任意点可播放。

[0184] 重放控制引擎 32

[0185] 重放控制引擎 32 执行各种功能，包括 (i) 对播放列表的重放控制和 (ii) 采集设置 PSR 组 23 的状态。重放控制功能包括根据当前 PL 信息和剪辑信息使显示引擎 31 执行以上 AV 重放功能中的播放和停止。功能 (i) 和 (ii) 根据从 DVD-like 模块 29a 和 Java 平台 29b 的函数调用被执行。

[0186] 接着描述重放控制引擎 32 执行的处理与 Java 虚拟机执行的处理的同步。当调用函数时重放控制引擎 32 执行基于 PL 信息的处理。在用于重放的 AVClip 的持续时间执行该处理，无论重放时间是 15 分钟还是 30 分钟。这里的问题是在 Java 虚拟机 30 返回成功响应的时间和重放控制引擎 32 实际结束处理的时间之间存在延时 (time lag)。作为事件驱动执行实体的 Java 虚拟机 30 在调用之后立即返回指示重放是否成功的响应，而重放控制引擎 32 在 15 或 30 分钟重放持续时间过去之后结束 AVClip 和播放项目的重放。因而，成功响应返回到应用程序的时间不能用作衡量 15 或 30 分钟以后处理结束的基础。当在 PL 重放期间执行快进或倒带时衡量处理结束变得更加困难，这是由于 15 或 30 分钟的重放时

间发生变化。鉴于此，当各个播放项目或 AVClip 的重放结束时，重放控制引擎 32 将指示播放项目和 AVClip 重放结束的事件输出到应用程序。这种输出使得应用程序知道重放控制引 擎 32 在哪个点结束播放项目或 AVClip 重放。

[0187] 模块管理器 33

[0188] 模块管理器 33 读取 INDEX.BDMV 并且选择 INDEX.BDMV 中的多条标题信息中的一条作为当前标题信息。模块管理器 33 读取当前标题信息所指示的 BD-J 对象，并且基于 BD-J 对象中所描述的 PL 信息控制重放控制引擎 32 执行重放控制。模块管理器 33 也控制 Java 虚拟机 30 以读取和执行 BD-J 对象中所描述的 Java 档案文件。

[0189] 如果基于 PL 信息的数字流的重放和应用程序的执行结束，或者如果用户调用菜单，模块管理器 33 读取定义另一标题的标题信息，并且选择该条标题信息作为当前标题信息。根据数字流重放或用户菜单调用选择另一条标题信息作为当前标题信息的过程被称为“标题变化”。

[0190] 图 21 中所示的状态转变可以通过重复地执行标题变化实现。图 21 中的椭圆窗口表示标题。

[0191] 标题包括当 BD-ROM 首次被装载时用于重放的“FirstPlayTitle”，构成顶部菜单的“Top_menuTitle”，和其他普通标题。图 21 中的箭头 jh1，jh2，jh3，jh4，jh5，jh6，jh7 和 jh8 象征性地指示标题之间的转移。

[0192] 图 21 中所示的状态转变包括当 BD-ROM 被装载时播放 FirstPlayTitle，然后转移到 Top_menuTitle 并且等待从顶部菜单选择。

[0193] 当用户从菜单选择时，在再次返回到 Top_menuTitle 之前根据选择播放各个标题。直到光盘被退出无限地重复该处理的过程是光盘内容所独有的状态转变。该状态转变在上述的模块管理器 33 的控制下实现。

[0194] 这完成了对 Java 虚拟机 30，显示引擎 31，重放控制引擎 32，和模块管理器 33 的描述。由 Java 虚拟机 30 对重放控制引擎 32 的控制通过虚拟包被执行。为了通过虚拟包实现对重放控制引擎 32 的控制，重放装置包括网络管理模块 37 和虚拟文件系统单元 38。接着描述这些元件。

[0195] 网络管理模块 37

[0196] 网络管理模块 37 根据从应用程序的方法调用从电影作品的供应商所管理的网站下载产生虚拟包所需的数据。该数据包括替换或添加到合并管理信息文件，签名信息文件，和 BD-ROM 上的文件的文件 (PL 信息，剪辑信息，AVClip，Java 档案文件，等等)。当工作存储器 54 中的应用程序提 出下载请求时，网络管理模块 37 通过网络下载产生虚拟包所需的数据，并且将下载的数据写入本地存储器 18。

[0197] 虚拟文件系统单元 38

[0198] 虚拟文件系统单元 38 是属于图 19 中的第二层的元件，其根据从应用程序的方法调用产生虚拟包。虚拟包的产生包括管理构成虚拟包的 AVClip 的状态的处理和生成虚拟包信息的处理。

[0199] 虚拟包信息

[0200] 虚拟包信息扩展 BD-ROM 上的卷管理信息。这里涉及的卷管理信息定义存在于记录介质上的目录结构，并且由与目录有关的目录管理信息和与文件有关的文件管理信息组

成。

[0201] 虚拟包信息通过将新文件管理信息添加到显示目录结构的卷管理信息来扩展 BD-ROM 上的目录结构。添加到卷管理信息的文件管理信息涉及存在于本地存储器 18 中的 PL 信息，剪辑信息，AVClip 和 Java 档案文件。产生该文件管理信息已被添加到其中的虚拟包信息和将该虚拟包信息提供给重放控制引擎 32 的过程允许重放控制引擎识别如存在于 BD-ROM 上的存储在本地存储器 18 中的 PL 信息，剪辑信息，AVClip 和 Java 档案文件。图 22 显示了由虚拟文件系统单元 38 典型地产生虚拟包信息。在图 22 的左上方是 BD-ROM 上的目录结构，这与图 2 相同。在左下方是本地存储器 18 中的目录结构，这与图 14 相同。与本地存储器 18 中的 PL 信息，剪辑信息，AVClip 和 Java 档案文件有关的文件管理信息被添加到 BD-ROM 上的卷管理信息。

[0202] 具体而言：

[0203] i) 与本地存储器 18 中的播放列表信息 #2(00002.mpls) 有关的文件管理信息被添加到 PLAYLIST 目录中的目录管理信息；

[0204] ii) 与本地存储器 18 中的剪辑信息 #2、#3 和 #4(00002.clip，00003.clip，00004.clip) 有关的文件管理信息被添加到 CLIPINF 目录中的目录管理信息；

[0205] iii) 与本地存储器 18 中的 AVClip#2、#3 和 #4(00002.m2ts，00003.m2ts，00004.m2ts) 有关的文件管理信息被添加到 STREAM 目录中的目录管理信息；

[0206] iv) 与本地存储器 18 中的 Java 档案文件“00002.jar”有关的文件管理 信息被添加到 BDJA 目录中的目录管理信息；

[0207] 由此获得虚拟包信息。换句话说，虚拟包信息是以上面的方式已被添加的卷管理信息。

[0208] 然后将该虚拟包信息提供给重放控制引擎 32，重放控制引擎由此能够在与 BD-ROM 上的 PL 信息，剪辑信息，AVClip，和 Java 档案文件同等的基础上处理本地存储器 18 中的 PL 信息，剪辑信息，AVClip，和 Java 档案文件。虚拟包信息的生成如上所述。

[0209] 下面描述虚拟包信息更新的定时。

[0210] 假设当按照图 21 中的箭头 jh1，jh2，jh3，jh4 等所指示的参考标记的数值顺序执行转移之后 BD-ROM 被退出。这允许从 BD-ROM 的装载到退出的连续时间间隙被看作单一时间轴。该时间轴被作为整个光盘的时间轴。图 23A 显示了整个光盘的时间轴，而图 23B 显示了该时间轴的结构。如图 23B 中所示，整个光盘的时间轴由播放 FirstPlayTitle 的间隔，播放 Top_menuTitle 的间隔，和播放普通标题（标题 #1 等）的间隔组成。关于定义这些标题的重放间隔的方式，由于每个标题仅仅由一个 BD-J 对象构成，因此任何给定的 BD-J 对象有效的间隔可以被当作标题的重放间隔。这些重放间隔之间的空隙，或者从一个标题变化到另一标题的些微间隔（即“标题变化”）是虚拟包信息更新的时间。

[0211] 接着使用图 24 描述由 Java 应用程序执行的下载新合并管理和签名信息文件以及内容文件的过程。

[0212] Java 应用程序首先将当前合并管理信息文件发送到服务器（步骤 S11），由此请求下载，并且判断是否已接收来自服务器的数据（步骤 S12）。当数据被下载时，Java 应用程序在相应光盘目录中创建新目录，并且将下载的合并管理信息文件和签名信息文件写入新目录（步骤 S13）。需要注意的是如果下载的合并管理和签名信息文件的文件名不与光盘目

录中的现有合并管理和签名信息文件一致，下载文件可以直接放置在现存目录（光盘 #1 文件）之下而不创建新目录。将下载的 AVClip，剪辑信息，PL 信息，和 Java 档案文件写入相应目录（步骤 S14）。Java 应用程序然后使用新合并管理和签名信息文件的文件路径作为自变量调用更新请求方法（步骤 S15）。Java 应用程序判断返回值是否为 false（步骤 S16），并且如果为 false 则终止处理。 如果返回值是不是 false，则 Java 应用程序使用更新的虚拟包信息执行处理（步骤 S17）。

[0213] 需要注意的是尽管根据当请求下载时 Java 应用程序将当前合并管理信息文件发送到服务器描述了以上处理，Java 应用程序可以仅仅发送合并管理信息文件的 ID。

[0214] 接着使用图 25 描述一旦接收更新请求由虚拟文件系统单元 38 执行的更新“准备”过程。

[0215] 虚拟文件系统单元 38 首先使用当调用方法时充当自变量的文件路径读取新合并管理和签名信息文件（步骤 S21），并且验证签名以便检验新合并管理信息文件是否已被窜改（步骤 S22）。如果签名不能被验证则执行异常终止。如果签名被验证，则虚拟文件系统单元 38 检验调用应用程序的权限（步骤 S23）。如果调用应用程序未被授权则执行异常终止。如果调用应用程序被授权，则虚拟文件系统单元 38 判断新合并管理信息文件所指定的文件是否实际存在于本地存储器中（步骤 S24）。如果这些文件并不存在则执行异常终止。如果这些文件存在，则虚拟文件系统单元 38 将新合并管理和签名信息文件以及从新合并信息文件引用的本地存储器中的所有文件变为只读（步骤 S25）。

[0216] 图 26 是由虚拟文件系统单元 38 执行的虚拟包“更新”处理的流程图。虚拟文件系统单元 38 首先标识对应于被装载的 BD-ROM 的光盘目录，并且用 Java 应用程序调用更新请求方法时、充当自变量的文件路径所指定的新合并管理和签名信息文件替换光盘目录中的合并管理和签名信息文件（步骤 S31）。虚拟文件系统单元 38 然后将本地存储器 18 中的合并管理信息文件所指定的 PL 信息的文件管理信息添加到 PLAYLIST 目录中的目录管理信息中（步骤 S32），并且执行步骤 S33 到 S37 的循环。该循环包括对于存在于本地存储器 18 中的每一条剪辑信息重复步骤 S34 到 S36。在这里，用于循环处理的一条剪辑信息假定为剪辑信息 x。虚拟文件系统单元 38 标识对应于剪辑信息 x 的 AVClip（步骤 S34），将本地存储器 18 中的合并管理信息文件所指定的剪辑信息 x 的文件管理信息添加到 CLIPINF 目录中的目录管理信息中（步骤 S35），并且将本地存储器 18 中的合并管理信息文件所指定的 AVClip x 的文件管理信息添加到 STREAM 目录中的目录管理信息中（步骤 S36）。通过对于本地存储器 18 中的所有剪辑信息和 AVClip 重复以上处理，与剪辑信息和 AVClip 有关的文件管理信息被添加到卷管理信息。由此获得的卷管理信息是虚拟包信息。虚拟文件系统单元 38 将该虚拟包信息提供给进行虚拟包调用的应用程序（步骤 S38），并且结束处理。

[0217] 图 27 是由模块管理器 33 执行的处理的流程图。

[0218] 模块管理器 33 首先选择 FirstPlayTitle 作为当前标题（步骤 S41），指定对应于当前标题的 BD-J 对象作为当前 BD-J 对象（步骤 S42），并且使重放控制引擎 32 基于当前 BD-J 对象中所描述的 PL 信息执行 PL 重放（步骤 S43）。模块管理器 33 然后使 Java 平台 29b 运行当前 BD-J 对象的应用程序管理表中其生命周期是当前标题的 Java 应用程序（步骤 S44），并且使 Java 平台 29b 终止其生命周期不是当前标题的 Java 应用程序（步骤 S45）。模块管理器 33 然后判断基于当前 PL 信息的 PL 重放是否已经完成（步骤 S46），并且如果

完成，模块管理器 33 则标识下一个标题（步骤 S47），并且选择该标题作为当前标题（步骤 S48）。如果当前 PL 重放还未完成，则模块管理器 33 判断是否已发生标题调用（步骤 S49），并且如果是这样的话则移动到步骤 S47。如果未发生标题调用，则模块管理器 33 判断是否已发生标题跳转（步骤 S50），并且如果是这样的话则移动到步骤 S47。如果未发生标题跳转，则模块管理器 33 判断当前标题的主应用程序是否已结束（步骤 S51）并且如果是这样的话则移动到步骤 S47。如果主应用程序还未结束，则模块管理器 33 返回到步骤 S46。

[0219] 图 28 是由重放控制引擎 32 执行的重放处理的流程图。在将当前 PL 信息中的第一条播放项目信息设置为播放项目 i 之后重放控制引擎 32 执行步骤 S62 到 S68 的循环（步骤 S61）。该循环中的控制变量是变量 i。在执行步骤 S62 到 S66 之后在步骤 S68 重放控制引擎 32 使控制变量 i 增加“1”直到变量 i 超过播放项目的数量（步骤 S67）。

[0220] 接着描述步骤 S62 到 S66。重放控制引擎 32 将播放项目 i 中的 Clip_information_file_name 中所描述的 AVClip 设置为 AVClipj（步骤 S62），并且指示驱动设备和解码器播放从 PlayItem.In_time 到 PlayItem.Out_time 的 AVClip j（步骤 S63）。重放控制引擎 32 判断在 Sync_PlayItem_Id 中是否存在指定播放项目 i 的子播放项目 k（步骤 S64），并且如果该子播放项目并不 存在则直接移动到步骤 S67。如果子播放项目 k 存在，则重放控制引擎 32 将子播放项目 k 的 Clip_information_file_name 所描述的 AVClip 设置为 AVClip h（步骤 S65），指示驱动设备和解码器重放从 Sync_Start_PTS_of_PlayItem 到 Out_time 的 AVClip h（步骤 S66），并且移动到步骤 S67。

[0221] 由于对于构成 PL 信息的所有播放项目信息重复该处理，PL 信息所定义的数据流序列被播放。

[0222] 图 29 显示了在标题变化期间怎样更新虚拟包信息。

[0223] 图 29 中的第一层级显示时间轴上的标题重放间隔，第二层级显示其生命周期是标题 #1 的 Java 应用程序，第三层级显示数字流，第四层级显示虚拟文件系统单元 38 的状态。

[0224] 一旦接受来自 Java 应用程序的更新请求，虚拟文件系统单元 38 被放置在“准备”状态，并且执行图 25 中所示的处理。

[0225] 在执行该处理之后，虚拟文件系统单元 38 在“准备好”状态等待标题变化。当发生标题变化时，虚拟文件系统单元 38 被放置在“更新”状态，并且在回复到“稳定”状态之前执行图 26 中所示的处理以更新虚拟包。如果在虚拟文件系统单元 38 回复到“稳定”状态之后，再次从 Top_menuTitle 选择标题 #1，则使用更新的虚拟包播放标题 #1。

[0226] 在这里，当模块管理器 33 例如由于数据流序列的重放结束或由用户调用菜单而选择不同标题作为当前标题时，发生标题变化。

[0227] 使用图 30 到 33 示意性地描述以上处理。

[0228] 图 30 显示了将当前合并管理信息文件发送到服务器的 Java 应用程序。ROOT 下所示的文件位于本地存储器中，而 BDMV 下的文件位于虚拟包中。

[0229] 图 31 显示了下载内容文件、新合并管理信息文件和新签名信息文件的 Java 应用程序。“00012.clpi”和“00012.m2ts”是下载的内容文件，而存储在 newMF 目录中的合并管理和签名信息文件已最新被下载。

[0230] 图 32 显示了请求虚拟文件系统单元 38 将现有合并管理和签名信息文件更新为最

新下载的文件的 Java 应用程序。通过使用文件路径来指定新合并管理和签名信息文件提出该更新请求。

[0231] 图 33 显示了合并管理和签名信息文件的替换以及内容文件的映射。在 图 33 的左侧显示了更新期间旧合并管理和签名信息文件的替换。在图 33 的右侧显示了标题变化后内容文件的映射。

[0232] 需要注意的是从旧合并管理信息而不是从新合并管理信息引用的文件的只读属性被移除，从而使这些文件可由 Java 应用程序写入。

[0233] 合并管理信息文件包括指示添加到本地存储器的内容的位置的信息。指示附加内容位置的信息包括内容 ID，内容存储于其中的目录的目录路径，或单个内容文件的文件路径。当将这些文件映射到虚拟包时可以在合并管理信息文件中描述文件名映射信息，从而允许在虚拟包中的不同文件名下访问这些文件。在这里，文件名映射信息用本地存储器中的文件名（包括文件路径）映射虚拟包中的文件名（包括文件路径）。

[0234] 在该情况下，作为虚拟包介质由虚拟文件系统单元 38 产生虚拟包，所述虚拟包介质由这样的文件构成，文件名映射信息中所描述的虚拟包中的所述文件的文件名已被添加到 BD-ROM 上的文件结构中。由 Java 应用程序访问的虚拟包中的文件被指定为虚拟包中的文件而不是 BD-ROM 上或本地存储器中的文件。当 Java 应用程序请求访问虚拟包中的文件时，虚拟文件系统单元 38 基于文件名映射信息将访问目的地切换到本地存储器或 BD-ROM。如果在文件名映射信息中描述了所期望的文件，访问目的地变为本地存储器中的相应文件。如果未在文件名映射信息中描述所期望的文件，访问目的地变为 BD-ROM 上的相应文件。

[0235] 换句话说，Java 应用程序的创作者不需要知道单个文件存储于其上的介质 (BD-ROM 或本地存储器)，因为虚拟文件系统单元 38 将虚拟包中由 Java 应用程序指定的文件的访问目的地切换到实际存储所述文件的介质，由此减轻了程序创作的负担。

[0236] 根据本实施例，虚拟包在标题变化期间被更新，这意味着重放对象的替换将不会导致重放装置的异常操作。

[0237] 实施例 2

[0238] 本实施例涉及当执行标题调用时的改进。标题调用导致在首先暂停当前标题之后播放被调用标题，然后在被调用标题的重放结束之后继续原始标题。由于标题调用以继续重放为前提，因此当调用标题时重放控制引擎 32 将存储在 PSR 中的用于重放控制的系统参数保存到备份 PSR 中，并且在 被调用标题的重放结束之后将被保存参数恢复到 PSR 中。

[0239] 下面是存储在 PSR 中的系统参数的列表。PSR 0 到 PSR 12 存储显示重放状态的系统参数，PSR 13 到 PSR 19 存储由播放器设置为优选的系统参数，PSR 20 到 PSR 32 是备份 PSR。

[0240] PSR0 :I 图形流编号

[0241] PSR1 :音频流编号

[0242] PSR2 :P 图形流 /TextST 流编号

[0243] PSR3 :角度编号

[0244] PSR4 :当前标题编号

[0245] PSR5 :当前章节编号

[0246] PSR6 :当前播放列表 ID

[0247] PSR7 :当前播放项目 ID

[0248] PSR8 :重放时间信息

[0249] PSR9 :导航定时器

[0250] PSR10 :选择键信息

[0251] PSR11 :I 图形流中的当前页 ID

[0252] PSR12 :P 图形流和 TextST 流中的用户样式 ID

[0253] PSR13 :视听年龄等级 (parental level)

[0254] PSR14 :字幕支持信息

[0255] PSR15 :播放器设定值 (音频)

[0256] PSR16 :音频流的语言代码

[0257] PSR17 :P 图形流和 TextST 流的语言代码

[0258] PSR18 :菜单的语言代码

[0259] PSR19 :播放器的版本信息

[0260] PSR20 :PSR0 的备份

[0261] PSR21 :PSR1 的备份

[0262] PSR22 :PSR2 的备份

[0263] PSR23 :PSR3 的备份

[0264] PSR24 :PSR4 的备份

[0265] PSR25 :PSR5 的备份

[0266] PSR26 :PSR6 的备份

[0267] PSR27 :PSR7 的备份

[0268] PSR28 :PSR8 的备份

[0269] PSR29 :PSR9 的备份

[0270] PSR30 :PSR10 的备份

[0271] PSR31 :PSR11 的备份

[0272] PSR32 :PSR12 的备份

[0273] 在标题调用期间更新虚拟包信息导致调用前后虚拟包信息的差异。

[0274] 由于当恢复原始标题时虚拟包信息将变化，因此如果重放控制引擎 32 试图使用备份值播放原始标题则会出现错误。该问题通过当 Java 应用程序请求更新时清除备份 PSR 得以避免。然而，假定变化没有影响取决于合并管理信息文件的内容，关于是否清除系统参数值的决定可以留待 Java 应用程序处理。

[0275] 图 34A 是当标题调用之后暂停当前标题的重放时由重放控制引擎 32 执行的处理的流程图。图 34B 是当被调用标题的重放结束之后继续重放原始标题时由重放控制引擎 32 执行的处理的流程图。

[0276] 当暂停当前标题重放时，重放控制引擎 32 将 PSR 0-12 保存到 PSR 20-32(步骤 S71)。

[0277] 当被调用标题的重放结束之后继续原始标题重放时，重放控制引擎 32 首先判断虚拟包是否已被更新 (步骤 S81)。如果未更新则将 PSR 20-32 恢复到 PSR 0-12(步骤 S83)，

如果虚拟包信息已被更新则在执行步骤 S83 之前初始化 PSR 20-32(步骤 S82)。

[0278] 根据本实施例,在标题调用期间当虚拟包信息已被更新时初始化备份 PSR,由此消除当继续原始标题重放时发生重放错误的危险。因而使重放控制引擎 32 稳定工作。

[0279] 需要注意的是当虚拟包信息被更新时,PSR 中的系统参数值可以被强制清除,而不是将该决定留待 Java 应用程序处理。

[0280] 实施例 3

[0281] 本实施例涉及一种方法,该方法用于管理合并管理信息的版本和从重放装置中的常驻应用程序指定用于合并的附加内容。图 35 显示了关于本实 施例的合并管理信息文件的典型内容。在实施例 1 中,通过改写旧合并管理信息更新合并管理信息文件(或更确切地说存储在其中的合并管理信息),从而导致旧信息被擦除。在本实施例中,新合并管理信息不断地被添加到文件中,而且即使对于相同的光盘 ID 也不改写旧信息。因而,如果虚拟文件系统单元 38 取消产生虚拟包并且回复到原始 BD-ROM,则反映该状态的信息被保留在合并管理信息文件中。在该情况下,合并管理信息文件的合并目标目录中的相应单元被留下空白或者记下指示原始 BD-ROM 的字符串。

[0282] 通过执行更新时不改写旧合并管理信息使将先前的合并管理信息(的历史)保留在合并管理信息文件中,然后如果用户想要虚拟包的旧版本,可以参考先前的合并管理信息产生旧版本。而且,先前由用户产生的虚拟包可以参考合并管理信息文件(或者更确切地说存储在其中的旧合并管理信息)不仅从 Java 应用程序而且从重放装置中的常驻应用程序产生。

[0283] 由常驻应用程序使用先前的合并管理信息的另一例子包括显示附加内容列表,从而用户可以从常驻应用程序删除不想要的附加内容。由于合并管理信息文件可以用于区分存储附加内容的目录,因此也可以从除了存储附加内容的 Java 应用程序之外的应用程序(即常驻应用程序)检索和删除附加内容。

[0284] 图 36 显示了关于本实施例的请求虚拟包更新的 Java 应用程序。与实施例 1 的差异在于这样的事实,即在不改写旧信息的情况下不断地添加合并管理信息,即使用于目标光盘 ID 的合并管理信息已经存在。当请求虚拟包更新时 Java 应用程序允许通过追加日期信息标识合并管理信息文件中最新的一条合并管理信息。日期信息不限于日期,也可以简单地为连续编号。

[0285] 图 37 显示了由使用合并管理信息文件的常驻应用程序向用户显示的典型附加内容列表。在这里,被显示的附加内容列表基于图 35 中所示的合并管理信息文件。理想的是显示使用户掌握附加内容涉及的信息。图 37 中的附加内容被显示为内容名。尽管仅仅日期信息被添加到图 36 中的合并管理信息,但是也可以添加附加内容的提要,因为常驻应用程序能够执行这样的显示。在该情况下,提供附加内容的提要用于当 Java 应用程序请求虚拟包更新时与内容 ID 一起输入。这些提要可以为包含各自提要的文件指定文 件路径,而不是简单地包括字符串的直接输入。因而除了更新日期之外合并管理信息存储内容提要,并且常驻应用程序能够连同日期信息一起在附加内容列表中显示这些提要。

