

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.  
G06F 3/12 (2006.01)



# [12] 发明专利说明书

专利号 ZL 200610066100.X

[45] 授权公告日 2009年3月25日

[11] 授权公告号 CN 100472425C

[22] 申请日 2006.3.28

[21] 申请号 200610066100.X

[30] 优先权

[32] 2005.3.29 [33] JP [31] 095697/2005

[73] 专利权人 佳能株式会社

地址 日本东京都

[72] 发明人 斋藤惠

[56] 参考文献

US6831752B1 2004.12.14

US2003/0112456A1 2003.6.19

US5339432A 1994.8.16

CN1326569A 2001.12.12

US2004/0111418A1 2004.6.10

US2003/0132956A1 2003.7.17

US2003/0053105A1 2003.3.20

审查员 张文

[74] 专利代理机构 北京怡丰知识产权代理有限公司

代理人 于振强

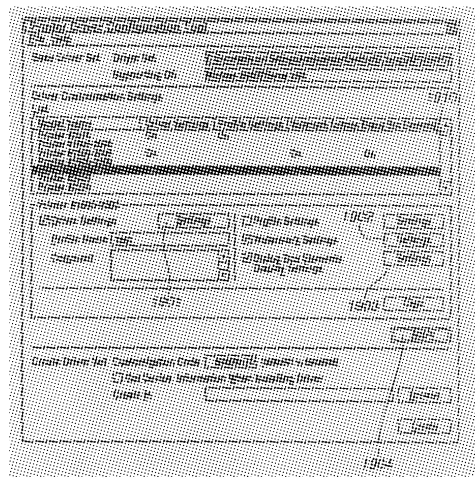
权利要求书4页 说明书24页 附图15页

[54] 发明名称

信息处理装置和方法

[57] 摘要

本发明提供一种信息处理装置，配置为通过设置属性定制适于控制打印机的设备驱动。该信息处理装置包括规定单元，适于规定多个设备驱动；以及设置单元，适于响应于一个设备驱动定制指令而在由规定单元规定的多个设备驱动上设置属性信息，其中所述属性信息对应于设备驱动。



1. 一种信息处理装置，用以执行定制工具并且配置为通过设置其属性信息定制适于控制设备的设备驱动，包括：

规定单元，适于通过执行定制工具而规定多个设备驱动；

确定单元，适于从所述规定单元所规定的设备驱动中确定所述定制工具能够定制的设备驱动；以及

设置单元，适于当设备驱动定制的执行被指示给定制工具时在由所述确定单元确定为可定制的多个设备驱动中为设备驱动定制设置属性信息。

2. 根据权利要求1的一种信息处理装置，其中该设置单元确定对应于多个设备驱动的属性信息是否被设置，以及

其中该设置单元将属性信息写在信息文件中。

3. 根据权利要求1的一种信息处理装置，进一步包括显示控制单元，适于根据对应于设置的属性信息的数据将定制的用户界面窗口显示在显示单元上。

4. 根据权利要求3的信息处理装置，其中显示控制单元根据由设置单元设置的属性信息控制在用户界面窗口中的项目的有效指示，无效指示以及不显示指示中的至少一个。

5. 根据权利要求4的信息处理装置，其中当设备驱动被激活时，显示控制单元根据通过参考对应于设备驱动的信息文

件而获取的属性信息来显示用户界面窗口。

6. 根据权利要求1的信息处理装置，其中该多个设备驱动分别对应于多个设备型号并且被包括在同一个设备驱动设置中，并且

其中该设置单元将属性信息写到可以由多个设备驱动读取的设置文件中。

7. 根据权利要求1的信息处理装置，其中规定单元从存储在预定的存储器中的预定的数据读取设备驱动列表或是设备型号的列表，并且其中所述规定单元规定一个设备驱动，设置单元对于该设备驱动设

置属性信息。

8. 根据权利要求1的信息处理装置，其中规定单元规定一个设备驱动，设置单元根据外部选择指令对于该设备驱动设置属性信息。

9. 根据权利要求1的信息处理装置，其中该设备是打印机。

10. 一种方法，用以执行定制工具并且配置为通过设置其属性信息定制用于控制设备的设备驱动，包括：

规定步骤，用于通过执行定制工具而规定多个设备驱动；

确定步骤，用于从所述规定步骤所规定的设备驱动中确定所述定制工具能够定制的设备驱动；以及

设置步骤，用于当设备驱动定制的执行被指示给定制工具时，在由确定步骤确定为可定制的多个设备驱动中为设备驱动定制设置属性信息。

11. 根据权利要求10的方法，其中，在设置步骤，确定对应于多个设备驱动的属性信息是否被设置，以及

其中在该设置步骤，将属性信息写在信息文件中。

12. 根据权利要求10的方法，进一步包括显示控制步骤，用于根据对应于设置的属性信息的数据将定制的用户界面窗口显示在显示单元上。

13. 根据权利要求12的方法，其中在显示控制步骤，根据在设置步骤设置的属性信息控制在用户界面窗口中的项目的有效指示，无效指示以及不显示指示中的至少一个。

14. 根据权利要求13的方法，其中在显示控制步骤，当设备驱动被激活时，根据通过参考对应于设备驱动的信息文件而获取的属性信息来显示用户界面窗口。

15. 根据权利要求10的方法，其中该多个设备驱动分别对应于多个设备型号并且被包括在同一个设备驱动组中，并且

其中在设置步骤，属性信息被写到可以由多个设备驱动读取的设置文件中。

16. 根据权利要求10的方法，其中在设置步骤，从存储在预定的存储器中的预定的数据读取设备驱动列表或是设备型号的列表，并且

其中在规定步骤中，规定一个设备驱动，对于该设备驱动在设置步骤中设置属性信息。

17. 根据权利要求 10 的方法，其中在规定步骤，根据外部选择指令规定一个设备驱动，在设置步骤对于该设备驱动设置属性信息。

18. 根据权利要求 10 的方法，其中该设备是打印机。

19. 一种信息处理装置，配置为创建包括用以安装多个设备驱动的驱动信息的安装设置，包括：

规定单元，适于规定多个将被定制的设备驱动；

确定单元，适于从所述规定单元所规定的设备驱动中确定定制工具能够定制的设备驱动；

设置单元，适于通过对于由所述确定单元确定为可定制的多个设备驱动的每一个定制打印设置数据而设置属性信息；以及

产生单元，适于产生包括多个且每一个都具有由所述设置单元设置的属性信息的设备驱动的安装设置。

20. 根据权利要求 19 的信息处理装置，其中所述的规定装置通过使用预先读进定制程序中的驱动信息规定该多个设备驱动。

21. 根据权利要求 19 的信息处理装置，其中驱动信息用于该多个具有由所述设置单元设置的属性信息的设备驱动的安装。

22. 一种信息处理方法，配置为创建包括用以安装多个设备驱动的驱动信息的安装设置，包括：

规定步骤，规定多个将被定制的设备驱动；

确定步骤，适于从所述规定步骤所规定的设备驱动中确定定制工具能够定制的设备驱动；

设置步骤，通过对于在所述确定步骤确定为可定制的多个设备驱动的每一个定制打印设置数据而设置属性信息；以及

产生步骤，产生包括多个且每一个都具有由所述设置单元设置的属性信息的设备驱动的安装设置。

23. 根据权利要求 22 的信息处理方法，其中在所述的规定步骤通过使用预先读进定制程序中的驱动信息规定该多个设备驱动的每一个。

---

24. 根据权利要求 22 的信息处理方法, 其中驱动信息用于该多个具有在所述设置步骤设置的属性信息的设备驱动的安装。

## 信息处理装置和方法

### 技术领域

本发明涉及一种信息处理技术或定制设备驱动。

### 背景技术

至今为止，一直有对于根据打印系统的安装环境来定制用于控制打印机的打印控制程序（打印机驱动）的需求。由于打印系统的用户环境和安装场所的不同状态使得这种需求在近年来变得更加迫切。

为了满足这种需求，传统上，提供了一种在打印机驱动的用户界面（此后简称为“UI”）上的定制技术（定制系统），其使得能够改变为打印机中的打印机驱动而设置的缺省值，并且使得能够通过指定项目而达到输入控制（此后称为禁止处理），其设置不能由用户根据用户条件而改变。在这种控制技术下，对应于指定项目的输入数据被拒绝。

相关的技术在例如，日本专利申请公开No.8-278871以及No.2003-208276中有所描述。

然而，在这种定制系统中，首先必须在支持打印机驱动的操作系统（OS）中为打印机驱动的安装创建安装设置。在这种环境下，用于创建定制的打印机驱动的安装设置的工具必须被激活。

同样，即使当定制涉及一个驱动设置的多个打印机驱动时，定制程序每次只能定制一个打印机驱动（对应于打印机的一个型号）。于是，当定制多个打印机驱动时，程序首先必须定制该多个打印机驱动中的一个，该一个打印机驱动对应于打印机的一个型号，并且接着该程序随后地且相似地定制下一个打印机驱动。

这是由于传统的定制工具只能定制已经安装在一信息处理装置中的驱动。为了定制另一个驱动，该工具必须在第一驱动被安装在装置

中之后被重新激活。

在管理员创建几个定制的驱动的情况下，管理员应当重复执行对应于每种打印机的类似的定制过程，其中每一个定制驱动对应于网络环境中提供的不同种类的打印机。这就给管理员强加了很大的负担。同时创建定制的驱动的效率较低。

## 发明内容

本发明的一个目的在于增加定制设备驱动的可操作性。

本发明的另一个目的在于在一个定制处理中对多个打印驱动进行定制。

在本发明的一个方面中，一种信息处理装置，用以执行定制工具并且配置为通过设置其属性信息定制适于控制设备的设备驱动，包括：规定单元，适于通过执行定制工具而规定多个设备驱动；确定单元，适于从所述规定单元所规定的设备驱动中确定所述定制工具能够定制的设备驱动；以及设置单元，适于当设备驱动定制的执行被指示给定制工具时在由所述确定单元确定为可定制的多个设备驱动中为设备驱动定制设置属性信息。

此外，根据本发明的另一方面，一种方法，用以执行定制工具并且配置为通过设置其属性信息定制用于控制设备的设备驱动，包括：规定步骤，用于通过执行定制工具而规定多个设备驱动；确定步骤，用于从所述规定步骤所规定的设备驱动中确定所述定制工具能够定制的设备驱动；以及设置步骤，用于当设备驱动定制的执行被指示给定制工具时，在由确定步骤确定为可定制的多个设备驱动中为设备驱动定制设置属性信息。

此外，根据本发明的另一方面，一种信息处理装置，配置为创建包括用以安装多个设备驱动的驱动信息的安装设置，包括：规定单元，适于规定多个将被定制的设备驱动；确定单元，适于从所述规定单元所规定的设备驱动中确定定制工具能够定制的设备驱动；设置单元，适于通过对于由所述确定单元确定为可定制的多个设备驱动的每一个定制打印设置数据而设置属性信息；以及产生单元，适于产生包括多个且每一

个都具有由所述设置单元设置的属性信息的设备驱动的安装设置。

此外，根据本发明的另一方面，一种信息处理方法，配置为创建包括用以安装多个设备驱动的驱动信息的安装设置，包括：规定步骤，规定多个将被定制的设备驱动；确定步骤，适于从所述规定步骤所规定的设备驱动中确定定制工具能够定制的设备驱动；设置步骤，通过对于在所述确定步骤确定为可定制的多个设备驱动的每一个定制打印设置数据而设置属性信息；以及产生步骤，产生包括多个且每一个都具有由所述设置单元设置的属性信息的设备驱动的安装设置。

