



(12) 发明专利

(10) 授权公告号 CN 103052934 B

(45) 授权公告日 2016. 05. 18

(21) 申请号 201180038293. 7

G06F 9/44(2006. 01)

(22) 申请日 2011. 06. 02

(56) 对比文件

(30) 优先权数据

61/350, 867 2010. 06. 02 US

61/410, 033 2010. 11. 04 US

CN 101063936 A, 2007. 10. 31,

US 2008307400 A1, 2008. 12. 11,

US 2004036719 A1, 2004. 02. 26,

CN 1125488 A, 1996. 06. 26,

(85) PCT国际申请进入国家阶段日

2013. 02. 04

审查员 傅重添

(86) PCT国际申请的申请数据

PCT/US2011/038932 2011. 06. 02

(87) PCT国际申请的公布数据

W02011/153352 EN 2011. 12. 08

(73) 专利权人 艾伦智能科技公司

地址 美国明尼苏达州

(72) 发明人 M·W·艾伦 S·M·伯思

J·L·科伊尔

(74) 专利代理机构 上海一平知识产权代理有限公司

公司 31266

代理人 须一平

(51) Int. Cl.

G06F 3/0481(2013. 01)

G06F 3/14(2006. 01)

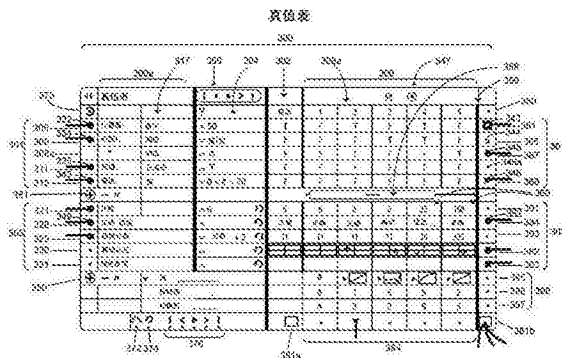
权利要求书2页 说明书24页 附图13页

(54) 发明名称

逻辑表

(57) 摘要

本发明公开的逻辑表包括真值表、回答表和状态表。上述逻辑表使用优雅的、可视工具组与一致的图形用户界面而能整合条件逻辑、复杂情况判断和状态处理。用户能在对象间建立逻辑交互，否则需要使用复杂的计算机编程语言。



1. 一种使用逻辑表作为图形用户界面 (GUI) 系统的程序对象来将多个程序对象相关联的方法, 该图形用户界面系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施, 该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上, 所述图形用户界面系统被显示于该图像显示器上, 所述方法包括:

通过所述结合编程与运行的实施程序在所述图形用户界面系统中显示第一程序对象、第二程序对象和该逻辑表, 该结合编程与运行的实施程序访问来自存储于所述数据库的对象库的该第一程序对象、该第二程序对象和该逻辑表, 调整多媒体计算机应用程序和存储经调整的多媒体计算机应用程序于该数据库中;

通过所述结合编程与运行的实施程序链接作为输入的所述第一程序对象的属性至所述逻辑表, 该结合编程与运行的实施程序进一步调整所述多媒体计算机应用程序和存储经进一步调整的多媒体计算机应用程序于所述数据库中;

通过所述结合编程与运行的实施程序链接所述第一程序对象的所述属性至所述逻辑表的属性状态, 该结合编程与运行的实施程序进一步调整所述多媒体计算机应用程序和存储经进一步调整的多媒体计算机应用程序于所述数据库中, 所述属性状态包括值语句, 所述第一程序对象的所述属性链接于该值语句, 及所述属性状态链接于作为输出的所述逻辑表的数值属性;

或者 (1) 通过所述结合编程与运行的实施程序链接作为所述输出的所述逻辑表的所述数值属性至所述第二程序对象的属性, 或者 (2) 通过所述结合编程与运行的实施程序链接所述第二程序对象的该数值属性至该输出, 所述结合编程与运行的实施程序进一步调整所述多媒体计算机应用程序和存储经进一步调整的多媒体计算机应用程序于所述数据库中, 从而通过该逻辑表链接所述第一程序对象的属性和所述第二程序对象的属性。

2. 根据权利要求 1 所述的方法, 其特征在于, 所述值语句是真或假语句。

3. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 所述值语句是字母数字的语句。

4. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 还包括基于求值表达式求所述第一程序对象的所述属性的值, 和基于该求值链接该第一程序对象的该属性至所述值语句。

5. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 不是所述第一程序对象的程序对象作为所述输入链接, 所述属性状态包括多个值语句, 其中所述值语句中的一个链接于不是该第一程序对象、作为该输入链接的该程序对象的属性, 并且求值表达式引用不是该第一程序对象、作为该输入链接的该程序对象的该属性, 而所述求值是基于对不是该第一程序对象、作为该输入链接的该程序对象的该属性的引用。

6. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 所述逻辑表包括用于链接所述程序对象的所述属性的输入端口和输出端口。

7. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 所述逻辑表包括用于链接所述值语句的值属性端口。

8. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 所述逻辑表包括用于集合多个链接的输出配线区。

9. 根据权利要求 1 和 2 中任一项所述的方法, 其特征在于, 所述逻辑表包括播放控制,

其中该播放控制允许用户从一个属性状态移动到另一个属性状态。

10. 根据权利要求 1 和 2 中任一项所述的方法,其特征在于,所述逻辑表包括间栏以操控一个属性状态到另一个属性状态的转换。

## 逻辑表

### 技术领域

[0001] 本发明主要涉及用于开发多媒体计算机应用程序的程序产品,特别涉及用于允许计算机装置使用逻辑表作为程序对象来链接程序对象的程序产品。

### 背景技术

[0002] 通常,判定表包含判定列表和这些判定所基于的条件。在计算机编程中,判定表用于根据不同情况下作出的判定来引导流程。判定表可能很复杂,特别是在涉及许多判定以解释所有可能情况时。普遍被接受的是,尽管可视编程环境是用户友好的,但这些环境往往不能满足交互媒体的建立者建立交互的需要,其依赖于复杂的判定,如那些涉及判定表的判定。因此,解决方案倾向于“脚本”功能,在其中建立者需使用编程。

### 发明内容

[0003] 描述了能整合条件逻辑、复杂情况判断和状态处理的可视工具组和方法。

[0004] 在本发明的一个方面中,逻辑表作为程序对象在图形用户界面(GUI)系统内被建立。使用逻辑表作为程序对象允许对对象和属性的复杂求值和操控,从而用户可使用具有高度复杂性的判定表建立应用程序,而不必学习复杂的计算机编程语言,如 Java, C++ 等。

[0005] 公开的逻辑表包括真值表、回答表和状态表。这些逻辑表使用优雅的、可视工具组与一致的图形用户界面以能整合条件逻辑、复杂情况判断和状态处理。该用户能在对象间建立逻辑交互,否则需要使用复杂计算机编程语言。

### 附图说明

[0006] 图 1 是用于提供用于建立多媒体计算机应用程序的结合编程与运行的实施程序的系统的概览示意图。

[0007] 图 2A 是通过网络浏览器,提供用于建立多媒体计算机应用程序的结合编程与运行的实施程序给用户的流程的流程图。

[0008] 图 2B 是通过存储在用户终端上的结合编程和运行的软件应用程序,提供用于建立多媒体计算机应用程序的结合编程与运行的实施程序给用户的流程的流程图。

[0009] 图 3 示出了真值表作为逻辑对象的一实施例。

[0010] 图 4A-4C 是公开的方法的一实施例的流程图。

[0011] 图 5 示出了 GUI 实施真值表作为逻辑对象的一实施例。

[0012] 图 6A-B 示出了回答表作为逻辑对象的一实施例。

[0013] 图 7 示出了 GUI 实施回答表作为逻辑对象的一实施例。

[0014] 图 8A-F 示出了状态表作为逻辑对象的一实施例。

[0015] 图 9 示出了 GUI 实施状态表作为逻辑对象的一实施例。

[0016] 图 10 是可用于实施本发明描述的系统及方法的计算装置的一例子。

[0017] 各附图中的类似参考标记指示类似元件。

## 具体实施方式

[0018] 现将具体提及实施例,其例子随附图示出。在以下的详细描述中,为了提供对本发明的全面理解,阐述了许多特定细节。但是,对于本领域的普通技术人员来说,没有这些特定细节,本发明也能实施是明显的。在其他情况下,公知方法、步骤、部件、电路和网络没有具体描述以避免不必要地使实施例的方面不清楚。

[0019] 还可以理解的是,尽管术语第一、第二等在本发明中可能用于描述各种元件,但这些元件不应被这些术语限制。这些术语仅用于区分一元件与另一元件。在不偏离本发明范围情形下,例如,第一动作可称为第二动作,而类似地,第二动作可称为第一动作。

[0020] 在本发明说明中使用的术语只是出于描述特定实施例的目的,并没有限制本发明的意图。在本发明说明和随附的权利要求书中使用的单数形式“一”、“一”和“所述”也意图包括复数形式,除非上下文另外清楚地指出。还可理解的是,本发明使用的术语“和/或”涉及和包含一个或多个相关列出项目的任何和所有可能组合。进一步理解的是,术语“包含”和/或“包括”在本发明中使用时指定了所述特征、整数、步骤、操作、元件、和/或部件的存在,但不排除一个或多个其他特征、整数、步骤、操作、元件、部件、和/或其群组的存在或增加。

[0021] 在本发明中使用时,术语“如果”可根据上下文被解释为“当...时”或“根据”或“响应确定”或“响应检测”。类似地,短语“如果确定了...”或“如果[所述条件或事件]被检测到”可根据上下文被解为“根据确定”或“响应确定”或“根据检测到[所述条件或事件]”或“响应检测到[所述条件或事件]”。

[0022] 术语“多媒体计算机应用程序”(即,项目)在本发明中用于提及可执行程序,其能从计算机输出一个或多个感官刺激(即,视觉、听觉、触觉等)而允许用户执行单个或多个相关特定任务。术语“编辑”在本发明中用于提及建立新多媒体计算机应用程序、调整存在的多媒体计算机应用程序、和/或与存在的多媒体计算机应用程序交互。

[0023] 术语“程序”在本发明中通常用于提及具有机器被构造成以理解和/或遵循的形式语言。术语“调试”在本发明中用于提及检测和定位程序缺陷以修复程序缺陷的过程。术语“编程语言”在本发明中用于提及人工语言,其用于表达引导机器执行特定功能和/或动作的计算、运算和逻辑。术语“编译器”在本发明中用于提及计算机程序,其将撰写的源代码转换成可被计算机处理器执行的机器语言。术语“对象相关映射”在本发明中用于提及编程技术,其用于在面向对象编程语言中的不兼容类型系统间转换数据。

[0024] 术语“资产”在本发明中用于提及在多媒体计算机应用程序中使用的任何图片、音频、视频、文本、SWF(冲击波快闪)、XML(可扩展标记语言)或其他外部多媒体格式。术语“对象”在本发明中用于包括但不限于原始对象、容器对象、和/或交互对象。术语“原始对象”在本发明中用于描述单个对象,如线、曲线、和/或其他基本形状,如几何形状。术语“容器对象”在本发明中用于描述可包含一个或多个其他对象的对象。例子包括但不限于场区域对象和/或小配件对象。容器对象可包含任何其他对象。术语“交互对象”在本发明中用于描述被配置成与用户交互的对象,其中对象的可视表示是根据与用户的交互而可变化的。例子包括滑块对象和/或路径对象。如果原始对象的可视表示是根据与用户的交互而可变化的,具有特定属性的原始对象可以是交互对象。

[0025] 术语“工具”在本发明中用于描述选择、建立、和 / 或操作对象的功能。术语“工具箱”在本发明中用于描述显示一个或多个工具图标的可视表示。术语“工具图标”在本发明中用于描述表示工具的图形图示,其显示于显示设备的屏幕上。该工具图标可由与 GUI 交互的用户点击和 / 或选择,以激活与选择的工具图标相关联用于选择、建立、和 / 或操作对象的工具。

[0026] 术语“应用数据”在本发明中用于提及该多媒体计算机应用程序的当前状态。术语“项目数据”在本发明中用于提及多媒体计算机应用程序中使用的所有对象、对象间的所有交互、以及所有资产的系列化集合。

[0027] 术语“图形用户界面”(即, GUI)在本发明中用于提及一种用户界面,该用户界面相对于必须键入文本命令通常提供可被用户操控的可视指示器以执行动作(如,例如,选择命令、调用文件、启动程序、以及做其他常规任务)。

[0028] 术语“编程环境”在本发明中用于提及在计算机处理器上运行的软件,其允许在编程多媒体计算机应用过程中使用编程表达式(如撰写的代码、图形、图纸、动画、或图标)。术语“运行环境”在本发明中用于提及在计算机处理器上运行的软件,其允许用户运行(如,发送指令给计算机处理器以及获取计算机的内存(RAM)和其他系统资源)已执行的多媒体计算机应用程序。术语“实时合作”在本发明中用于提及在计算机处理器上运行的软件,其允许多数人使用不同计算机同时编辑多媒体计算机应用程序。术语“结合编程与运行的单个实施程序”在本发明中用于提及在计算机处理器上运行的软件,其允许用户在编程多媒体计算机应用过程中使用编程表达式(如撰写的代码、图形、图纸、动画、或图标)并同时允许用户运行(如,发送指令给计算机处理器并获取该计算机的内存(RAM)和其他系统资源)已执行的多媒体计算机应用程序。

[0029] 描述了使用逻辑表作为逻辑对象将程序对象相关联的方法。本发明定义的多媒体计算机应用程序,是针对计算机程序,该计算机程序是完全交互体验,其能执行单个或多个相关特定任务。本发明讨论的对象涉及原始数据结构、逻辑数据结构和交互数据结构,这些数据结构由数据域、方法和它们的交互组成。

[0030] 本发明定义的交互对象,涉及可由其他交互对象、原始对象和 / 或逻辑对象组成的对象。在一些例子中,交互对象可以是独立的多媒体计算机应用程序,其用于在更大的多媒体计算机应用程序内执行功能。

[0031] 本发明讨论的项目包括选自解码版本的多媒体计算应用程序、对象、资产或资源及其组合中的至少一个。

[0032] 本发明讨论的资产或资源,涉及在多媒体计算机应用程序中使用的任何图片、音频、视频、文本、SWF(冲击波快闪)、或 XML(可扩展标记语言)。本发明讨论的应用数据涉及多媒体计算机应用程序的当前状态。本发明讨论的项目数据涉及多媒体计算机应用程序中使用的所有对象、这些对象间的所有交互、以及所有资产的系列化集合

[0033] 图 1 是根据一实施例的系统 100 的概览示意图,其提供用于建立多媒体计算机应用程序、调整多媒体计算机应用程序和 / 或与多媒体计算机应用程序交互的结合编程与运行的实施程序。该系统 100 包括多个用户终端 110-A 和 110-B,后端服务器 120,和网络 130。用户终端 110-A 和 110-B 通过网络 130 与后端服务器 120 通信。

