



[12] 发明专利申请公布说明书

[21] 申请号 200580009335.9

[43] 公开日 2007年3月21日

[11] 公开号 CN 1934539A

[22] 申请日 2005.2.4
 [21] 申请号 200580009335.9
 [30] 优先权
 [32] 2004.3.31 [33] US [31] 10/814,773
 [86] 国际申请 PCT/US2005/003386 2005.2.4
 [87] 国际公布 WO2005/103900 英 2005.11.3
 [85] 进入国家阶段日期 2006.9.22
 [71] 申请人 咕果公司
 地址 美国加利福尼亚州
 [72] 发明人 邓建功 斯蒂芬·劳伦斯
 克里斯托弗·M·普林斯
 米哈伊·F·约内斯库

[74] 专利代理机构 北京律盟知识产权代理有限责任
 公司
 代理人 王允方 刘国伟

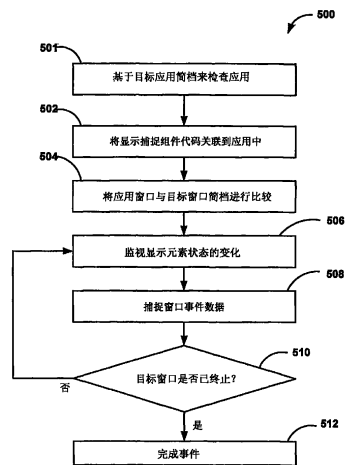
权利要求书5页 说明书21页 附图5页

[54] 发明名称

用于监控应用程序中的事件的基于简档的捕捉组件

[57] 摘要

一计算机系统中的一索引系统可包含应用程序、一捕捉处理器、一队列、一搜索引擎和一显示处理器。所述索引系统捕捉用户与所述应用程序互动的事件。事件排成队列，且如果可编入索引，那么将所述事件编入索引并存储以供用户通过所述搜索引擎进行访问。所述捕捉处理器中的捕捉组件可包含一键盘捕捉组件，其处理用户键击来确定事件。一显示捕捉组件从与所述应用程序关联的窗口捕捉事件数据。可以一轮询进度表或基于窗口元素的状态变化来捕捉显示事件数据。为了确定目标应用程序和所关注的窗口应用程序，可使用应用程序简档和窗口简档。



1. 一种基于计算机的方法，用于在一计算机系统中从一目标应用程序的一目标窗口捕捉事件数据的，所述方法包括：
 将窗口信息与一目标窗口简档进行比较，该目标窗口简档与所述目标应用程序的所述目标窗口关联；及
 一旦检测到一与所述窗口中一元素关联的一状态的一状态变化时，从所述目标窗口捕捉事件数据。
2. 根据权利要求 1 所述的方法，其进一步包括：
 从一第一应用程序接收窗口创建信息；及
 确定所述第一应用程序是否为所述目标应用程序的一实例。
3. 根据权利要求 2 所述的方法，其中确定包括将所述窗口创建信息与一与所述目标应用程序关联的目标应用程序简档进行比较。
4. 根据权利要求 2 所述的方法，其进一步包括：
 确定一与一现有窗口关联的第二应用程序是否为所述目标应用程序的一实例。
5. 根据权利要求 4 所述的方法，其中确定包括将来自所述第二应用程序的应用程序相关数据和与所述目标应用程序关联的所述目标应用程序简档进行比较。
6. 根据权利要求 4 所述的方法，其中所述应用程序相关数据包含与所述应用程序关联的一窗口类别名、一路径名或一注册表中的至少一者。
7. 根据权利要求 1 所述的方法，其中捕捉事件数据进一步包括根据一事件计划以一与一事件关联的数据结构存储捕捉的窗口信息。
8. 根据权利要求 7 所述的方法，其进一步包括索引并存储所述事件。

9. 根据权利要求 1 所述的方法，其进一步包括：

响应与所述窗口中所述元素关联的所述状态的一后续状态变化，以自所述窗口的附加的捕捉的事件数据更新一事件的一捕捉状态。
10. 根据权利要求 1 所述的方法，其中所述窗口中的所述元素包含一按钮、一下拉菜单或一超链接中的至少一者。
11. 根据权利要求 2 所述的方法，其中所述窗口创建信息包含与所述第一应用程序关联的一窗口类别名、一路径名或一注册表中的至少一者。
12. 根据权利要求 3 所述的方法，其中所述目标应用程序简档包含与所述目标应用程序关联的一窗口类别名、一路径名或一注册表中的至少一者。
13. 根据权利要求 1 所述的方法，其中所述目标应用程序与复数个目标窗口简档关联，每一目标窗口简档包括识别从其中捕捉事件数据的目标窗口的显示元素信息。
14. 根据权利要求 1 所述的方法，其进一步包括：

在一目标应用程序内关联显示捕捉代码。
15. 根据权利要求 14 所述的方法，其中所述显示捕捉代码修改所述目标应用程序对于操作系统请求的回应。
16. 根据权利要求 15 所述的方法，其中所述操作系统请求包含来自基于一与所述操作系统关联的应用程序编程界面的其它应用程序的消息。
17. 根据权利要求 15 所述的方法，其中所述操作系统请求包含对于所述目标窗口中的文本的请求。
18. 根据权利要求 14 所述的方法，其中关联显示捕捉代码包含设置一关于所述目标应用程序处理的挂钩函数。

19. 根据权利要求 14 所述的方法，其中所述挂钩函数经配置以在一操作系统与所述目标应用程序之间截取消息。
20. 一种确定索引事件的基于计算机的方法，其包括：
 - 接收与一计算机系统中的计算机应用程序关联的复数个显示呼叫；
 - 基于一目标窗口简档来处理所述复数个显示呼叫以便确定一目标窗口；
 - 至少部分地基于从所述目标窗口捕捉的数据来确定一事件；和
 - 将从所述目标窗口捕捉的所述数据编入索引并存储所述事件。
21. 根据权利要求 20 所述的方法，其进一步包括基于复数个已存储的目标应用程序简档来确定一目标应用程序，且此外其中所述复数个显示呼叫与目标应用程序关联。
22. 根据权利要求 20 所述的方法，其进一步包括捕捉一用户输入到所述目标窗口的文本，且其中所述事件包含所述捕捉的文本。
23. 根据权利要求 20 所述的方法，其进一步包括在处理每一显示呼叫之后更新一捕捉状态。
24. 根据权利要求 23 所述的方法，其中更新是响应于与所述目标窗口的一显示元素关联的一状态变化。
25. 根据权利要求 20 所述的方法，其进一步包括确定是否将所述事件编入索引。
26. 根据权利要求 20 所述的方法，其中将所述事件编入索引并存储所述事件在终止所述目标窗口之后发生。
27. 根据权利要求 20 所述的方法，其中处理所述复数个显示呼叫以便确定一目标窗口包括：使用显示呼叫分析写入到显示器的复数个显示项的 x , y 坐标、长度和相对位置中的一者或多者，并将它们与所述存储的目标窗口简档进行比较。

28. 一种含有用于从一计算机系统的一目标应用程序的一目标窗口捕捉事件数据的程序代码的计算机可读媒质，其包括：
- 用于将窗口信息与一与所述目标应用程序的所述目标窗口关联的目标窗口简档进行比较的程序代码；和
 - 用于当检测到一与一元素关联的状态的一状态变化时从所述目标窗口捕捉事件数据的程序代码。
29. 一种含有用于确定索引事件的程序代码的计算机可读媒质，其包括：
- 用于接收与一计算机系统中的应用程序关联的复数个显示呼叫的程序代码；
 - 用于基于一目标窗口简档来处理所述复数个显示呼叫以便确定一目标窗口的程序代码；
 - 用于至少部分地基于从所述目标窗口捕捉的数据来确定一事件的程序代码；
 - 和
 - 用于将从所述目标窗口捕捉的所述数据编入索引并存储所述事件的程序代码。
30. 一种方法，其包括：
- 将来自与在一计算机系统中执行的窗口关联的应用程序的应用程序相关数据和与复数个目标应用程序关联的已存储的目标应用程序简档进行比较；
 - 通过所述计算机系统的一操作系统来预订窗口创建事件；
 - 从一应用程序接收窗口创建请求信息；
 - 将所述窗口创建请求信息与所述已存储的目标应用程序简档进行比较；
 - 建立到达与一目标应用程序简档匹配的每一应用程序的处理过程的一具有捕捉组件代码的挂钩函数；
 - 将每一目标应用程序的每一处理过程的窗口元素信息和与所述目标应用程序的目标窗口关联的复数个已存储的目标窗口简档进行比较；
 - 监控与一目标窗口简档匹配的窗口中的显示元素相关联状态的状态变化；
 - 当检测到所述受监控状态的一状态变化时，从每一匹配的窗口将事件数据捕捉到与每一匹配的窗口关联的一事件中，其中捕捉包括基于所述捕捉组件代码来修改所述匹配的应用程序的应用程序行为；和

响应于所述匹配的窗口的关闭而完成与所述匹配的窗口关联的所述事件。

31. 一种方法，其包括：

通过将来自与在一计算机系统中执行的一窗口关联的一应用程序的应用程序相关数据与一与一目标应用程序关联的目标应用程序简档进行比较来确定所述应用程序是否为一目标应用程序的一实例，其中所述目标应用程序经配置以响应于一窗口文本请求而提供一空字符串；

建立到达所述应用程序的一处理过程的一具有捕捉组件代码的挂钩函数，所述捕捉组件代码经配置在所述应用程序与一操作系统之间截取消息；

通过将来自所述应用程序的一窗口的窗口信息与一与所述目标应用程序的一目标窗口关联的目标窗口简档进行比较来确定与所述应用程序关联的所述窗口是否为一目标窗口的一实例；

监视与所述窗口中的显示元素关联的状态的状态变化；

当检测到所述受监视的状态的一状态变化时捕捉所述窗口内的文本，其中捕捉包括截取一窗口文本请求和在所述应用程序响应所述窗口文本请求之前提供所述窗口文本；和

当所述窗口终止时将所述捕捉的文本编入索引。

32. 一种用于从一计算机系统的一目标应用程序的一目标窗口捕捉事件数据的计算机系统，所述系统包括：

用于将窗口信息与一与所述目标应用程序的所述目标窗口关联的目标窗口简档进行比较的构件；和

用于当检测到一与一元素关联的状态的一状态变化时从所述目标窗口捕捉事件数据的构件。

用于监控应用程序中的事件的基于简档的捕捉组件

本申请案是 Lawrence 等人 2004 年 3 月 31 日申请的标题为“Methods and Systems for Information Capture”的美国申请案序号第 10/8.14,773 号的部分接续申请案 (“CIP”), 且主张基于该案的优先权。