[0286] 胜于使 Java 应用程序指定内容提要,显示特定内容关于什么的元信息可以被追加到内容本身,常驻应用程序可以读取该信息并且基于读取的信息显示提要。

[0287] 图 37 中的“添加”栏显示各个附加内容首次与 BD-ROM 合并的日期。该信息也可

以从合并管理信息读取。

[0288] 需要注意的是首次合并附加内容的日期可以与合并管理信息分开保存。这些日期也可以从存储附加内容的目录被创建的日期确定。当显示在附加内容列表中的选择按钮之一被按压时，常驻应用程序将所选内容的目录路径 / 光盘 ID 和选择日期作为相应合并管理信息写入到合并管理信息文件中。换句话说，最近选择的附加内容变为最新合并管理信息。如果选择原始 BD-ROM，指示原始 BD-ROM 的值或空白单元被插入到合并管理信息文件的合并目标目录中。当显示在附加内容列表中的删除按钮之一被按压时，常驻应用程序参考合并管理信息文件读取用于删除的附加内容的目录，并且删除该目录。对应于该内容的内容 ID 的合并管理信息也从合并管理信息文件被删除。

[0289] 图 38 是关于本实施例的从 BD-ROM 的装载直到重放的处理流程的流程图。虚拟文件系统单元 38 首先检验被装载的 BD-ROM 的光盘 ID（步骤 S91），读取合并管理信息文件（步骤 S92），并且判断是否存在对应于被装载的 BD-ROM 的光盘 ID 的合并管理信息（步骤 S93）。如果判断是否定的（步骤 S93 ＝否），则仅仅使用原始 BD-ROM 执行重放（步骤 S94）。如果判断是肯定的（步骤 S93 ＝是），则使用最新合并管理信息产生虚拟包（步骤 S95）。在产生虚拟包期间，虚拟文件系统单元 38 判断是否已检测到错误（步骤 S96）。如果判断是肯定的（步骤 S96 ＝是），则虚拟文件系统单元 38 判断是否存在对应于 BD-ROM 的光盘 ID 的在先的合并管理信息（步骤 S97）。如果判断是肯定的（步骤 S97 ＝是），则使用在最新合并管理信息之前的合并管理信息的版本产生虚拟包（步骤 S98）。如果判断是否定的（步骤 S97 ＝否），则仅仅使用原始 BD-ROM 执行重放（步骤 S94）。如果在步骤 S96 未检测 到错误，使用产生的虚拟包执行重放（步骤 S99）。典型错误包括最新合并管理信息中的错误和不存在从播放列表引用的数据流等。

[0290] 根据本实施例，通过将先前的合并管理信息保存在合并管理信息文件中，可以参考合并管理信息文件的内容历史使用合并管理信息的旧版本从重放装置中的常驻应用程序产生虚拟包。如果在产生虚拟包期间出现错误，可以通过产生虚拟包的旧版本作为备选的动作过程避免所述问题。

[0291] 实施例 4

[0292] 本实施例涉及一种方法，该方法用于当 Java 应用程序请求虚拟包更新时指定虚拟包的有效期限，和仅仅在有效期限内使用虚拟包执行重放。

[0293] 图 39 显示了当请求虚拟包更新时被指定的有效期限。Java 应用程序指定用于合并的附加内容的内容 ID 和用于虚拟包的使用的有效期限。例如，如果用户想要作为虚拟包播放内容直到光盘被退出并且然后在重装载光盘之后仅仅使用原始 BD-ROM 播放内容，则到光盘被退出为止指示虚拟包是有效的值在从 Java 应用程序请求虚拟包更新时的自变量中被指定。

[0294] 图 40 是关于本实施例的从 BD-ROM 的装载（或重放装置的重启）直到重放的处理流程的流程图。虚拟文件系统单元 38 首先检验被装载的 BD-ROM 的光盘 ID（步骤 S101），读取合并管理信息文件（步骤 S102），并且判断是否存在对应于被装载的 BD-ROM 的光盘 ID 的合并管理信息（步骤 S103）。如果判断是否定的（步骤 S103 ＝否），则仅仅使用原始 BD-ROM 执行重放（步骤 S104）。如果判断是肯定的（步骤 S103 ＝是），则虚拟文件系统单元 38 判断相应的合并管理信息是否在有效期限内（步骤 S105）。如果不再有效，则删除相

应的合并管理信息（步骤 S106），并且仅仅使用原始 BD-ROM 执行重放（步骤 S104）。如果仍然有效，则相应的合并管理信息用于产生虚拟包（步骤 S107），并且使用虚拟包执行重放（步骤 S108）。

[0295] 需要注意的是作为本实施例的应用也可以想到其中仅 Java 模式虚拟包被产生的方式。如果当从 Java 应用程序请求产生虚拟包时仅指定 Java 模式，当存在从 DVD-like 模式到 Java 模式的过渡时，虚拟文件系统单元 38 产生虚拟包并且然后变换到 Java 模式。相反地，当存在从 Java 模式到 DVD-like 模式的过渡时，在取消虚拟包并回复到原始 BD-ROM 之后虚拟文件系统单元 38 变换到 DVD-like 模式。

[0296] 根据本实施例，有可能使用虚拟包指定重放的有效期限，由此允许使用仅一次虚拟包（即一旦 BD-ROM 被退出就被禁止的虚拟包）进行重放，和产生带有使用期限限制的虚拟包。

[0297] 需要注意的是在本实施例中有效期限在请求虚拟包更新时被指定，有效期限也可以在装载 BD-ROM 之后请求产生虚拟包时被指定。

[0298] 实施例 5

[0299] 下面是关于在实施例 1 的图 25 中的步骤 S23 的调用应用程序的权限的具体描述。具体而言，本实施例涉及拒绝来自未授权 Java 应用程序的虚拟包更新请求的方法。

[0300] 图 41 显示了用于筛选虚拟包更新请求的许可请求文件。如上所述，在来自 Java 应用程序的更新请求的基础上执行虚拟包更新。然而，当在来自未授权 Java 应用程序的请求的基础上执行更新时，由于查看限制被改变或者仅仅可以在某些条件下查看的视频剪辑的重放被允许，存在光盘内容被非法更新的危险。鉴于此，根据本实施例的虚拟包更新仅仅可以由具有更新许可的 Java 应用程序请求，所述更新许可是显示请求更新的许可已被准许的信息。通过检验与发布请求的 Java 应用程序对应的许可请求文件的内容判断是否具有更新许可。具体而言，类装载器根据各个许可请求文件的内容限制 Java 应用程序的功能。例如，如果许可请求文件中的更新属性值为“真”则处理更新请求，如果为“假”则拒绝。

[0301] 图 42 显示了强加在用于合并的本地存储器中的目录上的访问限制。如果用于合并的目录的内容由未授权 Java 应用程序更改，即使更新请求被筛选也存在虚拟包的内容被非法改变的危险。鉴于此，也在各个许可请求文件的内容的基础上限制对本地存储器访问的许可。例如，如果许可请求文件中的读和写属性值都为“真”，下载的内容可以被写入并且存储的文件可以被读取和编辑。然而，将在带有读和写属性之一或两者为“假”的许可请求文件的 Java 应用程序，或不带有许可请求文件的 Java 应用程序上强加文件访问限制。

[0302] 根据本实施例，可以防止未授权 Java 应用程序更新虚拟包和改变本地存储器中的目录的内容。

[0303] 下面是使用许可请求文件来限制虚拟包更新的具体例子。考虑一个例 子，其中本地存储器中的目录被分配给电影作品的特定供应商。具体而言，假设由 A 工作室、B 工作室和 C 公司提供的内容同时存储在本地存储器中和 BD-ROM 上。在这里，C 公司是数字杂志的供应商。当合并本地存储器和 BD-ROM 中的内容时，例如如果 B 工作室所提供的内容与 A 工作室所提供的内容合并将出现问题。鉴于此，仅仅准许 C 公司的更新许可（即许可请求文件中的更新属性值设置为“真”），由此允许使各种服务可用。A 工作室和 B 工作室仅仅能够合并它们自己的内容。

[0304] 实施例 6

[0305] 本实施例涉及一种方法，该方法用于在 Java 应用程序在多个标题上操作的情况下在标题变化期间更新虚拟包。

[0306] 图 43 显示了其生命周期被限制到单一标题的 Java 应用程序和其生命周期持续多个标题的 Java 应用程序。Java 应用程序的生命周期在应用程序管理信息中被显示，并且模块管理器 33 根据该应用程序管理信息管理 Java 应用程序的开始和结束。Java 应用程序包括仅仅在它们开始的标题内存在的那些应用程序（在下文被称为“标题绑定应用程序”）和可以在多个标题上存在的那些应用程序（在下文被称为“标题未绑定应用程序”）。应用程序管理信息包含标题数目，应用程序 ID 和显示特定 Java 应用程序是否被绑定的信息。

[0307] 在图 43 所示的应用程序管理信息中，例如，在标题 #1 中 Java 应用程序 #1 被绑定而 Java 应用程序 #2 未绑定。模块管理器 33 连同标题 #1 的结束一起终止绑定 Java 应用程序 #1。在另一方面，在标题 #1 结束之后允许未绑定的 Java 应用程序 #2 存活，并且根据下一个标题的应用程序管理信息作出是否终止该应用程序的决定。由于图 43 中的应用程序管理信息的例子表明 Java 应用程序 #2 可以同时在标题 #1 和标题 #2 中存在，因此允许该应用程序在从标题 #1 到标题 #2 的过渡期间存活。然而，由于 Java 应用程序 #2 在标题 #2 中是标题绑定的，因此模块管理器 33 连同该标题的结束一起终止应用程序。

[0308] 图 44 显示了在标题变化期间当更新虚拟包时在未绑定标题的应用程序上执行的处理。如图 43 中所示，可以在标题变化之前的标题和之后的标题中同时存在的标题未绑定应用程序在标题变化期间继续工作。然而，如果 请求虚拟包更新，包括标题未绑定应用程序的所有应用程序在标题变化期间被终止。在更新虚拟包之后，标题未绑定应用程序然后与属于下一个标题的标题绑定应用程序一起被重新启动。

[0309] 图 45 是考虑未绑定标题应用程序的标题变化处理的流程图。当标题重放开始时（步骤 S111），虚拟文件系统单元 38 首先判断在当前标题重放期间是否已从 Java 应用程序请求虚拟包更新（步骤 S112）。如果判断是肯定的（步骤 S112 ＝是），则虚拟文件系统单元 38 执行更新准备（步骤 S113）。当标题变化发生时（步骤 S114），虚拟文件系统单元 38 判断更新请求是否已被处理（步骤 S115）。如果判断是肯定的（步骤 S115 ＝是），终止包括标题未绑定应用程序的所有应用程序（步骤 S116），并且更新虚拟包（步骤 S117）。然后在标题变化之后播放下一个标题（步骤 S118）。如果在步骤 S112 虚拟包更新还未被请求或者如果在 S115 更新请求还未被处理，则当标题变化发生时模块管理器 33 仅仅终止标题绑定应用程序（步骤 S119）。

[0310] 由于本实施例保证所有应用程序在虚拟包的更新期间被终止，因此根据对旧的更新前文件的参考仍然保留在高速缓存中或者新文件与旧文件一起存在于高速缓存中，有可能在完成虚拟包更新之后防止任何连贯性损失。

[0311] 需要注意的是如果能够在多个光盘上存在的 Java 应用程序（“光盘未绑定应用程序”）在请求虚拟包更新之后发生标题变化时正在工作，通过以与光盘变化操作相同的方式对待虚拟包更新，光盘未绑定应用程序可以继续工作而不会被强制终止。

[0312] 还需要注意的是在发生标题变化之后模块管理器 33 可以根据完成虚拟包更新之后更新的应用程序管理信息，管理标题未绑定应用程序的开始和结束，而不会在更新期间终止标题未绑定应用程序。在该情况下，在更新完成之前，使标题未绑定应用程序参考更新

前的虚拟包。

[0313] 实施例 7

[0314] 本实施例涉及在 INDEX.BDMV 文件变化之后的虚拟包更新。一旦接收来自 Java 应用程序的虚拟包更新请求，虚拟文件系统单元 38 确认 INDEX.BDMV 文件存在于用于合并的目录中。如果 INDEX.BDMV 文件存在，虚拟文件系统单元 38 为准备更新读取 INDEX.BDMV 文件。然后现有的 INDEX.BDMV 文件被无效而新的 INDEX.BDMV 文件生效。如果例如 BD 播放器的常驻应用程序执行标题搜索或 Java 应用程序采集标题信息，在标题变化之前使用该新 INDEX.BDMV 文件。换句话说，在更新标题结构之后事先通知 Java 应用程序和用户使得有可能防止标题变化为将在更新之后停止存在的标题或标题变化到未预期的标题。

[0315] 图 46 是在 INDEX.BDMV 文件变化之后的虚拟包更新的流程图。首先，当在 Java 模式中播放标题时（步骤 S121），虚拟文件系统单元 38 判断 Java 应用程序是否已请求虚拟包更新（步骤 S122），并且如果已请求，则虚拟文件系统单元 38 接收请求并且执行更新准备（步骤 S123）。在检验文件和目录结构是否正确的同时，虚拟文件系统单元 38 判断 INDEX.BDMV 文件是否存在（步骤 S124）。如果 INDEX.BDMV 文件存在，则在判断是否已发生标题调用之前（步骤 S126），虚拟文件系统单元 38 使现有的 INDEX.BDMV 文件无效并且使新的 INDEX.BDMV 文件生效（步骤 S125）。参考在步骤 S125 中生效的 INDEX.BDMV 文件从 BD 播放器中的常驻应用程序或从 Java 应用程序执行标题变化。当标题变化发生时虚拟文件系统单元 38 执行更新（步骤 S127）。

[0316] 因而，尽管在更新请求后发生标题变化之前不执行虚拟包更新，在标题变化之前可以使新 INDEX.BDMV 文件可用。这意味着在更新请求之后，在标题搜索期间显示的标题列表将在标题变化发生之前已变化。

[0317] 由于用户然后基于改变的标题列表选择标题，因此可以防止由选择将在更新之后停止存在的标题引起的错误。因而可以在标题变化期间更新虚拟包而不会出现问题，即使由于更新而改变了标题结构。

[0318] 需要注意的是可以在重启 BD 播放器之后执行在 INDEX.BDMV 文件变化之后的虚拟包更新。

[0319] 变型

[0320] 以上基于优选实施例描述了关于本发明的重放装置，尽管本发明当然不限于这些实施例。

[0321] 关于其仅有的功能是播放记录介质的重放装置描述了以上实施例，尽管本发明并不限于此。例如，本发明可以是具有记录和重放功能的记录 / 重放装置。

[0322] 文件可以放置在使用任何类型结构的本地存储器中，只要与用于合并 的 BD-ROM 上的文件的对应关系被清楚地显示。

[0323] 在以上实施例中，Java（注册商标）用作虚拟机的编程语言，尽管也可以使用除了 Java 之外的编程语言，这样的例子包括 Perl Script，ECMA Script 和 B-Shell 等，其与 UNIX（注册商标）操作系统一起使用。

[0324] 关于播放 BD-ROM 的重放装置描述了以上实施例，尽管当然可以在如以上实施例中描述的 BD-ROM 上的必要数据被记录在可写光记录介质上的情况下实现与以上相同的效果。

[0325] 而且，当然可以在如以上实施例中描述的 BD-ROM 上的必要数据被记录在除了光记录介质之外的便携式记录介质（例如 SD 卡，小型闪存等）上的情况下实现与以上相同的效果。

[0326] 工业适用性

[0327] 构成本发明的重放装置可以在制造业中经营地、持续地和重复地被制造。该重放装置特别应用于关于视频内容产品的电影和消费用品产业。

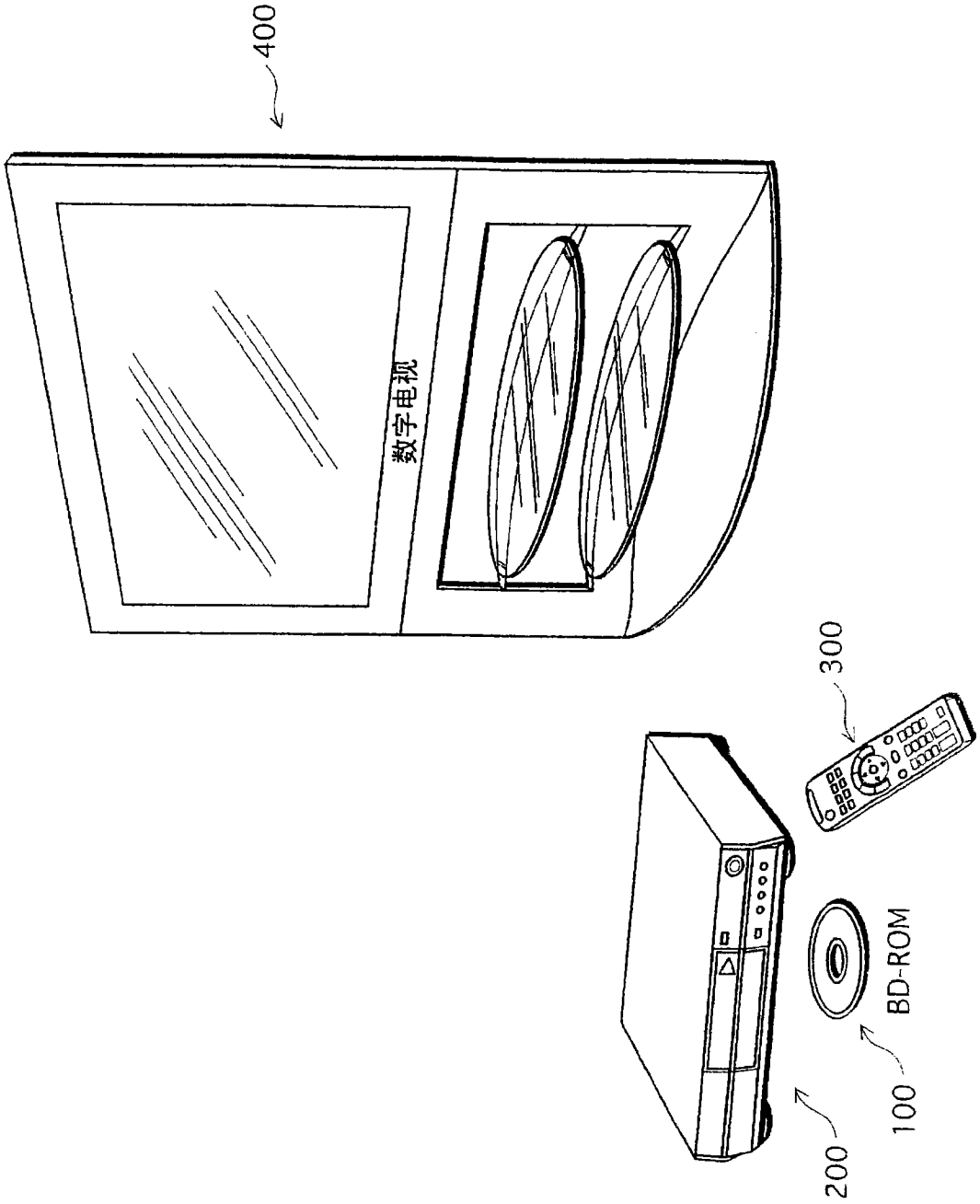


图 1

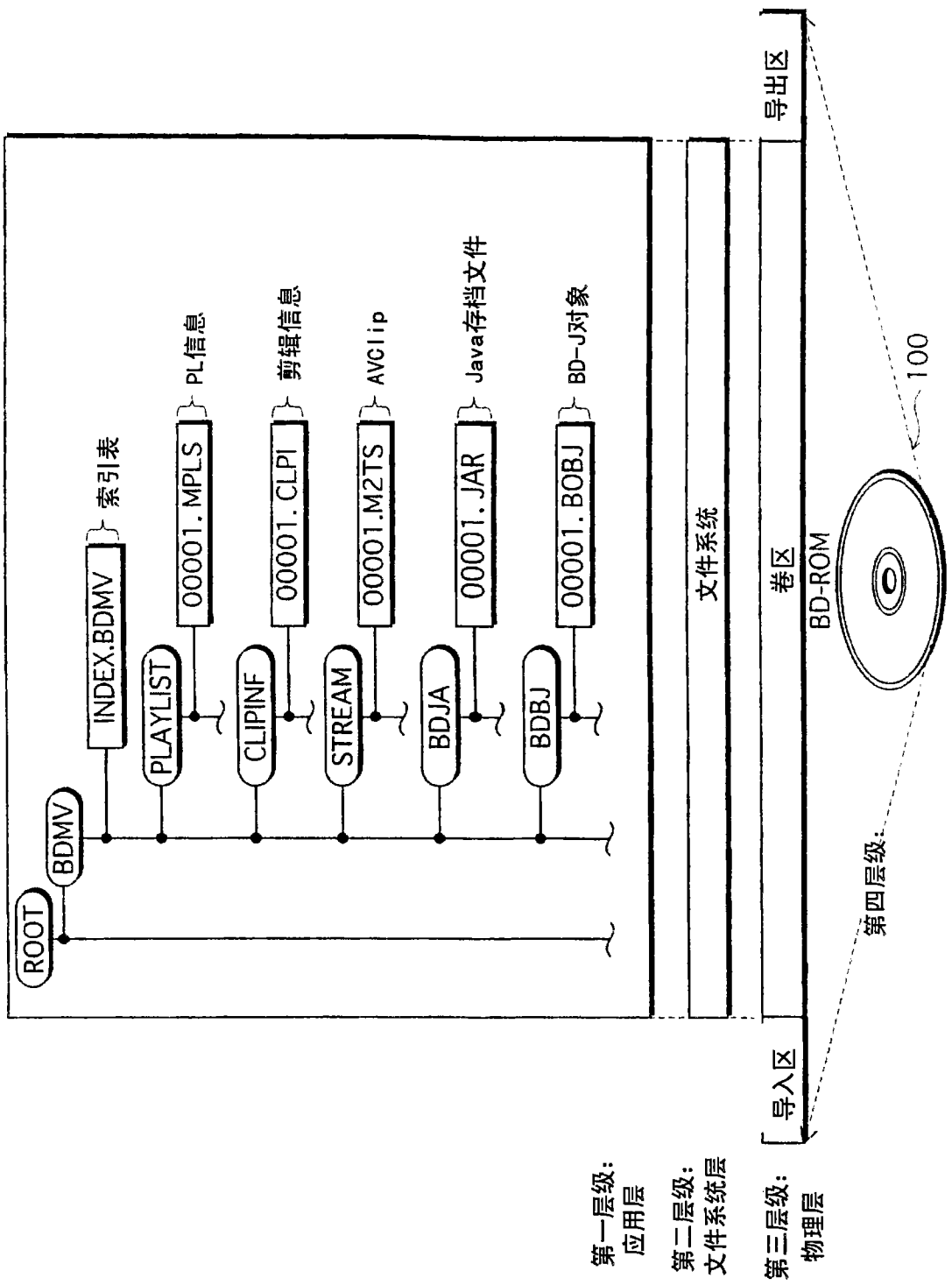

ROOT
BDMV
INDEX.BDMV
索引表
PLAYLIST
00001.MPLS
PL信息
CLIPINF
00001.CLPI
剪辑信息
STREAM
00001.M2TS
AVClip
BDJA
00001.JAR
Java存档文件
BDBJ
00001.BOBJ
BD-J对象
文件系统
导入区
卷区
导出区
BD-ROM
100
第四层级:
第一层级:
应用层
第二层级:
文件系统层
第三层级:
物理层

图 2

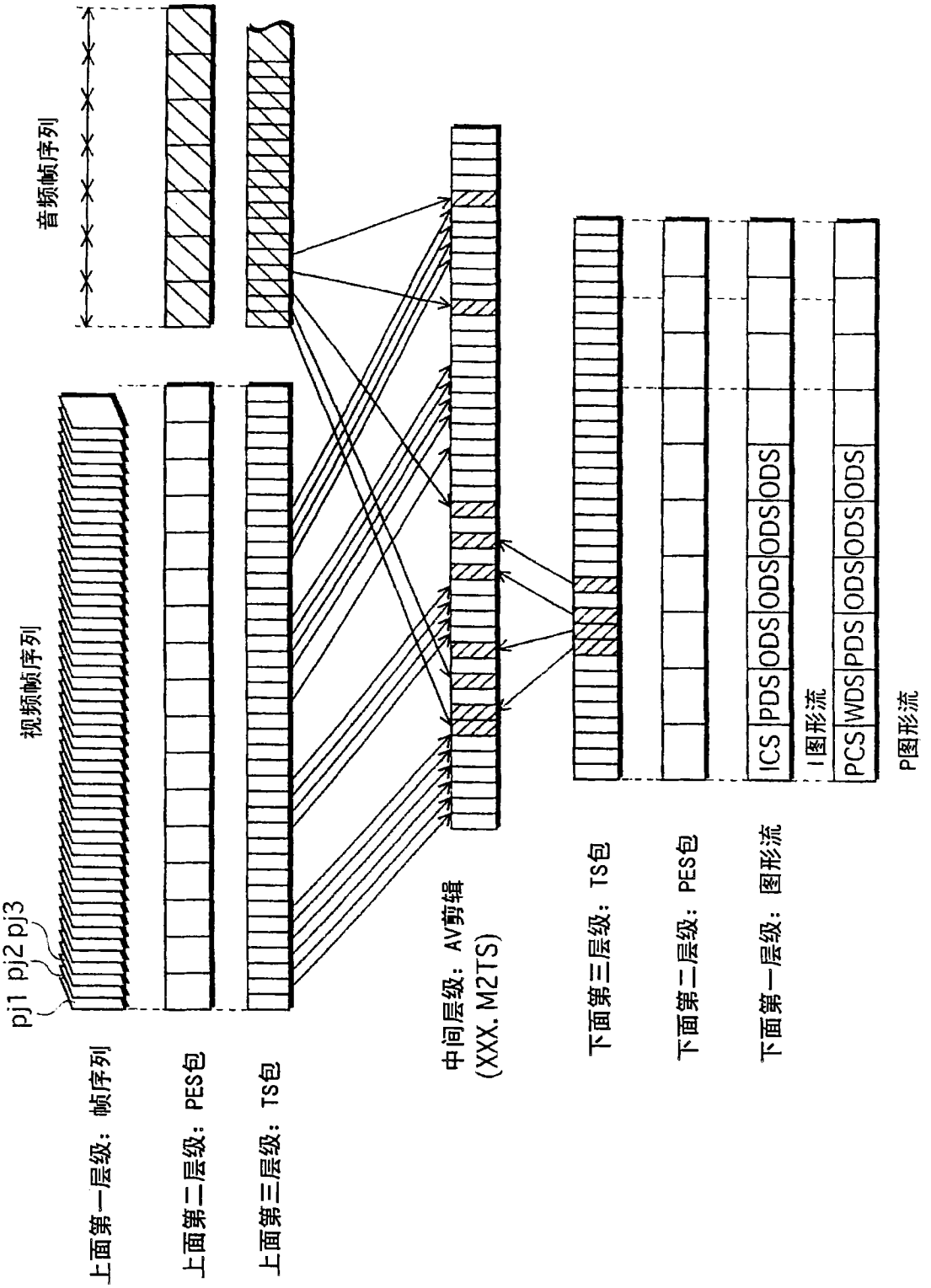

音频帧序列
视频帧序列
pj1 pj2 pj3
上面第一层级：帧序列
上面第二层级：PES包
上面第三层级：TS包
中间层级：AV剪辑
(XXX.M2TS)
下面第三层级：TS包
下面第二层级：PES包
下面第一层级：图形流
ICS
PDS
ODS
ODS
ODS
ODS
I图形流
PCS
WDS
PDS
ODS
ODS
ODS
P图形流