本发明的进一步特征将从以下参考附图对示例性实施例进行的描述中变得明显。

#### 附图说明

并入并且构成说明书的一部分的附图说明了本发明的实施例，并与说明书一起用以解释本发明的原理。

图 1 为表示根据本发明一个实施例的打印系统的配置的方框图；

图 2 为表示在预定应用程序和关于打印处理的程序被激活之后，在程序和数据被加载入主计算机 3000 的 RAM2 中的状态下，RAM2 的存储器映射的视图；

图 3 为表示定制工具的 UI 的视图；

图 4 为表示适于定制初始设置值的打印机驱动 UI 的视图；

图 5 为表示文档属性显示控制设置 UI 的视图；

图 6 为表示定制工具的基本操作的处理流的流程图；

图7为表示描述关于打印机驱动的安装信息的三个INF文件的视图；

图8为表示显示打印机驱动定制UI的操作的处理流的流程图；

图9为表示打印机驱动的通用打印机驱动(UPD)文件的数据格式的视图；

图10A为表示调用将被提供给打印机驱动的应用程序接口(API)的方法的视图；

图10B为表示了API函数和对应的伪API函数之间的关系表格；

图11为表示调用伪Windows(注册商标)API的操作的处理流的流程图；

图12为表示调用作为伪Windows(注册商标)API函数的一个例子的CSPL\_GetVersionEx函数的操作的处理流的流程图；

图13为表示控制用以对对应于定制的打印机驱动而设置属性的UI的操作的处理流的流程图；

图14为表示保持定制信息的一种形式的视图；

图15为表示保持定制信息的另一种形式的视图；

图16为表示保持定制信息的又一种形式的视图；

图17为表示共同定制包括在一个驱动设置中的多个打印机驱动的操作的处理流的流程图；

图18为表示用于定制设备驱动的定制数据的例子的视图；

图19为表示用于定制多个驱动的工具的UI的视图。

### 具体实施方式

以下将参考附图对本发明的实施例进行详细的描述。在以下对实施例的描述中，对一种在Windows(注册商标)，也即，用于PC(个人计算机)的OS(操作系统)中定制打印机驱动的方法作为例子进行了描述，其中Windows由微软(注册商标)公司制作。本发明同样可以应用到其它的操作系统，例如Linux或Mac OS X上。

图1为表示根据本发明的实施例的打印系统的配置的框图。该打印

系统包括信息处理装置（以下称为“主计算机”）3000和适于从主计算机3000接收打印数据并且执行打印的打印机1500。

在主计算机3000中，CPU（中央处理单元）1根据存储在RAM（随机存取存储器）2上的程序对连接到系统总线4上的设备进行全面地控制。RAM2也用作CPU1的主存储器和工作区域。ROM（只读存储器）3存储各种程序和数据并且被分割为适于存储各种字体的字体ROM3a，适于存储根程序和BIOS（基本输入/输出系统）的程序ROM3b，以及适于存储各种数据的数据ROM3c。

键盘控制器（KBC）5适于控制来自键盘（KB）9的键盘输入数据以及点击设备（鼠标（未显示））。CRT控制器（CRTC）6适于控制数据在CRT显示（CRT）10上的显示。盘控制器（DKC）7适于控制到例如硬盘的外部存储器11的存取。打印机控制器（PRTC）8通过双向接口21连接到打印机1500并且适于执行与打印机1500的通信控制。

外部存储器11为例如硬盘（HD），磁光盘（MO），以及floppy（注册商标）盘（FD）。如图所示，外部存储器11存储用户文件并编辑除了操作系统（OS）205，各种应用程序（例如，适于处理其中图形，图像，字符以及表格共存的文档的文档处理程序）201，以及关于打印处理的程序204之外的文件。关于打印处理的程序204适于创建由页面描述语言描述的打印数据，并且可以由同一个系统的多个打印机共同利用。该关于打印处理的程序204还包括打印机控制命令生成模块（以后称为打印机驱动）2041，以及打印机驱动UI控制模块2042。

存储在外存储器11中的应用程序201，其根据本发明包括有定制程序（以后简称为定制工具）被加载到RAM2并且由CPU1执行。CPU1执行空心字开发（或光栅化）并且使得在CRT显示10的屏幕上显示WYSIWYG（所见即所得）。CPU1还根据命令打开各种注册窗口以及执行各种数据处理，该命令由鼠标指针（没有显示）在CRT显示10的屏幕上指定。当执行打印时，用户可以打开一个打印设置窗口（由打印机驱动UI控制模块2042控制）并且在打印机驱动2041上执行打印机设置和打印处理设置，该打印驱动包括打印模式选择。

接着，描述打印机1500的配置。CPU12控制整个打印机1500的操作。RAM19用作CPU12的主要存储器和工作区域，并且还用作输出信息光栅化区域以及环境数据存储区域。另外，RAM19具有NVRAM（非易失RAM）区域并且被配置为使得其容量可以由可连接到扩展端口（未显示）的可选的RAM增加。ROM13包括适于存储各种字体的字体ROM13a，适于存储由CPU12执行的控制程序的程序ROM13b，以及适于存储各种数据的数据ROM13c。输入部分18从主计算机3000接收数据并将数据传输给主计算机3000。打印部分接口（I/F）16适于控制到用作打印机引擎的打印部分17的接口。

存储器控制器（MC）20控制到外部存储器14的存取，外部存储器14可以包括，例如，硬盘（HD），磁光盘（MO），floppy（注册商标）盘（FD），以及/或IC（集成电路）卡。外部存储器14存储字体数据，仿真程序，以及形式数据。当外部存储器14，例如硬盘，没有连接时，主计算机3000中使用的信息可以被存储在ROM13的数据ROM13c中。外部存储器14的数目并不限于1而是可以有多个。例如，打印机1500可以被配置为使得可选的字体卡被添加到外部字体，并且可以连接有多个存储有适于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。

在操作部分1501提供有操作面板，适于接受用户的操作。在该操作面板上布置有用于操作的开关和LED指示器（未显示）。该操作部分1501可以具有NVRAM（非易失RAM（未显示）），适于存储从操作面板发送的打印机设置信息。

CPU12通过打印部分接口16，根据存储在ROM13的程序ROM13b上的控制程序将图像信号作为输出信息输出到打印部分（打印机引擎）17。CPU12可以通过输入部分18执行与主计算机3000的通信，并且配置为接收从主计算机3000发送的打印数据并且将从打印机1500输出的信息通知给主计算机3000。

图2是表示在预定应用程序和关于打印处理的预定程序被激活之后，在程序和数据被加载入主计算机3000的RAM2中的状态下，RAM2

的存储器映射的视图。

如该图中所示，BIOS206，OS205，应用程序201，关于打印处理的程序204以及相关数据203被加载到RAM2中。还保证了空存储器区202。因而，可以执行应用程序201和关于打印处理的程序。包括在关于打印处理的程序内的打印机驱动UI控制模块2042（参见图1）响应于来自用户的打印设置命令在CRT显示10的屏幕上显示打印设置窗口并且使得用户通过键盘9执行设置。

图3是表示定制工具UI的例子的视图。该定制工具在Windows（注册商标）2000/XP/Server2003上操作并且定制，例如，用于Windows（注册商标）98/ME，Windows（注册商标）NT4.0，以及Windows（注册商标）2000/XP/Server2003的打印机驱动。然而，根据本发明的支持OS并不限于此。对应于OS的应用程序接口（API）的操作的仿真可以利用伪API（将在稍后进行详细描述）而无需实际安装打印机驱动。API的仿真可以通过改变或者添加各个函数的变量的设置而被支持。

当用户在如图3所示的菜单301上指定“开”时，显示出用以指定将被定制的驱动设置的文件夹的对话框（未显示）。这样，可以指定将被定制的驱动的存储器位置。被指定的文件夹在显示区域302上被指示。对应于指定的驱动设置的OS在显示区域303上被指示。

在有多个对应于打印机的型号的打印机驱动（在图14到16中称为“PRINTER MODEL”）被存储在一个驱动设置的文件夹中的情况下，当创建一个定制的驱动时，会列出多个打印机驱动（参见图19）。定制工具确定定制信息（或定制数据）是否对应于每个列出的打印机驱动而设置。接着，该定制工具创建与打印机驱动相关的定制的驱动，定制数据被设置到该定制的驱动上。对这个处理在对图6所示的步骤S608和图17所示的流程图的描述中进行了详细的描述。这个处理可以通过使用伪API而实现。在这种情况下，将被定制的驱动并不通过安装在OS上从而可以由OS认可而进入操作状态。这样，驱动就可以一个接着另一个地被定制。在图19所示的显示区域1910，显示了打印机驱动列表。每一个将被定制的驱动被选择并且用反转视频显示。接

着，按钮1901到1903被推下从而定制驱动。随后，可以通过按下定制按钮1904以生成多个定制的驱动。因此，通过按下定制按钮1904一次就可以提供定制工具并且该定制工具用作包括多个定制的驱动的安装设置。

定制数据包括各种数据，例如控制数据，其用于执行设备驱动的UI的控制项目的显示/不显示，并且用于设置设备驱动的值，如图18所示。

图18显示了在设备模式（DEVMODE）中的数据结构的例子。

该图表示了存储代表纸张方位的数据（dmOrientation）、代表纸张尺寸的数据（dmPapersize）、代表纸张宽度的数据（dmPaperWidth）以及代表副本数量的数据（dmCopies）的方式。此外，该DEMODE数据结构存储在注册表中。代表缺省打印设置环境的数据或“配置文件”（以下将进行描述）被读取，并且可以通过将数据写入存储在注册表中的DEMODE数据结构中或是从该DEMODE数据结构中读取数据而被反射到打印机驱动2041的UI。

“配置文件”为对应于打印机驱动2041的多个打印设置数据的集合。用户可以通过指定“配置文件”获得想要的操作环境，该“配置文件”包括在打印机驱动2041的UI中的设置数据的集合。例如，在用户通常使用水印和短纤的情况下，包括水印设置数据和短纤设置数据的设置数据的集合预先登记为配置文件。这样，用户可以仅通过选择“配置文件”将多个设置项目设置在想要的值上而实现打印。该定制工具保持这种数据作为定制数据并将这种数据写到UPD文件或GPD（通用打印机驱动）文件，对其将在后面进行描述，该定制数据为属性信息的一个例子。在本实施例中打印机驱动可以通过修改UPD文件以及GPD文件而得到定制。

在列表视图305中，规定指示“On”对应于每个打印机型号而显示，该打印机型号对应于将被定制的打印机驱动以及定制信息。当复选框310被选中，按钮310变得有效。当按钮311被按下，显示出将被定制的打印机驱动的UI（见图4）。打印设置数据（文档属性数据）以及设

备设置数据（打印机属性数据）的初始值可以被设置。

当复选框320被选中时，按钮321变得有效。当按钮321被按下时，显示出用于定制在打印机驱动UI上显示控制项目的方法的UI（见图5）。这样，在打印机驱动UI上显示控制项目的方法可以被设置。

当在如图3所示的UI窗口中按下设置按钮350时，从图4或图5所示的UI窗口设置的关于定制的数据被作为定制信息（或定制数据）保持在应用程序（定制工具）中。该定制工具可以将该定制信息保持在，例如，用于描述打印机驱动的安装信息的文件（称为“INF文件”（将在以后参考图7进行描述））中。该定制工具可以根据规定信息确定打印机驱动是否已经被定制。图14到16表示了多个存储在一个驱动设置中的打印机驱动。该定制工具确定对应于每个打印机驱动的定制信息的存在/缺乏，并且定制其定制信息已经准备好的打印机驱动（参见图6所示步骤S608，并且参见图17）。

定制信息以打印机型号名称（或对应于打印机型号的打印机驱动）和定制信息的形式保持着。定制信息可以图15或16所示的形式保持着。在采用如图15所示的形式的情况下，代表着“CUSTOMIZED”或“UNCUSTOMIZED”的规定信息与代表着打印机型号名称（或对应于打印机型号的打印机驱动）以及定制信息被合并地保持着。

在采用如图16所示的数据保持的形式的情况下，代表着“UNCUSTOMIZED”的规定信息（例如，零清除）由代表着未定制的打印机的型号名称的数据保持着。在定制工具确保了关于打印机型号的在INF文件中的定制保持区域的情况下，当按下设置按钮350时，定制信息对应于每个未定制的驱动，而不是对应于规定信息而被保持着。根据保持在定制保持区域内的规定信息以及定制信息，定制工具可以确定相关的打印机型号是否被定制。

通过按下按钮330，显示出用于指定定制驱动的创建位置的对话框（未显示）。这样，定制驱动的创建位置可以被指定。

当在执行各种设置之后按下按钮340，定制的打印机驱动根据代表着设置的数据而被创建。

图4表示了用以定制打印机驱动(以下简称为驱动)的初始值的打印机驱动UI,其中打印机驱动通过按下如图3所示的UI中的按钮311而显示。

定制工具在CPU1的控制下用作信息处理装置的设置部分。该设置部分在CRT显示的屏幕上显示设置窗口,如图4所示的UI窗口,也就是,用于设置文档属性和设备属性的设置窗口。

在用以设置通常安装在Windows(注册商标)NT4.0,或Windows(注册商标)2000/XP/Server2003上的打印机驱动的UI窗口中,用于设置文档的文档属性UI以及用以设置设备的打印机属性UI不能同时被显示。然而,用于根据本发明实施例的打印机驱动的定制的打印机驱动UI由利用打印机驱动的模块的定制工具显示。根据本发明的实施例,用于设置文档的文档属性UI以及用以设置设备的打印机属性UI作为一个UI窗口同时显示(见图4)。如图4所示,有四种UI窗口,也即,页面设置UI窗口401,结束UI窗口402,纸张馈送UI窗口403,以及打印质量UI窗口404可以作为文档属性UI窗口而显示。设备设置UI405可以作为打印机属性UI而显示。这样,文档属性以及打印机属性的初始值就可以通过UI窗口401到405而被设置。

