



(12) 发明专利

(10) 授权公告号 CN 101395569 B

(45) 授权公告日 2010.12.29

(21) 申请号 200780007604.7

G06F 9/445(2006.01)

(22) 申请日 2007.03.06

(56) 对比文件

(30) 优先权数据

075551/2006 2006.03.17 JP

JP 特开 2006-65839 A, 2006.03.09, 全文.

CN 1573704 A, 2005.02.02, 全文.

EP 0942355 A2, 1999.09.15, 全文.

(85) PCT申请进入国家阶段日

2008.09.02

审查员 邹予婷

(86) PCT申请的申请数据

PCT/JP2007/054757 2007.03.06

(87) PCT申请的公布数据

W02007/108340 EN 2007.09.27

(73) 专利权人 佳能株式会社

地址 日本东京

(72) 发明人 古谷智行

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 魏小薇

(51) Int. Cl.

G06F 3/12(2006.01)

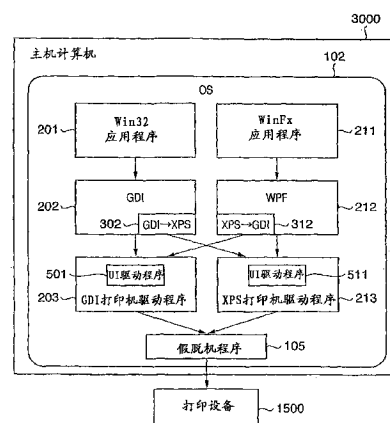
权利要求书 3 页 说明书 14 页 附图 16 页

(54) 发明名称

用于改变信息处理设备中的打印机驱动程序的方法

(57) 摘要

根据本发明的信息处理设备是包括下列单元的信息处理设备:判断单元,判断源应用程序和目的地打印机驱动程序的组合是第一类型应用程序和第一类型打印机驱动程序的组合还是第二类型应用程序和第二类型打印机驱动程序的组合,以及改变单元,在所述判断单元判断源应用程序和目的地打印机驱动程序不是任何一个所述组合的情况下,将由操作系统所参照的打印机驱动程序属性改变为满足所述组合的属性值。



1. 一种信息处理设备,其中能够操作第一图形处理单元和第二图形处理单元,所述第一图形处理单元根据来自第一类型应用程序的打印指令输出能够由第一类型打印机驱动程序处理的第一图形数据,所述第二图形处理单元根据来自第二类型应用程序的打印指令输出能够由第二类型打印机驱动程序处理的第二图形数据,所述信息处理设备包括:

选择单元,适用于选择目的地打印机驱动程序;

判断单元,适用于基于所述选择单元选择的目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的当前组合是第一类型应用程序和第一类型打印机驱动程序的第一组合还是第二类型应用程序和第二类型打印机驱动程序的第二组合;以及

改变单元,适用于在所述判断单元判断当前组合既不是第一组合也不是第二组合的情况下,无需来自用户的任何指令而将由源应用程序所使用的目的地打印机驱动程序改变为这样的打印机驱动程序,使得当前组合变为第一组合或者第二组合,

其中,所述第一打印机驱动程序和第二打印机驱动程序用于同一打印机。

2. 根据权利要求1所述的信息处理设备,其中,所述改变单元将作为响应由源应用程序传输到目的地打印机驱动程序的初始化指令生成的打印机驱动程序名称的驱动程序句柄改变为满足所述组合的打印机驱动程序的驱动程序句柄。

3. 根据权利要求1所述的信息处理设备,进一步包括存储单元,适用于存储由用户作为打印设置信息输入的打印设置,其中

所述改变单元将打印设置信息中包含的目的地打印机驱动程序的标识符改变为满足所述组合的打印机驱动程序的标识符。

4. 一种信息处理设备,其中第一图形处理单元和第二图形处理单元在操作系统的管理下操作,所述第一图形处理单元根据来自第一类型应用程序的打印指令输出能够由第一类型打印机驱动程序处理的第一图形数据,所述第二图形处理单元根据来自第二类型应用程序的打印指令输出能够由第二类型打印机驱动程序处理的第二图形数据,以及

操作系统参照打印机驱动程序属性,生成能够由与所参照的属性相对应的类型的打印机驱动程序处理的图形数据,所述信息处理设备包括:

选择单元,适用于选择目的地打印机驱动程序;

判断单元,适用于基于所述选择单元选择的目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的当前组合是第一类型应用程序和第一类型打印机驱动程序的第一组合还是第二类型应用程序和第二类型打印机驱动程序的第二组合;以及

改变单元,适用于在所述判断单元判断当前组合既不是第一组合也不是第二组合的情况下,无需来自用户的任何指令而将操作系统所参照的打印机驱动程序属性改变为这样的属性值,使得当前组合变为第一组合或者第二组合,

其中,所述第一打印机驱动程序和第二打印机驱动程序用于同一打印机。

5. 一种用于改变信息处理设备中的打印机驱动程序的方法,在所述信息处理设备中,能够操作第一图形处理单元和第二图形处理单元,所述第一图形处理单元根据来自第一类型应用程序的打印指令输出能够由第一类型打印机驱动程序处理的第一图形数据,所述第二图形处理单元根据来自第二类型应用程序的打印指令输出能够由第二类型打印机驱动

程序处理的第二图形数据,所述方法包括:

选择步骤,选择目的地打印机驱动程序;

判断步骤,基于在所述选择步骤中选择的目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的当前组合是第一类型应用程序和第一类型打印机驱动程序的第一组合还是第二类型应用程序和第二类型打印机驱动程序的第二组合;以及

改变步骤,在所述判断步骤中判断当前组合既不是第一组合也不是第二组合的情况下,无需来自用户的任何指令而将由源应用程序所使用的目的地打印机驱动程序改变为这样的打印机驱动程序,使得当前组合变为第一组合或者第二组合,

其中,所述第一打印机驱动程序和第二打印机驱动程序用于同一打印机。

6. 根据权利要求 5 所述的用于改变信息处理设备中的打印机驱动程序的方法,其中,在改变步骤中,作为响应由源应用程序传输到目的地打印机驱动程序的初始化指令生成的打印机驱动程序名称的驱动程序句柄被改变为满足所述组合的打印机驱动程序的驱动程序句柄。

7. 根据权利要求 5 所述的用于改变信息处理设备中的打印机驱动程序的方法,进一步包括存储步骤,用于存储由用户作为打印设置信息输入的打印设置,其中

在所述改变步骤中,打印设置信息中包含的打印机驱动程序的标识符被改变为满足所述组合的打印机驱动程序的标识符。

8. 一种用于改变信息处理设备中的打印机驱动程序的方法,在所述信息处理设备中,第一图形处理单元和第二图形处理单元在操作系统的管理下操作,所述第一图形处理单元根据来自第一类型应用程序的打印指令输出能够由第一类型打印机驱动程序处理的第一图形数据,所述第二图形处理单元根据来自第二类型应用程序的打印指令输出能够由第二类型打印机驱动程序处理的第二图形数据,以及

操作系统参照打印机驱动程序属性,生成能够由与所参照的属性相对应的类型的打印机驱动程序处理的图形数据,所述方法包括:

选择步骤,选择目的地打印机驱动程序;

判断步骤,用于基于在所述选择步骤中选择的目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的当前组合是第一类型应用程序和第一类型打印机驱动程序的第一组合还是第二类型应用程序和第二类型打印机驱动程序的第二组合;以及

改变步骤,用于在所述判断步骤中判断当前组合既不是第一组合也不是第二组合的情况下,无需来自用户的任何指令而将操作系统所参照的打印机驱动程序属性改变为这样的属性值,使得当前组合变为第一组合或者第二组合,

其中,所述第一打印机驱动程序和第二打印机驱动程序用于同一打印机。

9. 一种信息处理设备,其中能够操作第一图形处理单元和第二图形处理单元,所述第一图形处理单元根据来自第一类型应用程序的打印指令输出能够由第一类型打印机驱动程序处理的第一图形数据,所述第二图形处理单元根据来自第二类型应用程序的打印指令输出能够由第二类型打印机驱动程序处理的第二图形数据,所述信息处理设备包括:

选择单元,适用于选择目的地打印机驱动程序;

判断单元,适用于基于所述选择单元选择的目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的当前组合是第一类型应用程序和第二类型打印机驱动程序的第一组合还是第二类型应用程序和第一类型打印机驱动程序的第二组合;以及

改变单元,适用于在所述判断单元判断当前组合是第一组合或第二组合的情况下,无需来自用户的任何指令而将由源应用程序所使用的目的地打印机驱动程序改变为这样的打印机驱动程序,使得当前组合变为既不是第一组合也不是第二组合,

其中,所述第一打印机驱动程序和第二打印机驱动程序用于同一打印机。

10. 一种用于控制信息处理设备的方法,在所述信息处理设备中,能够操作第一图形处理单元和第二图形处理单元,所述第一图形处理单元根据来自第一类型应用程序的打印指令输出能够由第一类型打印机驱动程序处理的第一图形数据,所述第二图形处理单元根据来自第二类型应用程序的打印指令输出能够由第二类型打印机驱动程序处理的第二图形数据,所述方法包括:

选择步骤,选择目的地打印机驱动程序;

判断步骤,基于在所述选择步骤中选择的目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的当前组合是第一类型应用程序和第二类型打印机驱动程序的第一组合还是第二类型应用程序和第一类型打印机驱动程序的第二组合;以及

改变步骤,在所述判断步骤中判断当前组合是第一组合或第二组合的情况下,无需来自用户的任何指令而将由源应用程序所使用的目的地打印机驱动程序改变为这样的打印机驱动程序,使得当前组合变为既不是第一组合也不是第二组合,

其中,所述第一打印机驱动程序和第二打印机驱动程序用于同一打印机。

用于改变信息处理设备中的打印机驱动程序的方法

技术领域

[0001] 本发明涉及能够使用例如多个不同图形引擎进行打印处理等等的信息处理设备，以及用于改变信息处理设备中的打印机驱动程序的方法。

背景技术

[0002] 为了使用打印设备从由主机计算机执行的应用程序打印图像和文本，操作系统一般配备有诸如图 3 所示的配置。更具体地说，应用程序 101 向图形引擎 103 移交图形数据，在被移交到打印机驱动程序 104 之前，图形数据由图形引擎 103 进行处理。打印机驱动程序生成对应于打印设备的打印数据（一般是指 PDL[页面描述语言]）。打印数据存储在假脱机程序（spooler）105 中，然后被发送到打印设备 1500 中。

[0003] 具体来说，图形引擎 103 对图形数据进行处理，如转换由应用程序 101 所创建的图形数据的尺寸或者根据打印机驱动程序 104 的吞吐量进行仿真处理。因此，应用程序 101 和打印机驱动程序 104 两者都能够执行独立操作。图形引擎 103 通常是作为 OS（操作系统）102 的一部分而被提供的。

[0004] 另一方面，图形引擎 103 的数量不一定限于一个。某些配置（OS）允许两个或更多图形引擎共存。

[0005] 例如，微软公司在 2005 在美国西雅图举办了标题为 WinHEC-2005 的硬件工程会议。在该会议中，推出了微软的新 OS（名为 **Windows®** Vista），该操作系统具有这样的配置：两个图形引擎共存，如图 1 所示，（Microsoft Corporation，" Advances in WindowsPrinting:TWPR05001_WinHEC05.ppt "，[online May 7,2004;searched February28,2005]，<http://www.microsoft.com/whdc/device/print/default.msp>。）。

[0006] 微软的常规打印系统通过一个应用程序（以下称为 Win32 应用程序）使用叫做 GDI（图形设备接口）的图形引擎，该应用程序使用被称为 Win32API 的 API（应用程序编程接口），其中打印数据是由从 GDI 调用的打印机驱动程序（以下称为 GDI 打印机驱动程序）从图形数据创建的。在本实施例中，此打印处理流程将被称为 GDI 打印路径。

[0007] 对于 Windows Vista，除了常规 GDI 打印路径之外，还将提供被称为 XPS 打印路径的新的打印处理流程。对于 XPS 打印路径，由 OS 通过使用叫做 WinFx API 的 API 的应用程序（以下称为 WinFx 应用程序）提供了叫做 WPF（Windows 呈现基础，WindowsPresentation Foundation）的图形引擎。WPF 图形引擎向打印机驱动程序（以下简称为 XPS 打印机驱动程序）移交 XPS（XML 纸张规范，XML Paper Specification）格式图形数据，从而，打印机驱动程序将图形数据转换为打印数据。

[0008] GDI 和 WPF 图形引擎也能够彼此协作。此协作使图形数据能够从 Win32 应用程序移交到 XPS 打印机驱动程序，或从 WinFx 应用程序移交到 GDI 打印机驱动程序。

[0009] 为了使此协作起作用，当使用 GDI 打印机驱动程序从 WinFx 应用程序进行打印时，图形数据首先被 WPF 图形引擎转换为 XPS 格式图形数据，然后，从 XPS 格式图形数据转换为 EMF 格式图形数据。经过转换的图形数据被存储在 EMF 假脱机文件中，随后由 GDI 驱动程序

转换为打印数据。

[0010] 此外,当使用 XPS 打印机驱动程序从 Win32 应用程序进行打印时,图形数据首先被 GDI 图形引擎转换为 EMF 格式图形数据,然后,从 EMF 格式图形数据转换为 XPS 格式图形数据。经过转换的图形数据被存储在 XPS 假脱机文件中,随后由 XPS 打印机驱动程序转换为打印数据。

[0011] 如上文所描述的,在 Windows Vista 中提供了四个打印处理流程。如此,可以通过准备 GDI 打印机驱动程序或 XPS 打印机驱动程序中的任何一个,实现从 Win32 应用程序和 WinFx 应用程序两者的打印处理。

[0012] 然而,由于 XPS 格式和 EMF 格式彼此不同,因此,图形数据格式的转换在打印质量、打印功能、打印速度等等方面会有问题。

[0013] 例如,在 EMF 格式中得到支持的包含诸如 ROP 运算(或光栅操作处理运算)之类的逻辑运算的图形数据在 XPS 格式中得不到支持。因此,取决于由微软公司指定的规范,当经过用于将 EMF 格式转换为 XPS 格式的 GDI-XPS 转换模块的转换时,存在可能从包括逻辑运算的图形数据中丢失逻辑运算信息的风险。当删除了逻辑运算信息的图形数据被移交到 XPS 打印机驱动程序时,XPS 打印机驱动程序可能不能创建由 Win32 应用程序期望的输出结果。