技术领域

本发明大体上涉及搜索引擎。更明确地说, 本发明涉及信息捕捉的方法和系统。

背景技术

用户使用例如个人计算机、个人数字助理、移动电话或类似装置的客户端装置产生和访问例如电子邮件、网页、文字处理文档、电子数据表文档、即时通讯消息、演示文档、多媒质文件和类似项目的大量项目。一些项目存储在耦合到客户端装置、由客户端装置访问或与客户端装置关联的一个或一个以上存储装置上。用户有时希望搜索存储装置中的项目。

常规客户端装置搜索应用程序会使客户端装置的性能显著降级。例如, 某些常规客户端装置搜索应用程序通常使用成批处理将所有项目编入索引, 这导致成批索引期间客户端装置的性能显著减慢。另外, 成批处理仅周期性地发生。因此, 当用户执行搜索时, 最新近的访问或所创建的项目有时不包含在结果中。此外, 如果将成批索引安排在客户端装置不开启的时间, 那么在一段持续的时间内可能不会发生成批索引。在此情况下, 与客户端装置相关的项目索引可能变得明显过时。常规客户端装置搜索应用程序还可能需要在每次成批索引时重建索引或建立新的局部索引, 并执行可使用许多客户端装置资源的合并操作。

常规客户端装置搜索应用程序在操作时有时还会使用大量系统资源, 从而导致客户端装置的性能减慢。

另外, 常规客户端装置搜索应用程序通常需要用户向搜索应用程序明确地提供搜索查询来产生结果, 且可能限于检查特定应用程序的文件的文件名或内容。

发明内容

一种在计算机系统中从目标应用程序的目标窗口捕捉事件数据的基于计算机的方法包含：将窗口信息与与目标应用程序的目标窗口关联的目标窗口简档进行比较，和当检测到与窗口中元素关联的状态的状态变化时从目标窗口捕捉事件数据。本发明实施例包含信息捕捉、存储和索引的方法、系统和计算机可读媒质。

在一个实施例中，一种在计算机系统中从目标应用程序的目标窗口捕捉事件数据的方法还包含：从应用程序接收窗口创建信息，和确定应用程序是否为目标应用程序的实例。在一个实施例中，为了进行此确定，将窗口创建信息与与目标应用程序关联的目标应用程序简档进行比较。

在另一实施例中，一种在计算机系统中将事件编入索引的基于计算机的方法包含从目标窗口接收所捕捉的事件数据。目标窗口与目标应用程序关联。所述方法还包含基于目标应用程序并且基于用户输入来确定事件。在此实施例中，所述方法还包含确定是否将捕捉的事件数据编入索引并存储事件。

本发明的另一方面提供一种通过接收复数个显示呼叫来确定索引事件的基于计算机的方法。显示呼叫与计算机系统中的计算机应用程序关联。所述方法包含处理复数个显示呼叫来确定目标窗口。处理基于目标窗口简档。所述方法还包含至少部分地基于从目标窗口捕捉的数据来确定事件。

在又一实施例中，本发明提供一种将应用程序相关数据与与复数个目标应用程序关联的已存储的目标应用程序简档进行比较的方法。应用程序相关数据来自与在计算机系统中执行的窗口关联的应用程序。所述方法还包含通过计算机系统的操作系统预订窗口创建事件。从应用程序接收窗口创建请求信息。将窗口创建请求信息与已存储的目标应用程序简档进行比较。接着，建立到达每一目标应用程序的处理过程的挂钩函数。将每一目标应用程序的每一处理过程的窗口元素信息与与目标应用程序的目标窗口关联的复数个已存储的目标窗口简档进行比较。所述方法还包含监控与与目标窗口简档匹配的窗口中的显示元素关联的状态的状态变化。在此实施例中，所述方法包含当检测到受监控状态的状态变化时从每一匹配的窗口捕捉事件数据。将事件数据捕捉到与每一匹配的窗口关联的事件中。此捕捉包含基于捕捉组件代码来修改匹配的应用程序的应用程序行为。所述方法还包含响应于匹配的窗口的关闭而完成与匹配的窗口关联的事件。

本说明书中描述的特征和优点并非完全详尽，且明确地说，所属领域的普通技

术人员将根据附图、说明书和权利要求书而了解许多附加特征和优点。此外，应注意，本说明书中使用的语言已大体上经选择用于易读和指导的目的，且不可经选择而描绘或限制发明主题。

附图说明

图 1 是说明本发明的一个实施例可运作的示范性环境的图。

图 2 是说明一种捕捉并处理与客户端装置关联的事件数据的方法的示范性实施例的流程图。

图 3 是说明一种捕捉客户端装置上的键击信息的方法的示范性实施例的流程图。

图 4 是说明一种捕捉客户端装置上的显示信息的方法的示范性实施例的流程图。

图 5 是基于简档的显示捕捉方法的一个实施例的流程图。

附图和以下描述仅以说明的方式涉及本发明的优选实施例。从以下论述中应注意，本文揭示的结构和方法的替代实施例将当然地视为可在不脱离所主张的本发明原理的情况下使用的可行的替代形式。

具体实施方式

现参看附图，在几个附图中所有相似标号表示相似元件，图 1 展示说明本发明的一个实施例可运作的示范性环境的图。虽然图 1 中展示的环境反映一个客户端侧搜索引擎结构实施例，但其它客户端侧或服务器侧实施例为可能的。系统 100 包含可通过网络 106（例如，因特网）与服务器装置 150 通信的多个客户端装置 102a 到 102n。在其它实施例中，可作为替代而使用例如内联网本地局域网（“LAN”）或类似网络的其它网络。此外，在其它实施例中，关于分布式网络环境中客户端或服务器而描述的功能可在没有服务器装置或网络的情况下在单一客户端装置内发生。这些技术以及类似调试的实施方案属于本发明范围内。

图 1 所示的客户端装置 102a 到 102n 每一者包含计算机可读媒质 108，例如存储器装置、存储媒质和类似物。在一个实施例中，客户端装置 102a 包含耦合到处理器 110 的随机访问存储器（RAM）108。处理器 110 执行存储在存储器 108 中的计算机可执行程序指令。这些处理器可包含微处理器、ASIC、状态机或其它处理器，

且可为许多合适的计算机处理器中的任一者，例如加州圣克拉拉市（Santa Clara, California）的英特尔公司、伊利诺斯州绍姆堡（Schaumburg, Illinois）的摩托罗拉公司的处理器。这些处理器包含存储指令的媒质 108（例如，计算机可读媒质）或可与媒质 108 通信，当处理器执行所述指令时，所述指令促使处理器执行本文描述的步骤。计算机可读媒质 108 的实施例包含（但不限于）能够为处理器（例如，客户端装置 102a 的处理器 110）提供计算机可读指令的电子、光学、磁性或其它存储或传送装置。合适的媒质的其它实例包含（但不限于）软盘、CD-ROM、DVD、磁盘、存储器芯片、ROM、RAM、ASIC、经配置的处理器、所有光学媒质、所有磁带或其它磁性媒质，或计算机处理器可从其中读取指令的任何其它媒质。并且，各种其它形式的计算机可读媒质可将指令传送或携带到计算机，包含路由器、专用或公共网络、或其它有线或无线的传送装置或信道。指令可包含任何合适的计算机编程语言的代码，包含（例如）C、C++、C#、Visual Basic、Java、Python、Perl 和 JavaScript。

客户端装置 102a 到 102n 可耦合到网络 106，或者可为独立机器。客户端装置 102a 到 102n 还可包含许多外部或内部装置，例如鼠标、CD-ROM、DVD、键盘、显示器装置，或其它输入或输出装置。客户端装置 102a 到 102n 的实例为个人计算机、数字助理、个人数字助理、蜂窝式电话、移动电话、智能电话、寻呼机、数字图形输入板、膝上型计算机、因特网设备和其它基于处理器的装置。在一个实施例中，客户端装置 102a 到 102n 可为在任何合适的操作系统（例如，Microsoft® Windows® 或 Linux）上运作且能够执行一个或一个以上客户端应用程序程序的任何类型的基于处理器的平台。例如，客户端装置 102a 可包含执行客户端应用程序程序（也称作客户端应用程序 120）的个人计算机。客户端应用程序 120 可包含在存储器 108 中，且可包含（例如）文字处理应用程序、电子数据表应用程序、电子邮件应用程序、即时通讯应用程序、演示应用程序、因特网浏览器应用程序、日历/管理器应用程序、视频播放应用程序、音频播放应用程序、图像显示应用程序、文件管理程序、操作系统命令解释程序（operating system shell）和能够由客户端装置执行的其它应用程序。客户端应用程序 120 还可包含与其它客户端或远程应用程序互动或访问其它客户端或远程应用程序的客户端侧应用程序（例如，与远程电子邮件服务器互动以访问电子邮件的在客户端装置 102a 上执行的网络浏览器）。

用户 112a 可通过客户端装置 102a 的各种输入和输出装置与各种客户端应用程序 120 和与客户端应用程序 120 关联的项目互动。项目包含（例如）文字处理器文档、电子数据表文档、演示文档、电子邮件、即时通讯消息、数据库条目、日历条

目、约会条目、任务管理器条目、源码文件，和各种格式的其他客户端应用程序程序内容、文件、消息、项、网页，例如 HTML、XML、XHTML、可移植文档格式（PDF）文件和媒质文件（例如图像文件、音频文件和视频文件），或任何其它文档或项，或文档或项的群组，或适于在客户端装置中存储访问的电子形式的信息。

