



(12) 发明专利

(10) 授权公告号 CN 1501239 B

(45) 授权公告日 2010.07.28

(21) 申请号 03123838.6

(56) 对比文件

(22) 申请日 2003.05.14

US 5500937 A, 1996.03.19, 说明书第 1-3 栏.

(30) 优先权数据

WO 0167222 A2, 2001.09.13, 全文.

60/379,781 2002.05.14 US

60/379,749 2002.05.14 US

10/183,990 2002.06.28 US

审查员 詹芊芊

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 A·古纳里斯 S·道奇

T·A·杰林斯基 A·J·戈尔德贝格

K·费尔德曼 T·A·托塞特

R·L·钱伯斯 R·K·塞勒

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 陈斌

(51) Int. Cl.

G06F 9/44 (2006.01)

G06F 3/00 (2006.01)

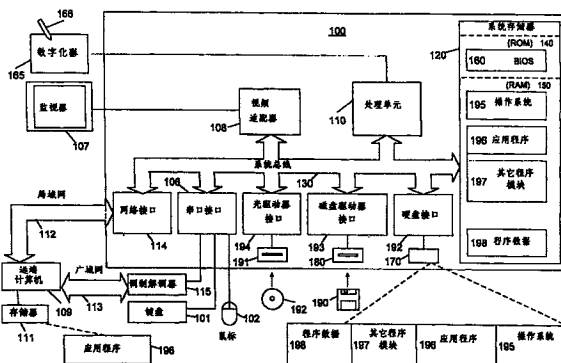
权利要求书 2 页 说明书 11 页 附图 6 页

(54) 发明名称

图形用户接口及其方法

(57) 摘要

一种控制和它的相关联的编程接口,用于允许电子墨水的输入,电子墨水的编辑和其它操作和/或墨水的识别。可以把墨水存储在诸如墨水目标之类的数据结构中,允许稍后通过应用程序获取。在文本可能要加粗、下划线、倾斜等的情况下,所述控制器和它的编程接口可以允许墨水信息象文本那样便于操作,同时提供手写墨水的浓厚度。



1. 一种配置成用控制与用户对接的图形用户接口,所述图形用户接口包括:  
第一显示空间,配置成接收墨水;  
具有第一模式开关的控制,把所述控制配置成当所述第一模式开关在第一设置时存储墨水,并当第一模式开关在第二设置时使墨水作为文本被识别;以及  
第二显示空间,配置成根据第一模式开关的设置第二显示空间显示墨水或文本。
2. 如权利要求 1 所述的图形用户接口,其特征在于,所述控制配置成使之当第一模式开关在第一设置时存储墨水。
3. 如权利要求 1 所述的图形用户接口,其特征在于,所述第一和第二显示空间是相同的显示空间。
4. 如权利要求 1 所述的图形用户接口,其特征在于,在触摸一敏感显示器上显示第一显示空间,根据输入笔相对于第一显示空间中的触摸一敏感显示器的运动接收墨水。
5. 如权利要求 1 所述的图形用户接口,其特征在于,进一步包括一种特性功能,用于当第一模式开关在第一设置时把至少一种特性施加于墨水,并当第一模式开关在第二设置时把至少一种特性施加于文本。
6. 如权利要求 5 所述的图形用户接口,其特征在于,所述至少一种特性是加粗、倾斜和颜色中的至少一种。
7. 如权利要求 1 所述的图形用户接口,其特征在于,所述控制进一步包括第二开关,并进一步配置成根据第二开关的设置接收或不接收对第一显示空间的墨水输入。
8. 如权利要求 1 所述的图形用户接口,其特征在于,所述控制进一步配置成,当第一开关在第二设置时,在使所述墨水作为文本被识别之前等待预定的时间量。
9. 如权利要求 8 所述的图形用户接口,其特征在于,预定时间量在 100 毫秒到 5000 毫秒的范围内。
10. 如权利要求 1 所述的图形用户接口,其特征在于,所述控制进一步包括用于把至少一种特性施加于所述墨水的装置,所述第二显示空间被配置成根据至少一种特性显示所述墨水。
11. 如权利要求 1 所述的图形用户接口,其特征在于,所述控制进一步包括第三开关,并配置成根据第三开关的设置调用多个识别器中的至少一个识别器使所述墨水作为文本被识别。
12. 如权利要求 1 所述的图形用户接口,其特征在于,所述第一模式开关是与显示的单元相关联的。
13. 一种计算机实现的图形用户接口,其特征在于,包括:  
配置成接收墨水并显示所接收的墨水的显示空间;以及  
响应于对相关联的显示元素的用户选择,可在第一设置和第二设置之间切换的模式开关,其中响应于接收墨水同时模式开关处于第二设置而非第一设置,显示的墨水被显示出的经识别的文本替代,所述经识别的文本是将墨水识别为文本的结果。
14. 如权利要求 13 所述的图形用户接口,其特征在于,响应于在模式开关处于第一设置时接收到的命令,显示的墨水被显示出的经识别的文本替代。
15. 如权利要求 13 所述的图形用户接口,其特征在于,响应于接收墨水同时模式开关处于第二设置而非第一设置,墨水被识别。

16. 如权利要求 13 所述的图形用户接口,其特征在于,响应于在模式开关处于第一设置时接收到的命令,墨水被识别。

17. 一种计算机实现的方法,包括:

接收包含墨水的用户输入;

显示所接收的墨水;

显示与模式开关相关联的显示元素,其中,响应于对所述显示元素的用户选择,所述模式开关可在第一设置和第二设置之间切换;以及

响应于接收墨水同时模式开关处于第二设置而非第一设置,显示的墨水被显示出的经识别的文本替代,所述经识别的文本是将墨水识别为文本的结果。

## 图形用户接口及其方法

### 技术领域

[0001] 本发明一般涉及软件应用程序和 / 或数据结构之间的接口。尤其,本发明的尤其涉及用电子墨水交换信息的接口和 / 或允许输入和 / 或编辑电子墨水的接口。

### 背景技术

[0002] 典型的计算机系统,尤其是使用诸如微软视窗之类的图形用户接口 (GUI) 系统的计算机系统,对于接收从一个或多个分立输入设备 (诸如用于输入文本的键盘、诸如用一个或多个按钮的鼠标来驱动用户接口的指针设备) 的用户输入是最佳的。普遍存在的键盘和鼠标接口提供文档、电子数据表、数据库字段、附图、照片等等的快速创建和修改。然而,与非-计算机 (即,标准的) 式的笔和纸相比,键盘和鼠标接口提供的灵活性还有相当的差距。用户用标准的笔和纸编辑文档,在页边距内写附注,以及画图画和其它形状等等。在某些情况中,因为可不受键盘和鼠标接口的限制而自由地作出附注,用户宁可使用笔在文档上作标记而不是在屏幕上审查文档。

[0003] 某些计算机系统允许用户在屏幕上画图。例如,微软 READER 应用程序允许把电子墨水 (这里也称之为“墨水”) 添加到文档中。系统存储墨水,并当请求时把它提供给用户。其它应用程序 (例如,与 Palm 3. x 和 4. x 和 PocketPC 操作系统相关联的本技术领域中公知的画图应用程序) 允许捕获和存储图画。还有,可以基于输入笔的输入产品诸如 Wacom 图形输入产品,使用诸如 CoralDraw 之类的各种画图应用程序和诸如 Photoshop 之类的照相和编辑应用程序。这些附图包括与用于构成附图的墨水笔划相关联的其它特性。例如,可以与墨水一起存储的线宽度和颜色。这些系统的一个目标是再现应用于一片纸上的实际墨水的外观和感觉。然而,在纸上的实际墨水可以具有未被坐标和连接线段的电子收集器捕获的大量的信息。这些信息中的某一些可能包括所使用笔尖的厚度 (如通过实际墨水的宽度所看到)、或笔对于纸的角度、笔尖的形状、墨水的沉积速度以及等等。

[0004] 用电子墨水还发生其它问题。已经考虑了用它书写的一部分应用程序。这导致基本上无能力向其它应用程序或环境提供电子墨水的丰厚度。当可以在多个应用程序之间输入文本时 (例如,通过使用剪贴板),墨水就不具备能够与墨水交互作用的这个能力。例如,不能够创建图 8 的图象,通过剪贴板把所创建的图象复制和粘贴到文档中,然后使墨水变粗。一个困难是图象在应用程序之间的不可移动性。