[0034] 用户终端 110-A 和 110-B 允许用户通过网络 130 建立存储于后端服务器 120 的多

媒体计算机应用程序,调整该多媒体计算机应用程序和/或与该多媒体计算机应用程序交互。用户终端 110-A 使用网络浏览器 135 通过网络 130 建立位于后端服务器 120 的多媒体计算机应用程序、调整位于后端服务器 120 的多媒体计算机应用程序和/或与位于后端服务器 120 的多媒体计算机应用程序交互。用户终端 110-B 可使用网络浏览器 135 或结合编程与运行的应用程序 140 通过网络 130 建立存储于后端服务器 120 的多媒体计算机应用程序,调整多媒体计算机应用程序和/或与多媒体计算机应用程序交互。不管上述多媒体计算机应用程序是通过网络浏览器 135 获得还是通过结合编程与运行的应用程序 140 获得,上述后端服务器提供统一的代码库给用户终端 110-A 和 110-B,以供用户建立该多媒体计算机应用程序、调整该多媒体计算机应用程序和/或与该多媒体计算机应用程序交互。

[0035] 在一些实施例中,多媒体计算机应用程序可储存于用户终端 110-B 和后端服务器 120 中。在这些实施例中,当用户终端 110-B 无法访问网络 130 时,该用户仍能通过结合编程与运行的应用程序 140,建立多媒体计算机应用程序、调整多媒体计算机应用程序和/或与多媒体计算机应用程序交互。当用户终端 110-B 连接网络 130 时,则该多媒体计算机应用程序的任何变化、调整和/或与其的交互可以更新至储存于后端服务器的多媒体计算机应用程序。

[0036] 此外,在一些实施例中,结合编程与运行的应用程序 140 包含除了网络浏览器 135 的附加功能,这些附加功能包括允许从用户终端 110-B 的本地操作系统输入资产、提供输出多媒体计算机应用程序及其资产到后端服务器 120 的高级方法,以及如上述讨论的,离线编辑该多媒体计算机应用程序。一个输出多媒体计算机应用程序及其内容的高级方法的例子是将该多媒体计算机应用程序及其内容转换成可被无缝导入和可被结合编程与运行的应用程序 140 解压的单个 zip 文件。

[0037] 在一实施例中,用户终端 110-A 和 110-B 中的每一个是计算设备,如图 10 所示的计算设备。可以使用的网络浏览器的例子包括,例如,微软浏览器、火狐浏览器,苹果浏览器等。

[0038] 后端服务器 120 在单个的结合编程与运行的实施程序 145 中营造编程和运行环境。该后端服务器 120 还存储使用该结合编程与运行的实施程序 145 时建立和提供的多媒体计算机应用程序和资产。该后端服务器 120 存储建立的多媒体计算机应用程序和资产于数据库 150 中。

[0039] 在一实施例中,网络 130 是基于英特网的网络,其允许任何能访问英特网的计算设备通过上述单个的结合编程与运行的实施程序 145 访问特定多媒体计算机应用程序或资产。在另一个实施例中,网络 130 是基于内联网的网络,允许只有那些授权安全访问网络 130 的通信设备(即,用户终端 110-A 和 110-B)通过上述单个的结合编程与运行的实施程序 145 访问特定多媒体计算机应用程序或资产。

[0040] 图 2A 是根据一实施例的通过网络浏览器提供用户对用于建立多媒体计算机应用程序的结合编程与运行的实施程序的访问的流程图 200。流程图 200 开始于步骤 210,在该步骤中,用户终端(如图 1 所示的用户终端 110-A 和 110-B)通过网络浏览器(如图 1 所示的网络浏览器 135)与后端服务器(如图 1 所示的后端服务器 120)连接。在该用户终端通过该网络浏览器成功与该后端服务器连接后,流程图 200 进入步骤 220。

[0041] 在步骤 220 中,用户通过该用户终端要求访问特定多媒体计算机应用程序或要求

建立新的多媒体计算机应用程序。该后端服务器确定该用户终端是否具有必需的访问权限去访问上述多媒体计算机应用程序或建立新的多媒体计算机应用程序。如果该后端服务器确定该用户终端具有必需的访问权限,流程图 200 进入步骤 230。如果该后端服务器确定该用户终端不具有必需的访问权限去访问上述多媒体计算机应用程序或建立新的多媒体计算机应用程序,流程图进入步骤 240。在步骤 240 中,该流程结束。

[0042] 该后端服务器可以根据不同因素确定该用户终端是否具有必需的访问权限。例如,在一些实施例中,该后端服务器检查该用户终端是否已提供给该后端服务器访问特定多媒体计算机应用程序或建立新的多媒体计算机应用程序正确的密码和登录信息。

[0043] 在步骤 230 中,如果该用户终端要求特定多媒体计算机应用程序,该后端服务器提供对用于建立多媒体计算机应用程序的结合编程与运行的实施程序的前端部分的访问和通过该结合编程与运行的实施程序提供对该特定多媒体计算机应用程序的访问。如果该用户要求建立新的多媒体计算机应用程序,该后端服务器提供用于建立多媒体计算机应用程序的该结合编程与运行的实施程序和新的多媒体计算机应用程序给该用户终端。

[0044] 图 2B 是根据一实施例的通过存储于用户终端的结合编程与运行的软件应用程序提供该用户对用于建立多媒体计算机应用程序的结合编程与运行的实施程序的访问的过程的流程图 250。流程图 250 开始于步骤 260,在该步骤中,用户终端(如图 1 所示的用户终端 110-B)通过存储于该用户终端的结合编程与运行的软件应用程序(如图 1 所示的结合编程和运行的应用程序 145)与后端服务器(如图 1 所示的后端服务器 120)连接。在该用户终端成功与该后端服务器连接后,流程图 250 进入步骤 270。

[0045] 在步骤 270 中,用户通过该用户终端要求访问特定多媒体计算机应用程序或要求建立新的多媒体计算机应用程序。该后端服务器确定该用户终端是否具有必需的访问权限去访问上述多媒体计算机应用程序或建立新的多媒体计算机应用程序。如果该后端服务器确定该用户终端具有必需的访问权限,流程图 250 进入步骤 280。如果该后端服务器确定该用户终端不具有必需的访问权限去访问上述多媒体计算机应用程序,流程图 250 进入步骤 290。在步骤 290 中,该流程结束。

[0046] 在步骤 280 中,如果该用户终端要求特定多媒体计算机应用程序,该后端服务器提供对该特定多媒体计算机应用程序的访问,该特定多媒体计算机应用程序可使用存储于该用户终端的该结合编程与运行的软件应用程序的前端部分来调整和/或与其交互。如果该用户终端要求建立新的多媒体计算机应用程序,该后端服务器提供新的多媒体计算机应用程序,该新的多媒体计算机应用程序可使用存储于该用户终端的该结合编程与运行的软件应用程序来调整。

[0047] 图 1、2A 和 2B 中所示的系统和方法可用于在多媒体计算机应用程序运行时编辑该多媒体计算机应用程序,和用于在多媒体计算机应用程序被编辑时运行该多媒体计算机应用程序。本发明描述的编程环境和运行环境可具有相同功能。该编程环境和该运行环境可在位于不同地理位置的用户间提供多媒体计算机应用程序的实时合作。

[0048] 在本发明的一些方面中,图形用户界面(GUI)系统由该单个结合编程与运行的实施程序 145 建立。在一例子中,该 GUI 系统是例如由单个结合编程与运行的实施程序 145 实施的用户终端 110-A 和 110B 的图像显示器上的环境。在一例子中,该 GUI 系统包含于网络浏览器 135 内。



[0049] 在一些实施例中,在实施该 GUI 系统时,多媒体计算机应用程序由用户建立。每当任何变化在该 GUI 系统中产生,该多媒体计算机应用程序由该单个结合编程与运行的实施程序 145 调整。

[0050] 在一些实施例中,该 GUI 系统包括对象被布置于其中的窗口和能使用户选择工具以在该窗口中建立对象的工具箱。在一例子中,该工具箱包括逻辑表作为一种类型的工具。在一实施中,用户从该工具箱选择逻辑表作为工具并在该窗口中建立逻辑表作为对象是可能的。

[0051] 在一实例中,“逻辑表”是指工具,其用于针对复合语句的组成命题的每个真值指示复合语句的真值。该逻辑表可用于针对每个输入值指示该表中提供的逻辑的输出值。

[0052] 在一实例中,该窗口中的逻辑对象由单个结合编程与运行的实施程序 145 建立,该单个结合编程与运行的实施程序 145 访问数据库 150 的数据结构中的对象库,并调整该多媒体计算机应用程序。建立的对象可包括公开的逻辑表。公开的逻辑表包括真值表、回答表和状态表。

[0053] 真值表

[0054] 真值表:概述

[0055] 图 3 显示了可作为程序对象使用的公开的逻辑表的一实施例。表名称行 300a 标识出该逻辑表 300 的类型。在本实施例中,该逻辑表是真值表 300,其允许用户基于求值表达式为真(T)或假(F)的条件逻辑和布尔求值,来操控多媒体计算机应用程序内的对象属性。

[0056] 该真值表 300 包括逻辑行 301、名称列 317、表达式列 304 和相关联数据表 306。

[0057] 真值表:逻辑行

[0058] 在图 3 所示的例子中,显示了五个逻辑行 305、308、308a、311 和 315,但逻辑行的数量没有特别限定。逻辑行 301 包括关于引入属性的信息,如引入属性的名称(名称列 317)、求值表达式 304、同行 308a 和相关联数据 306。

[0059] 真值表:属性名称列

[0060] 上述引入属性为被连接的程序对象的属性。在图 3 所示的例子中,上述引入属性列于逻辑行 305、308、311 和 315 的属性名称列 317 中,其中行 305、308、311 和 315 的每个包括作为输入被接收的另一个对象的引入属性的标识符或功能区名称。例如,在属性名称列 317 的逻辑行 311 中,包括按键程序对象的功能区名称。该按键具有属性“已点击”,且该属性的功能区名称显示于属性名称列 317 的逻辑行 311 中。

[0061] 真值表:求值表达式

[0062] 求值表达式是所连接属性的条件语句。在图 3 所示的例子中,求值表达式列于逻辑行 305、308、311 和 315 的表达式列 304 中。在一例子中,行 305、308、311 和 315 的每个包括等号。求值表达式是可包括字母或数值的条件逻辑语句。在一例子中,求值表达式默认为“= 50”(50 = 当前值),如表达式列 304 的逻辑行 305 中所示。在一些例子中,求值表达式可包括“大于”或“小于”(“<”,“>”),如表达式列 304 的逻辑行 315 中所示。在其他例子中,数字表达式的显示方法没有特别限制,可以是“= <”或“=>”分别为“小于等于”或“大于等于”。在另一个例子中,表达式可包括字符“#”。在该实例中,该字符“#”是指该引入属性的当前值。例如,在表达式列 304 的逻辑行 315 中显示“= 0<#<50”,其可被解读为:

如果当前值(#)“大于等于 0”或“小于 50”,所求值为真(T)。

[0063] 真值表 :同上行

[0064] 逻辑行 308a 是同行,是对直接位于该逻辑行 308a 的上行的复制。在图 3 中,逻辑行 308a 是逻辑行 308 的复制,该逻辑行 308 具有名称“问题 1 :回答”。通过如何建立同行的例子,当在程序对象内的属性功能区被连接于“添加逻辑行”按钮时,先建立逻辑行。然后,当在该程序对象内的相同属性功能区被连接于前逻辑行时,建立同行。在本例子中,当在程序对象“问题 1”的功能区“响应”被连接于“添加逻辑行”按钮 351 时,建立逻辑行 308。当另一配线从该相同程序对象“问题 1”的相同功能区“响应”拖曳至先前建立的逻辑行 308 时,在该前逻辑行 308 正下方建立“同行”308a。同行是前逻辑行的复制。这允许该表对相同功能区求不同表达式的值。例如,在图 3 中,逻辑行 308 对于词语“猫”或“狗”求程序对象“问题 1”中的功能区“响应”的值。而其次的“同行”行 308a 对于词语“鸟”求该相同程序对象中的相同功能区的值。使用从相同程序对象的相同功能区连接至其连接行的相同步骤,可建立逻辑行 301 或值行 303 的同行。

[0065] 真值表 :目前列

[0066] 真值表 300 中的第一列 302 具有标题“目前”。该第一列 302 显示了引入属性的当前状态和它们的相关联表达式。该列不移动。

[0067] 真值表 :相关联数据列

[0068] 上述相关联数据包括与该程序对象的引入属性相关联的数据。在图 3 所示的例子中,该相关联数据包含于相关联数据表 306 中,并包括布尔属性。布尔属性具有真或假值。当新引入属性被连接于真值表 300 时,每个引入属性出现于新逻辑行中,并且该引入属性与真或假值相关联,从而建立新逻辑行与其在相关联数据表 306 中的相关联 T/F 数据。

[0069] 相关联数据表 306 中的每一列表示“属性状态”,其为一串如图 3 所示的如 FFFF 或 TFFF 的布尔属性。通过对表达式列 304 中的表达式求引入属性的值,每个 T 或 F 值显示了相关联引入属性的“状态”。例如,在相关联数据表 306 的列 4 中,行 305 中的“T”显示了引入属性“计量器 :数字”的求值为“真”。这里,计量器的数字等于“50”。行 308 中,相关联数据表 306 的列 4 中的“T”显示了引入属性“问题 1 :响应”的求值为“真”。这里,对问题 1 的响应是“猫”、“狗”或“鸟”。行 311 中,相关联数据表 306 的列 4 中的“F”显示了引入属性“按钮 :已点击”为假。这里,该按钮没有被点击。

[0070] 在一实施中,当更多引入属性被连接于表 300,相关联数据表 306 中的各号码列 306 被建立以显示真(T)和假(F)的所有可能组合。在一实例中,当更多列被建立而无法被显示,滚动条 358 出现以允许上述列滚动进入视图。该表还可分别通过移动窄“调整大小”列被拉长或缩短以显示更多或更少列,其中一列在 359 处显示。

[0071] 在一些例子中,为引入属性及其相关联值的当前状态提供了当前状态列,而匹配该当前状态列中所有的 T 和 F 数据的列被突出,从而成为“满足”列。在图 3 所示的例子中,在相关联数据表 306 中具有标题“目前”的第一列 302 显示了引入属性及其相关联值的当前状态。当各列滚动进入视图时,该列不移动。在此例子中,当前满足列 306a 用白色突出颜色显示。如果该当前选择列 306a 因滚动或调整该表大小而移出视图,“快动”图标 347 出现。当点击该快动图标 347 时,选择列 306a 移入视图,而各列相应地调整。“满足”列意指该列中所有的真(T)和假(F)数据匹配表达式列 304 中求值表达式的结果。