用户 112a 与项目、客户端应用程序 120 和客户端装置 102a 的互动创建可被观察、记录、分析或以其他方式使用的事件数据。事件可为与项目、客户端应用程序 120 或客户端装置 102a 关联的任何可能的事件，例如在项目中输入文本、在显示器装置上显示项目、发送项目、接收项目、操作输入装置、打开项目、保存项目、打印项目、关闭项目、打开客户端应用程序、关闭客户端应用程序、空闲时间、处理器负载、磁盘访问、存储器使用、将客户端应用程序引到前台、改变应用程序的视觉显示细节（例如，重新规定尺寸或最小化）、与与客户端应用程序关联的窗口互动，和无论如何与项目、客户端应用程序或客户端装置关联的任何其它合适的事件。另外，当客户端装置 102a 与独立于用户 112a 的项目互动时（例如当接收电子邮件或执行已安排的任务时），可产生事件数据。

在一个实施例中，客户端装置 102a 的存储器 108 还可含有捕捉处理器 124、队列 126 和搜索引擎 122。客户端装置 102a 还可含有数据存储 140 或与数据存储 140 通信。捕捉处理器 124 可捕捉事件并将事件传递到队列 126。队列 126 可将已捕捉的事件传递到搜索引擎 122，或搜索引擎 122 可从队列 126 检索新事件。在一个实施例中，当新事件到达队列 126 时，队列 126 通知搜索引擎 122，且当搜索引擎 122 准备处理事件时，搜索引擎 122 从队列 126 检索事件。当搜索引擎接收事件时，其可经处理并可存储在数据存储 140 中。搜索引擎 122 可接收来自用户 112a 的显式查询（explicit query）或产生隐式查询（implicit query），且其可响应于所述查询而从数据存储 140 检索信息。在另一实施例中，队列位于搜索引擎 122 中。在又一实施例中，客户端装置 102a 不具有队列，且事件从捕捉处理器 124 直接传递到搜索引擎 122。根据其它实施例，使用信息交换协议来传输事件数据。信息交换协议可包含（例如）有助于数据交换的任何合适的规则或惯例，且可包含通信机制：例如，可扩展标识语言-远程程序呼叫协议（XML/RPC）、超文本传输协议（HTTP）、简单对象访问协议（SOAP）、共用存储器、插件、本地或远程程序呼叫，或任何其它合适的信息交换机制。

捕捉处理器 124 可通过识别并提取与事件关联的事件数据来捕捉事件。事件的实例包含发送或接收即时通讯消息、用户查看网页、保存文字处理文档、打印电子

数据表文档、输入文本以写作或编辑电子邮件、打开演示应用程序、关闭即时通讯应用程序、输入键击、移动鼠标、将鼠标停悬于超链接上、点击窗口中的按钮、从菜单中进行选择，或类似事件。捕捉处理器 124 捕捉到的涉及用户查看网页的事件的事件数据的实例包含网页的 URL、用户查看网页的时间和日期，和网页的内容。另外，在一个实施例中，可从远程位置捕捉包含与事件或项目有关的信息的附加事件数据。例如，可在网络 106 上从网络数据库服务器检索用户访问的电影或音乐文件的艺术家和其它相关信息，例如，专辑、导演、风格和类似信息。

在一个实施例中，捕捉处理器 124 可包含多个捕捉组件。例如，捕捉处理器 124 可包含用于每一客户端应用程序以便捕捉与每一应用程序相关的事件的分离的捕捉组件。捕捉处理器 124 还可包含用以监视并捕捉用户的键击输入的分离的捕捉组件，和可监视并捕捉显示在与客户端装置 102a 关联的显示器装置上的项（例如，文本）的分离的捕捉组件。键击捕捉组件和显示捕捉组件可在客户端应用程序捕捉组件不可用于客户端应用程序时用来捕捉事件，或与客户端应用程序捕捉组件结合。键击捕捉组件还可含有或访问键击数据库。键击数据库可提供键击与对应用程序的动作之间的相关。键击捕捉组件和显示捕捉组件还可含有或访问先前键盘或显示器事件的历史。

在一个实施例中，捕捉处理器 124 可包含监视整个网络活动以便捕捉与网络活动（例如，接收到即时通讯消息）关联的事件数据的分离的捕捉组件。捕捉处理器 124 可包含监视例如处理器负载、空闲时间、磁盘访问、使用中的客户端应用程序和可用的存储器数目的整个客户端装置性能数据的分离的捕捉组件。个别捕捉组件可监视多个客户端应用程序，且多个捕捉组件可监视单一客户端应用程序的不同方面。

在一个实施例中，捕捉处理器 124 可通过个别捕捉组件来监视客户端装置上的活动，且可通过一般化事件定义和注册机制（例如，事件计划）来捕捉事件。每一捕捉组件可定义其自身的事件计划，或可使用预定义的事件计划。事件计划可视客户端应用程序或捕捉组件正监视的活动而定而不同。通常，事件计划可（例如）通过提供与事件关联的事件数据（例如，事件的时间）的字段和与任何相关项目（例如，标题）以及任何相关项目的内容（例如，文档主体）有关的字段来描述事件的格式。事件计划可描述与事件有关的任何合适的事件数据的格式。例如，用户输入（例如，键入或显示到显示器装置的文字）的事件计划可包含用于输入的应用程序、文本的格式、文字和其它项输入，以及时间输入。用户接收的电子邮件事件的事件

计划可包含标题信息，例如主题内容、收件人、发件人、抄送和接收到的时间字段，以及主体信息。用户当前正查看网页的事件计划可包含网页的统一资源定位器（URL）、查看的时间和网页内容。用户保存文字处理文档的事件计划可包含文档标题、保存时间、文档位置、文档格式、文档文本，和文档位置的指针。

在一个实施例中，键击捕捉组件可能不使用事件计划。事实上，在此实施例中，键击捕捉组件可将键击转化成文字序列，并接着可将文字序列传递到另一组件（例如，文字处理捕捉组件）。文字处理捕捉组件可使用事件计划来表达键击捕捉组件所发送的信息。

更一般地，事件计划可描述大约在事件时间的系统状态。例如，事件计划可含有与用户从其中进行导航的先前网页关联的网页事件的 URL。另外，事件计划可描述具有如同列表的更复杂结构的字段。例如，事件计划可含有罗列多个收件人的字段。事件计划还可含有任选字段，使得应用程序可视需要而包含附加事件数据。

捕捉处理器 124 可捕捉目前发生的事件（“实时事件”），且可捕捉过去发生的事件（“历史事件”）。实时事件可为“可索引”或“不可索引”的。在一个实施例中，搜索引擎 122 将可编入索引的实时事件编入索引，但不将不可编入索引的实时事件编入索引。搜索引擎 122 可基于事件的重要性来确定是否将事件编入索引。可编入索引的实时事件可为与项目关联的更重要的事件，例如查看网页、加载或保存文件，和接收或发送即时消息（“IM”）或电子邮件（“email”）。不可编入索引的事件可能被搜索引擎 122 认为不够重要因而无需编入索引并存储事件，例如移动鼠标或选择项目中文本的一部分。不可编入索引的事件可由搜索引擎 122 用来更新当前用户状态。虽然所有的实时事件可与用户当前所从事的事情（或当前用户状态）有关，但可将可编入索引的实时事件编入索引并存储在数据存储 140 中。

或者，在一个实施例中，搜索引擎 122 可将所有的实时事件编入索引。实时事件可包含（例如）发送或接收项目（例如，即时通讯消息）、检查项目的一部分（例如，选择文本的一部分或在网页的一部分上移动鼠标）、改变项目（例如，在电子邮件中键入文字或在文字处理文档中粘贴句子）、关闭项目（例如，关闭即时通讯窗口或改变正查看的电子邮件消息）、加载、保存、打开或查看项目（例如，文字处理文档、网页或电子邮件，收听或保存 MP3 文件或其它音频/视频文件），或更新项目的元数据（例如，为网页设定书签、打印演示文档、删除文字处理文档或移动电子数据表文档）。

历史事件与可编入索引的实时事件相似，不同之处在于事件在安装搜索引擎

122 之前发生或者未被捕捉到，因为（例如）当客户端装置 102a 运作时的一段时间内搜索引擎 122 不运作，或因为事件发生时不存在用于特定类型历史事件的捕捉组件。历史事件的实例包含用户的已保存的文字处理文档、媒质文件、演示文档、日历条目和电子数据表文档，用户收件箱中的电子邮件，和用户设为书签的网页。捕捉处理器 124 可通过周期性地爬行（crawl）存储器 108 和任何相关的数据存储装置以便查找先前未被捕捉处理器 124 捕捉的事件来捕捉历史事件。捕捉处理器 124 还可通过请求某些客户端应用程序（例如，网络浏览器或电子邮件应用程序）以检索项目和其它相关信息来捕捉历史事件。例如，捕捉处理器 124 可请求网络浏览器应用程序获得用户查看的所有网页，或请求电子邮件应用程序获得与用户关联的所有电子邮件消息。这些项目可能当前并不存在于存储器 108 中或存在于客户端装置 102a 的存储装置上。例如，电子邮件应用程序可能必须从服务器装置检索电子邮件。在一个实施例中，搜索引擎 122 将历史事件编入索引。

在图 1 所示的实施例中，将捕捉处理器 124 捕捉的事件以事件计划所描述的格式发送到队列 126。捕捉处理器 124 还可将性能数据发送到队列 126。性能数据的实例包含当前处理器负载、一段预定时间内的平均处理器负载、空闲时间、磁盘访问、使用中的客户端应用程序，和可用的存储器数目。性能数据也可由特定的性能监视组件提供，其中一些特定的性能监视组件可能是（例如）搜索引擎 122 的一部分。队列 126 中的性能数据可由搜索引擎 122 和捕捉处理器 124 的捕捉组件检索。例如，捕捉组件可检索性能数据以改变发送到队列 126 的事件数目，或发送的事件的详细程度（当系统繁忙时事件较少或较小），或事件的发送频率（当系统繁忙时或有太多的事件等待处理时，不经常发送事件）。搜索引擎 122 可使用性能数据来确定其何时将各种事件编入索引，或其何时和每隔多久发布隐式查询。