[0014] 相反,在 XPS 格式中得到支持的复杂的图形数据在 EMF 格式中得不到支持。因此,当使用用于将 XPS 格式转换为 EMF 格式的 XPS-GDI 转换模块进行转换时,这样的图形数据在叫做“增色(flattering)”的过程中被转换为局部化位图。在此情况下,由于图形数据被转换为位图数据,因此,GDI 打印机驱动程序将不能判断原对象属性,因此,将不能创建对应于对象属性的输出结果。换句话说,通过数据格式转换而输出的图像的质量将恶化。例如,当使用打印机驱动程序每张纸打印多个页面时,因为被转换为位图的数据将按比例缩小和布局,所以图像质量将低于图形对象被以缩小的尺寸绘制的情况。

[0015] 此外,例如,在 XPS 格式中得到支持的电子签名信息在 EMF 格式中得不到支持。此外,在 EMF 格式中得到支持的、来自应用程序的信息被直接传输到 GDI 打印机驱动程序的处理,在 XPS 格式中得不到支持。因此,当在转换之前的格式的图形数据包含使用在转换后的格式中得不到支持的功能的指令的情况下进行从 XPS 格式到 EMF 格式的转换或相反的转换时,这样的功能将不会被执行。换句话说,执行数据格式转换处理时不能实现的功能可能存在于各个格式的图形数据中。

[0016] 另外,GDI-XPS 转换或 XPS-GDI 转换等等的处理的发生会导致处理开销,因此,与不需要进行转换的处理路径相比,打印处理速度可能降低。

[0017] 如上文所描述的,当进行数据格式转换时,在质量、功能和速度的所有方面都会出现问题。因此,希望避免牵涉数据格式转换的打印处理路径。

[0018] 另一方面,当在一台计算机中安装了同一个打印设备的 XPS 打印机驱动程序和 GDI 打印机驱动程序两者时,对要使用的打印机驱动程序的选择由用户自行决定。换句话说,在 **Windows®** Vista 中,图形引擎将不会自动地选择不需要数据格式转换的处理路径。此外,大多数用户不会意识到某一应用程序是 Win32 应用程序还是 WinFx 应用程序,也不会意识到存在四个打印处理流程。因此,当一个打印机驱动程序被选为默认打印机驱动程序时,将使用选定的打印机驱动程序,不管应用程序类型如何。如上文所描述的,在打印

处理路径中,避免诸如 GDI-XPS 转换或 XPS-GDI 转换之类的数据格式转换可能证明是困难的。

发明内容

[0019] 本发明是在考虑到上面的常规示例的情况下作出的,其目的是解决上文所描述的问题。此外,本发明的另一个目的是,在有多个打印机驱动程序可以供打印机使用的情况下,提供不管选择了哪一个打印机驱动程序,都能够使用与应用程序所使用的图形引擎相对应的打印机驱动程序进行打印处理的信息处理设备,还提供用于改变信息处理设备处的打印机驱动程序的方法。

[0020] 为了实现上文所描述的目的,本发明是按如下方式配置的。换句话说,一种信息处理设备,其中,能够操作第一图形处理单元和第二图形处理单元,所述第一图形处理单元将从第一类型应用程序输出的图形数据转换为能够由第一类型打印机驱动程序处理的第一打印信息,所述第二图形处理单元将从第二类型应用程序输出的图形数据转换为能够由第二类型打印机驱动程序处理的第二打印信息,该信息处理设备包括:

[0021] 判断单元,适用于基于目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的组合是第一类型应用程序和第一类型打印机驱动程序的组合还是第二类型应用程序和第二类型打印机驱动程序的组合;以及

[0022] 改变单元,适用于在所述判断单元判断源应用程序和目的地打印机驱动程序不是任何一个所述组合的情况下,将由源应用程序所使用的目的地打印机驱动程序改变为满足所述组合的打印机驱动程序。

[0023] 从不同的观点来看,本发明是一种信息处理设备,其中,第一图形处理单元和第二图形处理单元在操作系统的管理下操作,所述第一图形处理单元将从第一类型应用程序输出的图形数据转换为能够由第一类型打印机驱动程序处理的第一打印信息,所述第二图形处理单元将从第二类型应用程序输出的图形数据转换为能够由第二类型打印机驱动程序处理的第二打印信息,以及

[0024] 操作系统参照打印机驱动程序属性,如有必要,则将图形数据转换为能够由与所参照的属性相对应的类型的打印机驱动程序处理的打印信息,所述信息处理设备包括:

[0025] 判断单元,适用于基于目的地打印机驱动程序的类型和源应用程序的类型,判断源应用程序和目的地打印机驱动程序的组合是第一类型应用程序和第一类型打印机驱动程序的组合还是第二类型应用程序和第二类型打印机驱动程序的组合;以及

[0026] 改变单元,适用于在所述判断单元判断源应用程序和目的地打印机驱动程序不是任何一个所述组合的情况下,将由操作系统所参照的打印机驱动程序属性改变为满足所述组合的属性值。

[0027] 根据上文所描述的配置,在有多个打印机驱动程序可以供打印机使用的情况下,可以使用与应用程序所使用的图形引擎相对应的打印机驱动程序来进行打印,不管选择了哪一个打印机驱动程序。

[0028] 通过下列(参照附图)对示例性实施例的描述,本发明的其他特征将变得清晰。

附图说明

- [0029] 图 1 是示出了作为多个图形引擎共存的示例的根据**Windows®**Vista 的打印处理流程和对应于打印处理的组件的图；
- [0030] 图 2 是示出了根据**Windows®**Vista 的两个图形引擎共存的机构的图；
- [0031] 图 3 是示出了一般打印处理流程和对应于打印处理的组件的图；
- [0032] 图 4 是示出了应用了根据本发明的实施例的打印控制设备、控制方法及其打印机驱动程序的信息处理系统的配置的示例的方框图；
- [0033] 图 5 是示出了与实施例有关的 GDI 打印机驱动程序和 XPS 打印机驱动程序的配置的图；
- [0034] 图 6 是示出了安装了与实施例有关的 GDI 打印机驱动程序和 XPS 打印机驱动程序两者的情况的图；
- [0035] 图 7 是示出了根据实施例的用户界面驱动程序的配置的图；
- [0036] 图 8 是示出了用户界面驱动程序的组合判断单元进行的判断的准则的图；
- [0037] 图 9A 是示出了根据第一实施例的用户界面驱动程序的序列的图；
- [0038] 图 9B 是示出了根据第一实施例的用户界面驱动程序的序列的图；
- [0039] 图 10 是示出了当初始化指令被移交到用户界面驱动程序时根据第一实施例的用户界面驱动程序的处理流程的流程图；
- [0040] 图 11 是示出了根据第二实施例的用户界面驱动程序的配置的图；
- [0041] 图 12 是示出了根据第二实施例的用户界面驱动程序的序列的图；
- [0042] 图 13 是示出了当打印设置 UI 显示指令被移交到用户界面驱动程序时根据第二实施例的用户界面驱动程序的处理流程的流程图；
- [0043] 图 14 是示出了根据第三实施例的驱动程序的配置的图；
- [0044] 图 15 是示出了根据第三实施例的驱动程序的配置的图；以及
- [0045] 图 16 是根据第三实施例的接收了打印指令的用户界面驱动程序的处理流程图。

具体实施方式

[0046] [第一实施例]

[0047] < 计算机配置 >

[0048] 现在将描述根据本发明的打印控制设备的实施例。打印控制设备是通过计算机（信息处理设备）实现的。现在将描述计算机的软件配置的示例。计算机的硬件配置类似于通用计算机的硬件配置，并包括 CPU、存储器、用户界面设备（键盘、显示器等等），以及文件存储设备等等。

[0049] 图 1 示出了 Windows Vista 中的打印处理流程和用于进行打印处理的软件组件，作为多个图形引擎共存的示例。在 Windows Vista 中，应用程序可以分为两种类型的应用程序，即：作为第一类型应用程序的 Win32 应用程序 201，以及作为第二类型应用程序的 WinFx 应用程序 211。OS102 上存在两个图形引擎，即：作为第一图形处理单元的 GDI202，以及作为第二图形处理单元的 WPF212。Win32 应用程序 201 向 GDI202 移交作为第一图形数据的 GDI 函数，而 WinFx 应用程序 211 向 WPF212 移交作为第二图形数据的 WPF API 格式数据。此外，打印机驱动程序可以分为两种类型的打印机驱动程序，即：作为第一类型打印机驱动程序的 GDI 打印机驱动程序 203，以及作为第二类型打印机驱动程序的 XPS 打印机驱动程

序 213。GDI 打印机驱动程序 203 从 GDI202 接收图形数据,并将该图形数据转换为用于打印设备 1500 的打印数据。XPS 打印机驱动程序 213 从 WPF212 接收图形数据,并将该图形数据转换为用于打印设备 1500 的打印数据。由打印机驱动程序所创建的打印数据通过假脱机程序 105 传输到打印设备 1500。

[0050] 在此说明书中,用于创建打印数据的、从 Win32 应用程序 201 经由 GDI202 到 GDI 打印机驱动程序 203 的打印处理流程,将被称为 GDI 打印路径。另外,用于创建打印数据的、从 WinFx 应用程序 211 经由 WPF212 到 XPS 打印机驱动程序 213 的打印处理流程,将被称为 XPS 打印路径。此外,GDI 打印路径和 XPS 打印路径将被称为直接打印路径。然而,打印处理流程不限于 GDI 打印路径和 XPS 打印路径,还存在用于创建打印数据的、从 Win32 应用程序 201 经由 GDI202 到 XPS 打印机驱动程序 213 的打印处理流程。此外,还有用于创建打印数据的、从 WinFx 应用程序 211 经由 WPF212 到 GDI 打印机驱动程序 203 的打印处理流程。这些路径是通过由 GDI→XPS 转换单元 302 和 XPS→GDI 转换单元 312 进行的数据格式转换实现的。图形引擎将数据转换为彼此的数据格式的打印路径将被称为交叉打印路径。

[0051] 形成打印路径的相应的末端的应用程序和打印机驱动程序分别被称为源应用程序和目的地打印机驱动程序。

[0052] 下面,将参照图 2 描述打印处理流程。图 2 示出了两个图形引擎在 **Windows®** Vista 中共存的机制。在 **Windows®** Vista 中存在四种类型的打印处理路径,即:路径 351、352、353 和 354。

[0053] 打印处理路径 351 代表 GDI 打印路径。从 Win32 应用程序 201 移交的图形数据由 GDI202 以 EMF(增强型元文件)格式存储作为 EMS 假脱机文件。随后,由 GDI 打印机驱动程序 203 执行到打印数据的转换。

[0054] 打印处理路径 354 代表在 **Windows®** Vista 中添加的 XPS 打印路径。从 WinFx 应用程序 211 移交的图形数据由 WPF212 存储在 XPS 假脱机文件 311 中,随后由 XPS 打印机驱动程序 213 转换为打印数据。

[0055] 打印处理路径 352 代表来自 WinFx 应用程序 211 的数据由 GDI 打印机驱动程序 203 进行打印的情况下的打印处理路径。从 WinFx 应用程序 211 移交的图形数据在 XPS→GDI 转换模块 312 处经由 WPF212 转换为 EMF 格式的图形数据。经过转换的图形数据存储在 EMF 假脱机文件中,随后由 GDI 驱动程序 203 转换为打印数据。

[0056] 打印处理路径 353 代表来自 Win32 应用程序 201 的数据由 XPS 打印机驱动程序 213 进行打印的情况下的打印处理路径。从 Win32 应用程序 201 移交的图形数据在 GDI→XPS 转换模块 302 处经由 GDI202 转换为 XPS 格式的图形数据。经过转换的图形数据存储在 XPS 假脱机文件 311 中,随后由 XPS 打印机驱动程序 213 转换为打印数据。

[0057] 图 4 是描述了根据本实施例的包括计算机的打印控制系统的配置的方框图。除非另作说明,本发明可以应用于任何系统,如独立的设备,包括多个设备的系统,或经由诸如 LAN 或 WAN 之类的网络进行连接的并经由网络进行处理的系统,只要执行本发明的功能即可。

[0058] 在图 4 中,主机计算机 3000 配备有 CPU1,该 CPU1 基于存储在 ROM3 的程序 ROM 或外部存储器 11 中的文档处理程序等等,进行文档处理,其包括图、图像、文本、表(如电子表格)等等的混合。CPU1 对连接到系统总线 4 的相应的设备进行总体控制。

[0059] ROM3 的程序 ROM 或外部存储器 11 还存储操作系统程序（下面缩写为 OS）等等，它是 CPU1 的控制程序。ROM3 的字体 ROM 或外部存储器 11 存储在上文所提及的文档处理过程中所使用的字体数据等等，而 ROM3 的数据 ROM 或外部存储器 11 存储在上文所提及的文档处理过程中所使用的各种数据。

[0060] RAM2 充当 CPU1 的主存储器、工作区等等。键盘控制器 (KBC) 5 控制来自键盘 9 的键输入或来自指示设备（未显示）的输入。CRT 控制器 (CRTC) 6 控制 CRT 显示器 (CRT) 10 的显示。磁盘控制器 (DKC) 7 控制对外部存储器 11（硬盘 (HD)、软盘 (FD) 等等）的访问。外部存储器 11 存储了启动程序、各种应用程序、字体数据、用户文件、编辑文件、打印机控制命令生成程序（以下称为“打印机驱动程序”）等等。根据本发明的打印机驱动程序也存储在外存储器 11 中。

[0061] 打印机控制器 (PRTC) 8 经由预定双向接口 21 连接到打印机 1500，并对与打印机 1500 的通信进行控制处理。CPU1 例如通过对 RAM2 上配置的显示信息 RAM 上的轮廓字体进行部署（光栅化）处理而使能 CRT10 上的 WYSIWYG。

[0062] 此外，CPU1 还基于通过 CRT10 上的鼠标光标等等（未示出）发出的命令，打开各种注册的窗口，并执行各种数据处理。当进行打印时，用户可以打开涉及打印设置的窗口，以配置打印机设置或者包括打印模式选择的对于打印机驱动程序的打印处理方法的设置。

[0063] 在打印机 1500 处，打印机 CPU12 基于存储在 ROM13 的程序 ROM 中的控制程序等等，或存储在外存储器 14 中的控制程序等等，将图像信号作为输出信息输出到连接到系统总线 15 的打印单元（打印机引擎）17。

