

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/445 (2006.01)



[12] 发明专利说明书

专利号 ZL 200580014775.3

[45] 授权公告日 2008 年 9 月 24 日

[11] 授权公告号 CN 100421074C

[22] 申请日 2005.4.13

[21] 申请号 200580014775.3

[30] 优先权

[32] 2004.5.10 [33] EP [31] 04011063.7

[86] 国际申请 PCT/EP2005/003867 2005.4.13

[87] 国际公布 WO2005/109188 英 2005.11.17

[85] 进入国家阶段日期 2006.11.9

[73] 专利权人 汤姆森许可贸易公司

地址 法国布洛涅

[72] 发明人 约布斯特·霍伦特鲁普

德克·阿道夫 拉夫·奥特曼

卡斯腾·赫佩尔 乌韦·扬森

哈特穆特·彼得斯 安德烈·舍韦措

马尔科·温特

[56] 参考文献

CN1287669A 2001.3.14

US6173246B1 2001.1.9

CN1348185A 2002.5.8

CN1381838A 2002.11.27

Multimedia Systems: Enabling Computer Engineering Education. Wayne Burleson, Santhosh Thampuran, Nirupama Ramaswamy. IEEE. 2002

审查员 李俊

[74] 专利代理机构 中科专利商标代理有限责任公司

代理人 戎志敏

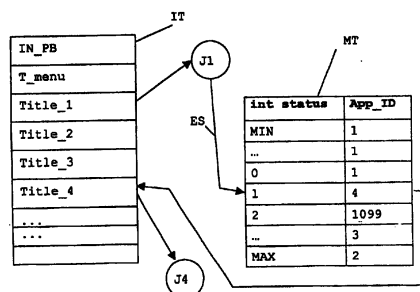
权利要求书 3 页 说明书 10 页 附图 3 页

[54] 发明名称

自动选择软件应用程序的方法

[57] 摘要

一种用于自动地选择作为要在虚拟机上执行的程序的软件应用程序(J4)的方法,包括:从终止的软件应用程序(J1)中接收退出状态值(ES);将所述退出状态值(ES)转变为要执行的下一个软件应用程序(J4)的标识符(App_ID),其中,使用映射表(MT);在所述虚拟机上执行由所述标识符(App_ID)表示的下一个软件应用程序(J4)。



1. 一种用于自动地选择存储在只读存储介质上的标题的方法，所述只读存储介质包括一个或多个标题、以及列出所述标题并且可用于标题选择的索引表（IT），以及包括未在所述索引表（IT）中列出的一个或多个其它标题（J41），其中，标题是包括可再现 AV 数据的软件应用程序、或者在虚拟机上执行的程序，以及，所有标题具有相关联的标题标识符，所述方法包括以下步骤：

- 执行第一标题（J1），其中，所述第一标题（J1）在终止时提供退出状态值（ES）；

- 将退出状态值（ES）转变为标识符（App_ID），所述标识符（App_ID）表示在所述索引表（IT）中的条目（Title_4），或者表示存储在所述存储介质上的标题的标题标识符，其中，为所述转变使用映射表（MT）；

- 确定所述标识符（App_ID）表示索引表条目（Title_4）还是标题标识符；

- 如果所述标识符（App_ID）表示索引表条目（Title_4），则使用所述索引表（IT），将所述标识符（App_ID）转变为标题标识符；以及

- 执行标题（J4），所述标题（J4）由所述标识符（App_ID）、或者由与所述索引表条目（Title_4）相对应的所述标题标识符来表示。

2. 如权利要求 1 所述的方法，其中，从光盘中获取表示所述第一标题（J1）和要执行的下一个标题（J4）的数据，以及从不是所述光盘的可重写存储介质中，获取表示所述映射表（MT）的数据。

3. 如权利要求 1 或 2 所述的方法，其中，所述标识符（App_ID）的值的范围表示所述索引表（IT）中的条目，以及所述标识符（App_ID）的值的第二范围表示存储在所述只读存储介质上的标题之一的标题标识符。

4. 如权利要求 1 或 2 所述的方法，其中，对于所述第一标题的每个可能的退出状态值（ES），所述映射表（MT）包括要执行或再现下

一个标题的有效标识符 (App_ID)。

5. 如权利要求 1 或 2 所述的方法, 其中, 所述只读存储介质包含与不同的标题相关联的多个映射表 (MT_JT、MT_34、MT_MT)。

6. 如权利要求 1 或 2 所述的方法, 其中, 如果下一个标题 (J4) 在所述索引表 (IT) 中没有相应的条目, 则在下一个标题 (J4) 的执行或再现时, 在所述索引表中仍然选择与所述第一标题 (J1) 相对应的所述条目 (Title_1)。

