[19]中华人民共和国专利局

[11] 公开号 CN 1124379A

[12] 发明专利申请公开说明书

[21]申请号 94109561.4

[51]Int.Cl[6]

G06F 15/00

[43]公开日 1996 年 6 月 12 日

[22]申请日 94.8.19

[30]优先权

[32]93.8.23 [33]US[31]110,235

[71]申请人 美国电报电话公司

地址 美国纽约

[72]发明人 丹尼尔·F·赫尔利

厄尔·H·韦斯特

[74]专利代理机构 中国国际贸易促进委员会专利商标事务所

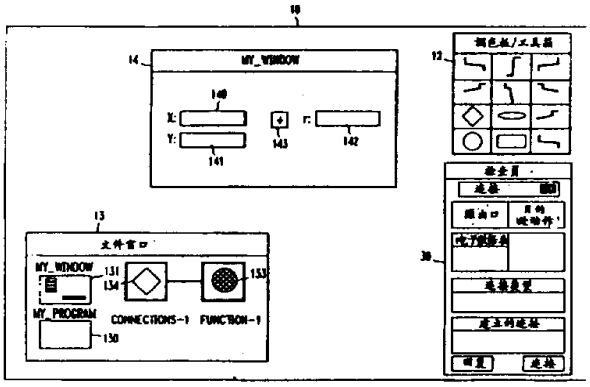
代理人 范本国

权利要求书 3 页 说明书 31 页 附图页数 17 页

[54]发明名称 使用可用子程序配置计算机程序的方法和装置

[57]摘要

一种图形程序配置系统，它允许用户建立完整计算机程序。它所提供的系统和过程允许一个子程序可具有任意个在程序配置期间可能定义的出口，其方法是激励子程序去建立新的方便于将子程序连接至其他子程序的出口。在本发明的特定实施例中，新出口的名称可或者 a）借助于由一个子程序到其他子程序所尝试的连接来获取，或 b）由程序配置系统的用户直接输入出口名称来提供，例如由用户在键盘上敲入出口名称。



权 利 要 求 书

1. 一种用于配置包含至少两个子程序的程序的系统，它包括：

用于向子程序发送信息，从而激励所述子程序去建立便于连接到其他子程序的新出口的装置；和

用于提供说明所述新出口已可用于连接的标志的装置。

2. 权利要求 1 中所定义的系统还包括用于在所述子程序的所述所建立的新出口与至少一个其他子程序之间建立连接的装置。

3. 权利要求 1 中所定义的系统，其中所述子程序通过导出一个出口名称列表而建立所述新出口，所述出口名称列表是从通过与第二子程序对象交换信息而获得的信息中导出的。