图4所示的UI配置为使得由用户为了设置而输入的不被定制的数据不被接受。这就由,例如由参考标记410指定的变灰的显示来表示。

图5表示显示了文档属性显示控制设置UI的视图。当如图3所示的按钮321被压下时,文档属性显示控制设置UI窗口在主计算机3000的CRT显示10的屏幕上显示。在图5所示的例子中,可以执行五种显示控制设置,也就是,页面设置UI窗口501,结束UI窗口502,纸张馈送UI窗口503,打印质量UI窗口504,以及共有UI窗口505。关于页面设置UI窗口501,结束UI窗口502,纸张馈送UI窗口503,打印质量UI窗口504,可以执行对应于图4所示的相关文档属性UI窗口的显示控制设置。共有窗口505可以执行在相关UI窗口之间共有的显示控制设置。

参考标号506指定了控制项目“任务操作”的实际的指示。可以指定有效指示(普通显示),无效指示(在所有的时间控制选项的不能显

示因此使得不能操作控制项目)，以及不显示（隐藏显示）。同样地，有效指示（项目的使能显示，使得控制项目可以被操作），无效指示，以及不显示可以对于其它的控制项目个别地设置以控制每一个文档属性UI的显示。这样，就可以控制对应于文档属性UI窗口的项目的设置数据的显示。当用户输入设置数据，该定制工具将输入数据写到GPD（通用打印机驱动）文件或是UPD文件。无效指示的一个例子是控制项目的变灰显示，该控制项目则不能由用户操作。

该定制工具在CPU1的控制之下作为信息处理装置的显示控制部分进行工作。该显示控制部分可以根据文档的设置属性（包括控制项目）控制UI中的项目的有效指示、无效指示、以及不显示。

接下来，参考图6对定制工具的基本操作进行描述。假定当执行基本操作时，由主计算机3000的CPU1执行的定制工具是执行基本操作的实体。

首先，在步骤S601，用户激活定制工具。执行定制工具的初始化。在步骤S602，定制工具将参考图3描述的UI窗口显示在CRT显示10的屏幕上。在步骤S603，将被定制的打印机驱动的安装设置由用户在图3所示的UI窗口上指定。在步骤S604，检查指定的安装设置。

在安装设置的检查中，定制工具从INF文件701，710以及720（其描述了图7所示的打印机驱动的安装信息）确定OS是否可以支持指定的安装设置，以及所指定的打印机驱动是否将由定制工具进行定制。

信息处理装置配置为通过设置属性（用于执行文档设置的文档属性以及用于执行设备设置的打印机属性的初始值）来定制适于控制打印机的设备驱动。优选地，该信息处理装置包括规定部分，适于规定多个设备驱动；以及设置部分，适于响应于一个设备驱动定制指令而为由规定部分规定的多个设备驱动设置对应于文档或设备属性的属性信息。在这种情况下，该信息处理装置的上述部分通过在CPU1的控制下执行定制工具而执行处理。

图7为表示在每个INF文件中的描述打印机驱动的安装信息的描述的视图。每一个INF文件包括模块描述信息（未显示）。假定根据

模块描述信息，定制工具可以规定一模块，其执行打印机驱动的UI处理。在图7中，参考标号700指的是对于Windows（注册商标）98/ME的INF文件的一部分。参考标号710指的是对于Windows（注册商标）NT4.0的INF文件的一部分。参考标号720指的是对于Windows（注册商标）2000/XP/Server2003的INF文件的一部分。当为INF文件700中的“签名”设置的值730为字符串资源\$CHICAGO\$的情况下，CPU1执行的定制工具确定打印机驱动是对于Windows（注册商标）98/ME的打印机驱动。

同样，当为INF文件710或720中的“签名”设置的值740或750为字符串资源\$Windows（注册商标）NT\$的情况下，该定制工具确定打印机驱动是对于Windows（注册商标）NT4.0或Windows（注册商标）2000/XP/Server2003的打印机驱动。接着，定制工具精炼确定处理并且当对于INF文件710或720中的“DISK1”设置的字符串包括字符串“NT”时，确定该打印机驱动为对于Windows（注册商标）NT4.0的打印机驱动。否则，该定制工具确定该打印机驱动为对于Windows（注册商标）2000/XP/Server2003的打印机驱动。

对应于可定制的打印机驱动的INF文件包括字符串资源指示着该打印机驱动支持定制。这样，当进行确定时，定制工具判断是否有一个字符串资源被包括在对应于可定制的打印机驱动的INF文件中。结果是，定制工具可以确定打印机驱动是否支持定制。

回到图6，在步骤S605，当定制工具确定打印机驱动不支持定制时（步骤S605的否），处理进行到步骤S610，在该处执行错误处理。接着，处理再次返回到步骤S603，在该处指定安装设置。

另一方面，当定制工具在步骤S605确定打印机驱动支持定制（步骤S605的是）时，该处理前进到步骤S606。接着，该定制工具控制CRT6将UI窗口显示在CRT显示10的屏幕上并且使得能够设置打印机驱动的定制，在前面对图3、图4以及图5的描述中已经描述过其中的UI窗口。在前面对图3、图4、图5的描述中的设置的输入数据被用作定制数据。该定制数据被设置在可以由打印机驱动读取的GPD文件中。当打印机

驱动被安装在装置中并且被设置在操作状态中时，打印机驱动读取包含在GPD文件中的设置值。接着，在GPD文件中的定制的值被反射到打印机驱动。例如，预先调整经常使用的打印设置值（例如，“两面”以及“1张2页”）并将其预先地反射给打印机驱动。这样，该经常使用的打印设置值或定制的缺省打印设置值被设置在打印机驱动中。当该驱动被指令将操作状态带到非操作状态时，该定制的缺省打印设置值被设置在驱动中。

当在步骤S607给出创建定制的驱动的命令时，在步骤S608执行定制的驱动的创建。图17是表示了步骤S608执行的实际操作流程图，同时也表示了共同地定制包括在一个驱动设置中的多个打印机驱动的处理的流。

在创建定制的驱动的处理中，定制工具首先在步骤S1701从定制的位置复制安装设置到定制的驱动的创建的位置。接着，为了规定对其进行定制数据的设置的设备驱动，在步骤S1702，定制工具检查定制数据是否对应于按照列在INF文件中的打印机驱动的顺序（例如，在图14所示的情况下，按照打印机型号3，打印机型号7，打印机型号1，以及打印机型号2的顺序）的打印机驱动的每一个而设置。定制工具可以根据规定信息的存在/缺乏确定该定制数据是否被设置（参考图15和图16对其已经进行过描述）。该定制工具可以响应与由例如用户外部地输入的选择指示而规定一设备驱动，对于该设备驱动设置定制数据。在这种情况下，在INF文件中描述的信息由用户输入，该信息代表了例如打印机驱动的种类以及定制数据的存储的位置。

如果定制数据对于对应的打印机驱动而设置（步骤S1703的是），处理就前进到步骤S1704，其中定制工具将定制数据写入包括在定制所复制到的安装设置中的驱动信息文件（也称为UPD文件或GPD文件）。接着，定制的驱动在步骤S1705被创建。随后，如果在步骤S1706确定出还没有对全部的所列的打印机驱动执行检查（步骤S1706的否），则处理返回到步骤S1703，在该处检查定制数据是否为下一个打印机驱动设置。如果设置，定制工具重复将定制数据写入到对应于

相关的打印机驱动的UPD文件的操作。当确定对于所有的所列出的打印机驱动都完成了检查（步骤S1706的是），则图17所示的流程图所显示的处理就结束。接着，处理返回到图6所示的步骤S608。上述处理的结果是，定制工具创建定制的驱动，定制数据从包括在一个驱动设置中的多个打印机驱动被反射到该定制的驱动。

信息处理装置配置为定制设备驱动（打印机驱动），该设备驱动适于通过设置属性（用以执行文档的设置文档属性以及用以执行设备设置的打印机属性的初始值）而控制打印机。该信息处理装置包括规定部分，该规定部分适于规定多个设备驱动，以及设置部分，适于响应于一个设备驱动指令而将对应于文档或设备属性的属性信息设置到由规定部分规定的多个设备驱动。该信息处理装置的上述的每一个部分通过在CPU1的控制之下执行定制工具而执行处理。

图9为表示驱动信息文件（UPD文件或GPD文件）的数据格式，该驱动信息文件为构成打印机驱动的一个文件。如图9所示，UPD文件（以及GPD文件）包括header 901，ID定义字段902，以及数据字段903。该ID定义字段902存储一组存储在数据字段903的数据的ID（识别信息）以及数据的存储的位置（或地址）。对应于ID的数量的数量集在此存储。根据本发明的定制工具改变数据值，该数据值对应于为定制而定义的ID。在将被使用的定制数据的ID没有在ID定义字段902定义时，对应于将被使用的定制数据的新的ID被添加到ID定义字段902。

定制的打印机驱动适于在安装时或是在第一次激活时读取GPD文件（或是UPD文件），定制工具将定制数据写到该GPD文件（或是UPD文件）。定制的打印机驱动也适于读取对应于存储在GPD文件（或是UPD文件）的ID定义字段902的预定ID而存储的数据903。例如，假定指示着“定制的”的ID存储在由定制工具创建的UPD文件的ID定义字段902，并且地址X作为数据的存储位置而存储在该处。

在代表“定制的”的ID信息存储在ID识别字段902的情况下，支持着定制的打印机驱动在安装完成或是第一次激活时适于读取对应于该信息的地址。这样，在安装完成或是第一次激活时，且当“定制的”包括

在UPD文件的ID识别字段902中时，该打印机驱动从ID识别字段902读取相应的数据地址X。随后，打印机驱动进一步从数据字段903中的地址X读取设备模式的数据结构，其为实际的驱动设置数据。地址X可以是绝对地址。还可以的是，数据格式可以被改编因此数据字段903的首地址被存储在头部901，并且从数据字段903的首地址的偏离地址被设置在地址X。

定制工具作为在CPU1的控制之下作为信息处理装置的显示控制部分工作。该显示控制部分涉及在打印机驱动的激活时打印机驱动的驱动信息文件（UPD文件和GPD文件），并且可以根据在UPD文件中设置的定制数据将UI窗口显示在CRT10的屏幕上。

在本实施例中，定制工具将关于用以设置文档的文档属性的初始值数据（由图4所示的UI窗口401到404设置的数据）、关于用以设置设备的打印机属性的初始值数据（由图4所示的UI窗口405设置的数据）、以及关于在对于文档属性的UI窗口上显示控制的方法的数据（由图5所示的UI窗口设置的数据）作为定制数据写到UPD文件上。该定制的打印机驱动在安装完成时或是在第一次激活时读取由定制工具生成的UPD文件，并且接着根据定制信息进行操作。随后对于定制的打印机驱动就可以轻易地构造操作环境。

回到图6，如果在定制的驱动在步骤S608被创建后用户在步骤S609命令终止，定制工具的操作在步骤S611结束。

以下接着参考图8对由CPU1执行的定制工具对用于打印机驱动的定制的UI的显示处理进行描述。由主计算机3000的CPU1执行的定制工具是执行该操作的实体。

当图3所示的按钮311被按下时，定制工具规定打印机驱动的一个模块，其根据包括在INF文件中的模块描述信息执行UI处理并且在步骤S801将模块加载入存储器是的模块可以被定制工具调用。

接着，在步骤S802，获取由打印机驱动的模块输出的函数地址。在步骤S803，用于定制工具和支持定制的打印机驱动之间的交换的CSPL结构（未显示）被初始化。在CSPL结构中存储有各种信息，因

此打印机驱动的UI可以被控制，即使当打印机驱动没有安装时。

之后，在步骤S804执行显示属性页（用于定制的驱动UI）的显示的操作。此时，显示该属性页的函数被调用，因此文档属性和打印机属性可以同时被定制，如图4所示，并且属性页都可以显示在一个UI窗口上。

当用于定制的驱动UI窗口被用户关闭时，由用于定制的驱动UI设置的信息被CSPL结构保持。这样，在步骤S805，该定制工具从CSPL结构（包括DeviceOption结构和Devmode结构）获取打印机驱动的信息，其对应于所保持的信息。

定制工具有一部分作为获取部分而工作，其适于在CPU1的控制下获取代表对应于文档属性或是设备属性的打印机驱动的数据。该获取部分保持关于打印机驱动的设置的数据，并且可以通过将数据传输到可以传送数据的数据保持部分（CSPL结构）并从该数据保持部分接收数据获取打印机驱动的信息。此时，该获取部分可以导致数据保持部分（CSPL结构）保持对应于由UI窗口（见图3至图5）设置的文档属性或设备属性的数据，并且可以从数据保持部分（CSPL结构）获得作为用于定制的设置信息的对应于属性的打印机驱动的信息。