7. 一种用于自动地选择存储在只读存储介质上的标题的设备, 所述只读存储介质包括一个或多个标题、以及列出所述标题并且可用于标题选择的索引表 (IT), 以及包括未在所述索引表 (IT) 中列出的一个或多个其它标题 (J41), 其中, 标题是包括可再现 AV 数据的软件应用程序、或者在虚拟机上执行的程序, 以及, 所有标题具有相关联的标题标识符, 所述设备包括:

- 用于从终止软件应用程序 (J1) 中接收退出状态值 (ES) 的装置;
- 用于将退出状态值 (ES) 转变为标识符 (App_ID) 的装置, 所述标识符 (App_ID) 表示在所述索引表 (IT) 中的条目 (Title_4), 或者表示存储在所述存储介质上的标题的标题标识符, 其中, 为了转变而使用映射表 (MT);

- 用于确定所述标识符 (App_ID) 表示索引表条目 (Title_4) 还是标题标识符的装置;

- 用于在所述标识符 (App_ID) 表示索引表条目时使用所述索引表 (IT), 将所述标识符 (App_ID) 转变为标题标识符的装置; 以及

- 用于执行标题 (J4) 的装置, 所述标题 (J4) 由所述标识符 (App_ID)、或者由与所述索引表条目 (Title_4) 相对应的所述标题标识符来表示。

8. 如权利要求 7 所述的设备, 其中, 从光盘中获取表示所述第一标题 (J1) 和要执行的下一个标题 (J4) 的数据, 以及从不是所述光盘的可重写存储介质中, 获取表示所述映射表 (MT) 的数据。

9. 如权利要求 7 或 8 所述的设备, 其中, 所述标识符 (App_ID) 的值的范围表示所述索引表 (IT) 中的条目, 以及所述标识符

(App_ID)的值的第二范围表示存储在所述只读存储介质上的标题之一的标题标识符。

自动选择软件应用程序的方法

技术领域

本发明涉及一种自动选择软件应用程序的方法。具体地，本发明涉及一种 Java 应用程序的生命周期管理。

背景技术

例如数字通用盘（DVD）的预先录制的光存储介质上的多媒体数据可以属于不同的应用程序，例如，不同的视频标题、游戏等。通常将这些应用程序嵌入介质的程序设计结构中，作为可以独立启动的应用程序。

然而，期望预先录制的光介质具有良好定义的内容外观和用户体验，例如交互性、以及复杂的创作可能性。由于 Java 技术独立于底层硬件和操作系统，并且提供了包括例如图形和网络接口的完备的编程环境，因而可以满足这些需求。Java 技术使用被称为 Java 虚拟机（JVM）的执行单元，来执行所谓的字节码，所述字节码从称为类文件的文件中获得。可以在程序库或程序包中组织类文件。JVM 被规定为抽象机，并且可以理解为需要执行处理单元硬件的独立软件层。

在消费电子设备中存在用于 Java 运行时间环境的不同的配置和简档，其中，每个简档适用于特定类型的设备。不同简档之间的主要差别在于它们支持哪个程序库。例如，像“个人基本简档”和“个人简档”之类的简档更加适用于消费电子设备。

在经典的应用程序模型中，应用程序处于激活或未激活（inactive）状态。除了该基础模型之外，所述简档还定义了被称为“Xlet”应用程序模型的增强型应用程序模型，该程序模型允许对应用程序生命周期的精密控制。系统包括“应用程序管理器”（AM），这是通过特定接口来控制所有运行的应用程序生命周期的软件实体。例如，AM 可以将应用程序加载至 JVM 中，启动及停止该应用程序。

因此，应用程序可以位于多个生命周期状态（例如，加载、启动或停止）中的一个状态中的任何时间点。应用程序可以自己改变生命周期状态。AM 提供了应用程序管理接口（AMI），通过 AMI，应用程序发送状态或状态改变的信号。因此，AM 在任何时刻都具有关于所有应用程序生命周期状态的更新的信息。

发明内容

本发明解决的问题是将Java集成于现有系统中。大体上，为了提高AM与应用程序之间交互的灵活性，需要定义了何时和怎样启动应用程序、以及当应用程序终止或被终止时会发生什么的生命周期模型，即，定义了系统行为的模型。

本发明是基于为存储在存储介质（例如，预先录制的存储介质）上的每个应用程序，来使用映射表，映射表可以存储在相同或另一存储介质上，以及用于将各个应用程序的应用程序退出状态信息（应用程序专用的）转换为表示标题或Java应用程序的标识符，该标题或Java应用程序也被存储在介质（例如，相同的介质）上，因而是盘专用的，以及下一步将会执行该标题或Java应用程序。因此，当Java应用程序结束时，播放器可以将结束应用程序的退出状态信息提供给相应映射表、读取映射表返回值并继续从表中得到的特定实体。