4. 权利要求 3 所定义的系统，其中所述第二子程序在送给所述子程序的所述信息中提供对象名称列表，而所述子程序则从所述对象名称列表中建立所述新出口。

5. 权利要求 1 所定义的系统，其中所述子程序通过从由人员提供的信息中导出一个出口名称列表来建立所述子程序的所述新出口。

6. 权利要求 5 所定义的系统，其中所述人员通过在键盘敲入来提供所述出口名称列表。

7. 权利要求 3 所定义的系统，其中在所述子程序的所述新建立的至少一个出口和第二个子程序之间的连接被约束为依从于至少一种特定的连接类型。

8. 权利要求 7 所定义的系统，其中所述特定的连接类型包括一项限制，即对在所述子程序的所述新建立的至少一个出口和第二个子程序之间的连结数量的限制。

9. 权利要求 7 所定义的系统，其中所述特定的连接类型包括一项要求，即要求第二个子程序对象具有特定的预定特性。

10. 一种在具有能显示至少第一和第二对象的图形显示的计算机程序配置系统中使用的方法，该方法包括以下步骤：

从所述第一对象到所述第二对象建立一个连接；

依据于利用所述连接从所述第二对象所提供的信息，将所述第一对象加以激励，以建立至少一个出口；

将依据于所述信息而由所述第一对象所建立的所述至少一个出口加以显示；以及

将所述第一对象的所述至少一个所建立的出口连接至其他对象。

11. 一种具有能显示至少第一和第二对象的图形显示的计算机程序配置系统，所述第二对象具有至少一个出口，所述计算机程序配置系统包括：

用 于显示由所述第一对象从所述第二对象所获取的出口的

装置，所述获取的出口是通过在所述第一和第二对象间交换信息而获取的；以及

用于将所述第一对象的所述获取的出口进一步连结到至少一个其他对象的装置。

12. 权利要求11所定义的系统，其中所述其他对象是所述第二对象。

13. 权利要求11所定义的系统，其中所述其他对象是一个第三对象。

14. 一种在具有至少第一和第二对象的计算机系统中使用的方法，在所述计算机系统中至少某些所述对象具有出口，所述出口具有名称，所述方法包括以下步骤：

从所述第一对象向所述第二对象发送一条信息，要求所述第二对象将它的出口名称发送给所述第一对象；

从所述第二对象接收所述出口名称；以及

为从所述第二对象接收到的每一个名称在所述第一对象内建立一个出口，所述所建立的出口适用于将所述对象连接至其他对象。

说　　明　　书

使用可用子程序配置
计算机程序的方法和装置

本发明一般涉及到通过将可用的子程序加以连接和配置，以开发类似所谓“面向对象”的程序那样的计算机程序。

计算机程序允许计算机完成很多有用的任务。然而，为了编写计算机程序，特别是所谓“面向对象”的计算机程序，人们必须具备必要的计算机编程技巧和该特定的编程语言的语法知识。为了掌握必要的编程技巧和知识，需要付出相当多的时间，通常还需金钱。因此，本技术的目的是为非计算机编程人员或只有初步培训的编程人员提供方便来开发计算机程序。

于 1992 年 11 月 10 日颁发给 ***Hullot*** 和转让给 ***NeXT*** 计算机公司的美国专利第 5,163,130 号中所公开的现有技术系统曾向这个目标迈进，该前技术系统为早已至少部分编写好的程序所用图形接口的开发和配置使用了图形用户接口技术。然而，这样的现有技术系统不能从一组子程序中开发完全的，随意的计算机程序。这是因为该现有技术系统处理出口的能力有限。

一个“出口”是一个数据结构或对象，它提供必需的信息，后者在一个特定“源”对象和一个或更多“目的”对象之间建立连接时是需要的。根据现有技术状况，出口是在特殊设计用来与其他对象交互工作的“源”对象中定义的。利用出口的名称，每个出口向程序配置系统的用户提供有关“目的”对象的种类的标帜，而源对象必须与该“目的”对象连接才能发挥作用。

本发明的目的是利用程序配置方法和装置为建立计算机程序提供方便，该程序配置方法和装置允许一个或更多个子程序单元和一个或更多个图形接口单元的变量和函数部分和其他子程序单元或图形接口单元中的变量或函数连接起来，以便将一定类型的计算机程序的功能完全加以配置并准备付诸执行，而不需用户直接阅读、编写或编辑该特定计算机语言的编码。程序配置实用系统用于具备可见显示的计算机系统内，在该可见显示上应用程序由一个图形图象所代表，而该应用程序具有至少两个子程序或至少一个子程序和至少一个图形接口单元，如该应用程序具有至少两个子程序，则每个子程序都有变量和动作部分。程序配置实用系统可使非编程人员或只有初步编程技巧的人有能力将一个或更多的子程序的集合配置成一个完整的计算机程序，而不是局限于只将程序用户接口配置成不可见的，功能性程序单元。

尤其是，本发明将现有技术加以改善，提供一个系统和过程，后者通过激励子程序去建立便于将该子程序和其他子程序连接的新

出口，可在程序配置期间为子程序定义任意数量的出口。在本发明的特定实施例中，新出口的名称或 *a*)依靠将一个子程序试图连接到其他子程序而获得，或 *b*)由程序配置系统用户直接输入出口名称而提供，例如在键盘上将出口名称敲入。此外，根据本发明的一个方面，出口可具有一个或更多约束参数而得到扩充，这些约束参数可用于保证这些连接只建立起合适的关系。这些约束参数包括：*a*)所允许的连接数量，*b*)所允许的连接种类，*c*)目的对象的类，和 *d*)所需要的目的对象操作。

附图中：

图 1、2、3、6 和 7 显示出给本发明的计算机程序配置系统和方法的用户所展现的屏幕显示；

图 4 和 5 分别显示出图 3 和 6 的屏幕显示的一部分；

图 8 至 13 显示用于实现本发明的过程的一部分的流程图；

图 14 显示根据本发明所存储信息的表格；

图 15 是实现本发明的系统和方法的计算机的典型硬件配置的框图；

图 16 显示出图 6 的屏幕显示的一部分；以及

图 17 显示出子程序，它们的出口和出口属性之间的关系。

虽然本发明的程序配置系统可用于使用任何数量的编程语言的计算机上，但最合适的情况是和运行面向对象的编程语言的计算机一起使用，本发明的最佳实施例是为执行用 *C* 编程语言编写的程序