### 发明内容

[0005] 通常,软件开发者熟悉传统的 Win32 RichEdit 控制,它允许用户输入和编辑格式化的文本。本发明通常涉及扩展 RichEdit 控制的功能而进一步提供接收电子墨水手写的的能力。手写可在一时间延迟后自动地,也可在以后根据请求,被进一步识别和转换成文本。在墨水没被转换成文本时,原始的电子墨水可以被存储,诸如存储在目标型数据结构中。因此,本发明的目的是使用电子墨水对现有的和将来的应用程序提供支持。

[0006] 本发明另一目的是提供一种灵活的和有效的接口,用于与各种特性交互作用,引

用方法和 / 或接收与电子墨水有关的事件,从而解决传统设备和系统所发现的一个或多个问题。本发明的再一个目的涉及改进所存储墨水的内容。其它目的涉及修改所存储的墨水。再有的目的涉及提供墨水和文本间的可操作性。

[0007] 结合下列较佳实施例的详细说明,将会更好地理解本发明的这些和其它特性。

#### 附图说明

[0008] 当和只是作为例子而不是作为对所申请发明专利限制的附图一起阅读时,会更好地理理解上述发明概要和下列较佳实施例的详细说明。

[0009] 图 1 是可以用于实施本发明的作为例子的通用数字计算环境的功能性方框图。

[0010] 图 2 是根据本发明可以使用的示例图形输入板计算机和输入笔的平面图。

[0011] 图 3-6 是根据本发明的图形用户接口的各种示例屏幕照相。

[0012] 图 7 是根据本发明可使用的各种示例手势。

[0013] 图 8 是根据本发明表示一系统环境的功能性方框图。

[0014] 图 9 是根据本发明的示例应用程序编程接口的功能性方框图。

#### 具体实施方式

[0015] 下面描述捕获、识别、与之接口、编辑或操作、和 / 或显示电子墨水的一种方法。

[0016] 一般计算平台

[0017] 图 1 是可以用来实施本发明的传统通用数字计算环境的一个例子的功能性方框图。在图 1 中,计算机 100 包括处理单元 110、系统存储器 120 以及把包括系统存储器之类的各种系统部件耦合到处理单元 110 的系统总线 130。系统总线 130 可以是数种总线结构类型中的任何一种,包括存储器总线或存储器控制器、外围总线、以及使用多种总线结构中的任何一种的本地总线。系统存储器 120 包括只读存储器 (ROM) 140 和随机存取存储器 (RAM) 150。

[0018] 在 ROM 140 中存储基本输入 / 输出系统 160 (BIOS),它包含在计算机 100 中的单元之间帮助传递信息的基本例行程序,诸如在启动期间。计算机 100 还包括在硬盘 (未示出) 进行读出和写入硬盘驱动器 170 ; ,在可拆卸磁盘 190 进行读出或写入的磁盘驱动器 180 ; 以及在诸如 CD ROM 或其它光媒体之类的可拆卸光盘 192 中进行读出和写入的光盘驱动器 191。分别通过硬盘驱动器接口 192、磁盘驱动器接口 193 以及光盘驱动器接口 194 把硬盘驱动器 170、磁盘驱动器 180 和光盘驱动器 191 连接到系统总线 130。驱动器和与它们相关联的计算机可读媒体提供计算机可读指令、数据结构、程序模块以及用于个人计算机 100 的其它数据的非易失性存储。熟悉本技术领域的人员可以理解,在示例操作环境中,也可以使用可以存储计算机可存取数据的其它类型的计算机可读媒体,诸如盒式磁带、快速存储卡、数字视频盘、Bernoulli 数据盒、随机存取存储器 (RAM)、只读存储器 (ROM) 等。

[0019] 可以把许多程序模块,包括操作系统 195、一个或多个应用程序 196、其它程序模块 197 以及程序数据 198,存储在硬盘驱动器 170、磁盘 190、光盘 192、ROM140 或 RAM 150 中。用户可以通过诸如键盘 101 和指示设备 102 之类的输入设备,把命令和信息输入计算机 100。其它输入设备 (未示出) 可以包括话筒、摇杆、游戏小键盘、卫星碟形天线、扫描器等。通常通过耦合到系统总线的串行端口接口 106 把这些和其它输入设备连接到处理单元

110,但是可以通过其它接口连接,诸如并行接口、游戏接口或通用串行总线(USB)。再进一步,可以通过合适的接口(未示出)把这些设备直接耦合到系统总线130。还可通过视频适配器108之类的接口,把监视器170或其它类型的显示设备连接到系统总线130。除了监视器之外,个人计算机一般还包括其它外围输出设备(未示出),诸如扬声器和打印机。在一个较佳实施例中,提供笔数字化器165和附带的笔或输入笔166,以便数字化地捕获徒手输入。虽然示出的笔数字化器165和处理单元110之间是直接连接的,实际上,如本领域中众知,可以通过串行端口、并行端口或其它接口和总线把笔数字化器165耦合到处理单元110。此外,虽然示出笔数字化器165是和监视器107分开的,但是最好是使数字化器165的可使用输入区和监视器107的显示区共同延伸。再进一步,可以把数字化器165集成到监视器107中,或可以作为分立的设备相重叠而存在,或附加到监视器107中。

[0020] 计算机100可以在网络化环境中操作,所述网络化环境使用到一台或多台远程计算机(诸如远程计算机109)的逻辑连接。远程计算机109可以是服务器、路由器、网络PC、同等设备或其它公共网络节点,一般包括上面介绍的与计算机100有关的许多单元或全部单元,虽然在图1中只有示出一个存储器存储设备111。在图1中描绘的逻辑连接包括局域网(LAN)112和广域网(WAN)113。这种网络环境在办公室、全企业的计算机网、内部网、互联网中是常见的。

[0021] 当在局域网网络环境中使用时,通过网络接口或适配器114把计算机100连接到局域网112。当在广域网网络环境中使用时,个人计算机一般包括调制解调器115或用于在广域网113上建立通信的其它手段,诸如互联网。通过串行端口接口106把可以是内接的或外接的调制解调器115连接到系统总线130。在网络化的环境中,可以把与个人计算机100或其部分有关的程序模块,存储在远程存储器存储设备中。

[0022] 可以理解,所示的网络连接是示例性的,可以使用在计算机之间建立通信链路的其它技术。假定存在诸如TCP/IP,以太网、FTP、HTTP等各种众所周知的协议中的任何一种,并且系统可以在用户-服务器配置中操作以允许用户从基于网的服务器取得网页。可以使用各种传统的网浏览器中的任何一种来显示和操作网页上的数据。

[0023] 图2示出可以根据本发明而使用的基于输入笔的计算机处理系统(又称为图形输入板PC(个人计算机))201。在图1中的系统的任何或所有特征、子系统以及功能都可以包括在图2的计算机中。图形输入板PC201包括一个大显示表面202,例如,数字化平面显示器,最好是液晶显示器(LCD)屏幕,在其上显示多个窗口203。可以使用的其它显示技术可以包括,但是不限于,OLED显示器、等离子显示器等。用户可以使用输入笔204的笔尖(这里还把笔尖称为“光标”)在数字化显示区上选择、突出显示和写入。数字化显示板的合适的例子包括电磁笔数字化器,诸如Mutoh和Wacom笔数字化器。还可以使用其它类型的笔数字化器,例如,光数字化器。图形输入板PC201解译使用输入笔204作出的标记,以便操作数据、输入文本以及执行传统计算机应用程序任务,诸如电子数据表、处理程序等。

[0024] 可以使输入笔配备按钮或其它特征以扩大它的选择能力。在一个实施例中,可以实施输入笔作为“铅笔”或“钢笔”,其中,一端构成写入部分,另一端构成“擦除”端,当“擦除”端在显示器上移动时,指示出要擦除的显示部分。可以使用诸如鼠标、跟踪球等其它类型的输入设备。此外,在触摸-敏感或接近-敏感的显示器上,用户可以使用自己的手指来选择或指示所显示图象部分。因此,这里所使用的“用户输入设备”具有宽广的定义,并包