对于软件应用程序来说，提供状态信息的通常方式是，使用退出代码来进行终止，退出代码可以由上层的或者接下来的软件层进行分析。

本发明允许对于存储在存储介质上的一个、多于一个、或者所有应用程序来使用特定的映射表。即，在本发明的一个实施例中，存储介质包含与其保存的Java应用程序一样多的映射表，而在另一实施例中，存储介质包含较少的、但是至少一个映射表，其中，由存储在介质上的多个或所有应用程序来共享所述映射表。根据本发明，存储介质还可以仅对于其保存的一些、但不是所有的应用程序，包含一个或多个映射表。

本发明构建了灵活且鲁棒的方法，通过该方法，在盘的重放期间，

尤其是在重放设备上的Java应用程序的生命周期结束的情况下，例如多媒体盘的作者（author）可以控制重放设备的系统行为。因此，对于盘的作者来说，可以在盘上预先定义不同的可独立获得的应用程序的灵活的序列，因而还能够改进在运行期间对输入（例如，用户输入、来自播放器（例如，计时单元）的输入等）的反应（reaction）。

在权利要求7中公开了利用该方法的设备。

将在所附权利要求、以下的描述及附图中公开本发明的优选实施例。

附图说明

参照附图，对本发明的示例性实施例进行描述，其中

图1是存储在多媒体盘上的重放数据的结构以及相关的Java应用程序；

图2是用于将退出状态代码映射至下一标题标识符的映射表的使用；

图3是用于将退出状态代码映射至下一应用程序标识符的映射表的使用；

图4是Java标题菜单的结构；以及

图5是不同应用程序共享映射表的示例性映射表。

具体实施方式

图1描述了存储在预先录制的存储介质（例如，DVD或蓝光盘）上的多媒体数据和菜单数据的结构。当将存储介质插入播放器（即，再现设备）时，播放器搜索并检测索引表IT。接着，播放器检测索引表的第一条目IN_PB是Java应用程序JF。例如，在第一条目IN_PB是具有指示Java类文件的扩展的文件名时，这可以被检测到。然后，播放器将建立Java应用程序管理器（AM）和Java虚拟机（JVM），来执行特定文件的程序。例如，特定的应用程序JF开始诸如简介序列之类的特定视听（AV）序列或电影对象的重放。

当AV序列结束时，Java应用程序JF终止，以及播放器继续在索

引表 IT 中步进。下一个表条目 T_menu 包含至另一 Java 应用程序 JT 的链接，在 JVM 中执行该 Java 应用程序 JT，导致显示最高级菜单和背景视频序列。可以显示直至用户选择项目为止。

进行以下操作是有用的：测量用户未给出输入的时间，以及在特定时间之后，自动地启动应用程序，例如，索引表条目之一或者附加的动画，随后返回最高级菜单。

当选择项目时，自动地终止运行应用程序 JT，并返回退出值。播放器可以使用该退出值来识别要启动的下一个应用程序，例如特定标题 Title_1。根据本发明，将映射表用于要启动的下一个应用程序的识别。映射表提供了应用程序返回值与要启动的下一个应用程序的标识符之间的链接。

使用这样的映射表的优点在于，对于每个应用程序，可以提供一个或多个后继应用程序。因此，内容作者可以在应用程序终止之后，在保存了多个标题的盘上定义灵活的标题序列，而不是定义固定序列或返回主菜单。例如，当重放特定选择标题 Title_1 的应用程序 J1 终止时，其返回可以映射至启动另一个 Java 应用程序 J3 的另一个标题 Title_3、或者映射至可能不可以从最高级菜单中选择的第三 Java 应用程序 J5。在这种情况下，由于未从索引表 IT 中选择新的标题，所以先前激活的标题保持激活。先前选择的标题，即索引表条目，是激活的。这样的索引表条目可以是与 Java 应用程序的链接，也可以是与另一程序的链接。

根据本发明的映射表是环境（context）和应用程序专用的，即，对于当前的环境，例如编辑，每个应用程序可以具有自己的映射表。然而，在许多情况下，不同的应用程序可以共享相同的映射表。由映射表设置的标识符可以是数字值或字母数字值，例如，标题或与应用程序相关联的应用程序标识符（APPL_ID），并且通过该标识符，可以调用该应用程序。标识符还可以是特定类型或类的参数，例如文件名。可以通过使用与每个 Java 应用程序相关联的应用程序标识符（APPL_ID），来将一个或多个 Java 应用程序分配给标题。

在图 2 中，使用映射表 MT 的应用程序管理器从终止 Java 应用程

序 J1 的退出状态代码 ES 中，确定指向索引表 IT 的条目 Title_4 的标识符，Title_4 指定要在 JVM 上执行的另一 Java 应用程序 J4。映射表 MT 为对于其是有效的应用程序 J1 的每个可能的退出状态代码 ES 提供了要执行的下一个应用程序的有效标识符 App_ID。由于应用程序返回值的数据类型是已知的、从而比特宽度是已知的，所以这是可能的。图 2 的映射表 MT 包括针对最小值 MIN 的条目、针对最小值与 0 之间的值的一个条目、针对退出状态代码 0、1 和 2 的每个的一个条目、针对大于 2 但小于最大值的退出状态代码的一个条目、以及最后地，针对最大值的条目。如在描述的示例中的，退出代码值可以是四比特宽，因而可以编译为带符号整数值 -8、...、7。返回值还可以是无符号整数类型、或任何其它类型。