的计算机所设计的，因此，下面的讨论将至少部分地适合于面向对象的编程，尤其适合于面向对象的C编程。然而应该理解本发明并不只局限于运行面向对象编程语言的系统。

本发明的程序配置系统是一个自包含实用系统，它和将被配置的程序是分开的。假定任何被配置的计算机程序所需的部件函数、对象、和子程序在和程序配置系统一起使用之前都已编写并在一般情况下能够执行。

要理解下面的解释，必须定义几项术语。

一个“对象”是一个由一个或更多个在一个特定的子程序内一起定义的数据结构和有关的操作所组成的自包含群体。

一个“连接”是一个由程序配置系统建立的对象，用于获取“源”对象和“目的”对象间的关系细节。连接是在程序配置过程中建立的，它们在程序执行期间被其他对象使用。连接对象通常获取图14所示的关系细节，这在下面还将解释。现时发明的程序配置系统的关键功能是如适当的对象间交互作用所需要的那样为对象间连接的定义提供方便。

一个“出口”是一个用于提供信息的数据结构或对象，该信息是在从一特定“源”对象到一个或更多个“目的”对象建立连接时所需要的。根据本发明的一个方面，出口是在特殊设计用来与其他对象交互工作的“源”对象内所定义，或由“源”对象的功能操作所建立。借助于出口的名称或相关的出口变量的名称，每个出口向程序配置系

统的用户提供有关“目的”对象的种类的标帜，而源对象必须与该“目的”对象连接才能发挥作用。除名称之外，根据本发明的一个方面，出口可具有一个或更多约束参数而得到扩充，这些约束参数可用于保证这些连接只建立起合适的关系。这些约束参数包括：*a*)所允许的连接数量，*b*)所允许的连接种类，*c*)目的对象的类，和 *d*)所需要的目的对象操作。

决定于应用程序的设计，一定的对象为了要发挥功能，就要求它们出口的初始集合在程序配置时加以扩充(例如增加数量)。根据本发明的原理，当由来自程序配置工具包的信息所激励时，为扩充出口的初始集合而提供附加的出口的对象就进行扩充操作。用此种方法附加到对象的出口的原始集合的出口在这里称作“获取的”出口，并用程序配置工具包个别地赋予名称。通过从合适的“目的”对象的功能取得一串名称和/或通过提示用户手动输入所希望的出口名称，例如用键盘，程序配置工具包可用于自动地为获取的出口命名。

根据本发明，获取的出口 *a*)由程序配置系统显示和注解，*b*)由用户使用程序配置系统加以改变，以及 *c*)可以具有对用户有特殊意义的名称，例如，目的电子数据表子程序的范围名称，或数据库特性。

为了有利于解释子程序，它们的出口、和出口属性间的关系，读者可将注意力放在图 17。为简明起见，也只为了下面图 17 的解释，出口将显式地看成出口对象。一个子程序对象 1701 可能有例如指针指向一个或更多出口对象 1702。每个出口对象 1702 有至少一个

名称 1703，并 可能有例如指针指向一个或更多被命名的连接类型对象 1705。每个连接类型的对象 1705 有一个名称 1706 和不同的参数，例如，这些参数用于描述 *a*)所需要的连接数量 1707，*b*)目标子程序所需要的方法 1708(这里用作技术术语，即功能子程序)，*c*)所需要的目标子程序的类 1709，以及 *d*)一个用于描述出口的可能用途的标识参数 1710，例如出口是否可能用于从一个目标子程序获取其他出口名称的目的。

现转向图 1，其中显示了计算机显示的图象，在下列情况时能看到该显示：1)用户进入本发明的程序配置系统，2)为所要配置的程序将程序文件对系统加以标识，和 3)为了用于所配置程序将所需部件子程序和图形接口域加以标识。此种情况下，命名为 *MY—PROGRAM* 的所配置的程序是一个简单程序，用于将两个数 *X* 和 *Y* 作为输入量和将它们的和数作为结果 *r* 加以显示。显示 10 显示了程序配置系统。文件窗口 13 包括一个代表 *MY—PROGRAM* 的图形单元 130，代表 *MY—WINDOW*14 的图形单元 131，代表一个称为 *FUNCTION*—1 的子程序的图形单元 133，和代表另一个命名为 *CONNECTIONS*—1 的子程序的图形单元 134。在图形单元 131 上按鼠标，用户即可打开或关闭 *MY—WINDOW*14。