[0064] 此外，ROM13 的程序 ROM 还存储 CPU12 的控制程序等等。ROM13 的字体 ROM 存储了当生成上文所提及的输出信息时所使用的字体数据等等，而 ROM13 的数据 ROM 在打印机未配备诸如硬盘之类的外部存储器 14 的情况下存储在主机计算机上所使用的信息等等。

[0065] CPU12 是这样布置的，以便经由输入单元 18 能够进行与主机计算机的通信的处理，打印机内的信息等等可通知到主机计算机 3000。RAM19 充当 CPU12 的主存储器、工作区等等。RAM19 被配置为使得可以通过连接到未示出的扩展端口的可选 RAM，进行存储器扩展。RAM19 被用作输出信息部署区、环境数据存储区，以及 NVRAM 等等。对上文所提及的外部存储器 14（硬盘 (HD)、IC 卡等等）的访问是由存储器控制器 (MC) 20 进行控制的。

[0066] 外部存储器 14 可选地连接，以存储字体数据、仿真程序、表单数据等等。此外，还在操作面板 18 上安置了用于操作的开关、LED 显示装置等等。此外，上文所提及的外部存储器不必限于一个。作为替换方案，可以提供一个或多个外部存储器，使得除内部字体之外，还可以连接多个可选字体卡和存储解释不同语言系统的打印机控制语言的程序的外部存储器。此外，还可以提供未示出的 NVRAM，用于存储来自操作面板 1501 的打印机模式设置信息。

[0067] < 驱动程序的详细配置 >

[0068] 图 5 是示出了与本实施例有关的 GDI 打印机驱动程序 203 和 XPS 打印机驱动程序 213 的配置的图。在图 5 中，有关应用程序和图形引擎的描述已经省略，示出了单个应用程序和单个图形引擎的情况。毫无疑问，如图 1 所示，由图形引擎移交分别对应于每一个打印机驱动程序的格式的图形数据。在本实施例中，相对于打印设备 1500，在 OS102 中安装了 GDI 打印机驱动程序 203 和 XPS 打印机驱动程序 213 两者。换句话说，GDI 打印机驱动程序

203 和 XPS 打印机驱动程序 213 共享同一个打印机作为它们的输出目的地。

[0069] GDI 打印机驱动程序 203 由 GDI UI(或用户界面)驱动程序 501 和 GDI 图形驱动程序 502 构成。GDI 用户界面驱动程序 501 存储了显示在打印机驱动程序的用户界面上的内容和由用户输入到注册表中的打印设置信息 520。GDI 图形驱动程序 502 从图形引擎 103 接收相应的图形数据(DDI[设备驱动程序接口]函数),并根据存储在 GDI 用户界面驱动程序 501 中的打印设置信息 520 从图形数据创建打印数据。

[0070] XPS 打印机驱动程序 213 由 XPS 用户界面驱动程序 511 和 XPS 图形驱动程序 512 构成。XPS 用户界面驱动程序 511 存储了显示在打印机驱动程序的用户界面上的内容和由用户输入到注册表中的打印设置信息 520。XPS 图形驱动程序 512 从图形引擎 103 接收对应的图形数据(XPS 数据),并根据存储在 XPS 用户界面驱动程序 511 中的打印设置信息 520 从图形数据创建打印数据。

[0071] 尽管将 GDI 用户界面驱动程序 501 和 XPS 用户界面驱动程序 511 描述成单独的组件,但是,作为替换也可以使用配备有两种用户界面驱动程序的功能和接口的单一公用用户界面驱动程序。在此情况下,GDI 打印机驱动程序 203 将由公用用户界面驱动程序和 GDI 图形驱动程序 502 组成,而 XPS 打印机驱动程序 213 将由公用用户界面驱动程序和 XPS 图形驱动程序 512 组成。

[0072] 图 6 示出了一个窗口的示例,该窗口示出了在已经安装了与本实施例有关的 GDI 打印机驱动程序和 XPS 打印机驱动程序两者的情况下打印机的状态。如图 6 所示,相对于单一打印设备,(例如打印机 A),GDI 打印机驱动程序 203 被注册为“Printer A(GDI)”601。同时,相对于同一个打印机 A,XPS 打印机驱动程序 213 被注册为“打印机 A(XPS)”。显然,相对于单一打印设备,有两个打印机驱动程序存在,即,GDI 打印机驱动程序 203 和 XPS 打印机驱动程序 213。顺便说一句,Microsoft **Windows®** OS 允许用户选择默认打印机驱动程序。对于被选为默认打印机驱动程序的打印机驱动程序,由操作系统存储了表示该选择的信息,以便可访问。当要从应用程序进行打印并且给出了用于打印的指令时,通常显示出应用程序的打印设置用户界面。然后,用户能够从应用程序的用户界面中选择打印机驱动程序。如果没有进行选择,则选择默认打印机驱动程序,并将使用该打印机驱动程序进行打印。图 6 示出了已经选择“打印机 A(XPS)”602 作为默认打印机驱动程序。

[0073] <用户界面驱动程序的配置和操作>

[0074] 图 7 是最佳地代表本发明的特征的图,并示出了根据第一实施例的用户界面驱动程序 501 和用户界面驱动程序 511 的配置。在第一实施例中,GDI 图形驱动程序 502 和 XPS 图形驱动程序 512 不进行任何特殊处理,只作为一般图形驱动程序操作。GDI 用户界面驱动程序 501 和 XPS 用户界面驱动程序 511 或公用用户界面驱动程序(下面统称为用户界面驱动程序)执行本实施例的处理特征。

[0075] 如图 7 所示,根据本实施例的用户界面驱动程序 501、511 从应用程序 201、211 或图形引擎 202、212 接收初始化指令。作为响应,用户界面驱动程序 501、511 通过应用程序信息获得单元 703 获取有关调用用户界面驱动程序的应用程序是 Win32 应用程序 201 还是 WinFx 应用程序 211 的信息。

[0076] 根据本实施例的用户界面驱动程序从应用程序 201、211 或图形引擎 202、212 接收初始化指令。作为响应,用户界面驱动程序 501、511 使用应用程序信息获得单元 703 获取

正在调用用户界面驱动程序的应用程序 101 的名称。此外,用户界面驱动程序 501、511 还获取有关应用程序是 Win32 应用程序 201 还是 WinFx 应用程序 211 的信息(即,应用程序类型信息)。在本实施例中,可以通过使用由 Win32API 所提供的 GetModuleFileName 函数等等来获取调用用户界面驱动程序的进程名称(可执行文件名称)。另外,还可以按如下方式判断调用用户界面驱动程序的应用程序是 Win32 应用程序 201 还是 WinFx 应用程序 211。更具体地说,从由可执行文件保持的标头部分,获取由可执行文件(其名称是由上文所提及的函数获取的)导入的 DLL(动态链接库)。然后,判断在导入的 DLL 内是否存在 WinFx 应用程序所特有的模块。例如,通过预先制表和存储 WinFx 唯一的模块名称,并且当对从表中获取的模块名称进行的搜索导致命中时,模块可以被判断为 WinFx 唯一的模块。

[0077] 如此判断应用程序是 Win32 应用程序 201 还是 WinFx 应用程序 211。除上述情况之外,例如,如果 OS 提供了输送应用程序类型的 API,则可以使用该 API 获取应用程序类型信息。作为替换方案,可以采用这样的方法:用户界面驱动程序存储应用程序表,该表与进程名称(可执行文件名称)相关联地存储应用程序名称或表示应用程序是 Win32 应用程序还是 WinFx 应用程序的类型信息。为了维护应用程序表,此方法最好这样安排,以便显示维护用户界面,并且可以通过用户输入来更新应用程序表。用户界面驱动程序使用 GetModuleFileName 函数等等来获取调用用户界面驱动程序的进程的名称,并使用进程名称来搜索应用程序表。命中将意味着,对应于相关的进程名称的应用程序名称是调用用户界面驱动程序的应用程序的名称。此外,对应于相关进程名称的应用程序类型是调用用户界面驱动程序的应用程序的类型。

[0078] 还可以使用诸如下面将描述的方法来确定应用程序类型。Win32 应用程序企图以称为 DEVMODE 的格式从用户界面驱动程序中获取打印设置信息。另一方面,WinFx 应用程序企图以称为 PrintTicket 的格式从用户界面驱动程序中获取打印设置信息。用户界面驱动程序能够创建任意一种格式的打印设置信息。如此,可以基于这样的获取请求,确定应用程序类型。

[0079] 接下来,组合判断单元 704 就应用程序和由应用程序调用的用户界面驱动程序的组合是否适当作出判断。图 8 示出了用于进行此判断的准则。适当的结果意味着,要由应用程序使用的图形引擎和打印机驱动程序一致。换句话说,在此情况下没有必要对从图形引擎输出的图形数据的格式进行转换(例如,EMF-XPS 转换)。根据应用程序类型,要由应用程序使用的图形引擎是固定的。因此,如果应用程序类型信息和由应用程序选定的打印机驱动程序是已知的,则可以对组合是否适当进行判断。可以按照前面所描述的方式获取应用程序类型。如果有关打印机驱动程序的选择的信息存储在计算机上,则可以通过参照这样的信息来识别选定的打印机驱动程序。此外,如果要使用的打印机驱动程序的信息(标识符等等)从进行打印的应用程序移交到图形引擎,并可由用户界面驱动程序参照,则可以通过参照这样的信息来标识打印机驱动程序。然后,如果可以参照打印机驱动程序的标识符,则可以确定打印机驱动程序类型(打印机驱动程序是 XPS 打印机驱动程序还是 GDI 打印机驱动程序)。例如,通过预先创建并存储将打印机驱动程序的名称与其类型信息关联的表,可以通过参照该表来确定打印机驱动程序类型。顺便说一句,“适当”组合还可以被表达为同一个类型的应用程序和打印机驱动程序的组合。由于图形引擎还需要判断类型是否相同,以便判断是否需要进行数据格式转换,因此,可以采用相同的判断标准。

[0080] 在组合判断单元 704 判断组合是不适当的情况下,初始化指令单元 705 向配置直接路径的适当用户界面驱动程序发出初始化指令。适当的用户界面驱动程序被包括在与由用户最初选定的打印机驱动程序的同一个目的地打印机相对应的打印机驱动程序中,同时,还被包括在与对应于应用程序的图形引擎一起形成直接路径的打印机驱动程序中。假设 GDI 打印机驱动程序具有表示共享同一个目的地打印机的 XPS 打印机驱动程序是否被安装在信息处理设备中的信息。类似地,假设 XPS 打印机驱动程序具有表示共享同一个目的地打印机的 GDI 打印机驱动程序是否被安装在信息处理设备中的信息。此外,如果用户界面驱动程序是公用用户界面驱动程序,则用户界面驱动程序显然具有表示 GDI 图形驱动程序和 / 或 XPS 图形驱动程序是否存在的信息。此描述假设 GDI 用户界面驱动程序和 XPS 用户界面驱动程序被配置为单独的用户界面驱动程序模块。

[0081] 随后,从响应初始化指令执行初始化处理的用户界面驱动程序接收驱动程序句柄。由驱动程序句柄返回单元 707 接收来自用户界面驱动程序的驱动程序句柄。驱动程序句柄返回单元 707 将接收到的驱动程序句柄传输(移交)到从中向用户界面驱动程序发出了初始化指令的应用程序 201、211 或图形引擎 202、212。

[0082] 在本实施例中,对于每一个打印机,存在两个驱动程序。因此,如果其中一个驱动程序被判断为不适当,则另一个驱动程序被选为适当的驱动程序。毫无疑问,可以使用表,例如图 8 所示的表(稍后描述),选择适当的用户界面驱动程序。此外,初始化处理还包括例如存储器分配、驱动程序句柄响应等等。驱动程序句柄是用于唯一地标识驱动程序的标识符。句柄使得能标识用户界面驱动程序和打印机驱动程序。

[0083] 同时,在组合判断单元 704 判断组合适当的情况下,驱动程序初始化单元 706 按原样执行用户界面驱动程序的初始化。通过初始化处理生成的驱动程序句柄被驱动程序句柄返回单元 707 返回到发出初始化指令的应用程序或图形驱动程序。

[0084] 从用户界面驱动程序接收到驱动程序句柄的应用程序或图形引擎激活由所移交的驱动程序句柄标识的用户界面驱动程序,并允许用户输入打印设置。此外,由于从用户界面驱动程序接收到驱动程序句柄的应用程序或图形引擎将图形数据输出到由所移交的驱动程序句柄标识的打印机驱动程序,构成直接打印路径的适当的打印机驱动程序进行打印数据转换,或者,换句话说,创建打印作业。可以看出,通过让接收激活请求的用户界面驱动程序移交适用于将用来进行打印的应用程序的驱动程序句柄,可以以快速的方式进行能够产生高质量结果的打印处理。

[0085] 顺便说一句,可以有这样的情况:用户界面显示指令与驱动程序的初始化指令一起被移交到用户界面驱动程序,以允许用户进行打印设置。在这样的情况下,如果应用程序和打印机驱动程序的组合被组合判断单元 704 判断为适当,则在用户界面显示单元 701 上显示用户界面对话框。用户通过显示出的 UI 在用户界面设置单元 702 处改变打印设置。另一方面,如果组合被组合判断单元 704 判断为不适当,则初始化指令单元 705 除驱动程序初始化指令之外,还向适当的驱动程序发出用户界面显示指令。以这种方式,用户可以使用适当的驱动程序改变打印设置。

[0086] 图 8 示出了由用户界面驱动程序的组合判断单元进行的判断的准则的示例。在当要从 Win32 应用程序 201 进行打印时选择了 GDI 打印机驱动程序 203 的情况下,判断将会导致“适当”。因此,在图 8 所示的参照表的对应的字段中注册了“适当”。这是因为,在通

过 Win32 应用程序 201 进行打印的过程中使用了 GDI 图形引擎 202, 而 GDI 打印机驱动程序 203 使用 GDI 图形驱动程序 502 作为直接打印路径来进行打印处理。此外, 当从 Win32 应用程序 201 进行打印时选择 XPS 用户界面驱动程序 511 将会导致判断为“不适当”。因此, 在对应的字段中注册了“不适当”。这是因为, 在通过 Win32 应用程序 201 进行打印的过程中使用了 GDI 图形引擎, 而 XPS 打印机驱动程序 213 使用 XPS 图形驱动程序 512 作为交叉打印路径来进行打印处理。