通常，有两种终止 Java 应用程序的方式：由应用程序管理器发起终止、或者由应用程序自身发起终止。对于从存储介质中获取的应用程序，在前一种情况中，这意味着用户定义了怎样继续进行，例如标题搜索，而在后一种情况中，介质的作者定义了下一步将发生什么。因此，本发明为内容作者提供了灵活的方式，以使得能够在应用程序终止时提供进行至何处的信息。

通常，映射表是数据结构，用于将输入值与输出值相关联，即，将盘上的一个程序的返回值与盘上另一应用程序的标识符相关联。在 Java 应用程序结束的情况下，映射表用于指定系统行为，以及由应用程序管理器（AM）进行处理。在表 1 中示出了概念上的映射表的示例。

退出状态	标识符	request_resume_flag
Integer.MIN_VALUE	Title 1	0
...	...	...
0	Title 1	1
1	应用程序 98765	0
...	...	...
Integer.MAX_VALUE	Title 2	0

表 1：示例性概念映射表

表中的第一列定义了可以从应用程序中传递的退出状态值。在本发明的优选实施例中，这是整数。该表覆盖了应用程序可能返回的所有值，从最小值至最大值。在另一实施例中，给出了缺省值而非最小和最大值。例如，如果终止应用程序不返回值、或者返回值与其它表条目不匹配，则也可以使用该缺省值。

第二列指定了表示标题或 Java 应用程序的特定标识符。目前的预先录制的光盘在逻辑上被分为所谓的标题，其中，每个标题具有相关联的标题号。本发明的映射表允许在“标识符”列中指定标题号。这允许预先定义、或者创作出至另一标题的转变。该转变可以包括播放器的“模式改变”，其中，通过不同于 Java 的已知方式（例如，定义了预命令、AV 重放命令和后命令的脚本）来创作出标题。该表还允许在标识符列中指定 Java 应用程序，而不是标题。在这种情况下，播放器可以在没有标题或模式的转变的情况下，在当前标题中启动指定的应用程序。

此外，在转变为另一标题的情况下，播放器可以在该标题的运行时间内保存恢复信息。如果作者想恢复该标题，则请求播放器估计存储的恢复信息，这可以通过在表的相关域中设置“request_resume_flag”来表示。例如，这可以用来控制被多次调用的应用程序的不同行为或不同返回值。

图 3 示出了使用退出状态代码 ES “2” 来终止的应用程序 J1。AM 在映射表中查找该值，并在相关的条目中发现相关联的应用程序标识符 App_ID “99”，AM 将该标识符 App_ID “99” 识别为 Java 应用程序 J99 的标识符。AM 将指定的 Java 应用程序 J99 加载至执行该应用程序 J99 的 JVM。在这种情况下，还可以直接从索引表 IT 中启动指定的 Java 应用程序 J99。当应用程序 J99 终止时，AM 接收返回值 ES99，其在映射表 MT 中查找 ES99，以确定要执行的下一个应用程序。

图 4 示出了可以怎样处理用户输入。当用户选择了菜单项时，执行最高级菜单的 Java 应用程序 JM 返回退出状态值。依据用户通过输入设备（例如，遥控器）选择哪个菜单项，应用程序 JM 具有不同的退出状态值。使用上述本发明的映射表，将退出状态值映射至索引表

IT 的其它条目中的一个。所选索引表条目之一 Title_2 可以由 AM 控制、但不使用 JVM，来启动例如电影的重放，而另一个索引表条目 Title_3 在 JVM 上启动另一个 Java 应用程序 JT，该应用程序 JT 然后重放另一个视频标题。

还可以由播放器或用户输入之外的其它因素（例如，计时单元、伪随机发生器、或类似于个人计算机的另一个连接的设备）来影响应用程序返回的退出状态值。

图 5 示例性地描述了具有四个标题 Title_1、...、Title_4 的存储介质的小菜单。当播放器读取存储介质时，其估计索引表 IT，以及 AM 在条目中步进。第一条目 FirstPlayback 包含被载入 JVM 并执行的 Java 应用程序 JF 的标识符。在终止时，该应用程序不返回退出代码值，以及没有映射表与该应用程序相关联。因此，AM 将继续在索引表 IT 中步进，以及针对下一个条目，发现另一个 Java 应用程序 JT 的标识符 T_menu。该应用程序也被载入 JVM 并执行，导致为用户显示菜单。但是该应用程序 JT 仅在用户输入时终止，或者如果用户不提供输入，则在预定时间之后自动终止。计时单元可以对此进行控制。例如，当用户从菜单中选择 Title_3 时，运行的应用程序 JT 将退出代码值“3”返回 AM。AM 将使用与终止的应用程序相关联的映射表 MT_JT，来将退出代码值“3”映射至应用程序或标题标识符“103”，例如，“103”指向相应的索引表条目 Title_3。如果用户选择另一个菜单条目，则菜单应用程序 JT 的返回值可以指向另一个索引表条目，或者直接指向应用程序 J41。所选索引表条目 Title_3 表示另一个 Java 应用程序 J3，J3 然后被载入 JVM 并执行，例如播放电影或游戏。