用户已定义 *MY—WINDOW*14 为包括 1) 前面标以文字标号“*X*:”的域 140，2)前面标以文字标号“*Y*:”的域 141，3)前面标以文字标号“*r*:”的域 142，和 4)带有标号“+”的按钮 143。*MY—WIN-*

*DOW*14，域 140—142 和按钮 143 由用户按下法建立：用户从调色板/工具箱 12 中选取合适的项目，并把所选项目复制到显示 10 上所希望的位置上。应用本程序时可将代表鼠标光标的箭头 20 移动至域 140 并按鼠标按钮，这就允许输入数值 *X*，接着对域 141 进行同样操作，将数值 *y* 输入，再将箭头移至按钮 143 并按鼠标，这样的操作将 *MY—PROGRAM*130 中所用子程序—本例中是 *CONNECTIONS*1 134 和 *FUNCTION*—1 133—加以结合，以便计算 *r* 值并将它在域 142 中显示。

根据本发明的原理，要使这样一个程序运行的一个方法是将包括为用于较大程序而编写的对象的子程序分别加以配置，将它们连接在一起并和一个用户接口子程序连接起来，该用户接口子程序能提供一个具有一定数量用户接口域—例如 *MY—WINDOW*14 的域 140—143—的显示。*FUNCTION*—1 133 是一个这样的可配置的子程序对象，在本例中它是一个特定的电子数据表子程序对象，它提供用于计算特定类型的和数所需全部程序逻辑。“*CONNECTIONS*—1”134 是另一个这样的子程序对象，它是一个连接函数，能提供同步触发功能。一旦所有子程序，例如用于控制 *MY—WINDOW*14、*FUNCTION*—1 133、*CONNECTIONS*—1 134 的代码，都已编写完，它们可连接在一起以定义一个完整的可运行的程序。要做到这点，必须在调色板/工具箱 12 中提供一个用于代表称为“*FUNCTION*—1”和“*CONNECTIONS*—1”的子程序的图形图象，

并至少一次将它们每一个复制到上面讨论的程序文件窗口 13 中去。一般讲来，和调色板/工具箱中所代表的其他函数一样，*FUNCTION*—1 133、*CONNECTIONS*—1 134 事先必须至少部分编写好，以便在它们的图象存入调色板/工具箱 12 时每个这样的函数或子程序都是可执行的。熟悉技术的人知道怎样制做这样的子程序。

为建立所需连接，程序配置系统按下列示范方式工作：

为特定地将子程序 *CONNECTIONS*—1 134 和子程序 *FUNCTION*—1 133 相联，用户操作如下：为了选用 *CONNECTIONS*—1 134 将箭头 20 移至文件窗口 13 中它的图形表示处，并在按下计算机键盘的控制(*CTRL*)键的同时按鼠标按钮。虽然在图 1 中没有其他子程序，有时会有不止一个其他子程序要加以选择。如图 2 所示，按着鼠标按钮不放—控制键可以放开—，用户移动鼠标来将箭头 20 移向感兴趣的特定子程序处，从最初选用的子程序表示如 *CONNECTIONS*—1 134 画一直线 21。当该线抵达感兴趣的子程序如 *FUNCTION*—1 133 时，用户放开鼠标按钮，结果就在所增亮的两个子程序之间建立了连接。图 3 中显示了这样的增亮的连接。如果检查员窗口 30 没有在以前的活动中显示出来，则这时候该窗口 30 即在显示 10 上出现。

图 4 显示检查员窗口 30 更详细的图象。在所示操作模式即已知为链接或“连接”模式中，检查员窗口的功能是 *a*)在子窗口 400 中显示上一次所选子程序如 *CONNECTIONS*—1 134 的出口，*b*)在

另一个子窗口 401 中显 示目标子程序如 *FUNCTION*—1 133 的合适的可执行函数，而其他子程序或图形单元可以和它连接起来，c)显示从所选出口至所选子程序所允许的连接类型（如图 5 中 *TYPE*—1 501），以及 d)便于完成这种链接或连接。标号为“出口”的子窗口 400 列举了所选子程序即本例中的 *CONNECTIONS*—1 134 的初始变量。标号为“目的处的动作”的子窗口 401 列举了可用于连接目标子程序函数程序，在本例中是 *FUNCTION*—1 133 内可用的一套适当的函数。