定制工具将由这个数据保持部分保持的数据写到将在后面进行描述的UPD文件中。当被调用时，打印机驱动确定是否存在定制的UPD文件。当打印机驱动确定存在定制的UPD文件时，打印机驱动读取包含在UPD文件中的定制的UI控制数据打印设置数据。这样，打印机驱动的定制设置就实现了。

也就是说，根据用于定制的设置信息，作为显示控制部分工作的定制工具部分调用打印机驱动的UI模块。通过使用将在后面进行描述的伪API，被调用的UI模块导致CRT显示10显示定制的UI窗口，其通过了经由图3至图5所示的UI窗口设置的定制信息，该定制信息指定用于控制控制项目的显示的数据并且表示了打印设置信息的缺省值。

打印机属性设置信息被设置在DeviceOption结构（未显示）中，并且文档属性设置信息被设置在Devmode结构（未显示）中，二者都

作为定制设置信息。该定制工具获取作为定制设置信息的存储的设置信息。

之后，在步骤S806，之前在步骤S801加载到存储器上的模块之后被卸载。随后，在步骤S807，用于定制的驱动UI的显示终止。

流程图中表示的处理流所描述处理与对应于打印机驱动的OS相比较稍微有些变化。例如，Windows（注册商标）NT4.0和Windows（注册商标）2000/XP/Server2003利用了共有属性页UI，其为OS运行的计算机系统的一个组件。这样，共有属性页UI函数在必要的处理执行之后被调用。另一方面，根据Windows（注册商标）98/ME，属性页函数被调用并且在必要的处理执行之后显示UI。基本的处理流如图8所示。

图10A为表示了提供在打印机驱动中的Windows（注册商标）应用程序接口（API）的调用方式。当打印机驱动安装在装置中时，Windows（注册商标）API（在附图中简称为“API”）在打印机驱动UI处理1010的情形1040（在普通调用上）中被调用。当支持定制的打印机驱动没有安装在装置中，并且当伪Windows（注册商标）API（在附图中简称为“伪API”）1060在打印机驱动UI处理1010的情形1050中被定制工具调用，通过使用相关的伪Windows（注册商标）API而以伪方式模仿由Windows（注册商标）API执行的处理。接着，结果数据被返回到打印机驱动UI处理1010。

图10B是表示了Windows（注册商标）API函数和相关的伪API函数名称（在该图中分别称为“API”和“伪API名称”）之间的对应关系的图表。假定这种对应关系可以预先设置在安装的打印机驱动中。通常地，对于打印机驱动控制着用户界面的程序适于根据输入到打印机驱动的数据调用Windows（注册商标）API。本实施例配置为使得在打印机驱动没有安装的状态下虽然Windows（注册商标）API不能被调用，但是可以调用伪API。

当伪Windows（注册商标）API从定制工具被调用时，该伪Windows（注册商标）API模仿由Windows（注册商标）API执行的处理并且将

该处理返回到打印机驱动UI处理1010。当打印机驱动对于OS安装后，打印机驱动UI作为普通驱动UI而被调用，普通的Windows（注册商标）API1020被调用并且该处理返回到打印机驱动UI处理1010。

图11为表示调用伪Windows（注册商标）API（在附图中简称为“伪API”）的处理的流的流程图。由主计算机3000的CPU1执行的定制工具为执行该处理中的操作的实体。

首先，在步骤S1101，定制工具确定包括在伪Windows（注册商标）API函数的变量中的打印机句柄（变量）是否与当被从定制工具调用时被传送的CSPL结构的地址信息（此后简称为“地址信息”）相匹配。确定的方法被改编使得CSPL结构的首成分指示着CSPL结构的尺寸。这样，传送的打印机句柄（变量）指的是被作为CSPL结构的首地址而对待。当变量的打印机句柄与CSPL结构的尺寸匹配时，定制工具确定打印机句柄（变量）为CSPL结构的地址（步骤S1110的是）。打印机驱动的UI模块将经由UI输入的代表处理结构的数据存储在由句柄指定的结构中，其中数据例如，代表打印设置项目的值以及控制的显示的指定的数据。接着，该处理结束。

当变量的打印机句柄不是CSPL结构的地址时（步骤S1110的否），执行打印机驱动的操作。这样，在步骤S1120，执行调用Windows（注册商标）API的操作。接着，该处理结束。

如果在步骤S1110确定变量的打印机句柄匹配CSPL结构的地址（步骤S1110的是），处理进行到步骤S1130，在此处Windows（注册商标）伪API被从定制工具调用。在这种情况下，当打印机驱动没有安装时，Windows（注册商标）API不能被调用。因此，在步骤S1130，由伪Windows（注册商标）API执行一个等同于Windows（注册商标）API所执行的处理。接着，等同处理的结构被返回到打印机驱动UI处理1010。接着，该等同处理结束。

这样，定制工具就作为在CPU1的控制之下的信息处理装置的选择部分而工作。该选择部分根据代表由数据保持部分保持的主地址的地址信息（CSPL结构），选择将被使用（被调用）的应用程序接口

(Windows (注册商标) API或者伪Windows (注册商标) API)。

图12为表示作为伪Windows (注册商标) API函数的一个例子,从打印机驱动调用CSPL\_GetVersionEx函数的操作(见图10B所示的1040)的处理流的流程图。假定当这个操作被执行时,打印机驱动UI被定制工具调用,由主计算机3000的CPU1执行的定制工具为执行该处理的实体。同样,CSPL\_GetVersionEx函数被定制工具函数调用。由CSPL\_GetVersionEx函数模仿的Windows (注册商标) API函数为Windows (注册商标) GetVersionEx(见图10B所示的表1030)。该Windows (注册商标) API函数执行将关于运行在主计算机上的OS的版本的返回到在打印机驱动UI端执行的处理1010。

首先,在步骤S1210,定制工具中的CSPL\_GetVersionEx函数确定打印机句柄,其为函数的变量,是否为CSPL结构的地址,当API函数被调用时该CSPL结构被传送。如果不是(步骤S1210的否),该处理进行到步骤S1220,在此处对于普通打印机驱动而操作的Windows(注册商标)CSPL\_GetVersionEx函数1040被调用。这种确定方法类似于对图11所示的步骤S1110的描述所描述的方法。接着,给处理结束。

另一方面,如果在步骤S1210确定函数的变量与CSPL结构的地址相匹配(步骤S1210的是),处理就进行到步骤S1230,在该处根据关于对应于保持在CSPL结构中的打印机驱动的OS的信息,以伪方法执行将由GetVersionEx函数执行的处理(也就是,等同于由GetVersionEx函数执行的处理的处理)。在步骤S1230,该CSPL\_GetVersionEx函数将与由GetVersionEx函数获取的处理结果相同的处理结果返回到打印机驱动UI模块。对于其它的函数也同样。

例如,应用程序接口(伪Windows(注册商标)API)对应于在图3至图5所示的UI窗口所设置的文档属性或设备属性而被调用。该应用程序接口以伪方式(也就是,执行等同于由Windows(注册商标)API执行的处理的处理)执行处理,该处理在打印机驱动中对应于设置的文档属性或是设置的设备属性而执行。

在步骤S1230或S1220执行处理之后,图12表示的流程所显示的处

理结束。

随后，例如，当通过从定制工具调用API而执行关于Windows（注册商标）98/ME驱动模块UI的控制的处理时，实际运行的OS是对应于定制工具的Windows（注册商标）2000/XP/Server2003。然而，在CSPL结构中，包括指示着对应于驱动模块的OS是Windows（注册商标）98/ME的信息。这样，指示着“Windows（注册商标）ME”的值作为表示OS版本的信息被返回。这样，就可以防止Windows（注册商标）98/ME驱动模块的UI控制处理发生崩溃。

也就是说，将被定制的目标是Windows（注册商标）98/ME的打印机驱动以及由例如INF文件和驱动用户界面模块提供的驱动信息。在本实施例中，定制工具由Windows（注册商标）2000/XP/Server2003支持。假定其中执行有定制工具的环境由这些OS中的一个控制。CSPL结构在图8所示的步骤S803进行初始化。此时，指示着例如支持着驱动模块的OS是Windows（注册商标）98/ME的信息被作为表示OS版本的OS版本信息写到CSPL结构。从而，CSPL结构就包含了指示着支持着驱动模块的OS是Windows（注册商标）98/ME的信息。这样，CSPL\_GetVersionEx函数访问包含在CSPL结构中的OS版本信息并且将表示“Windows（注册商标）98/ME”的值返回到调用打印机驱动UI模块。

对于其它的伪Windows（注册商标）API也是同样。即使在驱动模块UI控制的处理中由于没有实际安装打印机驱动而由调用Windows（注册商标）API导致故障的情况下，伪函数（伪API函数）可以模仿由Windows（注册商标）API执行的处理。这样，Windows（注册商标）API的操作可以被虚拟地反射到UI控制上。

虽然没有明确地描述，但是Windows（注册商标）98/ME打印机驱动是一个模块，其在16位环境下操作，因此UI处理在16位环境下执行。然而，根据本实施例定制的Windows（注册商标）98/ME打印机驱动通过使用形实转换机制的形实转换函数而在32位的环境中执行实质的UI处理。这样，根据本实施例的定制工具可以相对容易地调用在

32位环境中操作的UI处理函数，在该环境中执行实质的UI处理，并且可以相对容易地显示并且控制Windows（注册商标）98/ME UI。

也就是，订制工具获取包括打印设置数据的属性。接着，订制工具在OS上登记可操作的设备驱动。也就是，订制工具实施OS的管理权以在设备驱动中设置包括打印设置数据的所获取的属性，使得该设备驱动可以被读取。该设备驱动没有通过将属性登记在由OS管理的注册表区域而被安装。当该属性在该处被设置时，设备驱动被存储在由OS管理的文件系统的用户区域内。该属性由订制工具存储在UPD文件中。创建的UPD文件与将被定制的打印机驱动、打印机驱动UI模块以及INF文件一起构成了安装设置。创建的安装设置通过记录介质和通信介质被分布到每个客户计算机上。安装者可以根据INF文件所描述的信息执行安装。安装者将对应于每个从UPD文件读取的设置项目的值作为devmode结构和 deviceOption结构的每一个的对应项目的值进行复制。通过参考存储在devmode结构和 deviceOption结构中的每一项的值而操作打印机驱动和打印机驱动UI模块。

上述的处理是由作为应用程序的一个例子的订制工具执行。订制工具代替操作系统对通过设备驱动输入的信息进行响应。

这种由订制工具作出的伪代理响应允许设备驱动认可由应用程序作出的响应作为由OS作出的响应，而不会导致错误处理，并且使用从用户输入的数据继续随后的设置处理。

该订制工具包括对应于设备驱动的伪API。同样，订制工具可以在对于各种操作系统（例如，Windows（注册商标）NT以及Windows（注册商标）95）的设备驱动而设置属性，该属性包括打印设置数据。

订制工具可以响应于一个指令在多个设备驱动上设置信息，该信息涉及打印设置并且由订制工具获取。

如上所述，根据本实施例的伪API允许订制工具对于设备驱动如同操作系统一样运行，因此设备驱动表现为安装在对于操作系统的操作条件中。当订制工具不以伪方式响应设备驱动时，该设备驱动切换到错误处理条件。结果是，订制工具可以继续设备驱动的定制而无需

安装设备驱动。

图13为表示控制用以设置对应于定制的打印机驱动属性的UI的处理流的流程图。由主计算机3000的CPU1执行的定制工具是执行处理的实体。在图13所示的过程中，图11和图12所显示的伪API函数被调用。这样，就实现了属性页的显示，参数的获取以及设置值的输入。

首先，在步骤S1300，当UI控制指示输入时，定制工具被激活。接着，在步骤S1310，UI被初始化。接着，在步骤S1320，如果关于UI现实的定制信息包括在图9所示的驱动信息文件中（步骤S1320的是），则处理进行到步骤S1330。该控制指示传递到UI显示控制（无效控制和隐藏控制）。这样，在步骤S1340，使得该装置等待用户的操作。另一方面，如果没有驱动信息文件（步骤S1320的否），则处理略过步骤S1330。接着，在步骤S1340，使得该装置等待用户的操作。

之后，在步骤S1340，装置待机直到用户执行操作。如果用户执行关闭UI窗口的操作（步骤S1350的是），则UI处理在步骤S1370结束。如果用户的操作是另一种（步骤S1350的否），则对应于用户操作的UI控制处理在步骤S1360执行。随后，处理返回到步骤S1320。