[0087] 类似地, 当从 WinFx 应用程序 211 进行打印时选择 XPS 打印机驱动程序 213 将会导致判断为“适当”。这是因为, 在通过 WinFx 应用程序 211 进行打印的过程中使用了 WPF 图形引擎 212, 而 XPS 打印机驱动程序 213 使用 XPS 图形驱动程序 512 作为直接打印路径来进行打印处理。此外, 当从 WinFx 应用程序 211 进行打印时选择 GDI 用户界面驱动程序 501 将会导致判断为“不适当”。这是因为, 在通过 WinFx 应用程序 211 进行打印的过程中使用了 WPF 图形引擎 212, 进行 XPS→GDI 转换 312, 而 GDI 打印机驱动程序 203 使用 GDI 图形驱动程序 502 作为交叉打印路径来进行打印处理。因此, 为对应的项目, 注册根据判断结果的值。

[0088] 通过在图 8 所示的表中的打印机驱动程序 203、213 的字段中记下对应的驱动程序句柄, 可以从图 8 的表中获取适用于应用程序的驱动程序句柄。

[0089] 图 9A 和 9B 是根据第一实施例的用户界面驱动程序的序列图。在图 9A 中, 当从应用程序移交初始化指令和用户界面显示指令 901 时, 组合判断单元 704 在步骤 902 中判断该驱动程序是否是适用于发出初始化指令的应用程序的驱动程序。如果该驱动程序被判断为不适当, 则初始化指令单元 705 在步骤 903 中向适用于该应用程序的用户界面驱动程序发出初始化指令和用户界面显示指令。在接收到初始化指令时, 用户界面驱动程序在步骤 904 中执行打印机驱动程序 (图形驱动程序) 的初始化处理, 并将驱动程序句柄返回到初始化指令的发出源 (即, 用户界面驱动程序)。在步骤 905 中, 驱动程序句柄返回单元 707 进一步将已返回的驱动程序句柄返回到应用程序。应用程序使用对应于驱动程序句柄的打印机驱动程序执行打印处理。以这种方式, 可以通过适当的图形驱动程序执行打印, 而同时避免诸如处理路径 352 或处理路径 353 之类的所谓的交叉打印路径。

[0090] 同时, 在图 9B 中, 用户界面驱动程序接收初始化指令和用户界面显示指令 911。在步骤 912 中, 组合判断单元 704 判断用户界面驱动程序是否适用于初始化指令的发出源应用程序。图 9B 示出了驱动程序被判断为适当。在此情况下, 接收到初始化指令的用户界面驱动程序在步骤 913 中执行对应的图形驱动程序的初始化处理, 并在步骤 914 中将驱动程序句柄返回到初始化指令的发出源, 或者, 换句话说, 应用程序。该应用程序使用对应于驱动程序句柄的打印机驱动程序执行打印处理。以这种方式, 可以通过适当的图形驱动程序执行打印, 而同时避免诸如处理路径 352 或处理路径 353 之类的所谓的交叉流程。

[0091] < 用户界面驱动程序的处理过程 >

[0092] 图 10 是示出了当初始化指令被移交到用户界面驱动程序时根据第一实施例的用户界面驱动程序 501、511 的处理流程的流程图。当向应用程序 201、211 发出打印指令时, 用户通过对应用程序 201、211 进行操作来发出打印机驱动程序的打印设置的指令。应用程序向指定的打印机驱动程序的用户界面驱动程序发出初始化指令和用户界面显示指令。图 10 示出了在接收到初始化指令时由用户界面驱动程序执行的处理。

[0093] 首先,在步骤 10-1 中,用户界面驱动程序接收初始化指令。用户界面驱动程序获取调用用户界面驱动程序的应用程序的名称(可执行文件名称)。基于所获取的名称,用户界面驱动程序在步骤 10-2 中获取表明应用程序的类型的信息(类型信息),或者,更具体地说,表明正在调用的应用程序是 Win32 应用程序还是 WinFx 应用程序的信息。此处理如参照图 7 所描述的。基于获取的应用程序类型信息,用户界面驱动程序在步骤 10-3 中判断包括用户界面驱动程序的打印机驱动程序是否适用于该应用程序。此判断是例如使用图 8 所示的表执行的。如果判断为不适当,则用户界面驱动程序在步骤 10-5 中向适当的驱动程序发出初始化指令。然后,在步骤 10-6 中,接收从适当的驱动程序返回的驱动程序句柄,接着,由用户界面驱动程序返回到应用程序。

[0094] 在步骤 10-3 中,当判断打印机适用于正在进行调用的源应用程序时,用户界面驱动程序在步骤 10-4 中初始化其自己的驱动程序,并将在初始化过程中创建的驱动程序句柄返回到该应用程序。

[0095] 从另一个打印机驱动程序接收初始化指令的打印机驱动程序也根据图 10 所示的过程进行操作。由于在本实施例中有两种类型的打印机驱动程序,因此,在图 10 所示的过程中的步骤 10-3 中判断的结果对于从另一个用户界面驱动程序接收初始化指令的用户界面驱动程序将始终是“适当”。另外,在此情况下,驱动程序句柄的返回目的地将是用户界面驱动程序即初始化指令的传输源。

[0096] 从用户界面驱动程序接收驱动程序句柄的应用程序或图形引擎使用由驱动程序句柄标识的打印机驱动程序(图形驱动程序),创建打印数据,并将该打印数据传输到打印机。打印机驱动程序将从图形引擎输出的图形数据移交到打印机驱动程序,以便转换为可以被打印设备解释的以 PDL、XML 等等描述的预定格式的打印数据。

[0097] 作为上文所描述的处理的结果,应用程序可以使用适当的图形驱动程序进行实际的打印。换句话说,由图形引擎输出的数据不再必须转换为可由打印机驱动程序处理的格式的数据。如此,在有多个打印机驱动程序可以供打印机使用的情况下,可以使用与应用程序所使用的图形引擎相对应的打印机驱动程序来进行打印,不管选择了哪一个打印机驱动程序。结果,打印速度可以改善,可以防止由于数据转换而造成的信息损失,并可以提高打印质量。

[0098] [第二实施例]

[0099] 第一实施例是这样的打印系统,其中,在从应用程序执行打印设置的用户界面显示的同时,执行打印机驱动程序的初始化。然而,这不一定适用于所有打印系统。取决于 OS 的规范,可以预见这样的情况:在执行用于配置用户界面驱动程序的打印设置等等的用户界面显示时,不执行打印机驱动程序的初始化处理。在实际执行打印处理时,向打印机驱动程序的图形驱动程序发出初始化指令。在这样的打印系统中,由于不向用户界面驱动程序发出初始化指令,因此,甚至在用户界面驱动程序执行第一实施例的图 10 所示的处理的情况下,也不能切换应用程序所使用的驱动程序。考虑到此情况,本发明是这样安排的,以便响应用户界面显示指令,用户界面驱动程序执行切换到适当的驱动程序的处理。下面将描述的项目之外的系统配置与第一实施例相同,因此将省略其描述。

[0100] 图 11 是最佳地代表第二实施例的特征的图,并示出了根据第二实施例的用户界面驱动程序的配置。在根据第二实施例的打印系统中,用户界面驱动程序不初始化驱动程

序。因此,当与第一实施例的图 7 相比时,第二实施例不包括初始化单元 707、驱动程序初始化单元 706 和驱动程序句柄返回单元 707。相反,已经添加了驱动程序名称改变单元 1101 和驱动程序名称返回单元 1102。

[0101] 类似于第一实施例,根据第二实施例的 GDI 图形驱动程序 502 和 XPS 图形驱动程序 512 不进行任何特殊处理,只作为一般图形驱动程序操作。另一方面,GDI 用户界面驱动程序 203 和 XPS 用户界面驱动程序 213 或公用用户界面驱动程序(下面统称为用户界面驱动程序)执行第二实施例所特有的处理。

[0102] 在从应用程序 201、211 或图形引擎 202、212 接收到设置改变指令时,根据第二实施例的用户界面驱动程序在用户界面显示单元 701 处执行用户界面显示。用户界面设置单元 702 存储由用户输入的打印设置。作为响应,应用程序信息获得单元 703 获取有关正在调用用户界面驱动程序的应用程序 201、211 是 Win32 应用程序 201 还是 WinFx 应用程序 211 的信息。由应用程序信息获得单元 703 和组合判断单元 704 执行的判断处理与在第一实施例中执行的判断处理相同。

[0103] 当组合判断单元 704 判断该组合不适当时,用户界面驱动程序的驱动程序名称改变单元 1101 将适当的驱动程序名称移交到应用程序。更具体地说,在本实施例中,这是使用从用户界面驱动程序移交到应用程序的打印设置信息(GDI 用户界面驱动程序的 DEVMODE, XPS 用户界面驱动程序的 Print Ticket(打印作业单))执行的。换句话说,驱动程序名称改变单元 1101 存储适用于打印设置信息的驱动程序名称,打印设置信息由驱动程序名称返回单元 1102 移交到应用程序。

[0104] 应用程序从移交的打印设置信息中获取驱动程序名称。然后,应用程序使用具有该驱动程序名称的打印机驱动程序执行打印。这就允许避免了处理路径 352 或 353。

[0105] 另一方面,当组合判断单元 704 判断该组合是适当的时,不执行任何特定操作,接收到用户界面显示指令的用户界面驱动程序将打印设置信息移交到应用程序,而不改变已经存储在打印设置信息中的驱动程序名称。

[0106] 图 12 示出了根据第二实施例的用户界面驱动程序的序列。在步骤 1201 中,用户界面显示指令被应用程序移交到用户界面驱动程序。在步骤 1202 中,当组合判断单元 704 判断驱动程序不适当时,在步骤 1203 中,用户界面驱动程序使用驱动程序名称改变单元 1101 将打印设置信息内的驱动程序名称改变为适当的驱动程序名称。然后,在步骤 1204 中,用户界面驱动程序将打印设置信息返回到应用程序,该打印设置信息包括用户界面显示单元 701 和用户界面设置单元 702 处的用户的用户界面设置。应用程序执行与打印设置信息内的驱动程序名称相对应的打印初始化处理 and 打印处理。

[0107] 图 13 是示出了当打印设置用户界面显示指令被移交到用户界面驱动程序时根据第二实施例的用户界面驱动程序的流程图的流程图。当从应用程序 201、211 进行打印时,用户向应用程序 201、211 发出配置打印机驱动程序的打印设置的指令。作为响应,应用程序向指定的打印机驱动程序的用户界面驱动程序发出用户界面显示指令。图 13 所示的处理流程由接收指令的用户界面驱动程序执行。

[0108] 用户界面驱动程序在步骤 13-1 中接收用户界面显示指令,在步骤 13-2 中执行用户界面对话框显示,以接受来自用户的改变打印设置的输入。然后,在步骤 13-3 中,用户界面驱动程序存储由用户输入到打印设置信息中的打印设置值。

[0109] 用户界面驱动程序接下来获取调用用户界面驱动程序的应用程序的名称（可执行文件名称）。基于该名称，在步骤 13-4 中，用户界面驱动程序获取表明正在进行调用的应用程序是 Win32 应用程序还是 WinFx 应用程序的类型信息。此处理类似于在第一实施例中执行的打印处理。基于获取的应用程序类型信息，用户界面驱动程序在步骤 13-5 中判断包括用户界面驱动程序的打印机驱动程序是否适用于该应用程序。如果判断为不适当，则在步骤 13-6 中打印设置信息中的驱动程序名称项被更改为适当的驱动程序名称。在步骤 13-7 中，包括更改的驱动程序名称的打印设置信息被返回到应用程序。

[0110] 当在步骤 13-5 中判断为适当时，用户界面驱动程序在步骤 13-8 中将在步骤 13-3 中创建的打印设置信息原样返回到应用程序。

[0111] 适当的驱动程序名称可以通过例如另外将驱动程序名称关联到并注册到打印机驱动程序类型被在图 8 所示的表中注册到的字段中而获得。例如，搜索相对于获取的应用程序类型被定义为“适当”的驱动程序名称。所获得的驱动程序名称是适用于该应用程序的驱动程序名称。打印设置信息中的驱动程序名称用所获得的驱动程序名称代替。应用程序向在打印设置信息中具有驱动程序名称的打印机驱动程序发出打印初始化指令，并使用该打印机驱动程序执行打印处理。结果，可以避免打印处理路径 352 或 353。

[0112] 以这种方式，甚至在应用程序不向用户界面驱动程序发出初始化指令的情况下，本实施例也能使要由应用程序所使用的驱动程序切换到适当的驱动程序。因此，由图形引擎输出的数据不再必须转换为可由打印机驱动程序处理的格式的数据。结果，打印速度可以改善，可以防止由于数据转换而造成的信息损失，并可以提高打印质量。

[0113] [第三实施例]

[0114] 在第二实施例中，在应用程序基于已经由用户界面驱动程序改变的打印设置信息中所包含的驱动程序名称，执行打印初始化或打印处理的情况下，可以改变驱动程序。然而，还有这样的情况，例如，应用程序将当调用用户界面驱动程序时使用的驱动程序名称存储到本区域中，存储的驱动程序名称的驱动程序也在打印处理过程中使用。在这样的情况下，用户界面驱动程序不能改变应用程序所使用的驱动程序。

[0115] 考虑到上述情况，第三实施例是这样安排的，以便如图 14 所示，图形驱动程序 1401 能够通过 GDI 接口和 XPS 接口接收来自 GDI202 的打印数据和来自 WPF212 的打印数据。因此，只提供一个用户界面驱动程序，这是当使用任何一个接口时激活的。然后，对由 OS 标识的驱动程序属性（表示 GDI 驱动程序或者 XPS 驱动程序的类型信息）进行改变。换句话说，被选为要由应用程序使用的驱动程序的打印机驱动程序的驱动程序属性被存储，以便可从操作系统被访问。基于驱动程序属性信息，就从图形引擎输出的数据的格式是否需要转换作出判断。

[0116] 图 15 是示出了根据本实施例改变驱动程序属性的配置的示意图。在本实施例中，由操作系统参照的驱动程序的属性信息被用户界面驱动程序 1503 重新写入。换句话说，在从应用程序 1501 接收到初始化指令时，用户界面驱动程序 1503 使用与第一实施例相同的方法获取应用程序类型信息。接下来，读取由操作系统保留和管理的表示驱动程序类型等等的驱动程序属性信息 1505，并将其与应用程序类型信息进行比较，以便判断驱动程序属性是否适当。图 8 所示的表可以用于此判断，前提是在打印机驱动程序字段中注册了驱动程序属性信息。如果适当，则继续进行处理，而不作修改。如果不适当，则将由操作系统参