[0072] 真值表 :值行

[0073] 真值表 300 还包括值行 303。在图 3 所示的例子中,显示了五个值行 321、328、329、330 和 331,但值行的数量没有特别限定。在一例子中,新值行可通过从程序对象连接配线至“添加值行(+)”按钮 350,或通过点击“添加值行(+)”按钮 350 加入。

[0074] 值行 303 包括表达式以分配值和存放从真值表 300 输出的数据。值行 321、328、329、330 和 331 的名称列可被编辑以命名各行具有的值类别。值行 321、328、329、330 和 331 的表达式列 304 可用于输入相同值或序列值。在一例子中,如果使用相同值,输入该值至值行 303 的列 304 中。例如,值行 321 的列 304 具有输入值“6”,因此值行 321 重复该值“6”。如果值行 321 要具有序列值,“-”可添入值“6”,于是该表达式被输入为,例如“6 -”,而不是“6”。在值行 328 中,每个值是非重复的,因此不需要在表达式列 304 中输入表达式。个别值也可手动编辑以具有自定义值,这些自定义值是一序列数字的例外,如单元格 368 中所示。

[0075] 表达式可通过名称引用其他值行。例如,在图 3 中,值行 329 的列 304 中的表达式提及值行 321 的列 317 中的值“分数”的名称。值行 329 的列 304 中的表达式“=分数 +5”是指“5”分值添入值行 321 的各值中。表达式还可建立其他类型的引用其他值行的格式。例如,表达式“=求和(栏 1…栏 3)”表示“对于每一列,计算包含于值行的“栏 1”、“栏 2”和“栏 3”的值的总和”。

[0076] 值行 321、328、329、330 和 331 的相关联数据表 306 包括值域。该值域中的数据可以是字母或数字。该数字或字母值数据的长度可以是任何长度。该值域还可具有其他类型数据。例如,值行 330 具有“移动标记”属性的位置和旋转信息(XYR)。该位置信息包括具有水平“X”、垂直“Y”和旋转角“R”的对象。在相关联数据表 306 的值行 330 中,该位置和旋转信息由网格图和每个单元格中的标记图形地表示。值行还可具有颜色信息。例如,在值行 331 中,相关联数据表 306 的值行 331 中的每列具有表示为色卡的颜色值。

[0077] 真值表 :引入属性端口

[0078] 引入属性在引入属性端口处被真值表 300 接收作为输入。在图 3 所示的例子中,显示了用于各引入属性配线的四个引入属性端口 332、334、335 和 342。上述引入属性端口 332、334、335 和 342 允许连接另一对象的属性至上述引入属性端口 332、334、335 和 342。它们还允许从上述引入属性端口 332、334、335 和 342 连接至另一对象的属性,从而使上述引入属性配线可绑定,双向端口。双向端口允许数据被发送给该表,也允许数据被发出给程序对象。

[0079] 真值表 :导出属性端口

[0080] 真值表 300 还包括导出属性端口 301a。导出属性端口 301a 是界面,其能通过连接的属性配线,从该表发送数据出去。在图 3 所示的例子中,显示了用于各引入属性配线的四个导出属性端口 361、365、367 和 369。在一例子中,上述导出属性端口 361、365、367 和 369 可在三个模式间切换 :单向或只读绑定,双向绑定和输出。

[0081] 在单向或只读绑定模式中,是从上述导出属性端口 301a 绑定至程序对象,从而真值表 300 的属性变化将被反映于所连接的程序对象的属性中,但不是反之亦然。该单向、只读绑定模式分别由一大一小长方形 343 和 344 可视表示。

[0082] 在双向绑定模式中,类似地从上述导出属性端口 301a 绑定至程序对象,从而真值

表 300 的属性变化将被反映于所连接的程序对象的属性中,而且反之亦然。该双向绑定模式分别由一小及一大圆点 345a 和 345 可视表示。在输出模式中,该导出属性端口引发一单向信号至程序对象的输入端口。信号在表达式的求值变为真“T”时被引发。该信号启动该程序对象的输入动作,如:显示,隐藏,转到页面,和其他动作。该输出绑定模式由一大三角形 346 或一小三角形可视表示。

[0083] 在列行中,有双向导出属性端口 360,其发送当前列码至连接的程序对象。因为该端口具有双向绑定模式,其也可以从连接的程序对象接收列码。当该端口从连接的程序对象接收列码时,真值表 300 使那列为所选择的列,而引入属性值改变以匹配显示于新选择列中的那些值。

[0084] 真值表:值属性端口

[0085] 值行 303、303a 包含在行 321、328、329、330、331 和行 391、394、393、382、383 的左端和右端的双向绑定属性端口。它们可从程序对象接收数据和发送数据出去至程序对象。值行 303、303a 可连接于左端或右端。例如,在图 3 中,值行 328 的配线 349、352 将发送来自值行 328 中的选择列 306a 的值“正确”,或接收数据以改变该值“正确”为不同值。

[0086] 真值表:输出端口和输出区

[0087] 当列被满足时,输出端口 384 可被连接于对象以触发动作。当列变为当前状态,输出端口 384 引发单向信号至程序对象的输入端口以发出该列已被满足的信号。如果有由于该表已被压缩而不在视图中的连接列,输出配线区 381a、381b 可用于收集这些配线。

[0088] 真值表:播放控制 356 和输入端口

[0089] 真值表 300 还包括播放控制 356 和几种类型的输入端口:判断目前输入 372、重置表输入 374、控制输入 376 和电源输入 373。播放控制 356 允许创建者逐步判断相关联数据表 306 的各列。当点击判断目前输入 372 时,该表求当前数据的值。当点击重置表输入 374 时,该表重置所有求值到原始设置。控制输入 376 允许外部按键连接于该表以控制逐步判断其各列或“播放”该表。控制输入 376 的功能复制播放控制 356 的功能,但控制输入 376 可外部地连接。电源输入允许该表判断(求值)的性能打开或关上。该表判断的电源默认为“打开”。输入配线可从程序对象连接于电源输入 373 以外部地打开或关闭该表判断的电源。

[0090] 真值表:间栏

[0091] 真值表 300 还包括间栏 398。该间栏包括间行 395、秒转换行 396 和秒保持行 397。间行 395 包括数据表 306 的列 2-5 的各单元格中的 XY 直线或曲线图。如直线和曲线的可视线表示代表算法,其可应用于状态间的转换。秒转换行 396 包括数据表 306 的列 2-5 的各单元格中的数值。秒转换行 396 中的这些数值表示秒数,在该秒数内状态间的转换应该发生。秒保持行 397 还包括数据表 306 的列 2-5 的各单元格中的数值。秒保持行 397 中的这些数值表示从前一状态转换到新状态开始前延迟的秒数。

[0092] 间栏 398 可用于操控对象的状态间的转换。例如,音频对象可具有两个属性,音量和 XY 坐标,并且该两个属性的每一个可具有两个状态,其中各属性的每个状态具有不同于另一状态的值。在一例子中,用户可设置由如数据表 306 的列 2 的间行 395 中所示的直线表示的线性算法、“秒转换”值为数据表 306 的列 2 的秒转换行 396 中所示的 1,和“秒保持”值为数据表 306 的列 2 的秒保持行 397 中所示的 3。在此例子中,当该逻辑表接收触发以将该音频对象的状态从初始状态改变为后续状态,将有三秒的延迟,然后该音频对象将从第

一位置移动到第二位置,而该音量将在一秒转换周期中以线性方式从第一音量调节到第二音量。如果该用户作为替代地设置由如数据表 306 的列 3 的间行 395 中所示的唇形的 v 形曲线表示的“跳跃”算法,该音频对象将“跳跃”进入和 / 或出去该转换。如果该用户作为替代地设置由如数据表 306 的列 4 的间行 395 中所示的曲线表示的减缓算法,该音频对象将渐入和 / 或渐出该转换。在一些例子中,上述间栏具有至少大约 20 个间转换。间栏可具有的间转换数没有特定限定。在其他例子中,每个间转换由可视线表示和各自的名称代表。

[0093] 真值表 :流程图和例子

[0094] 图 4A 是根据所公开的方法的一实施例,显示由系统 100 执行的流程的步骤的流程图。

[0095] 参阅图 4A 和图 5,一旦该程序开始,图形用户界面(GU I)窗口 504 显示于用户终端 110-A 和 110-B 的图像显示器上。

[0096] 图 4B 是涉及建立对象的流程的一实施例的流程图。这相当于图 4B 中的子程序 A。参阅图 4B,工具箱是被打开的。该工具箱可包括一组工具,该组工具包括按键工具、真值表工具、回答表工具、状态表工具和场区域工具,等等。如图 5 所示的例子中,工具箱 500 显示于窗口 504 中。该工具箱 500 包括按键工具 501、真值表工具 502 和场区域工具 503。该用户可通过用指针式装置选择在工具箱 500 中的可应用工具,然后在窗口 504 中点击建立相关程序对象以建立对象。

[0097] 图 5 示出了显示六个对象 506、508、510、512、514 和 516 的窗口 504。对象 516 是由真值表工具 502 建立的真值表 516。真值表 516 具有可视图形部件 516a。虽然没有显示,真值表 516 还可包括信息中心。该信息中心通常提供对象的属性列表,在该列表中用户可通过从所列属性的一个拖曳配线至另一个对象的属性,在对象间链接属性。真值表 516 的可视图形部件 516a 包括逻辑行 530、532 和 534、名称列 536、表达式列 538、值行 540 和相关联数据表 545。

[0098] 其他对象包括按键 506、508 和 510,其为由按键工具 501 建立的按键对象,和场区域 514,其为由场区域工具 503 建立的场区域对象。虽然在以后描述的例子中,按键 506、508 和 510 及场区域 514 是用于与真值表 516 链接,其他类型的对象也可用于与该真值表链接,如草图、长方形、椭圆形、线、动画路径、文本、文本输入、复选框、滑块、时钟、其他表格和其他程序对象。这些对象可由如铅笔、椭圆、滑块等工具建立。可在 504 中建立的对象数也没有特定限定。

[0099] 图 4C 是涉及链接按键 506、508 和 510 至真值表 516 的流程的流程图。这相当于图 4C 中的子程序 B。该子程序以调用子程序 A 开始。参阅图 4C 和 5,按键 506、508 和 510 分别具有可视图形部件 506a、508a 和 510a 以及信息中心 506b、508b 和 510b。在所示的例子中,可视图形部件 506a、508a 和 510a 显示为油漆刷,但可视图形部件可采用任何用户想要的形式。

[0100] 参阅图 4C 和 5,通过从按键 506、508 和 510 的各信息中心 506b、508b 和 510b 中的属性拖曳配线 518、520 和 522 并将上述配线落在真值表 516 的可视图形部件 516a 左边缘上的各输入配线锚 524、526 和 528 上,而在按键 506、508 和 510 的属性和真值表 516 间建立链接。输入配线锚 524、526 和 528 在值行 530、532 和 534 的左边缘上。如果上述配线既没拖动也没落入,该流程处于等待状态直到实施拖动或落入操作。

[0101] 一旦配线 518、520 和 522 落入输入配线锚 524、526 和 528, 在各值行 530、532 和 534 的名称列 536 中提供按键 506、508 和 510 的引入属性的标识符或功能区名称。例如, 对于按键 506 连接的属性是该按键被选择时的可见度。在值行 530 的名称列 536 中, “红色选择: 可见” 出现。

[0102] 配线 518、520 和 522 的落入导致系统 100 链接按键 506、508 和 510 的属性至显示于数据表 545 的属性状态。此处的术语“属性状态”是指任何值和任何数量的值的组合表达。属性状态列举了任何数量的具有数值属性的引入属性的所有可能排列, 其作为该真值表的输出。在一例子中, 该属性状态是两个值, 真或假, 的组合表达。在此例子中, 该属性状态可以是如 TTF 的一串值, 属性数量相应于输入数量。在本例子中, 有三个输入, 而属性状态的总集将是: TTT、FTT、TFT、FFT、TTF、FTF、TFF、FFF。

[0103] 在图 5 中, 在号码为 0、4、5、6 和 7 的列中分别提供了 TTT、TTF、FFT、TFT 和 FTT。列码 0、4、5、6 和 7 标识各列中的每一列, 并表示属性状态。

[0104] 图 4A 还包括用配线链接导出端口至场区域的属性, 调用子程序 C。

[0105] 采用列码以在属性状态和程序对象的引入属性间关联。例如, 如图 5 所示, 在列行 535 中提供列码 0、4、5、6 和 7, 该列行 535 具有在该列行 535 最右侧的导出端口 537。当导出端口 537 被连接于程序对象中的属性(未显示), 导出端口 537 发送列码给该程序对象。导出端口 537 是双向端口, 因此该程序对象也可发送列码给导出端口 537。当真值表 516 从该程序对象接收该号码, 适当的列成为所选择列, 并且连接的输入属性值变化为显示于所选列中的那些值。

[0106] 在图 5 所示的例子中, 属性状态 TTT、TTF、FFT、TFT 和 FTT 分别具有列码 0、4、5、6 和 7。列 0 具有当前或“目前”属性状态。列 6 显示为当前选择的属性状态。在图 5 所示的例子中, 具有名称“场区域 1: 页码”的值栏 545 对于列码 0、4、5、6 和 7 分别具有值 6、0、5、6、7、8。值栏 545 通过属性端口 550 经由配线 552 连接至场区域对象 514。特别是, 配线 552 被连接于“页码”功能区, 该功能区是信息中心 514b 的一部分。当列在真值表 516 中被满足时, 值栏端口 550 发送所选列码 6 的值栏 545 中的值到场区域 514 的信息中心 514b。在此例子中, 该数字控制显示于场区域 514 的页面, 该控制数字为页 6。该页码显示于场区域 514 的控制栏 514c 中。

[0107] 在另一例子中, 用户可具有第二值栏, 其对于列 4、5、6、和 7 分别具有值 20、30、40 和 50。在一例子中, 该第二值栏可被命名为“分数”。在此例子中, 数字 20、30、40 和 50 是发送给文本对象的分数值, 该文本对象显示所获得的分数。

[0108] 虽然没有显示, 在一例子中, 引入属性的求值表达式显示于值行 530、532 和 534 的表达式列 538 中。在此例子中, 这些表达式具有格式“=表达式”。在一实施中, 当引入功能区具有真(T)或假(F)的值时, 这些表达式将求值为真(T)或假(F)。该例子中的表达式可包括: “= T”, “= F”, “=T|F”。表达式“=T|F”是指当该引入属性是真(T)时和当该引入属性是假(F)时, 该引入属性可被求值为真(T)。

[0109] 新值行可以多种方式被建立。一种方式是从程序对象连接配线至左侧或右侧属性端口 550, 或拖曳并将该配线落在“添加值栏”按键 529 上。另一种方式是点击“添加值栏”按键 529 以添加新值栏。