括许多已知输入设备的变型。

[0025] 电子墨水和墨水目标的概念

[0026] 这里所使用的墨水称为电子墨水。电子墨水被设计成一个序列的笔或一组笔划，其中，每一笔划包括一序列的点或一组点。通过所捕获的时间和 / 或通过在上页面上笔划和 / 或点出现的地方可以使笔划和 / 或点的序列排序。笔划组可以包括笔划和 / 或点的序列、和 / 或未排序的笔划和 / 或点。点可以使用本技术领域中的多种技术来表示，包括笛卡尔坐标 (X, Y)、极坐标 (r,  $\theta$ ) 以及其它技术。另一方面，可以用一个点和下一个点的方向上的矢量来表示笔划。打算使笔划包括与墨水有关的任何点或分段的显示，不考虑点下面的显示和 / 或点与什么连接。一般在数字化器（诸如显示表面 202 的数字化器）处开始墨水收集。用户可以把输入笔放在数字化器上，并开始写入或画图。那时，可以产生新墨水包（即，与墨水有关的数据包）。用户还可以在与数字化器足够近的空空中移动输入笔以致使数字化器检测到。当发生这种情况时，根据所检测的输入笔在空中的运动，可以产生数据包（这里称为“空中包”）。包不但可以包括位置信息，还可以包括输入笔压力和 / 或角度信息。

[0027] 为了存储墨水，可以创建表示输入笔 204 在显示表面 202 和 / 或其它输入上划出的墨水原始笔划的墨水目标。可以从显示表面 202 上任何地方或只从其上确定部分（诸如特定的窗口）收集墨水的收集笔划。墨水目标实质上是墨水数据的容器。墨水如何存储在墨水目标中的特定格式对于本发明是不重要的。然而，最好把原始划出的笔划存储在墨水目标中。

[0028] 可以定义墨水目标的两种示例类型。可以实施 t 墨水 (tInk) 目标（“t”的意思是“text(文本)”）作为 OLE 目标，它表示期望形成字母或字的墨水。t 墨水 (tInk) 目标允许通过诸如文本识别器把手写墨水转换成文本。t 墨水 (tInk) 目标是指涉及墨水并具有原文上下文的墨水目标。原文墨水的颜色和 / 或字体的字号以及原文墨水是否应该下划线、加粗、倾斜和 / 或等等可以编程设置，并且可以基于围绕 t 墨水 (tInk) 目标的文本的属性。换言之，可以把在 t 墨水 (tInk) 目标指定插入点处的环境特性施加于 t 墨水 (tInk) 目标。在一个实施例中，t 墨水 (tInk) 目标只包括用于提交给文本识别器的一个单字，致使一句句子可以包括多个 t 墨水 (tInk) 目标。另一方面，还可以定义 s 墨水 (sInk) 目标（“s”的意思是“草图 (sketch)”），表示不期望形成字的墨水。s 墨水 (sInk) 目标也可以是 OLE 目标。因此，可以把 s 墨水 (sInk) 目标解译为画图或任何其它非原文上下文。s 墨水 (sInk) 目标对于显示多个字也是有用的。墨水兼容的应用程序（和 / 或用户）可以对某些墨水目标作出 t 墨水 (tInk) 目标的标记，并对其它墨水目标作出 s 墨水 (sInk) 目标的标记。为了说明的目的，这里把两种类型的墨水描述为“tInk(t 墨水)”和“sInk(s 墨水)”。然而可以理解，可以使用其它名称来表示可以采用的各种类型的墨水目标。还有，可以使用另外类型的目标来存储任何要求格式的电子墨水。

[0029] 可以定义一个目标（这里称为“InkCollector(墨水收集器)”目标）以及用来从墨水输入设备捕获墨水和 / 或向应用程序提供墨水。在某种意义上，墨水收集器 (InkCollector) 目标的作用是水龙头，它通过收集作为一划或多划墨水笔划的墨水并把墨水存储在一个或多个相关联的墨水目标中，使墨水“倾注”到一个或多个不同的和 / 或有区别的墨水目标中。墨水收集器 (InkCollector) 目标可以把它自己附加到已知的应用程序

窗口中。然后,它可以通过使用任何或所有可得到的图形输入设备(可以包括输入笔 204 和 / 或鼠标)而提供在该窗口上的实时涂墨。为了使用墨水收集器(InkCollector)目标,开发者可以创建它,分配哪个窗口把墨水收集进来,以及启动目标。在启动墨水收集器(InkCollector)目标之后,可以进行设置,用多种墨水收集模式(在其中收集墨水笔划和 / 或手势)收集墨水。手势是输入笔 204 的一种运动或其它动作,解译成不是作为涂抹的墨水,而是解译成为执行某种动作或功能的请求或命令。例如,为了选择墨水的目的,可以执行特定的手势,而另外的手势可以是为了使墨水倾斜的目的。每一时刻当输入笔在数字化器输入上时或与之接近时,墨水收集器(InkCollector)目标会收集笔划和 / 或手势。

[0030] 墨水编辑控制(InkEdit Control)

[0031] 可以定义控制,这里称为“InkEdit(墨水编辑)”控制,它为捕获、识别和 / 或显示墨水(例如,以文本的形式)提供一种方便的方法。InkEdit(墨水编辑)控制可以进一步支持显示墨水为具有墨水-格式化和 / 或文本格式化能力的嵌入的目标(例如,作为嵌入的 t 墨水(tInk)目标,经过编程可通过墨水选择特性存取),诸如加粗、下划线、倾斜、上标、下标、对齐等。这个目标的主要指定应用是允许输入墨水以及显示墨水或从墨水识别的文本。这个控制可以进一步允许对墨水和 / 或所识别的文本进行编辑和 / 或格式化。

[0032] 众知的微软 WINDOWS 用户接口是编辑控制(即, RichEdit 和 RichTextBox)。InkEdit(墨水编辑)控制的实施例向开发者提供了这些控制的扩充墨水版本,它们已经是熟悉的和可能已经使用在它们的应用程序中,除了从键盘接收文本的现有能力之外,把各种特征添加到现有的 RichEdit 控制,以接收来自输入笔、鼠标和 / 或其它指针的文本。例如,在一个实施例中,InkEdit(墨水编辑)控制提供接收电子墨水手写和识别的能力,并把该墨水转换成文本。InkEdit(墨水编辑)控制可以进一步提供接收手写作为用于以后识别的墨水,致使手写本身是可编辑的能力。

[0033] 为了使用 InkEdit(墨水编辑)控制,开发者需要简单地用具体例子说明 InkEdit(墨水编辑)控制。开发者和 / 或运行时用户可以进一步把一种或多种模式应用于 InkEdit(墨水编辑)控制各种特征。例如,一种模式可以指示是否应该插入墨水作为墨水或文本。InkEdit(墨水编辑)控制可以管理许多内部世俗细节,用于建立图形输入上下文、收听数字化器事件、收集笔划、把笔划馈送到识别器和 / 或把识别的结果(例如,它可以是 OLE 嵌入目标)馈送到 InkEdit(墨水编辑)控制进行显示以及以后的保留。

[0034] 在一个示例实施例中,可以在 ActiveX 和 Win32 中实施 InkEdit(墨水编辑)控制,并基于传统的微软 Win32 Rich Edit 控制。在另一个示例实施例中,可以在微软 .NET 实施 InkEdit(墨水编辑)控制,并基于 Win32 InkEdit、RichEdit 控制以及微软 .NET RichTextBox 控制。如所众知,微软 RichEdit 和 Rich TextBox 控制允许用户输入、编辑、格式化、打印和存储文本,同时提供各种先进的格式化特征(诸如文本字体、颜色、格式化等)。因此,除了现在可应用于墨水和传统文本的那些特征之外,本发明的 InkEdit(墨水编辑)控制可以具有许多由 RichEdit 和 RichTextBox 控制提供的特征。墨水可以成为它自己中的以及它自己的第一级公民。在可能对文本进行加粗、下划线、倾斜等的情况中,InkEdit(墨水编辑)控制和它的编程接口可以允许象文本那么方便地操作墨水信息,同时提供手写墨水的浓厚度。

[0035] 在某些实施例中,设计 InkEdit(墨水编辑)控制使之在单行和多行文本输入和编

辑的形式情况中能很好地工作。InkEdit(墨水编辑)控制可以以原文手写的形式从一个用户得到墨水输入。可以识别墨水输入,可以把打印的文本插入到位。除了用户正在用墨水写时,用于 InkEdit(墨水编辑)的缺省用户接口可以类似于传统 Rich TextBox 控制的缺省用户接口。InkEdit(墨水编辑)控制可以显示原始的墨水或经识别的文本(或两者)。可以把所显示的墨水定标到 InkEdit(墨水编辑)控制的当前输入字体的字号,还可以与其它文本串联式地显示,和/或可以改变它的位置、字号和/或颜色。另一方面,所显示的墨水可以保持它的原始位置、字号和/或颜色。

