

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/00 (2006.01)

G06F 3/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 200480043357.2

[45] 授权公告日 2009年12月23日

[11] 授权公告号 CN 100573503C

[22] 申请日 2004.7.8

[21] 申请号 200480043357.2

[30] 优先权

[32] 2004.7.8 [33] US [31] 10/887,543

[86] 国际申请 PCT/US2004/022053 2004.7.8

[87] 国际公布 WO2006/016865 英 2006.2.16

[85] 进入国家阶段日期 2006.12.15

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 A·J·麦克格林奇 A·芭拉

J·D·雅各布

[56] 参考文献

CN1151559A 1997.6.11

CN1239253A 1999.12.22

US6219047B1 2001.4.17

US5481667A 1996.1.2

CN1327681A 2001.12.19

CN1155202A 1997.7.23

US2001009420A1 2001.7.26

US6532023B1 2003.3.11

审查员 梁 韬

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 顾嘉运

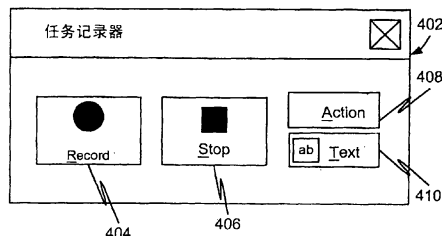
权利要求书 3 页 说明书 20 页 附图 11 页

[54] 发明名称

自动生成内容的方法和系统

[57] 摘要

本发明针对用于自动记录指示创作者在完成用户界面(205)上执行的整体任务的各步骤时的动作的信息的系统(500)。一记录系统(200)被用于记录在具有不同配置的不同计算机(502、504)上执行任务所采取的步骤。所记录的步骤然后被导入创作组件(240)，在那里所记录的步骤被安排成期望的内容。



1. 一种用于生成指示为执行任务而在用户界面上采取的步骤的内容的方法，所述方法包括：

接收第一用户输入，所述第一用户输入指示对第一配置中的第一计算设备的用户界面上的用户界面元素的用户操纵；

响应于所述第一用户输入，在所述第一计算设备上记录指示对所述用户界面元素的操纵的第一信息；

接收第二用户输入，所述第二用户输入指示对第二配置中的第二计算设备的用户界面上的用户界面元素的用户操纵；

响应于所述第二用户输入，在所述第二计算设备上记录指示对所述用户界面元素的操纵的第二信息；以及

将所述第一和第二信息组合以获取所述内容；

其中，所述第一和第二信息包括一图像，该图像指示了用户为执行任务而在一计算设备上所执行的步骤；以及

其中，所述组合包括自动生成指示用户为执行任务而在一计算设备上所执行步骤的内容。

2. 如权利要求 1 所述的方法，其特征在于，所述组合是在第三计算设备上执行的。

3. 如权利要求 1 所述的方法，其特征在于，所述组合是在所述第一和第二计算设备之一上执行的。

4. 如权利要求 1 所述的方法，其特征在于，其中自动生成内容还包括：

自动生成对应于所述第一用户输入的第一文本；以及

自动生成对应于所述第二用户输入的第二文本。

5. 如权利要求 4 所述的方法，其特征在于，自动生成所述第一文本是在所述第一计算设备上执行的。

6. 如权利要求 4 所述的方法，其特征在于，自动生成所述第二文本是在所述第二计算设备上执行的。

7. 如权利要求 1 所述的方法，其特征在于，所述内容包括对为了在具有所述第一和第二配置的计算设备上完成所述任务而执行的步骤的描述。

8. 如权利要求 1 所述的方法, 其特征在于, 所述第一信息包括在所述第一计算设备的用户界面上操纵的用户界面元素的至少一部分的第一图像, 所述第二信息包括在所述第二计算设备的用户界面上操纵的用户界面元素的至少一部分的第二图像。

9. 如权利要求 8 所述的方法, 其特征在于, 所述组合包括:

在编辑器组件上显示所述第一和第二图像, 所述编辑器组件被配置成接收对所述第一和第二计算设备上的用户界面元素所做的用户操纵的文字描述。

10. 如权利要求 9 所述的方法, 其特征在于, 所述组合包括:

将所述第一和第二图像嵌入到所述文字描述中。

11. 一种用于生成描述由用户采取以便在用户界面上执行任务的步骤的内容的内容生成系统, 包括:

部署在具有第一配置的第一计算设备上的第一记录系统, 所述第一记录系统被配置成接收关于用户已经在所述第一计算设备上采取了步骤的指示, 并记录指示所述步骤的第一信息;

部署在具有第二配置的第二计算设备上的第二记录系统, 所述第二记录系统被配置成接收关于用户已经在所述第二计算设备上采取了步骤的指示, 并记录指示所述步骤的第二信息; 以及

被配置成接收所述第一和第二信息并组合所述第一和第二信息以形成所述内容的编辑器组件;

其中, 所述第一和第二信息包括一图像, 该图像指示了用户为执行任务而在一计算设备上所执行的步骤; 以及

其中, 所述编辑器组件包括自动文本生成系统, 配置用于自动生成指示用户为执行任务而在一计算设备上所执行步骤的内容。

12. 如权利要求 11 所述的内容生成系统, 其特征在于, 所述编辑器组件被部署在第三计算设备上。

13. 如权利要求 11 所述的内容生成系统, 其特征在于, 所述编辑器组件被部署在所述第一和第二计算设备之一上。

14. 如权利要求 11 所述的内容生成系统, 其特征在于, 所述第一和第二记录系统被配置成将所述第一和第二信息分别记录为指示所述用户在所述第一和第二计算设备上所执行的步骤的第一和第二图像。

15. 如权利要求 14 所述的内容生成系统, 其特征在于, 所述编辑器组件被配

置成显示所记录的第一和第二图像并接收相关联的文本。

16. 如权利要求 15 所述的内容生成系统，其特征在于，所述编辑器组件被配置成生成带有嵌入在所述相关联文本中的图像的最终内容。

17. 如权利要求 11 所述的内容生成系统，其特征在于，

其中，自动文本生成系统被配置成接收所述第一信息并基于所述第一信息自动生成描述所述用户在所述第一计算设备上所执行的步骤的文本。

18. 如权利要求 17 所述的内容生成系统，其特征在于，所述自动文本生成系统被配置成接收所述第二信息并基于所述第二信息自动生成描述所述用户在所述第二计算设备上所执行的步骤的文本。

19. 如权利要求 18 所述的内容生成系统，其特征在于，所述自动文本生成系统被部署在第三计算设备上。

20. 如权利要求 18 所述的内容生成系统，其特征在于，所述自动文本生成系统包括：

部署在所述第一计算设备上并基于所述第一信息生成文本的第一自动文本生成器；以及

部署在所述第二计算设备上并基于所述第二信息生成文本的第二自动文本生成器。

自动生成内容的方法和系统

发明背景

本发明涉及生成内容，诸如帮助内容。更具体地，本发明涉及对指示用户在用户界面上的动作自动生成的内容的导入。

图形用户界面（GUI）是广泛使用的界面机制。GUI 适于定位任务（例如，对矩形调整大小）、视觉修改工具任务（例如，对某物加上难以名状的蓝色阴影）或选择任务（例如，这是一百个图片中我想要旋转的一个）。GUI 也适于对快速单步特征的迅速访问。应用程序的 GUI 是从功能角度（例如，组织成菜单、工具栏等）而非面向任务角度（例如，由用户想要进行的高级任务组织的，诸如“使我的计算机免受黑客威胁”）组织的有用的工具箱。

然而，GUI 也对用户提出了众多问题。使用工具箱类似物，用户难以在箱中找到工具或了解如何使用工具来完成由多个步骤组成的任务。由强制表现为不透明层次的单个词语、小按钮和标签描述的界面使其自身不适用于人们思考他们的任务的方式。GUI 要求用户分解任务以便确定什么元素是完成任务所必需的。这种要求产生了复杂性。除复杂性以外，组装 GUI 元素（即，菜单点击、对话点击等）是花费时间的。这即使对专家用户而言也是低效且消耗时间的。

解决 GUI 问题的一个现有机制是书面的帮助程式。帮助程式通常采用帮助文档、PSS（产品支持服务）KB（知识库）文章、和新闻组公告的形式，这填补了顾客需求与 GUI 问题之间的缺口。它们类似于与工具箱一起提供的手册，并具有许多好处。这些好处包括，作为示例：

- 1) 技术上而言，即使对非技术作者，它们也相对易于创作；
- 2) 它们易于在服务器上更新，因此连接的用户能够轻易访问新的内容；以及
- 3) 它们指示 GUI，从而使用户能够控制解决问题。

然而，帮助文档、PSS KB 文章和新闻组有它们自己的问题。这些问题包括，作为示例：

- 1) 复杂任务需要在用户一方上进行大量处理。用户需要进行从每一步骤的描述到 GUI 的映射。在被跳过、被不正确或不充分描述或被乱序描述了的步骤处可

导致错误。

2) 故障诊断程序甚至程式性帮助文档通常也包括在帮助主题下产生复杂分支的状态信息,使得主题过长而难以由最终用户阅读和处理。工具栏可能遗失,或可能在采取下一步骤之前需要被开启。故障诊断程序通常询问关于状态的问题,这从最好的角度而言也是令人灰心的(因为,故障诊断程序应能够自己找出答案),从最坏的角度而言则对非专家而言是不能回答的。

3) 存在成百万的文档,搜索答案涉及从哪里开始搜索以及如何从返回的数千结果中挑选最佳搜索结果的问题。

4) 不存在共享的创作结构。新闻组公告、KB 文章、故障诊断程序和程式性帮助文档皆具有不同的结构和创作策略,然而它们都解决类似的问题。

5) 对用户而言,仅是难以阅读逐步的文本,然后视觉上在 UI 中搜索正被描述的元素并对该元素采取所描述的动作。

解决 GUI 问题的另一现有机制是向导。创建向导来解决 GUI 和书面的帮助程式的弱点。现在存在数千向导,可在所制造的几乎每个软件产品中找到这些向导。这是因为向导解决了目前现有的基于文本的帮助和协助未能解决的实际需求。它们允许用户以面向任务的方式访问功能,并可自动组装 GUI 或工具。向导向项目主管(program manager)和开发人员提供了解决顾客任务的手段。这类似于工具箱中的专家,引导用户逐步通过为任务成功而必需的步骤。某些向导帮助顾客安装系统(例如,安装向导),某些向导包括具有特征的内容,并帮助顾客创建内容(例如,Newsletter 向导或 PowerPoint 的 AutoContent 向导),某些向导帮助顾客诊断并解决问题(例如,故障诊断程序)。

向导向用户提供众多好处。向导的好处中的某些在于:

1) 向导能够实体化“任务”的概念。用户通常清楚向导在帮助他们完成什么任务。使用逐步的页面,用户易于作出选择,并且在良好设计的向导的情况中,通常减少了用户在视觉上被淹没的发生率。

2) 向导可自动组装并与软件的底层特征交互,且包括顾客作出选择所需的信息或专家意见。这节省了用户执行任务的时间。

3) 向导可自动生成内容,并可通过创建文本和规划布局来节省用户时间。

4) 向导也是用于询问问题、获得回答并分支到最相关的下一问题或特征的良好手段。

然而,向导也有它们自身的问题。这些问题中的某些包括,人们试图完成的

任务远多于用于完成这些任务的向导。向导和 IUI（引导性用户界面）不教导用户如何使用底层 GUI，且通常当向导完成时，用户不确定下一步做什么。创作向导的成本仍旧高，且要求具有技术专业知识的（例如，软件开发员）来创作向导。