[0110] 在图 5 所示的例子中, “提交混合”按键 512 被连接 512b 于“反馈场区域”对象。

“提交混合”按钮的输出“点击”被连接于该场区域对象的输入“显示”。当点击按钮“提交混合”时,反馈场区域被显示,使其反馈信息可见于该用户。在另一例子中,该“提交混合”按钮,或另一程序对象,可被连接以在不需反馈场区域时隐藏该反馈场区域。

[0111] 回答表

[0112] 回答表:概述

[0113] 图 6A 显示了可用作程序对象的公开的逻辑表的一实施例。名称行 600a 标识出该逻辑表 600 的类型。在本实施例中,该逻辑表是回答表 600,该回答表 600 允许该表评估用户回答和响应,并基于条件逻辑和回答或响应的顺序在多媒体计算机应用程序内操控对象的属性。

[0114] 图 6A 显示了回答表 600 的例子。图 6A 所示的回答表 600 类似于真值表 300,因为回答表 600 也包括下列项目,它们以与其在真值表 300 中作用的方式类似的方式作用。在这点上,参考上面真值表 300 中相应项目的描述。一表名称行 600a

[0115] 一逻辑行 601

[0116] 一属性名称列 617

[0117] 一表达式列 604

[0118] 一同上行 608a

[0119] 一“目前”列 602

[0120] 一相关联数据表 606

[0121] 一值行 603

[0122] 一引入属性端口 632、635

[0123] 一导出属性端口 601a 包括 644、645、646

[0124] 一值属性端口 648、649、651、652、653、654

[0125] 一输出端口 684

[0126] 一输出配线区 681a、681b

[0127] 一播放控制 656

[0128] 一判断目前输入 672

[0129] 一重置表输入 674

[0130] 一控制输入 676

[0131] 一电源输入 673

[0132] 一间栏 698 包括 695、696、697

[0133] 回答表:回答判断

[0134] 在一例子中,回答表 600 在如何在相关联数据列中判断表达式上不同于真值表 300。参阅图 6A,上述相关联数据列以打勾 606b 和数字 606a 求依据顺序的表达式的方法来求表达式的值。每列被认为是一个“回答”。该用户必须满足相关联数据表 606 中所示的顺序中的每个逻辑行表达式。当逻辑行表达式被求值为真(T)值,逻辑行表达式是被满足的。

[0135] 如果数据列中的单元格具有打勾标记 606b,该逻辑行中的表达式必须为真,且其就任何顺序可为真。当表达式就任何顺序可为真时,该列满足为回答的标准是所有具有打勾标记的表达式被求值为真(T),但这些表达式按什么顺序被求值为真(T)并不重要。

[0136] 如果数据列中的单元格具有数字 606d,则该逻辑行中的表达式必须都为真,并且其必须按该列中的相关联数据单元格中的一系列数字所示的顺序变为真。例如,在图 6A 中,在列 3 中,逻辑行 608 必须变为真,首先因为其相关联数据单元格具有顺序数字“1”。接着逻辑行 609 在逻辑行 608 后必须变为真,因为其相关联数据单元格具有顺序值“2”。然后逻辑行 608a 在逻辑行 608 后必须变为真,因为其相关联数据单元格具有顺序数字“3”。如果该用户的选择促使这些属性按不同顺序变为真,则列 3 中的回答未被满足。

[0137] 打勾标记和顺序数字可相结合。例如,在图 6A 中,在列 2 中,逻辑行 608 必须在逻辑行 608a 前变为真,但逻辑行 608 可按任何顺序,即在逻辑行 608 和 608a 变为真前、后或在两者之间,变为真。

[0138] 回答表 600 使用符号系统,该符号系统使用打勾标记、数字和字符表示不同形式的排序。该顺序符号的系统在图 6B 中被描述。

[0139] 当相关联数据列 606 中的单元格被满足时,单元格改变颜色。例如,在图 6A 中,逻辑行 608a 的列 1 中的单元格已被满足,因此逻辑行 608a 的列 1 的单元格中的颜色已变化为绿色 606c。如果单元格已被满足,其颜色已变为绿色,并且该列也是当前选择列,则已被满足的单元格的颜色变为淡绿,如逻辑行 608a、列 2 的单元格所示。

[0140] 回答表 600 从左至右判断。那即是,回答表 600 先求列 1 的值,接着移到列 2,接着到列 3,等等,并继续移向右边,直到到达完全被满足的一列。当标为打勾标记或顺序数字的列中的所有单元格已满足,回答表 600 判断该列完全满足。当发生该情况时,整列变为绿色,包括列码、逻辑单元格和值单元格。例如,在图 6A 中,列 5 是完全被满足的。如果回答表 600 无法找到被满足的列,回答表 600 再一次从列 1 开始,并从左至右继续进行求每一列的值,直到回答表 600 找到被满足的列。回答表 600 重复该模式的判断,除非该表被关闭。

[0141] 回答表:分支

[0142] 回答表 600 包括分支系统。这显示于分支行 610 中。当列被满足时,回答表 600 通过该列中的分支按键 610a,遵循该分支设置。可点击该分支按键以在三种类型的分支间循环:

[0143] 1. 继续:显示为向右箭头 610a。

[0144] 该分支继续从左向右求值。求值移至下一列。

[0145] 2. 再试一次:显示为指向左边的具有“逆转—”C”曲线的箭头。

[0146] 该分支停止该左至右判断扫描,并等待来自该用户的更多输入。

[0147] 3. 退出:显示为向下指的箭头。

[0148] 该分支停止判断扫描。退出回答表 600 并关闭其电源。

[0149] 回答表:状态端口和状态信息

[0150] 回答表 600 包括状态端口和状态信息 610b。其显示了回答表 600 的当前操作状态。该状态端口是单向的、只读端口,其发送与每个状态条件相关联的数字。该状态信息显示当前状态条件,以及与该状态条件相关联的数字。例如,在图 6 中,状态信息 610b 显示回答表 600 处于“准备”状态条件,其是指回答表 600 准备接收更多输入。状态条件的数量将为大约五个状态条件,但不限于这个数量。

[0151] 回答表:例子

[0152] 图 7 是使用中的回答表的一个例子。在该实施例中,提供了判断多媒体计算机应



用程序的回答,其显示了该回答表如何判断不同顺序的回答。

[0153] 图 7 示出了显示下列对象的窗口 700:复选框 720、滑块 730、动物文本输入栏位对象 740、重置按键 750 和颜色长方形反馈对象 760 及文本反馈对象 770。还有回答表对象 705。该回答表对象使用工具箱 702 中可用的回答表工具 703 建立。其他对象 720、730、740、750、760 和 770 分别用工具箱 702 中的复选框工具、滑块工具、文本输入工具、按键工具、长方形工具和文本工具来建立。

[0154] 复选框 720、滑块 730 和动物输入文本 740 的属性分别被连接到回答表对象 705 的逻辑行 710、711、712。复选框 720 的信息中心 722 中的已勾选属性被配线 721 连接于回答表对象 705 中的逻辑行 710。滑块 730 的信息中心 732 中的值-%属性被配线 731 连接于回答表对象 705 中的逻辑行 711。动物文本输入栏位对象 740 的信息中心 742 中的文本属性被配线 741 连接于回答表对象 705 中的逻辑行 712。在表达式列 718 中,已输入表达式以求每个引入属性的值。当表达式被求值为真(T),则该表达式被确定为已被满足。

[0155] 相关联数据列 713 一列 1、2、3、4、5 一包含命令以判断属性 710、711、712 是否已被满足,和按什么顺序属性 710、711、712 已被满足。如果属性可按任何顺序被满足,其单元格具有打勾标记。如果属性必须按特定顺序被满足,其单元格具有顺序数字。例如,在列 1 中,属性 710、711、712 必须按顺序被满足。在列 2 中,属性 711 必须先被满足,接着属性 710 必须被满足,然后属性 712 必须被满足。在列 3 中,这三个属性可按任何顺序被满足。在列 4 中,属性 711 可按任何顺序被满足,但属性 712 必须在属性 710 被满足前被满足。列 5 没有判断标准,所以其“捕捉”不匹配先前的列 1、2、3 和 4 的任何响应。

[0156] 在上述逻辑行下面是两个值行 716 和 717。值行 716 根据哪列被完全满足,来改变颜色反馈长方形 760 的颜色。颜色反馈长方形 760 的信息中心 761 中的填充属性被配线 762 连接于回答表 705 中的值行 716。值行 717 根据哪列被完全满足,来改变文本反馈对象 770 的文本。文本反馈长方形 770 的信息中心 771 中的文本属性被配线 772 连接于回答表 705 中的值行 717。

[0157] 列 713 示出了所有三个属性 710、711、712 已被满足,但它们不是都按一些列要求的顺序被满足。对于列中要满足的属性,其必须被求值为真(T),并且其必须按该列要求的顺序做。当列中的属性被满足,该单元格的顏色改变,例如,变为绿色 719。在图 7 所示的该例子中,列 3 已完全被满足。当列完全被满足时,在用打勾标记或顺序数字判断的所有那些逻辑行中的所有属性已被满足。当列完全被满足时,其被回答表对象 705 选择。当列被选择时,该列码单元格改变,例如,为白色。在图 7 所示的例子中,列 3 被选择,从而其列码单元格为,例如,白色。当列被满足时,其值从导出属性端口 701 发送到这些值被连接于的外部程序对象中的属性。在图 7 所示的该例子中,由于列 3 被选择,值行 716 已发送颜色红色至该颜色反馈长方形,并且值行 717 已发送文本“明白了!”到文本反馈对象 770。

[0158] 列 4 也被满足,但其没被选择。那是因为回答表对象 705 从左至右判断各列。在其他实施例中,回答表对象可按不同顺序判断各列。分支 704 控制在列被满足和被选择时,回答表对象 705 做什么。在此例子中,列 3 被满足和被选择。列 3 的分支 702 已被设置为“再试一次”,其是指回答表对象 705 发送与该列相关联的数据值,并暂停以等待来自该用户的更多响应。回答表对象 705 不继续从左至右判断,因此尽管列 4 完全被满足,但回答表对象 705 没有选择列 4。如果列 3 的分支 702 已被设置为“继续”(未显示),回答表对象 705

将继续从左至右判断,并且将确认列 4 完全被满足。如果列 4 的分支已被设置为“再试一次”,则该回答表将停止于列 4,并发送值数据至那些连接于导出属性端口 701 的属性。

[0159] 该例子还包括重置按键 750。重置按键 750 的信息中心 752 中的输出“点击一开始”被配线 751 连接于回答表对象 705 中的重置表输入 780。当重置按键 750 被点击,所有判断结果被清除,所有列单元格改变为灰色以显示它们还没判断引入属性,所有引入属性被重置为其默认值,并且回答表对象 705 等待其可判断的来自该用户的响应。

[0160] 在一实施例中,当该回答演示被运行,信息中心、配线和回答表对象 705 隐藏(未显示)。该用户直接控制这三个对象,并接收颜色反馈和文本反馈。在另一实施例中,当该回答演示被运行,信息中心、配线和回答表对象 705 可见于该用户(未显示)。

[0161] 状态表

[0162] 状态表:概述

[0163] 图 8A 示出了可用作程序对象的公开的逻辑表的一实施例。名称行 800a 标识出该逻辑表 800 的类型。在本实施例中,该逻辑表是状态表 800,其允许该表存储多媒体计算机应用程序内的对象属性的多个值,以为连接的程序对象建立一系列“状态”,这些“状态”可按使用时间轴按时间顺序被播放。

[0164] 图 8A 示出的状态表 800 类似于真值表 300,因为状态表 800 也包括下列项目,它们以与其在真值表 300 中作用的方式类似的方式作用。在这点上,参考上面真值表 300 中相应项目的描述。

[0165] 一表名称行 800a

[0166] 一属性名称列 817

[0167] 一表达式列 804

[0168] 一同上行 808a

[0169] 一“目前”列 802

[0170] 一相关联数据表 806

[0171] 一值行 803

[0172] 一值属性端口 848、849、851、851a、852、853、854

[0173] 一输出端口 884

[0174] 一输出配线区 881a、881b

[0175] 一播放控制 856

[0176] 一判断目前输入 872

[0177] 一重置表输入 874

[0178] 一控制输入 876

[0179] 一电源输入 873

[0180] 一间栏 898

[0181] 在一例子中,状态表 800 不使用逻辑行、引入属性端口或导出属性端口。

[0182] 状态表:状态

[0183] 参阅图 8A,状态表 800 使用值行 803、803a 集合和存储连接于状态表 800 的程序对象的属性的多个值。例如,在图 8A 中,值行 803 具有就每个引入属性 821、828、829、808a、833、836、830、831 的多个值。值可包括文本、数字、屏幕位置、对象旋转、颜色和其他类型的

值。对于值行 828,其属性名称是“鱼:名称”,并且其具有 5 个值:“鲨鱼”、“鲸鱼”、“虹鳟”、“鲶鱼”和“鲑鱼”。每列示出了一组连接属性的值。其为该组连接属性的“状态”。例如,在图 8A 中,对于在图 8A 中为当前选择列的列 3,值行 821、828、829、808a、833、836、830、831 的值分别为“13”、“虹鳟”、“印度洋”、“河流”、“3”、“水族馆”,如单元格内的位置图表可视化示出的连接的属性“移动标记”的大约右下屏幕坐标和 45 度旋转角、以及如单元格内的色卡可视化示出的颜色黄色。共同地,该组值是列 3 的“状态”。当列被选择时,其单元格中的值被发送到连接于状态表 800 的属性。当不同列被选择时,连接于状态表 800 的属性接收不同组的值。结果为,因为一组属性被连接于状态表 800,其可响应于用户或程序动作呈现多个“状态”,或作为渐进序列或动画逐步判断多个“状态”,而状态表 800 的各列相应地变化以改变每个连接属性的状态。

[0184] 属性 803 被连接于具有双向绑定端口的值行,因此状态表 800 既可发送值给连接属性,也可从连接属性接收值。

[0185] 状态表:控制和控制输入

[0186] 状态表 800 可用真值表使用的相同项目控制。这些项目包括播放控制 856 和控制输入 876。播放控制 856 允许创建者在编辑多媒体计算机应用程序时,逐步判断状态表 800 的“状态”。控制输入 876 允许外部按键连接于状态表 800,以控制逐步向前或向后判断其各列,或按顺序“播放”该表。

[0187] 状态表:时间轴

[0188] 参阅图 8A-F,状态表 800 还可用其综合时间轴 892 控制。时间轴 892 允许设置持续时间 825,其建立时间的总长度。在时间轴 892 的右侧,有分时的时间轴。圆点 823 可从每一列被拖曳至该分时时间轴以在总持续时间 825 内创建时间,在该时间该列被显示。这是该列的“起始时间”。当圆点被按下并沿该分时时间轴被拖曳时,该圆点的颜色变化 824,并且其沿着该时间轴的时间位置以文本显示于该圆点 824 上方。