[0036] 在一个示例实施例中,InkEdit(墨水编辑)控制的缺省特性是识别墨水并在短的识别时间过去之后,把墨水转换成文本。这个识别时间可以是所要求的任何时间量,诸如约 2000 毫秒,或在约 100 毫秒到约 5000 毫秒的范围内,或在约 200 毫秒到约 2000 毫秒的范围内。在把识别时间设置成零时,可以禁止自动识别。因为 InkEdit(墨水编辑)控制可以是超级的 RichEdit,所以它又可能在 InkEdit(墨水编辑)控制中嵌入和显示墨水。可以把每个墨水字插入控制作为墨水目标(例如,t 墨水(tInk)目标)。墨水目标可以包括墨水以及与墨水相关联的一个或多个特性。

[0037] 当施加插入时,可以使墨水定标为当前字体的字号和其它环境特性,诸如倾斜或加粗。如果用户选择编辑墨水目标的文本,则用户可以首先把墨水转换成文本。

[0038] 参考图 3, InkEdit(墨水编辑)控制可以出现在与图形用户接口 301 相关联的显示器上。图形用户接口 301 可以包括一个或多个显示空间 302,用于接收汲取的墨水数据和/或用于显示墨水和/或文本。图形用户接口可以接收数据,诸如在显示空间 302 中通过输入笔 204 汲取的墨水数据,并且 InkEdit(墨水编辑)可以把数据解译为墨水。InkEdit(墨水编辑)控制可以进一步使墨水与通过应用程序开发者、用户、和/或自动地选择的诸如加粗、下划线、倾斜上标、下标、对准、颜色、字号和/或等等之类的一个或多个特性相关联。如在图 3 和 4 中所示,可以选择某些墨水和提供一种特性,例如,选择手写墨水词“conceived in liberty”,然后使之倾斜,并选择手写墨水词“created equal”(如图 3 中的虚线框所示),然后增加它的字号。InkEdit(墨水编辑)控制可以进一步使墨水与它相关联的特性存储而作为墨水目标。因此,例如,可以把在显示空间 302 中的墨水存储在一个或多个墨水目标中,而可以把词“conceived in liberty”存储在墨水目标中并与倾斜特性相关联。InkEdit(墨水编辑)控制可以根据接收到的墨水数据使墨水得以显示。在一个实施例中,例如,在接收到墨水的显示空间 302 中的相同位置处的显示空间 302 中显示墨水。

[0039] 墨水可以在显示空间 302 中保持显示(即,余晖),或可以识别墨水和/或把它转换成文本。这种识别和转换可以立即发生,或在识别时间过去之后和/或根据命令(例如,来自用户或来自应用程序。在墨水被识别和/或根据时间过去之后条件对墨水转换情况下,这个识别时间可以是所要求的任何时间量,诸如约 2000 毫秒,或在约 100 毫秒到约 5000 毫秒的范围内,或在约 200 毫秒到约 2000 毫秒的范围内。可以根据提起输入笔 204 使之离开显示表面 202 的时间和/或根据笔划的结束来起用于对识别时间进行计时的定时器,并且在时间条件发生之前可以根据输入笔 204 回到显示表面 202 而取消计时。可以使定时器起动的其它事件包括输入笔 204 停止在显示表面 202 上的运动,来自输入笔 204 的手势,诸如按钮和/或等等的其它输入。根据一个或多个模式开关的设置,决定可以或不可以识别和/或显示墨水作为打印的文本。可以使一个或多个模式开关与一个或多个显示单元

(例如,显示单元 303) 相关联或可以使用户不知道(即,不显示在显示器上)。当开关是第一种设置时,将不作为文本而识别或显示墨水。在某些实施例中,根据特定的命令仍可以识别和 / 或显示墨水作为文本。当开关是第二种设置时,可以识别和 / 或显示墨水作为文本(如这里所述,立即、在时间过去之后、或根据命令)。当显示墨水作为文本时,可以显示从识别产生的文本来代替显示空间 302 中的墨水,致使被替代的墨水不再显示在显示空间 302 中。图 5 示出已经被识别和转换成文本的显示空间 302 中的一部分墨水,而图 6 示出已经被识别和转换成文本的显示空间 302 中的所有墨水。在这些示意图中,手写体不是原始手写墨水,而印刷字体表示经识别的文本。

[0040] 可以通过单个识别器或诸如识别器的集合之类的多个识别器识别墨水。识别器可以包括一个或多个手势识别器和 / 或文本识别器。可以进一步使待识别墨水的选择与特定识别上下文相关联,所述特定识别上下文可以包括连同一个或多个工作识别特性的所谓的“阶乘识别符 (factoid)”特性。可以认为阶乘识别符特性是提供给识别上下文的一组“线索”,有助于使墨水上下文的识别更为正确。可以定义表示执行墨水识别能力、获取识别结果和 / 或获取另外识别结果的识别器上下文目标。识别器上下文目标可以启动安装在系统中的各种识别器执行墨水识别而适当地处理输入。可以执行包括后台识别和前台识别的至少两类识别。在系统的后台处理中发生后台识别,并可能由于其它系统事件(由用户创建或其它情况)而停止。相反,前台识别一般由用户启动,并且直到完成识别才停止。识别器上下文目标可以接收待识别的墨水笔划,阶乘识别符特性可以定义输入墨水和所要求的识别输出上的限制和 / 或其它参数。例如,可能设置的限制包括在识别期间要使用的语言、字典和 / 或文法。当联系一种形式使用 InkEdit(墨水编辑)控制时,对于该形式中的每个数据输入字段可以设置不同的识别器上下文和 / 或阶乘识别符。各种数据输入字段对于某些信息组可能是特定的,诸如具有数字、加号、虚线和括号的电话号码字段;只具有数字和虚线的邮政编码字段;只具有大写字母的州的缩写;通用资源定位器 (URL);以及等等。

[0041] 具有不同的识别器上下文和 / 或阶乘识别符的一个潜在的优点在于:在某些实施例中,可以调整发生识别的时间帧或什么触发识别。例如,对于在一种形式中的名称或说明字段,有可能要求识别发生于每个字的结束处。然而,当使用州、街道 / 公寓号码或邮政编码字段(例如)时,有可能要求识别发生于写入每个字符之后。在某些情况中,可以使用一个字符到一个字符的识别,以提供比通过同时识别一个字符组可以得到的识别正确度更高的识别正确度。

[0042] InkEdit(墨水编辑)控制可以进一步提供手势支持,并可以根据手势产生事件。参考图 7,可以支持各种手势,诸如所示的示例手势 701-704。例如,手势 701 可以表示托架返回,手势 702 可以表示跳格命令,手势 703 可以表示空格字符,以及手势 704 可以表示退格命令。可能有许多其它手势。

[0043] InkEdit(墨水编辑)控制可以进一步提供校正用户接口,该接口允许用户按需要观看另外的识别结果,使用屏幕上键盘和 / 或使用字符、字母和 / 或块文本识别器。还有,InkEdit(墨水编辑)控制可以允许使用相同的存储和装载机构象传统 Windows Form Rich Textbox 控制那样来保持和装载它的数据。

[0044] InkEdit API(墨水编辑应用程序接口)

[0045] InkEdit(墨水编辑)控制通过它的应用程序接口 (API) 向用户揭示许多功能。根

据主机应用程序,可以提供 InkEditAPI(墨水编辑应用程序接口)的各种风味。参考图 8,示例系统可以包括下列一个或多个:ActiveX 主机应用程序 801、Win32 主机应用程序 802、和 / 或通用语言运行时 (CLR) 主机应用程序 803。此外,可以定义 InkEdit ActiveX 控制 804,并且可以与 ActiveX 主机应用程序 801 对接。可以进一步定义 InkEdit Win32 控制 805,并可以与 Win32 主机应用程序 802 对接。可以定义 InkEdit WinForms 控制 806,并可以与 CLR 主机应用程序 803 对接。最后,可以定义 RichEdit 4.5 Win32 控制 807,它与 InkEdit(墨水编辑)控制 804、805、806 的各种风味中的任何一种或所有的相对接。InkEdit Win32 控制 805 可以是其它两种控制 (ActiveX 控制 804 和 WinForms 控制 806) 的基础。在一个实施例中,可以在 Win32 控制 805 中执行关键功能,诸如墨水的收集、具有识别器的迭代和 / 或进行 RichEdit Win32 控制 807 的子分类。ActiveX InkEdit 控制 804 可以使用通过 Win32 InkEdit 控制 805 定义的 C++ 分类,并可以基于该功能来创建 ActiveX 支持。可以从 Winforms Rich TextBox 808 得到 Winforms InkEdit 控制 806,并可以用这里讨论的加墨水功能使它扩展。例如,首先通过在 Rich TextBox 808 提供的现有 API 单元的顶层上装载代替 RichEdit 控制的 Win32 InkEdit 控制和按需要添加新方法、特性和 / 或事件,可以扩展 Winforms InkEdit 控制 806 的功能。注意图 8 描绘在各种主机环境中控制之间关系的示例组,但绝不是把控制之间的箭头作为限制。