此外，所有这些类型的内容还遭受另一问题。执行任何给定任务所必须采取的步骤可能基于其上执行该任务的计算机的配置而改变。例如，取决于用户计算机的操作系统，改变计算机上的背景显示（即“壁纸”）可能要求用户执行不同的步骤。事实上，如果操作系统的版本号不同，所需步骤也甚至可能不同。类似地，取决于计算机的网络配置（例如，取决于计算机位于一网络域还是一工作组上），步骤可能不同。这要求用户在书面内容中创作相当复杂的分支逻辑。同样，用户可能觉得导航通过一复杂的分支帮助文本来执行必要步骤是麻烦。

因此，创作描述将由用户采取的过程的所有这些类型的内容通常易于产生错误。很容易就遗失步骤、不正确地描述步骤、或在 UI 操纵的长序列中丢失对当前正描述什么步骤的跟踪。然而，这种书面程式性帮助内容非常常见。这样的帮助内容通常与产品一起装运，为产品支持组提供在线帮助内容，且公司内的程式通常以这样的方式进行文档编写以用于专用商业过程。因此，这种类型的信息是难以创作且通常包含错误的。

此外，最终用户一般遵循已创作的步骤。难以阅读逐步的文本，然后在 UI 中搜索正被描述的特定控件元素，然后针对该控件元素采取适当的措施。发现众多用户发觉这是相当的负担，他们仅扫视文本的前一两个步骤，然后尽力确定下一步需要启动哪个 UI 元素，很少回头参考书面的文本步骤。也发现，相对于阅读单词并在心里将单词转换成图片然后找到相应的 UI 控件元素而言，眼睛能更容易地找到和识别图片。然而，在过去，正是进行了如下步骤，创作者必须煞费苦心地获得每一步骤的屏幕截图、裁剪图像然后将它们粘贴到文档中的适当位置中，以便拥有对要采取的措施的任何类型的视觉描述。

发明概述

本发明的一个实施例通过提供一种内容组件解决了向导、帮助、知识库文章和故障诊断程序的问题中的某些问题，该内容组件提供了简单的方式来创作数千任务（或向导）的，并且或与 GUI 集成并教导用户如何使用 GUI 来执行任务，或代表用户来执行任务。在一个特定实施例中，本发明涉及创作活动内容向导（ACW）脚本，以及作为 ACW 脚本的一部分的文本和图像。

本发明针对用于自动记录指示创作者在完成用户界面上执行的整体任务的各步骤时的动作的信息的系统。记录系统被用于记录为在具有不同配置的不同计算机上执行任务所采取的步骤。所记录的步骤然后被导入创作组件，在那里所记录的步骤被编排成期望的内容。

在一个实施例中，记录系统记录对应于创作者在用户界面上的动作的图像。记录系统然后将所记录的图像传递给创作组件，在那里创作者可生成对应于该图像的文本以描述这些动作。如有需要，图像也可被嵌入在文本中来发布。

在一个实施例中，记录系统包括自动生成对应于图像的文本的文本生成器。这种文本然后可用于形成文本文档，它向用户提供指令或其它信息。在生成文本文档的过程期间或之后，文本可使用编辑器来编辑以增强文档的可理解性。

附图简述

图 1 是可在其中使用本发明的一个示例性环境的框图。

图 2 是示出根据本发明的一个实施例的记录系统的组件的框图。

图 3 是示出图 2 中所示的系统在记录和生成与用户动作相关的文本时的操作的流程图。

图 4 是描绘记录系统的控件的一个实施例的屏幕截图。

图 5 是示出可用于生成文本的示例性显示的屏幕截图。

图 6 是示出根据本发明的一个实施例的高亮特征的屏幕截图。

图 7 是更详细示出自动文本生成的一个实施例的流程图。

图 8 是示出向创作者呈现所生成的文本以便编辑的一种方式的屏幕截图。

图 9 是更好地示出了其中图像可用于创作内容的一个实施例的屏幕截图。

图 10 示出了根据本发明的一个实施例的带有嵌入图像的最终文本的一个实施例。

图 11 是用于对在具有不同配置的系统上执行任务的用户界面步骤进行纪录的系统的框图的一个实施例的框图。

示例性实施例的详细描述

本发明涉及对指示用户在用户界面上的动作的内容的自动记录。在详细描述本发明的之前，将讨论可在其中使用本发明的一个示例性环境。

图 1 示出了可在其上实现本发明的合适的计算系统环境 100 的示例。计算系

统环境 100 只是合适的计算环境的一个示例，并不旨在对本发明的使用范围或功能提出任何限制。也不应该把计算环境 100 解释为对示例性操作环境 100 中示出的任一组件或其组合有任何依赖性要求。

本发明可用众多其它通用或专用计算系统环境或配置来操作。适合在本发明中使用的公知的计算系统、环境和/或配置的示例包括，但不限于，个人计算机、服务器计算机、手持或膝上型设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费者电子产品、网络 PC、小型机、大型机、包含上述系统或设备中的任一个的分布式计算机环境等。

本发明可在由计算机执行的诸如程序模块等的计算机可执行指令的通用语境中描述。一般而言，程序模块包括例程、程序、对象、组件、数据结构等，它们执行特定任务或实现特定抽象数据类型。本发明也可以在分布式计算环境中实现，其中任务由通过通信网络连接的远程处理设备执行。在分布式计算环境中，程序模块可以位于包括存储器存储设备在内的本地和远程计算机存储介质中。

参考图 1，用于实现本发明的一个示例性系统包括计算机 110 形式的通用计算设备。计算机 110 的组件可以包括，但不限于，处理单元 120、系统存储器 130 和将包括系统存储器在内的各种系统组件耦合至处理单元 120 的系统总线 121。系统总线 121 可以是若干类型的总线结构中的任一种，包括存储器总线或存储器控制器、外围总线和使用各种总线体系结构中的任一种的局部总线。作为示例，而非限制，这样的体系结构包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MCA) 总线、扩展的 ISA (EISA) 总线、视频电子技术标准协会 (VESA) 局部总线 and 外围部件互连 (PCI) 总线 (也被称为 Mezzanine 总线)。

计算机 110 通常包括各种计算机可读介质。计算机可读介质可以是能够被计算机 110 访问的任何可用介质，且包括易失性和非易失性介质、可移动和不可移动介质。作为示例，而非限制，计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以任何方法或技术实现的用于存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括，但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘 (DVD) 或其它光盘存储、磁带盒、磁带、磁盘存储或其它磁性存储设备、或能用于存储所需信息且可以由计算机 110 访问的任何其它介质。通信介质通常具体化为诸如载波或其它传输机制等已调制数据信号中的计算机可读指令、数据结构、程序模块或其它数据，且包含任何信息传递介质。术语

“已调制数据信号”指的是这样一种信号，其一个或多个特征以在信号中编码信息的方式被设定或更改。作为示例，而非限制，通信介质包括有线介质，诸如有线网络或直接线连接，以及无线介质，诸如声学、RF、红外线和其它无线介质。上述中任一个的组合也应包括在计算机可读介质的范围之内。

系统存储器 130 包括易失性和/或非易失性存储器形式的计算机存储介质，诸如只读存储器（ROM）131 和随机存取存储器（RAM）132。基本输入/输出系统 133（BIOS）包含有助于诸如启动时在计算机 110 中元件之间传递信息的基本例程，它通常存储在 ROM 131 中。RAM 132 通常包含处理单元 120 可以立即访问和/或目前正在操作的数据和/或程序模块。作为示例，而非限制，图 1 示出了操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137。

计算机 110 也可以包括其它可移动/不可移动、易失性/非易失性计算机存储介质。仅作为示例，图 1 示出了从不可移动、非易失性磁介质中读取或向其写入的硬盘驱动器 141，从可移动、非易失性磁盘 152 中读取或向其写入的磁盘驱动器 151，以及从诸如 CD ROM 或其它光学介质等可移动、非易失性光盘 156 中读取或向其写入的光盘驱动器 155。可以在示例性操作环境下使用的其它可移动/不可移动、易失性/非易失性计算机存储介质包括，但不限于，盒式磁带、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等。硬盘驱动器 141 通常由不可移动存储器接口，诸如接口 140 连接至系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常由可移动存储器接口，诸如接口 150 连接至系统总线 121。

以上描述和在图 1 中示出的驱动器及其相关联的计算机存储介质为计算机 110 提供了对计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 1 中，硬盘驱动器 141 被示为存储操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147。注意，这些组件可以与操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同或不同。操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 在这里被标注了不同的标号是为了说明至少它们是不同的副本。

用户可以通过输入设备，诸如键盘 162、麦克风 163 和诸如鼠标、跟踪球或触摸垫的定点设备 161 向计算机 110 输入命令和信息。其它输入设备（未示出）可以包括操纵杆、游戏垫、圆盘式卫星天线、扫描仪等。这些和其它输入设备通常由耦合至系统总线的用户输入接口 160 连接至处理单元 120，但也可以由其它接口或总线结构，诸如并行端口、游戏端口或通用串行总线（USB）连接。监视器 191 或其

它类型的显示设备也经由接口，诸如视频接口 190 连接至系统总线 121。除监视器以外，计算机也可以包括其它外围输出设备，诸如扬声器 197 和打印机 196，它们可以通过输出外围接口 195 连接。

计算机 110 可使用至一个或多个远程计算机，诸如远程计算机 180 的逻辑连接在网络化环境下操作。远程计算机 180 可以是个人计算机、手持式设备、服务器、路由器、网络 PC、对等设备或其它常见网络节点，且通常包括上文相对于计算机 110 描述的许多或所有元件。图 1 中所示逻辑连接包括局域网（LAN）171 和广域网（WAN）173，但也可以包括其它网络。这样的网络环境在办公室、企业范围计算机网络、内联网和因特网中是常见的。

当在 LAN 连网环境中使用时，计算机 110 通过网络接口或适配器 170 连接至局域网 171。当在 WAN 连网环境中使用时，计算机 110 通常包括调制解调器 172 或用于通过诸如因特网等 WAN 173 建立通信的其它装置。调制解调器 172 可以是内置或外置的，它可以通过用户输入接口 160 或其它合适的机制连接至系统总线 121。在网络化环境中，相对于计算机 110 所描述的模块或其部分可以存储在远程存储器存储设备中。作为示例，而非限制，图 1 示出了远程应用程序 185 驻留在远程计算机 180 上。可以理解，所示的网络连接是示例性的，且可以使用在计算机之间建立通信链路的其它手段。

图 2 是示出根据本发明的一个实施例的记录系统的组件的框图。图 2 也示出了带有记录器和自动文本生成器的系统。记录器包括记录组件 210、挂钩组件 212 以及用户界面（UI）自动化组件 214。文本生成器包括文本数据库（或数据存储）220 以及文本生成组件 230。系统与用户界面 205 交互。创作者可配置系统的组件来自动记录对应于在用户界面 205 的控件上执行的动作的图像，文本生成器自动生成描述这些动作的文本 235。创作者 201 也可在编辑器 240 上编辑自动生成的文本 235 或从草稿生成文本以获取描述任务（或 UI 控件动作）的最终文本 245。最终文本 245 也示例性地包括嵌入其中的图像。现在将详细描述图 2 中的多个组件。