图 3

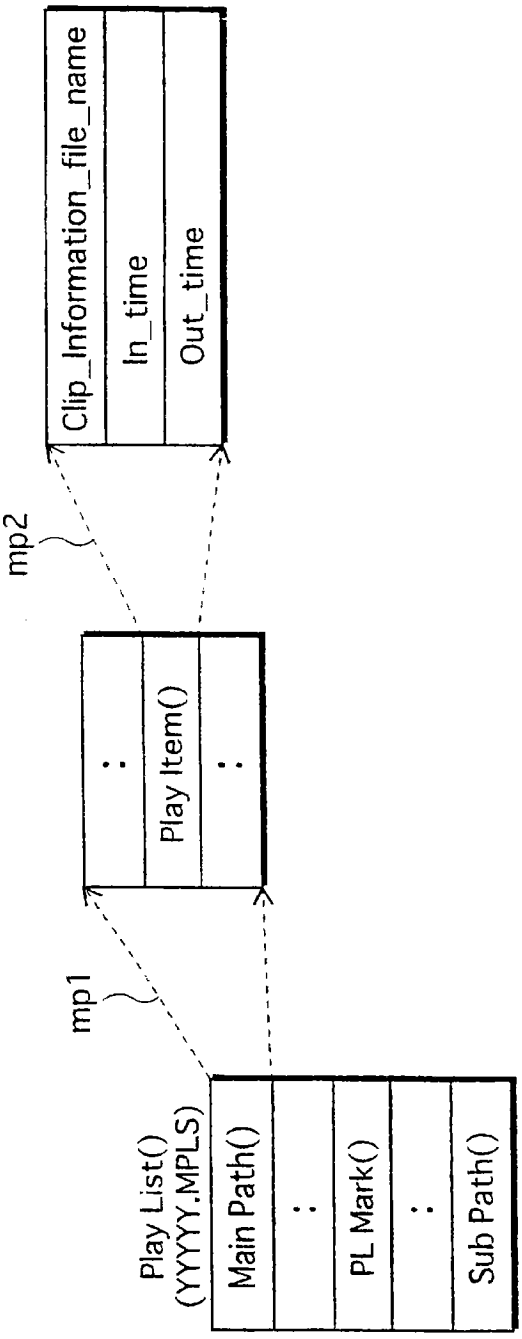

Clip_Information_file_name
In_time
Out_time
mp2
:
Play Item()
:
mp1
Play List()
(YYYYY.MPLS)
Main Path()
:
PL Mark()
:
Sub Path()

图 4

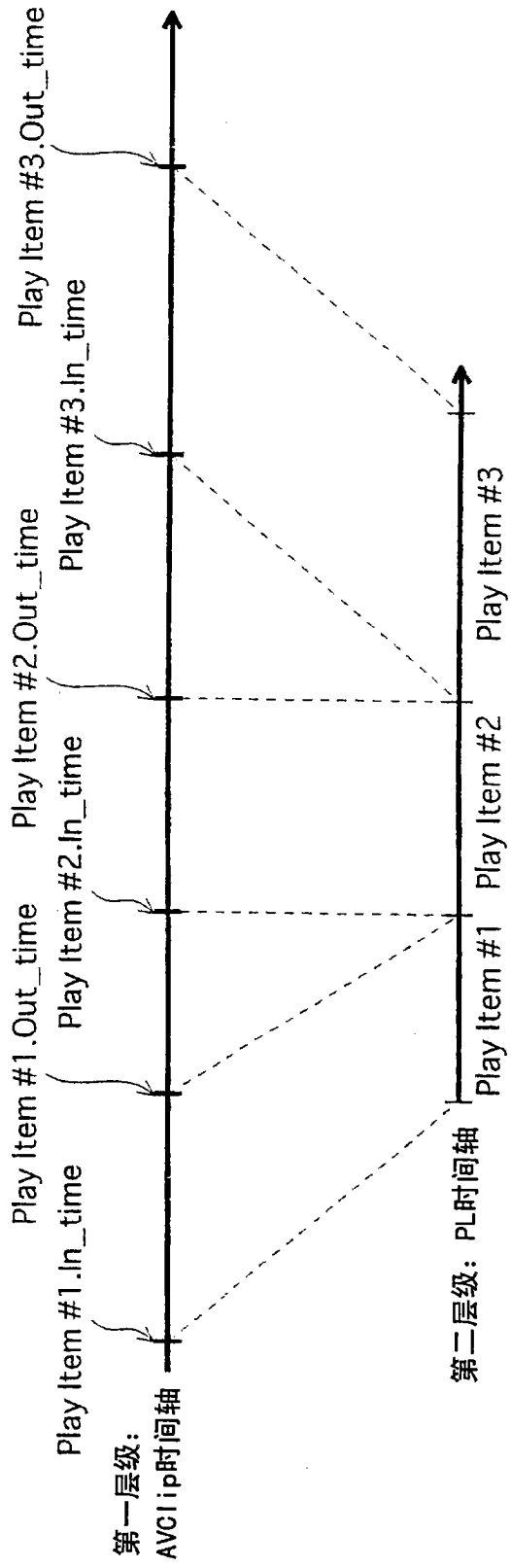

Play Item #3.Out_time
Play Item #3.In_time
Play Item #2.Out_time
Play Item #2.In_time
Play Item #1.Out_time
Play Item #1.In_time
第一层级：
AVClip时间轴
Play Item #1
Play Item #2
Play Item #3
第二层级：PL时间轴

图 5

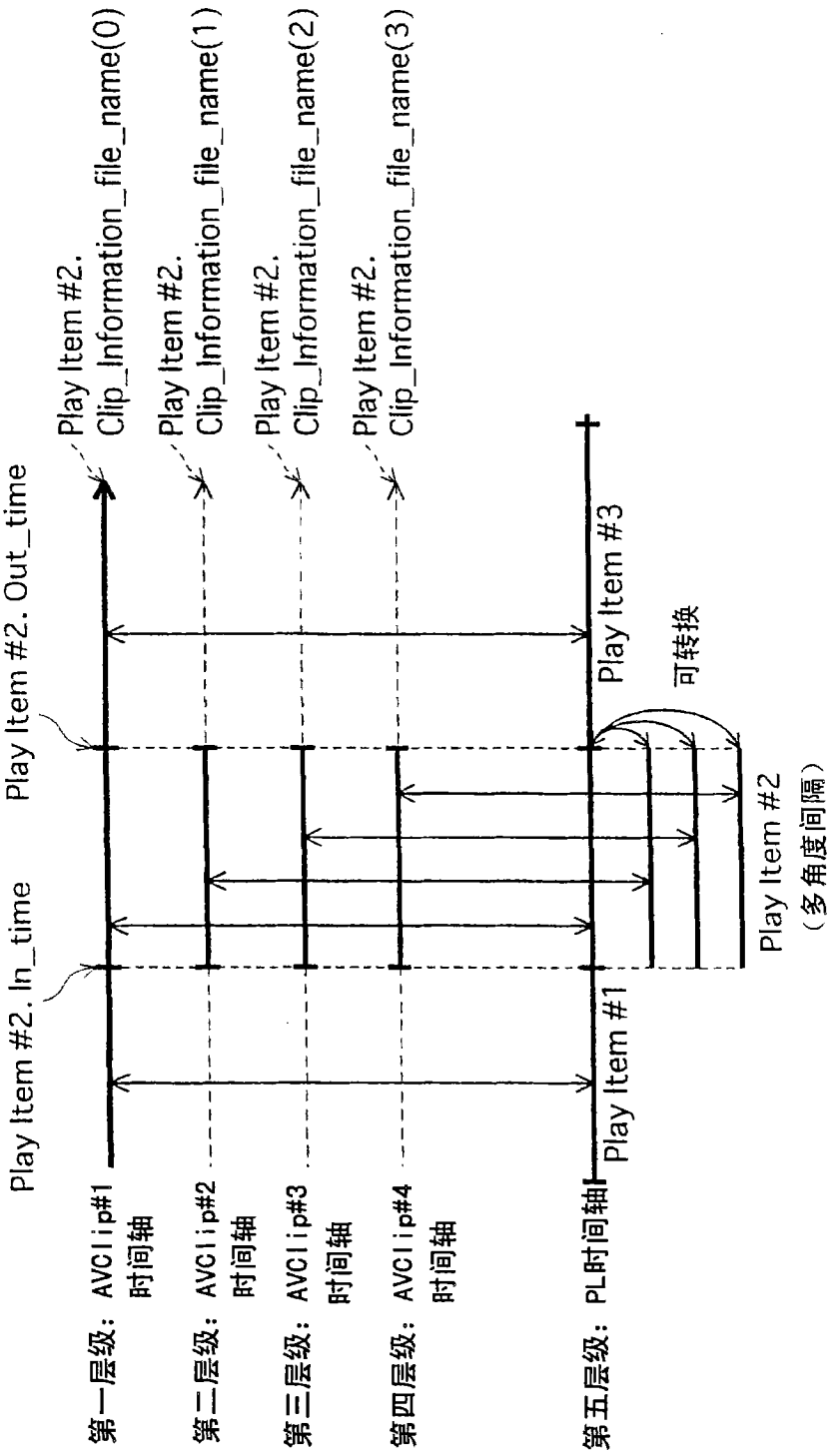

Play Item #2. In_time
Play Item #2. Out_time
Play Item #2. Clip_Information_file_name(0)
Play Item #2. Clip_Information_file_name(1)
Play Item #2. Clip_Information_file_name(2)
Play Item #2. Clip_Information_file_name(3)
第一层级：AVClip#1 时间轴
第二层级：AVClip#2 时间轴
第三层级：AVClip#3 时间轴
第四层级：AVClip#4 时间轴
第五层级：PL时间轴
Play Item #1
Play Item #3
可转换
Play Item #2
（多角度间隔）

图 6

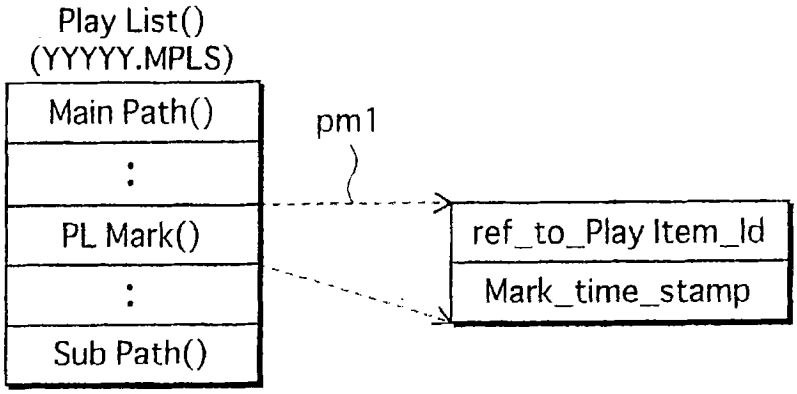

Play List()
(YYYYY.MPLS)
Main Path()
:
PL Mark()
:
Sub Path()
pm1
ref_to_Play Item_Id
Mark_time_stamp

图 7

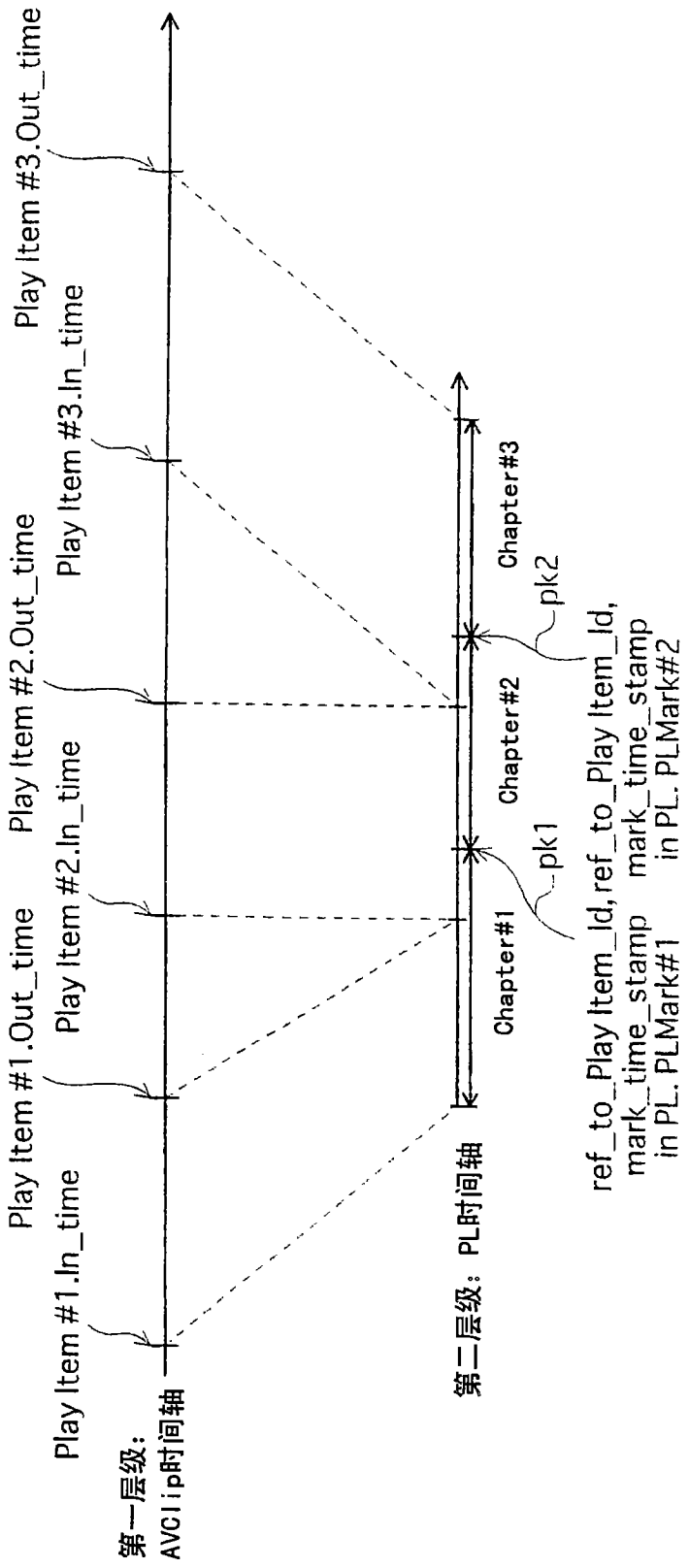

第一层级：
AVClip时间轴
Play Item #1.In_time
Play Item #1.Out_time
Play Item #2.In_time
Play Item #2.Out_time
Play Item #3.In_time
Play Item #3.Out_time
第二层级：PL时间轴
Chapter#1
Chapter#2
Chapter#3
pk1
pk2
ref_to_Play Item_Id, mark_time_stamp in PL. PLMark#1
ref_to_Play Item_Id, mark_time_stamp in PL. PLMark#2

图 8

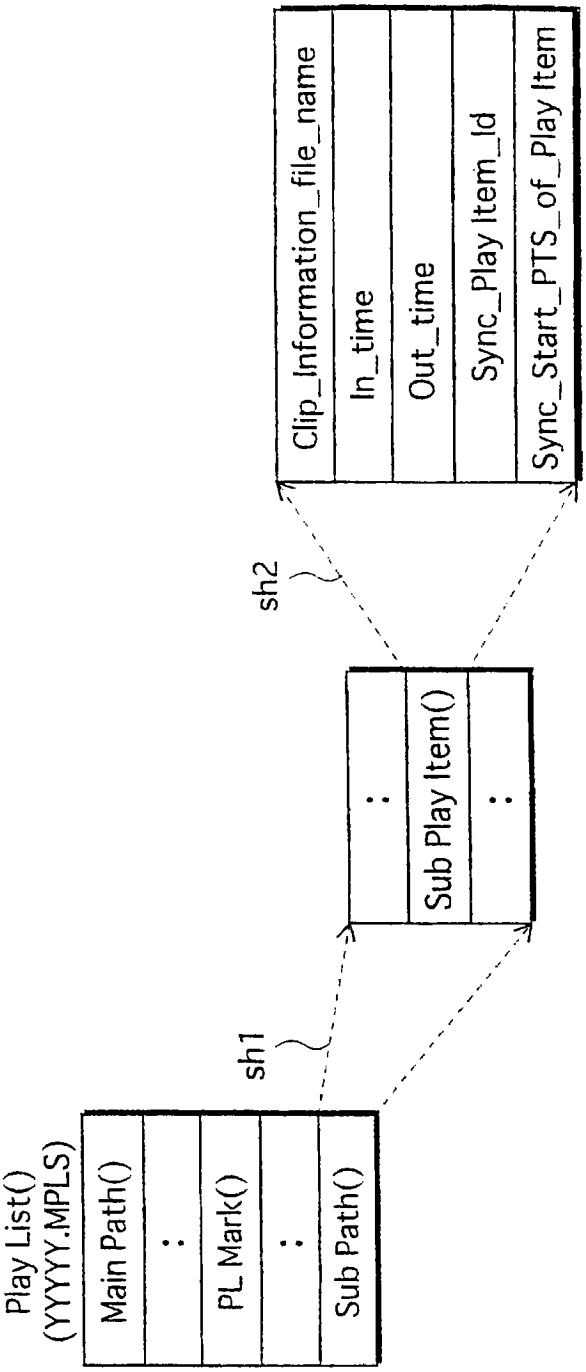

Play List()
(YYYYY.MPLS)
Main Path()
PL Mark()
Sub Path()
sh1
Sub Play Item()
sh2
Clip_Information_file_name
In_time
Out_time
Sync_Play Item_Id
Sync_Start_PTS_of_Play Item

图 9

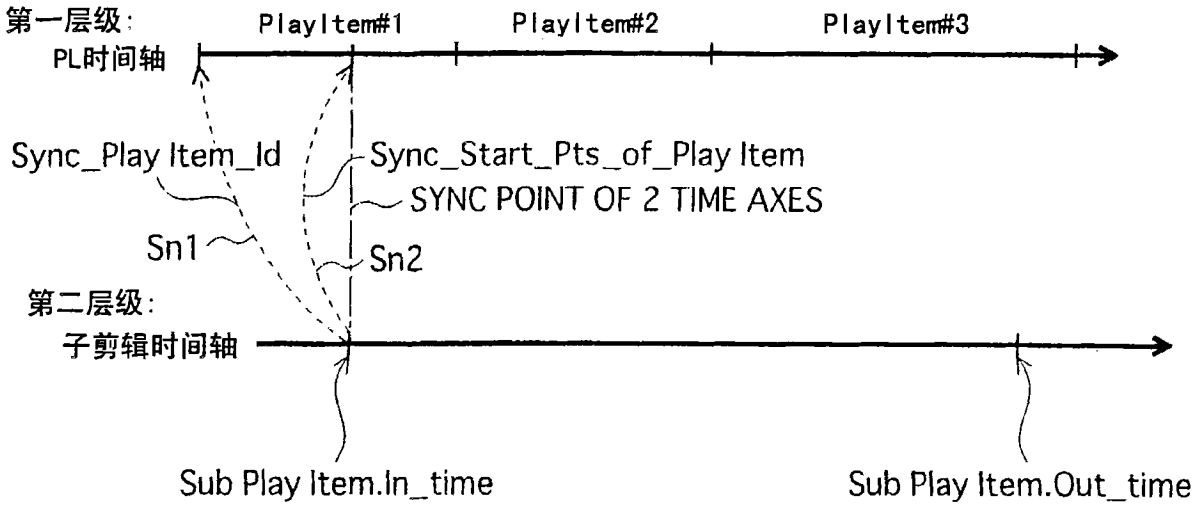

第一层级:
PlayItem#1
PlayItem#2
PlayItem#3
PL时间轴
Sync_Play Item_Id
Sync_Start_Pts_of_Play Item
SYNC POINT OF 2 TIME AXES
Sn1
Sn2
第二层级:
子剪辑时间轴
Sub Play Item.In_time
Sub Play Item.Out_time

图 10

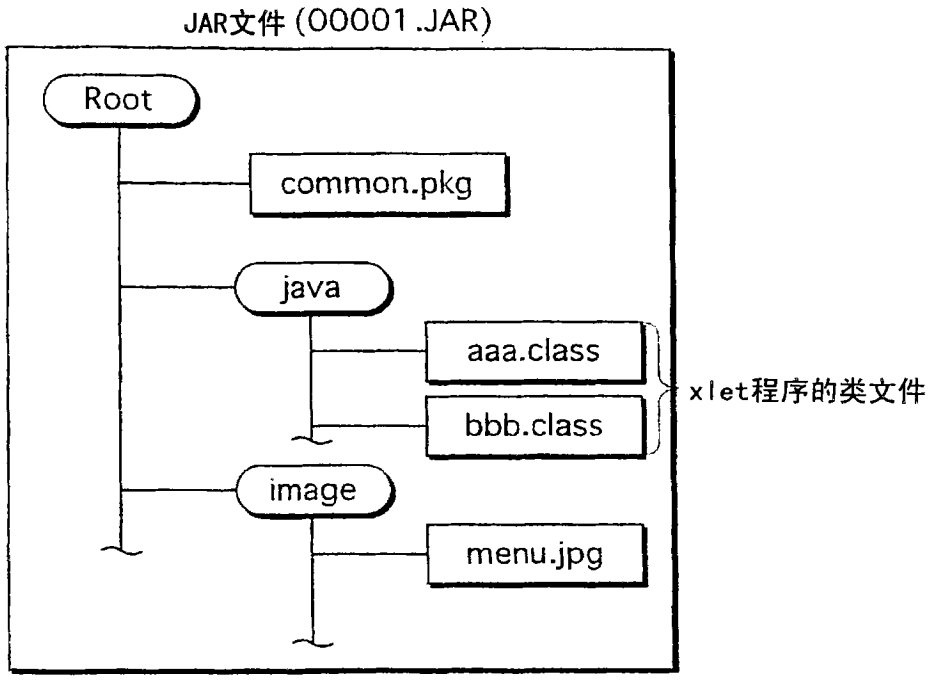

JAR文件 (00001 .JAR)
Root
common.pkg
java
aaa.class
bbb.class
xlet程序的类文件
image
menu.jpg

图 11A

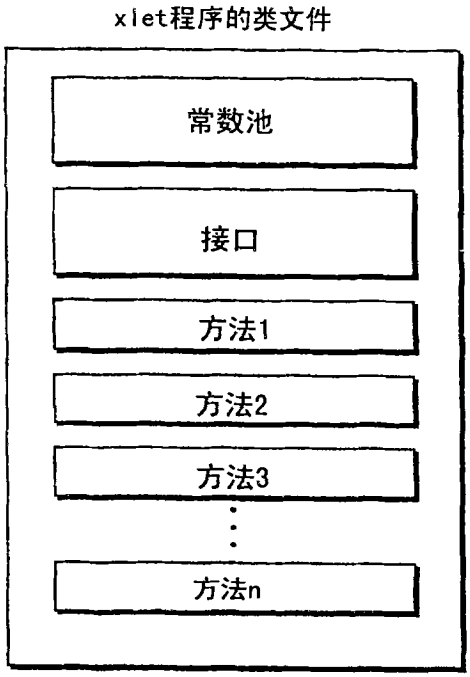

xlet程序的类文件
常数池
接口
方法1
方法2
方法3
方法n

图 11B

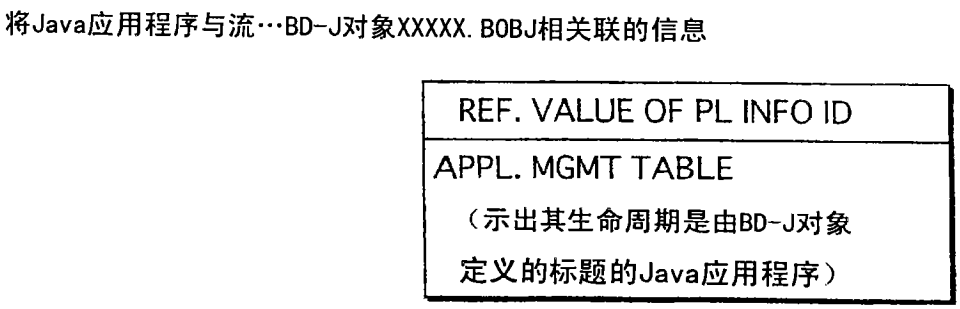

将Java应用程序与流…BD-J对象XXXXX.BOBJ相关联的信息
REF. VALUE OF PL INFO ID
APPL. MGMT TABLE
（示出其生命周期是由BD-J对象
定义的标题的Java应用程序）