[0046] 现在参考图 9 讨论 InkEdit(墨水编辑)控制的 API(应用程序接口)。在图 9 中,通过一个方框来表示 InkEdit(墨水编辑)控制 901,并示出 API 的各种单元(或功能上分组的单元),标有射出和 / 或进入表示 InkEdit(墨水编辑)控制 901 的方框的箭头 940-960。一般,进入 InkEdit(墨水编辑)控制 901 方框的箭头是指大部分修改 InkEdit(墨水编辑)控制 901(例如,通过改变它的特性中之一)和 / 或把信息提供给 InkEdit(墨水编辑)控制 901 的其它单元的 API 单元(或功能上分组的单元)。从 InkEdit(墨水编辑)控制 901 方框射出的箭头是指大部分表示标志或某些其它信息(通过 InkEdit(墨水编辑)控制 901 提供给它的环境)的 API 单元(或功能上分组的单元)。然而,箭头的方向是示意性的,并不是作为限制,所以并不阻止用进入 InkEdit(墨水编辑)控制 901 的箭头来表示 InkEdit(墨水编辑)控制 901 把信息提供给它的环境。同样,并不阻止从 InkEdit(墨水编辑)控制 901 射出的箭头修改或提供到 InkEdit(墨水编辑)控制 901 的信息。图 9 进一步示出 InkEdit(墨水编辑)控制 901 的多个特性 902-921。可以按 InkEdit 型控制的任何风味(包括但是不限于,这里讨论的 Win32、ActiveX 和 .NET 风味)的任何组合或子组合来利用下面讨论的 API 单元。

[0047] 在示例实施例中的 InkEdit API(墨水编辑应用程序接口)具有按任何组合或子组合的下列细目和结构(未示出)中的某一些或全部。例如,外形细目定义一个或多个值,该值规定当显示时 InkEdit(墨水编辑)控制 901 的外形是平的还是 3- 维的。边界格式细目定义一个或多个值,该值规定 InkEdit(墨水编辑)控制 901 是否具有边界。墨水-模式细目定义一个或多个值,该值规定用于汲取墨水的收集模式设置——是否禁止墨水收集,是否只收集墨水,或是否收集墨水和手势两者。插入模式细目定义一个或多个值,该值规定墨水是如何插入到 InkEdit(墨水编辑)控制 901 中的,是作为墨水还是作为经识别的文本。InkEdit 状态细目定义一个或多个值,该值规定 InkEdit(墨水编辑)控制 901 是否空闲、收集墨水或识别墨水。装载 / 存储细目定义一个或多个值,该值规定是否装载文件和 /

或作为 Rich TextFormat (RTF) 文件、文本文件或其它格式的文件而存储。鼠标 - 按钮细目定义一个或多个值, 该值规定按下或正在按下哪个鼠标按钮。注意, 除非另行规定, 这里对于鼠标或鼠标按钮的所有参考都同样适用于输入笔或输入笔按钮。滚动条细目定义一个或多个值, 该值规定 InkEdit(墨水编辑) 控制是否具有水平和 / 或垂直滚动条。对准细目定义一个或多个值, 该值规定显示的段落是否对准 InkEdit(墨水编辑) 控制左或右页边线, 或在左和右页边线之间。笔划 - 信息结构包括有关特定笔划的信息, 诸如使用哪个光标来创建笔划以及把笔划存储在哪里 (例如, 作为特定笔划目标)。手势 - 信息结构包括有关特定手势的信息, 诸如使用哪个光标来创建手势、构成手势的笔划, 和 / 或把手势存储在哪里 (例如, 作为特定手势目标)。可以在 InkEdit(墨水编辑) 控制的 Win32 风味中使用特别成功的笔划 - 信息和手势 - 信息结构。识别 - 结果结构包括有关文本识别结构的信息, 并且根据已准备的识别结果而发送。例如, 通过分立的通知消息, 可以提供有关这些本文讨论的结构的一些或全部使用在哪里的通知。

[0048] 在示例实施例中的 InkEditAPI 还具有按任何组合或子组合的、可以设置的以及可以返回它们表示的信息的一个或多个下列特性。例如, 外形特性 902 表示所显示的 InkEdit(墨水编辑) 控制 901 是平的还是 3- 维的。后台 - 颜色特性 903 表示 InkEdit(墨水编辑) 控制 901 的后台颜色。边界格式特性 904 表示 InkEdit(墨水编辑) 控制 901 是否具有边界。创建 - 参数特性 905 表示当创建 InkEdit(墨水编辑) 控制 901 时创建参数。光标特性 906 表示当鼠标指针在 InkEdit(墨水编辑) 控制 901 上时所显示的光标。画图 - 属性特性 907 表示当画图和 InkEdit(墨水编辑) 控制 901 中显示墨水 (尚未被识别为文本的墨水) 时使用的缺省画图属性或当画墨水时施加到墨水的画图属性。滚动条禁止特性 908 表示是否启动或禁止 InkEdit(墨水编辑) 控制 901 中的滚动条。拖拉 - 图标特性 909 表示待显示作为拖拉 - 和 - 丢下操作中指针的图标。阶乘识别符 910 特性表示阶乘识别符 (这里进一步讨论), 识别器用它来限制它对于识别结果的搜索。各种字体和文本特性 911 表示 InkEdit(墨水编辑) 控制 901 所显示文本的字体以及当前选择文本的或在插入点处的字体的名称和字号。其它字体和文本特性 911 表示当前选择的文本 (或在插入点处) 是否为加粗、倾斜或下划线。再其它字体和文本特性 911 表示当前选择的文本 (或在插入点处) 是否出现在基线上、作为上标、或作为下标, 它们的对准, 以及当前选择的文本或在插入点处的颜色。各种墨水 - 模式特性 912 表示当在控制上画图时如何收集墨水以及是否禁止收集墨水, 是否只收集墨水, 或是否收集墨水和手势两者。控制 - 锁定特性 913 表示是否可以编辑 InkEdit(墨水编辑) 控制 901 的内容。各种鼠标特性 914 表示要显示的当前定制鼠标图标以及当鼠标指针在 InkEdit(墨水编辑) 控制 901 的图形描绘上时所显示的鼠标指针类型。多线特性 915 表示 InkEdit(墨水编辑) 控制 901 是否为多线控制。各种识别器特性 916 表示使用哪个识别器来进行识别, 以及在已经结束墨水笔划之后开始文本识别的时间量。滚动条特性 917 表示在 InkEdit(墨水编辑) 控制 901 中显示的滚动条的类型。墨水目标特性 918 表示在当前选择的文本中的墨水目标。设置这个墨水 - 目标特性可能导致从输入的墨水目标清单识别的识别结果替代当前的选择。各种选择 - 文本特性 919 表示在 InkEdit(墨水编辑) 控制 901 中的当前选择的文本、所选择的字符数、所选择的经 Rich Text Format (RTF) 格式化的文本以及所选择文本的开始点。各种受控制文本特性 920 表示在 InkEdit(墨水编辑) 控制 901 的文本方框中显示的当前文本以及在包括所有 RTF 格

式化代码的 InkEdit(墨水编辑)控制 901 中的文本。状态特性 921 表示 InkEdit(墨水编辑)控制 901 是否空闲、收集墨水或识别墨水。

[0049] 在示例实施例中的 InkEdit API 还具有按任何组合或子组合的多个相关联的消息、事件以及方法。由于 InkEdit(墨水编辑)控制是超级的 RichEdit 控制,所以直接传递每个 RichEdit 消息(在大多数情况中),具有如 RichEdit 中的相同效果。这也应用于事件通知消息,它把已经发生特定事件通知 InkEdit(墨水编辑)控制的父亲窗口。

[0050] 例如,根据光标尖端(例如,输入笔 204 的尖端)物理地接触数字化表面(例如,表面 202)而发送光标-向下消息 940。根据正完成的笔划发送笔划-完成消息 941,并发生笔划-完成事件。根据笔划-完成事件的发生而发生笔划-完成方法。根据正完成的手势发送手势-完成消息 942,这可以通过手势-完成事件来指示。