在一个实施例中，用户界面 205 是具有允许用户采取动作来执行任务的控件的常规图形用户界面。用户界面 205 示例性地显示在图 1 中所示的显示设备 191 上。这种类型的图形用户界面（GUI）是广泛使用的界面机制。

记录组件 210 在一个实施例中是允许创作者 201 或另一用户在用户界面 205 上执行任务并通过捕捉任务中每一步骤的图像来记录任务的应用程序。如将在以下更详细描述的，当创作者 201 在用户界面 205 上执行与任务相关联的步骤时，记录

组件 210 记录关于创作者在用户界面 205 上与什么控件和窗口交互的信息。这种信息被示例性地提供给文本生成器 230 以自动在诸如帮助文档的文档中生成文本。

记录组件 210 经由挂钩组件 212 和用户界面 (UI) 自动化组件 214 与用户界面 205 交互。这些组件可与记录组件 210 分开,或在某些实施例中,这些组件可与记录组件 210 集成在一起。

在一个实施例中,挂钩组件 212 是由计算机使用的操作系统内的模块或组件。例如,当挂钩被设定为鼠标点击时,指示鼠标点击的信息(诸如消息)被转发给挂钩组件 212,在那里消费该信息,且当其相关联的图像由记录组件 210 记录之后,为注册以接收鼠标点击的计算机中的其它组件回放该信息。从而,一般,挂钩组件 212 用作操作系统和目标应用程序之间的缓冲区。

挂钩组件 212 可被配置成查找基本上任何输入动作,诸如所接收的信号的类型,例如单击、双击、右击或左击、键盘动作、触敏屏输入等。一旦表示动作的信息由记录组件 210 记录之后,表示鼠标点击(或所记录的任何动作)的信息就由挂钩组件 212 向应用程序回放。对此的一个理由是,用户可在记录第一动作之前采取第二动作。第二动作可使用户界面的状态改变,并因此导致对第一动作的不正确记录。通过消费第一鼠标消息并当一旦完成记录之后对其回放,这确保了第一动作将被正确记录。

也应注意,由挂钩组件 212 执行的功能(即,监听鼠标点击并对其回放)被示例性地在分开的线程上执行。这确保了所有用户界面动作(例如,鼠标点击、键盘动作等)将没有遗漏地被正确记录并回放。此外,挂钩组件 212 的记录和回放机制可示例性地无视操作系统内隐含的任何超时特征。这可能是必须的,如果操作系统的超时周期过短而无法允许正确的记录动作。

用户界面自动化组件 214 示例性地是被配置成解释由创作者或用户通过用户界面 205 执行的整体任务的原子步骤的计算机程序。在一个实施例中,用户界面自动化组件 214 是使用华盛顿州雷德蒙市的微软公司提供的微软用户界面自动化来实现的 GUI 自动化模块。该模块提供了一种程序化方式,以访问关于可视用户界面的信息并在程序上与可视用户界面交互。然而,取决于系统设置,用户界面自动化组件 214 可使用能够程序上导航图形用户界面并检测(和可任选地在程序上导航 GUI 来执行和运行)用户界面上的命令的任何应用程序来实现。

用户界面自动化组件 214 因此检测与创作者 201(或另一用户)以任务顺序在用户界面 205 上执行的期望任务相关联的步骤中的每一个。例如,如将在以下更详

详细描述，当任务需要用户在 GUI 上点击按钮以显示新菜单或窗口时，用户界面自动化组件 214 确定哪一控件位于用户界面 205 上的鼠标光标的位置处及其大小和它的父窗口。记录组件 210 使用来自挂钩组件 212 的信息（例如，控件的类型、名字和状态）来记录被用于执行该步骤的控件的名字和属性。该信息由用户界面自动化组件 214 和挂钩组件 212 向记录组件 210 提供，使得记录组件 210 可记录由创作者用来执行该步骤的按钮或控件的图像。以下将参考图 3 更详细描述对图像的获取。

文本生成组件 230 是被配置成生成描述在记录过程期间被运行或执行的动作的自然语言文本的程序或模块。文本生成组件 230 使用所记录的图像和由记录组件 210 记录的其它信息来搜索数据库 220，并从文本数据库 220 中选择对应于所记录步骤的正确的模板或条目。

文本数据库 220 示例性地是可由文本生成器 230 搜索的数据库或其它信息存储系统。文本数据库 220 包含与在用户界面 205 上可使用的控件相关的信息。这种信息可包括例如，控件的名字、控件的类型、在控件上执行的动作、以及作为自然语言句子的动作的文字描述。

在某些实施例中，以多种语言提供对条目的文字描述。当以多种语言提供文字描述时，每一条目配备语言标识符，以允许正确的语言被选择。

然而，取决于系统需求，可在文本数据库 220 中提供其它信息。在一个实施例中，文本数据库 220 中的某些条目含有与由顺序执行的多个控件例示的两个或多个动作相关的信息。当由文本数据库 220 中的单个条目表示多个控件上的多个动作时，条目的文本包含作为单个句子的在两个控件上执行的动作的自然语言描述。通过将两个命令的描述组合成单个句子，改进了最终文本文档的可读性。

在一个实施例中，文本数据库 220 以可扩展标记语言（XML）编写。每一条目的数据可作为一连串子条目被存储，其中条目的每一子条目指的是标识任务所需的单独的信息片段。然而，可使用其它格式来存储数据。

在一个实施例中，文本生成组件 230 当在文本数据库 220 中搜索条目时查看所记录的动作中的两个或多个。这样做是为了提供更流畅的文本文档。例如，良好的程式性文档一般将多于一个的步骤组成单个句子，以增强可读性。如果文本生成组件 230 标识了文本数据库 220 中与所记录的信息相匹配的两个或多个条目时，则文本生成组件 230 可使用任何已知方法来确定要选择数据库中的哪个条目，诸如通过基于对每一条目记分并选择具有最高分的条目来去除条目的歧义。

根据一个实施例，基于用户界面上所启动的控件的类型和所执行的动作，文

本生成组件 230 在文本数据库 220 中搜索匹配所执行的控件类型和动作的条目。一旦在文本数据库 220 中标识了匹配之后,文本生成组件 230 从文本数据库 220 中获取对动作的相关联的自然语言描述,并将其作为句子指令而放置于所生成的文本文档 235 中。在替换实施例中,文本生成组件 220 也可基于由 UI 自动化模块 214 提供的信息生成文本文档的可执行版本。

当从文本数据库 235 中选择文字描述时,文本生成组件也可察看控件的状态。这当控件是复选框或者是可展开或折叠的树时是重要的。在这种情况下,仅在框上点击可能不适于描述该动作,因为无论期望的结果如何,控件上的动作都是相同的。因此,在这些情况中,控件的新状态将影响所选择的文本。例如,如果控件是复选框,且它要被取消选定,则所匹配的文本将基于控件的新状态加上控件的名字。

文本编辑器 240 是被配置成对自动生成的文本 235 进行更正、改变或添加信息或文本的编辑器。取决于由文本生成器 230 生成的结果文本,以及创作者执行的动作,可能有必要编辑文本以进一步增强其可理解性。从而,文本编辑器 240 接收所生成的文本 235,并允许创作者 201 编辑所生成的文本。

例如,由于语法上的必要性或因为所记录的步骤之一要求一用户动作而系统在记录该步骤时没有要求对用户动作的描述,就需要文本编辑。在这样的情况中(当需要用户输入时),根据一个实施例,在执行将被记录的动作的同时,文本生成组件 230 仅在文本中提供空格,以便创作者提供关于用户在此步骤中应做的动作的指令/描述。

例如,假定正由用户执行并由记录组件记录的任务是改变铺在计算机屏幕上的背景。这要求用户为背景选择图案。从而,由文本数据库返回的用于改变背景的用户动作的文本可以是“请选择[插入动作的描述]”,其中创作者将编辑文本成“请从列表中选择所需的背景”。同样在编辑阶段期间,如果在记录任务之前未提供对整个任务的描述,则创作者 201 可提供对此的描述。一旦文本被编辑之后,最终文本 245 从创作工具中输出,并以允许最终文本当用户期望时可被检取的适当的存储模式来存储。

图 3 是根据本发明的一个实施例,更详细地示出系统如何操作的流程图。首先,创作者 201 启用记录组件 210。这由图 3 中的框 400 指示。为此,创作者 201 仅启动诸如图 4 中所示的控件 402 的控件。Record(记录)按钮 410 可由创作者 201 启动以开始记录。启动 Stop(停止)按钮 406 以停止记录。如将在以下描述,创作者 201 可在文本编辑期间将对用户动作的描述插入到文本编辑器 240 中。为此,

用户启动按钮 408。类似地，如以下描述的，创作者 201 可编辑自动生成的文本。这可通过启动图 4 中的 Text（文本）按钮 410 来完成。或者，在某些实施例中，可提供附加的按钮来允许用户设置一值。这是为了允许当例如在运行时用户被要求将文本键入编辑框中的情况中生成正确的文本。当然，也可使用其它或不同的控件。

再次参考图 3，一旦创作者 201 启用记录组件 210，系统仅等待用户在用户界面 205 上采取动作。注意到，图 2 示出了该用户是创作者 201，但用户也可以是不同的用户。

一旦用户在用户界面 205 上采取了动作（诸如，通过在用户界面上操纵控件元素）之后，挂钩组件 212 接收指示用户动作的消息或其它信号。如上所述，对于挂钩组件 212，挂钩组件 212 消费该消息，并将其置于记录队列中。图 3 中的框 412 指示了用户在 UI 205 上采取动作。

记录组件 210 然后从 UI 自动化组件 214 接收图像标识信息。这由图 3 中的框 414 指示。在一个示例性实施例中，UI 自动化组件 214 向记录组件 210 提供允许记录组件 210 记录显示屏上的表示或对应于用户在用户界面 205 处采取的动作的图像的多个信息项。在一个示例性实施例中，这些信息项是用户启动或以其它方式操纵的控件元素在显示屏上的位置、该控件元素的大小、以及包含该控件元素的父窗口。

记录组件 210 然后获取指示与用户界面 205 相关联的并对应于或反映由用户采取的动作的屏幕截图的实际图像信息。这由图 3 中的框 416 指示。

为了执行该步骤，记录组件 210 可做许多事以增强系统的操作。例如，记录组件 210 可确定记录比用户仅操纵的控件元素要多的实际图像信息（或实际屏幕截图）可能是有帮助的。这可能是对的，例如，如果当前在正由用户操纵的显示上存在多于一个类似的控件元素时。假定例如，用户在用户界面上点击了“OK 按钮”。然而，此时在显示屏上可能存在多于一个的“OK 按钮”。从而，为了去除各个“OK 按钮”之中的歧义，记录组件 210 可获取不仅是由用户操纵的特定“OK 按钮”的实际屏幕截图信息，还有围绕该“OK 按钮”的所需多个像素的屏幕截图信息。这提供了具有比单单控件本身的图像更多上下文的图像。