图 12

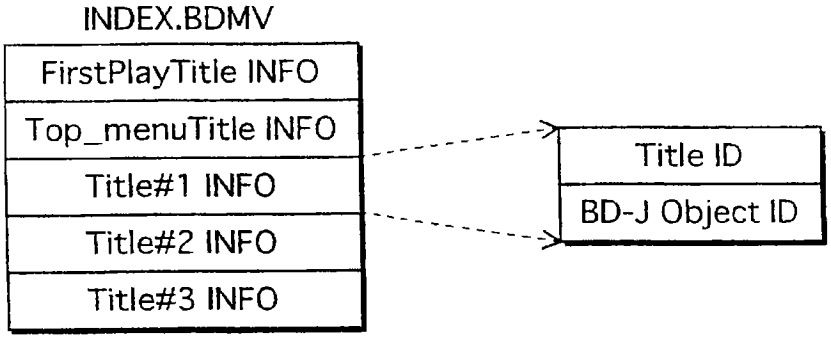

INDEX.BDMV
FirstPlayTitle INFO
Top_menuTitle INFO
Title#1 INFO
Title#2 INFO
Title#3 INFO
Title ID
BD-J Object ID

图 13

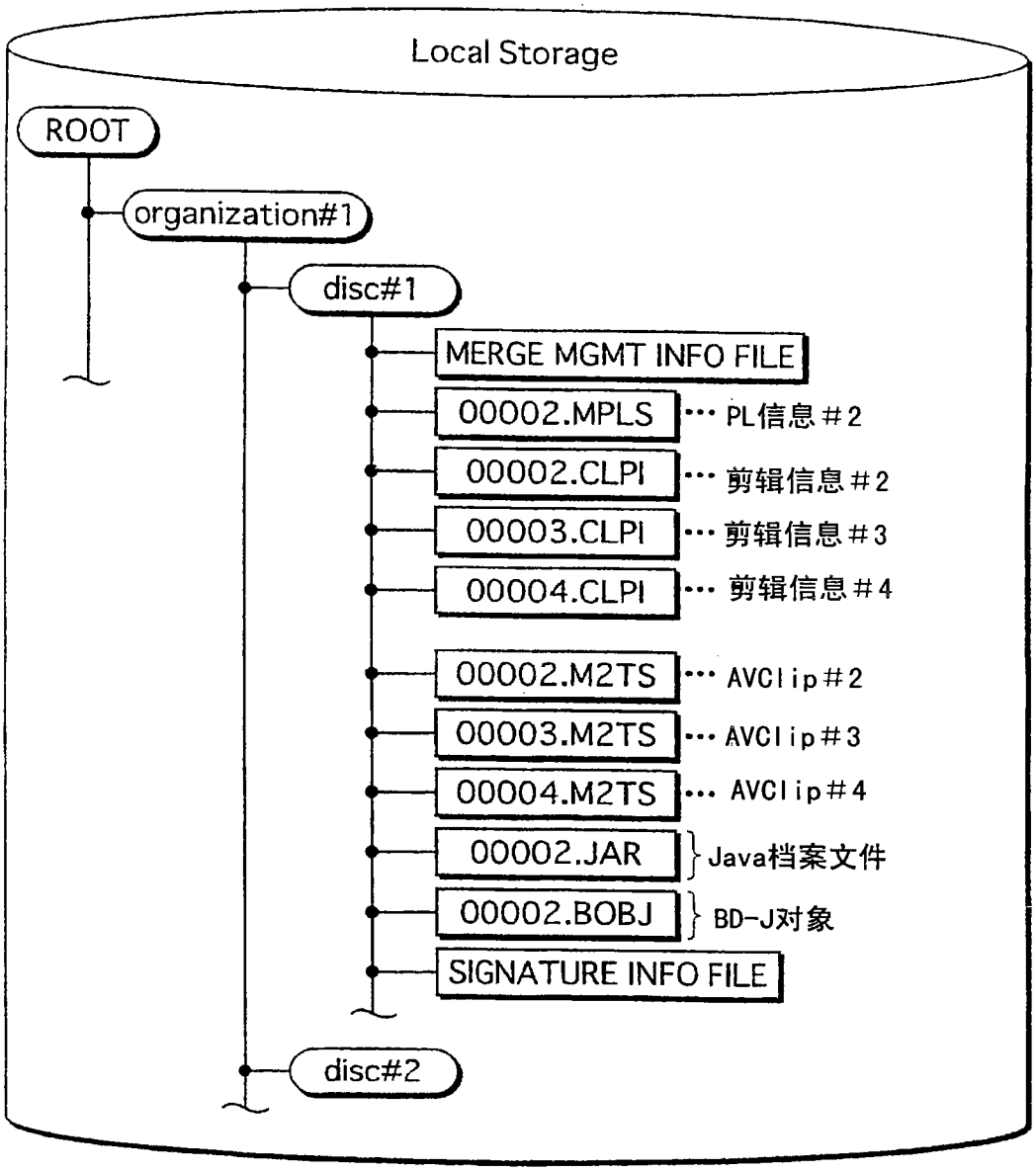

Local Storage
ROOT
organization#1
disc#1
MERGE MGMT INFO FILE
00002.MPLS
PL信息＃2
00002.CLPI
剪辑信息＃2
00003.CLPI
剪辑信息＃3
00004.CLPI
剪辑信息＃4
00002.M2TS
AVClip＃2
00003.M2TS
AVClip＃3
00004.M2TS
AVClip＃4
00002.JAR
Java档案文件
00002.BOBJ
BD-J对象
SIGNATURE INFO FILE
disc#2

图 14

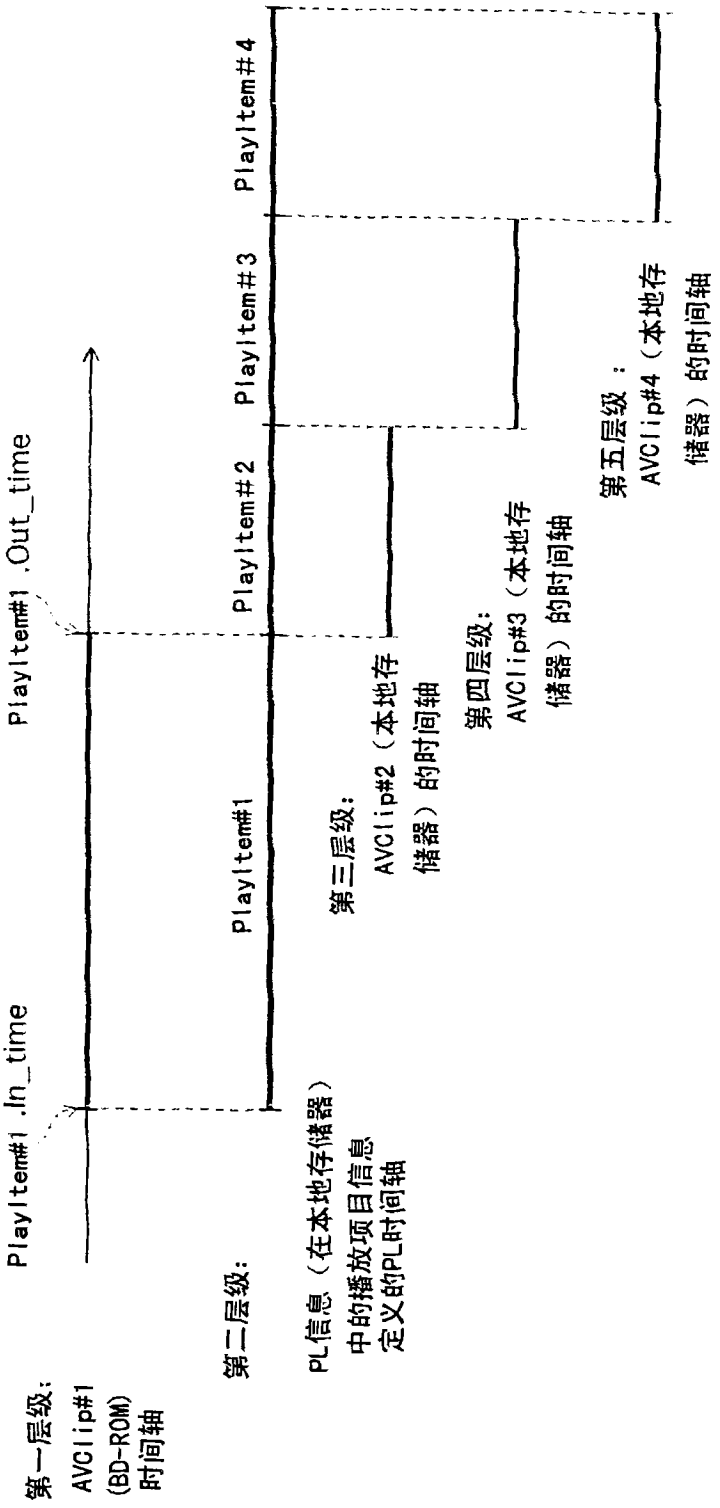

第一层级：
AVClip#1
(BD-ROM)
时间轴
Playitem#1 .In_time
Playitem#1 .Out_time
第二层级：
PL信息（在本地存储器）
中的播放项目信息
定义的PL时间轴
Playitem#1
Playitem#2
Playitem#3
Playitem#4
第三层级：
AVClip#2（本地存
储器）的时间轴
第四层级：
AVClip#3（本地存
储器）的时间轴
第五层级：
AVClip#4（本地存
储器）的时间轴

图 15

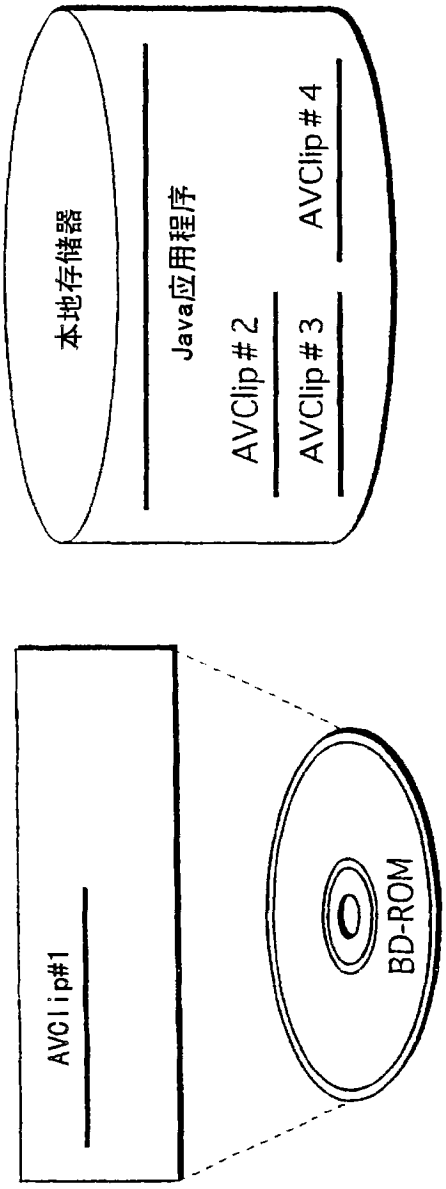


图 16A

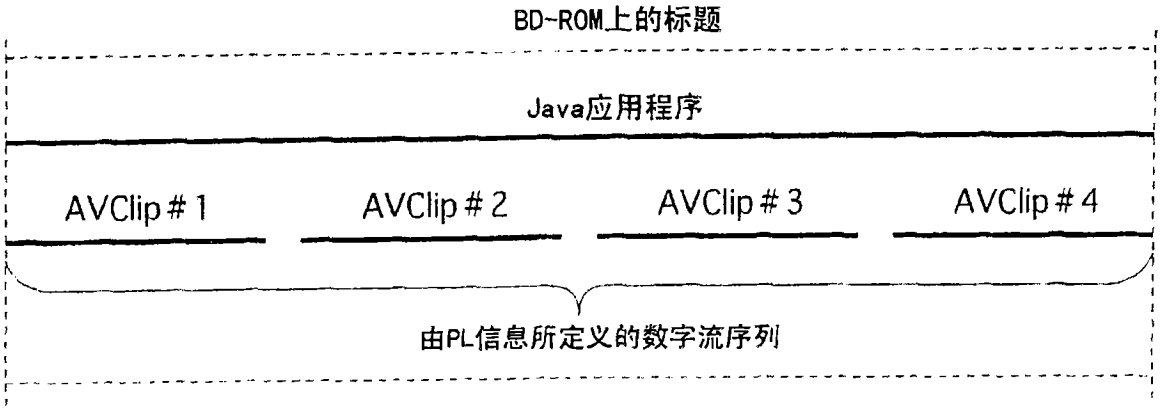


图 16B

合并管理信息文件

本地存储器中的存储位置	虚拟包中的ID
/organization #1/disc #1/00002.mpls	BDMV/PLAYLIST/00002.mpls
/organization #1/disc #1/00002.clpi	BDMV/CLIPINF/00002.clpi
/organization #1/disc #1/00003.clpi	BDMV/CLIPINF/00003.clpi
⋮	⋮

图 17

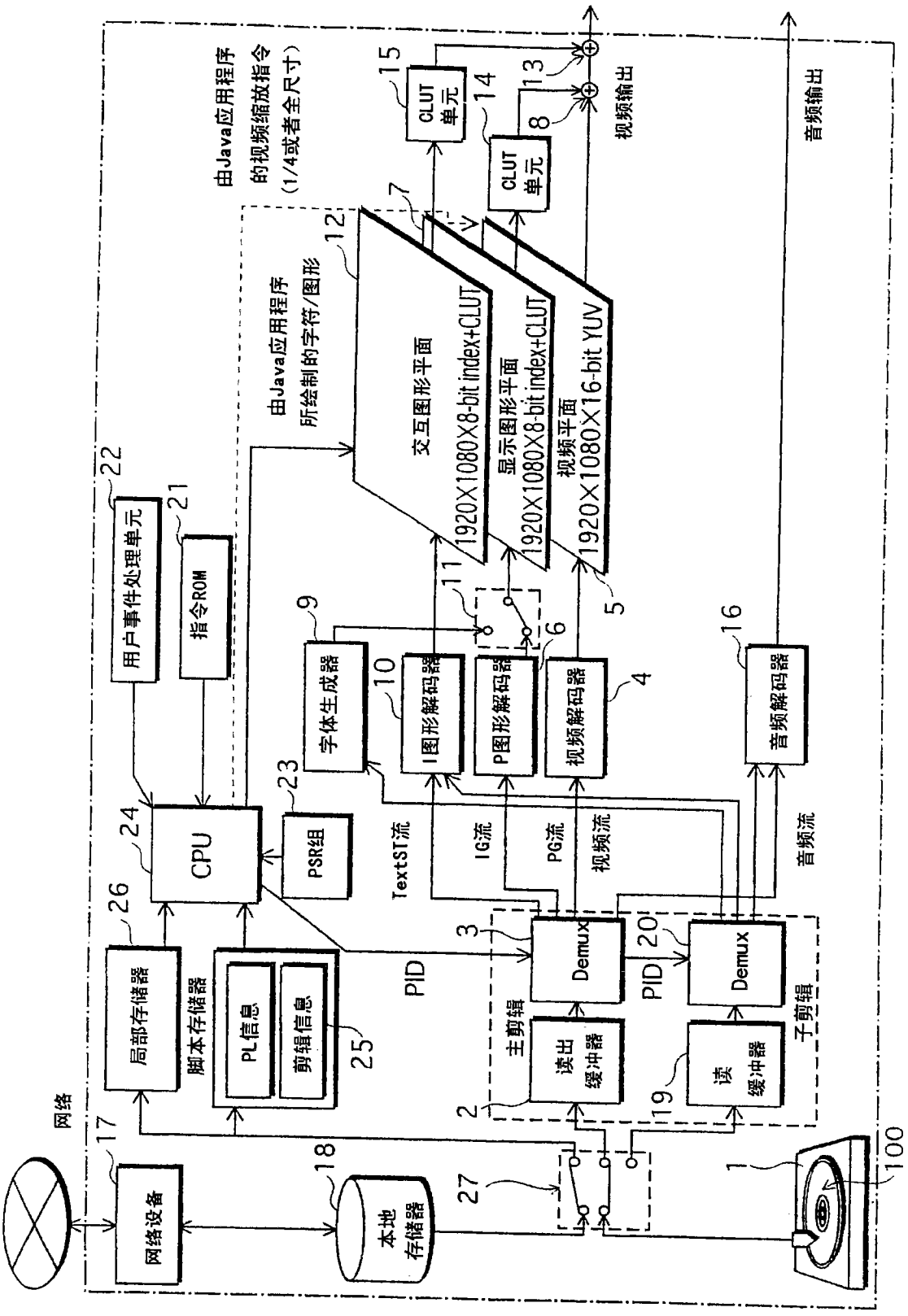

网络
网络设备
本地存储器
局部存储器
脚本存储器
PL信息
剪辑信息
用户事件处理单元
指令ROM
CPU
PSR组
字体生成器
由Java应用程序所绘制的字符/图形
由Java应用程序的视频缩放指令（1/4或者全尺寸）
交互图形平面
1920X1080X8-bit index+CLUT
显示图形平面
1920X1080X8-bit index+CLUT
视频平面
1920X1080X16-bit YUV
CLUT单元
CLUT单元
视频输出
音频输出
I图形解码器
P图形解码器
视频解码器
音频解码器
TextST流
IG流
PG流
视频流
音频流
PID
PID
Demux
Demux
主剪辑
子剪辑
读出缓冲器
读缓冲器
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
100

图 18

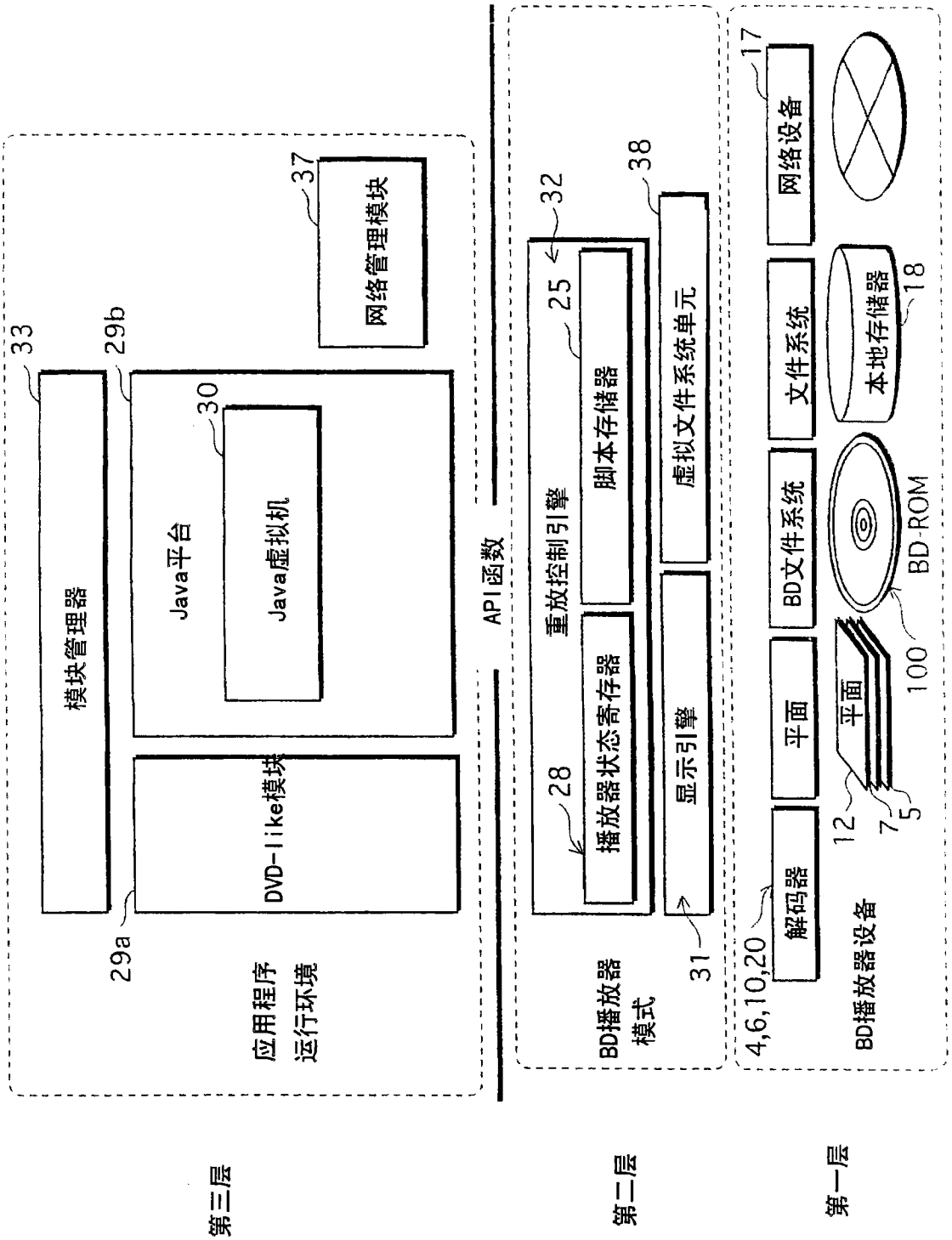

第三层
第二层
第一层
应用程序
运行环境
BD播放器
模式
BD播放器设备
模块管理器
Java平台
Java虚拟机
DVD-like模块
网络管理模块
API函数
重放控制引擎
播放器状态寄存器
显示引擎
脚本存储器
虚拟文件系统单元
解码器
平面
平面
BD文件系统
文件系统
本地存储器
网络设备
BD-ROM
33
29b
30
37
29a
32
25
28
38
31
4,6,10,20
12
7
5
100
17
18

图 19

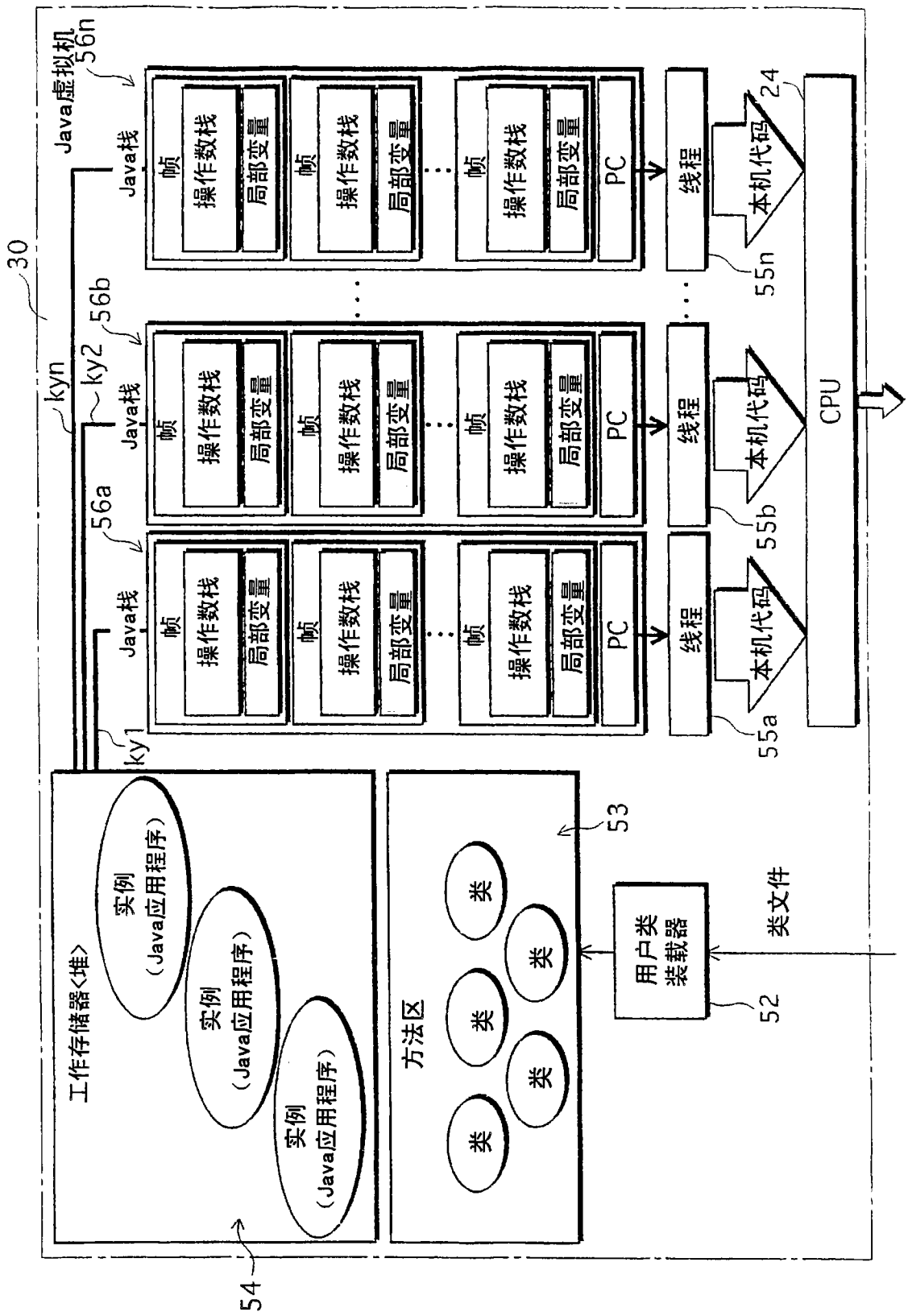

30
Java虚拟机
56n
Java栈
帧
操作数栈
局部变量
PC
线程
本机代码
24
55n
56b
ky2
kyn
56a
55b
CPU
ky1
55a
工作存储器<堆>
实例
（Java应用程序）
实例
（Java应用程序）
实例
（Java应用程序）
方法区
类
类
类
类
53
用户类
装载器
52
类文件
54