在终止该应用程序时，其将退出状态值返回 AM，AM 使用例如对于两个不同的应用程序 J3 和 J4 有效的映射表 MT_J4。实际上，可以对于另一个存储介质、介质的另一个版本等，来对 Java 应用程序 J3 进行编程，以及可以返回 2 和 7 之间的退出代码值。然而，作者可以决定，对于该版本的介质，应当忽略来自应用程序 J3 的退出代码值 2、6 和 7，而在另一版本中，这些代码可以具有另一功能。因此，对于另一个应用程序 J4，可以使用相同的映射表 MT_J4。

通常，如果具有相同的相关返回值，则两个或多个应用程序可以共享映射表。因此，映射表是介质专用的，而应用程序的返回值是应用程序专用的。

根据本发明，有多种方式将应用程序与映射表相关联。可以在任何时刻进行有效退出代码映射表的确定，但是优选地，当运行应用程序时进行所述确定。应用程序可以是任何可执行程序，以及映射表可以是将输入值与输出值进行链接的任何数据结构。应用程序与映射表之间的关联的示例是通过文件名、通过目录结构、或者通过独立的映射表进行关联。在第一情况下，例如，应用程序和保存映射表的文件具有有着不同文件名扩展的相同文件名。

在第二情况下，应用程序和保存映射表的文件可以存储在介质上的相同的目录或文件夹中。

在第三情况下，可以在介质上存储独立的映射表，用于将运行的应用程序的应用程序标识符映射至各个退出代码映射表的标识符。

根据本发明，在应用程序开始其生命周期的结束时，执行以下步骤：

首先，提供退出状态值。其次，使用映射表，将该退出状态值转变为标识符，第三，播放器继续由该标识符表示的实体。以下更加详细地解释这些步骤。

对于“经典的”应用程序模型，通过 `System.exit (int 状态)` 方法来提供退出状态值，即，终止的应用程序将表示其状态的整数值返回下一个控制层，即 AM。在 Xlet 应用程序模型的情况下，应用程序通过 AMI，仅将其生命周期结束的信号发送给 AM。此外，根据本发明，终止的应用程序将退出状态提供给 AM。有许多实现方法。在优选实施例中，扩展 AMI，使得应用程序可以直接将退出状态值传递给 AM。在本发明的另一实施例中，在发送生命周期结束信号之前，在应用程序和 AM 已知并可用的指定位置处存储退出状态值。为此，可以使用任何种类的易失性或非易失性存储器。

在应用程序未将退出状态消息递送至 AM 的情况下，例如，当使用现有的 AMI 时，索引表可以提供指定条目来处理该情况，或者使用

缺省值。

在第二步骤中，AM 使用映射表中的消息，以将实际退出状态转变为标识符，即查找表。

在第三步骤中，播放器继续由标识符表示的实体。如上所述，如果指定了标题号，则可以转变为另一标题，可以包括“模式改变”。如果标识符表示 Java 应用程序，则播放器在当前标题中启动指定的应用程序，即不需要标题或模式转变。

上述表 1 是概念性的表。例如，通过使用如在下表的表 2 中示出的缺省值，可以比以上所示出的表更加有效地存储该表。

int 状态	Title# / App_ID
0	3781
1	8273
2	3922
缺省值	1

表 2：更加有效地编码的映射表

在优选实施例中，该表覆盖了相应应用程序的可能退出状态值的完整的范围。在另一实施例中，该表不需要覆盖完整的范围、或者留下一些空白的标识符域。在这种情况下，如果不能将返回值映射至应用程序或标题，则根本不执行转变。

对于标识应用程序或标题的表 2 的右手列中的值，可以预留值的范围，例如，第一范围预留给标题标识符，第二范围预留给应用程序标识符。

有利地，本发明提供了非常鲁棒的方式，来指定 Java 应用程序生命周期结束处的系统行为。具体地，在盘公开发行之之前，针对连贯性，可以容易地验证映射表，因此确保在应用程序的运行时间内，对于任何退出状态值来启动（launch）有效的标题或应用程序。

本发明的另一个优点是，易于重新使用 Java 应用程序。这包括，如上所述，在相同盘的不同配置中重新使用应用程序，以及使用特定的应用程序来创作出不同的盘。由于由映射表定义了系统转变行为，因而与应用程序本身独立，所以易于在不同的环境中重新使用相同的