照的驱动程序属性信息 1505 更改为适当的打印机驱动程序的驱动程序属性值。

[0117] 图 16 示出了由用户界面驱动程序执行的处理的过程。在接收到初始化指令时,在步骤 1601 中,用户界面驱动程序获取由操作系统参照的驱动程序属性。在步骤 1602 中,用户界面驱动程序获取应用程序类型信息。在步骤 1603 中,就应用程序类型信息和驱动程序属性信息是否一致(匹配)作出判断。如果不一致,则由操作系统参照的驱动程序属性被重新写入到与应用程序一致的驱动程序的驱动程序属性中。

[0118] 可以看出,在本实施例中,获取由 OS 标识的信息(注册表、文件的可用性等等),如果适当,则执行由用户界面驱动程序进行的处理,不用修改,而如果不适当,则改变表示驱动程序属性的 OS 的信息。因此,由图形引擎输出的数据不再必须转换为可由打印机驱动程序处理的格式的数据。结果,打印速度可以改善,可以防止由于数据转换而造成的信息损失,并可以提高打印质量。

[0119] 此外,本发明还可以应用于包括多个设备(例如,主机计算机、接口设备、阅读器或打印机)的系统,或由单一设备组成的装置(例如,复印机或传真机)。此外,本发明的目的还可以通过这样来实现:将存储了能够实现上文所描述的实施例的功能的程序代码的存储介质提供到系统或装置,并让系统的计算机或装置读出并执行存储在存储介质中的程序。在此情况下,从存储介质中读出的程序代码本身实现了上文所提及的实施例的功能,而程序代码本身和存储了程序代码的存储介质将构成本发明。

[0120] 此外,本发明还包括这样的配置:如前所述的实施例的功能通过由操作系统(OS)等等执行的处理来实现,所述操作系统在计算机上运行,所述计算机基于来自程序代码的指令,执行一部分或所有实际处理。此外,本发明还适用于这样的情况:从存储介质读出的程序代码被写到插入到计算机中的功能增强卡上或内置在连接到计算机的功能增强单元中的存储器上。在这样的情况下,如前所述的实施例的功能还可以通过让内置在功能增强卡或功能增强单元中的 CPU 等等基于来自程序代码的指令执行一部分或所有实际处理。

[0121] 虽然参照示例性实施例描述了本发明,但是应该理解,本发明不限于所公开的示例性实施例。下列权利要求的范围应该有最宽的解释,以便包含所有这样的修改以及等同结构和功能。

[0122] 本申请要求 2006 年 3 月 17 日提交的编号为 No. 2006-075551 的日本专利申请的优先权,这里引入了该申请的全部内容作为参考。