在一个实施例中，队列 126 保留事件直到搜索引擎 122 准备好处理事件为止。或者，队列 126 使用性能数据来帮助确定多快将事件提供给搜索引擎 122。队列 126 可包含一个或一个以上分离的队列，例如用户状态队列和索引队列。在一个实施例中，索引队列可将可编入索引的事件排成队列。或者，队列 126 可具有附加队列或包含单一队列。可使用存储器映射文件将队列 126 实施为循环优先队列。队列可为多优先权队列，其中在较低优先权事件之前先服务较高优先权事件，且其它组件可能指定其所关注的事件类型。通常，可给予实时事件比历史事件更高的优先权，且可给予可编入索引的事件比不可编入索引的实时事件更高的优先权。队列 126 可能有其它实施方案。在另一实施例中，客户端装置 102a 不具有队列 126。在此实施

例中，事件直接从捕捉处理器 124 传递到搜索引擎 122。在其它实施例中，可使用合适的信息交换机制在捕捉组件与搜索引擎之间传输事件，所述合适的信息交换机制例如：可扩展标识语言-远程程序呼叫协议(XML/RPC)、超文本传输协议(HTTP)、简单对象访问协议(SOAP)、共用存储器、插件、本地或远程程序呼叫，或任何其它合适的信息交换机制。

再次参看图 1，在一个实施例中，搜索引擎 122 可含有索引器 130、查询系统 132 和格式器 134。查询系统 132 可从队列 126 检索所有的实时事件和性能数据。查询系统 132 可使用性能数据和实时事件来更新当前用户状态并产生隐式查询。隐式查询可为基于当前用户状态的自动产生的查询。查询系统 132 还可接收并处理来自用户 112a 的显式查询。性能数据还可由搜索引擎 122 从队列 126 检索以便用于确定搜索引擎 122 可能进行的活动的数目。

在图1所示的实施例中，索引器 130 从队列 126 检索可编入索引的实时事件和历史事件（可编入索引的事件）。或者，队列 126 可将可编入索引的事件发送到索引器 130。索引器 130 可将可编入索引的事件编入索引，且可将它们发送到数据存储 140，在数据存储 140 中存储可编入索引的事件。数据存储 140 可为任何类型的计算机可读媒质，且可与客户端装置 102a 集成（例如，硬盘驱动器），或在客户端装置 102a 外部（例如，外部硬盘驱动器）或在通过网络 106 访问的另一数据存储装置上。数据存储可为一个或一个以上逻辑或物理存储区域。在一个实施例中，数据存储 140 可在存储器 108 中。数据存储 140 可有助于一种存储数据的方法或方法的组合，其中包含（但不限于）阵列、哈希表、列表和配对（pair），且可包含压缩和编码。在图 1 所示的实施例中，数据存储包含索引 142、数据库 144 和存储库 146。

在图 1 所示的实施例中，当索引器 130 接收事件时，索引器 130 可从事件计划中确定与事件关联的术语（如果存在的话）、事件的时间（如果可用的话）、与事件关联的图像（如果存在的话），和/或定义事件的其它信息。索引器 130 还可确定事件是否与其它事件有关，并使事件与相关事件关联。例如，对于与网页有关的事件，索引器 130 可使此事件与与相同网页有关的其它事件关联。此关联信息可以每一相关事件群组的文档的形式存储在数据库 133 中。

索引器 130 可发送与事件关联的术语和时间并将其并入在数据存储 140 的索引 142 中。可将事件发送到数据库 144 以便进行存储，且相关项目的内容和任何相关图像可存储在存储库 146 中。与即时通讯消息关联的通话对象可存储在数据库 144 中。

在图 1 所示的实施例中，用户 112a 可将显式查询输入到客户端装置 102a 上的搜索引擎界面中，搜索引擎 122 接收显式查询。在一个实施例中，客户端装置 102a 上的搜索引擎界面包含图形/文本显示，例如基于 HTML 的网页。在替代实施例中，搜索引擎界面包含各种输入/输出模式，例如声音激活系统、触控式系统 (touch based system) 或类似模式。

搜索引擎 122 还可基于当前用户状态产生隐式查询，查询系统 132 可从实时事件中确定当前用户状态。基于查询，查询系统 132 可在数据存储 140 中定位相关信息，并提供结果集合。在一个实施例中，结果集合包含与客户端应用程序 120 关联的项目或客户端项目的项目识别符。客户端项目包含与用户 112a 或客户端装置 102a 关联的项目，例如用户的电子邮件、文字处理文档、即时通讯消息、先前查看的网页，和与客户端装置 102a 或用户 112a 关联的任何其它项目或项目的一部分。项目识别符可为 (例如) 统一资源定位器 (URL)、文件名、链接、图标、本地文件的路径、任何字数识别符，或可识别项目的任何合适的信息。在另一实施例中，结果集合还包含定位在网络 106 上的其它系统上的项目 (例如，由搜索引擎定位在服务器装置上的网络项目) 的项目识别符。网络项目包含定位在网络 106 上的系统上的用户 112a 先前未查看或以其他方式参考的项目，例如用户 112a 先前未查看的网页。

格式器 134 可从搜索引擎 122 的查询系统 132 接收搜索结果集合，且可格式化所述结果以供输出到显示处理器 128。在一个实施例中，格式器 134 可将结果格式化为 XML、HTML 或用制表符描绘的文本。显示处理器 128 可包含在存储器 108 中，且可控制结果集合在与客户端装置 102a 关联的显示器装置上的显示。显示处理器 128 可包含各种组件。例如，在一个实施例中，显示处理器 128 包含超文本传输协议 (HTTP) 服务器，其接收对于信息的请求，并通过建立和传送超文本标识语言 (HTML) 页来进行响应。在一个此实施例中，HTTP 服务器包含 Apache 网络服务器的缩小版。显示处理器 128 可与一组 API 关联以便允许各种应用程序接收结果并以各种格式显示所述结果。显示 API 可以各种方式实施，包含 (例如) 作为 DLL 出口、COM 界面、VB、JAVA 或 .NET 库，或作为网络服务。

用户 112a 到 112n 可通过客户端装置 102a 到 102n 在网络 106 上彼此通信，且与耦合到网络 106 的其它系统和装置通信。如图 1 所示，服务器装置 150 可耦合到网络 106。在图 1 所示的实施例中，搜索引擎 122 可将包含在显式或隐式查询中或两者中的搜索查询传送到服务器装置 150。用户 112a 还可在搜索引擎界面中输入搜索查询，客户端装置 102a 可通过网络 106 将搜索查询传送到服务器装置 150。在另

一实施例中，可改为将查询信号发送到代理服务器（未图示），代理服务器接着将查询信号传送到服务器装置 150。也可能有其它配置。

服务器装置 150 可包含执行搜索引擎应用程序（例如，Google™搜索引擎）的服务器。在其它实施例中，服务器装置 150 可包含相关信息服务器或广告服务器。与客户端装置 102a 到 102n 相似，服务器装置 150 可包含耦合到计算机可读存储器 162 的处理器 160。描绘为单一计算机系统的服务器装置 150 可实施为计算机处理器的网络。服务器装置 150 的实例包含服务器、大型计算机、连网的计算机、基于处理器的装置，和类似类型的系统和装置。服务器处理器 160 可为许多计算机处理器中的任一者，例如加州圣克拉拉市的英特尔公司和伊利诺斯州绍姆堡的摩托罗拉公司的处理器。在另一实施例中，服务器装置 150 可存在于客户端装置上。在又一实施例中，可存在多个服务器装置 150。

存储器 162 含有搜索引擎应用程序，也称为网络搜索引擎 170。搜索引擎 170 可响应于来自客户端装置 102a 的搜索查询而定位来自网络 106 的相关信息。搜索引擎 170 接着可将结果集合通过网络 106 提供给客户端装置 102a。

结果集合可包含一个或一个以上项目识别符。项目识别符可为（例如）统一资源定位器（URL）、文件名、链接、图标、本地文件的路径，或识别项目的任何其它信息。在一个实施例中，项目识别符可包含与项目关联的 URL。

在一个实施例中，服务器装置 150 或相关装置先前已执行网络 106 的爬行以便定位存储在耦合到网络 106 的其它装置或系统处的项目（例如网页），并已在存储器 162 中或另一数据存储装置上编入索引项目。

应注意，本发明的其它实施例可包含具有与图 1 所示的结构不同结构的系统。例如，在本发明的一些其它实施例中，客户端装置 102a 为独立装置且不耦合到网络。

现参看图 2，其展示说明一种捕捉并处理与客户端装置关联的事件数据的方法的示范性实施例的流程图。应注意，可执行根据本发明实施例的各种事件数据捕捉和处理方法。例如，在一个实施例中，基于用于与应用程序互动的与应用程序关联的应用程序界面（“API”）来捕捉应用数据。在另一实施例中，捕捉与应用程序关联的键击，且尤其基于与键击关联的动作来确定事件。在另一实施例中，捕捉显示元素和相关信息，且至少部分地基于所捕捉的显示信息来确定事件。这些是根据本发明实施例的多种事件数据捕捉和处理方法的一些非详尽的实例。

在一个实施例中，可确定是否索引事件，且如果如此确定，那么可将事件编入索引并进行存储。

图 2 说明示范性方法 200，其提供一种捕捉并处理实时事件的方法。此示范性方法是以举例的方式提供，从对示范性实施例的前述描述中应了解，存在执行本发明其它实施例的方法的多种方式。图 2 所示的方法 200 可由各种系统中的任一者实施或以其他方式执行。下文以举例的方式描述方法 200 由图 1 所示的系统 100 执行，且在解释图 2 的实例方法的过程中参考系统 100 的各种元件。

在 202 中，捕捉处理器 124 捕捉事件。捕捉处理器 124 可通过在事件发生时识别并编辑与事件关联的事件数据来捕捉事件。捕捉处理器 124 可具有用于每一客户端应用程序的分离的捕捉组件、网络监视、性能数据捕捉、键击捕捉和显示捕捉。例如，事件可在用户 112a 在客户端应用程序中键入预定数目的文字时发生。与此事件关联的事件数据可为（例如）用于输入的应用程序、文本格式、文字输入和时间输入。在一个实施例中，捕捉组件可使用例如先前已向客户端装置 102a 注册的事件计划的一般化事件定义机制来捕捉或表达事件。