类似地，记录组件 210 也可记录包含控件元素的整个父窗口的屏幕截图。当然，这包含了可用于特别标识用户操纵的控制元件的大量额外上下文。

为了确定附加的上下文是否需要由记录组件 210 记录，记录组件 210 可使用各种不同的技术中的任一个进行判断。例如，记录组件 210 可采用将标识要记录的

上下文数量的试探法。这种试探法可基于所操纵的控件元素的大小和形状、所操纵的控件元素的特定功能、控件元素在屏幕上的位置（例如，如果控件元素位于左上角，则记录组件 210 可在控件元素的右下侧取得较多的像素），或试探法可仅仅反映将围绕控件元素取得的固定数量的像素，而不考虑它位于何处或控件元素执行了何种功能。

记录组件 210 可使用任何已知技术获取实际屏幕截图信息。例如，在大多数操作系统中，存在公开的应用程序编程接口（API）允许应用程序或其它计算组件获取当前正显示的屏幕截图信息。从而，在一个示例性实施例中，一旦记录组件 210 了解到它需要的屏幕截图信息的坐标、上下文信息的数量以及可任选地控件元素的父窗口之后，它简单地进行 API 调用以获取信息。

获取了实际图像信息之后，记录组件 210 对其记录以供稍后使用。这由图 3 的框 418 指示。当然，此时也注意到，记录组件 210 可记录由 UI 自动化组件 214 提供的其它信息。例如，UI 自动化组件 214 示例性地向记录组件 210 提供控件名、控件类型、在控件上执行的动作、所执行的操纵的类型（诸如鼠标点击、鼠标滚轮旋转、键盘键击、触摸垫输入等）。这些信息均可由记录组件 210 记录。

根据本发明的一个实施例，文本生成组件 230 结合文本数据库 220 自动生成与所捕捉的图像相关联并与用户在用户界面 205 上采取的动作相关联的文本。在使用这些项的实施例中，记录组件 210 将所捕捉的实际图像信息发送给文本生成组件 230。这由图 3 中的框 420 指示。自动生成的文本示例性地提供对应于每一用户对用户界面 205 的操纵的逐步指令的书面程式以便执行需要对用户界面 205 的多个操纵的整体任务。

以下参考图 5-8 描述用于自动生成文本的一个实施例。简要地，为了生成该文本，文本生成组件 230 可使用任何合适的方法。在一个示例性实施例中，文本生成组件 230 在文本数据存储 220 中搜索对应于从记录组件 210 接收的信息的条目。例如，文本数据存储 220 示例性地可以是包含多个条目的 XML 数据库，该多个条目包括由用户在用户界面 205 上操纵的控件或其它项的类型、动作的类型和对应于该动作的文本。当然，可使用其它数据存储方法来实现数据存储 220，且数据存储 220 也可包含其它或不同的信息。

例如，假定从记录组件 210 接收的信息指示用户点击了（或以其它方式调用）“OK 按钮”。然后，文本生成组件 230 在文本数据存储 220 中搜索匹配这种类型的动作的条目。一旦找到匹配，则文本生成组件 230 从文本数据存储 220 中的该条

目检索描述这种类型的动作的文本。该文本例如可仅叙述“点击 OK”。

在任何情况中，文本生成组件 230 示例性地自动生成描述在用户界面 205 上采取的并由记录组件 210 记录的用户动作的文本。这由图 3 中的框 422 指示。

所生成的文本由图 2 中的框 235 指示。在一个示例性实施例中，由记录组件 210 记录的图像被自动嵌入在所生成的文本 235 中，或至少与所生成的文本 235 相关联，使得它们能够在稍后的处理中彼此结合地被回调和显示。

接着，由记录组件 210 记录的图像数据和自动生成的文本被提供给编辑器组件 240。由记录组件 210 记录的图像和自动生成的文本示例性地被显示在编辑器 240 处的显示屏上，使得创作者 201 可生成对应于这些图像的文本。图像和由生成组件 230 生成的文本的显示由图 3 中的框 424 指示。

创作者然后可修改自动生成的文本或生成新文本。这由图 3 中的框 436 指示。最后，根据期望的格式或模式输出或保存最终文本 245。最终文本 245 可任选地包括嵌入其中的所捕捉的图像。这由图 3 中的框 438 指示。

为了设置系统来自动生成文本，创作者通过首先打开类似于图 5 中所示的窗口来激活记录组件 210。这里，创作者可编辑由线 450 和 452 指示的屏幕的一部分，以包含诸如正被创建的文档的标题的信息以及关于该任务的任何介绍性的信息。然而，这种信息可在稍后的编辑期间被添加到文本文档。

当系统开始记录图像时，UI 自动化组件 214 确定用户界面 205 上的可用用户界面元素。而且记录组件 210 使用从 UI 自动化组件 214 提供的信息在用户界面上提供关于哪个控件当前被标识为正被访问的控件的指示。对控件的高亮由图 6 中的参考标号 454 示出。

图 7 是根据本发明的一个实施例，更详细示出被执行以便为所接收的已记录的步骤生成文本的步骤的流程图。框 460 示出，与所记录的步骤相关的信息在文本生成组件 230 处从记录组件 210 接收。

一旦与所记录的步骤相关的信息由文本生成器组件 230 接收之后，文本生成器组件 230 就进而在文本数据库 220 中搜索匹配所接收的命令的条目。这由框 462 指示。在一个实施例中，文本数据库 220 是包含多个条目的 XML 数据库，其中多个条目包括控件或与之交互的其它项的类型、动作类型、控件的新状态（例如，被选中、取消选中、展开、折叠等）以及动作的相应文本。然而，可使用其它数据存储方法来保存文本。此外，其它信息可被保存在文本数据库 220 中。以下在表 1 中提供根据一个实施例的文本数据库的一部分的示例。

表 1

```

<EnglishTemplate actionTypeID="value"
controlTypeID="check box" ActionText="Select"
specialValueID="checked">
  <Sentence>Select the <tag id="1"></tag>
checkbox</Sentence></EnglishTemplate>

<EnglishTemplate actionTypeID="invoke"
controlTypeID="button" ActionText="Click">
  <Sentence>Click <tag
id="1"></tag></Sentence></EnglishTemplate>

<EnglishTemplate actionTypeID="invoke"
controlTypeID="list item" ActionText="Double-click">
  <Sentence>In the <tag id="2"></tag> list, double-
click <tag id="1"></tag></Sentence></EnglishTemplate>

<EnglishTemplate actionTypeID="expand_collapse"
controlTypeID="tree item" ActionText="Expand"
specialValueID="expanded">
  <Sentence>Click the minus sign next to <tag
id="1"></tag> to collapse
it</Sentence></EnglishTemplate>

```

例如，假定从记录组件接收到的命令的信息是：动作类型="调用"（action type = "invoke"），控件类型="按钮"（control type = "button"），控件名="点击 OK"（control name = "click OK"），则文本生成组件 230 在文本数据库 220 中搜索并找到匹配该信息的条目。然后它从该条目中检索对应于"点击 OK"（click OK）的文本。在框 464 处示出对与所匹配的条目相关联的文本的获取。

在记录所注意的步骤期间，如果创作者通过选择图 4 中所示的用户界面 402 上的用户动作（action）按钮 408 来指定该步骤为用户动作步骤，或者如果文本数

数据库 220 中的条目指示该动作是用户动作,则文本生成组件 230 在一个实施例中可提示创作者向所获取的文本添加对该动作的描述。这在框 466 处示出。创作者然后可在框 468 处提供所需文本。然而,创作者可忽视该步骤并在稍后编辑阶段期间添加信息。任何添加的文本在框 470 处被添加到文本 235。

如果不需要任何用户动作,或已经由创作者提供了必要的用户动作信息,则文本生成器 230 将新获取的文本添加到文本文档。这在框 472 处示出。

在图 8 中由参考标号 474-486 示出了所生成的文本的示例。该文本向用户提供用于所期望的任务的逐步指令。该文本可如参考文本数据库 220 和文本生成组件 230 所描述的或根据允许从所接收的输入命令中自动生成文本的任何方法来生成。自动生成的文本以及所捕捉的图像然后可呈现给创作者用于编辑。

图 9 是示出所捕捉的图像和自动生成的文本两者均可在编辑器 240 处向创作者 201 显示的一种方式的屏幕截图。图 9 包括与本发明相关的两个主要部分。第一部分包括文本显示部分 426,第二部分包括图像显示部分 428。文本显示部分 426 显示可任选自动生成的文本 235 或由创作者 201 创作的文本。图像显示部分 428 显示由记录组件 210 记录的所捕捉的屏幕截图。

在自动生成文本的实施例中,该文本在显示部分 426 中在 235 处示出,它可由创作者使用编辑器组件 240 来编辑(其一个屏幕截图在图 9 中示出)。因此,创作者可高亮显示部分 426 上显示的动作或其它项中的每一个。与高亮项相关联的图像在显示部分 428 中显示。因此,在图 9 中所示的实施例中,创作者高亮了对应于启动 Accessibility Options (可访问性选项)控件的动作编号 2,相应的图像在显示部分 428 中显示。用户然后可按需输入文本或修改文本,以便获得对由用户在用户界面 205 处执行的步骤的完全描述。在图 9 中所示的实施例中,用户启动了用户界面 205 上的 Accessibility Option 控件 430。可见,记录组件 210 不仅获取了对应于 Accessibility Options 控件的控制框 430,还获取了包含围绕 Accessibility Option 按钮 430 的多个像素的更大的上下文框 432。上下文框 432 显示了围绕 Accessibility Options 控件 430 的额外的上下文,使得它可在屏幕上更容易地被定位。

图 9 也示出了记录组件 210 已经获取了包含 Accessibility Options 控件 430 的父窗口 434。从而,创作者可精确地看见哪个控件由用户在用户界面 205 上操纵,以及该控件驻留在其父窗口上的何处。这极大地协助了创作者生成或修改描述用户界面上所采取的动作的文本。

可以理解,可按照各种不同方式使用如这里所述的本发明。例如,本发明可

用于生成内容（诸如，帮助内容），其中所捕捉的图像嵌入在文本或帮助文本内。对此的一个示例在图 10 中示出。可见，处理中的每一步不仅包含描述做什么来完成该步骤的文本，还包含指示将在何处执行该步骤的视觉图像。在此实施例中，本发明被用于在文本中生成嵌入的图像，最终文本 245 与所嵌入的图像一起输出。这由图 4 的可任选框 438 指示。

然而，本发明不必用于将图像嵌入到文本中。相反，本发明可仅用于向创作者显示所捕捉的图像，其中创作者正生成对所采取的并对其捕捉图像的步骤的书面描述。例如，创作者可能正生成描述修理病毒或将其从计算机移除的处理的文本。创作者可能不希望在该文本中包括嵌入的图像。在这种情况下，仅显示图像，可任选地，也向创作者显示自动生成的文本，创作者可从草稿中生成文本，或者，在提供了自动生成的文本的实施例中，创作者也可修改自动生成的文本。

在任何情况中，最终文本 245 将示例性地以最终发布内容的任何模式来创建，或被转换成该模式。这种模式可以是，例如，Microsoft Assistance 标记语言（MAML）、HTML 或某种其它的 XML 格式。