经由UI输入的属性信息经由CSPL结构再次存储在打印机驱动信息文件（也就是，UPD文件）。此时，如图3所示的定制码与预先的识别码一起存储在打印机驱动信息文件中。

结果是，关于显示在图5所示的窗口中指定的控制项目的方法的定制数据被保持在图9所示的驱动信息文件中。这样就可以实现根据定制数据的UI控制显示。

本实施例改编为使得定制数据被写到为UPD文件的驱动信息文件。然而，当打印机驱动可以被查到并且可以被分析时，驱动信息文件并不限于UPD文件。例如，UPD文件可以作为打印机驱动的标准驱动信息文件使用在Longhorn中，其为下一个Windows（注册商标）版本，并且该定制信息可以被并入这个文件。

根据本发明，在定制操作在打印机驱动上执行之后，多个打印机驱动可以同时由一个定制完成指令定制。

即使当打印机驱动提供给不支持定制的OS时，也可以创建反射用户想要的定制的定制的打印机驱动。

打印机驱动的初始设置值的定制可以由等同于打印机驱动的UI窗口的UI窗口执行。这样，初始设置值的定制就可以轻易地实现。通过控制显示项目在反射定制的UI窗口上的有效指示、无效指示以及不显示，用户操作的效率和装置的可操作性就得到了提高。

定制的驱动类似于安装设置的未定制驱动而配置。这样，定制的驱动就可以在由微软（注册商标）公司制作的Windows（注册商标）操作系统或者作为下一个Windows（注册商标）OS的Longhorn中的指向和打印环境中作为驱动而操作。结果是，可以改进定制驱动的多功能性。

本发明的特征可以通过将存储介质（或记录介质）提供给系统或是装置而实现，在该存储介质上存储有实现上述实施例的功能的程序码，该系统或是装置的计算机（或是CPU或是MPU）读取并且执行存储在存储介质中的程序码。

在这种情况下，从存储介质读取的程序码本身实现上述实施例的功能。这样，该存储有程序码的存储介质可以构成本发明。上述实施例的功能不仅通过在计算机中执行读取的程序码来实现，而且还通过根据运行在计算机上的OS（操作系统）的指令执行部分或全部的实际的处理来执行，其中的指令由程序码发出。

同样，上述实施例的功能也可以通过根据程序码所发出的指令执行部分或全部的CPU或类似的实际处理而实现，其中CPU或类似提供在插入计算机的功能扩展板或是连接到计算机的功能扩展单元中，其中程序码在被写入提供在功能扩展板或功能扩展单元中的存储器上后被从该存储介质读取。