正如上面所蕴含的，检查员窗口 30 具有几种不同的操作模式。检查员窗口 30 包括模式子窗口 425，后者显示出检查员窗口 30 的当前操作模式。如图 4 和 5 所示，检查员窗口 30 处于“连接”模式。图 16 显示处于“ 属性” 模式的图 5 的检查员。检查员窗口 30 还有“按钮“423、424 和 427，当按鼠标时，它们中每一个都将一个特定函数加以初始化。当前标以“连接”的按钮 423 可用作连接和断开两种函数，因此有时候如图 5 中更详细地描述的和显示的那样，标号为“断开”。

用户将箭头 20 放置于出口名称上方并按鼠标按钮、即可在子窗口 400 中选择一个出口。用户接着可以将所选出口和目的子程序内可用的动作加以联接。这是通过将箭头 20 放置于子窗口 401 中的描述字上方并再次按鼠标按钮而完成的。如果一个建议的连接有可能规定不止一种连接类型，可用的类型列举于子窗口 420 中。用户将

箭头 20 放置于所希望的连接类型名称上方，并按鼠标按钮，即可选择或改变连接类型。

将箭头 20 放置于“连接”按钮 423 上并按鼠标按钮，即可在所选子程序和 目标子程序的有关动作之间完成所选类型的连接。在本例中，在程序运行期间，当子程序 *CONNECTIONS*—1 134 中以前所定义的程序指令的顺序操作使该子程序向另一个由所连接出口所指向的子程序发送一个动作信息时，检查员窗口 30 中为该连接所选的特定动作信息将送至本例中的 *FUNCTION*—1 133。

根据本发明，用户不时地可能在任意选择的源和目的对象之间规定一个所希望的连接。如上面所描述那样，通常做法是将箭头 20 放置在屏幕的源对象表示上方并如上描述画条线到某些目的对象，然后放开鼠标按钮。每次当用户操作时，系统将显示同源对象相关联的检查员窗口 30 并在检查员窗口 30 内列举源对象的出口变量名称。当检查员窗口 30 中特定出口被选择和/或增亮时，系统从目的对象获取全部所需信息，从决定 *a*)何种连接类型可用于连接至目的对象，*b*) 多少个连接可连至目的对象，以及 *c*)何种目的对象函数可由从所增亮出口到目的对象的信息所触发。从目的对象如此获得的信息用于决定在检查员窗口 30 中列举何种函数和连接类型，作为选项供用户选择。根据本发明的一个方面，用户被系统限制为只能选择那些对所选目的对象适用的连接的类型和数量。有利的是，这些连接将只触发目的对象中那些有意义的函数。

根据本发明，一定子程序中的“源”出口用于从一个或更多个目的子程序获取和使用附加出口。当这种特定类型的源子程序和出口连接至合适的目的子程序时，这些源子程序将获取附加出口，然后也列举在子窗口 400 中。另外，当那些由一定的子程序所获取的源出口连接到合适的目的子程序时，这些源出口也可用于获取附加的源出口。如图 5 所示，所获取出口从被获取时起即列举于子窗口 400 中并一直停留在那里，直至遇到下列情况之一：或其他一些源对象被选择，或那些促使所获取对象被获取的出口本身被断开。在子窗口 400 内，*a*) 符号“@”500 紧挨着任何所获取出口列举出来以及 *b*)符号“+”406 紧挨着任何在所选子程序的上下文中用来或可能用来从目的子程序中获取源出口的出口列举出来。星号符号“*”404 也紧挨着任何 用 在源和目的子程序间的一个或更多连接内的出口列举出来。

根据本发明，用于从众多源变量同步地复制数据到一个目标对象时所必需的对象特别适合于使用所获取出口。本例中，众多用户接口对象连接至由 *CONNECTIONS*—1 134 所代表的对象，后者接着又连接到对象 *FUNCTION*—1 133，———一个特定的电子数据表。对象 *CONNECTIONS*—1 134 具有能力通过特定命名为“*SPREADSHEET*”的出口从任何它所连接的电子数据表中获取附加出口。如此获得的出口被理解为目标电子数据表的范围名称。一旦获得这些出口，所获取出口可用于连接至其他对象，这包括但不局

限于提供用户接口的对象。然而这些所获取出口，以及特别是传送给与之连接的其他对象的信息，仍然处于进行获取的对象的控制之下。因此，进行获取的源对象随时提供送往目的对象的信息同步操作，例如，该同步操作可由被用户接口所捕获的单个按鼠标操作所产生的送往目的对象的单个信息所触发。