图 3 提供捕捉步骤 202 的实时事件的实例。明确地说，图 3 展示说明一种捕捉客户端装置上的键击信息的方法的示范性实施例的流程图。在此实施例中，接收与应用程序关联的复数个键击，处理每一键击来确定形成复数个相关动作的相关动作，且至少部分地基于复数个相关动作来确定事件。可首先确定处于焦点中的应用程序（application in focus）。

在一个实施例中，事件可为许多文字，且可确定复数个相关动作可形成文字。可至少部分地通过接收指示空格或标点符号的至少一个键击来确定文字。在另一实施例中，事件可为许多字符，且可确定复数个相关动作可形成字符。

在一个实施例中，在处理每一键击之后，可更新捕捉状态。在一个实施例中，可至少部分地基于事件来更新当前用户状态。可将事件编入索引并进行存储。

在一个实施例中，可至少部分地通过将键击与键击表中的条目匹配并确定键击表中的与条目关联的动作来确定每一相关动作。动作可包含在文字中添加字符、从文字中删除字符、插入字符、重写字符、删除文字、删除段落、选择项，和重新定位光标中的一者。键击表可与应用程序关联，或可为通用键击表。

在另一实施例中，接收与应用程序关联的键击，基于用户输入来确定事件，且确定是否将事件编入索引。用户输入可为从键击确定的许多文字、从键击确定的许多字符中的一或多者，且变化是焦点从所述应用程序到另一应用程序。确定是否将事件编入索引可包含确定事件对于用户是否重要。

在 302 中，提供键击数据库。键击数据库可含有用于各种客户端应用程序的键

击命令和每一命令的个别动作的表。在一个实施例中，键击数据库对于每一分离的客户端应用程序具有一个分离的表。如果不存在用于特殊应用程序的特定的表，那么可将通用表用于应用程序。在一个实施例中，可将相同的表用于应用程序的群组。例如，可将相同的表用于来自相同开发商的应用程序。例如，可将相同的表用于所有 Microsoft®应用程序，且可将不同的表用于来自 Lotus®的所有应用程序。另外，可将相同的表用于相关应用程序，例如来自 Microsoft®的 Office®应用程序或来自任何开发商的文字处理应用程序。可为每一应用程序手动地创建键击表。或者，可通过自动地确定每一应用程序的每一键击的个别动作来自动地创建键击表。

在 303 中，呼叫键击捕捉组件。键击捕捉组件可有规律乃至恒定地运作，使得其监视所有键击活动，或可在需要时或在不同时间由捕捉处理器 124 其它捕捉组件选择性地呼叫。在 304 中，键击捕捉组件可确定处于焦点中的客户端应用程序。处于焦点中的应用程序是当前正由用户 112a 使用的客户端应用程序。例如，其可为用户 112a 正在其中输入文本的客户端应用程序。在一个实施例中，确定处于焦点中的应用程序，使得键击捕捉组件可使用来自键击数据库的与处于焦点中的应用程序关联的键击表。例如，如果处于焦点中的应用程序是特定文字处理应用程序，那么键击捕捉组件可调用相应的特定文字处理应用程序键击表。

在 306 中，键击捕捉组件重置事件数据。在一个实施例中，以事件计划定义的格式在事件中编辑事件数据，且键击捕捉组件可注册特定的键击捕捉事件计划，且重置事件包含清除任何现有的事件数据。也可重置计数器 N。在一个实施例中，计数器计算键击捕捉组件捕捉的文字数目。在此实施例中，当事件被重置时，计数器被重置为零。

在 308 中，键击捕捉组件捕捉用户 112a 输入的键击。用户可通过多种输入装置（例如键盘）来输入键击。键击可包含单键输入或多键输入，且可包含文本输入和编辑命令输入。例如，键击可包含“G”键或“上档（Shift）”和“G”键。键击还可包含“回格（Backspace）”键或“控制（Control）”和“回格”键。键击捕捉组件可通过可由操作系统调用的应用程序回叫接收键击消息来接收键击。键击捕捉组件还可通过监视操作系统层级的输入缓冲器（例如，键盘缓冲器）来接收键击。可捕捉其它输入字符的方法（例如，手写）。

在 310 中，键击捕捉组件处理键击。在所展示的实施例中，键击捕捉组件通过将键击与处于焦点中的特定应用程序的键击表中的条目进行匹配并确定键击的个别动作来处理接收到的键击。例如，如果在特定文字处理应用程序中接收到的键击

是“上档”键和“G”键，那么键击捕捉组件可在特定应用程序的键击表中匹配“上档”键和“G”键，并确定相关动作（例如，输出字母“G”）。如果接收到的键击是“控制”和“回格”键，那么键击捕捉组件可将这些键与特定应用程序的键击表中的个别动作进行匹配，并确定相关动作（例如，删除光标前的文字的动作）。其它可能的动作包含（例如）将光标上移或下移一行、将光标上移或下移一段，和将光标移动到文档的开始或结尾处。键击捕捉组件可跟踪光标相对于先前输入的字符的当前位置。例如，如果用户键入 5 个字符，敲击“左箭头”三次，键入一个字符，并敲击“右箭头”三次，那么组件可识别用户后退并在文字中插入字符（对于“左箭头”对应于将光标后移一个字符的应用程序）。组件可进一步跟踪每一应用程序是否处于“插入”模式，使得其知晓新的字符是否会覆盖任何现有的字符。

在另一实施例中，键击捕捉组件可能不会直接处理键击。作为替代，键击捕捉组件可首先将键击传递给操作系统（仿佛用户已主动将其发送给键击捕捉组件）而不是应用程序。以此方式，操作系统可进行将键击转译成更高层级消息（字符、键移动等）的工作，并接着可将这些消息传回键击捕捉组件以供进一步处理。出于若干原因，将键击发送给操作系统以便进行中间处理而不是直接对其进行处理可能是有利的。例如，其可避免处理多键击字符（包含非英文字符条目）的一些困难，且其可确保以与操作系统行为一致的方式处理键击。

在 312 中，键击捕捉组件将经处理的键击添加到事件数据。在所展示的实施例中，键击捕捉组件捕捉事件计划所描述的事件数据，并将经处理的键击添加到事件数据。例如，如果经处理的键击指示添加字母 G，那么将字母 G 添加到事件数据。如果经处理的键击指示删除光标前的文字且文字包含在事件数据中，那么可删除所述文字。另外，键击捕捉组件可将经处理的键击添加到捕捉状态。键击捕捉组件可在确定用户最近已输入了什么文字时维持含有先前键击的捕捉状态。可在产生事件之前更新捕捉状态。

在 314 中，键击捕捉组件确定是否已将完整的文字添加到事件。文字可以是代表口语文字、缩写词、术语或语义单位的一连串字符。例如“WORLD SERIES”、“ASCII”、“MR”和“GOOGLE”可以是文字。键击捕捉组件可确定是否已通过多种方法输入完整的文字。在一个实施例中，键击捕捉组件查找后面有空格的一连串字符（例如“GOOGLE ”）或后面有某些标点字符的一连串字符（例如，“GOOGLE,”），以便确定是否已输入完整的文字。如果未输入完整的文字，那么键击捕捉组件前进到步骤 316。在 316 中，键击捕捉组件可确定先前处于焦点中的应用程序是否仍处

于焦点中，或处于焦点中的应用程序是否已改变且另一应用程序现处于焦点中。如果处于焦点中的应用程序未改变，那么键击捕捉组件返回到步骤 308 并接收另一键击。如果处于焦点中的应用程序已改变，那么键击捕捉组件返回到步骤 304 并确定当前处于焦点中的应用程序。如果键击捕捉组件识别例如将光标上移一行的动作，那么其可更新捕捉状态以反映新的光标位置。如果键击捕捉组件不具有所述位置的捕捉状态，或无法正确地精确定位新位置，那么其可简单地重置任何部分捕捉的文字。作为另一实例，键击“左移箭头”可在特定应用程序中选择先前字符。键击捕捉组件可识别出此动作和例如删除选择或将选择粘贴在文档中其它地方的可能的后续动作。

如果在步骤 314 中键击捕捉组件确定已接收到完整的文字，那么在步骤 318 中计数器 N 加一。在步骤 320 中，捕捉处理器确定计数器 N 是否等于整数 T。整数 T 可为（例如）文字数目。文字数目可为预定的，或可基于多种合适的因素（例如处于焦点中的当前应用程序或其它用户活动）来确定。在一个实例中，T 等于一，使得键击捕捉组件可在识别用户 112a 输入的每一文字时创建事件。或者，T 可为字符数目。与文字数目一样，字符数目可为预定的，或可基于多种合适的因素来确定。如果在步骤 320 中发现 N 不等于 T，那么键击捕捉组件转到步骤 316，在步骤 316 中键击捕捉组件确定处于焦点中的应用程序是否已改变。如果键击捕捉组件确定 N 确实等于 T，那么键击捕捉组件前进到步骤 322。例如，如果 N 是计算所捕捉的文字的数目的计数器且 T 设置为一并接收到“GOOGLE”，那么键击捕捉组件前进到步骤 322。

在 322 中，键击捕捉组件编辑事件。在所展示的实施例中，键击捕捉组件通过编辑与事件计划描述的特定字段关联的事件数据来编辑事件。例如，事件数据可包含用于输入的应用程序、文本的格式、文字输入和时间输入。当键击捕捉组件已编辑了事件时，方法 200 可在图 2 所示的步骤 204 处继续。

图 4 提供基于显示信息来捕捉事件（例如，如图 2 所示的实时事件 202）的另一实例。图 4 展示说明一种捕捉客户端装置上的显示信息的方法的示范性实施例的流程图。通常，客户端装置中的应用程序会调用显示呼叫来在用户显示器（例如，监视器、液晶显示器、等离子显示器或类似物）上显示与应用程序关联的信息。在一个实施例中，捕捉处理器 124 包含用于至少部分地基于与应用程序关联的显示或窗口来确定事件的显示捕捉组件。应注意，与应用程序关联的窗口可被隐藏或显示且在任何给定时间对于用户为可见的。可至少部分地通过使用显示器当前状态的阵

列并用显示呼叫更新阵列来确定显示器上的信息。还可（至少部分地）通过至少部分地基于显示呼叫的显示位置建立显示项来确定显示器上的信息。可通过使用显示呼叫分析输出到显示器的若干项的 x , y 坐标、长度或相对位置中的一或多者来处理显示呼叫。