图 11 是用于生成描述用户为在两个不同的系统上（或具有两种不同配置的系统上）执行一任务所采取的步骤的内容（诸如，最终文本 245）的系统 500 的框图。如发明背景中所述的，现有的用于生成内容以描述用户为执行整体任务而采取的步骤的系统遭受如下的事实：执行任何给定任务所必须采取的步骤可能会基于执行任务的计算机的配置而改变。

从而，即使捕捉了指示要执行的步骤的图像，且即使自动或手动生成了描述这些步骤的文本，但基于由计算机运行的特定操作系统或基于计算机系统配置中的任何其它差异，步骤仍可能改变。事实上，如果操作系统的版本号或由计算机运行的其它软件不同，所需步骤甚至也可能不同。当然，配置中的大量其它差异也可使执行给定任务所需的步骤不同。例如，对于机器的不同状态，用户界面可有所不同。作为示例，取决于计算机是位于网络域上还是位于工作组上，连接至远程机器的用户界面步骤的顺序可能有所不同。

从而，为了自动记录执行给定任务所需的步骤，图 2 中所示的记录系统应记录以该内容旨在供使用的以所有不同配置来配置的机器上的步骤。记录在所有这些不同配置上的所记录的用于执行任务的步骤随后应当被组合成一个描述如何在不同配置下执行该任务的整体内容集。可任选地，最终向用户显示该内容的系统将检测用户系统的配置，并仅显示相关的内容（例如，仅与最终用户相关的那些步骤）。

因此，系统 500 示出第一计算机 502 和第二计算机 504，它们各自包括诸如图 2 中所示的记录系统。计算机 502 示例性地根据此处指定为配置 A 的第一配置来配置。另一方面，计算机 504 根据此处指定为配置 B 的第二配置来配置。

配置 A 和 B 可以是影响执行任何给定任务所需的步骤的任何类型的不同配置，这些步骤将被存储并对此创作内容。从而，作为示例，配置 A 可以是位于网络域上的计算机，而配置 B 是位于工作组上的计算机。类似地，配置 A 可以是运行在第一操作系统上的计算机，而配置 B 是运行在第二操作系统上的计算机。类似地，配置 A 可以是操作在操作系统的版本上的计算机，而配置 B 是操作在同一操作系统的不同版本上的计算机。除此之外，按照内容创作者的期望也可使用其它不同的配置。当然，也应注意，计算机 502 和 504 实际上可以是不同的计算机或安装在虚拟机上的计算机。

图 11 中所示的记录系统 200 可类似于图 2 中所示的记录系统，且相应地对其标号。然而，如将在以下更详细描述，在仅需要一个自动文本生成部分的情况下，自动文本生成部分可被部署在创作计算机 506 上。

计算机 502 和 504 示例性地被连接至创作计算机 506。创作计算机 506 示例性地包括导入器机制 508 和内容编辑器组件（诸如参考图 2 描述的编辑器组件 240）。

创作计算机 506 示例性地可以是计算机 502 或 504 中的任一个，或者是不同的计算机。类似地，计算机 506 可运行在与计算机 502 和 504 之一相同的操作系统上，或运行在不同的操作系统上。此外，如上所述，编辑器组件 240 可配备自动文本生成系统，或它可仅仅是参考图 2 所述的编辑器组件 240。

操作中，计算机 502 和 504 上的记录系统 200 允许用户在与计算机 502 和 504 中的每一个相关联的用户界面上执行任务，而记录系统 200 记录并存储被采取以执行所需任务的用户界面步骤。记录系统 200 因此如上所述示例性地记录对应于用户所采取的动作的屏幕截图，也可任选地自动生成对应于这些图像的文本。

记录系统 200 然后示例性地将所记录的步骤格式化可由导入器组件 508 接受的格式。例如，计算机 502 上的记录系统 200 将示例性地记录用户在计算机 502 的用户界面上为执行给定任务而进行的一连串步骤，并根据一 XML 格式将所记录的这些步骤格式化所记录的片段 A（也由标号 510 指定）。类似地，计算机 504 上的记录系统 200 将示例性地记录步骤，并根据一 XML 格式将它们格式化所记录的片段 B（也由标号 512 指定）。

所记录的片段 510 和 512 然后由计算机 502 和 504 提供给导入器组件 508。导

入器组件 508 被示为位于计算机 506 上，但当然可以理解，导入器组件 508 可以是使功能分布在系统 500 中所示的各个计算机之中，或与所有这些计算机分离驻留，或安置在除计算机 506 以外的计算机之一上的组件。在任何情况中，导入器组件 508 可以是用于接收所记录的片段 510 和 512 并将其提供给编辑器组件 240 以便向用户呈现的任何合适的导入机制。

导入器组件 508 示例性地在记录计算机 506 上存储所记录的片段 510 和 512，并以各种不同方式之一将它们导入到编辑器组件 240 中。例如，这可通过经由共享目录导入片段 510 和 512 来完成。类似地，这可通过将所记录的片段 510 和 512 存储到磁盘或另一外部存储器设备上来完成，其中该外部存储器设备物理上被运送至计算机 506，在那里片段 510 和 512 被加载到编辑器组件 240。类似地，可远程使用计算机 506 来操作计算机 502 和 504，然后计算机 506 可将所记录的片段 510 和 512 存储在可由计算机 506 访问的剪贴板或其它共享的应用程序空间上。这样的共享空间的一个实施例是由华盛顿州雷德蒙市的微软公司提供的 Windows Clipboard 系统。

在使用共享应用程序空间来提供片段 510 和 512 的实施例中，计算机 506 从共享的应用程序空间中读取片段 510 和 512，并将其导入到编辑器组件 240 内。例如，在使用 Windows Clipboard 应用程序空间的实施例中，一种同样由华盛顿州雷德蒙市的微软公司提供的被称为远程桌面（Remote Desktop）的程序允许在远程与本地计算机之间（诸如在计算机 502 和 504 与计算机 506 之间）共享剪贴板空间。

在任何情况中，导入器组件 508 用于将所记录的片段 510 和 512 导入至编辑器组件 240 内。在计算机 502 和 504 上的记录系统 200 包括自动文本生成系统 270 的实施例中，所记录的片段 510 和 512 将包括自动生成的文本。或者，文本可在编辑器组件 240 处自动或手动生成。

编辑器组件 240 向用户显示片段 510 和 512，并因此允许由用户插入适当的条件。例如，图 11 中所示的最终文本 245 的实施例示出了：当用户已经插入了条件语句 514 和 516 之后，所记录的片段 510 和所记录的片段 512 被插入整体文本 245 中。条件语句 514 例如是“条件: If 配置 A, then:”。这指示，如果该计算机以配置 A 配置，则将向用户显示或自动执行由所记录的片段 510 指示的步骤。

条件 516 陈述“条件: If 配置 B, then:”。这指示，如果该计算机根据配置 B 配置，则将向用户显示或执行由所记录的片段 512 所指示的步骤。

当然，也可理解，创作者也可按照任何期望的方式删除或修改文本。这可通

过在编辑器组件 210 上使用常规编辑技术来完成。

表 2 示出了由来自两个不同配置的记录形成的示例性 XML 文档。这两个 XML 片段被记录在其一运行在网络域上另一运行在网络工作组上的计算机上。注意，`condition="Network_connection: domain"`属性被标记以处理用于判断运行哪一片段的 if-then 逻辑。

表 2

```
<task>
  <title>Add a new user to the computer</title>
  <introduction>When you add a user to your computer, you are allowing that
individual to have access to files and programs on your computer. </introduction>
  <commandStep>
    <command>control.exe userpasswords</command>
    <description>Open User Accounts in Control Panel</description>
  </commandStep>

  <!-- *** 这部分是位于网络域上的机器上的记录器 *** --/>
  <stepGroup condition="Network_connection:domain">
    <uiActionsStep><actionRef id="0"/><actionRef id="1"/>
      <description>On the <ui>Users</ui> tab, click Add</description>
    </uiActionsStep>
    <uiActionsStep><actionRef id="2"/>
      <description>Follow the instructions in the wizard to add a new
user</description>
    </uiActionsStep>
  </stepGroup>

  <!-- *** 这部分是位于工作组上的机器上的记录器 *** -/>
  <stepGroup condition="Network_connection:workgroup">
    <uiActionsStep><actionRef id="3"/>
      <description>Click <ui>Create a new account</ui></description>
    </uiActionsStep>
```

```
<uiActionsStep><actionRef id="4"/>
    <description>Type a name for the new user account</description>
</uiActionsStep>
<uiActionsStep><actionRef id="5"/>
    <description>Click Next</description>
</uiActionsStep>
<uiActionsStep><actionRef id="6"/>
    <description>Click either Computer Administrator or Limited
depending on the type of account you want to assign the new
user</description>
</uiActionsStep>
<uiActionsStep><actionRef id="7"/>
    <description>Click create</description>
</uiActionsStep>
</stepGroup>
</task>
```

因此，可见本发明允许创作者在不同配置下的不同机器上记录所有或部分宏（或执行给定任务的另一连串步骤），然后在一个创作机器上组装所记录的片段以得到最终内容。这允许宏被记录在不同机器上，但主创作步骤可在一个位置处执行。这使得内容的创作相比现有系统而言较不昂贵和复杂，并更高度自动化。