应用程序。例如，可以容易地将相同的应用程序集成于不同的盘中。只有映射表需要改编，但是不需要编程来改变应用程序代码。

本发明的另一个优点是，在应用程序终止时，即在应用程序生命周期结束时，可以启动另一个 Java 应用程序。这提供了简单的方式以将多个应用程序集成于一个标题中。目前的解决方案会需要编程工作来将不同的应用程序集成于单个应用程序。因此，本发明支持灵活和精密的标题结构化。这种改进的模块为内容作者提供了附加的灵活性。

此外，本发明可以用作构建模块，以将 Java 应用程序集成至目前的系统中。

在本发明的一个实施例中，从只读存储介质中获取终止的软件应用程序和要执行的下一个软件应用程序的程序数据，以及从可重写存储介质（例如，硬盘驱动器（HDD））中获取表示映射表的数据。这具有可以在以后修改映射表的优点。对于两个映射表可用，其中一个映射表在存储了终止应用程序的介质上，以及一个映射表在诸如 HDD 之类的本地存储介质上的情况，可以提供例如“HDD 中的映射表具有较高优先级”之类的规则。

本发明的总体思想是概念性的映射表，用于将应用程序退出状态信息（应用程序专用的）转变为表示标题或 Java 应用程序（盘专用的）的标识符。当完成 Java 应用程序时，播放器估计映射表，以及继续从该表中得到的特定实体。

本发明的主要优点在于，由于提高的鲁棒性、易于重新使用应用程序、以及更高的灵活性，而使得允许灵活地转变为其它标题或 Java 应用程序。尤其是，由于可以模块化逐步地执行，而不是必须在不同的环境中立即验证整个介质，所以使在创作出之后进行介质的验证更加容易。

本发明可以用于数据结构，尤其是用于存储在存储介质上的数据结构，这包含用于在虚拟机上执行的应用程序（如在 JVM 上执行的 Java 应用程序）的数据。本发明可以与以 Java 编程语言编写的应用程序、或者与以在 JVM 或通用虚拟机上执行的其它语言编写的程序一起使用。