在一个实施例中，首先确定处于焦点中的应用程序，从中捕捉相关显示信息以产生事件。事件可包含（例如）确定将要在显示器上输出的许多文字。在一个实施例中，可在处理每一显示呼叫之后更新捕捉状态，且可至少部分地基于事件来更新当前用户状态。

参看图 4，最初呼叫显示捕捉组件 400。显示捕捉组件可有规律地（例如，按进度表）乃至恒定地运作，使得其监视所有显示活动，或可在需要时被选择性地呼叫。接着，确定处于焦点中的应用程序 402。在一个实施例中，处于焦点中的应用程序是当前正使用的客户端应用程序。例如，在基于窗口的操作系统环境中，显示捕捉组件确定用户现在正使用的窗口。

对于第一或新的呼叫，重置事件 404。可能由于多种原因而重置事件，例如，因为用户已切换处于焦点中的应用程序，事件已被捕捉并发送以进行处理，或由于其它多种原因。在一个实施例中，重置 404 事件包含指示或产生新事件。重置 404 之前的现有事件数据被以捕捉状态进行保存并维持，直到重置为止。新事件的指示触发发送先前事件中的现有事件数据以进行处理。例如，在一个实施例中，重置功能保存先前事件并在初始化新事件之前将其发送到队列 126。在一个实施例中，可根据与事件类型关联的事件计划来产生新事件。事件计划可包含与不同类型的事件关联的各种数据字段的格式化。

接着，接收显示呼叫 406。例如，在一个实施例中，显示捕捉组件接收到显示呼叫。显示呼叫可为操作系统发送到与客户端装置关联的显示器装置的指令，其指示显示器装置显示项或改变项在显示器装置上的显示。例如，显示呼叫可以是以字体“Times New Roman”在显示位置（例如，位置 x , y ）处显示文本“Google”。此显示呼叫可（例如）促使在计算机监视器的屏幕上在活动窗口中位置 x , y 处且以字体“Times New Roman”显示文本“Google”。显示呼叫还可以是（例如）删除给定位置处的项，改变文字字体、改变项在显示器上的位置，或类似操作。在一个实施例中，还可通过将代码注入到应用程序中或通过如下文进一步详细描述使用动态链接库（“DLL”）挂钩函数并截取操作系统 API 呼叫来确定显示呼叫。在另一实施例中，还可捕捉其它输出方法，例如对于打印机驱动器、音频驱动器或类似物的呼叫。

在接收 406 显示呼叫之后，处理 408 显示呼叫来确定与当前显示关联的事件数据。显示包含用户在窗口中看到的视觉信息。在一个实施例中，捕捉与当前显示关联的事件数据并以根据与特定事件类型或应用程序关联的事件计划的事件格式放置。在一个实施例中，可（例如）通过保留已输出到当前显示的项的阵列，来保存所捕捉的事件数据并将其维持在捕捉状态中。例如，在一个实施例中，显示捕捉组件可使用项的 x 和 y 坐标来确定文字、文字之间的空格和新的行的开始。显示捕捉组件可使用显示呼叫来更新当前显示的阵列。例如，如果显示呼叫是以字体“Times New Roman”在先前显示文字“Google,”的位置 x, y 处写入文本“Google”，那么将字体为“Times New Roman”的文本“Google”添加到位置 x, y 处的显示，从而取代文本“Google”。接着可将包含文本“Google”的与当前显示关联的已更新阵列添加到事件。也可通过用已更新阵列“Google”取代先前状态阵列“Google,”来更新捕捉状态。因此，在事件不完整的时候，对于显示的更新可在相关事件中得到反映。

接着确定 410 是否已发生完整的事件。在一个实施例中，为了确定 410 事件是否完整，检查完整条件。例如，完整条件可包含将单一文字或预定数目的文字添加到显示器装置上的显示。或者，作为另一实例，完整条件可包含显示器上的特定用户动作，例如超链接、按钮、菜单项或类似物上的鼠标点击。如果尚未发生完整事件，那么确定 412 处于焦点中的应用程序是否已改变。如果处于焦点中的应用程序未改变，那么接收 406 与活动窗口关联的下一显示呼叫。相反，如果处于焦点中的应用程序已改变，那么再次确定 402 处于焦点中的新应用程序，且开始新事件的捕捉过程。

如果确定 410 已发生完整事件，那么编辑 414 完成的事件。例如，在一个实施例中，显示捕捉组件通过根据事件计划提供所有的与事件相关数据来编辑事件。事件数据可包含所使用的应用程序、显示的格式、显示的时间、所显示的内容、所显示的任何文字的字体和字体尺寸，和随后可用于索引或对用户而言有用的与事件关联的任何其它数据。

现参看图 5，展示基于简档的显示捕捉方法 500 的一个实施例的流程图。在此实施例中，显示捕捉组件可经配置以捕捉与选定的应用程序的窗口关联的信息。应用程序窗口可显示在用户的视野内或隐藏在计算机系统内。显示捕捉组件可为之进行配置的选定的应用程序的一些实例包含不提供公共 API 的应用程序、加密或以其他方式保护数据以防从其应用程序窗口捕捉数据的应用程序，或类似应用程序。此实施例中的捕捉处理器可包含若干捕捉组件。对于从其中捕捉事件数据的一些目标

应用程序，使用基于简档的捕捉组件。

因此，在一个实施例中，在捕捉组件处理初始化时，首先检查 501 具有相关窗口的先前存在的应用程序以便确定其身份。在一个实施例中，使用目标应用程序简档来确定运行的应用程序的身份。应用程序简档包含与目标应用程序关联的识别信息。应用程序简档可包含（例如）与目标应用程序窗口（隐藏或显示）关联的唯一窗口类别名、与目标应用程序关联的存储的二进制文件的唯一系统路径名、与目标应用程序关联的操作系统注册表，或类似信息。在一个实施例中，初始化时且在检查 501 现有的应用程序之后，（例如）通过经由操作系统预订窗口创建事件来接收后续窗口创建事件的通知。对于每一窗口创建事件，检查 501 发起的应用程序以确定其是否为选定的目标应用程序。

一旦识别目标应用程序，将应用程序特定的显示捕捉组件的计算机代码与目标应用程序关联 502。例如，在一个实施例中，设置基于 Windows® API 的挂钩函数将显示捕捉代码从动态链接库（“DLL”）加载到目标应用程序处理中。在一个实施例中，显示捕捉代码通过修改、截取或绕过其与操作系统的互动来规避目标应用程序的加密/数据捕捉保护特征。

在一个实施例中，从与目标应用程序关联的窗口的子集中获得目标应用程序的事件数据。因此，将与目标应用程序关联的窗口与存储的窗口简档进行比较 504。窗口简档含有可用来识别目标窗口的信息，包含（例如）将要包含在窗口标题中的文本、显示元素（例如，按钮、下拉菜单、文本输入框和类似物）的数目和类型，将要显示在显示元素上的文本、显示元素的位置和排列，和类似信息。

例如，在一个实施例中，即时通讯（“IM”）目标应用程序的文本窗口是目标窗口。即时消息文本（输入的和接收到的）是将被捕捉且与 IM 事件关联的事件数据（即，IM 事件中的一个数据字段包含用户在一次对话中交换的文本）。此实例中的窗口简档（即，作为 IM 通话窗口的目标窗口的窗口简档）包含在窗口标题中“聊天”的字符串、“发送”按钮和较大的自由文本输入框。

对于目标应用程序中的每一目标窗口，监视 506 显示元素或其它目标窗口显示相关变量的状态。例如，监视与显示按钮或等效的快捷键（例如，“发送”按钮或 Ctrl + Enter 键序列）关联的处理程序（handler）的状态。当检测到所关注的显示元素（例如，按钮）的状态变化时，捕捉 508 目标窗口中的事件数据。

在一个实施例中，由挂钩函数到目标应用程序处理中的显示捕捉组件代码来启用数据捕捉。此代码在目标应用程序与操作系统之间截取操作系统消息，并修改对

于这些操作系统消息的回应以便启用数据捕捉。例如，在基于 Windows®的系统中，经配置以响应于每一“获取文本”命令（例如，WM_GETTEXT）而提供空字符串的应用程序由捕捉组件代码修改，以便使用默认的处理程序来提供窗口显示文本。因此，内嵌于目标处理中的捕捉组件代码允许修改应用程序行为（例如）以便绕过来自所述处理内的数据捕捉预防措施。

可基于状态变化而重复事件数据显示捕捉 508，且可用新捕捉的事件数据来更新与事件关联的捕捉状态。例如，对于 IM 应用程序中“发送”按钮上的每次用户点击，可捕捉 508 与即时消息聊天对话关联的文本并以与当前对话的 IM 事件关联的数据结构将其存储在存储器中。“发送”按钮上的后续点击可促使捕捉用户之间的 IM 对话中附加的文本交换。此基于状态的数据捕捉有益地提供一种在用户与目标窗口互动时捕捉显示数据的有效方案。基于时间的轮询方案可能不太有效，因为即使用户不向目标程序提供任何新的输入，其也可运作。此外，基于状态的数据捕捉可减少遗漏一些输入的数据的可能性，因为通常在用户输入之后，需要某一状态变化（例如，点击“发送”按钮、“OK”按钮、“继续”按钮，或键入等效的快捷键或类似状态变化）。

再次参看图 5，监视 510 目标窗口终止的状态。当目标窗口终止时，可完成 512 事件。可完成到达事件数据存储中的任何事件数据条目，且可将捕捉的事件发送到索引器。当目标窗口保持活动的同时，可为了窗口中的每一状态变化而继续进行显示捕捉。

现在再次参看图 2，一旦捕捉到事件（例如，如参看图 4 和图 5 所描述），事件数据处理方法继续步骤 204。除了其它捕捉组件外，参看图 4 而描述的键击捕捉组件和参看图 5 而描述的显示捕捉组件可一起用在捕捉处理器 124 中以便确定用户活动（例如，用户正在视觉输出装置（例如，显示器）上查看的内容）。其它捕捉组件可包含对于可包含在索引中的音频和视频事件的音频和视频数据捕捉技术，例如声音识别、图像处理 and 类似技术。