尽管参考特定实施例描述了本发明，但本领域的技术人员将认识到，可对形式和细节进行改变，而不背离本发明的精神和范围。

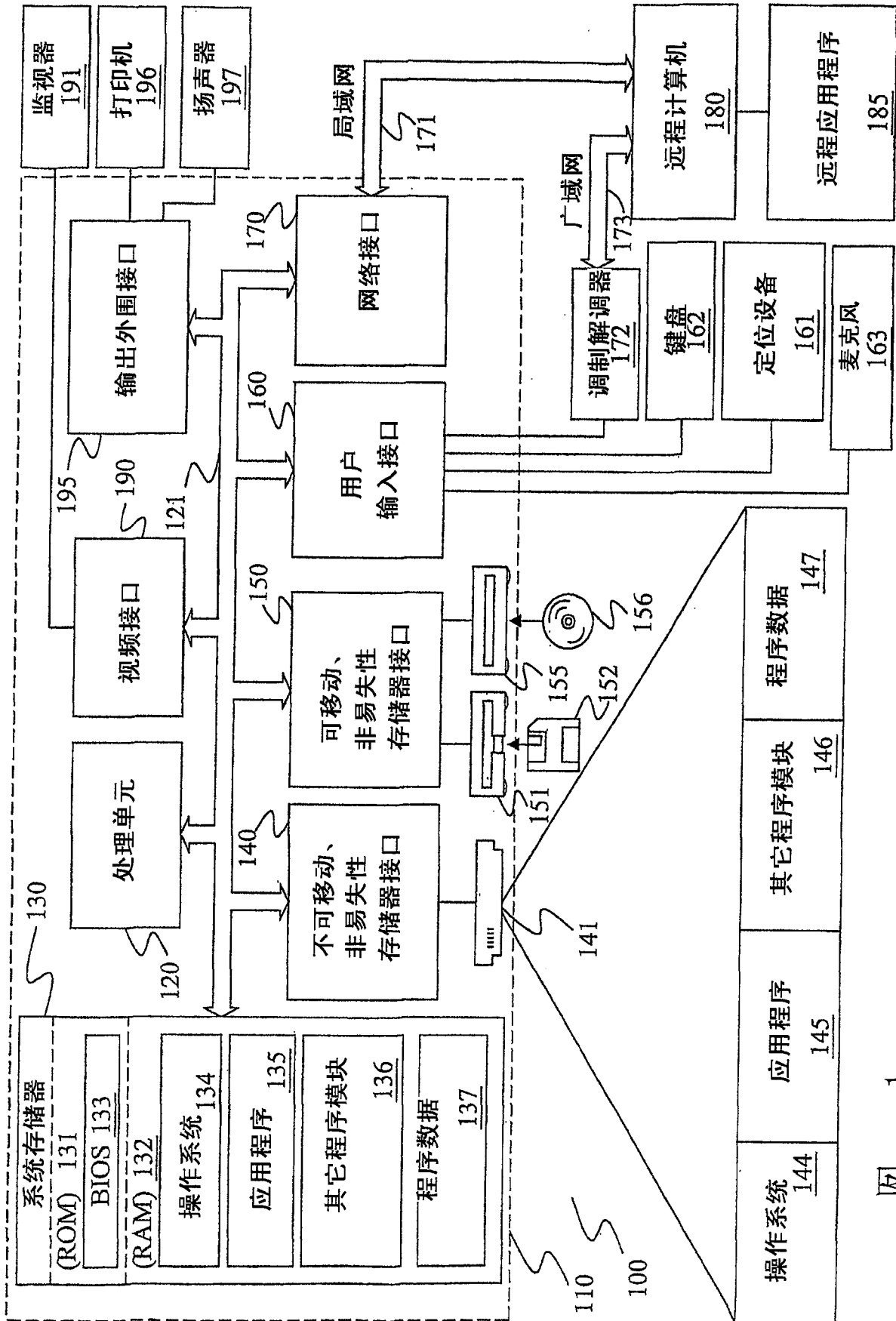


图 1

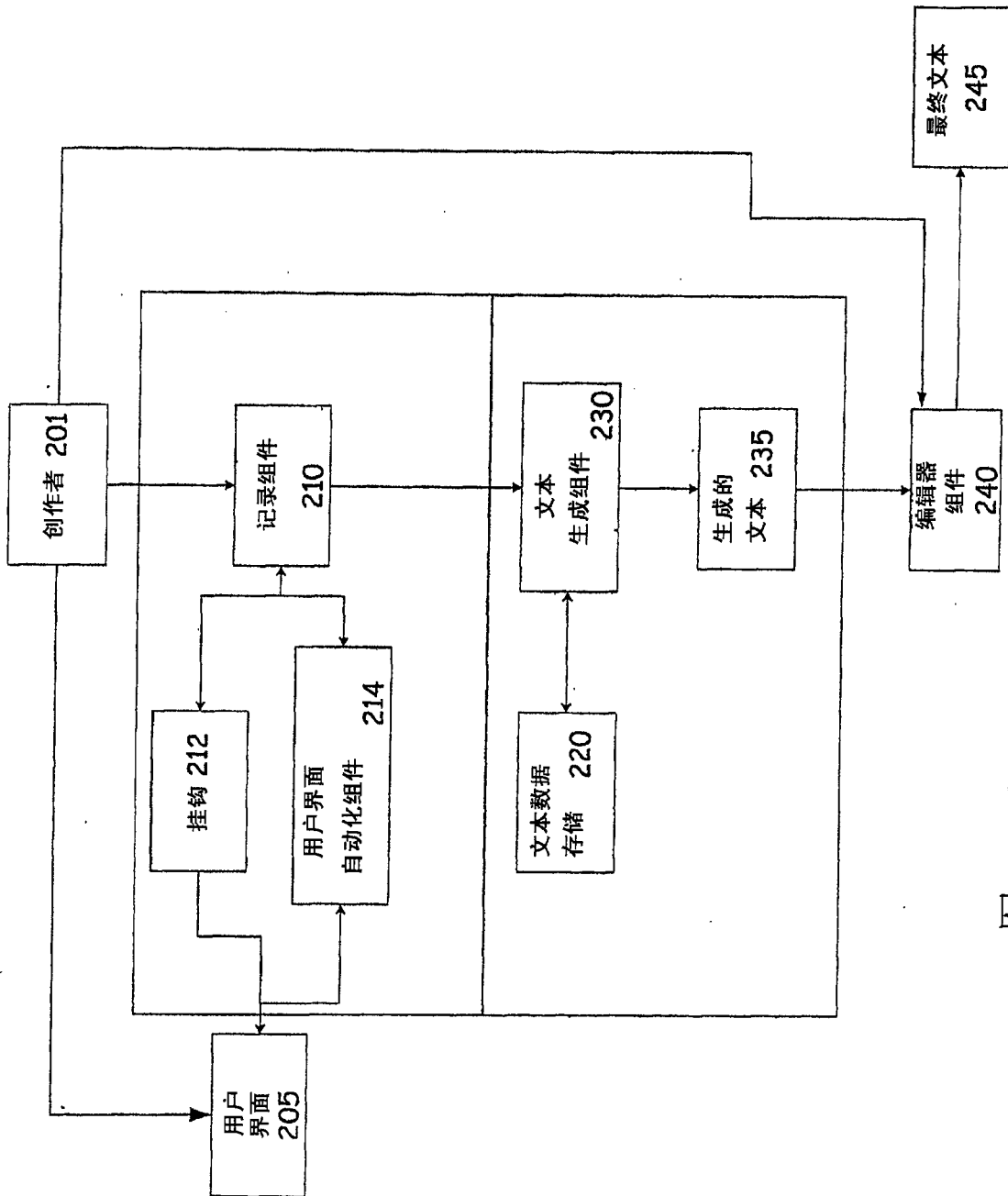


图 2

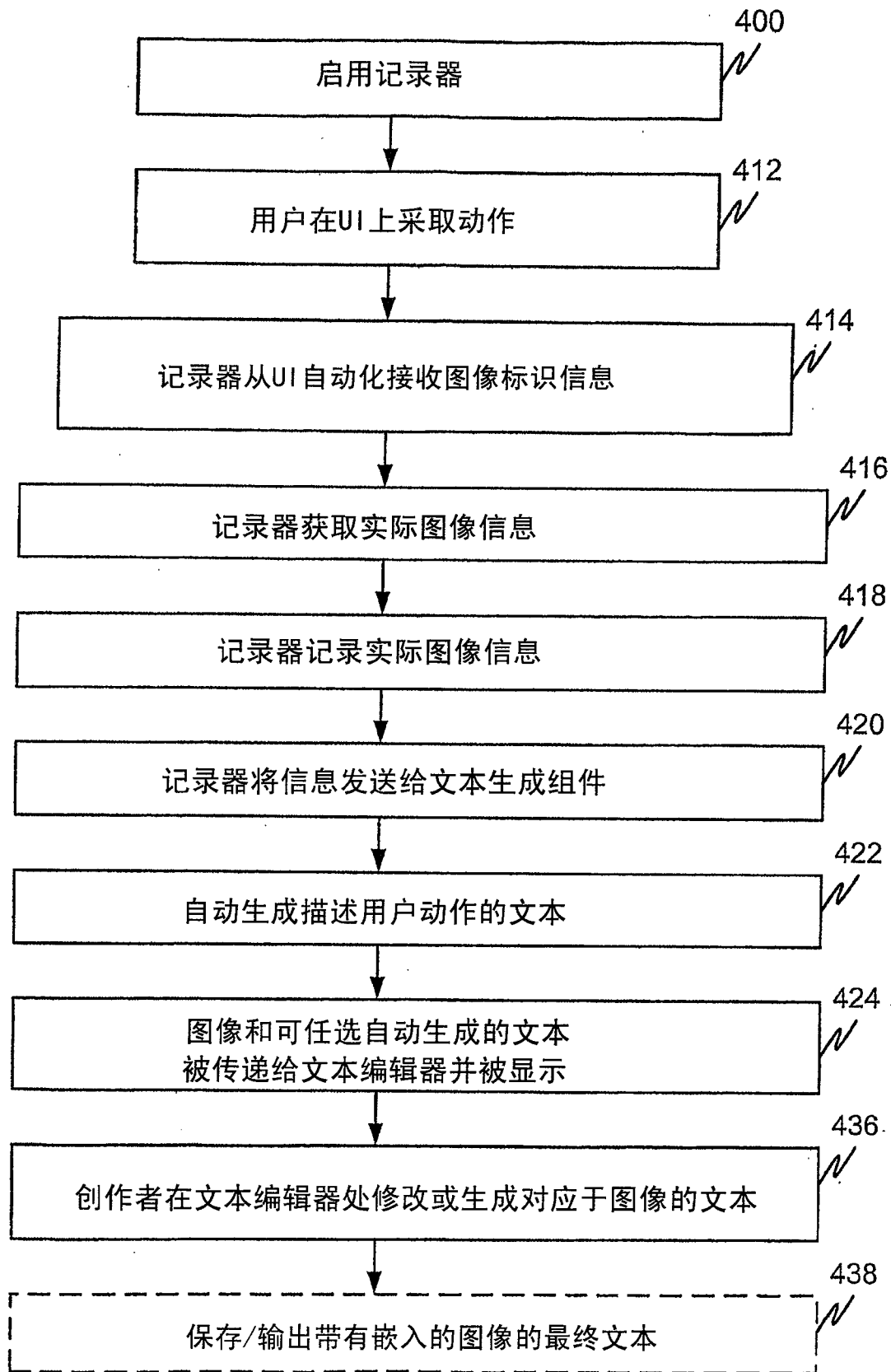
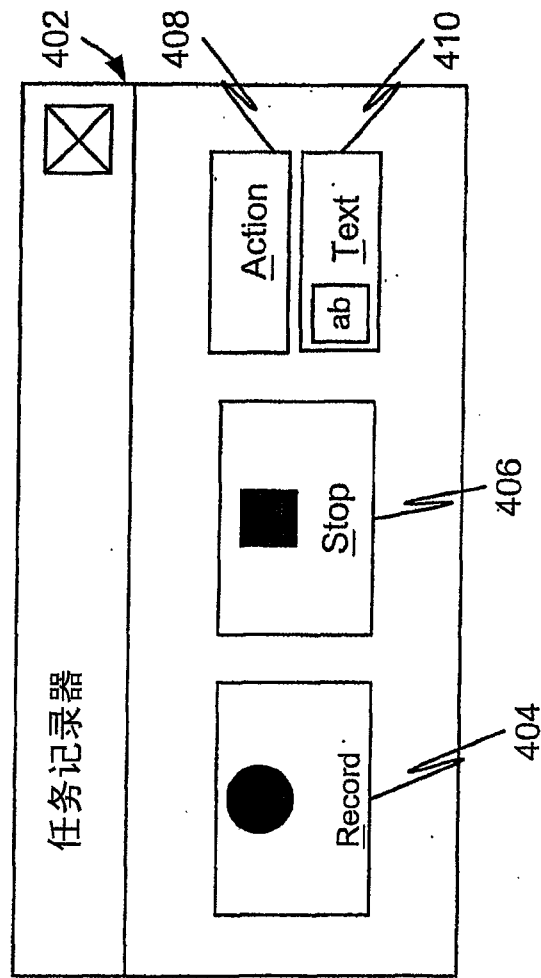
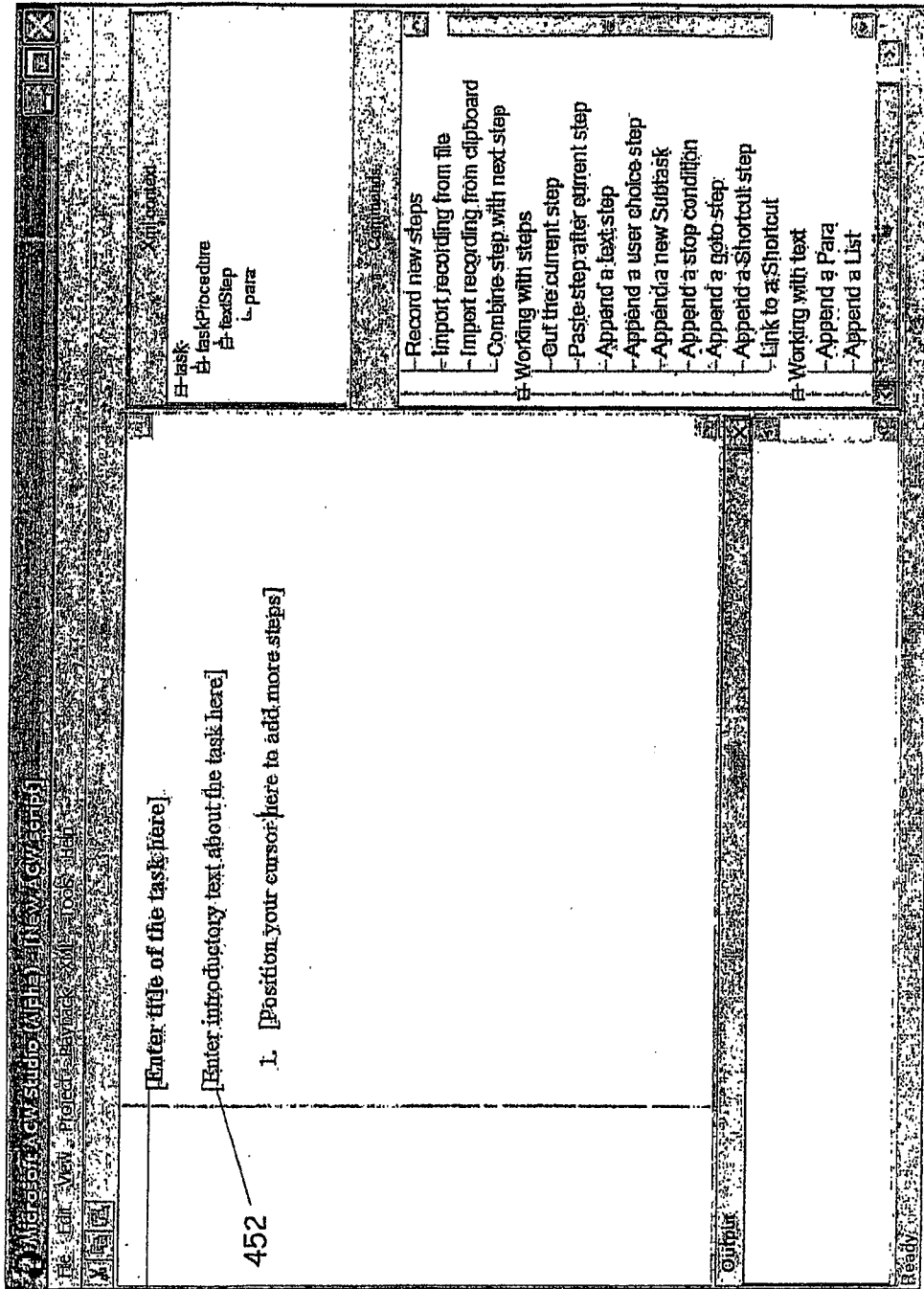