相似地，设计用于连接到数据库模块的对象也可能随时准备获取对应于数据库特性的出口。例如，已经获取了代表数据库特性的出口的对象对于收集那些形成数据库查询所必需的数据以及使收集操作同步化特别有用。

根据本发明的一个方面，对一定类型的进行获取的对象讲，如图16所示，当检查员处于“属性”模式时，用户可以直接将出口名称输入到检查员窗口30中去(图6)，从而规定将要获取的出口。任何时候只要检查员窗口30以属性模式显示出来，用户就可以选用出口列表921中的空白区并键入所希望的出口名称。另一种方法是，用户可在列表内选用已有的所获取的出口，并在所增亮的名称上键入，从而改变出口名称。用这种方法获取出口的对象在提供多重同步化的逻辑、算术或语法比较时特别有用。例如，借助于这种方法可由一个对象获取多个出口，可由一个事件或由其他对象来的一个信息触发多个单独的但同时的比较，并且单一的结果可送至一个或更多其他对象。

图5显示出检查员窗口30，它是在一个目的子程序例如图1中

FUNCTION—1 133 的变量自动地和直接地加以获取并由所选子程序例如 *CONNECTIONS*—1 134 的功能子程序所采用，然后又在子窗口 400 中显示之后显示的。用户将箭头 20 移动至所希望的变量处，例如图 5 中的变量 *A*1*B*1 407，再按鼠标使它增亮。如图 6 所示，一旦一个变量被选增亮，以及当箭头 20(图 1)位于和当前检查员窗口 30 相关联的子程序图形表示上方时，用户再按一个鼠标，并画另一条线 111(图 6)至另一个子程序，或如本例中显示上的接口域 140。

当鼠标箭头接近程序用户接口中所定义的任何域或文件窗口 13(图 1) 中任一其他子程序时，由情况决定，目标域或子程序将增亮。当一个域或子程序如此增亮而用户放开鼠标按钮时，在源和目的对象间画了一条可见的连线，用于显示未决逻辑连接。当未决逻辑连接被显示时，*a*)相对应的源出口在子窗口 400(图 4)中得到增亮，*b*)目的处所选动作在子窗口 401 中得到增亮，*c*) 所选连接类型在子窗口 420 中得到增亮，以及 *d*)如果它是一个早已存在的连接，它将和从其他所选对象出口来的其他连接一起列举于子窗口 419 中。当用户将箭头 20 放置于 *a*) "连接类型"子窗口 420 中的字 510 上方，*b*)"源出口"子窗口 400 中的字 407 上方，或 *c*)"目的处动作"子窗口 401 中字 403 上方时，他只要按鼠标按钮，即可进一步标明或改变所建议要建立的连接的类型。

如所建议连接得到证实，只要按"连接"按钮 423，该连接将 *a*)

被建立起来以及 b)在子窗口 419 内列举出来，并有增亮的显示。当一个出口第一次连接时，一个像星号 404 那样的指示符将在子窗口 400 中紧挨着出口名称显示出来，标志着该出口至少已连接过一次。如图 5 所示，当一个特定连接被选择并在子窗口 419 中增亮时，连接/断开按钮 423 的标号即改变为“断开”。每个所选连接和它的类型将一直停留在子窗口 419 的列表中，直至当箭头 20 位于特定连接上方时按鼠标和又一次在标号为“断开”的按钮 423 上按鼠标。

当检查员窗口 30 第一次打开时，如在子窗口 400 中已列举了连接好的出口名称，例如从前次连接操作来的出口名称，则每个所连接出口都将有星号 404 显示出来。如用户选用一个早就连接好的出口名称，像线 111(图 6)那样的线将显示出来，表示连接列表中和所选出口到一个接口域或子程序有关的第一次连接。当箭头 20 位于子窗口 419 中其他连接上方而用户按鼠标按钮时，将显示其他线。这样可以检查连接，或确认它们的正确性，或找出它们是什么。如用户已画了线 111(图 6)，意图是连接一个尚未连接的出口，但欲错误地将一个连接到其一个其他图形表示的出口加以增亮，则线 111 将会消失，并将被一条线所代替，该线是在该出口连接的域或子程序与例如 *CONNECTIONS*—1　134 那样的所选子程序之间画出的。在一个实施例中，在将所选出口 407 连接至其他域或子程序之前，用户必须重新画制线 111。可以应用一个“废除”函数，它使早已连好的出口减亮并自动恢复线 111。再者，任何时候只要有一个连接被显示按