图 20

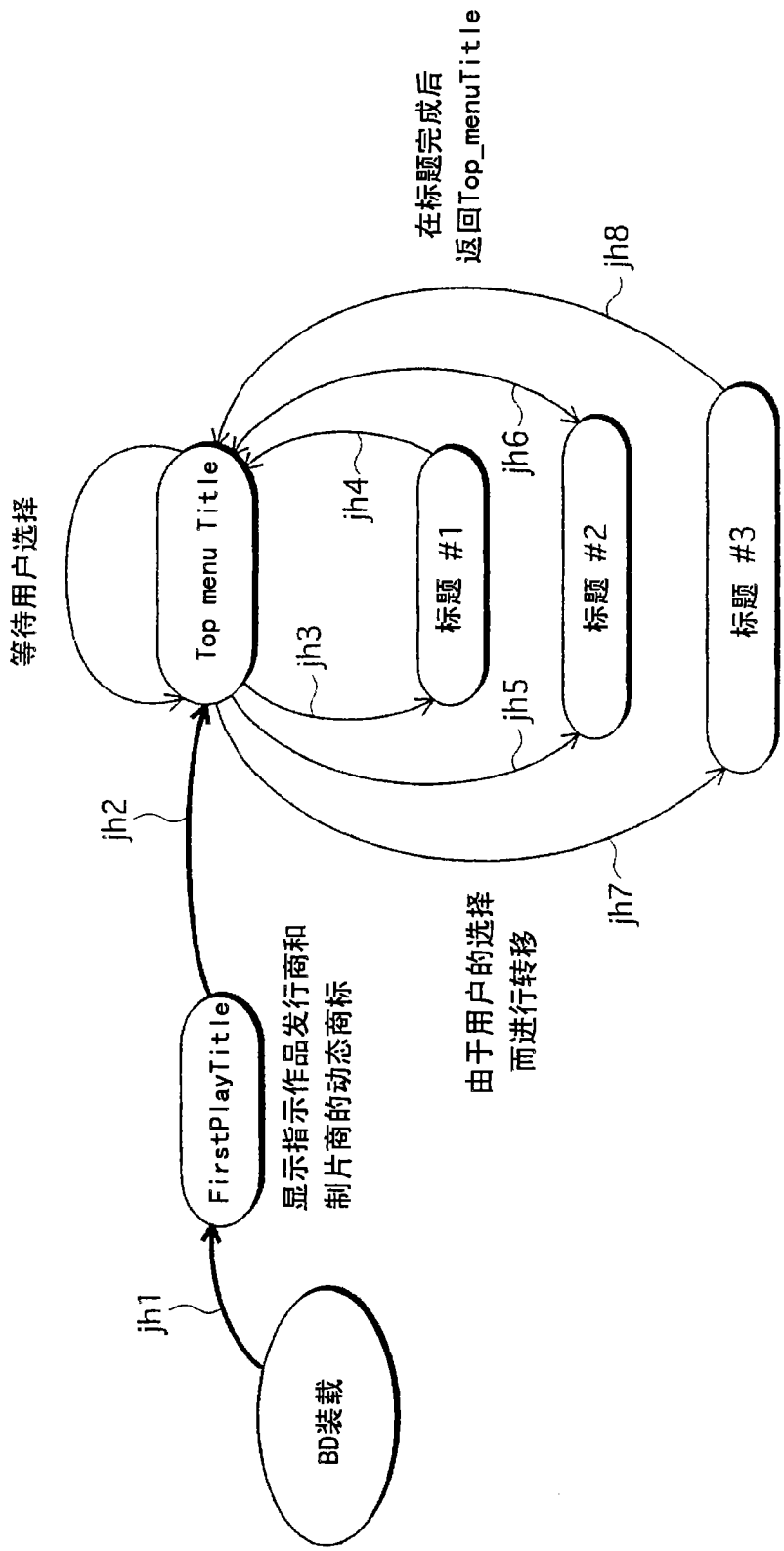

BD装载
jh1
FirstPlayTitle
显示指示作品发行商和
制片商的动态商标
jh2
Top menu Title
等待用户选择
jh3
jh4
标题 #1
jh5
标题 #2
jh6
标题 #3
由于用户的选择
而进行转移
jh7
在标题完成后
返回Top_menuTitle
jh8

图 21

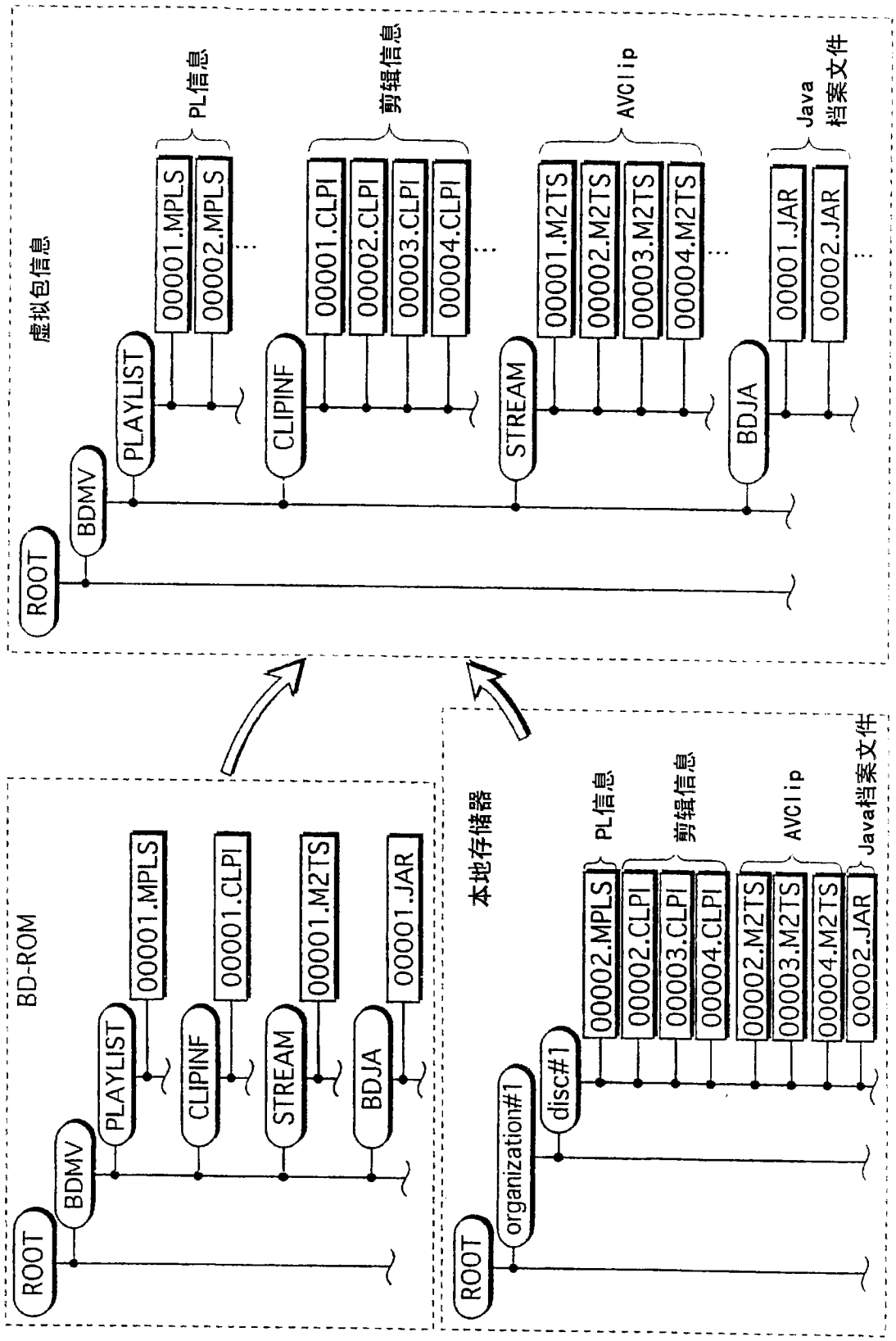

虚拟包信息
ROOT
BDMV
PLAYLIST
CLIPINF
STREAM
BDJA
00001.MPLS
00002.MPLS
00001.CLPI
00002.CLPI
00003.CLPI
00004.CLPI
00001.M2TS
00002.M2TS
00003.M2TS
00004.M2TS
00001.JAR
00002.JAR
PL信息
剪辑信息
AVClip
Java
档案文件
BD-ROM
本地存储器
organization#1
disc#1
Java档案文件

图 22

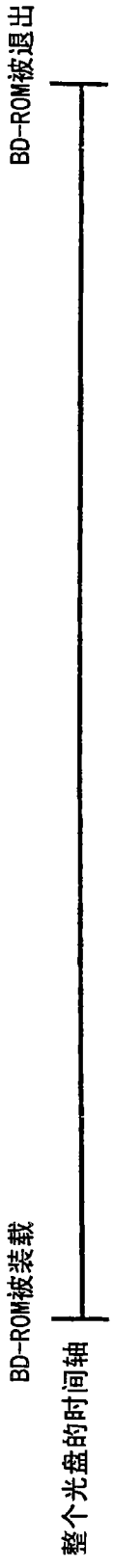

BD-ROM被退出
BD-ROM被装载
整个光盘的时间轴

图 23A

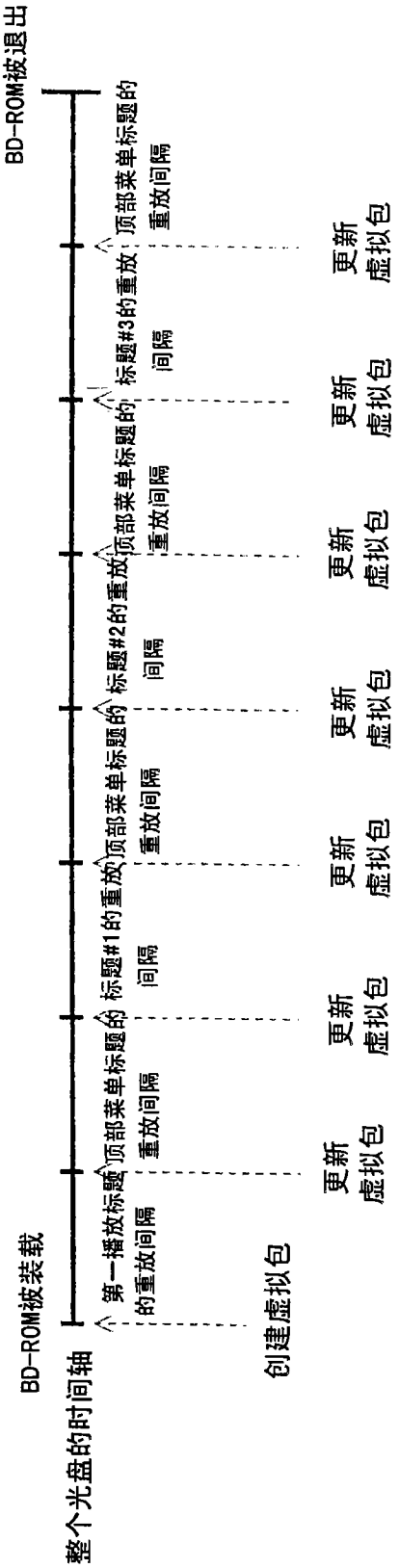

BD-ROM被退出
BD-ROM被装载
整个光盘的时间轴
第一播放标题的重放间隔
顶部菜单标题的重放间隔
标题#1的重放间隔
顶部菜单标题的重放间隔
标题#2的重放间隔
顶部菜单标题的重放间隔
标题#3的重放间隔
顶部菜单标题的重放间隔
创建虚拟包
更新虚拟包
更新虚拟包
更新虚拟包
更新虚拟包
更新虚拟包

图 23B

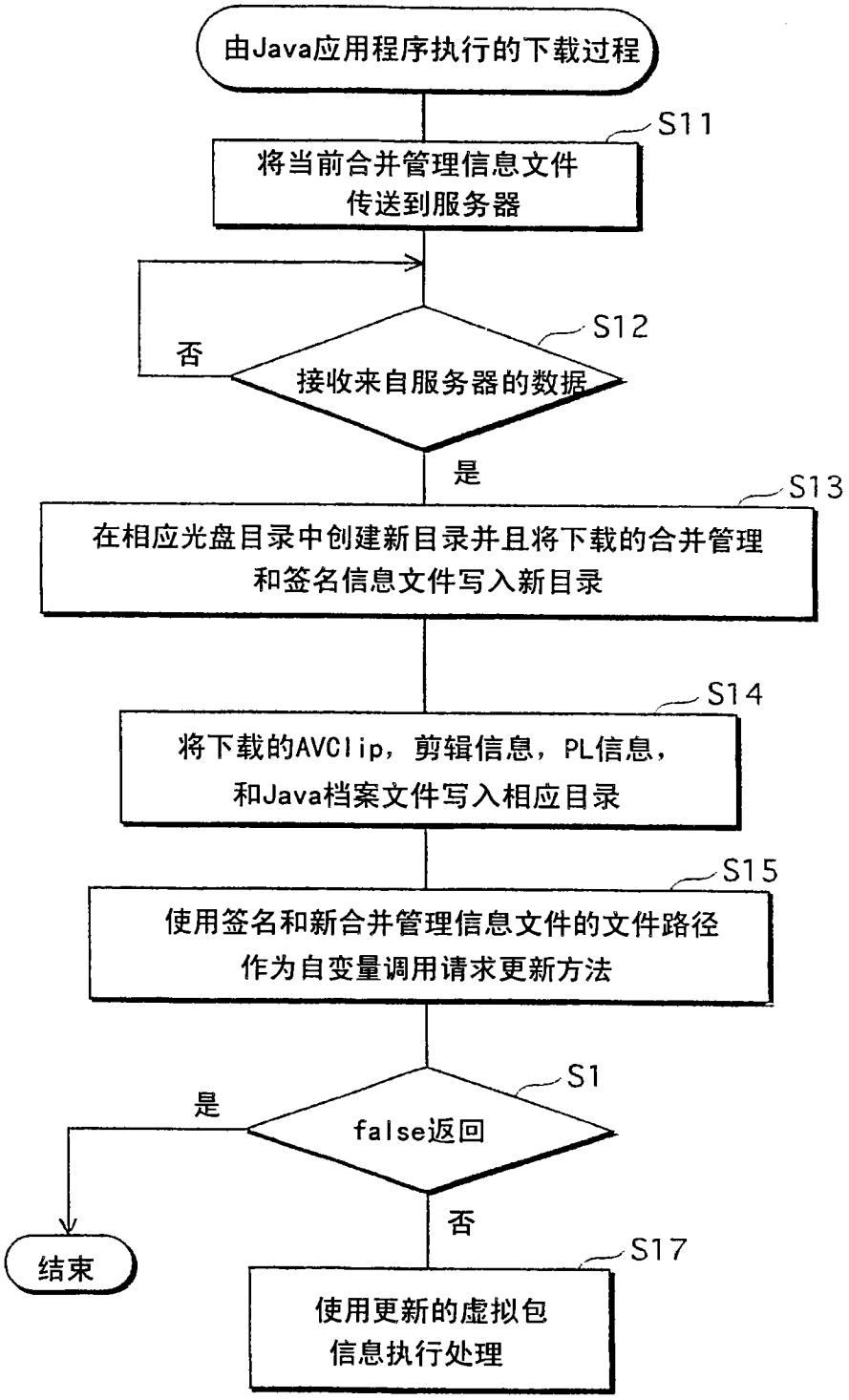

由Java应用程序执行的下载过程
S11
将当前合并管理信息文件
传送到服务器
S12
否
接收来自服务器的数据
是
S13
在相应光盘目录中创建新目录并且将下载的合并管理
和签名信息文件写入新目录
S14
将下载的AVClip，剪辑信息，PL信息，
和Java档案文件写入相应目录
S15
使用签名和新合并管理信息文件的文件路径
作为自变量调用请求更新方法
S1
是
false返回
否
结束
S17
使用更新的虚拟包
信息执行处理

图 24

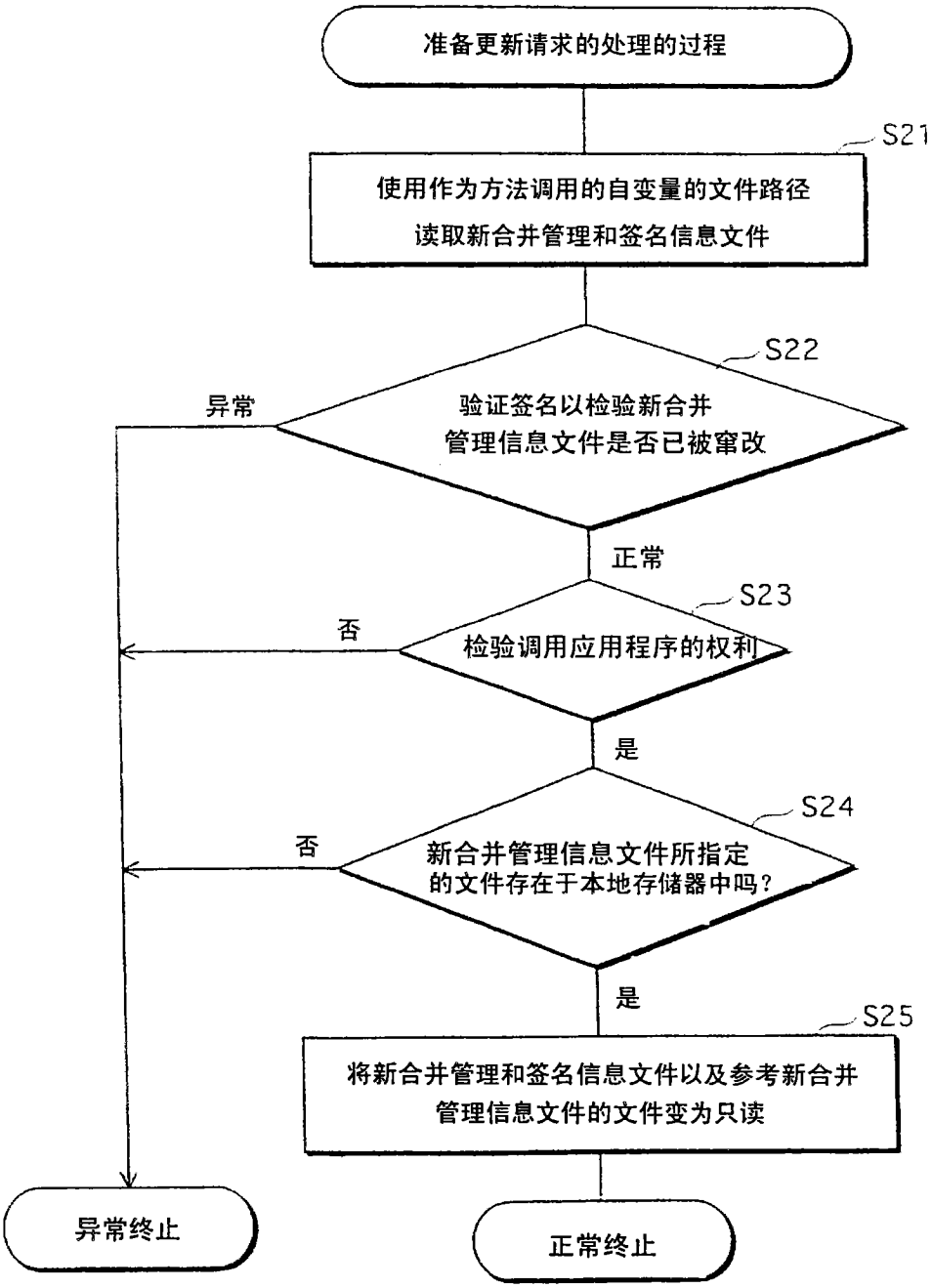

准备更新请求的处理的过程
S21
使用作为方法调用的自变量的文件路径
读取新合并管理和签名信息文件
S22
验证签名以检验新合并
管理信息文件是否已被窜改
异常
正常
S23
检验调用应用程序的权利
否
是
S24
新合并管理信息文件所指定
的文件存在于本地存储器中吗？
否
是
S25
将新合并管理和签名信息文件以及参考新合并
管理信息文件的文件变为只读
异常终止
正常终止

图 25

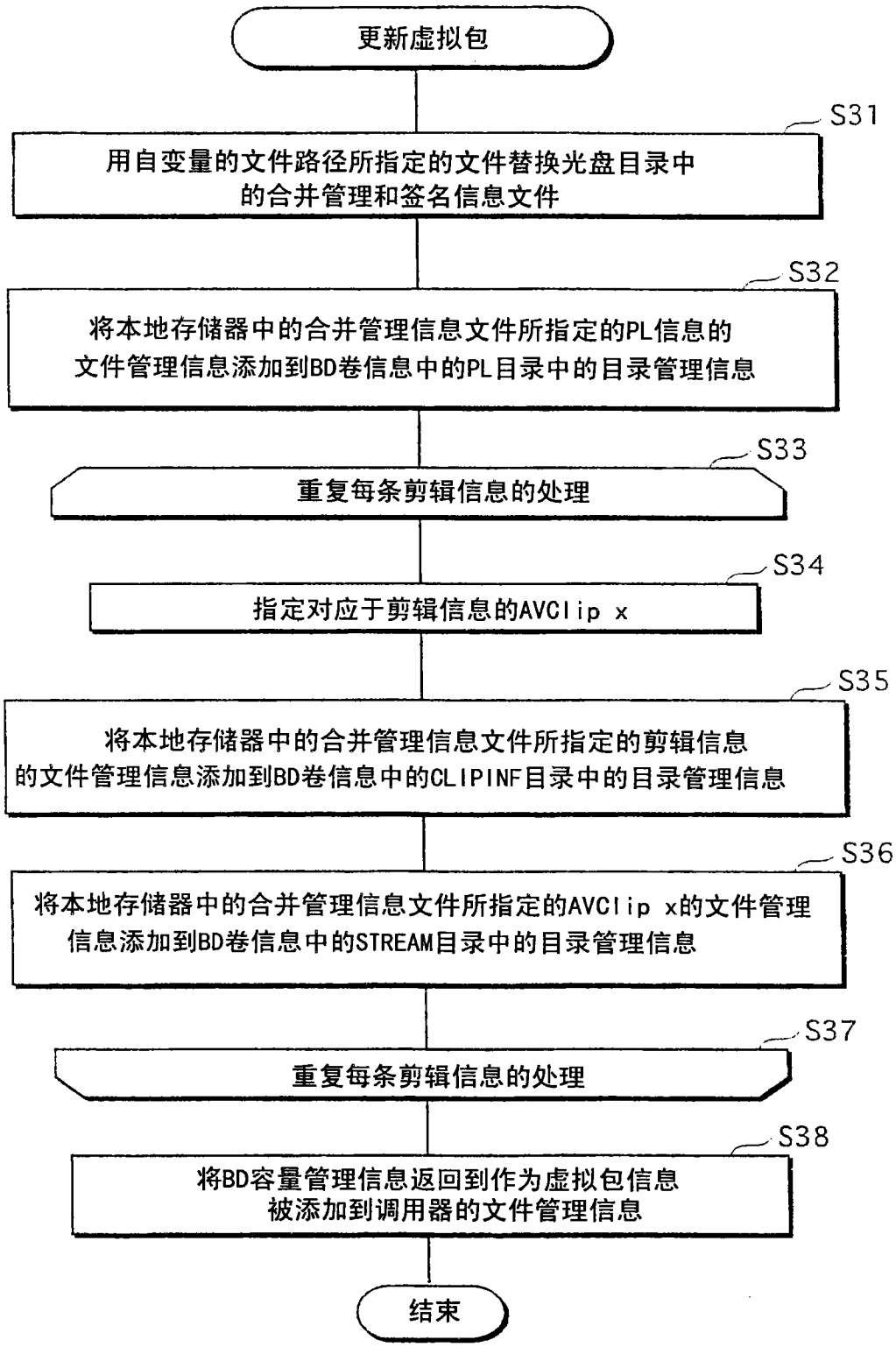

更新虚拟包
S31
用自变量的文件路径所指定的文件替换光盘目录中的合并管理和签名信息文件
S32
将本地存储器中的合并管理信息文件所指定的PL信息的文件管理信息添加到BD卷信息中的PL目录中的目录管理信息
S33
重复每条剪辑信息的处理
S34
指定对应于剪辑信息的AVClip x
S35
将本地存储器中的合并管理信息文件所指定的剪辑信息的文件管理信息添加到BD卷信息中的CLIPINF目录中的目录管理信息
S36
将本地存储器中的合并管理信息文件所指定的AVClip x的文件管理信息添加到BD卷信息中的STREAM目录中的目录管理信息
S37
重复每条剪辑信息的处理
S38
将BD容量管理信息返回到作为虚拟包信息被添加到调用器的文件管理信息
结束