在本发明应用在上述的存储介质上时，该存储介质存储分别对应于在上面对于实施例的描述中所描述的流程图（图6，8，11到13，以及17）的程序码。

虽然参考着示例性的实施例对本发明进行了描述，但是应当理解

---

的是本发明并不限于所披露的示例性的实施例。下面的权利要求的范围应当被赋予最为广泛的解释，从而包括所有的修改，等同结构和功能。

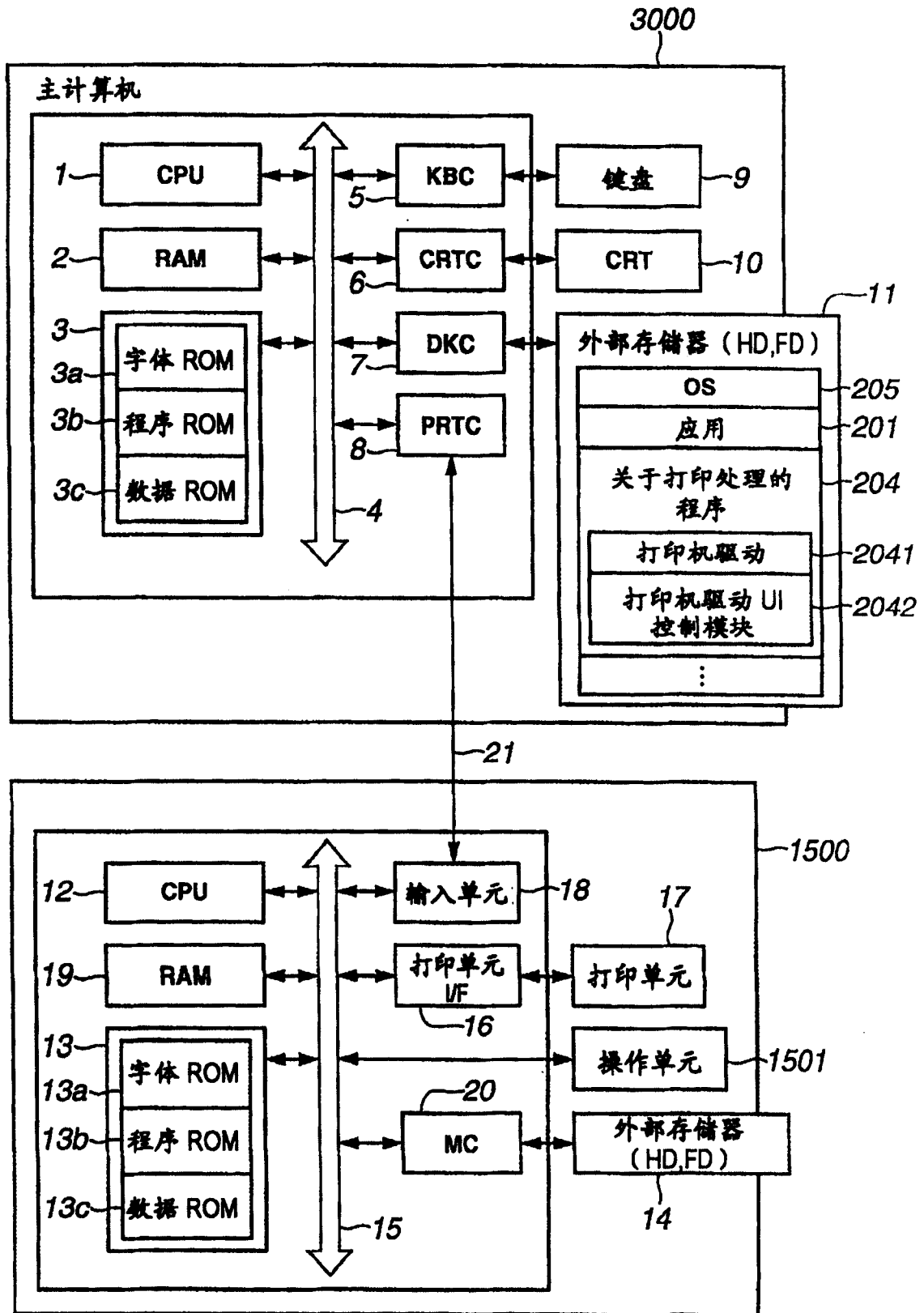


图 1

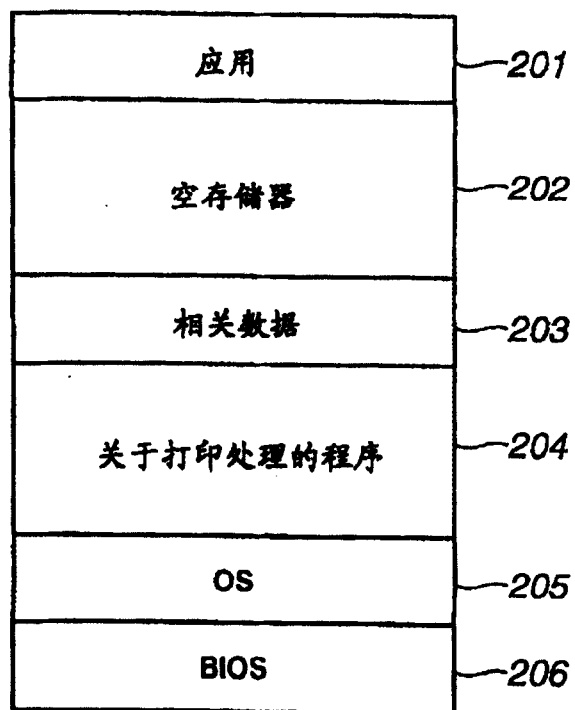


图 2

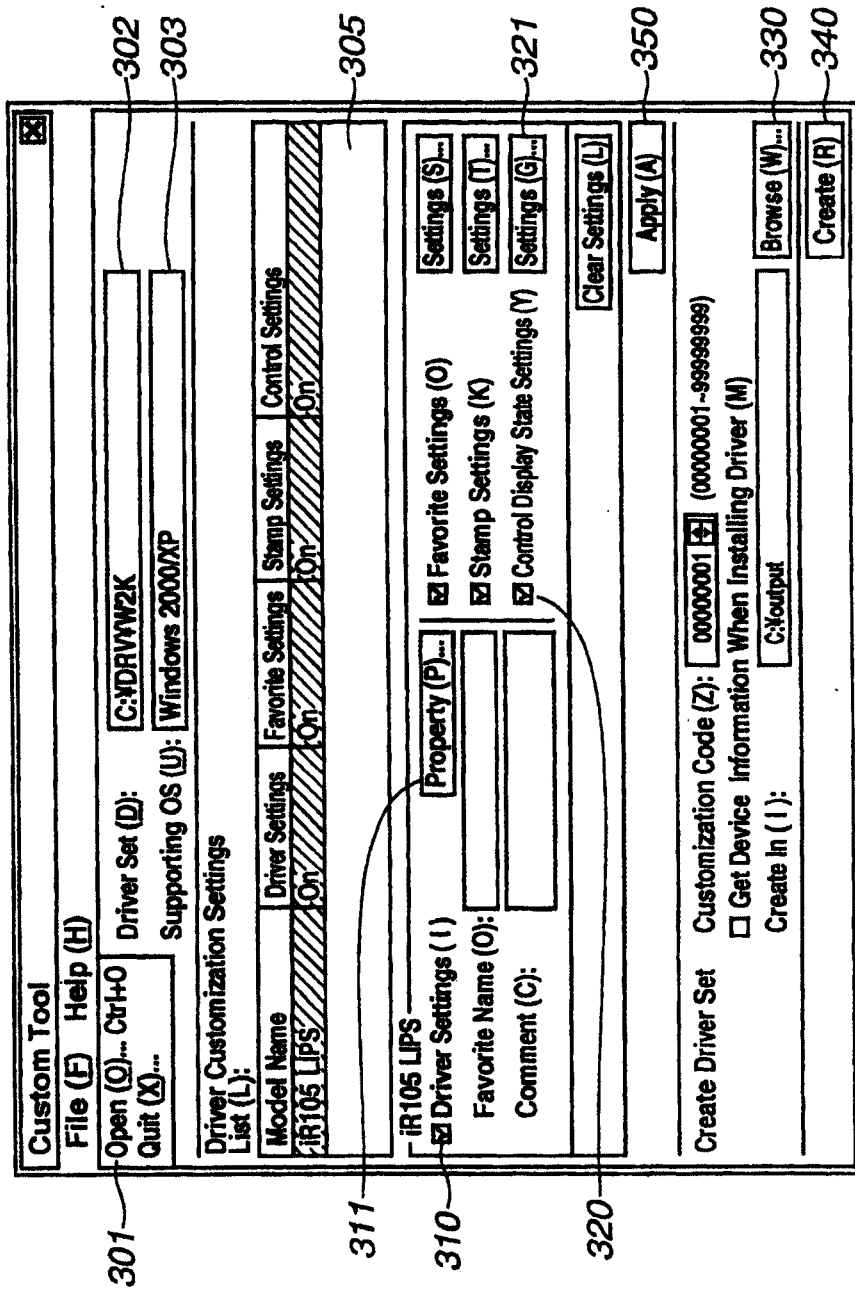


图 3

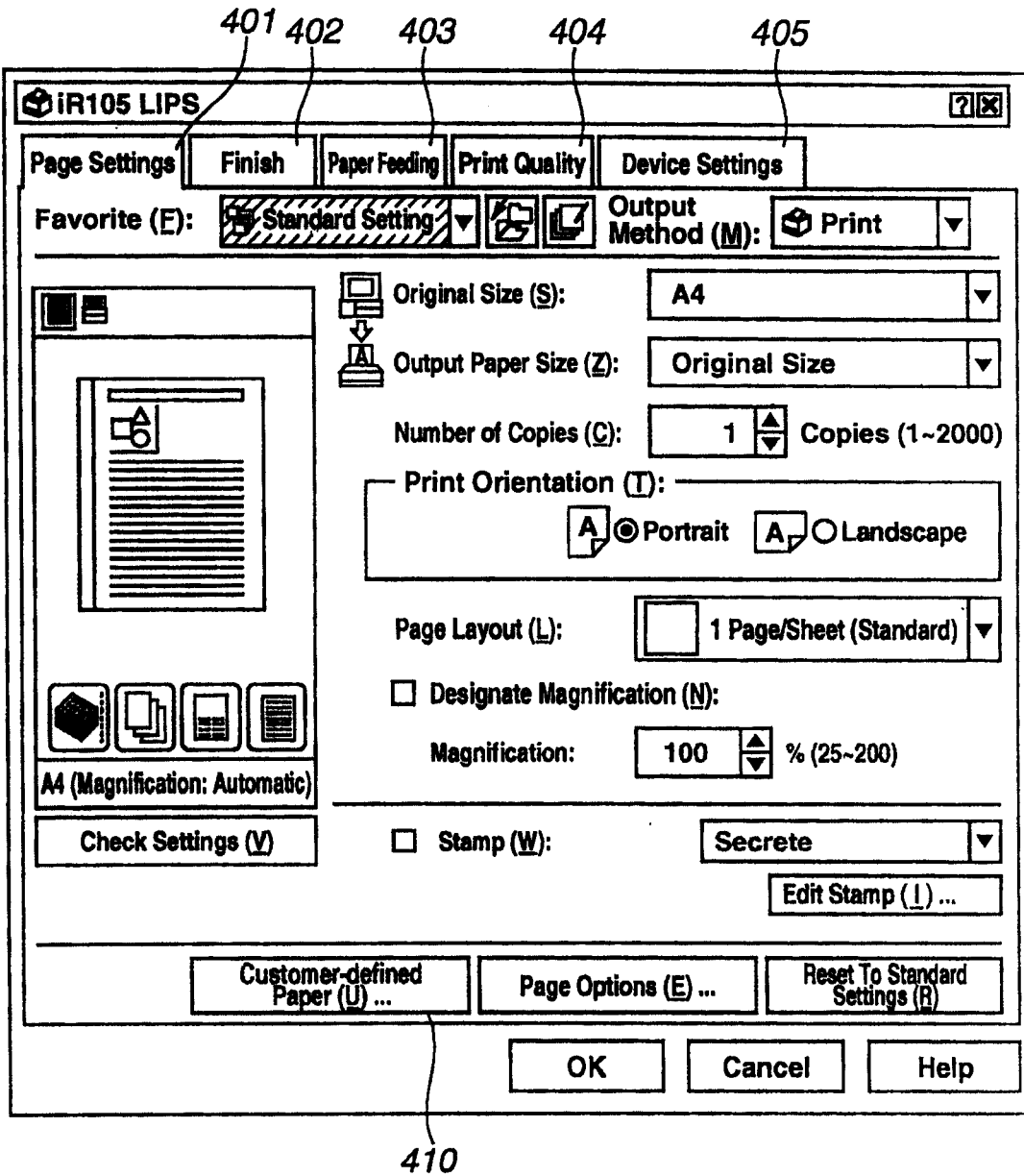


图 4

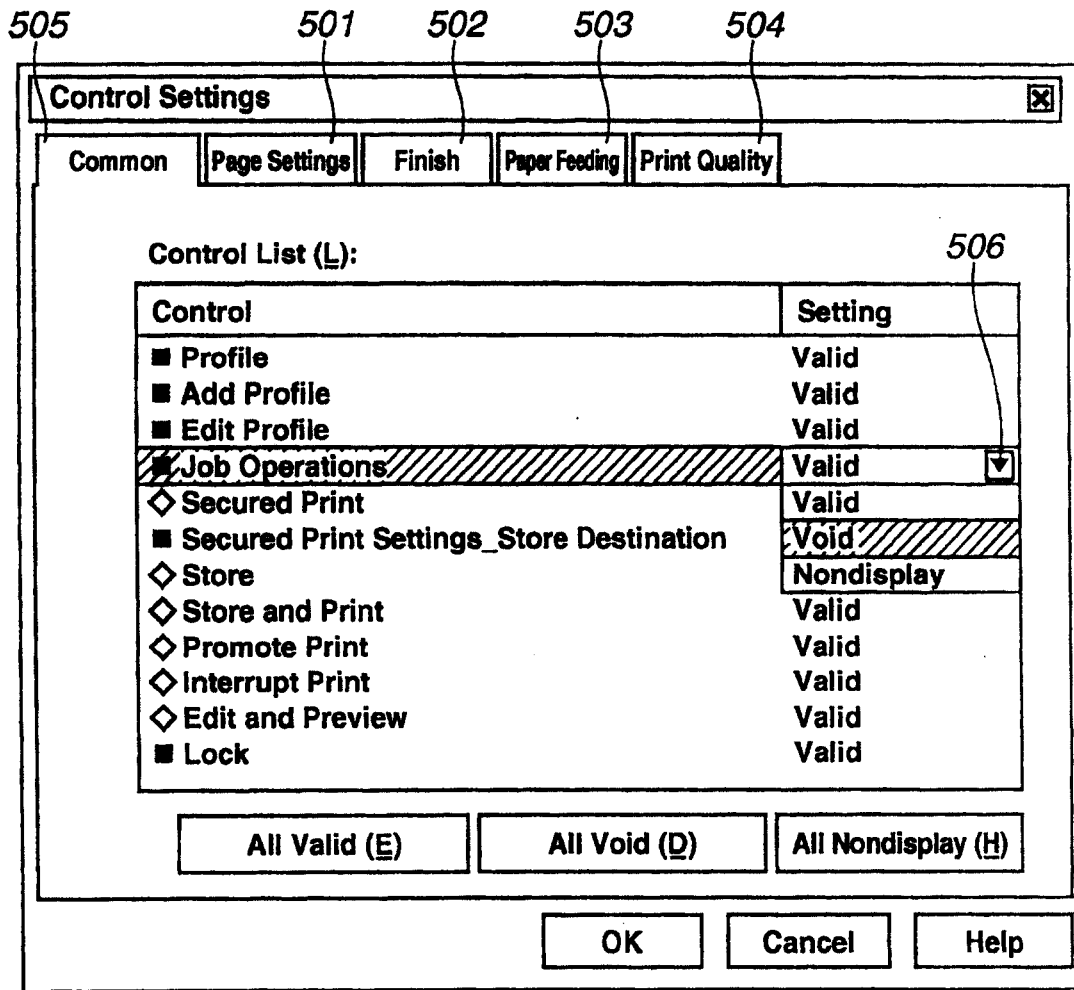


图 5

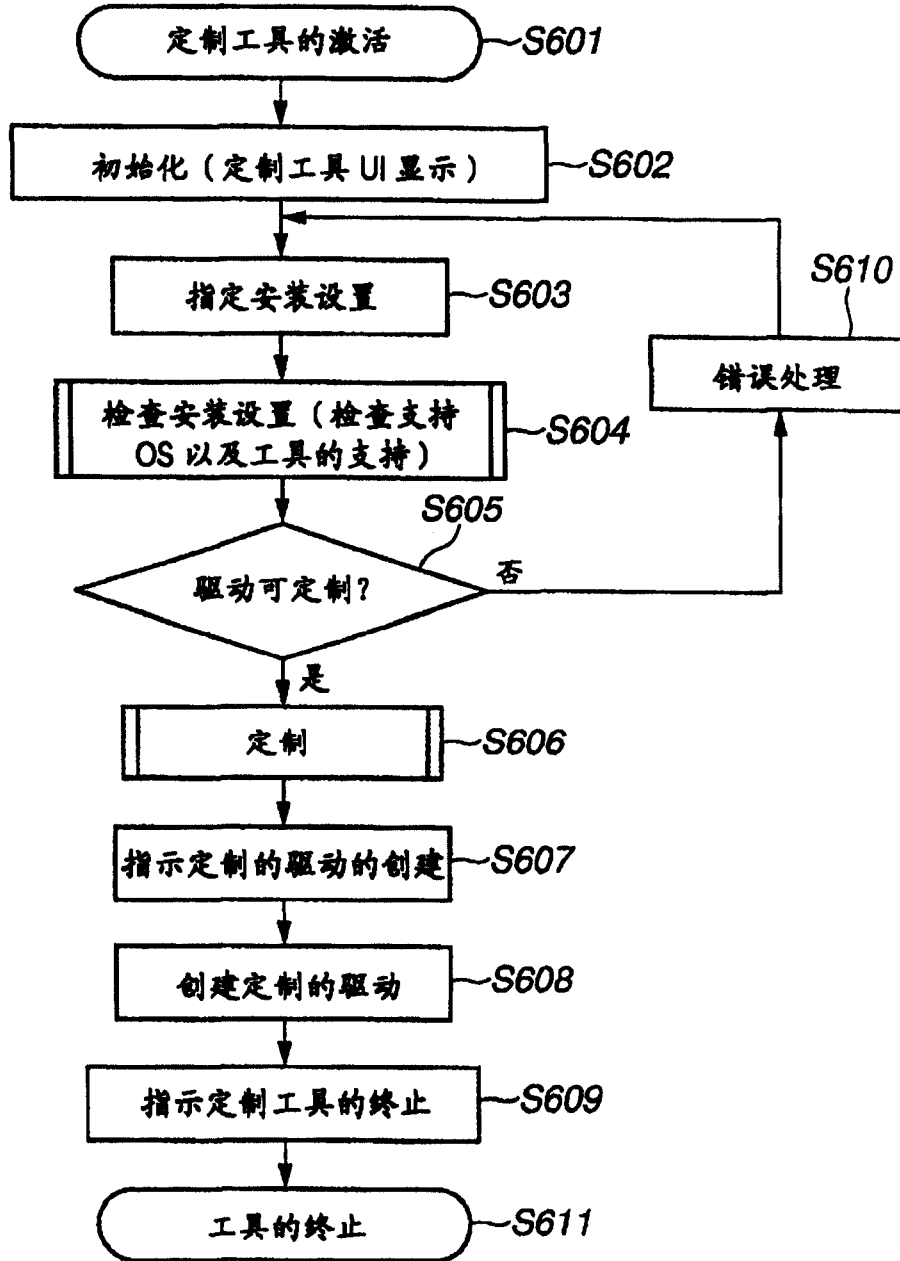


图 6