钮 423 将用“断开”为标号，这时用户有一个选项供选择，或在该按钮上按鼠标，从而将出口断开，或不理睬“断开”按钮而选择另一个出口、连接、连接类型、或动作。

根据本发明的一个方面，如果某特定子程序的任何出口用于从它可能与之连接的目的子程序中获取附加出口，则检查员子窗口 400 在列举该出口名称时同时显示一个特殊显示符，如符号“＋”406。这就告知用户，该特定子程序在连接至一个或更多个其他共同合作的子函数时将会获取附加出口。

根据本发明的一个方面，如果从一个特定子程序例如 *CONNECTIONS*—1 134 中的一个出口建立连接，而该特定子程序是用于从它与之连接的目的子程序如 *FUNCTION*—1 133 中获取附加出口的，则附加出口名称被自动获取并在同一子窗口 401 中和符号“🔗”500 一起列举出来。这提醒用户，所获取出口和它们的连接依赖于某些已经建立的连接。

在本发明的一个实施例中，所有出口名称都有一套和它们相关联的“连接类型”，因此，如上所述，在建立连接之前都为每个连接选择一个特定的连接类型。当一个特定的出口在子窗口 400 内增亮时，它的当前所选连接类型在子窗口 420 内列举并增亮。用户要改变选择时，可将箭头 20 置于子窗口 420 中所需连接类型 510 的字上方，同时按鼠标按钮。

上述过程可以一再重复，直至其他子程序内的所有域或出口都

连结到其他所需子程序的所需出口和动作。本例中，三个域 140、141、142 都连接到子程序 *CONNECTIONS*—1 134 内的出口 *A1B1*、*A2B1* 和 *A3B1*，而子程序 134 本身又有 4 个出口 *A1B1*、*A2B1*、*A3B1*、和 *SPREADSHEET* 连接至 *FUNCTION*—1 133。此外，域 143 的出口也连接至子程序 *CONNECTIONS*—1 134 中的一个动作，而子程序 134 本身又触发所连接的 *FUNCTION*—1 133 的所需操作。

在本发明的一个实施例中，只有指定的“出口”的变量或函数才能连接至子程序。因此，当在目的子程序中要起动一个动作时，这由送至一个出口的信息所触发，而该出口支持携带动作信息给目的子程序的连接的类型。如图 5 所示，在动作连接过程中一个出口能送给目的子程序的特定动作信息在子窗口 400 列举出来。所有由增亮的出口名称所支持的连接的类型都在连接类型窗口 420 中列举出来。

正如在所有出口的情况下那样，一个星号或其他标示符(未示出)紧挨着所连接动作的名称显示出来，而当两者都增亮时，按钮 423 变为“断开”按钮。类似地，如连接动作被增亮，则现有的到子程序的连接将显示出来。

本例中，根据本发明的一个方面，当箭头 20 位于按钮“+”143 上方时按鼠标按钮，即触发子程序 *CONNECTIONS*—1 134，以提供适当的动作信息来触发 *FUNCTION*—1 的操作。为建立此连接，程序配置系统的用户将箭头 20 放置于按钮“+”上方，并按鼠标按

钮，以选择代表按钮“＋”的函数的图形单元143，然后画一条线（未示出）到程序文件窗口13中的子程序*CONNECTIONS—1* 134的图形单元。如图7所示，当放开鼠标按钮时，两个图形单元间的连接211将显示出来。在建立该连接后，一个和检查员窗口30相类似但和第一个所选图形单元—本例中是按钮“＋”143—相关连的检查员窗口（未示出）显露出来，并且如上面描述那样可用于将从“＋”域子程序中定义的出口到子程序“*CONNECTIONS—1*”134中定义的动作的连接给予建立、改变或检查。

参照以上所述，在程序早已存在的例子中，通过被认为是工作空间管理程序的操作系统的功能，该程序的特性和子程序可供本发明的程序配置系统使用。图1示出，工作空间管理程序提供了窗口15，其中显示了不同可用文件的表示。用户通过*a*）在工作空间管理程序窗口15中增亮早已存在的程序例如*MY—PROGRAM*的图形表示，*b*）把它拉向文件窗口13，以及*c*）将它放在那里，可以让程序配置系统了解早已存在的程序如*MY—PROGRAM*的特性。

当一个程序配置完毕和用户通过从菜单（未示出）选择“存放”项来表示配置过程的结束时，程序配置系统在一个文件中存储和每个子程序有关的全部编程对象的列表，并且如果有的话，图形用户接口，和连接列表。对于每一个由用户规定的连接，系统将图14所示表格501的复制件作为程序配置文件的一部分加以存储。每一个这样的表格501复制件包括*a*）以不同连接类型列表中索引形式出现的