[0189] 时间轴 892 具有“刷子”控制按键 826。该“刷子”按键可沿该分时时间轴被拖曳。当其跨越该分时时间轴被拖曳,所选择列随该“刷子”移动过“起始时间”圆点而变化。这允许创建者通过沿该时间轴前后移动该“刷子”来测试序列。

[0190] 时间轴 892 与状态表 800 中的间栏 898 一起作用。当转换时间被设置于间栏 898 的秒转换行 896 中,其秒数在该时间轴中可视显示为从起始时间圆点向右边延伸的蓝色长条 823a 以恰当地显示转换秒数。当保持时间被设置于间栏 898 的秒保持行 897 中,它告诉状态表 800 和时间轴 892 在显示下一列前保持该秒数。这为下一列建立了“起始时间”,因此与该列相关联的“起始时间”圆点 823 沿分时时间轴移动以反映保持秒数 897。如果列的“起始时间”圆点 823 移动,前一列的保持秒 897 相应变化。

[0191] 时间轴 892 具有“信息”按键 844。当点击该信息按键 844 时,新的“时间轴数据”表 844a 出现。它示出每一列的“起始时间”823a、824a 为时间格式的数字。这些“起始时间”反映了该分时时间轴上“起始时间”圆点 823、824 的位置。例如,在图 8A 中,“起始时间”圆点 823 位于沿该分时时间轴的 00:35 处。该时间数字 00:35 也在“时间轴数据”844a 起始时间栏位 823a 中。“时间轴数据”表 844a 中的“起始时间”栏位数量反映了状态表 800 中的列数量。每个“起始时间”栏位中的时间值可被编辑。当该“起始时间”栏位被编辑,“起始时间”圆点 823 在该时间轴上相应移动以匹配输入到“时间轴数据”844a “起始时间”栏

位中的时间。

[0192] “时间轴数据”表 844a 具有“刷子”825a,其模拟时间轴 892 中“刷子”825 的功能。该“时间轴数据”表 844a 可通过点击关闭按钮“X”844b 被关闭。

[0193] 时间轴 892 还可作为独立的时间轴 837a。工具箱 838 具有时间轴工具 837。当它被点击时,它建立独立于状态表 800 的时间轴 837a。程序对象的属性可直接被连接于沿分时时间轴的“起始时间”圆点 837b。这允许属性和输入动作被连接于该时间轴,其允许一系列属性按时间顺序被引发或输入动作按时间顺序被引发。

[0194] 状态表:例子

[0195] 图 9 提供了使用时的状态表的一个例子。在该实施例中,状态表在多媒体计算机应用程序中用于教学相机的光圈设置如何影响景深。

[0196] 图 9 示出了显示数个对象的窗口 900:树背景对象 930、景深比例 940、包括左光圈按钮 950、右光圈按钮 960 以及光圈设置显示 970 的相机对象 980。还有状态表对象 905。状态表对象 905 使用工具箱 902 中可用的状态表工具 903 建立。

[0197] 光圈设置显示 970、树背景对象 930 和景深比例 940 的属性分别被连接入状态表对象 905 的值行 910、911、912。值列 913 一列 1、2、3、4、5 一包含光圈设置显示 970、树背景对象 930 和景深比例 940 的值的 5 个状态。光圈设置显示 970、树背景对象 930 和景深比例 940 每个被连接于状态表对象 905。树背景对象 930 的信息中心 931 中的模糊属性 932 被连接于状态表对象区 922 中的端口。景深比例 940 的信息中心 941 中的尺寸-高度属性 942 被连接于状态表对象区 922 中的端口。光圈设置显示 970 的信息中心 971 中的文本属性 972 被连接于状态表对象区 922 中的端口。

[0198] 状态表对象 905 中的列 3 被当前选择,因此这些值分别显示于光圈设置显示 970、树背景对象 930 和景深比例 940 中。光圈设置显示 970 示出值行 910 中状态表对象 905 的所选择列 914 中的值 5.6f。树背景对象 930 示出值行 911 中状态表对象 905 的所选择列 914 中的 30%模糊。景深比例 940 的高度示出值行 912 中状态表对象 905 的所选择列 914 中的 200 像素。由于状态表对象 905 中有 5 列,相机仿真可显示所连接属性的 5 个状态。状态表对象 905 中的列数可根据应用而改变,并且没有特定限定。

[0199] 在此例子中,相机对象 980 的左光圈按钮 950 和右光圈按钮 960 为可用于改变仿真状态的按钮。它们在该仿真中行为,就像光圈按钮会在实体相机上行为的。左光圈按钮 950 的信息中心 951 中的输出“点击开始”属性 952 被连接于状态表对象 905 中的“返回一步”控制输入 920。当左光圈按钮 950 被点击,状态表对象 905 向后移动一列并相应改变所连接属性的值。右光圈按钮 960 的信息中心 961 中的输出“点击开始”属性 962 被连接于状态表对象 905 中的“前进一步”控制输入 921。当右光圈按钮 960 被点击,状态表对象 905 向前移动一列并相应改变所连接属性的值。

[0200] 当时间轴不用于该例子,在其他实施例中,时间轴可用于按顺序呈现一序列所有五个状态。

[0201] 在一些实施例中,当该仿真被运行,信息中心、配线和状态表对象 905 对该用户来说是隐藏的(未显示)。该用户直接控制该相机并查看结果。然而,在其他实施例中,当该仿真被运行,信息中心、配线和状态表对象 905 对该用户来说是可见的(未显示)。

[0202] 图 10 是计算机系统 2300 的示意图。系统 2300 可用于与本发明描述的任何计算

机实施方法相关联的描述的操作。系统 2300 可包括各种形式的数字计算机,如笔记本电脑、台式电脑、工作站、个人数字助理、服务器、刀锋服务器、主机及其他合适的计算机。系统 2300 还可包括移动装置,如个人数字助理、移动电话、智能手机及其他类似计算装置。此外,系统 2300 可包括便携式存储介质,如闪存盘(USB Flashdrives)。例如,闪存盘可存储操作系统和其他应用程序。闪存盘可包括输入/输出部件,如无线传送器或可插入另一个计算装置的 USB 端口的 USB 连接器。

[0203] 在此例子中,系统 2300 包括处理器 2310、存储器 2320、存储装置 2330 和输入/输出装置 2340。部件 2310、2320、2330 和 2340 的每一个使用系统总线 2350 相互连接。处理器 2310 能处理系统 2300 内的执行命令。处理器 2310 可使用一些架构中的任何架构设计。例如,处理器 2310 可以是 CISC(复杂指令集计算机)处理器、RISC(精简指令集计算机)处理器,或 MISC(最小指令集计算机)处理器。

[0204] 在一些实施中,处理器 2310 是单线程处理器。在一些实施中,处理器 2310 是多线程处理器。处理器 2310 能处理存储器 2320 中或存储装置 2330 存储的命令以在输入/输出装置 2340 上显示用户界面的图形信息。

[0205] 存储器 2320 在系统 2300 内存储信息。在一些实施中,存储器 2320 是计算机可读介质。在一些实施中,存储器 2320 是易失性存储单元。在一些实施中,存储器 2320 是非易失性存储单元。

[0206] 存储装置 2330 能为系统 2300 提供海量存储。在一些实施中,存储装置 2330 是计算机可读介质。例如,存储装置 2330 可以是软碟装置、硬碟装置、光学磁盘装置或磁带装置。

[0207] 输入/输出装置 2340 为系统 2300 提供输入/输出操作。在一些实施中,输入/输出装置 2340 包括输入装置(如键盘、如鼠标或追踪球的指针式装置;触摸屏;手指动作装置;等等)。在一些实施中,输入/输出装置 2340 包括用于显示图形用户界面的显示单元。

[0208] 本发明描述的特征可在数字电子线路中,或计算机硬件、固件、软件或其结合中实施。装置可在计算机程序产品中实现,计算机程序产品有形地承载于信息载体中,例如,机器可读存储装置中或传播信号中,由可编程处理器执行;以及方法步骤可由可编程处理器实现,可编程处理器通过对输入数据进行操作执行程序命令以实现所描述实施的功能并生成输出。上述特征可在一个或多个计算机程序中有利地实施,这些计算机程序在可编程系统上可执行,该可编程系统包括至少一个联接以从数据存储系统接收数据和指令和传送数据和指令给数据存储系统的可编程处理器、至少一个输入装置和至少一个输出装置。计算机程序是可直接或间接在计算机中使用以执行某个活动或产生某个结果的一组指令。计算机程序可以任何形式编程语言撰写,包括编译语言或解释语言,并且计算机程序可以任何形式布局,包括作为独立程序或作为模块、部件、子程序或适用于计算环境中的其他单元。

[0209] 执行程序指令的合适处理器包括,例如,任何种类计算机的多处理器之一或单处理器,以及通用微处理器和专用微处理器。通常,处理器从只读存储器或随机存取存储器或两者接收指令和数据。计算机的必要元件是执行指令的处理器、存储指令和数据的一个或多个存储器。通常,计算机还包括或可操作地联接以与一个或多个用于存储数据文件的大容量存储装置通信;该类装置包括磁盘,如内部硬盘和可移动磁盘;磁光盘;以及光盘。适用于有形承载主算机程序指令和数据的存储装置包括所有形式的非易失性存储器,包括例

如半导体存储装置,如电可编程只读存储器(EPROM)、电可擦除只读存储器(EEPROM)和闪存装置;磁盘如内部硬盘和可移动磁盘;磁光盘;和 CD-ROM、DVD-ROM 磁盘。处理器和存储器可由专用集成电路(application-specific integrated circuits,简称“ASIC”)提供,或嵌入 ASIC 中。

[0210] 为了提供与用户的交互,本发明描述的特征可在计算机上实施,该计算机具有显示装置,如用于显示信息给用户的 CRT(阴极射线管)显示器或液晶显示器(liquid crystal display,简称“LCD”),和输入装置(如键盘;如鼠标或追踪球的指针式装置;触摸屏;手指动作装置;等等)通过这些装置用户可提供输入给计算机。

[0211] 上述特征可在计算机系统中实施,该计算机系统包括后端部件,如数据服务器,或包括中间部件,如应用服务器或因特网服务器,或包括前端部件,如具有图形用户界面或浏览器的客户端计算机,或其中的任何组合。该系统的部件可通过任何形式或任何数字数据通信介质如通信网络连接。通信网络的例子包括局域网(LAN)、广域网(WAN)、点对点网络(具有特别或静态成员)、网格计算构架和因特网。

[0212] 该计算机系统可包括客户端和服务端。客户端和服务端通常相互远离,并且一般通过网络交互。客户端和服务端的关系利用各计算机上运行并且相互具有客户端-服务端关系的计算机程序产生。

[0213] 所公开的方法的优点之一是资产可直接从操作系统桌面替换。而且,资产可在多个对象中替换,而不用像传统创建程序的情形一个一个地替换它们。更甚者,资产可不用下拉菜单替换。如此,所公开的方法允许多个对象中的资产更有效率地被替换。

[0214] 虽然所公开的系统及其方法已配合使用优选实施例描述,对于本领域的技术人员来说,已公开系统及其方法的其他对象和改进可在公开的范围和范畴内作出是明显的。

[0215] 在其各个方面和所公开形式中的本发明非常适合于达到所述对象和其他优点。已公开的细节不是要被作为对权利要求书的限定。

[0216] 本发明的各方面:

[0217] 1. 一种非临时性计算机可读介质,其具有计算机可执行指令存储于其上,所述指令用于计算机装置执行使用逻辑表作为图形用户界面(GUI)系统中的程序对象以将多个程序对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括:

[0218] 通过该结合编程与运行的实施程序在该 GUI 系统显示第一程序对象、第二程序对象和该逻辑表,该结合编程与运行的实施程序访问来自存储于该数据库的对象库的该第一程序对象、该第二程序对象和该逻辑表,调整多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序;

[0219] 通过该结合编程与运行的实施程序链接作为输入的该第一程序对象的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序;

[0220] 通过该结合编程与运行的实施程序链接该第一程序对象的该属性至该逻辑表的属性状态,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序,该属性状态包括值语句,该第一程序对象

的该属性链接于该值语句,以及该属性状态链接于作为输出的该逻辑表的数值属性;

[0221] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的该逻辑表的该数值属性至该第二程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接该第二程序对象的该数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序,从而通过该逻辑表,链接该第一程序对象的属性和该第二程序对象的属性。

[0222] 2. 方面 1 的计算机可读介质,其中该值语句是真或假语句。

[0223] 3. 方面 1 和 2 中任一计算机可读介质,其中该值语句是字母数字的语句。

[0224] 4. 方面 1—3 中任一计算机可读介质,还包括基于求值表达式求该第一程序对象的属性的值,并基于该求值链接该第一程序对象的属性至该值语句。

[0225] 5. 方面 1—4 中任一计算机可读介质,其中不是该第一程序对象的程序对象被链接作为该输入,该属性状态包括多个值语句,其中这些值语句中的一个连接于不是该第一程序对象、被链接作为该输入的该程序对象的属性,并且该求值表达式引用不是该第一程序对象、被链接作为该输入的该程序对象的属性,而求值是基于对不是该第一程序对象、被链接作为该输入的该程序对象的属性的引用。

[0226] 6. 方面 1—5 中任一计算机可读介质,其中该逻辑表包括用于链接上述程序对象的属性的输入端口和输出端口。

[0227] 7. 方面 1—6 中任一计算机可读介质,其中该逻辑表包括用于链接该值语句的数值属性端口。

[0228] 8. 方面 1—7 中任一计算机可读介质,其中该逻辑表包括用于集合多个链接的输出配线区。

[0229] 9. 方面 1—8 中任一计算机可读介质,其中该逻辑表包括播放控制,其中该播放控制允许用户从一个属性状态移动到另一个属性状态。

[0230] 10. 方面 1—9 中任一计算机可读介质,其中该逻辑表包括间栏以操控一个属性状态到另一个属性状态的转换。

[0231] 11. 一种非临时性计算机可读介质,其具有计算机可执行指令存储于其上,所述指令用于计算机装置执行使用逻辑表作为图形用户界面(GUI)系统中的程序对象以将多个程序对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括:

[0232] 通过该结合编程与运行的实施程序在该 GUI 系统中显示多个程序对象和逻辑表,该结合编程与运行的实施程序访问来自包括该数据库中的对象库的数据库结构的该多个程序对象和该逻辑表,和调整该多媒体计算机应用程序;

[0233] 通过该结合编程与运行的实施程序链接作为各输入的上述程序对象的每个的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储所调整的计算机应用程序;

[0234] 通过该结合编程与运行的实施程序链接上述程序对象的属性中的每一个至该逻辑表的属性状态,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储所调整的计算机应用程序,该属性状态包括多个值语句,该多个程序对象的属性链接于