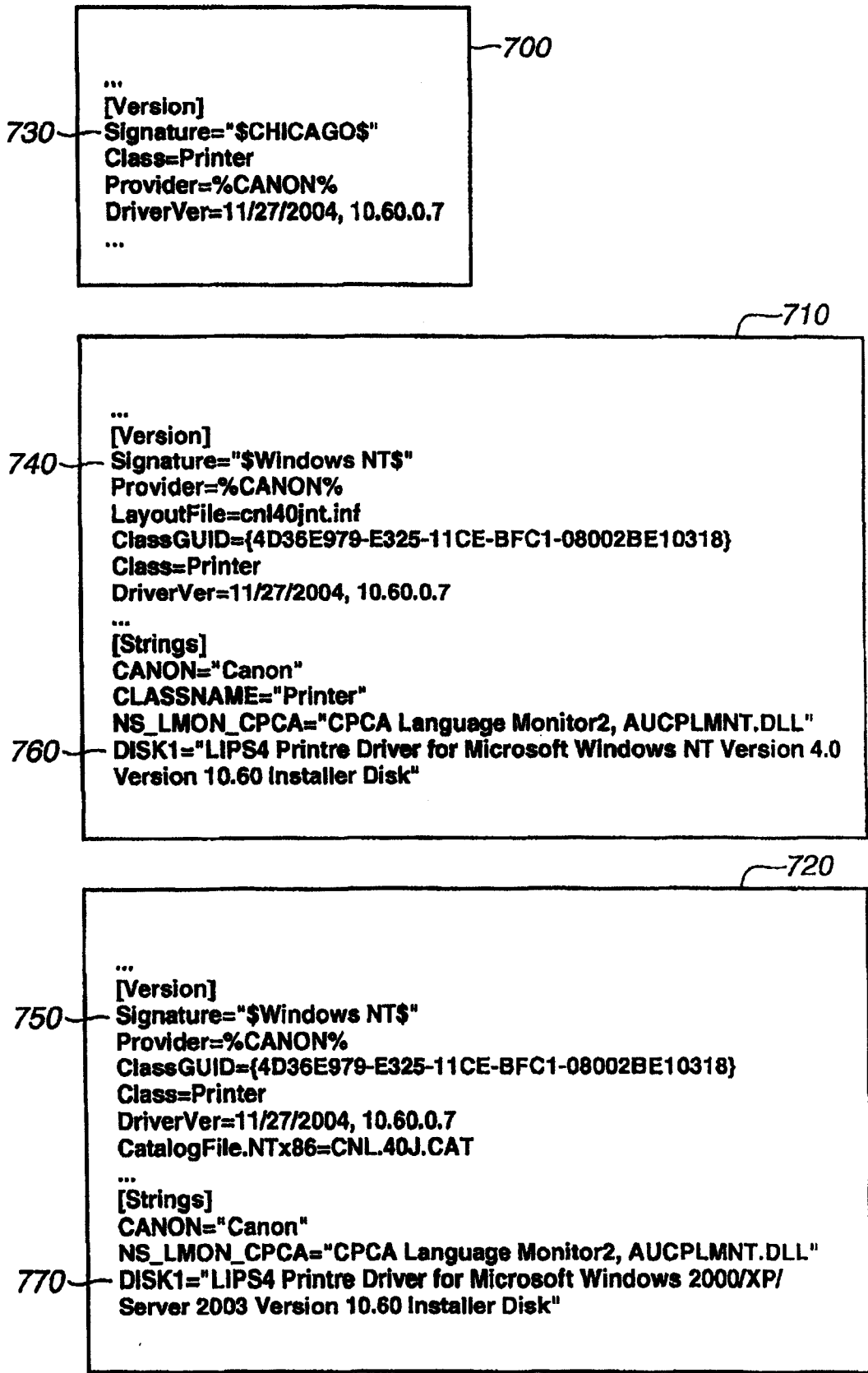


图 7

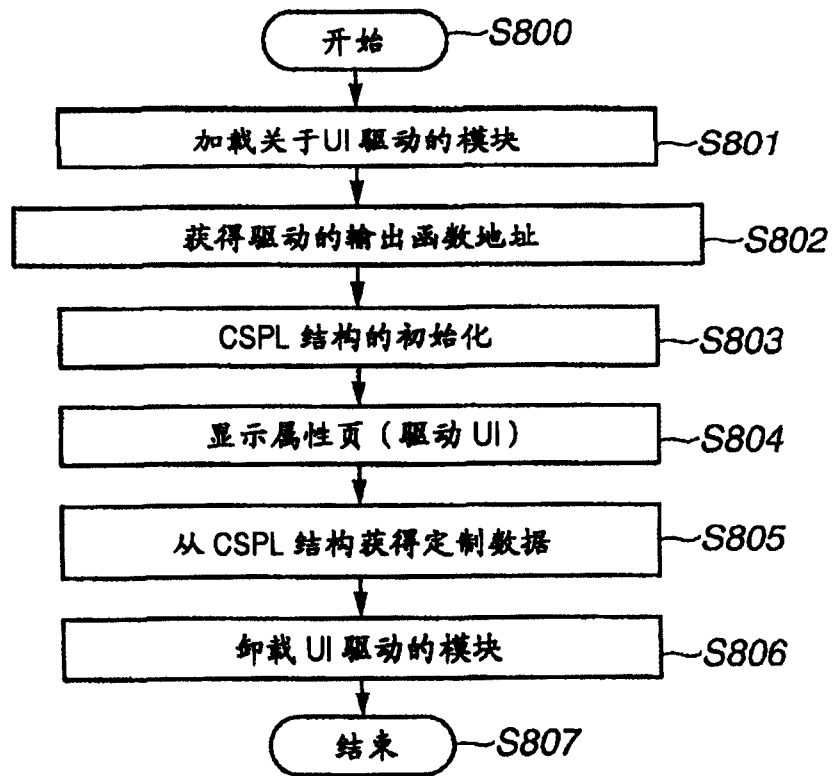


图 8

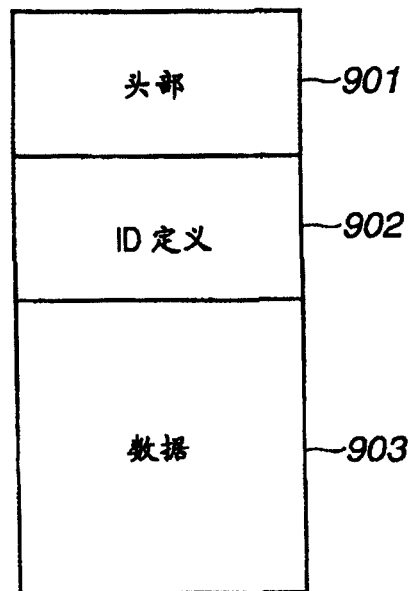


图 9

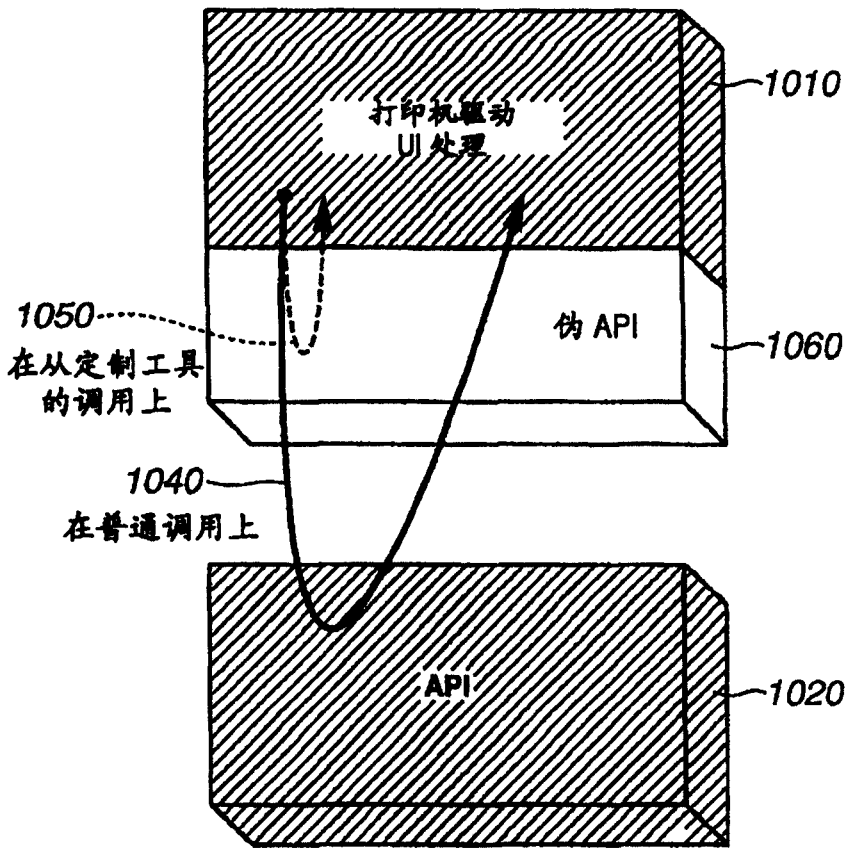


图 10A

1030

API	伪 API 名称
GetVersionEx	CSPL_GetVersionEx
OpenPrinter	CSPL_OpenPrinter
ClosePrinter	CSPL_ClosePrinter
SetPrinter	CSPL_SetPrinter
GetPrinter	CSPL_GetPrinter
GetPrinterDriver	CSPL_GetPrinterDriver
SetPrinterData	CSPL_SetPrinterData
GetPrinterData	CSPL_GetPrinterData