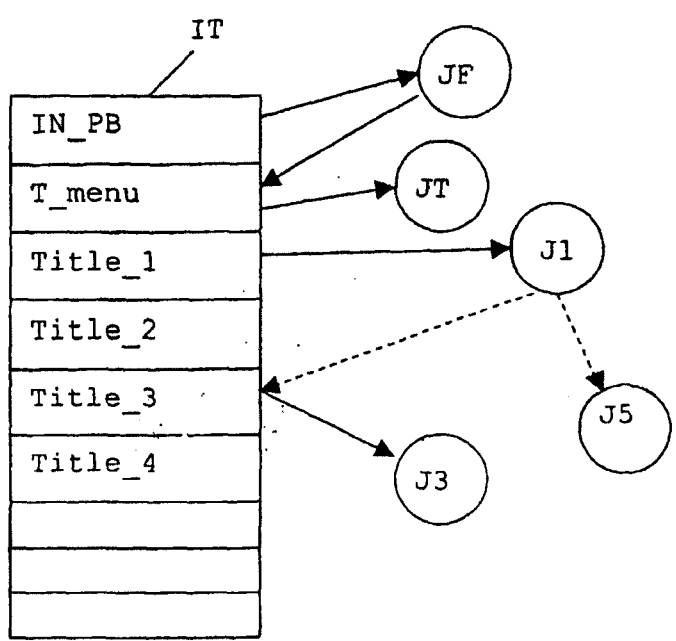


图 1

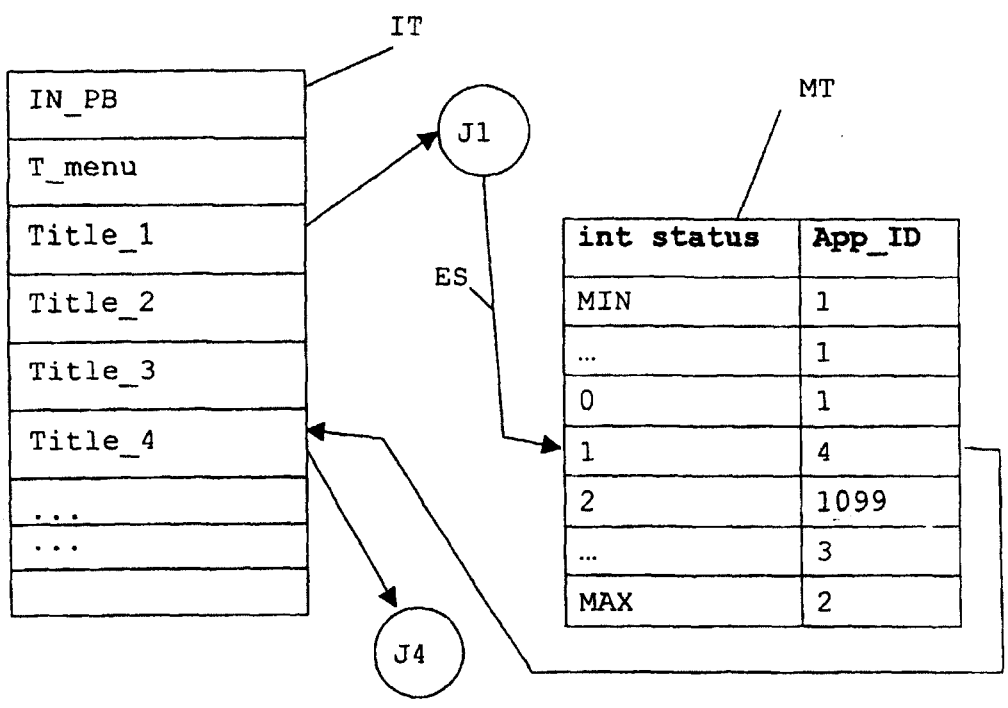


图 2

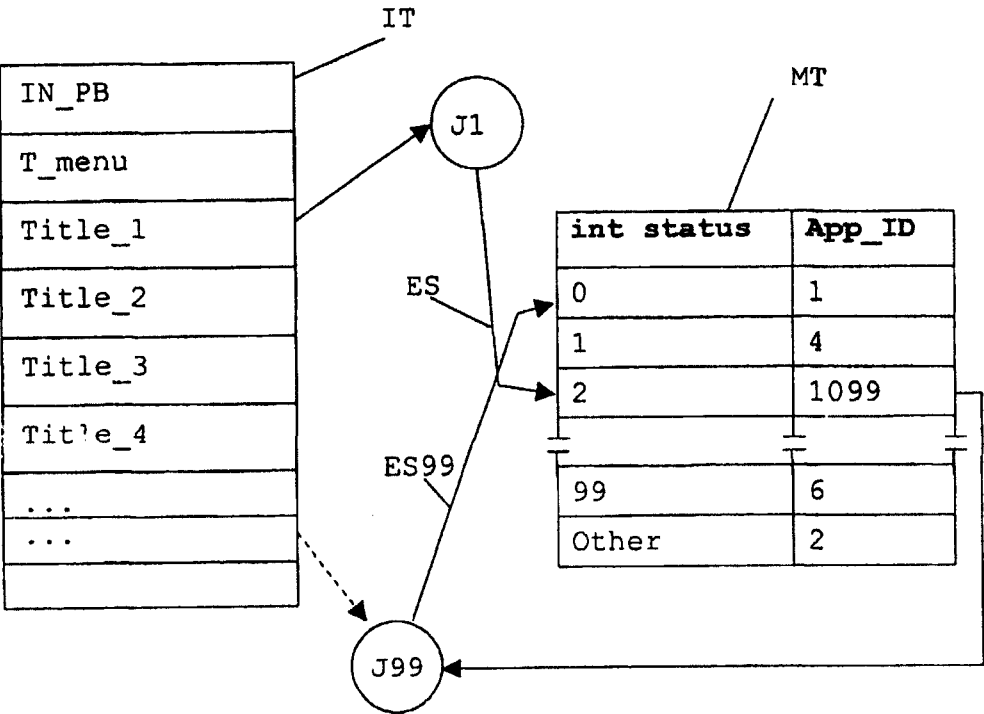