各该值语句的每一个,该多个值语句被按顺序地安排于该属性状态,并且上述值语句的至少一个顺序安排链接于作为输出的该逻辑表的数值属性,上述值语句的该至少一个顺序安排还链接于回答;

[0235] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的逻辑表的数值属性至另一个程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接另一个程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储进一步调整的多媒体计算机应用程序。

[0236] 12. 方面7的计算机可读介质,其中该逻辑表包括多个属性状态和多个回答,并且上述属性状态的每个链接于上述回答的一个。

[0237] 13. 一种非临时性计算机可读介质,其具有计算机可执行指令存储于其上,所述指令用于计算机装置执行使用逻辑表作为图形用户界面(GUI)系统中的程序对象以将多个程序对象相关联的方法,该GUI系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该GUI系统显示于该图像显示器,该方法包括:

[0238] 通过该结合编程与运行的实施程序在该GUI体系统中显示第一程序对象、第二程序对象和逻辑表,该结合编程与运行的实施程序访问来自该数据库中的对象库的该第一程序对象、该第二程序对象和该逻辑表,和调整该多媒体计算机应用程序;

[0239] 通过该结合编程与运行的实施程序链接第一程序对象的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的计算机应用程序,该逻辑表包括多个属性状态,该第一程序对象的属性是使得该第一程序对象能够从该逻辑表的一个属性状态切换到另一个属性状态;

[0240] 通过该结合编程与运行的实施程序链接第一程序对象的属性至该逻辑表的上述属性状态中的一个,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序,上述属性状态的每一个包括值语句,每个属性状态链接于作为输出的该逻辑表的数值属性;

[0241] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的该逻辑表的数值属性至该第二程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接该第二程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序,包含于上述属性状态中的每个的值语句定义该第二程序对象的属性。

[0242] 14. 一种计算机装置,该计算机装置包括具有程序指令的计算机可读介质和控制单元,这些程序指令由处理器执行时在图像显示器上处理和显示多个图像,该控制单元适于执行使用逻辑表作为图形用户界面(GUI)系统中的程序对象将多个程序对象相关联的方法,该GUI系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该GUI系统显示于该图像显示器,该方法包括:

[0243] 通过该结合编程与运行的实施程序在该GUI系统中显示第一程序对象、第二程序对象和该逻辑表,该结合编程与运行的实施程序访问来自存储于该数据库的对象库的该第一程序对象、该第二程序对象和该逻辑表,调整多媒体计算机应用程序和在该数据库中存



储所调整的多媒体计算机应用程序；

[0244] 通过该结合编程与运行的实施程序链接作为输入的该第一程序对象属筒至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序；

[0245] 通过该结合编程与运行的实施程序链接该第一程序对象的属性至该逻辑表的属性状态,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序,该属性状态包括值语句,该第一程序对象的属性链于该值语句,以及该属性状态链接于作为输出的该逻辑表的数值属性；

[0246] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的该逻辑表的数值属性至该第二程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接该第二程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序,从而通过该逻辑表链接该第一程序对象的属性和该第二程序对象的属性。

[0247] 15. 一种计算机装置,该计算机装置包括具有程序指令的计算机可读介质和控制单元,这些程序指令由处理器执行时在图像显示器上处理和显示多个图像,该控制单元适于执行使用逻辑表作为图形用户界面(GUI)系统中的程序对象将多个程序对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括：

[0248] 通过该结合编程与运行的实施程序在该 GUI 系统中显示多个程序对象和逻辑表,该结合编程与运行的实施程序访问来自包括该数据库中的对象库的数据库结构的该多个程序对象和该逻辑表,和调整该多媒体计算机应用程序；

[0249] 通过该结合编程与运行的实施程序链接作为各输入的上述程序对象的每个的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储所调整的计算机应用程序；

[0250] 通过该结合编程与运行的实施程序链接上述程序对象的属性中的每一个至该逻辑表的属性状态,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储所调整的计算机应用程序,该属性状态包括多个值语句,该多个程序对象的属性链接于各该值语句的每一个,该多个值语句被按顺序地安排于该属性状态,并且上述值语句的至少一个顺序安排链接于作为输出的该逻辑表的数值属性,上述值语句的该至少一个顺序安排还链接于回答；

[0251] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的逻辑表的数值属性至另一个程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接另一个程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储进一步调整的多媒体计算机应用程序。

[0252] 16. 一种计算机装置,该计算机装置包括具有程序指令的计算机可读介质和控制单元,这些程序指令由处理器执行时在图像显示器上处理和显示多个图像,该控制单元适于执行使用逻辑表作为图形用户界面(GUI)系统中的程序对象将多个程序对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结

合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括:

[0253] 通过该结合编程与运行的实施程序在该 GUI 体系统中显示第一程序对象、第二程序对象和逻辑表,该结合编程与运行的实施程序访问来自该数据库中的对象库的该第一程序对象、该第二程序对象和该逻辑表,和调整该多媒体计算机应用程序;

[0254] 通过该结合编程与运行的实施程序链接第一程序对象的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的计算机应用程序,该逻辑表包括多个属性状态,该第一程序对象的属性是使得该第一程序对象能够从该逻辑表的一个属性状态切换到另一个属性状态;

[0255] 通过该结合编程与运行的实施程序链接第一程序对象的属性至该逻辑表的上述属性状态中的一个,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序,上述属性状态的每一个包括值语句,每个属性状态链接于作为输出的该逻辑表的数值属性;

[0256] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的该逻辑表的数值属性至该第二程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接该第二程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序,包含于上述属性状态中的每个的值语句定义该第二程序对象的属性。

[0257] 17. 一种使用逻辑表作为图形用户界面(GUI)系统中的程序对象来将多个程序对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括:

[0258] 通过该结合编程与运行的实施程序在该 GUI 系统中显示第一程序对象、第二程序对象和该逻辑表,该结合编程与运行的实施程序访问来自存储于该数据库的对象库的该第一程序对象、该第二程序对象和该逻辑表,调整多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序;

[0259] 通过该结合编程与运行的实施程序链接作为输入的该第一程序对象属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序;

[0260] 通过该结合编程与运行的实施程序链接该第一程序对象的属性至该逻辑表的属性状态,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序,该属性状态包括值语句,该第一程序对象的属性链于该值语句,以及该属性状态链接于作为输出的该逻辑表的数值属性;

[0261] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的该逻辑表的数值属性至该第二程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接该第二程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储进一步调整的多媒体计算机应用程序,从而通过该逻辑表链接该第一程序对象的属性和该第二程序对象的属性。

[0262] 18. 一种使用逻辑表作为图形用户界面(GUI)系统中的程序对象来将多个程序

对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括:

[0263] 通过该结合编程与运行的实施程序在该 GUI 系统中显示多个程序对象和逻辑表,该结合编程与运行的实施程序访问来自包括该数据库中的对象库的数据库结构的该多个程序对象和该逻辑表,和调整该多媒体计算机应用程序;

[0264] 通过该结合编程与运行的实施程序链接作为各输入的上述程序对象的每个的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储所调整的计算机应用程序;

[0265] 通过该结合编程与运行的实施程序链接上述程序对象的属性中的每一个至该逻辑表的属性状态,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储所调整的计算机应用程序,该属性状态包括多个值语句,该多个程序对象的属性链接于各该值语句的每一个,该多个值语句被按顺序地安排于该属性状态,并且上述值语句的至少一个顺序安排链接于作为输出的该逻辑表的数值属性,上述值语句的该至少一个顺序安排还链接于回答;

[0266] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的逻辑表的数值属性至另一个程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接另一个程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和存储进一步调整的多媒体计算机应用程序。

[0267] 19. 一种使用逻辑表作为图形用户界面(GUI)系统中的程序对象来将多个程序对象相关联的方法,该 GUI 系统由与多媒体计算机应用程序交互的结合编程与运行的实施程序实施,该结合编程与运行的实施程序被提供在包括数据库和图像显示器的计算机装置上,该 GUI 系统显示于该图像显示器,该方法包括:

[0268] 通过该结合编程与运行的实施程序在该 GUI 体系统中显示第一程序对象、第二程序对象和逻辑表,该结合编程与运行的实施程序访问来自该数据库中的对象库的该第一程序对象、该第二程序对象和该逻辑表,和调整该多媒体计算机应用程序;

[0269] 通过该结合编程与运行的实施程序链接第一程序对象的属性至该逻辑表,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的计算机应用程序,该逻辑表包括多个属性状态,该第一程序对象的属性是使得该第一程序对象能够从该逻辑表的一个属性状态切换到另一个属性状态;

[0270] 通过该结合编程与运行的实施程序链接第一程序对象的属性至该逻辑表的上述属性状态中的一个,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序,上述属性状态的每一个包括值语句,每个属性状态链接于作为输出的该逻辑表的数值属性;

[0271] 或者(1)通过该结合编程与运行的实施程序链接作为该输出的该逻辑表的数值属性至该第二程序对象的属性,或者(2)通过该结合编程与运行的实施程序链接该第二程序对象的数值属性至该输出,该结合编程与运行的实施程序进一步调整该多媒体计算机应用程序和在该数据库中存储所调整的多媒体计算机应用程序,包含于上述属性状态中的每个的值语句定义该第二程序对象的属性。