1040

图 10B

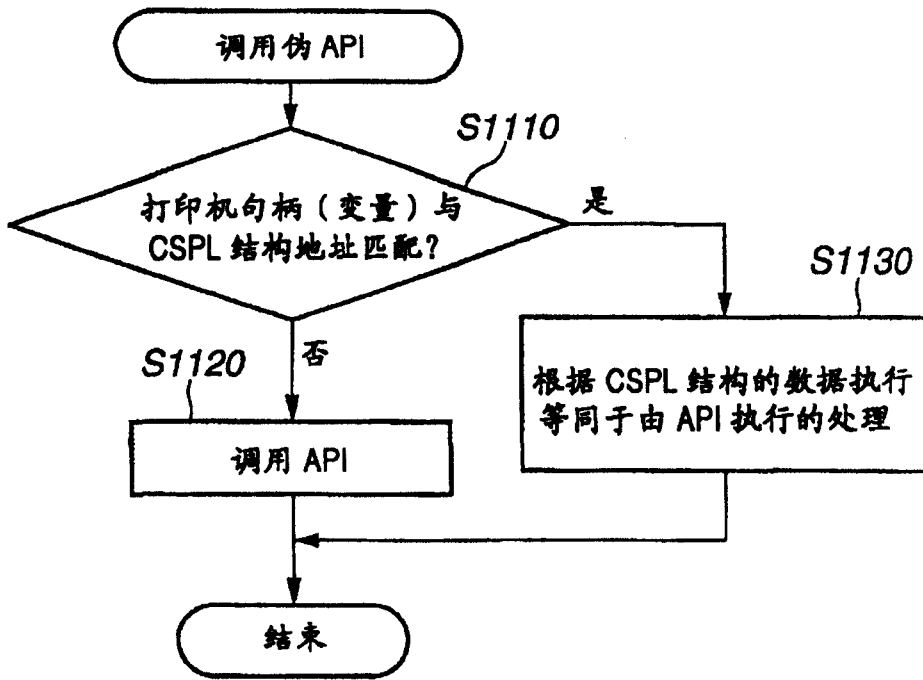


图 11

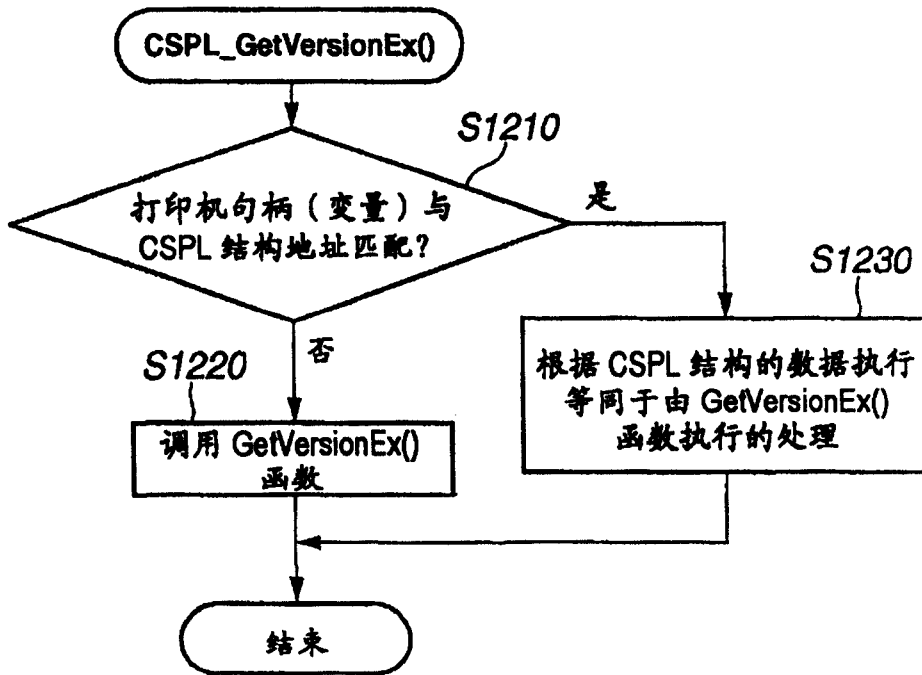


图 12

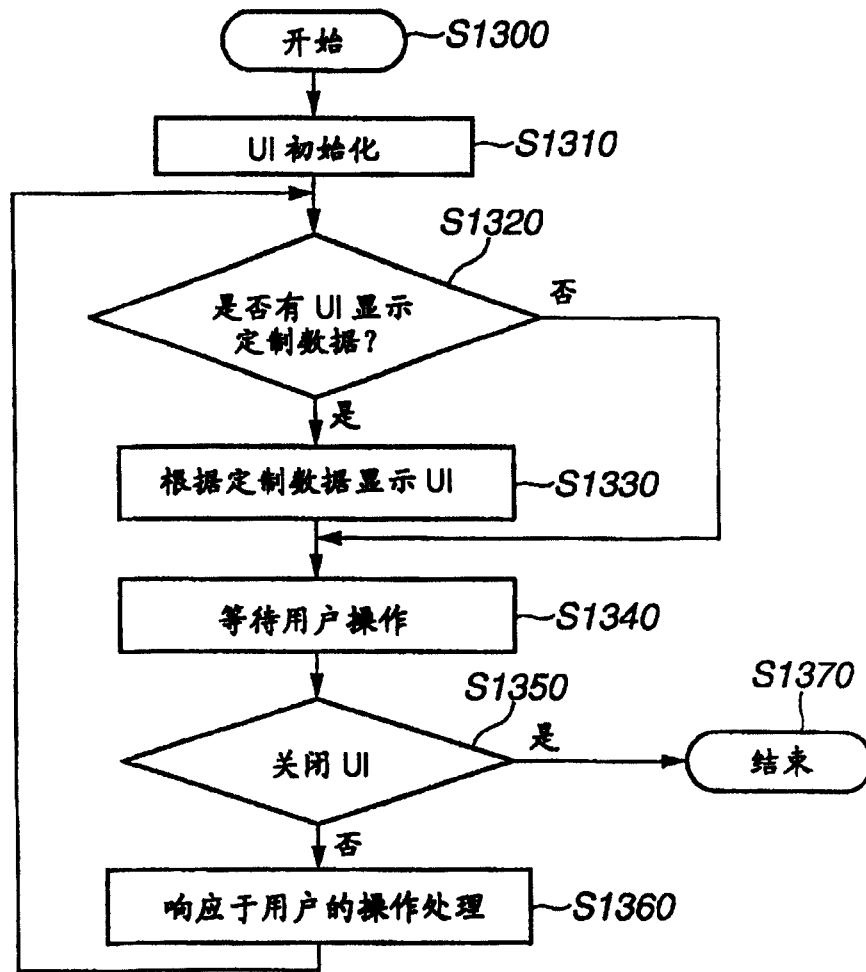


图 13

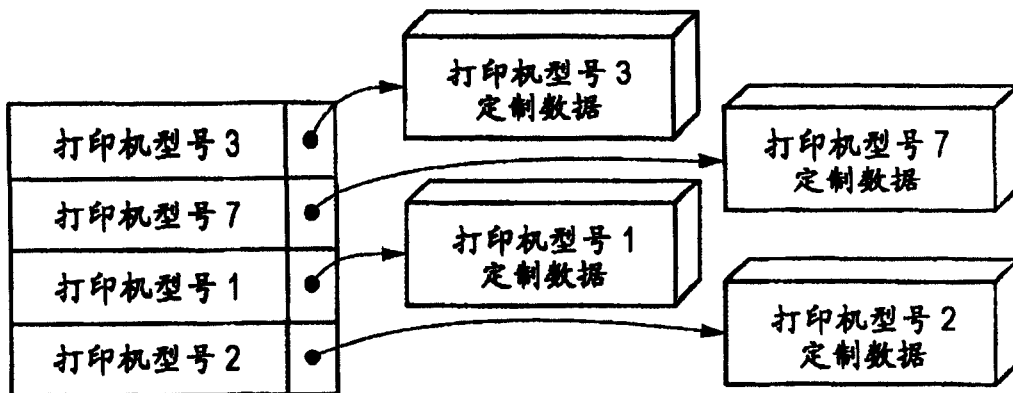


图 14

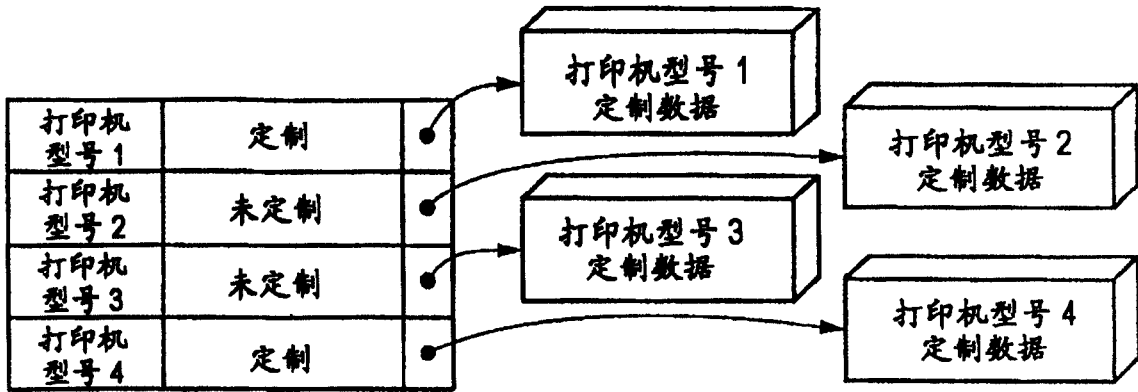


图 15

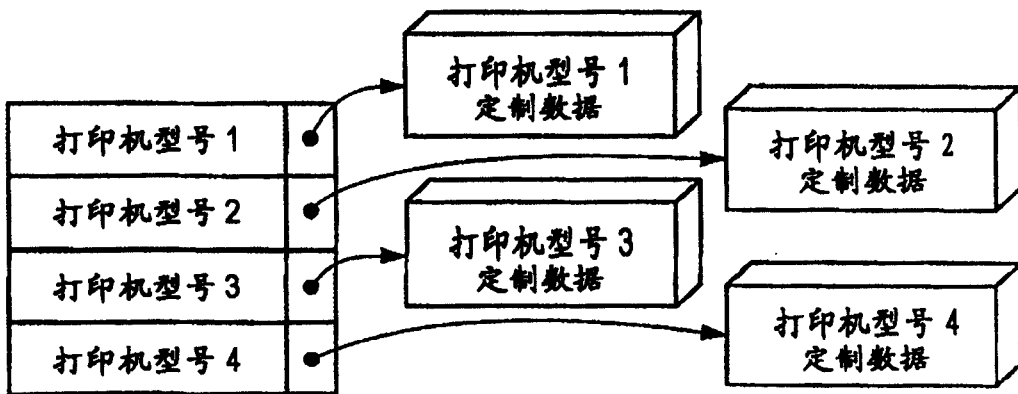


图 16

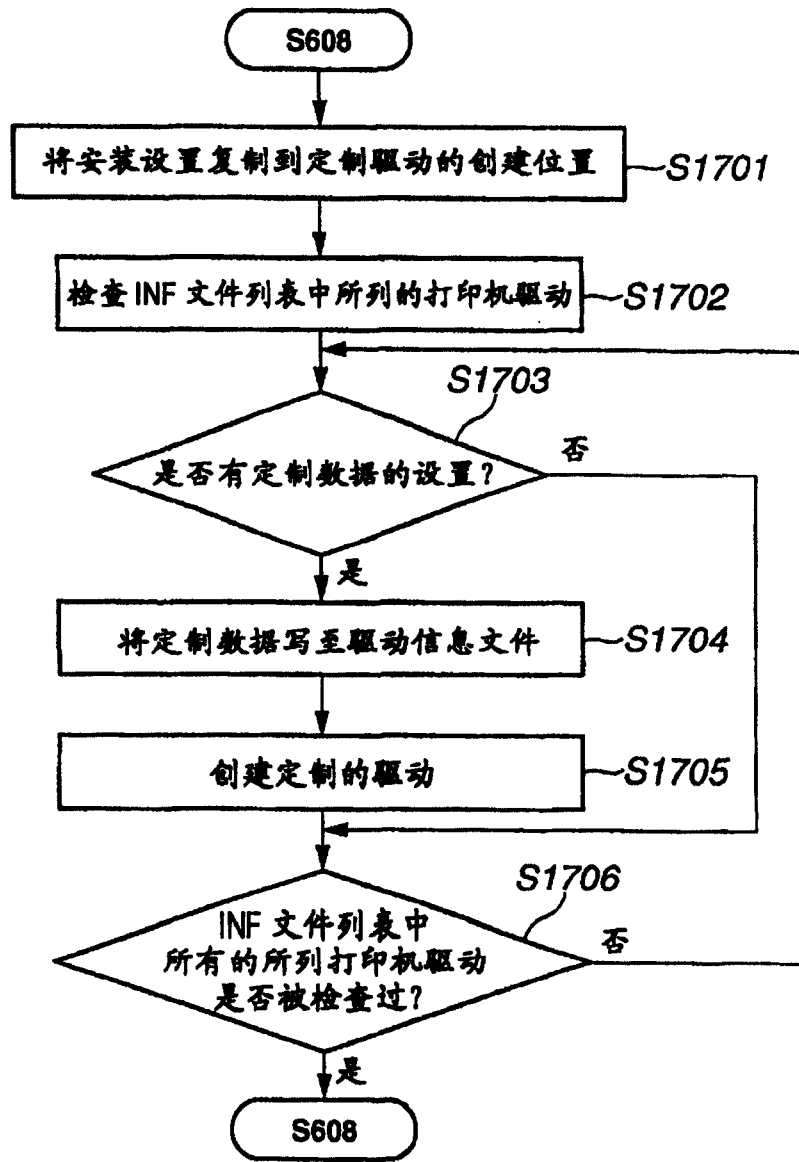


图 17

```
typedef struct _devicemode{ /*dvmd*/  
    TCHAR dmDeviceName[32];  
    WORD dmSpecVersion;  
    WORD dmDriverVersion;  
    WORD dmSize;  
    WORD dmDriverExtra;  
    DWORD dmFields;  
    short dmOrientation;  
    short dmPaperSize;  
    short dmPaperLength;  
    short dmPaperWidth;  
    short dmCopies;  
}DEVMODE;
```

图 18

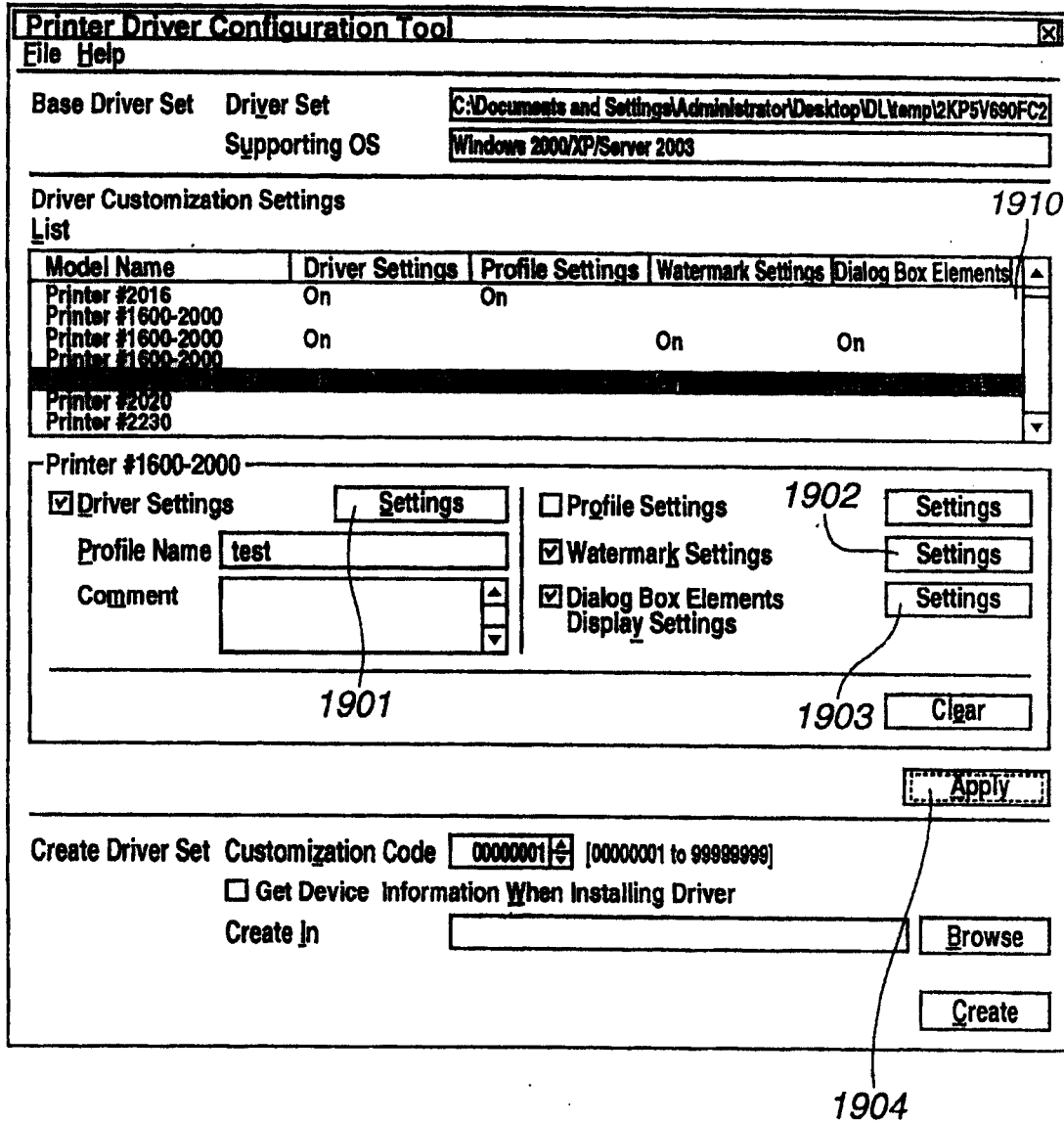


图 19