[0051] InkEdit API 还可以具有与识别有关的事件、方法和消息 943。根据已经发生的识别发送这种与识别有关的消息,或得到或设置所使用的识别器。其它与识别有关的消息,在识别时间过去将导致发生识别之前,具体地强迫进行识别。与识别有关的事件,根据待识别的应用-特定的手势以及根据一般的识别而发生。与识别有关的方法,根据正发生的识别事件或规定应识别笔划的收集以及要返回该识别结果而发生。

[0052] InkEdit API 可以进一步具有根据经点击 InkEdit(墨水编辑)控制 901 而发生的事件 944,根据按下或放开一个键而发生的事件 945,根据在 InkEdit(墨水编辑)控制 901 上的鼠标指针和/或输入笔,以及按下、放开、或双击而发生的事件 946 以及根据在 InkEdit(墨水编辑)控制 901 上移动的鼠标指针而发生的事件 947。

[0053] InkEdit API 可以进一步具有消息 948,它用于根据在墨水编辑状态细目中定义的值而获取 InkEdit(墨水编辑)控制 901 的状态;方法 949,它把特定类型的文件装载到 InkEdit(墨水编辑)控制 901 中或把 InkEdit(墨水编辑)控制 901 的内容存储到特定类型的文件中;以及用于处理 Windows 消息的方法 950。

[0054] InkEdit API 可以进一步具有消息 951,它根据在墨水模式细目中定义的值而获取或设置 InkEdit(墨水编辑)控制 901 的加墨水模式;以及消息 952,它根据在墨水模式细目中定义的值而获取或设置 InkEdit(墨水编辑)控制 901 的墨水插入模。

[0055] InkEdit API 可以进一步具有消息 953,它获取当前画图属性或设置要在 InkEdit(墨水编辑)控制 901 中使用的将来画图属性;以及消息 954,它获取或设置 InkEdit(墨水编辑)控制 901 的识别时间。例如,识别时间可以按毫秒来测量。

[0056] InkEdit API 可以进一步具有与手势状态有关的消息和方法 955。定义与手势状态有关的消息获取或设置用于 InkEdit(墨水编辑)控制 901 的手势状态。与手势状态有关的方法使用手势状态的状态来限制可能被 InkEdit(墨水编辑)控制 901 识别的手势的设置。

[0057] InkEdit API 可以进一步具有消息 956,它获取或设置 InkEdit(墨水编辑)控制 901 待使用的识别器;消息 957,它获取或设置用于识别的阶乘识别符;消息 958,它获取或设置当前选择的墨水;消息 959,它获取或设置所显示的鼠标图标;以及消息 960,它获取或设置要显示的鼠标指针。

[0058] InkEdit API 可以进一步具有构造者 961,用于创建新 InkEdit(墨水编辑)控制;以及当为 InkEdit(墨水编辑)控制 901 创建句柄时发生的方法 962。在 InkEdit(墨水编

辑)控制的 .NET 风味中,可以认为该方法是构造者。InkEdit API 可以进一步具有根据在控制变化中的文本选择而发生的选择改变的事件。

[0059] 这里所描述的实施例本发明各方面的示例系统和方法只是作为例子而示出,可以理解,本发明当然不限于这些实施例。特别是在上述学说的指导下,熟悉本技术领域的人员可以进行各种修改。例如,可以单独利用上述实施例中的每个单元,或与其它实施例中的的单元组合。虽然已经使用所附的权利要求书来定义本发明,但是示出这些权利要求在于打算使本发明按任何组合或子组合来包括这里所描述的单元和步骤。因此,本发明可由任何数目的变化组合来定义,它按各种组合或子组合而结合包括说明书、权利要求书和附图的规格中的一个或多个单元。对于有关技术领域中的人员是显而易见的,在本规格的指导下,可以利用本发明的各方面的独立的或与这里定义的一个或多个单元或步骤组合的另外组合作为本发明的修改或变型或作为本发明的一部分。打算使这里包括的本发明的书面说明包含所有这种修改和变型。还有,应该认识到,虽然这里提供目标和其它 API 单元的各种名称,但是这种名称只是作为示意,可以使用任何名称而不偏离本发明的范围。