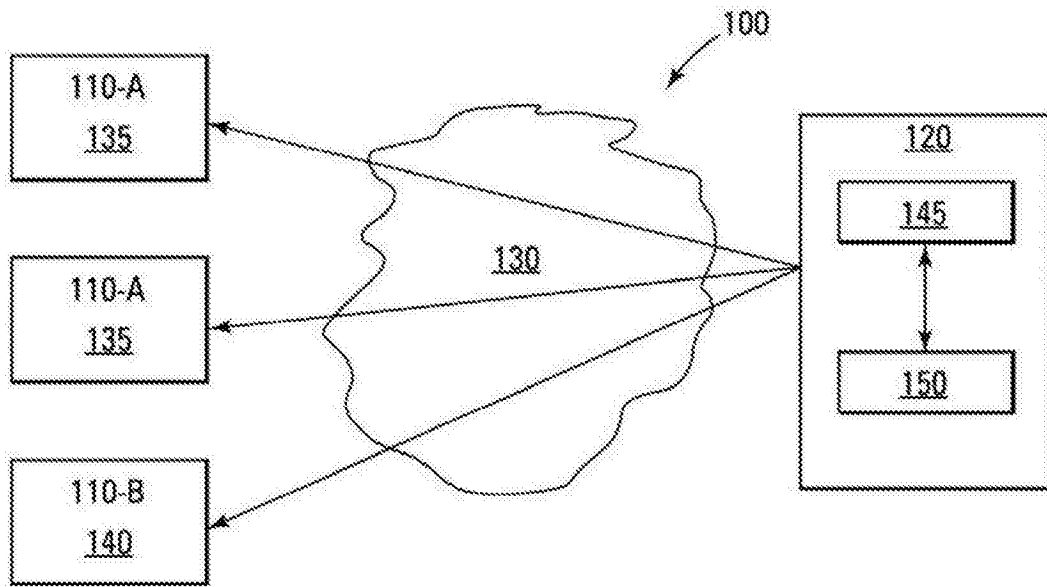


图 1

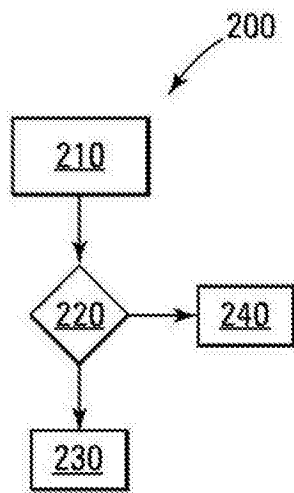


图 2A

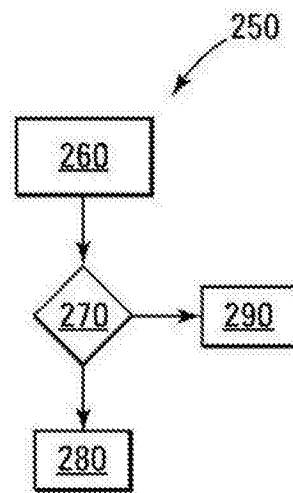


图 2B

真值表

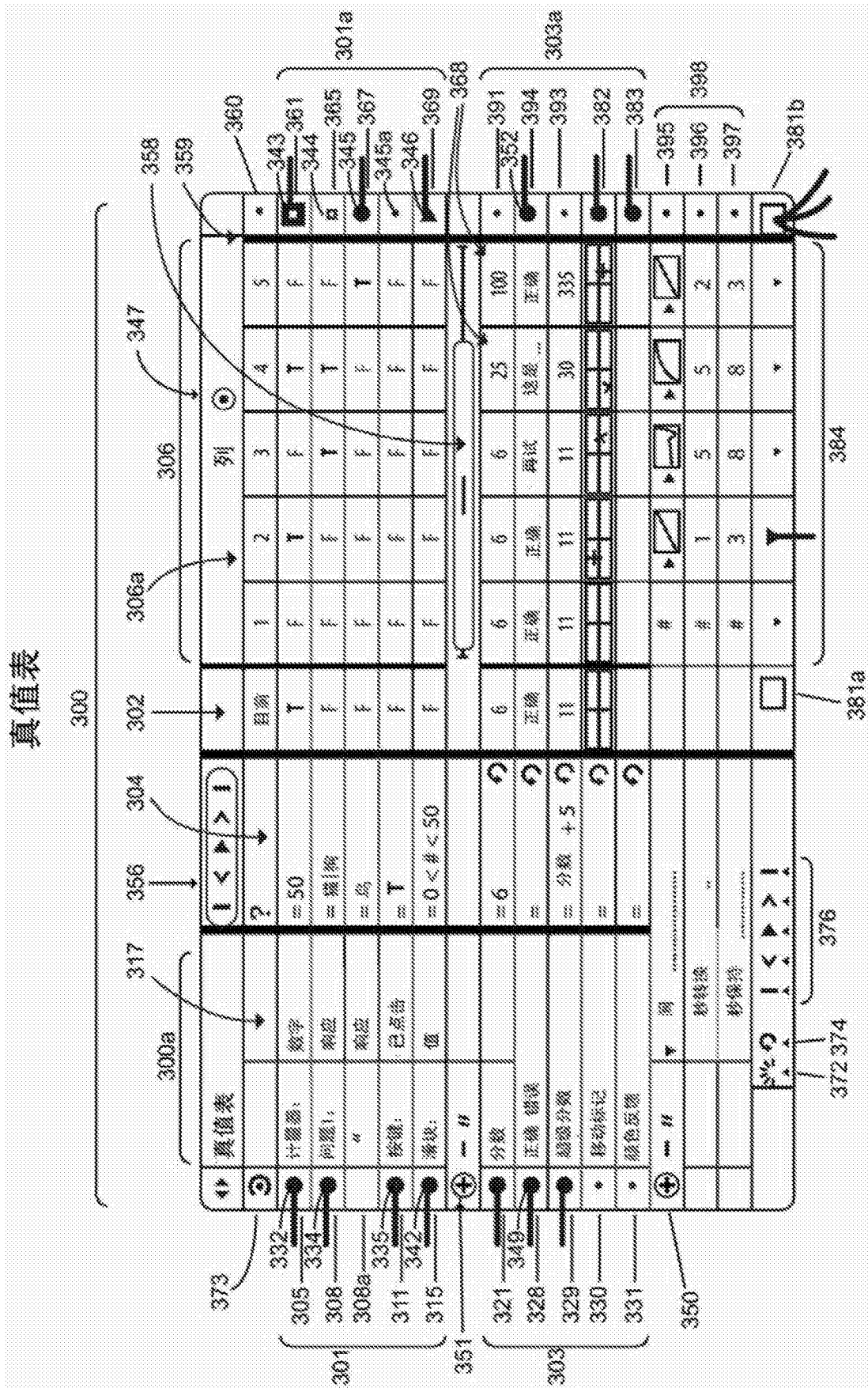


图 3

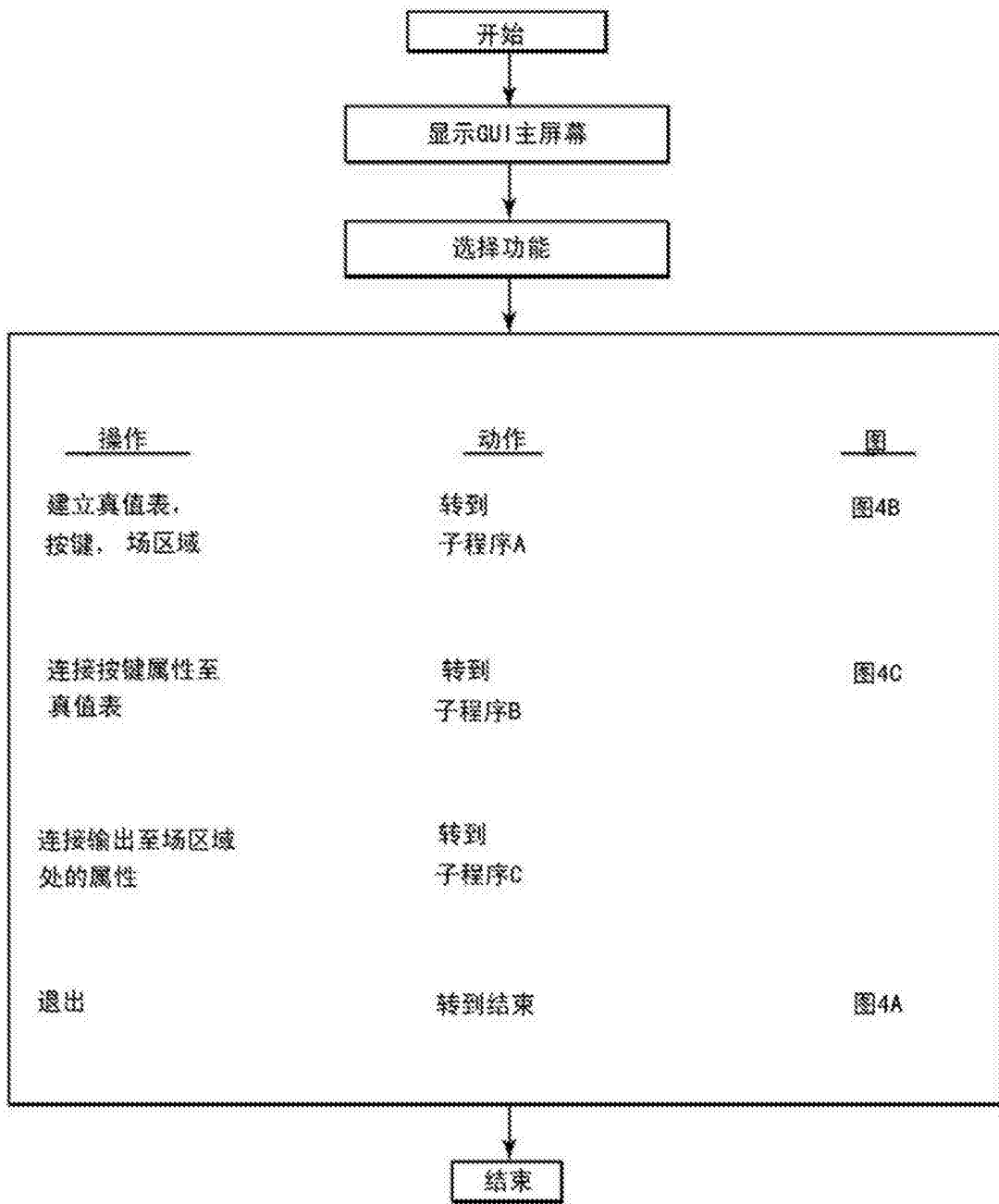


图 4A

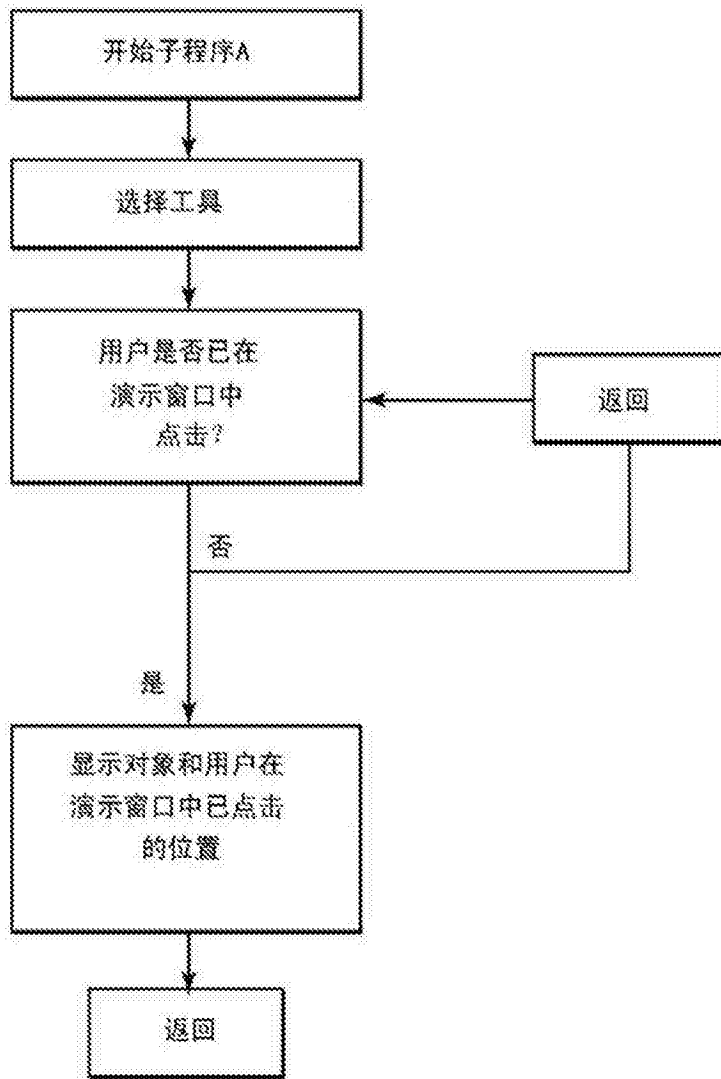


图 4B

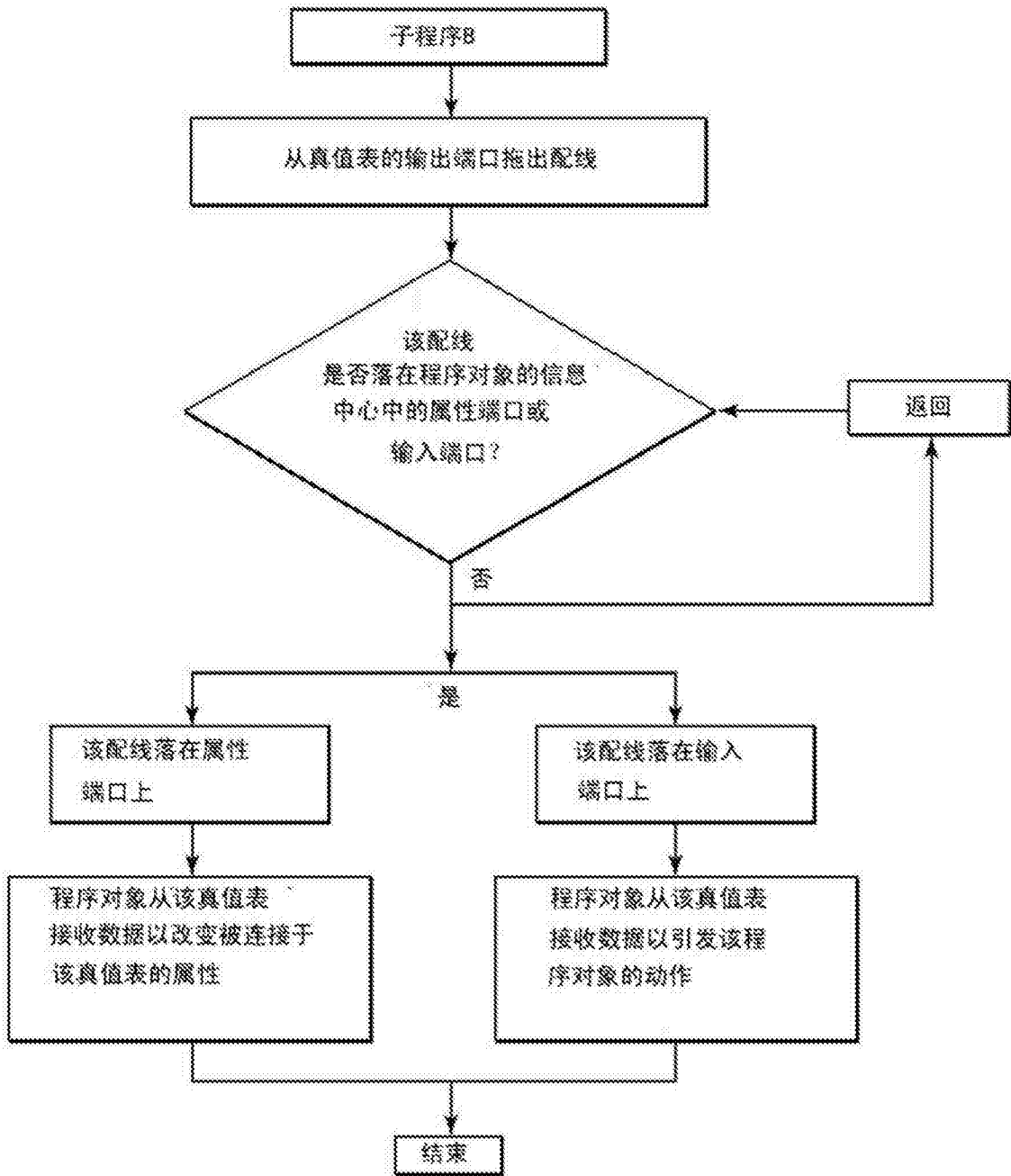


图 4C





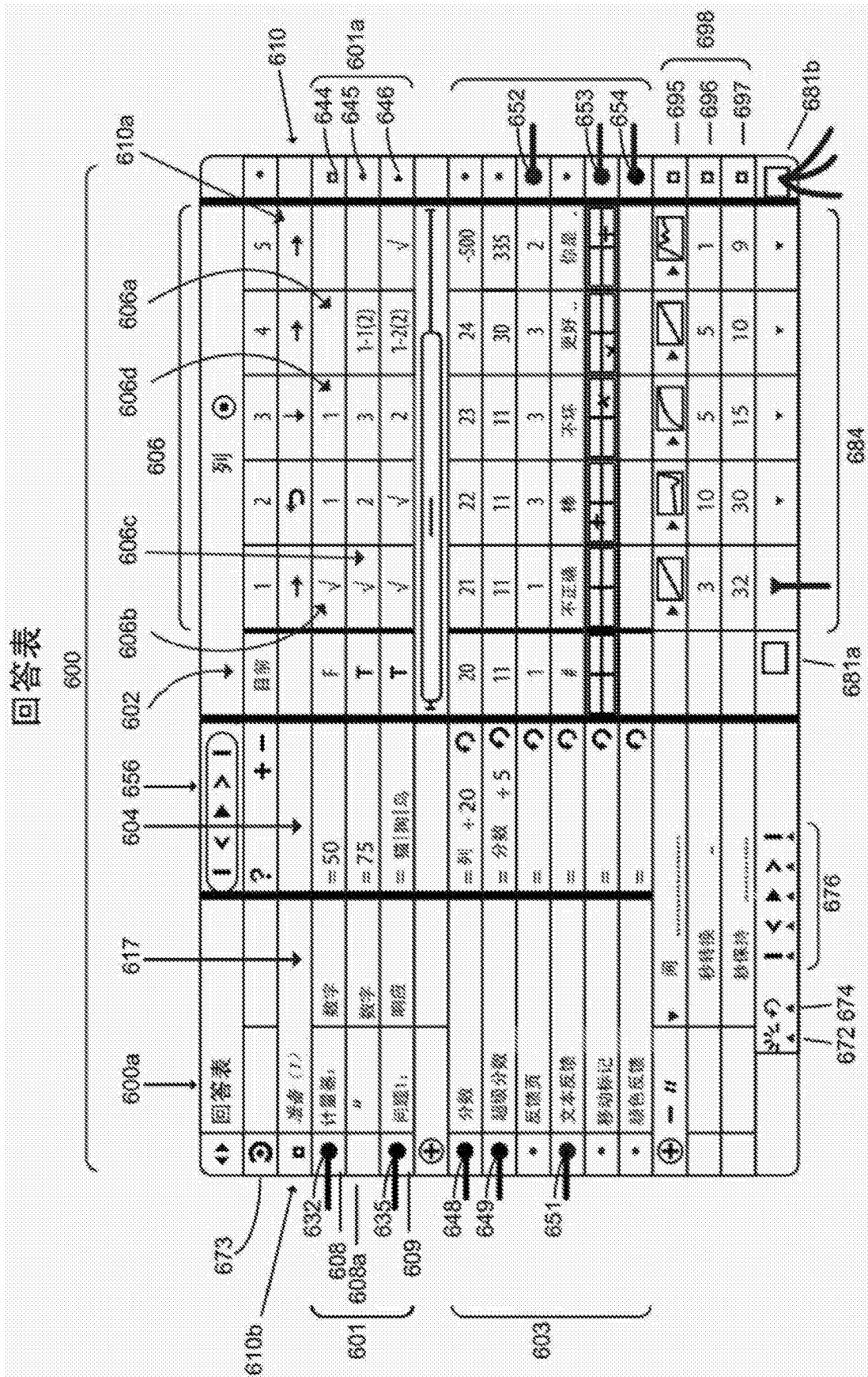


图 6A

回答表—顺序符号

该回答表基于表达式和顺序某值。  
 引入属性必须都为真...而且它必须按正确的顺序做。  
 顺序符号被键入到逻辑单元格中。  
 本页示出了符号协议...和案例。

	1	2	3	4	5	6	7	8	9	10	11
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

案例：描述...  
 列表：不同符号...  
 这是另一种方式以符号表示组中的顺序...  
 不使用1-4的符号。组“1”的组以符号组表示为1、2、3、4...  
 而组“2”被变为各“2”，等等...