图 26

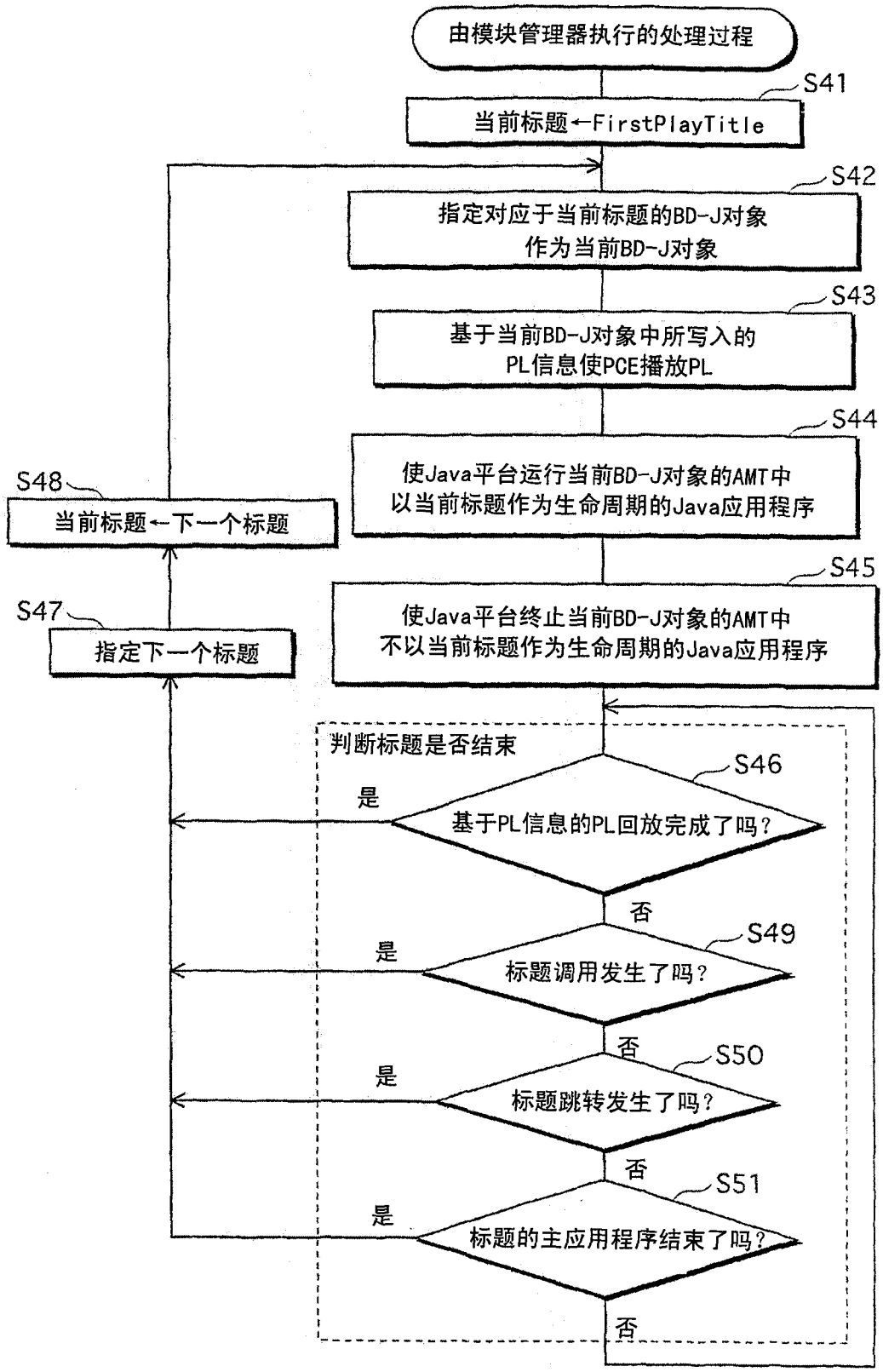

由模块管理器执行的处理过程
S41
当前标题←FirstPlayTitle
S42
指定对应于当前标题的BD-J对象
作为当前BD-J对象
S43
基于当前BD-J对象中所写入的
PL信息使PCE播放PL
S44
使Java平台运行当前BD-J对象的AMT中
以当前标题作为生命周期的Java应用程序
S45
使Java平台终止当前BD-J对象的AMT中
不以当前标题作为生命周期的Java应用程序
判断标题是否结束
S46
基于PL信息的PL回放完成了吗?
是
否
S49
标题调用发生了吗?
是
否
S50
标题跳转发生了吗?
是
否
S51
标题的主应用程序结束了吗?
是
否
S47
指定下一个标题
S48
当前标题←下一个标题

图 27

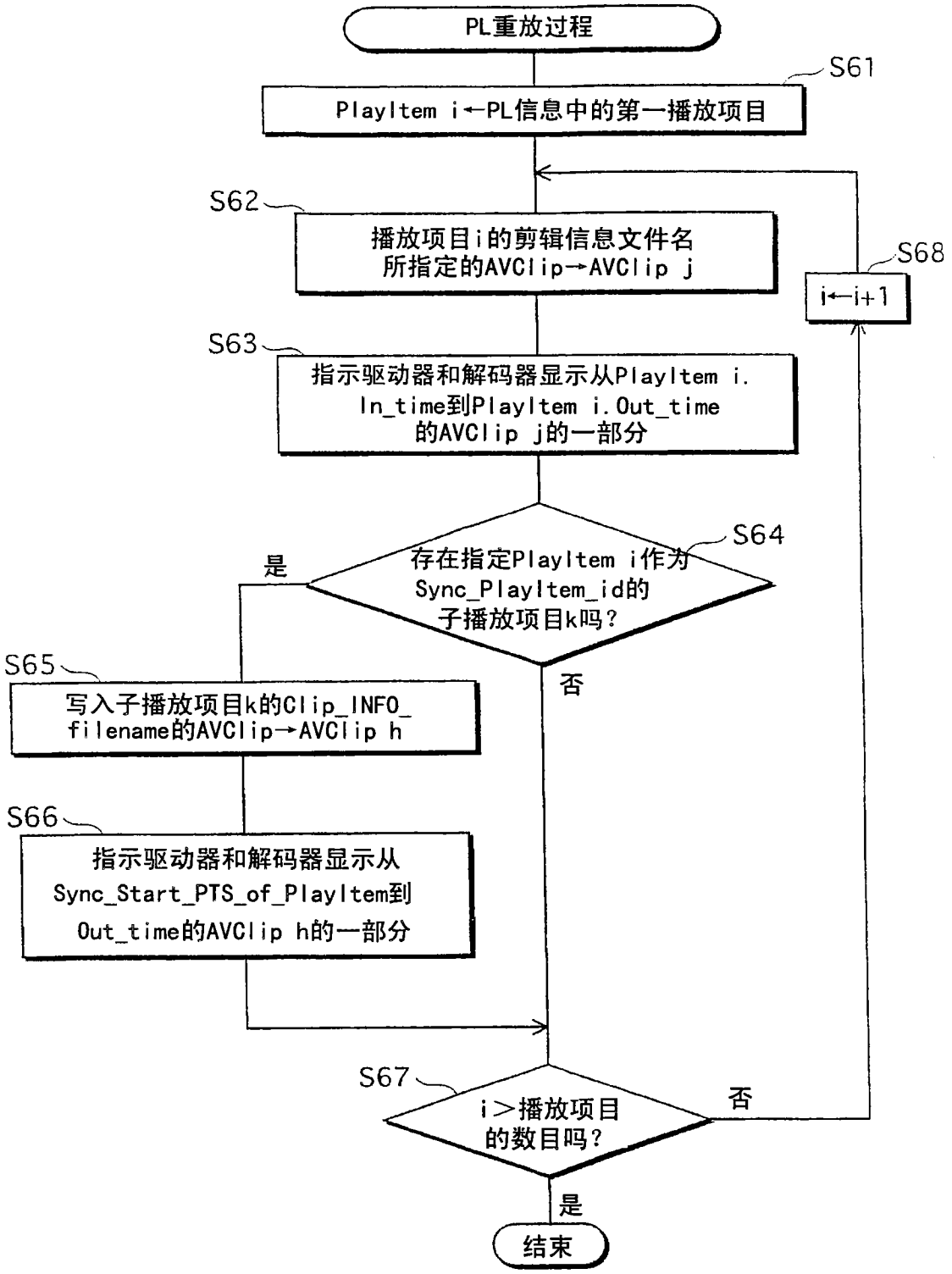

PL重放过程
S61
PlayItem i←PL信息中的第一播放项目
S62
播放项目i的剪辑信息文件名
所指定的AVClip→AVClip j
S68
i←i+1
S63
指示驱动器和解码器显示从PlayItem i.
In_time到PlayItem i.Out_time
的AVClip j的一部分
S64
存在指定PlayItem i作为
Sync_PlayItem_id的
子播放项目k吗?
是
否
S65
写入子播放项目k的Clip_INFO_
filename的AVClip→AVClip h
S66
指示驱动器和解码器显示从
Sync_Start_PTS_of_PlayItem到
Out_time的AVClip h的一部分
S67
i>播放项目
的数目吗?
否
是
结束

图 28

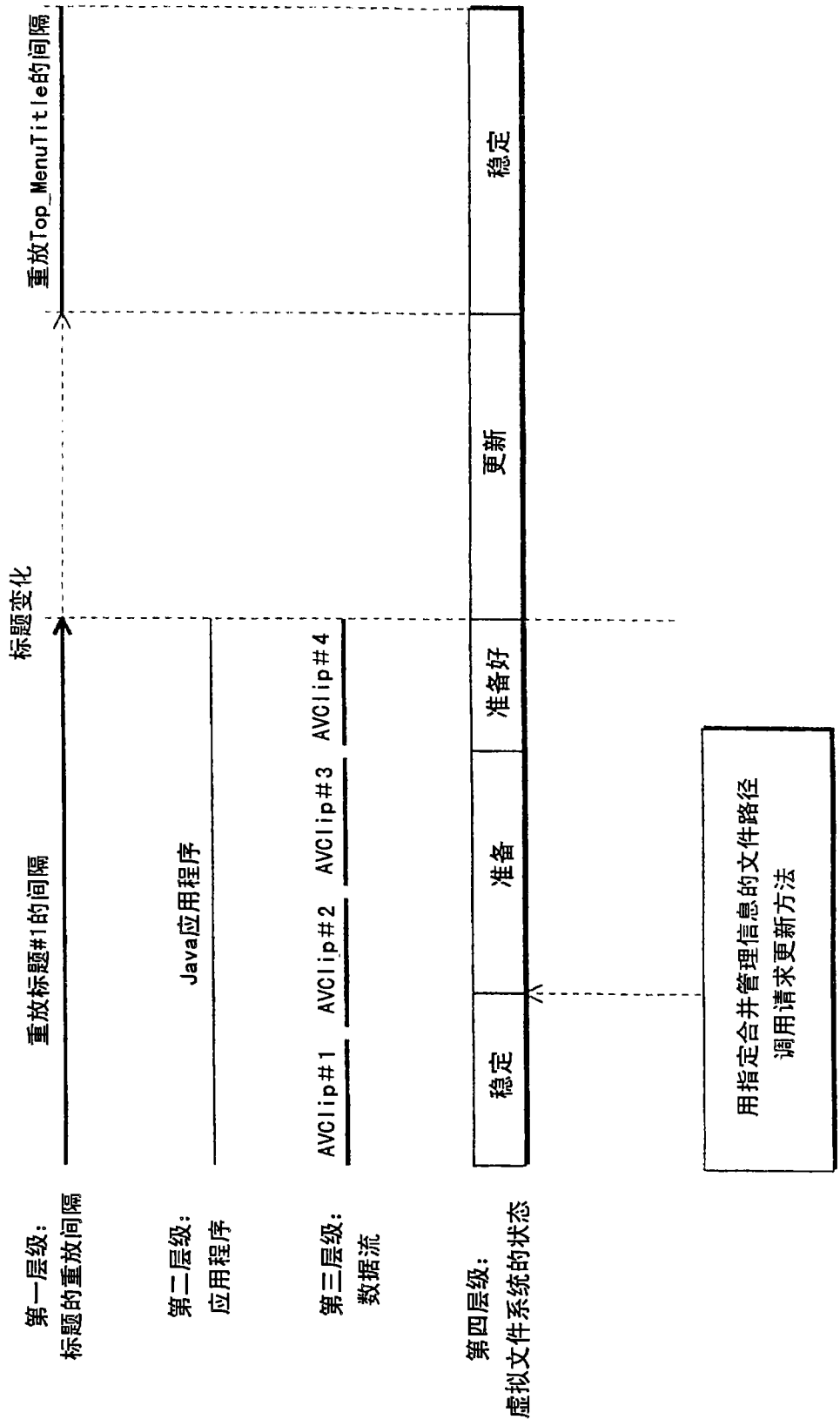

第一层级：
标题的重放间隔
第二层级：
应用程序
第三层级：
数据流
第四层级：
虚拟文件系统的状态
重放标题#1的间隔
标题变化
重放Top_MenuTitle的间隔
Java应用程序
AVClip#1
AVClip#2
AVClip#3
AVClip#4
稳定
准备
准备好
更新
稳定
用指定合并管理信息的文件路径
调用请求更新方法

图 29

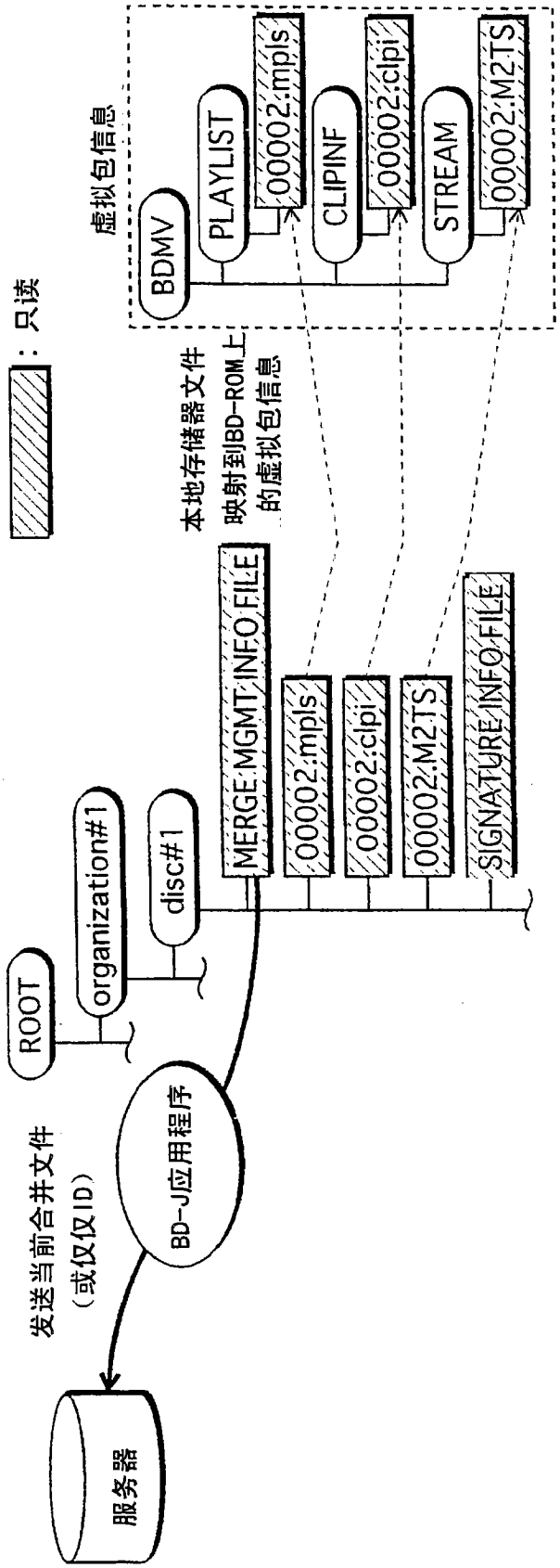

虚拟包信息
BDMV
PLAYLIST
00002:mpls
CLIPINF
00002:clpi
STREAM
00002:M2TS
：只读
本地存储器文件
映射到BD-ROM上
的虚拟包信息
MERGE.MGMT.INFO.FILE
00002:mpls
00002:clpi
00002:M2TS
SIGNATURE.INFO.FILE
ROOT
organization#1
disc#1
BD-J应用程序
发送当前合并文件
（或仅仅ID）
服务器

图 30

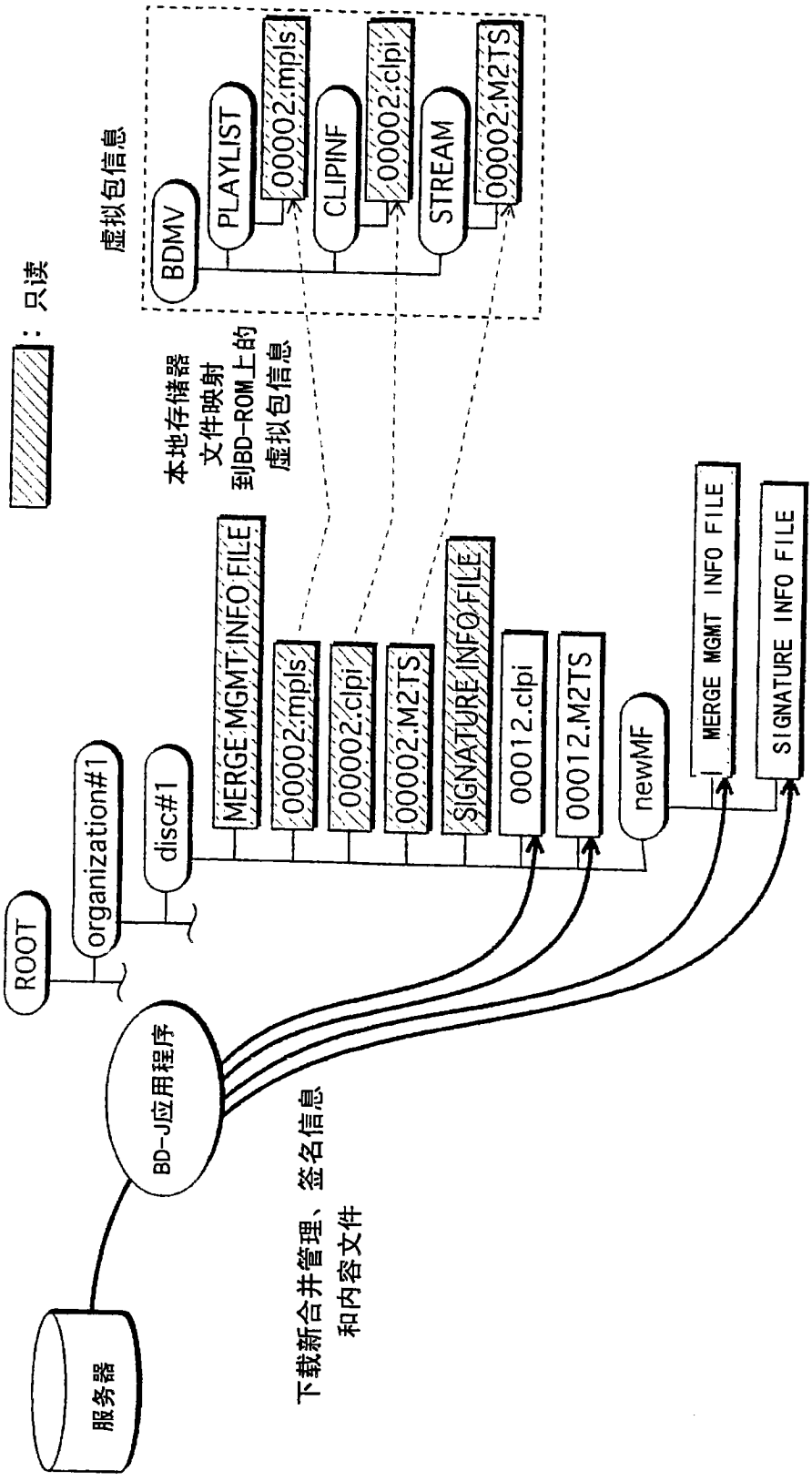

虚拟包信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00002.clpi
STREAM
00002.M2TS
：只读
本地存储器
文件映射
到BD-ROM上的
虚拟包信息
MERGE MGMT INFO FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE INFO FILE
00012.clpi
00012.M2TS
newMF
MERGE MGMT INFO FILE
SIGNATURE INFO FILE
ROOT
organization#1
disc#1
BD-J应用程序
服务器
下载新合并管理、签名信息
和内容文件

图 31

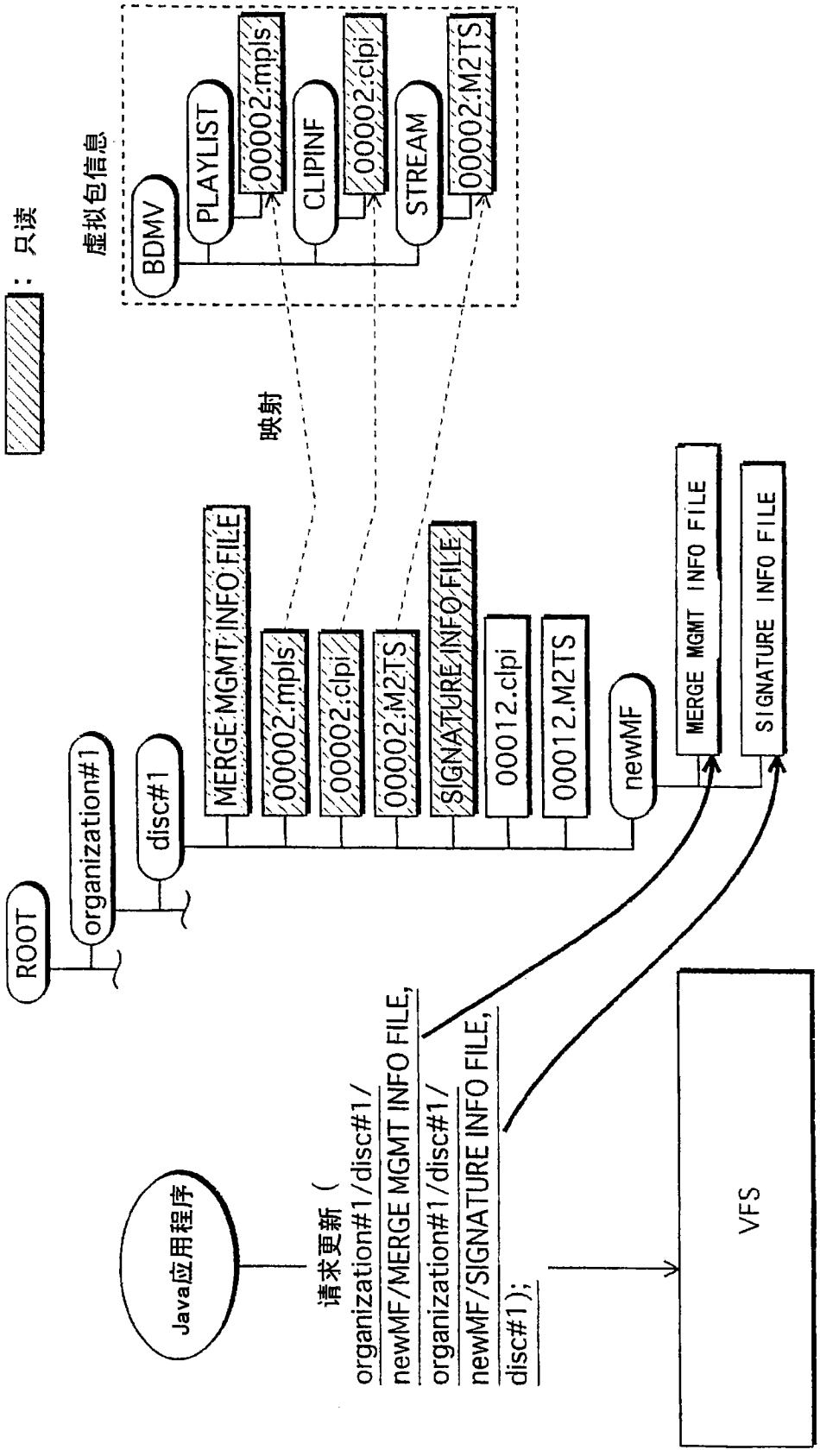

：只读
虚拟包信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00002.clpi
STREAM
00002.M2TS
映射
ROOT
organization#1
disc#1
MERGE MGMT INFO FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE INFO FILE
00012.clpi
00012.M2TS
newMF
MERGE MGMT INFO FILE
SIGNATURE INFO FILE
Java应用程序
请求更新（
organization#1/disc#1/
newMF/MERGE MGMT INFO FILE,
organization#1/disc#1/
newMF/SIGNATURE INFO FILE,
disc#1）;
VFS