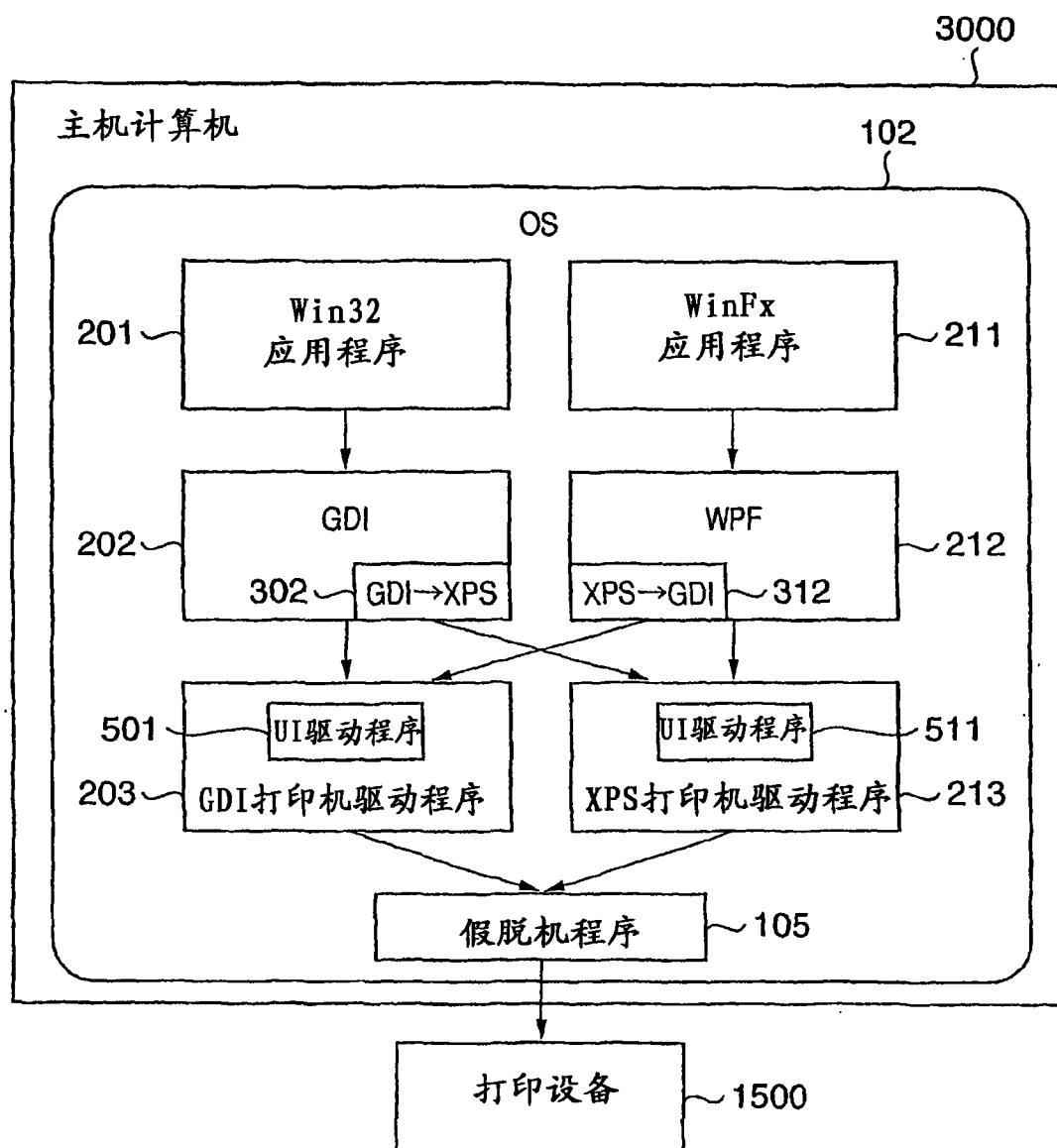


图 1

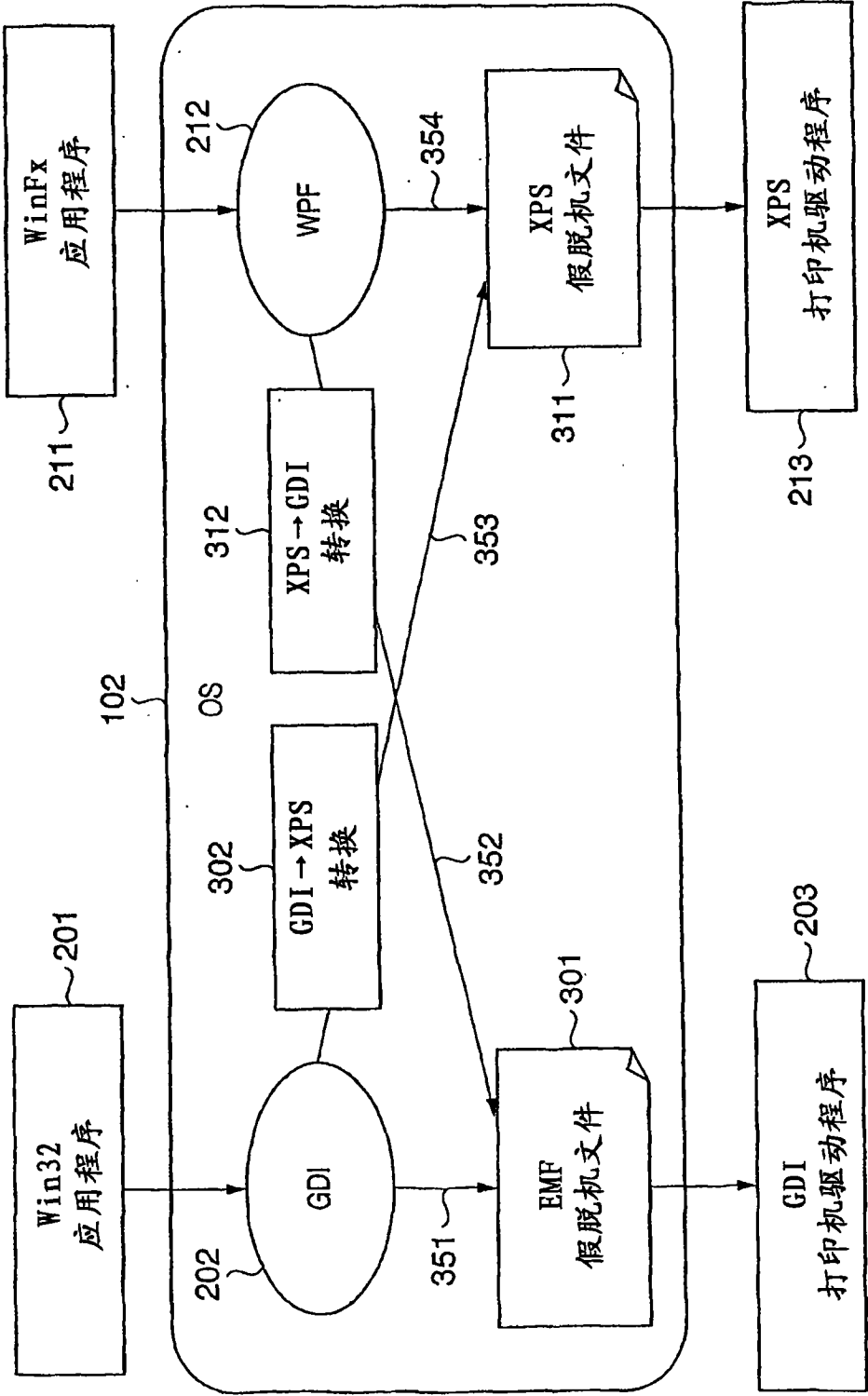


图 2

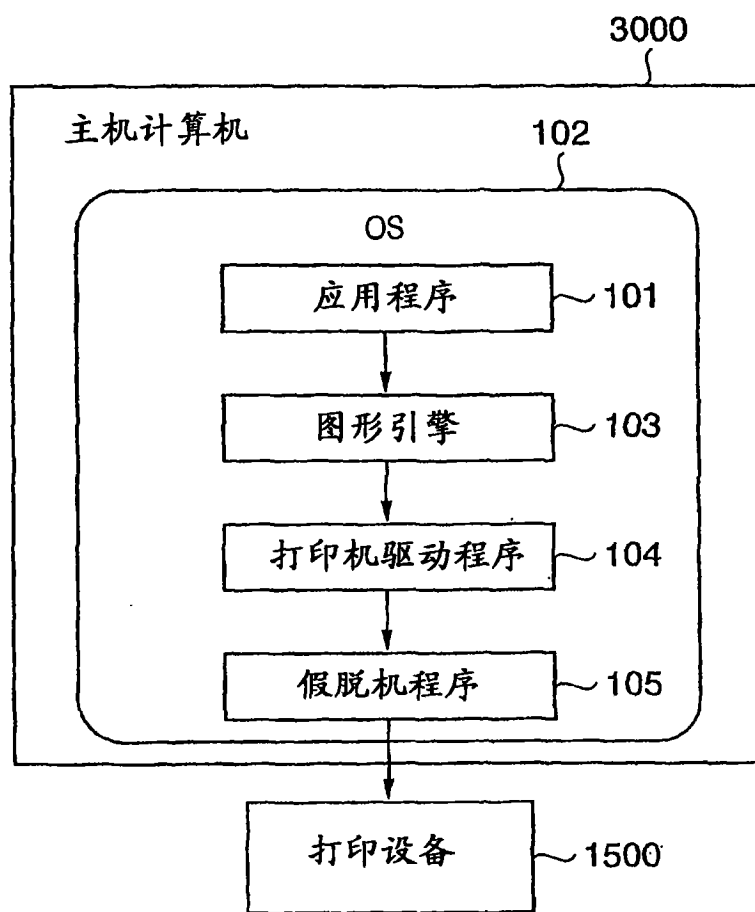


图 3

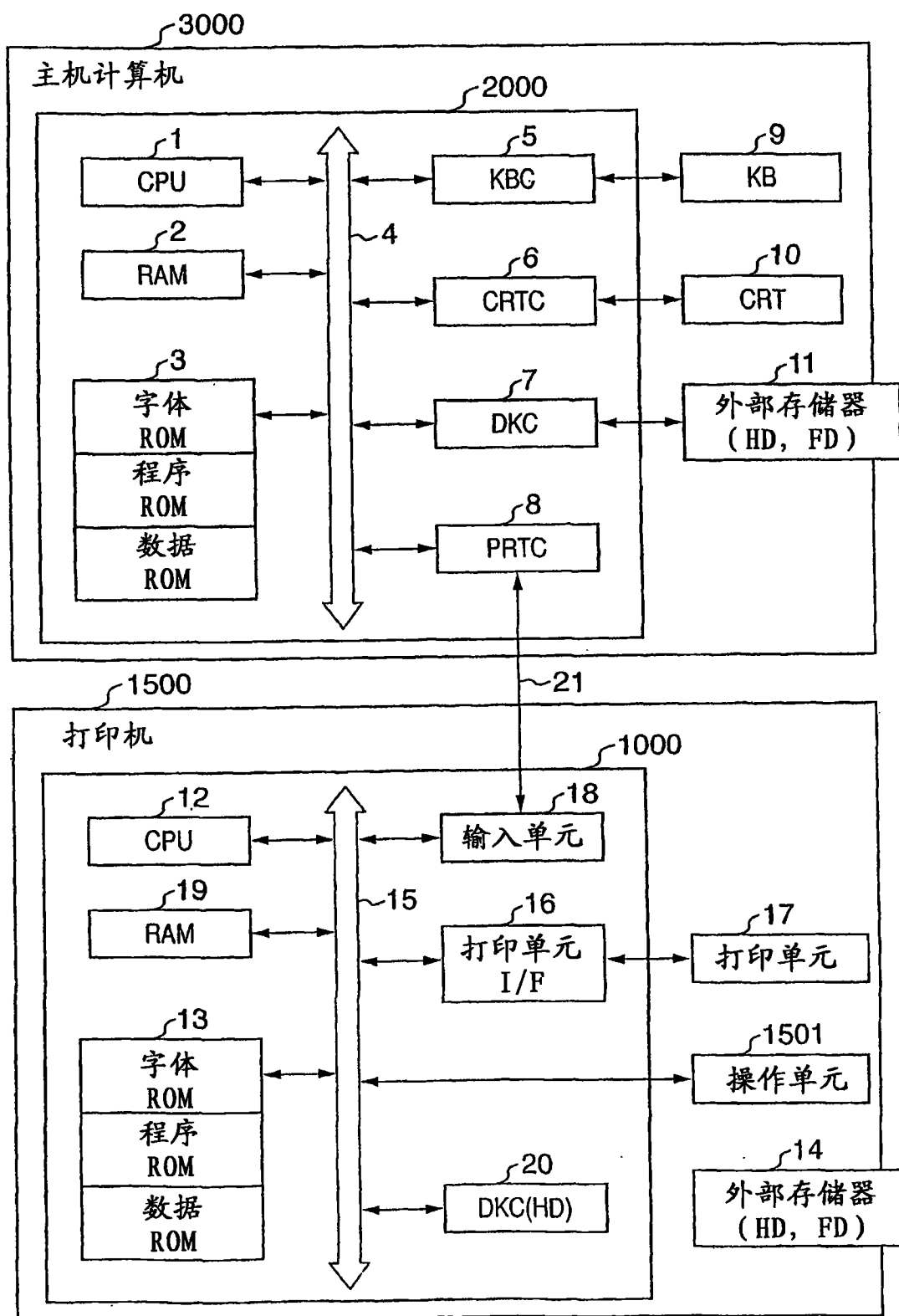


图 4

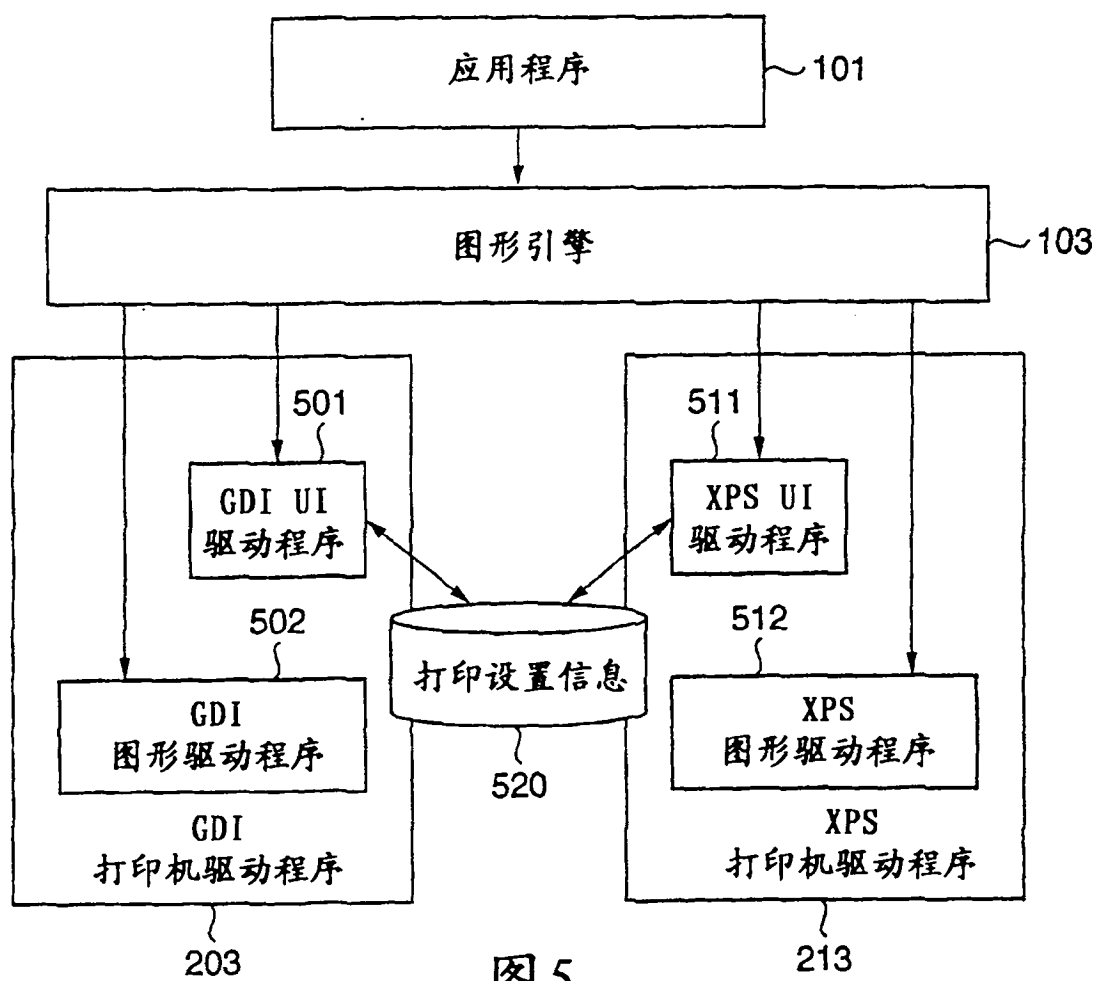


图 5

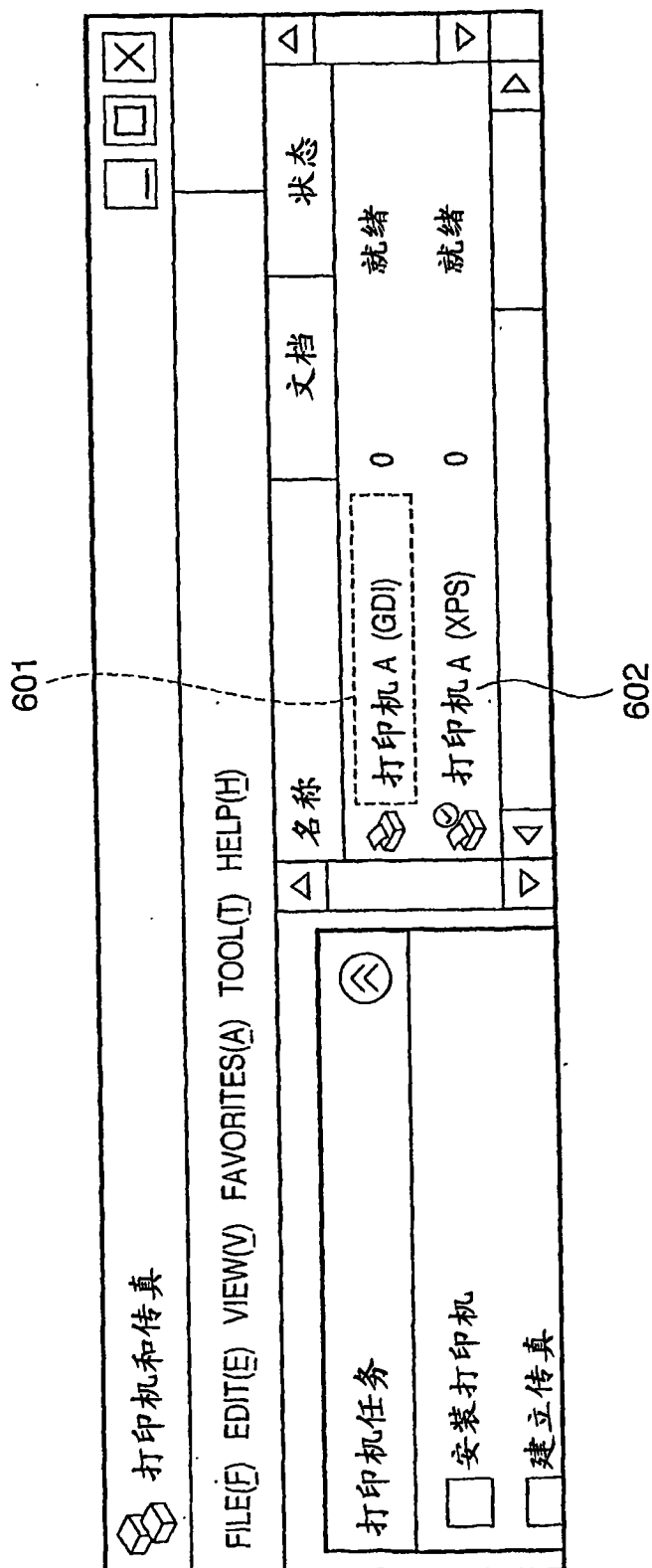


图6

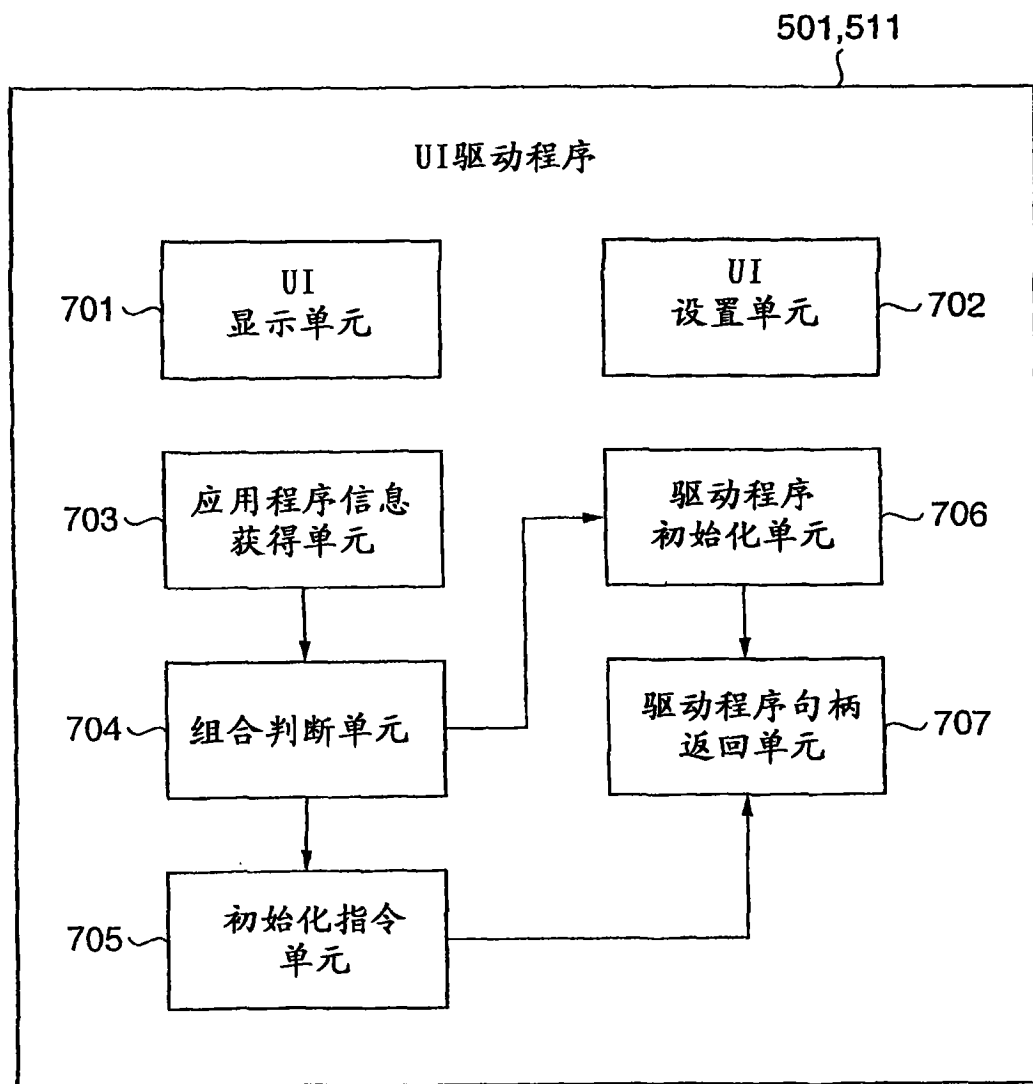