捕捉事件之后，确定 204 事件是否为可编入索引的事件。如上文所解释，一些实时事件可能不被编入索引（不可编入索引的实时事件）。在一个实施例中，不可编入索引的实时事件可用来更新当前用户状态，且可为（例如）检查项目的一部分、改变项目和关闭项目。在此实施例中，索引器 130 不将不可编入索引的事件编入索引或发送以进行存储。可编入索引的事件可为可编入索引的实时事件或历史事件。键击和显示呼叫可为不可编入索引的事件，例如用户输入许多文字或在显示器装置

上显示许多文字。键击和显示呼叫也可为可编入索引的事件，例如当整个项目或项目的一部分由用户输入或显示在显示器装置上时。

如果确定 204 可编入索引的事件，那么可将事件发送 206 到（例如）队列 126 以供编入索引，同时指示其为可编入索引的事件。在图 1 所示的实施例中，将可编入索引的实时事件发送到队列 126 内的用户状态队列和索引队列两者，且将历史事件发送到队列 126 内的索引队列。或者，可不将可编入索引的实时事件发送到用户状态队列以节省计算时间。捕捉处理器 124 可以事件计划所描述的格式将事件发送到队列 126。

如果确定 204 事件是不可编入索引的事件，那么可将不可编入索引的事件发送 206 到（例如）队列 126 的用户状态队列以供进一步处理，同时指示不将其编入索引。例如，当用户正撰写电子邮件时，捕捉组件可捕捉用户输入在电子邮件作文中的许多文字。捕捉组件可捕捉输入文本以作为事件。虽然此事件可能有助于确定用户 112a 的当前状态，但其重要程度可能不足以使搜索引擎将其编入索引并发送以进行存储使得用户随后可检索此信息。或者，在用户打开接收到的电子邮件的实例中，此事件可提供关于当前用户状态的信息，但其也可由搜索引擎 122 编入索引并发送以进行存储，因为用户 112a 随后可能关注于检索此电子邮件。

在一个实施例中，队列 126 保留事件直到搜索引擎准备接收事件为止。基于事件数据，可在队列 126 上将事件区分优先次序以供处理。例如，与实时事件相比，可给予历史事件由队列 126 进行处理的较低优先权。在一个实施例中，当索引器 130 准备处理另一事件时，其可从队列 126 中的索引队列检索一个或一个以上事件。查询系统 132 在准备更新用户状态时可从队列 126 的用户状态队列检索一个或一个以上事件。在另一实施例中，不使用队列，且将事件从捕捉处理器 124 直接发送到搜索引擎 122。

作出 204 索引确定且适当地将事件排成队列 206 之后，将事件编入索引并进行存储 208。在一个实施例中，当索引器 130 准备处理事件时，其可从队列 126 检索事件。索引器 130 确定事件是否为复制事件，且如果不是，那么向事件分配事件 ID。索引器 130 还可将事件与相关事件关联。例如，在图 2 所示的实施例中，索引器 130 使用事件计划来确定与事件关联的可编入索引的术语，与事件关联的日期和时间，以及与事件关联的其它数据。索引器 130 可将事件 ID 与包含在索引 142 中的可编入索引的术语关联。事件可存储在数据库 144 中，且事件的内容可存储在存储库 146 中。

所展示的环境反映客户端侧搜索引擎结构实施例。可能有其它实施例，例如独立的客户端装置或网络搜索引擎。

虽然本文已说明并描述了本发明的特定实施例和应用，但应了解，本发明不限于本文揭示的精确构造和组件，且在不脱离所附权利要求书中界定的本发明精神和范围的情况下，可对本发明方法和设备的配置、操作和细节作出各种修改、变化和变更。