图 32

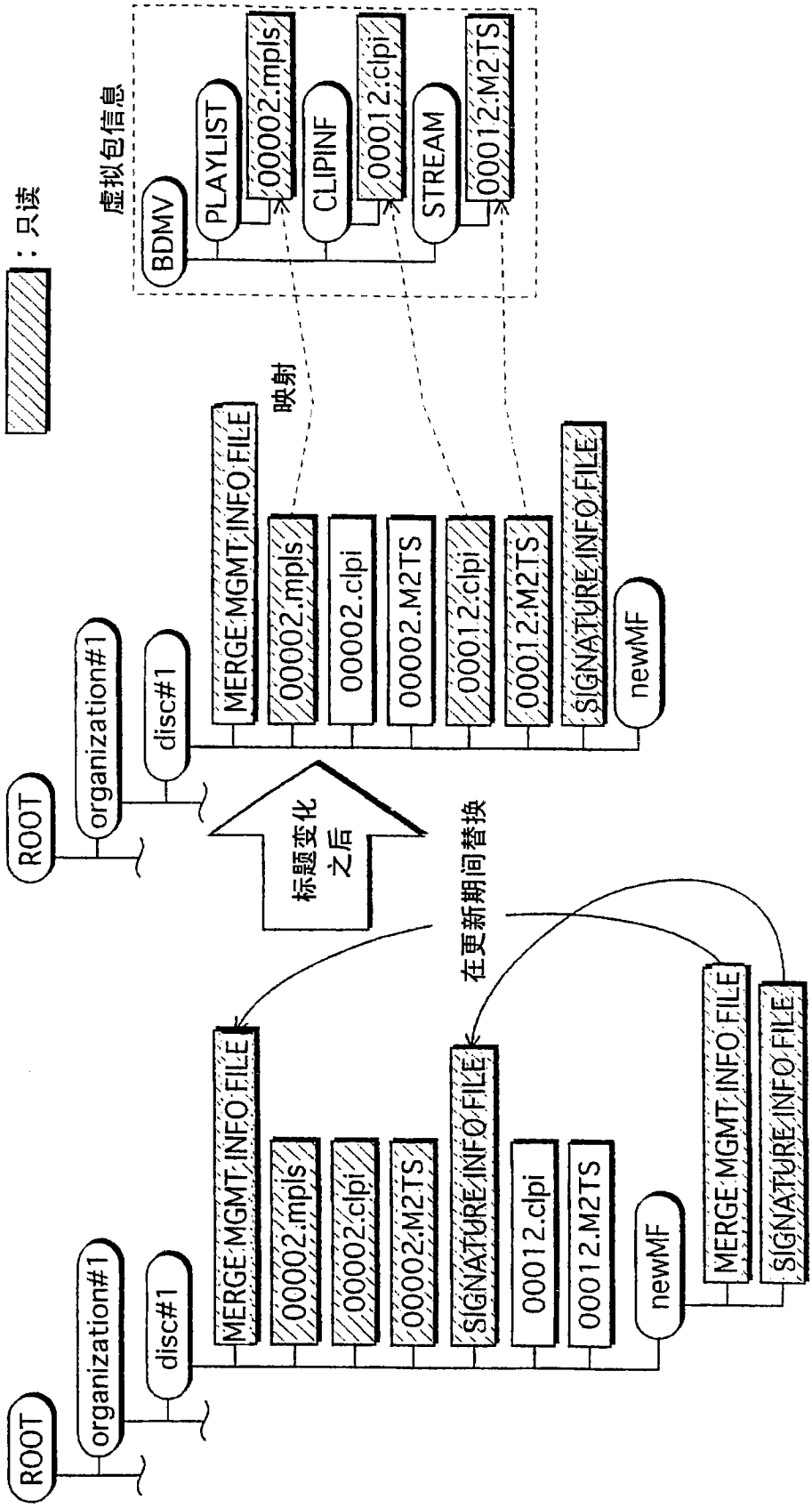

：只读
虚拟包信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00012.clpi
STREAM
00012.M2TS
映射
ROOT
organization#1
disc#1
MERGE.MGMT.INFO.FILE
00002.mpls
00002.clpi
00002.M2TS
00012.clpi
00012.M2TS
SIGNATURE.INFO.FILE
newMF
标题变化之后
在更新期间替换
ROOT
organization#1
disc#1
MERGE.MGMT.INFO.FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE.INFO.FILE
00012.clpi
00012.M2TS
newMF
MERGE.MGMT.INFO.FILE
SIGNATURE.INFO.FILE

图 33

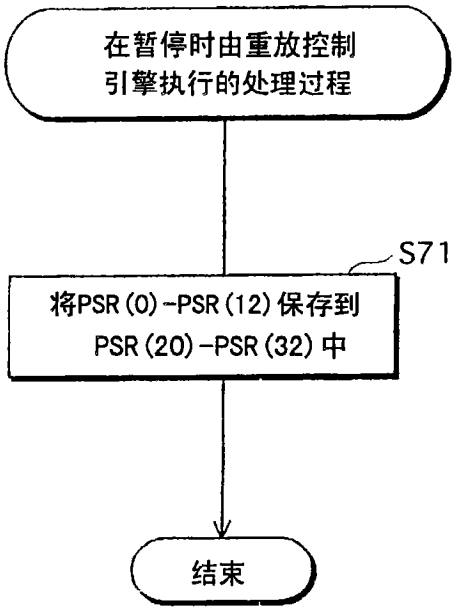

在暂停时由重放控制
引擎执行的处理过程
S71
将PSR(0)-PSR(12)保存到
PSR(20)-PSR(32)中
结束

图 34A

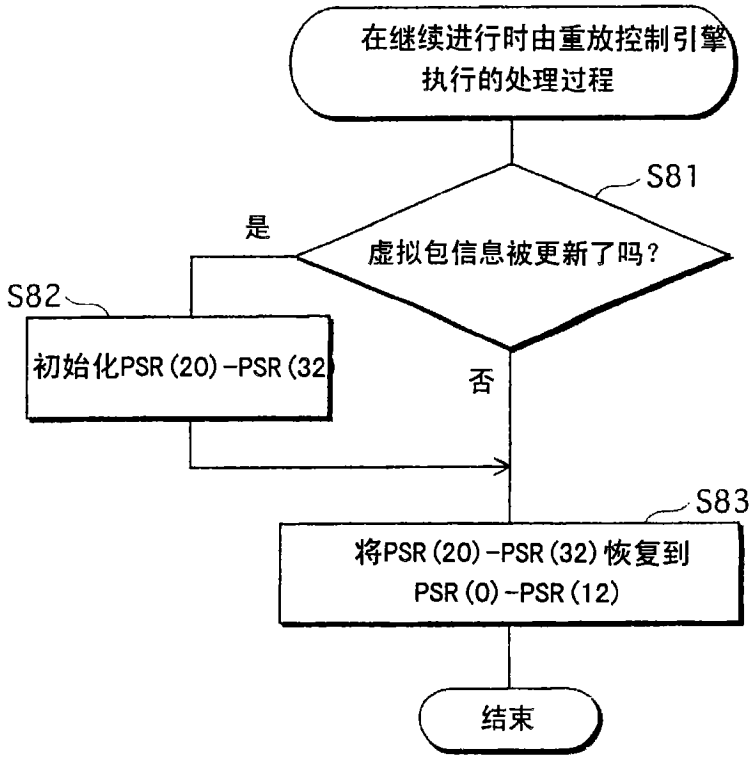

在继续进行时由重放控制引擎
执行的处理过程
S81
虚拟包信息被更新了吗?
是
否
S82
初始化PSR(20)-PSR(32)
S83
将PSR(20)-PSR(32)恢复到
PSR(0)-PSR(12)
结束

图 34B

合并管理信息

光盘ID	合并目标目录	更新日期
disc #1	/organization #1/disc #1/content #1	2004/01/01
disc #1	/organization #1/disc #1/content #2	2004/02/01
disc #1	-	2004/03/01
disc #1	/organization #1/disc #1/content #3	2004/04/01

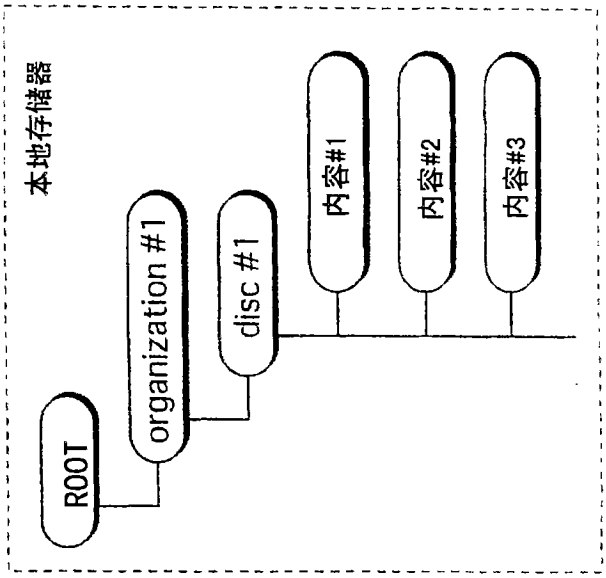


图 35

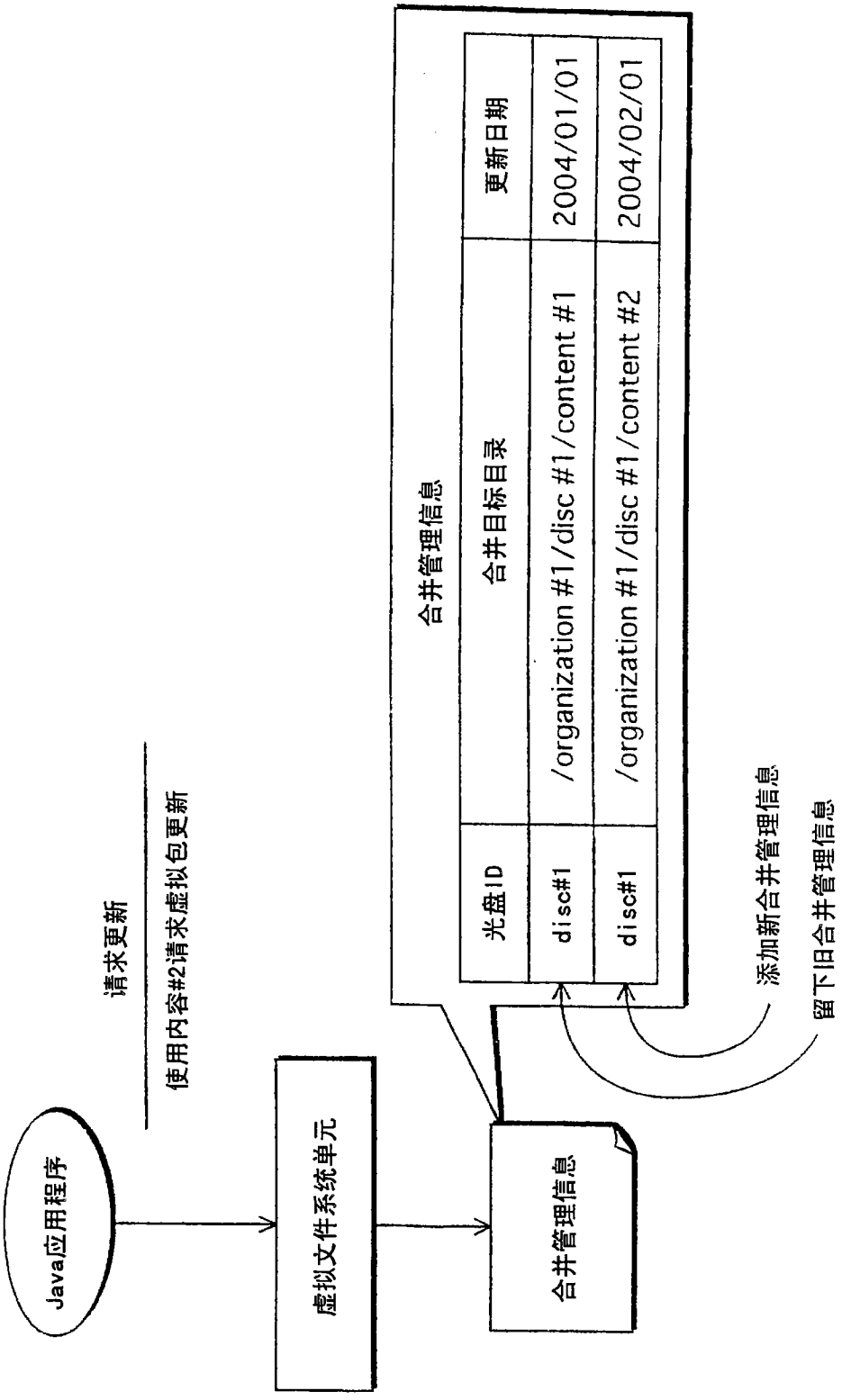

Java应用程序
请求更新
使用内容#2请求虚拟包更新
虚拟文件系统单元
合并管理信息
合并管理信息
光盘ID
合并目标目录
更新日期
disc#1
/organization #1/disc #1/content #1
2004/01/01
disc#1
/organization #1/disc #1/content #2
2004/02/01
添加新合并管理信息
留下旧合并管理信息

图 36

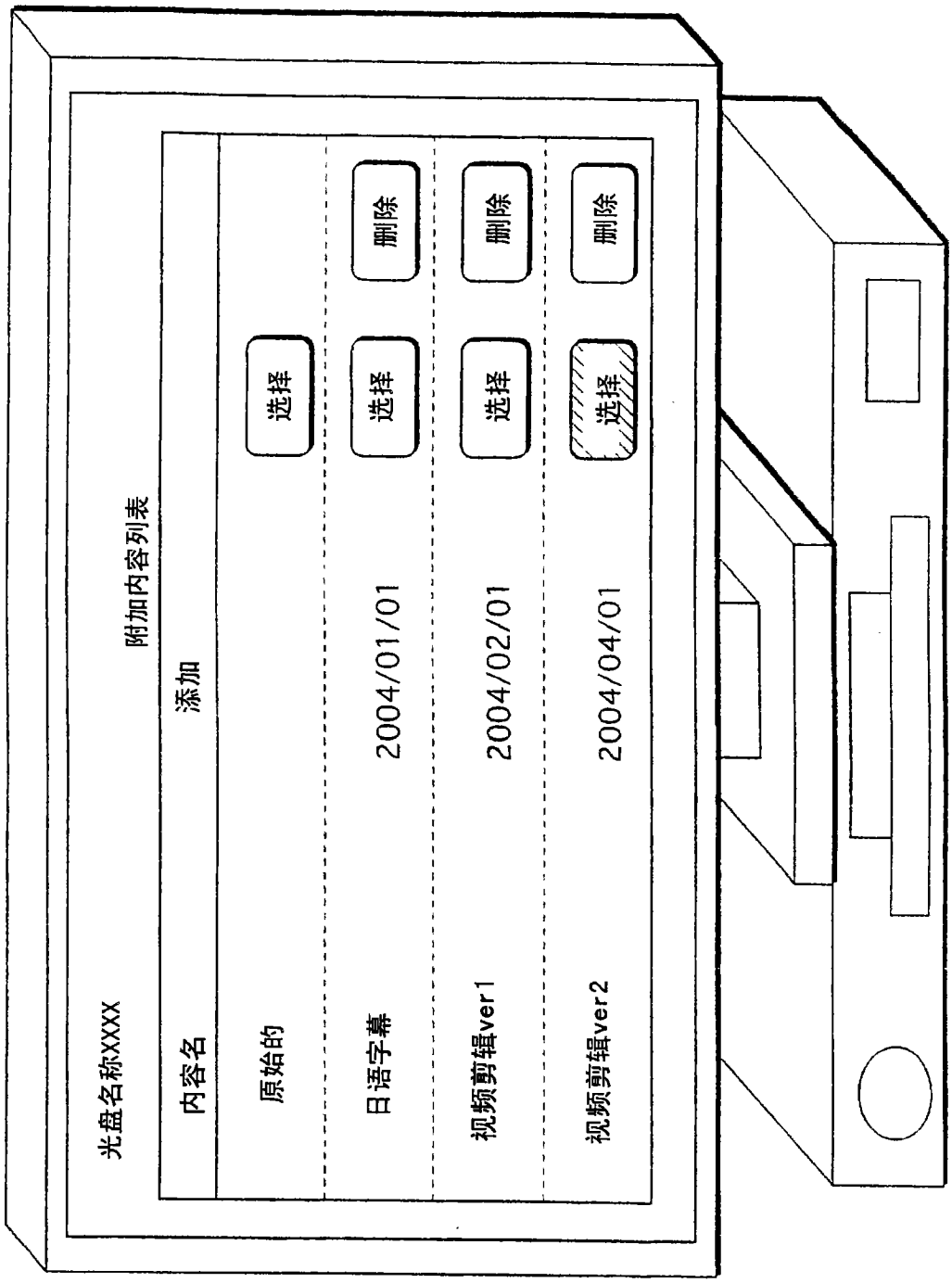

光盘名称XXXX
附加内容列表
添加
内容名
原始的
选择
日语字幕
2004/01/01
选择
删除
视频剪辑ver1
2004/02/01
选择
删除
视频剪辑ver2
2004/04/01
选择
删除

图 37

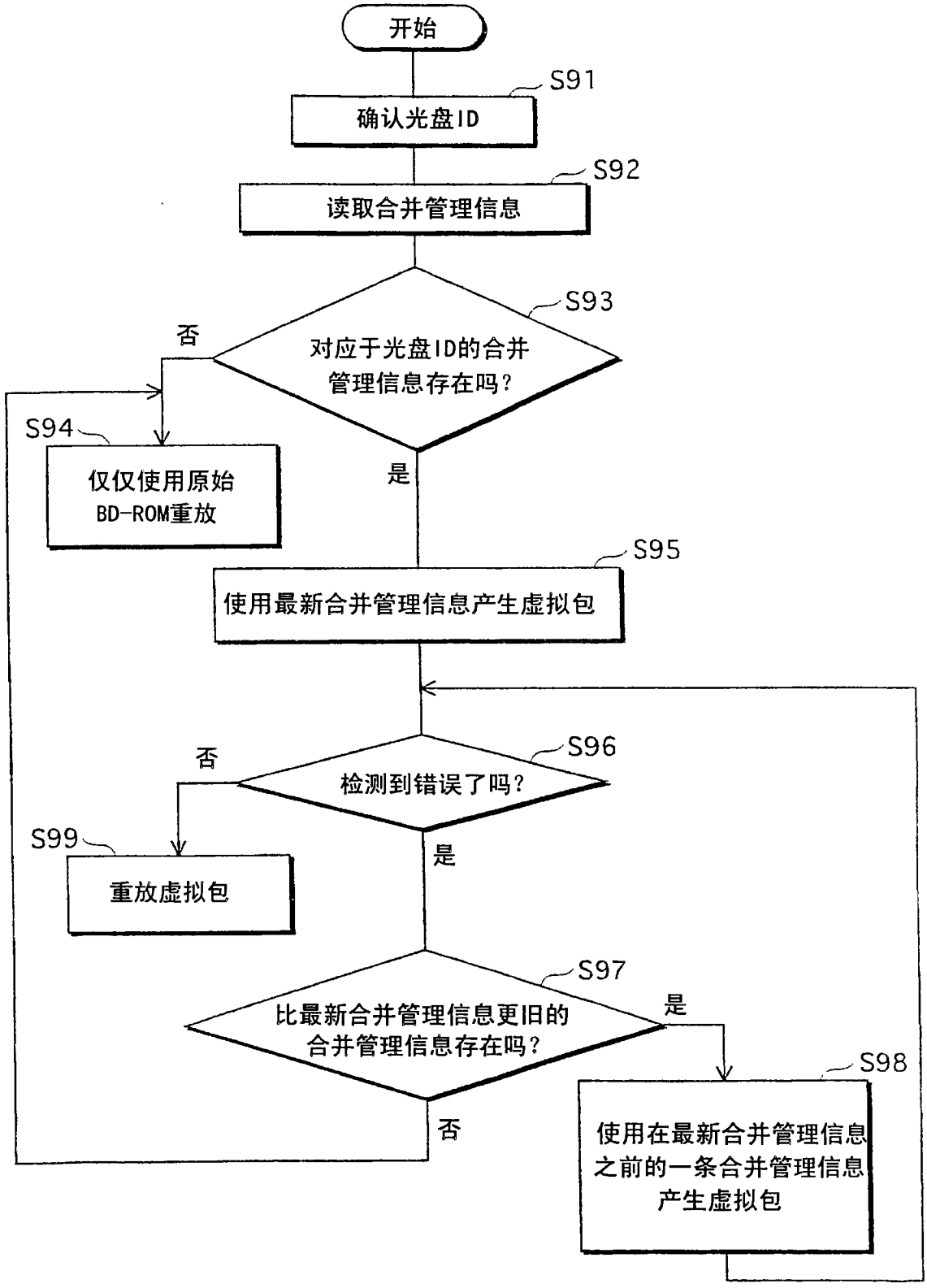

开始
S91
确认光盘ID
S92
读取合并管理信息
S93
对应于光盘ID的合并管理信息存在吗？
否
是
S94
仅仅使用原始BD-ROM重放
S95
使用最新合并管理信息产生虚拟包
S96
检测到错误了吗？
否
是
S99
重放虚拟包
S97
比最新合并管理信息更旧的合并管理信息存在吗？
是
否
S98
使用在最新合并管理信息之前的一条合并管理信息产生虚拟包

图 38

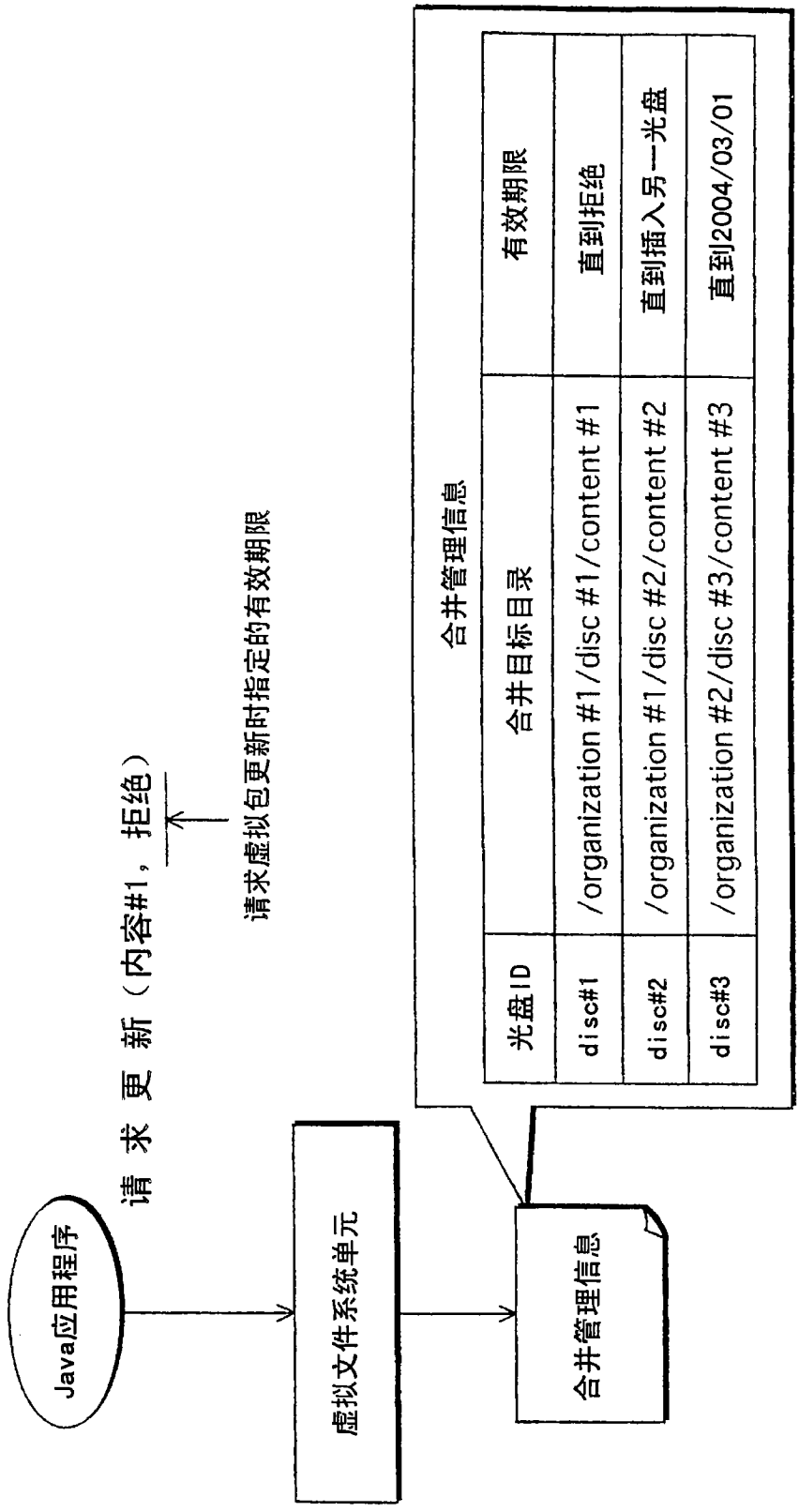

Java应用程序
请求更新（内容#1，拒绝）
请求虚拟包更新时指定的有效期限
虚拟文件系统单元
合并管理信息
合并管理信息
光盘ID
合并目标目录
有效期限
disc#1
/organization #1/disc #1/content #1
直到拒绝
disc#2
/organization #1/disc #2/content #2
直到插入另一光盘
disc#3
/organization #2/disc #3/content #3
直到2004/03/01