图 7

	201 Win32 应用程序	211 WinFx 应用程序
203 GDI 打印机驱动程序	适当	不适当
213 XPS 打印机驱动程序	不适当	适当

图 8

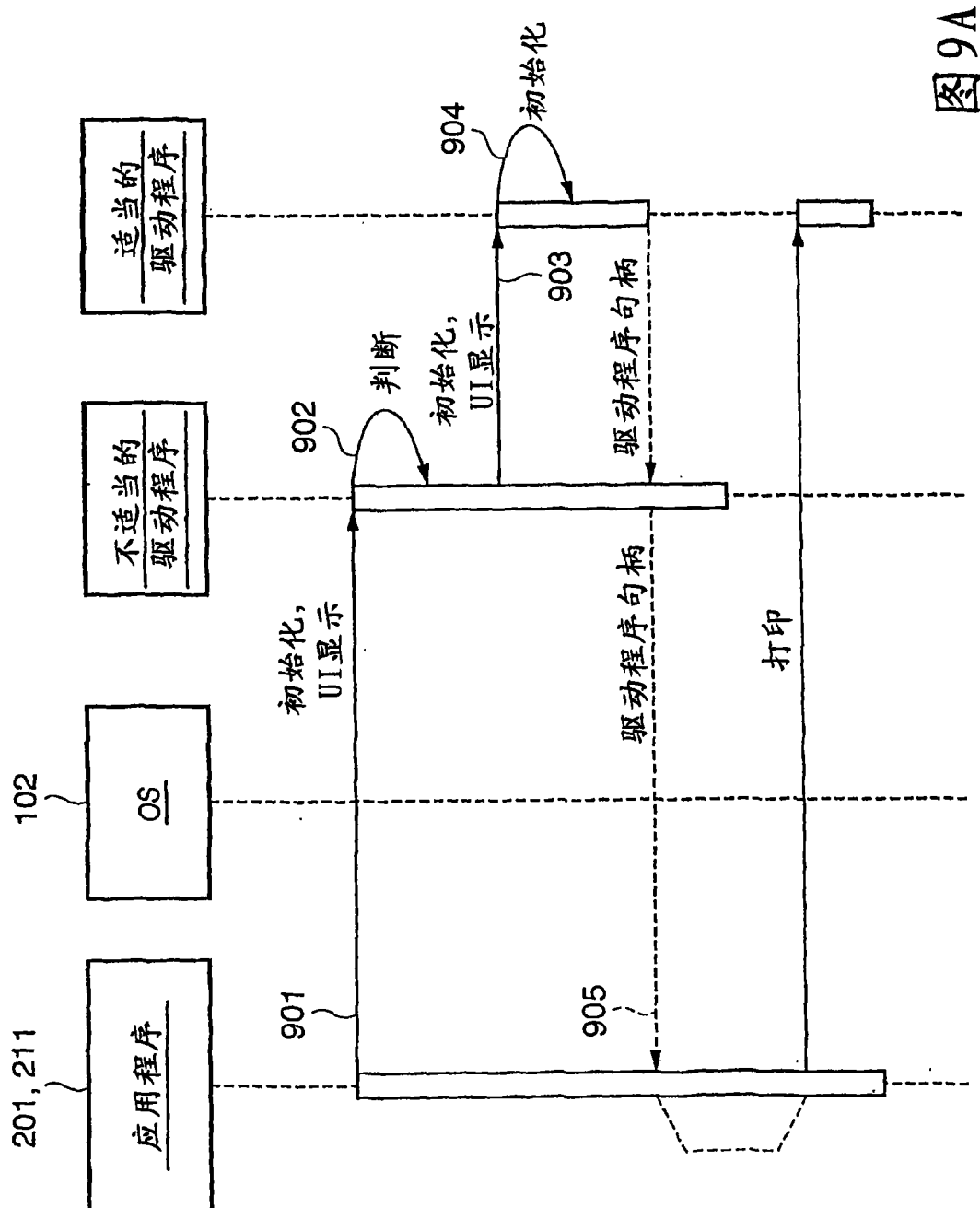


图9A

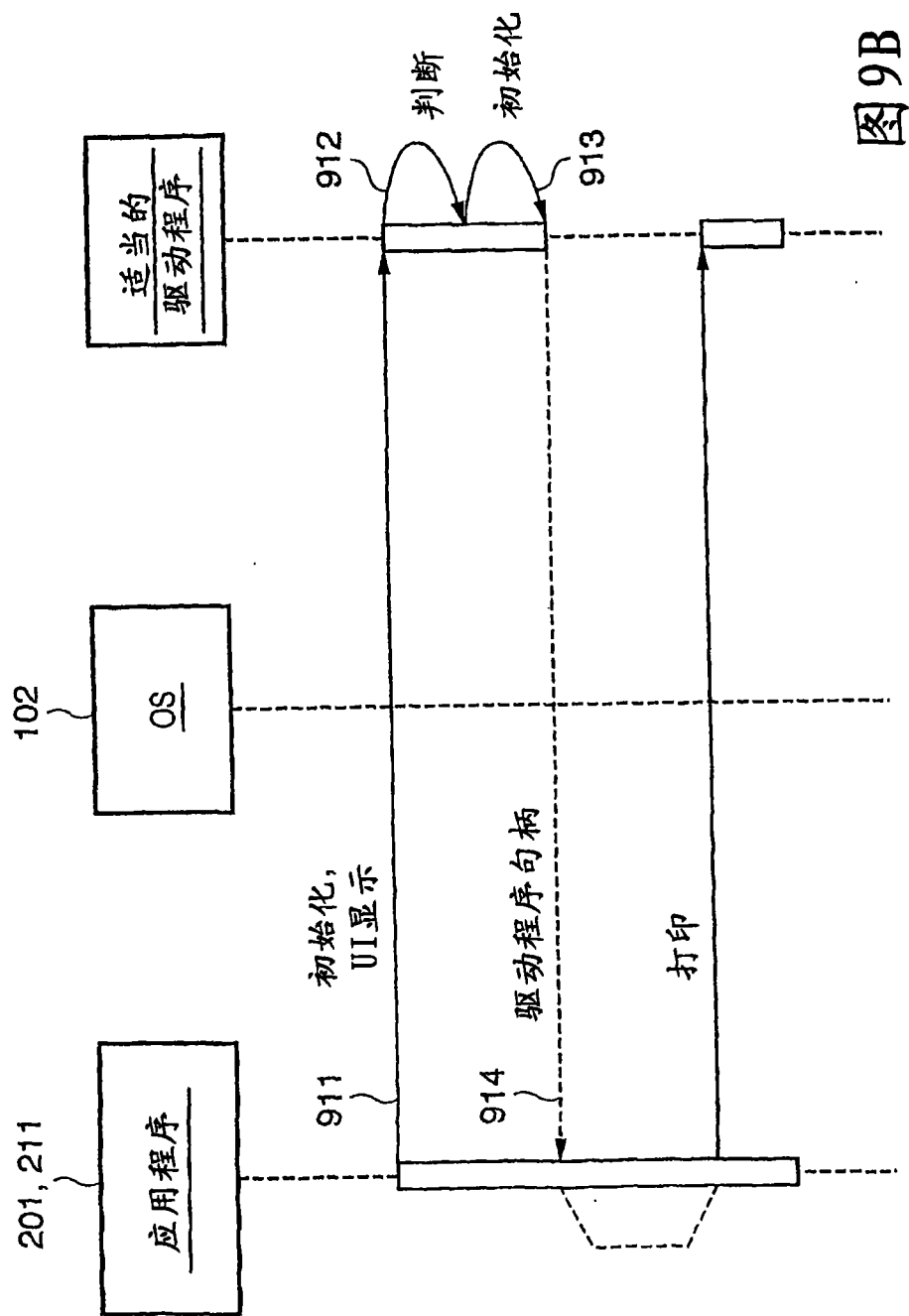


图9B

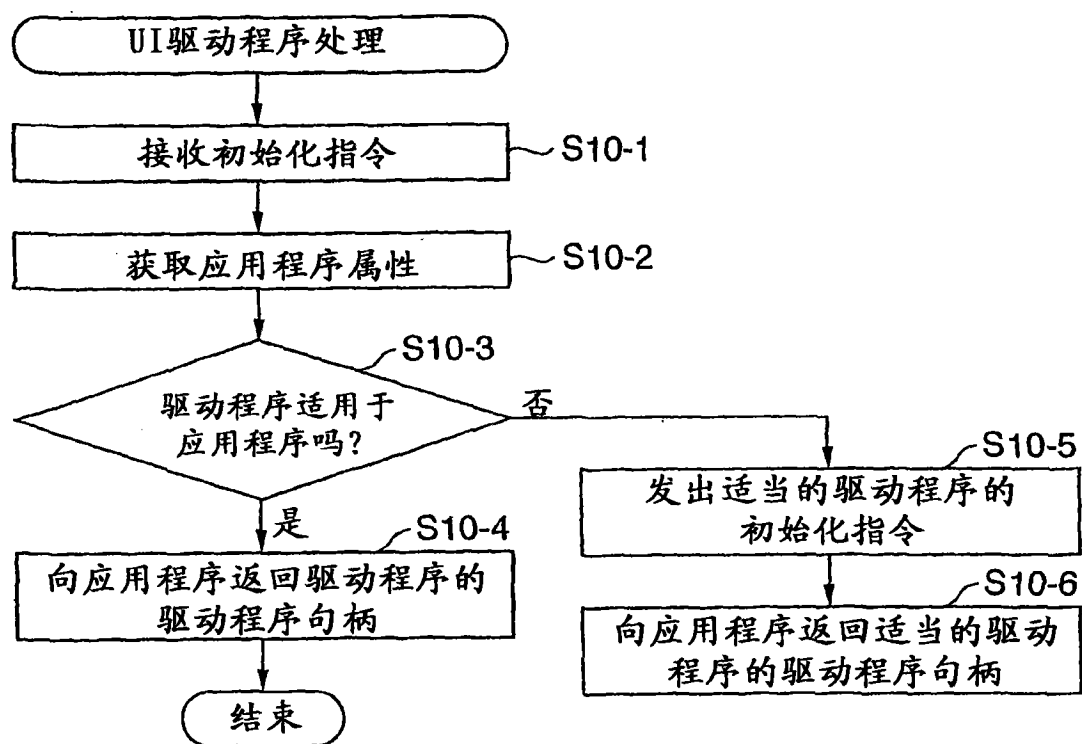


图 10

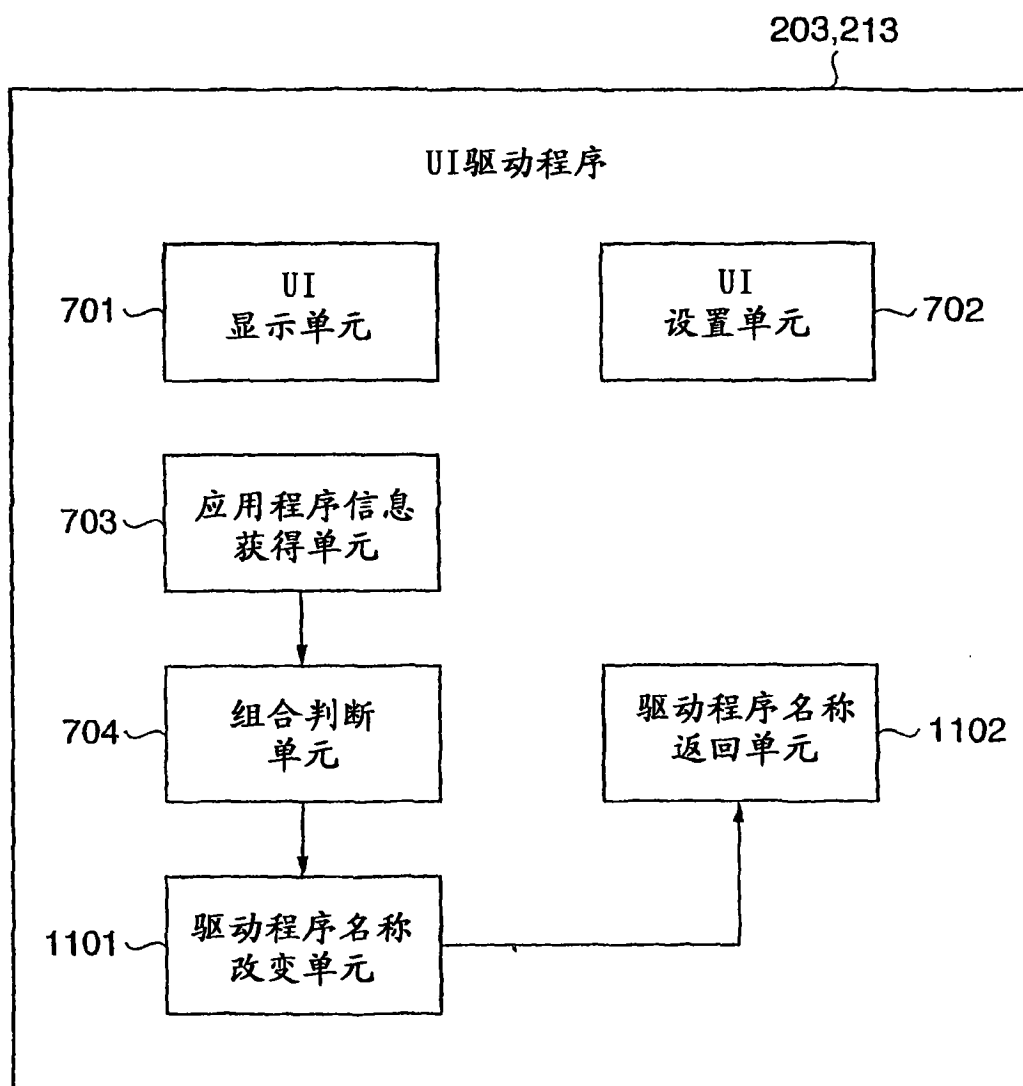
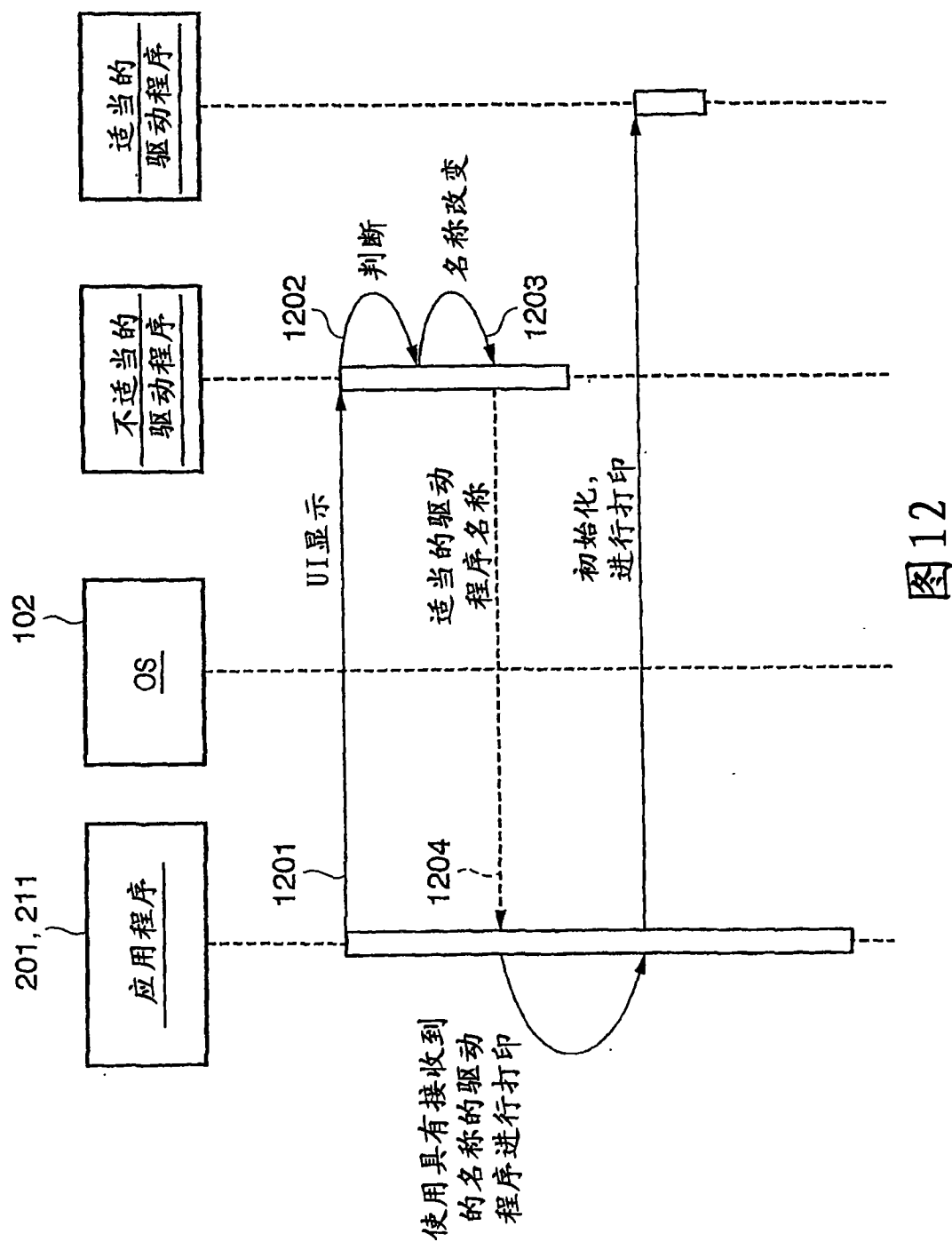