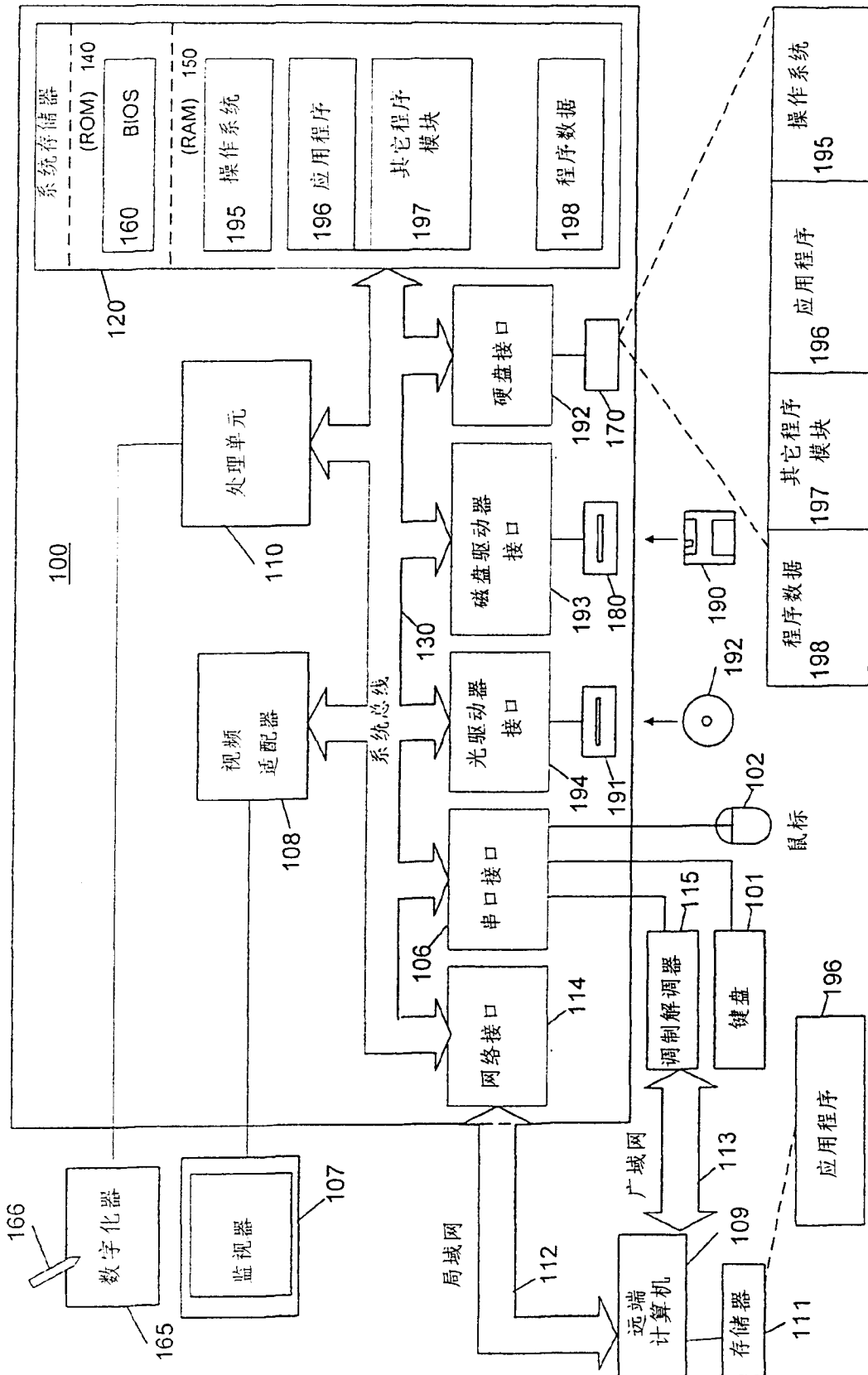


图 1

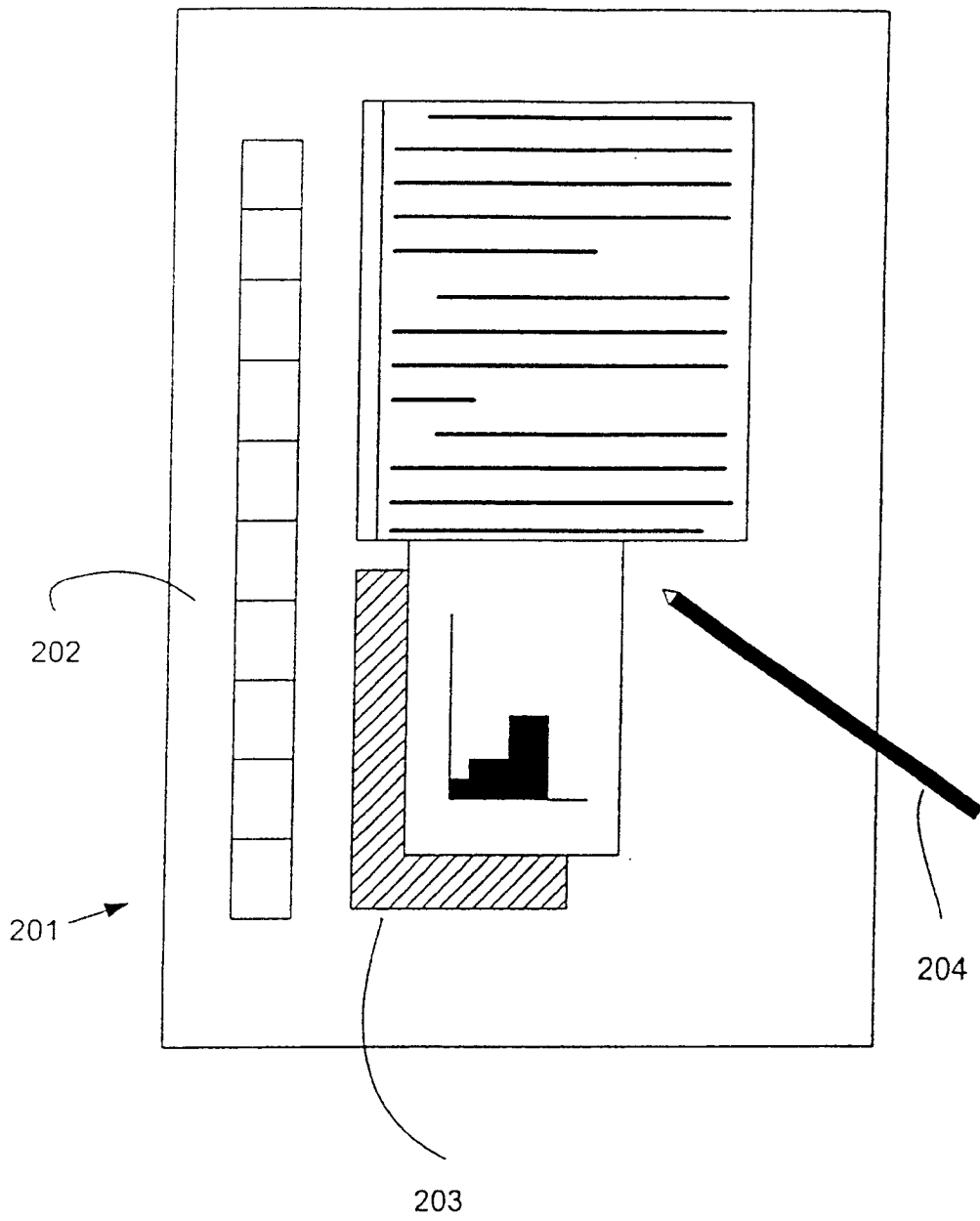


图 2

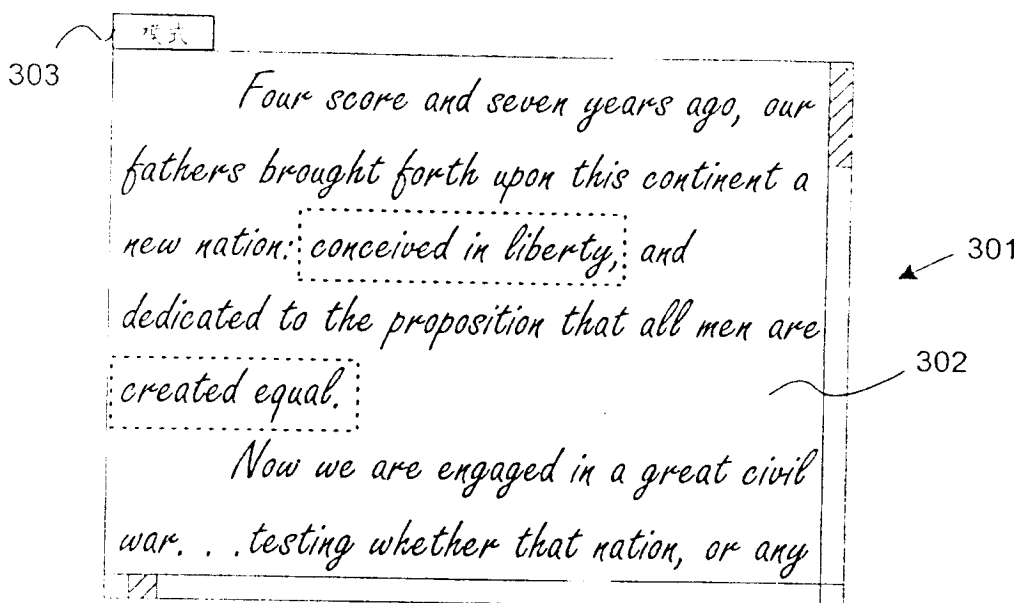


图 3

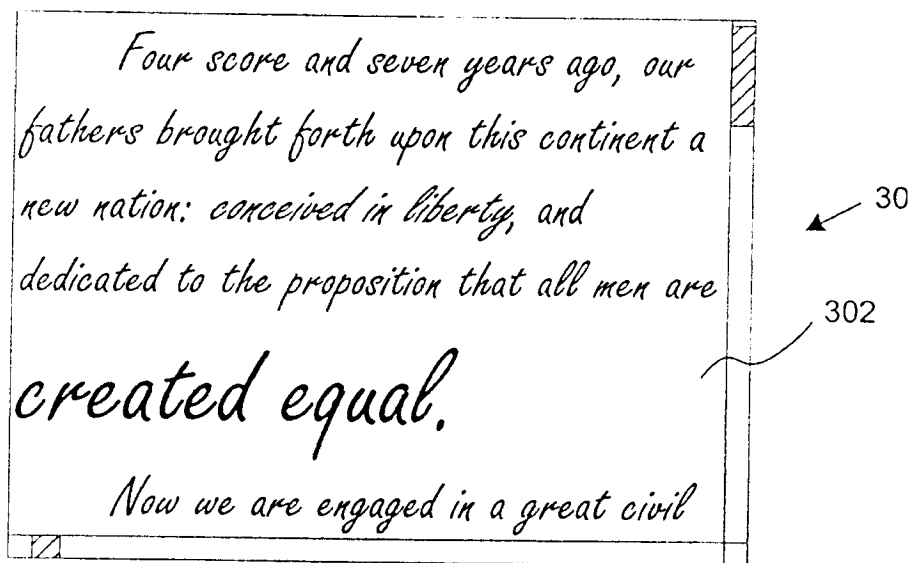


图 4

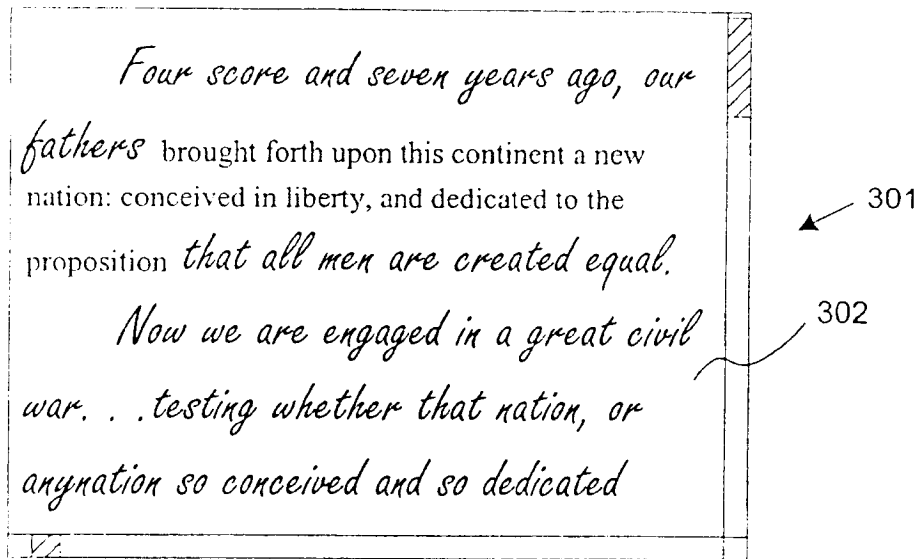