列表：包括具有子集选择的非顺序步骤的顺序组...  
 顺序号指示选择范围...  
 所以在该例子中，在组“1”内，只有各“1”中的两个必须在...  
 相连到各“2”前被完成。所有“2”必须按任何顺序做。但在组...  
 “3”内，从5个项目组中只有1个项目必须被完成。

列表：包括具有子集选择的顺序步骤的顺序组...  
 在组“1”中，4个步骤必须按顺序做，1-4 至 1-4...但只有上...  
 述步骤的两个步骤被完成。例如，用户可完成1-4和1-3...或...  
 1-3和1-4...等等以完成该组。然后其将前进至组2和3。

列表：包括具有子集选择的非顺序步骤的顺序组...  
 非案例的符号组以指示“非顺序”。在此例子中...  
 ...组1必须先做。组1内的步骤可按任何顺序做...  
 ...组2必须第二个做。在组“2”内，有2个项目“a”和“b”...  
 任何一组可按任何顺序做，但两组必须在组2前做。在组...  
 “2a”内，只有2个项目“(2)”必须被完成。在组“2b”内...  
 只有一个项目“(1)”必须被完成...  
 ...组3必须最后做。

列表：具有顺序小组和非顺序小组的顺序组...  
 此案例使用字母以指示“非顺序”。在此例子中...  
 ...组A、B、C可以按任何顺序做...  
 ...在组A内，四个步骤必须在一起并且按顺序做...  
 ...在组B内，三个步骤必须在一起并且按顺序做...  
 ...组C具有两个步骤必须在一起但按任何顺序做...  
 列表：列表上的变化...  
 这与图10相同，但形式上在组c中的步骤，您可以按任何顺序做...  
 它们不能要在一起作为组做。

图 6B

回答表：例子

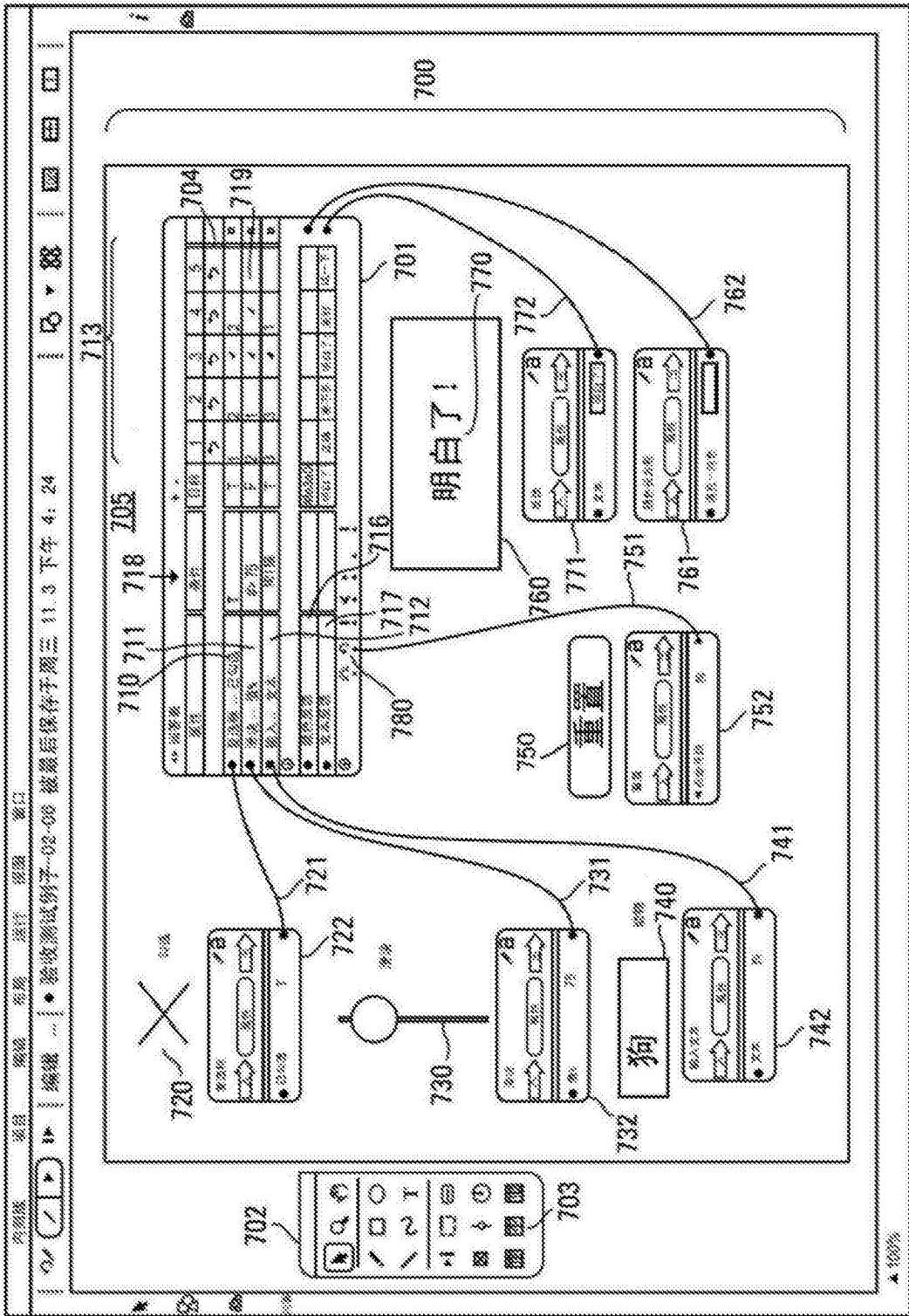


图 7

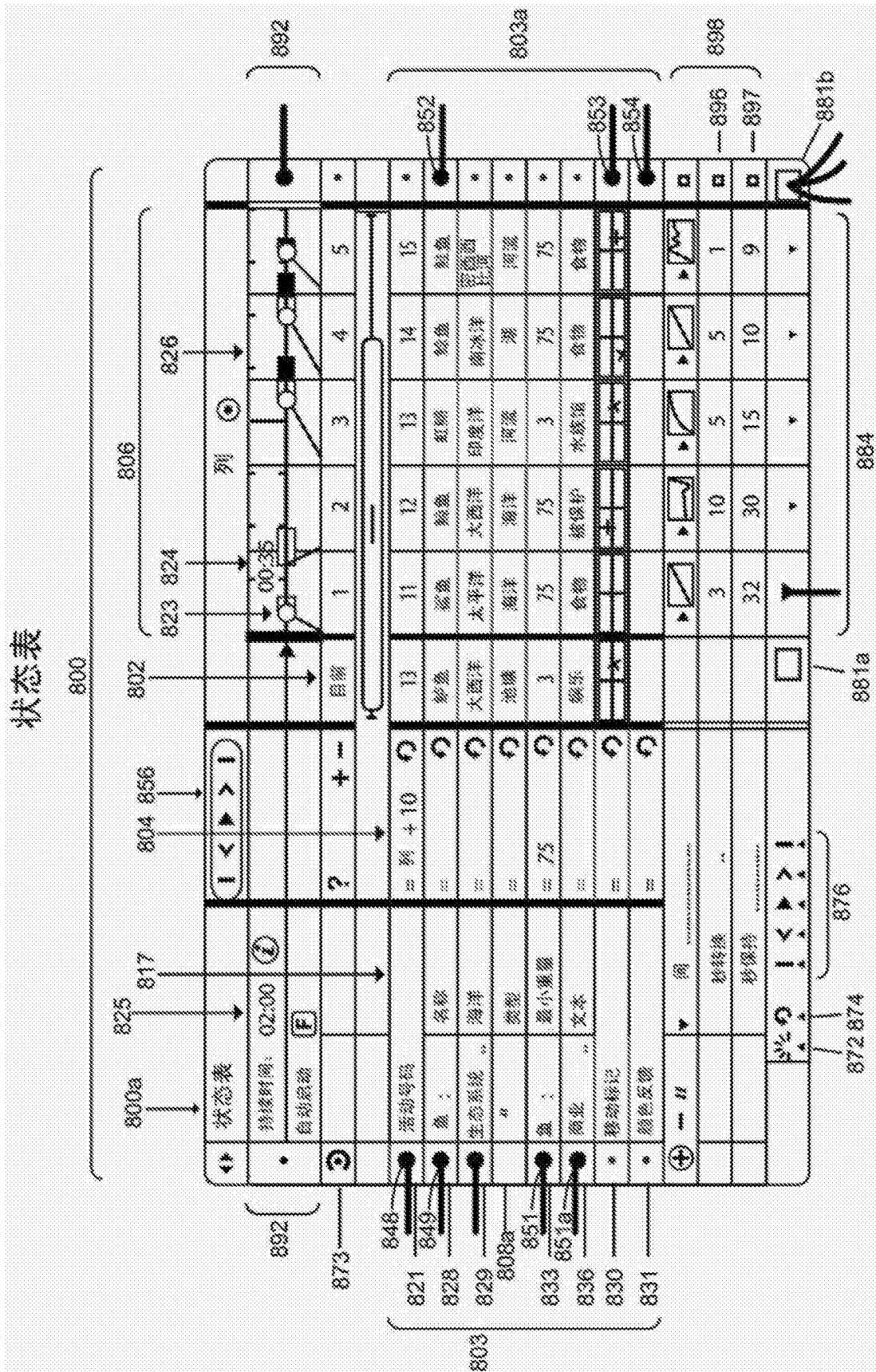


图 8A

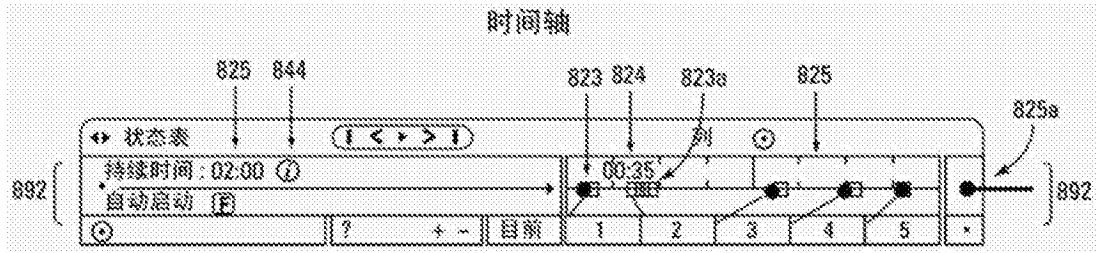


图 8B

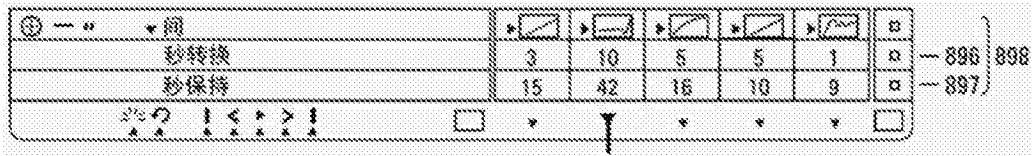


图 8C



图 8D

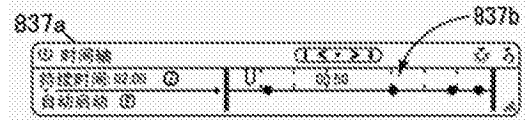


图 8E

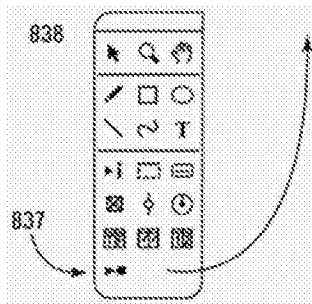


图 8F



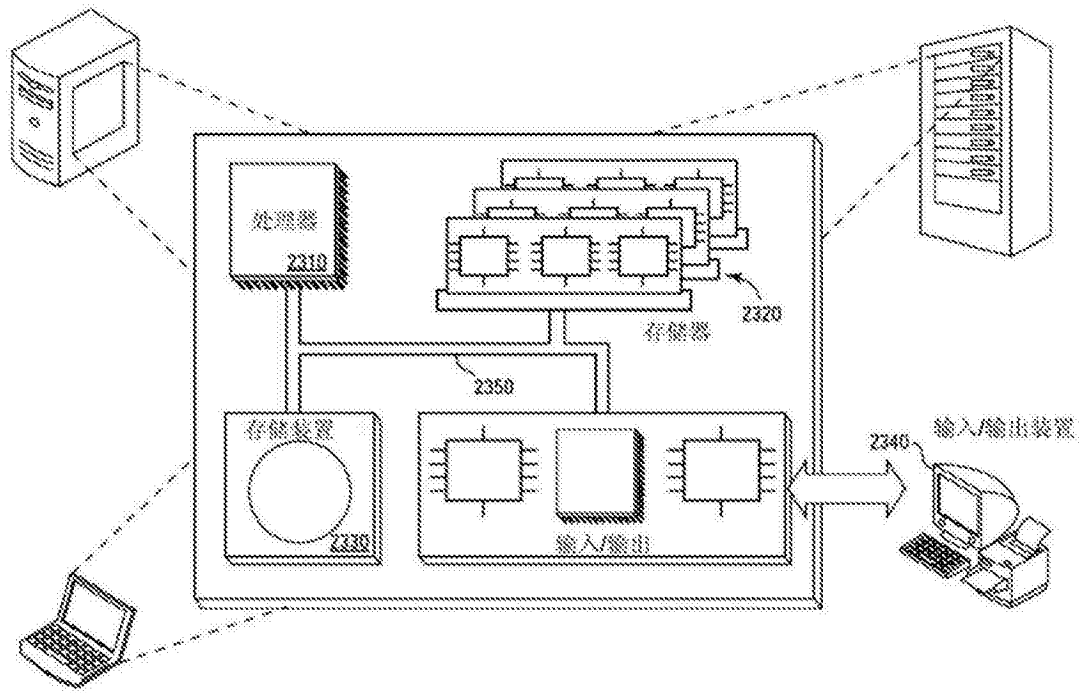


图 10