图 3



4



454

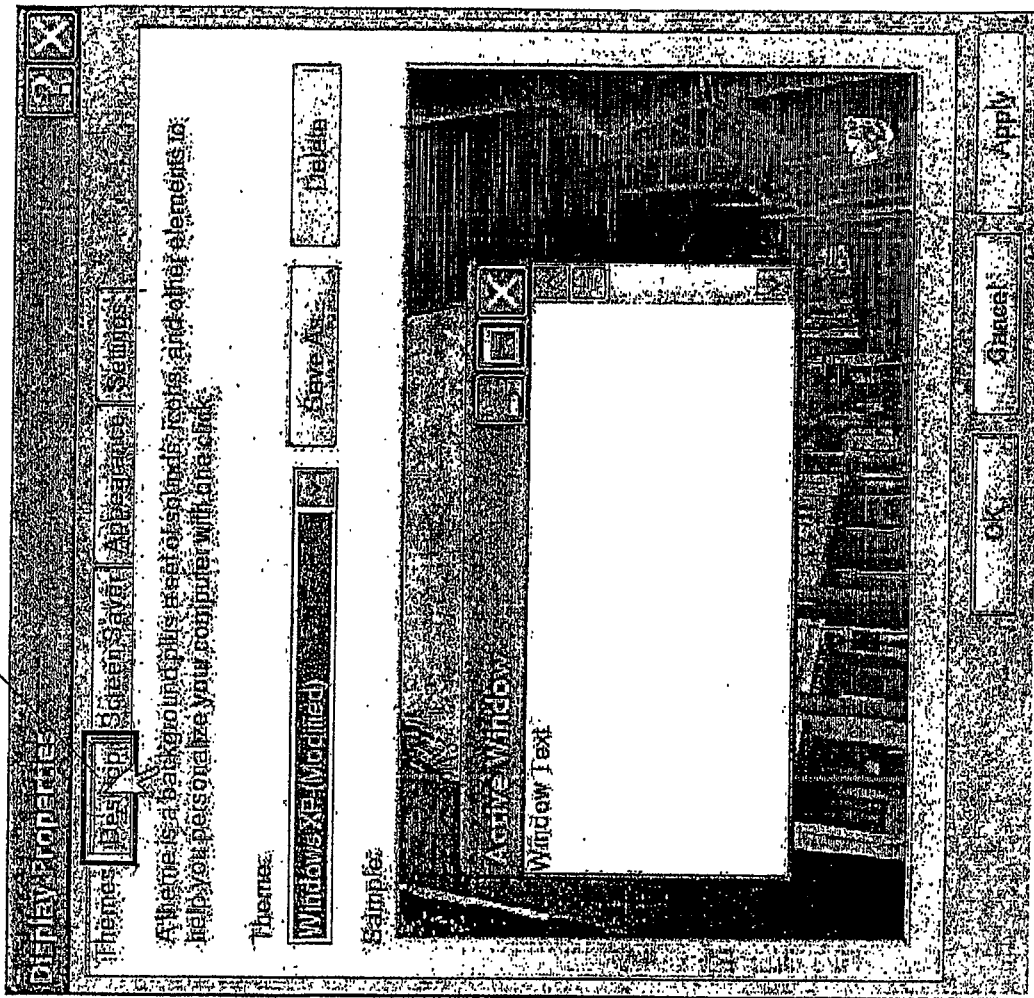


图 6

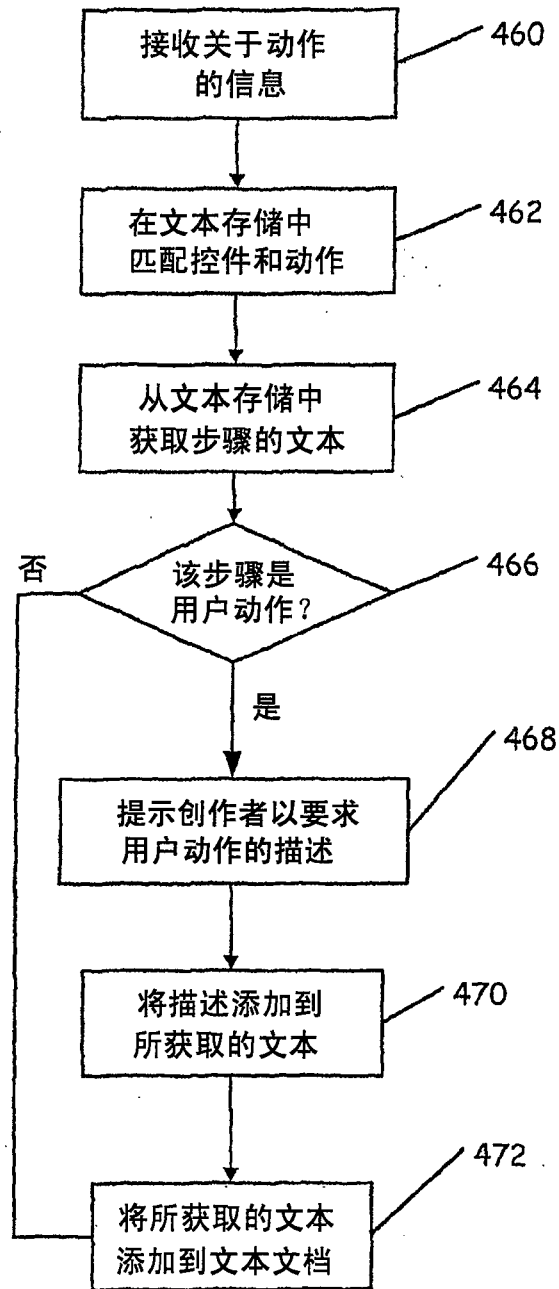
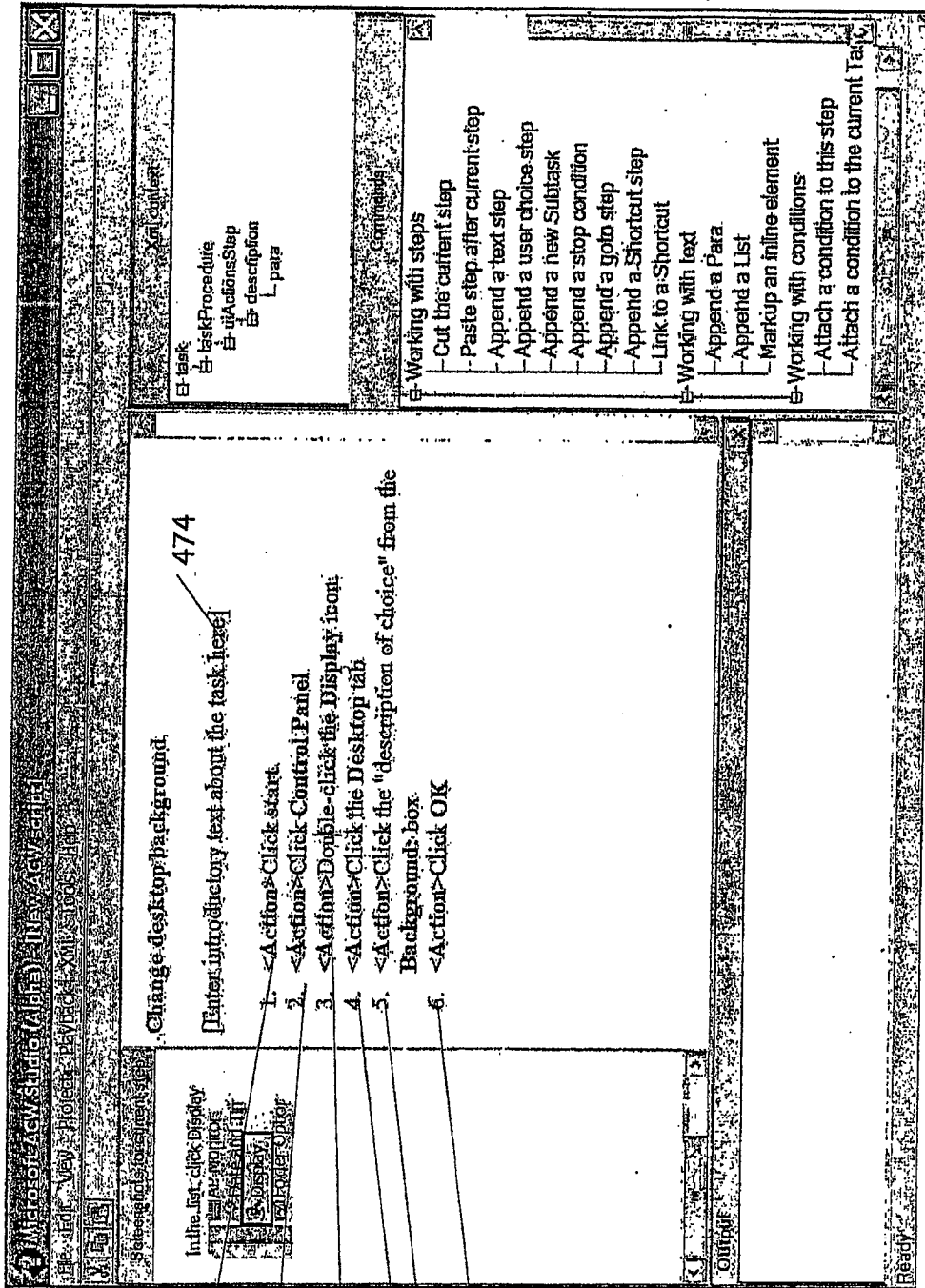