图 5

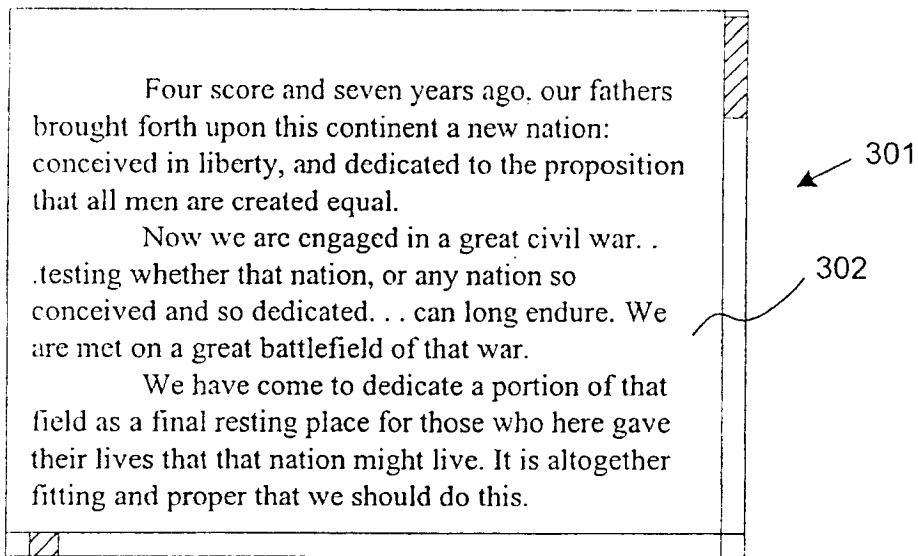


图 6

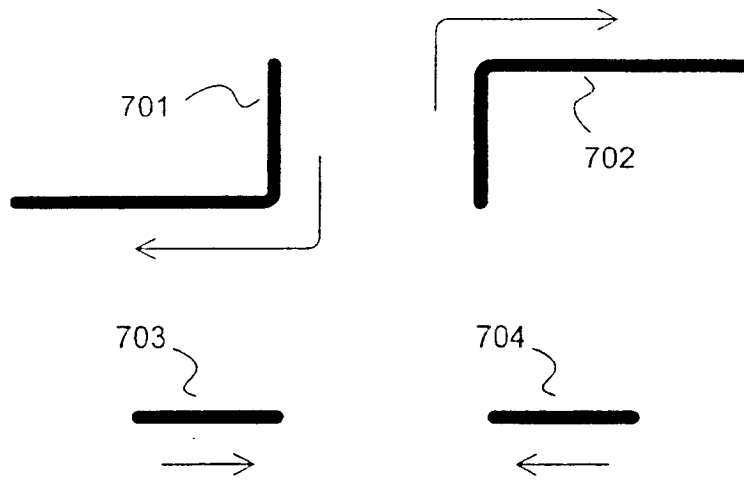


图 7

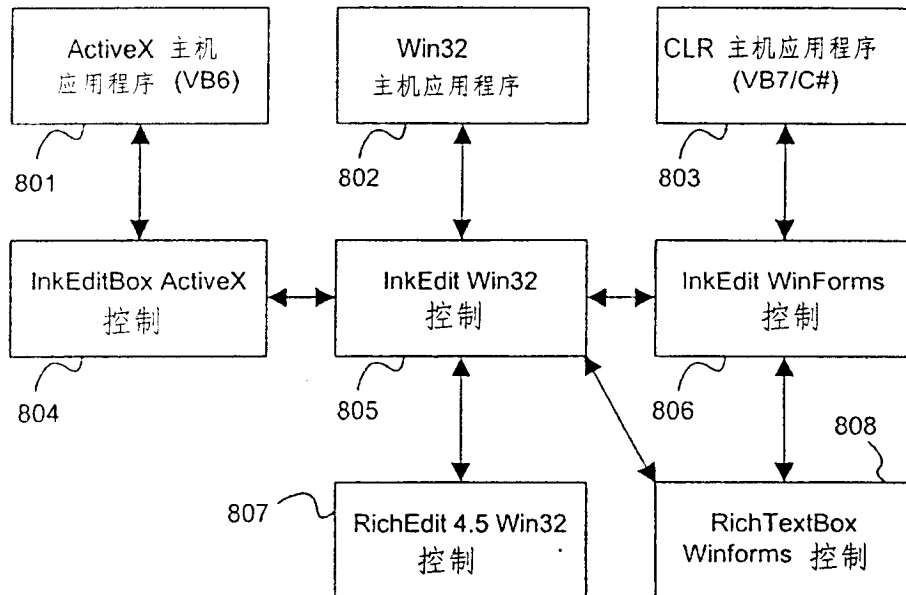


图 8

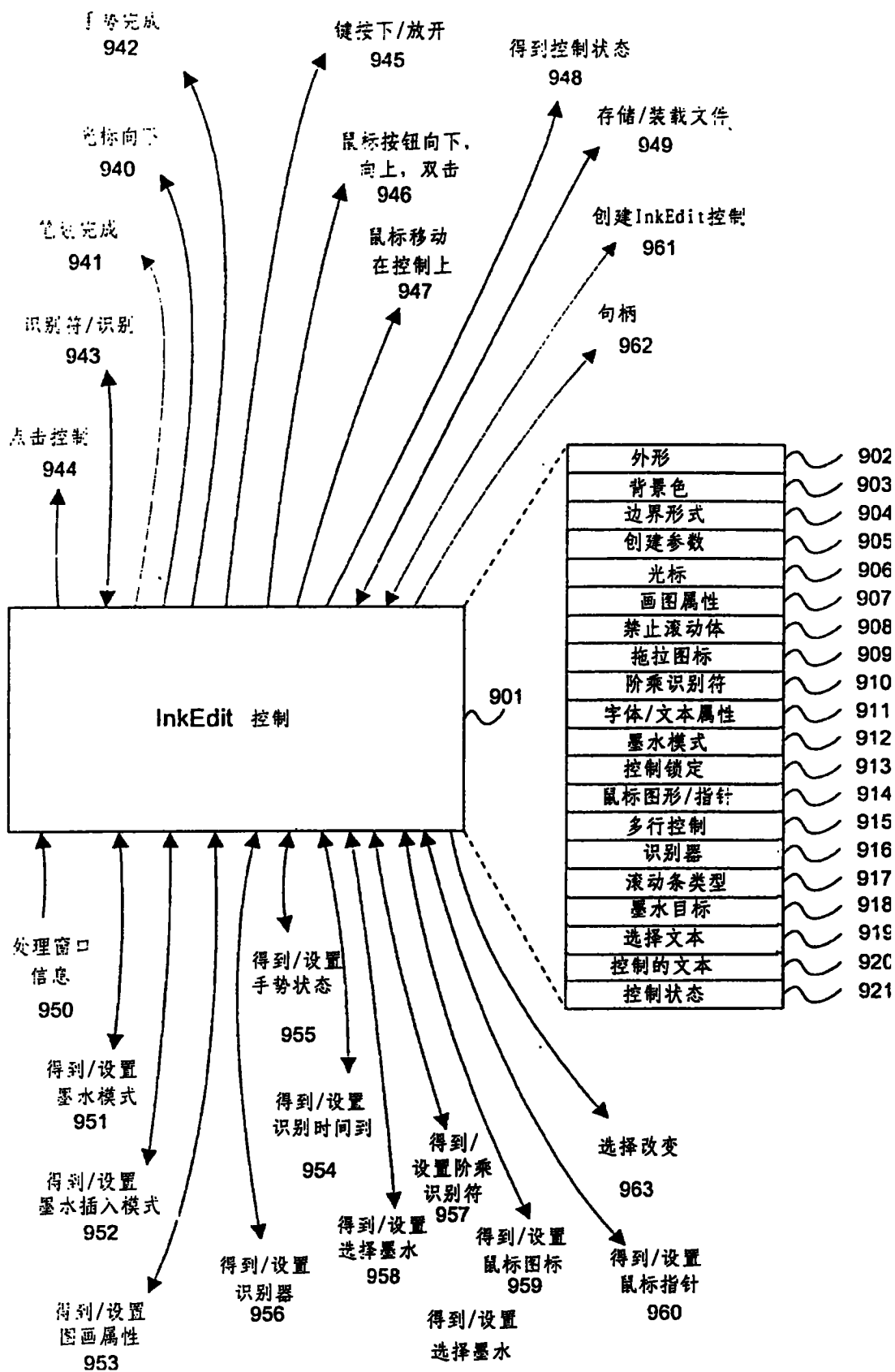


图 9