图 39

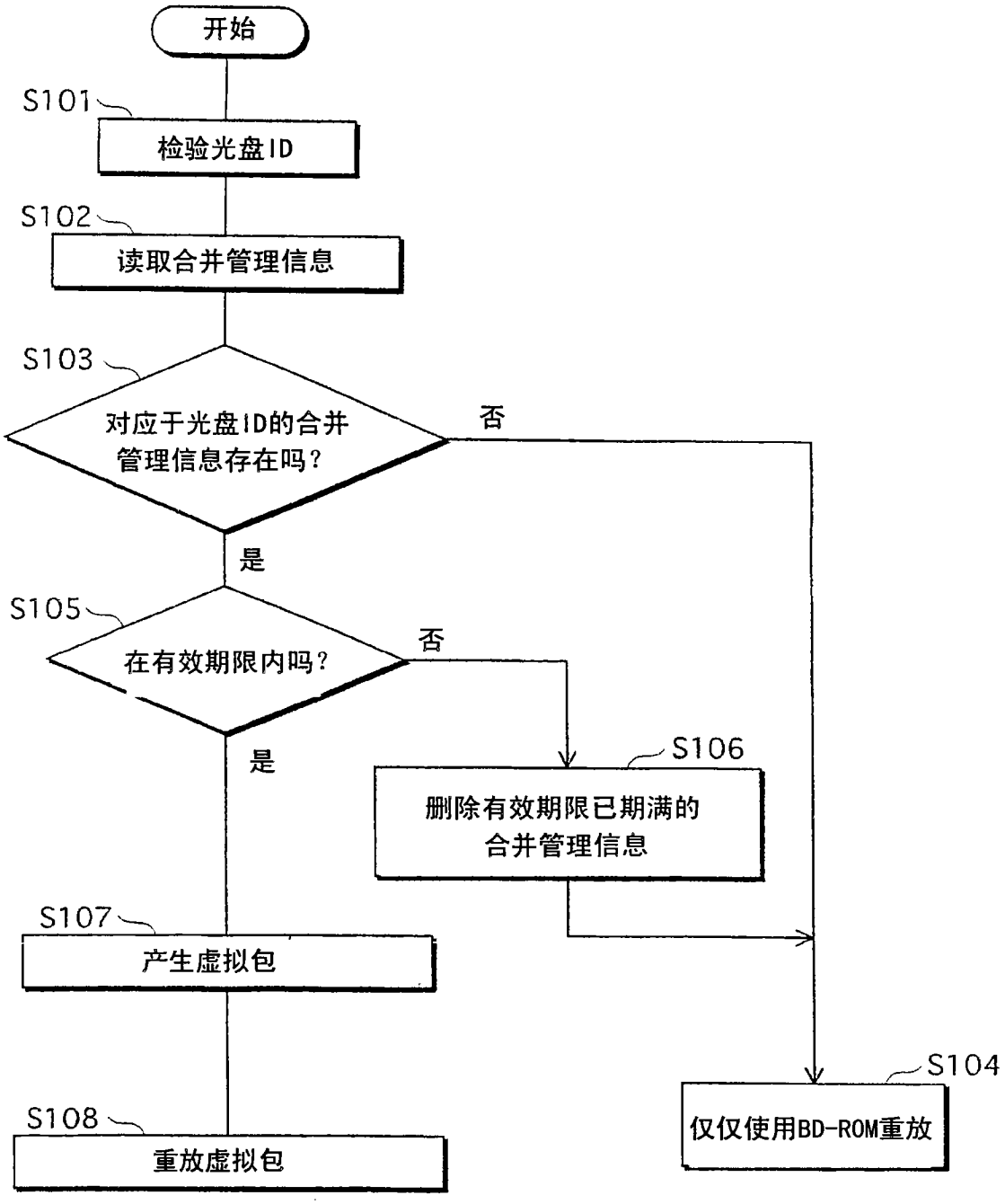

开始
S101
检验光盘ID
S102
读取合并管理信息
S103
对应于光盘ID的合并管理信息存在吗？
否
是
S105
在有效期限内吗？
否
是
S106
删除有效期限已期满的合并管理信息
S107
产生虚拟包
S108
重放虚拟包
S104
仅仅使用BD-ROM重放

图 40

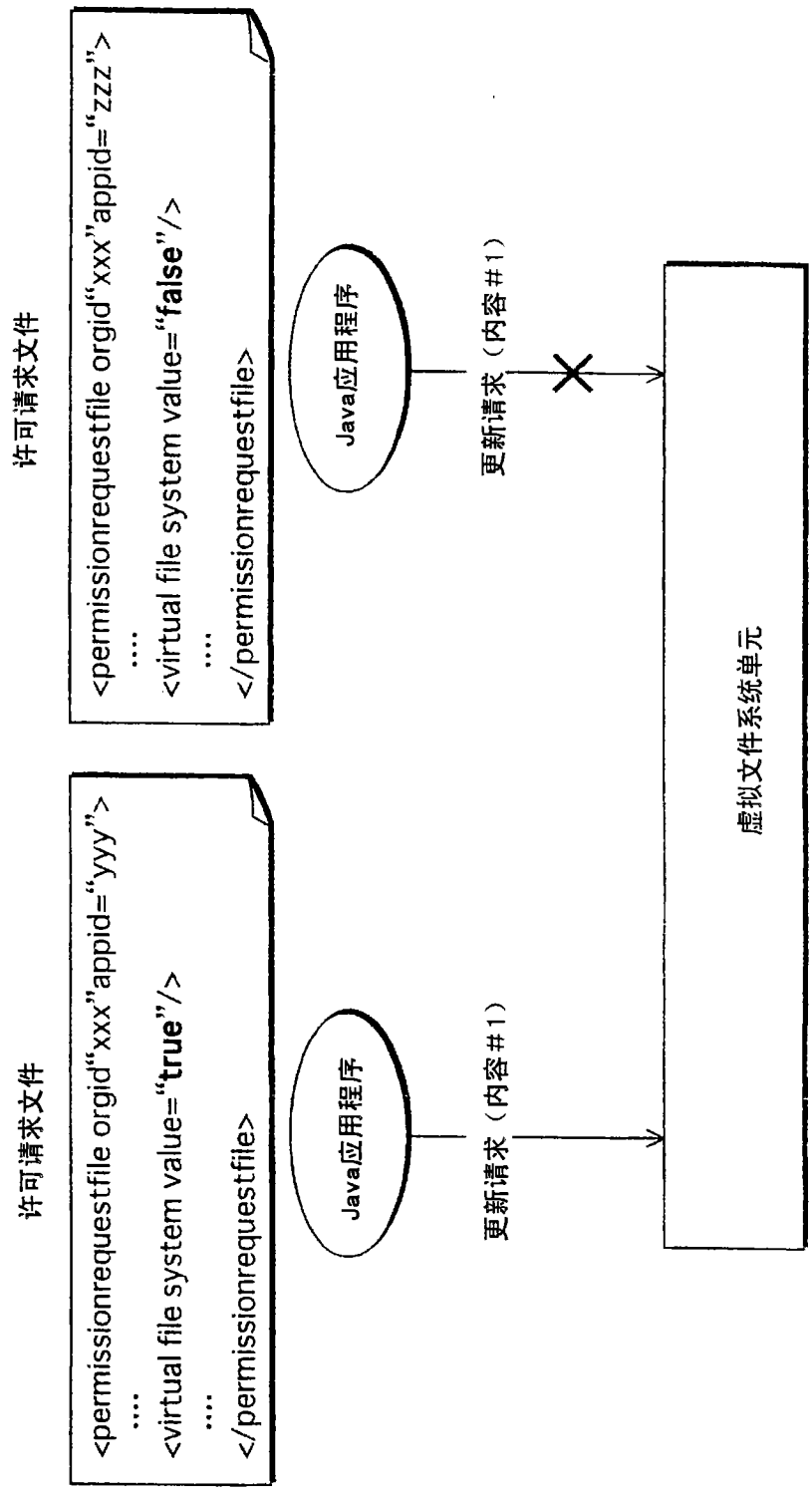

许可请求文件
<permissionrequestfile orgid“xxx”appid=“zzz”>
....
<virtual file system value=“false”/>
....
</permissionrequestfile>
Java应用程序
更新请求（内容#1）
许可请求文件
<permissionrequestfile orgid“xxx”appid=“yyy”>
....
<virtual file system value=“true”/>
....
</permissionrequestfile>
Java应用程序
更新请求（内容#1）
虚拟文件系统单元

图 41

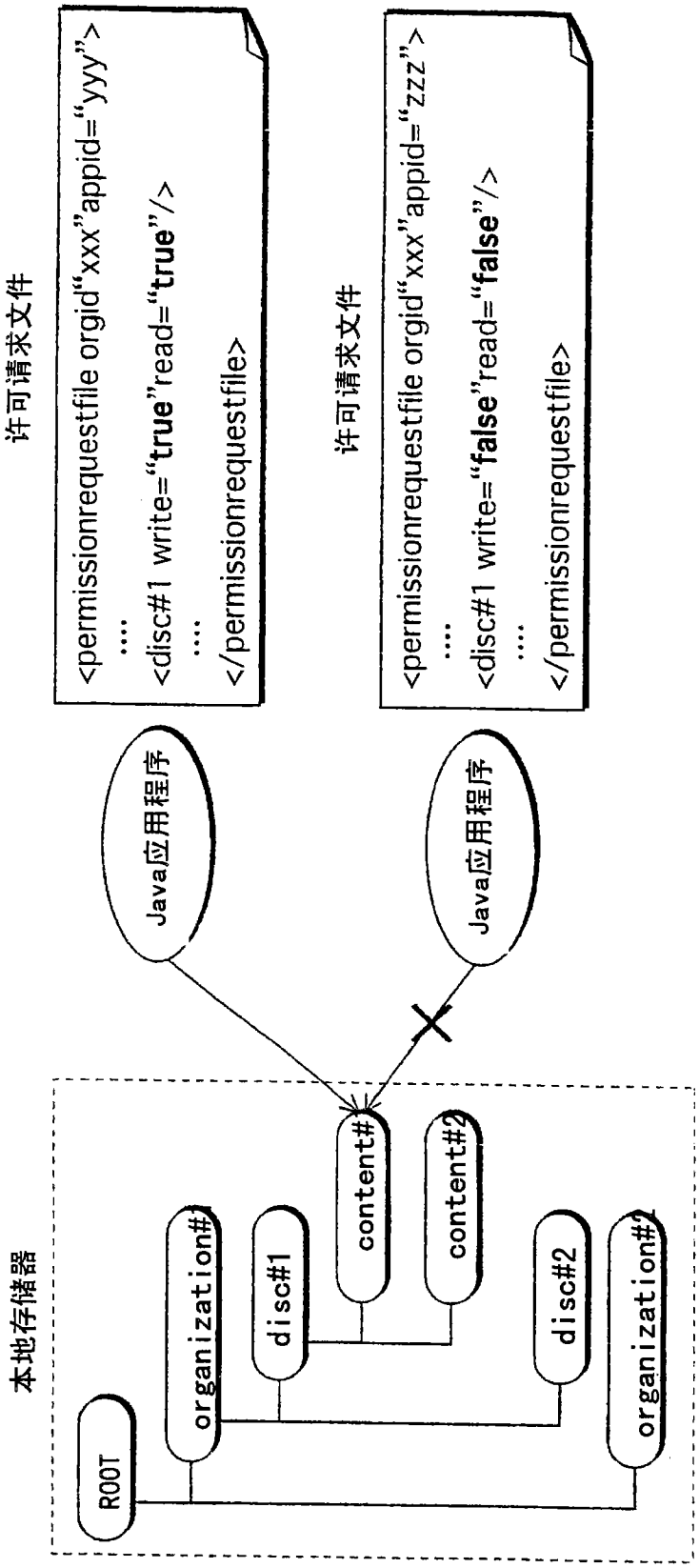

许可请求文件
<permissionrequestfile orgid“xxx”appid=“yyy”>
....
<disc#1 write=“true”read=“true”/>
....
</permissionrequestfile>
许可请求文件
<permissionrequestfile orgid“xxx”appid=“zzz”>
....
<disc#1 write=“false”read=“false”/>
....
</permissionrequestfile>
Java应用程序
Java应用程序
本地存储器
ROOT
organization#
disc#1
content#
content#2
disc#2
organization#2

图 42

应用程序管理信息

标题	应用程序ID	标题绑定属性
标题#1	Java应用程序#1	绑定
	Java应用程序#2	未绑定
标题#2	Java应用程序#2	绑定
	Java应用程序#3	未绑定
标题#3	Java应用程序#3	绑定
	Java应用程序#4	绑定

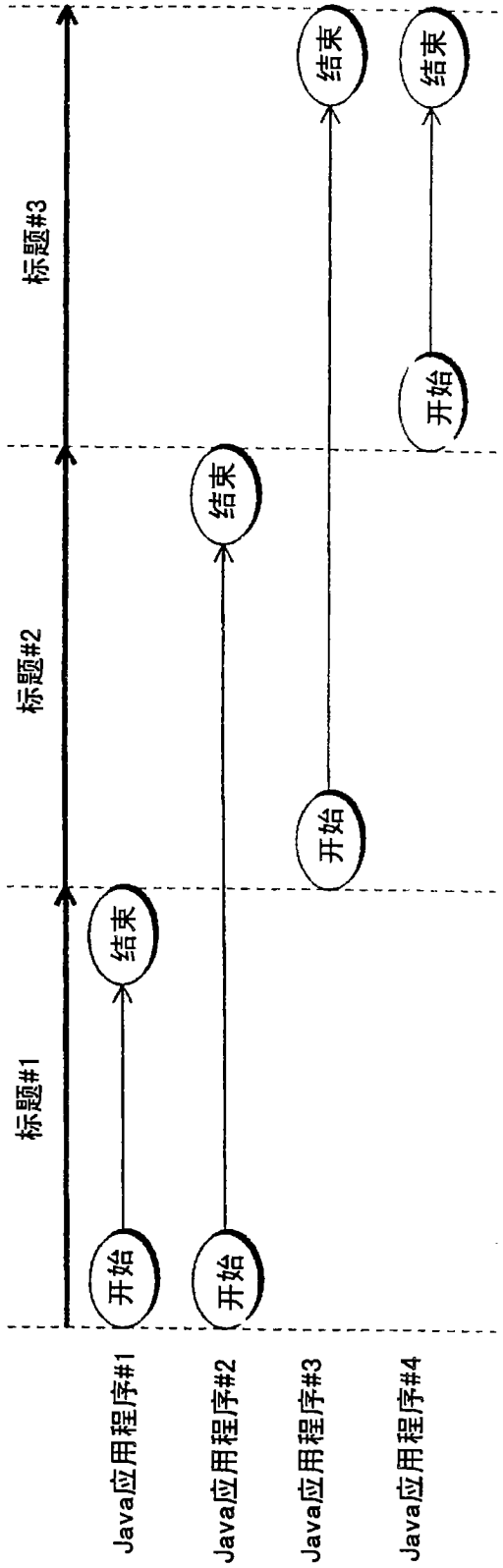


图 43

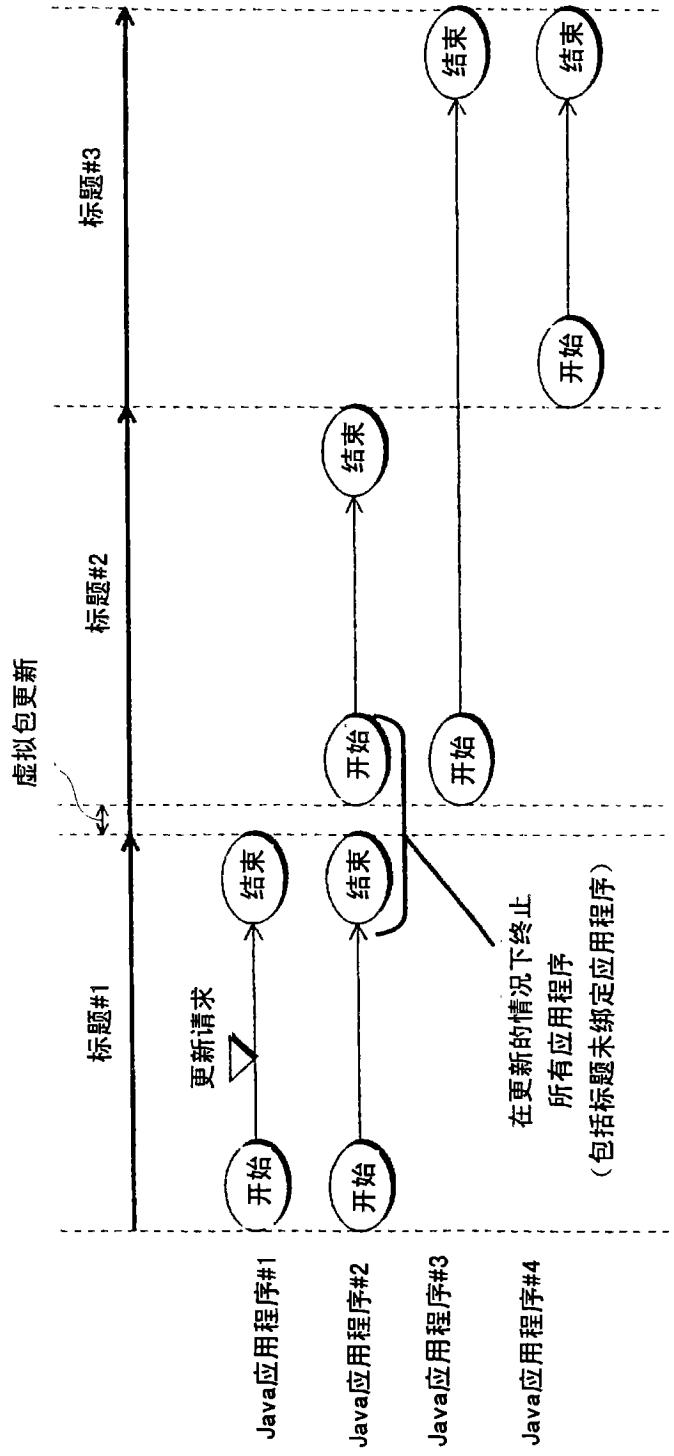

虚拟包更新
标题#1
标题#2
标题#3
更新请求
开始
结束
开始
结束
开始
结束
开始
结束
在更新的情况下终止
所有应用程序
（包括标题未绑定应用程序）
Java应用程序#1
Java应用程序#2
Java应用程序#3
Java应用程序#4

图 44

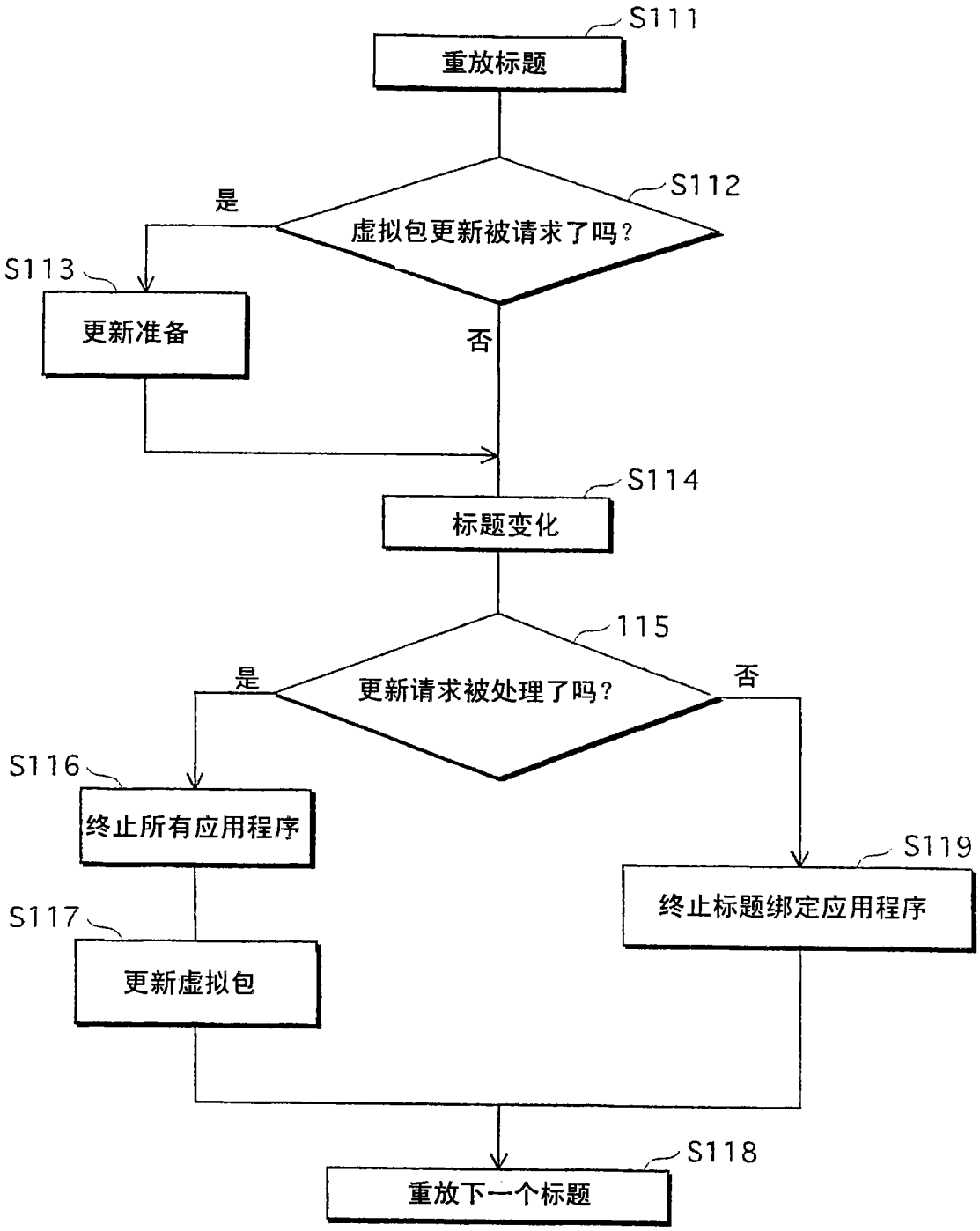

S111
重放标题
S112
虚拟包更新被请求了吗？
是
否
S113
更新准备
S114
标题变化
115
更新请求被处理了吗？
是
否
S116
终止所有应用程序
S117
更新虚拟包
S119
终止标题绑定应用程序
S118
重放下一个标题

图 45

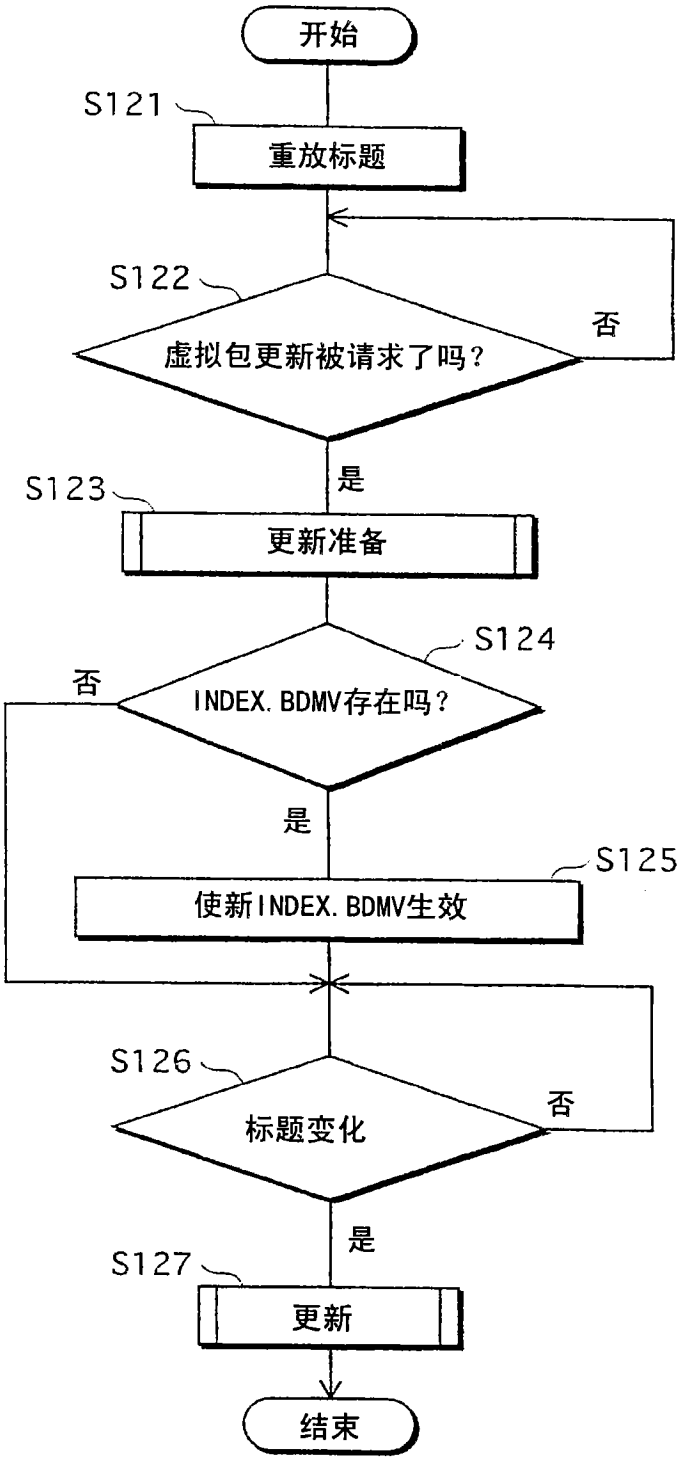

开始
S121
重放标题
S122
虚拟包更新被请求了吗？
否
是
S123
更新准备
S124
否
INDEX.BDMV存在吗？
是
S125
使新INDEX.BDMV生效
S126
标题变化
否
是
S127
更新
结束

图 46