图 7



8

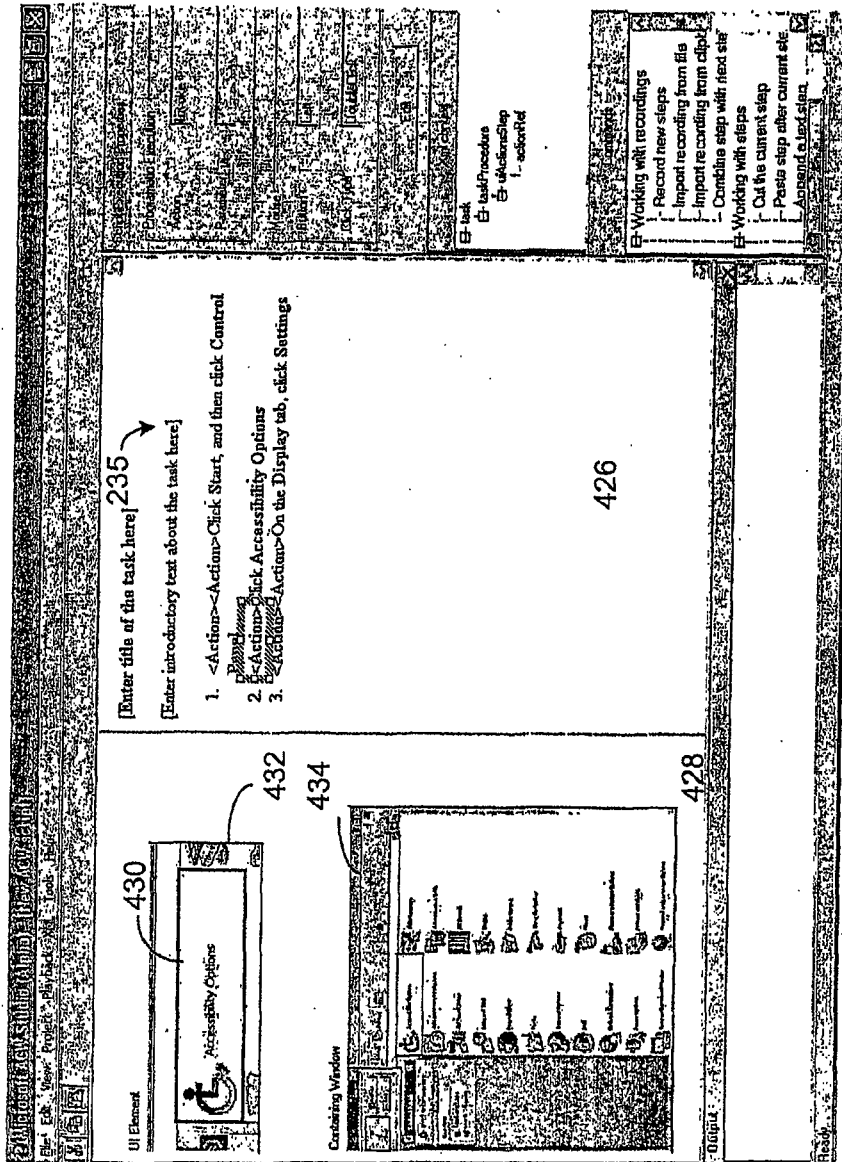


图 9

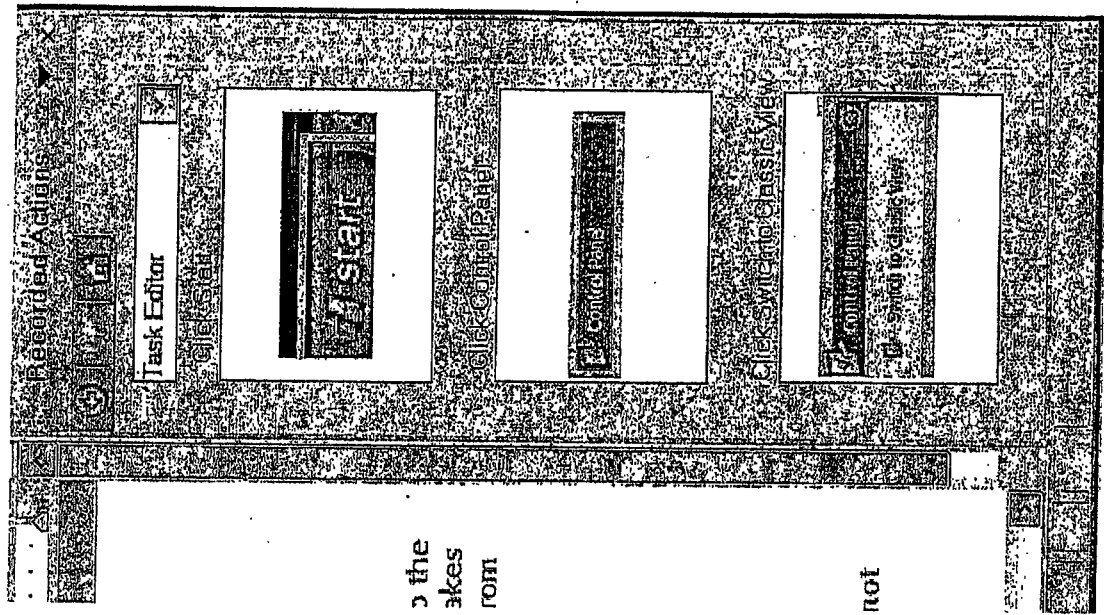


图 10

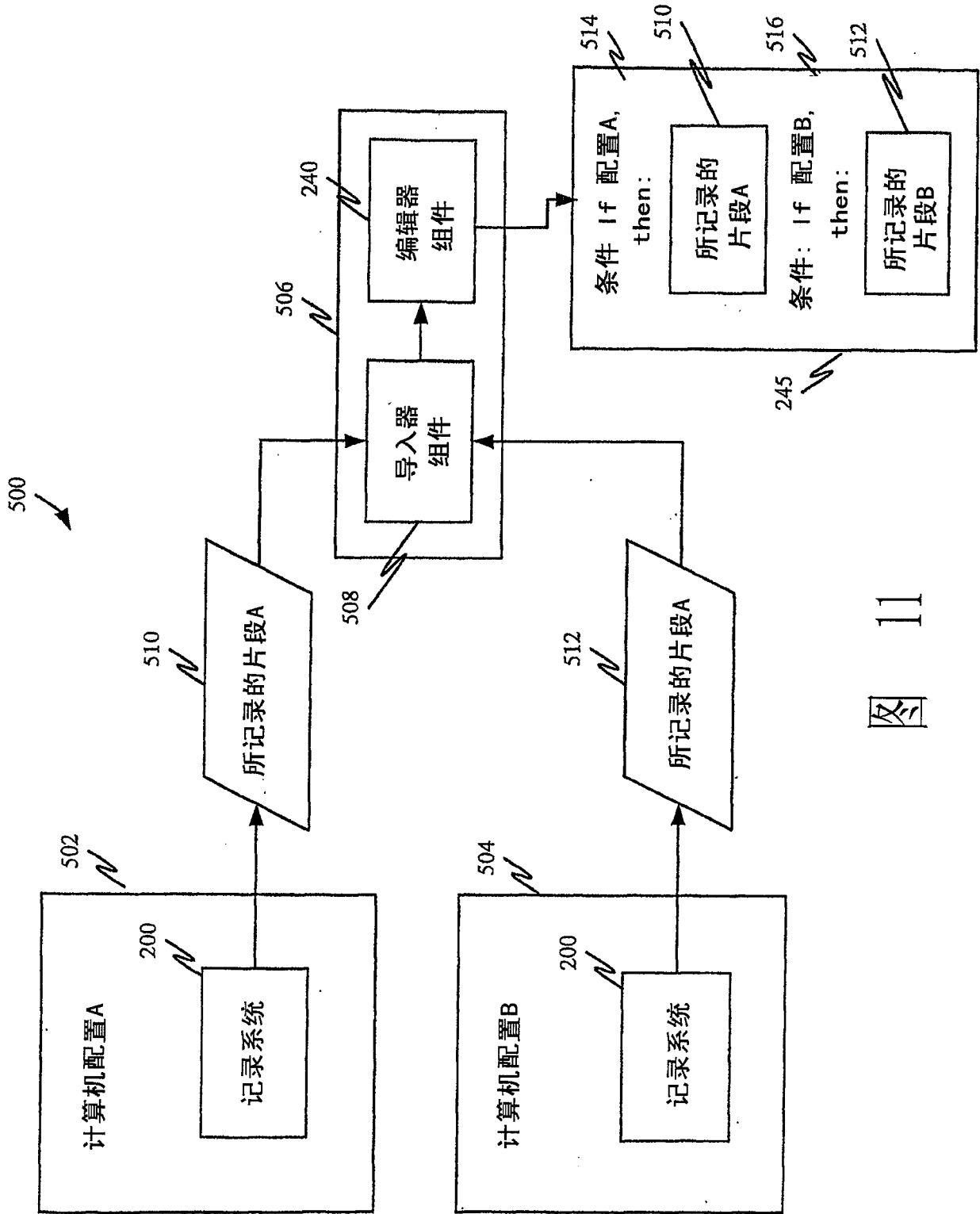


图 11