图 3

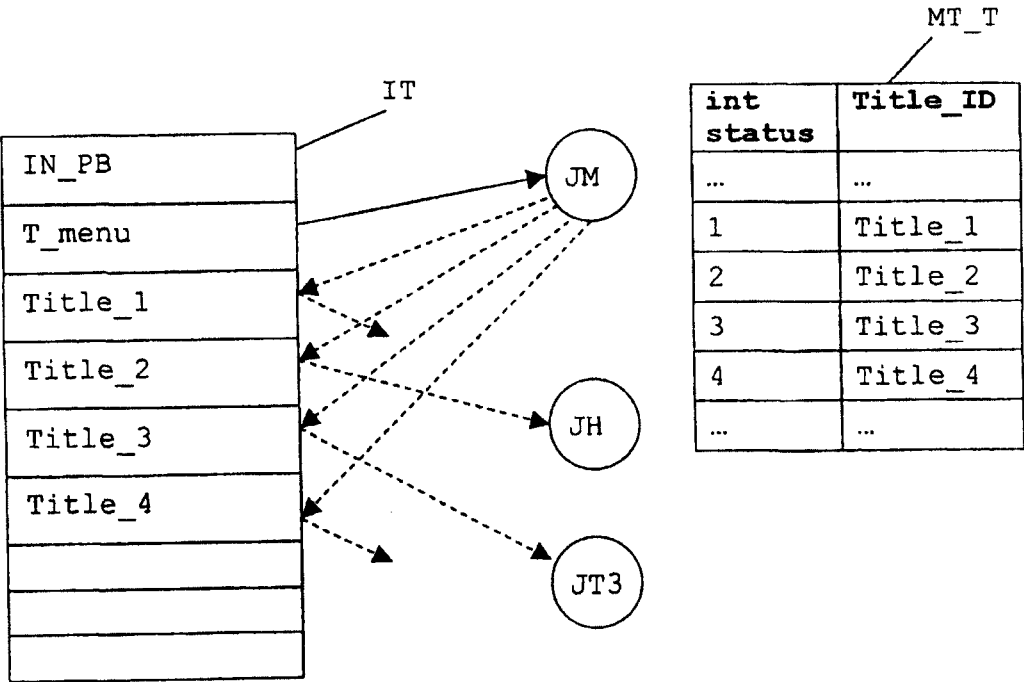


图 4

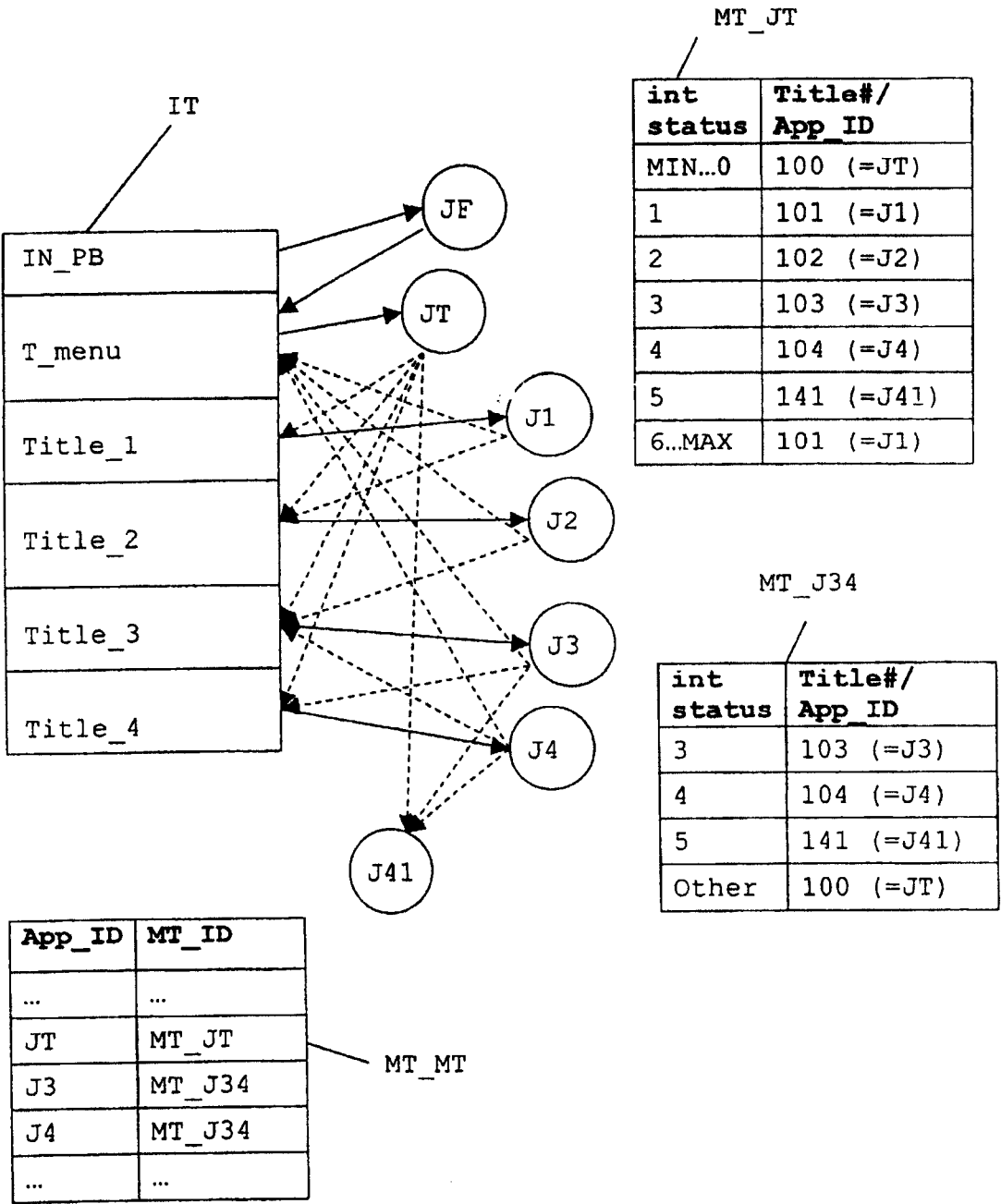


图 5