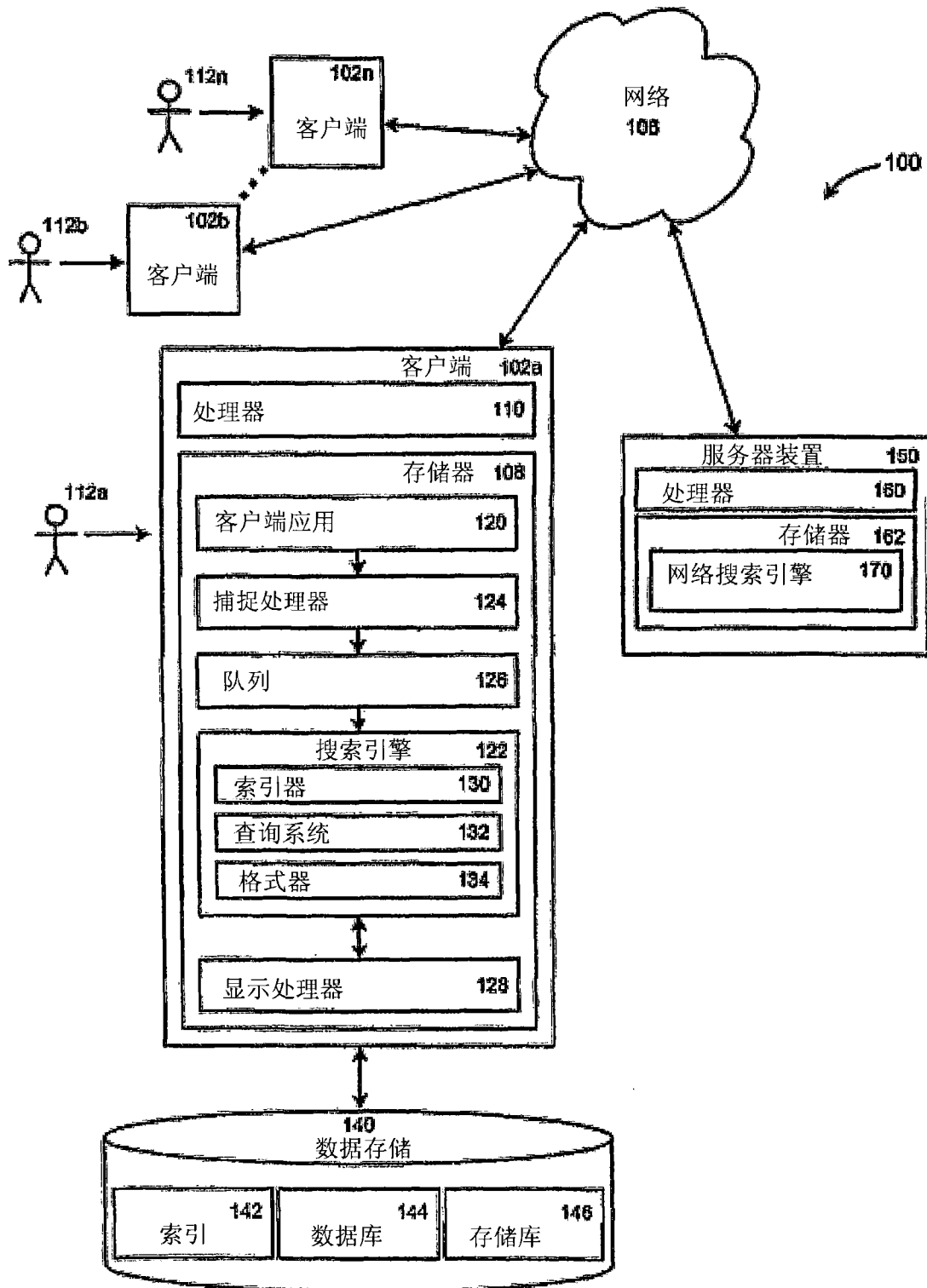


图1

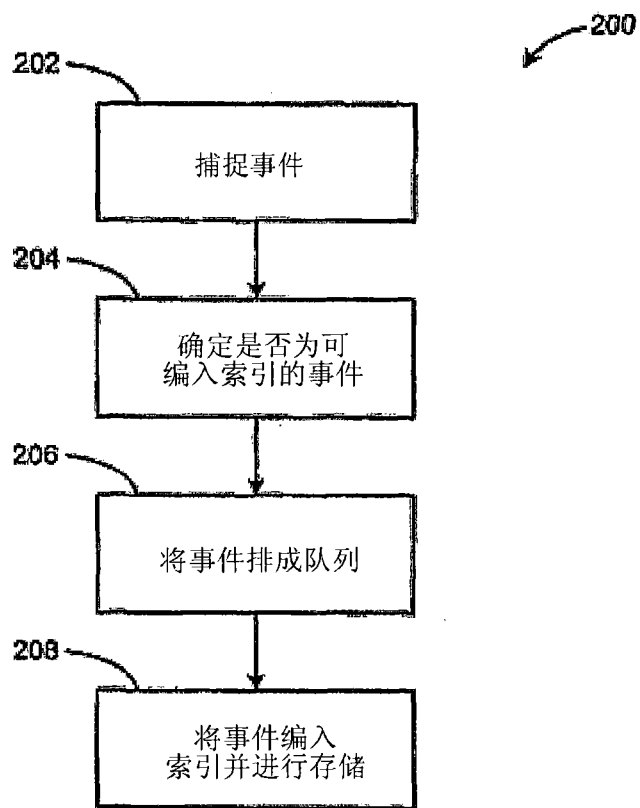


图2

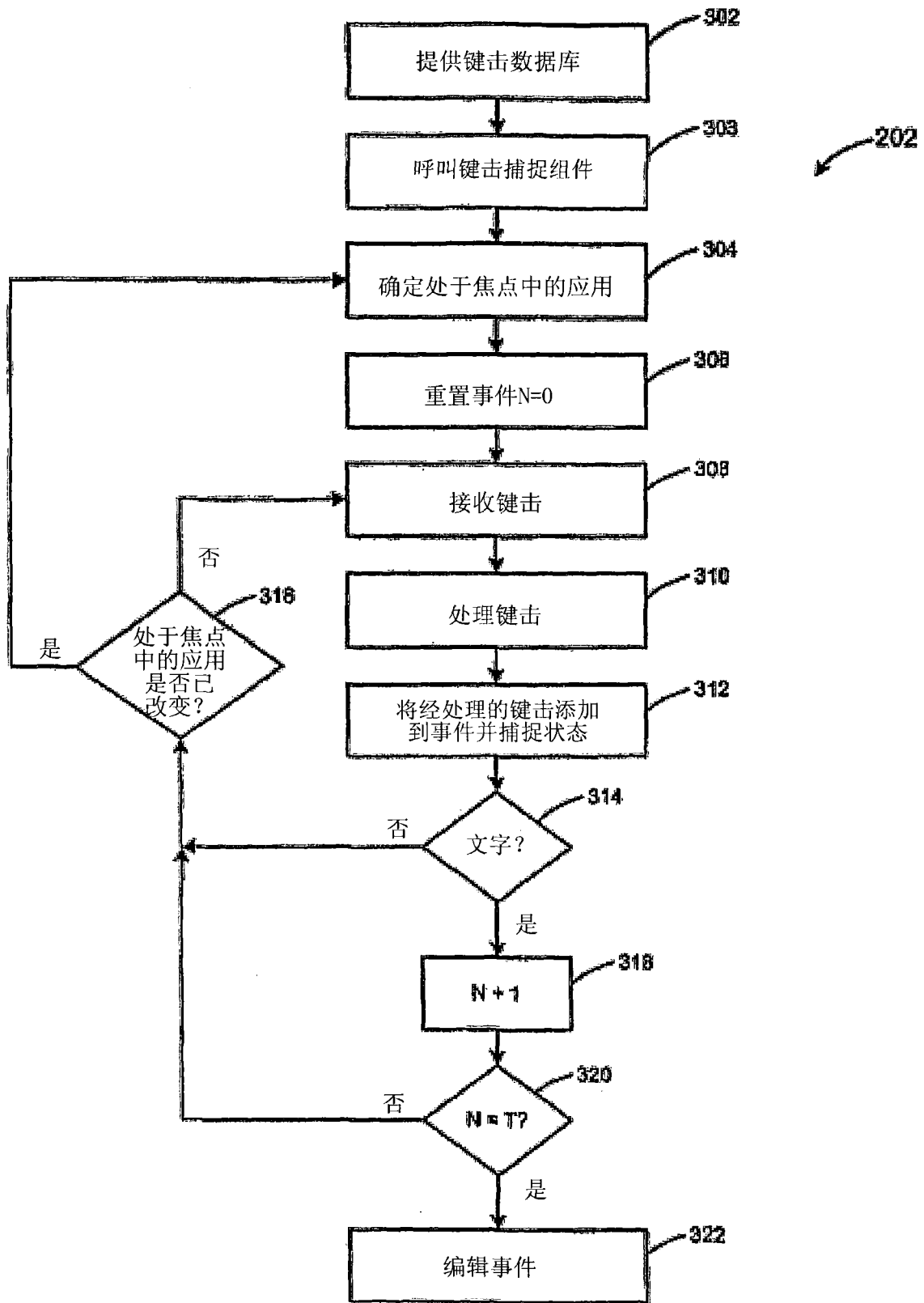


图3

202

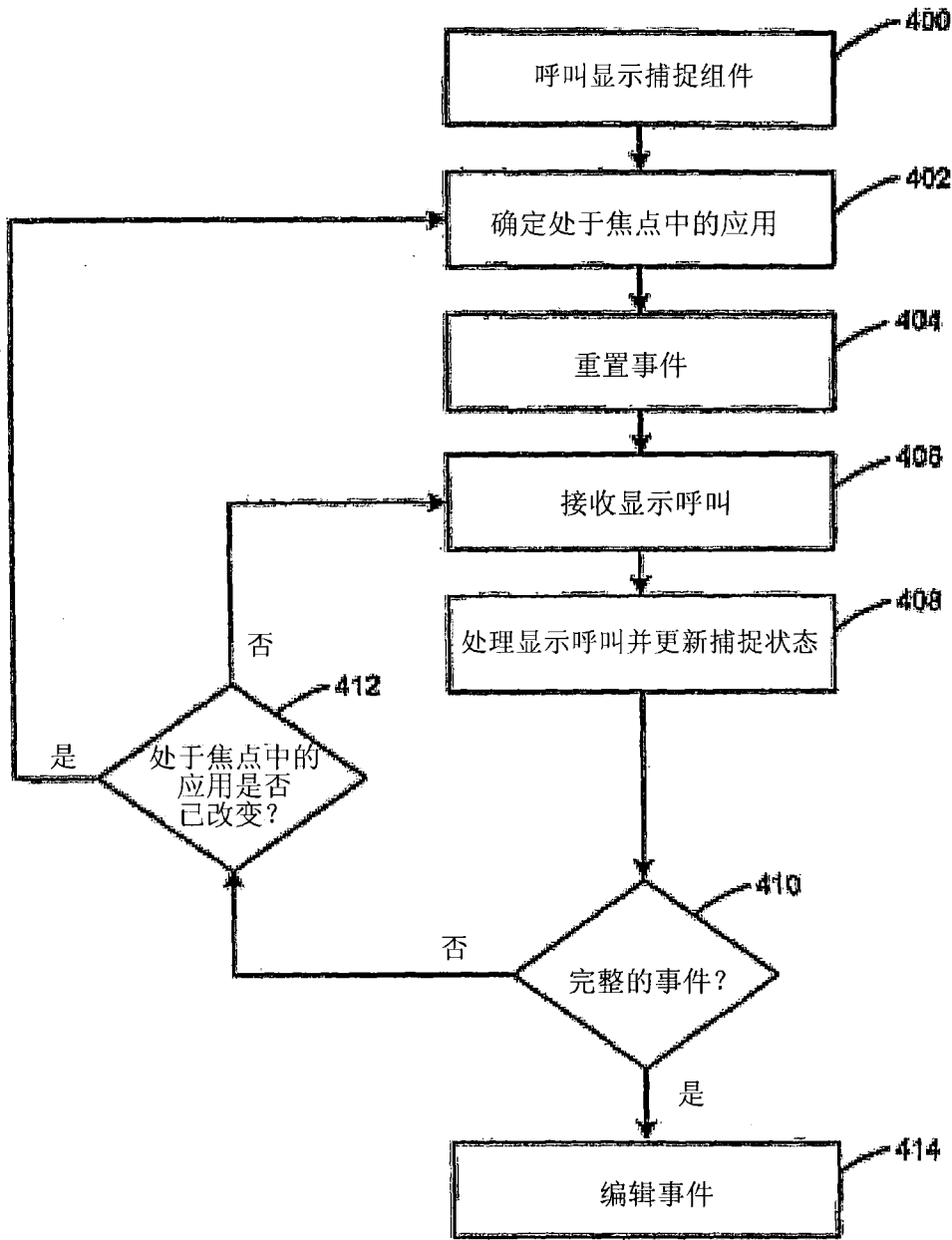


图4

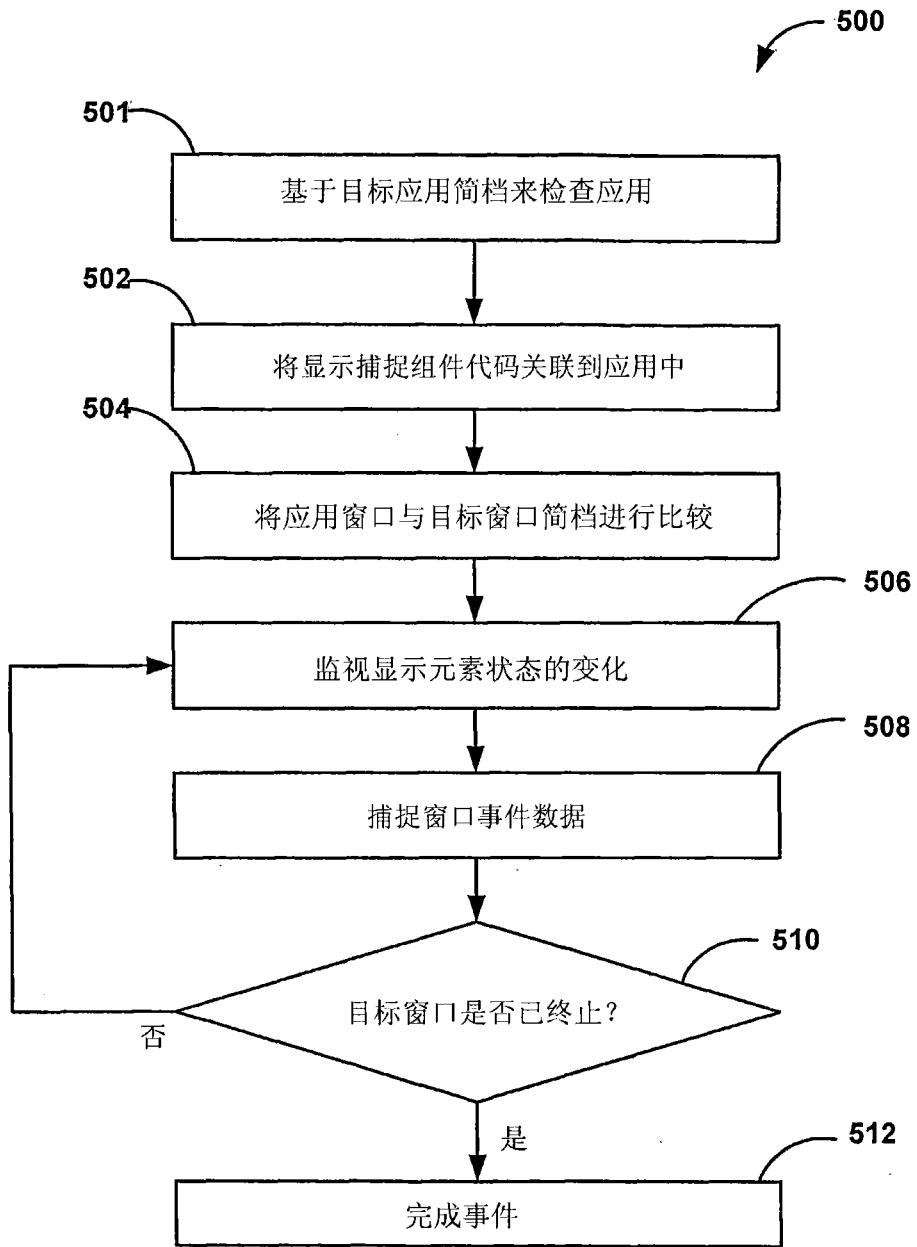


图5