图 11



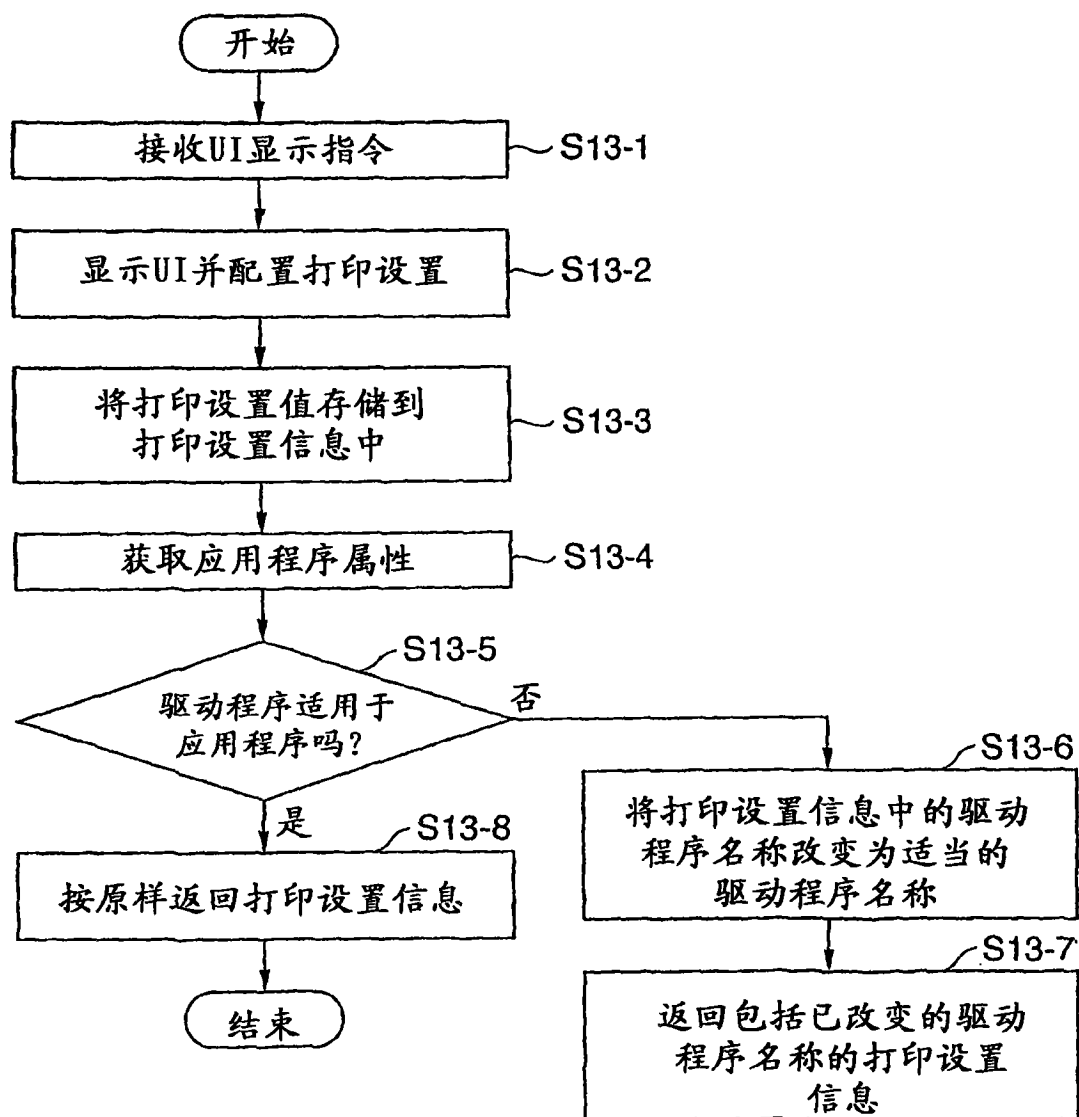


图 13

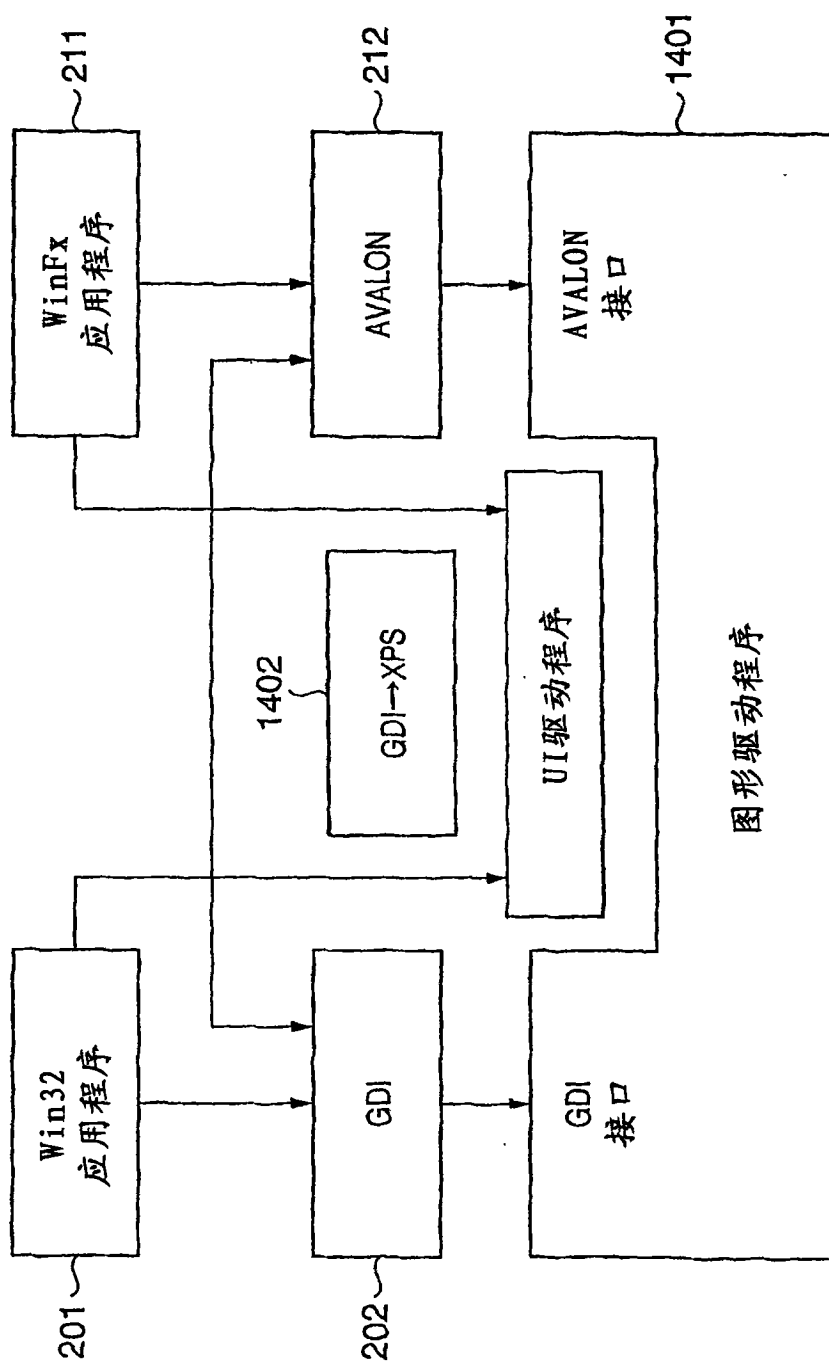


图14

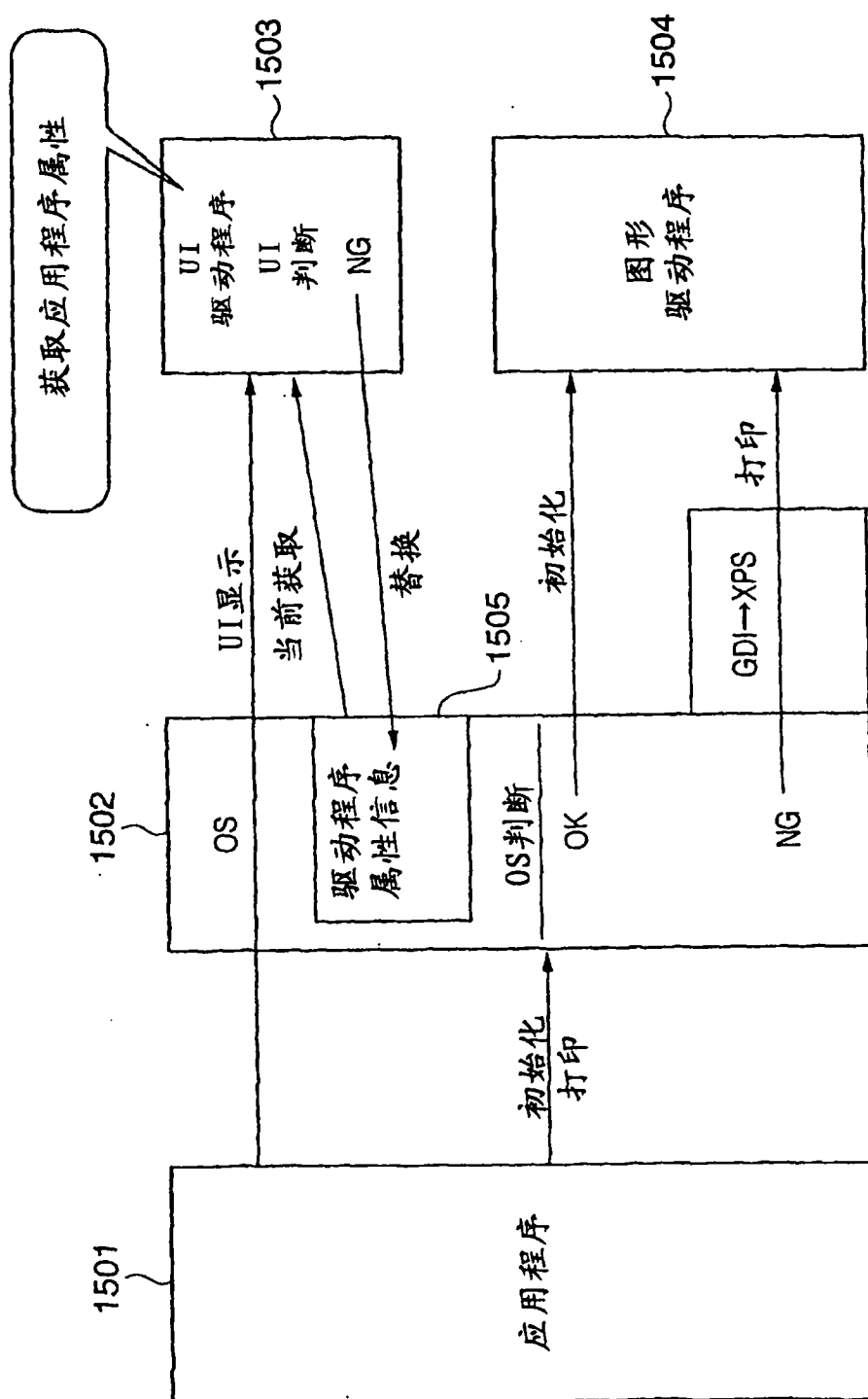


图15

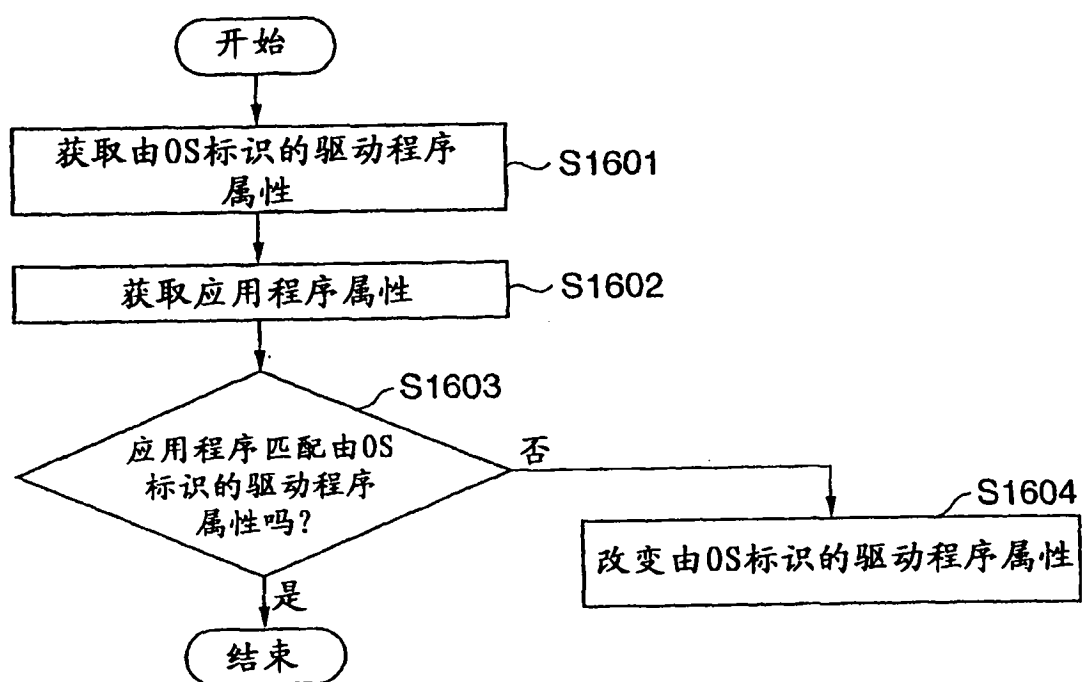


图 16