连接域的类型(类型)502,*b*)都以指向对象列表中正确对象的指针形式出现的源子程序域(源)503和目的子程序域(目的)504两项,以及*c*)出口域505中的出口变量名称和动作域506中的动作函数名称(如有的话)。另外,对于每一个由用户建立的连接,系统在表格501中的"所有者"域507中存放用来获取和特定的连接(如果有的话)有关的"出口"域505中所命名的出口的出口变量名称。当包括一个或更多个表格501的程序配置文件存放以后,本发明的程序配置系统的工作即完成。

程序配置系统提供了一个程序,其第一行是一条指令,用于装载并执行程序配置文件。执行配置文件的结果是:装载正确文件;部分依据于一个或更多表格501来建立恰当连接;以及显示由于不同子程序和它们互相间的连接之间的相互作用所提供的正确功能。

图8至11显示了今后称之为主连接过程的这一过程的流程图,在步骤60处,当*CTRL*键按下而鼠标箭头20位于如*FUNCTION*—1 133和*CONNECTIONS*—1 134这样的子程序表示上方或位于如按钮143这样的屏幕按钮对象上方时,如用户按鼠标按钮就能引用该主连接过程。条件转移点61用于测试确定箭头是否移动过。如条件转移点61测试结果是*YES*,这标志着箭头已移动过,接着条件转移点62测试确定是否有早已存在的连接线。如步骤62的测试结果是*YES*,这标志着有早已存在的连接线,因此在步骤63将这些连接线从屏幕上清除。如步骤62的测试结果是*NO*,或者执行

完步骤63后，控制即交给步骤64，此早显示出如*CONNECTIONS*—1　134那样的所选子程序和箭头20之间的新连接线。接着控制又交回给步骤61，以便测试进一步的箭头移动操作。

如步骤61的测试结果是*NO*，这标志着箭头没有移动过，控制即交给条件转移点65，在那里测试确定鼠标按钮是否已经放开。如果步骤65的测试结果是*NO*，这标志着鼠标按钮尚未放开，则控制又交回给步骤61去继续检查箭头移动操作。如步骤65的测试结果是*YES*，这标志着鼠标按钮已经放开，控制交给条件转移点66，在那里测试确定箭头20是否位于某一类型对象上方，如连接线从子程序*CONNECTIONS*—1　134出发，则该子程序134可连接至这一类型对象，如连接线从143处所用按钮出发，则该按钮可连接至这一类型对象。

如步骤66的测试结果是*NO*，这标志着鼠标按钮已放开，但箭头20尚未放置于可与之建立连接的对象的上方，于是在步骤67中将任何已显示的连接线加以消隐，并且主连接过程在步骤68中结束退出。系统接着等待用户选择另一任务。

如步骤66的测试结果是*YES*，这标志着箭头位于可与之建立连接的对象上方，于是控制交给步骤69，其中围绕该对象显示一个方框，因而将它增亮。在步骤69后或67后的任一种情况下，条件转移点600测试确定检查员窗口30是否可见。如步骤600的测试结果是*NO*，这标志着检查员窗口30不可见，则在步骤601使它变为

可见。如步骤600的测试结果是*YES*，或在执行步骤601后，控制交给条件转移点602(图9)，其中测试确定检查员窗口30是否处于连接模式。如步骤602的测试结果是*NO*，则检查员窗口30不处于连接模式，控制交给步骤603，在步骤603检查员窗口30设置为连接模式。如步骤602的测试结果是*YES*，或在执行步骤603后，控制交给步骤604，其中在检查员窗口30的左列内显示出可用的程序出口。

接着，条件转移点630测试确定由步骤604显示的任何出口是否早已连接。如有任何出口早已连接，条件转移点631测试确定所有可能获取的出口是否在检查员子窗口400(图5)中列举出来。如步骤631(图9)的测试结果是*NO*，控制交给步骤632，在步骤632所有已获取但未列举的出口都加到列表中去。接着控制交回给条件转移点631，以测试所有可能获取的出口是否都已列举。

在可能的几个周期后，当所有出口和已获取出口都已标识和列举后，步骤631的测试结果成为*YES*，控制即交给步骤633，其中任何所列举出口只要用在连至一个或更多个目的子程序的一个或更多个连接中，它们即被标以星号404(图5)。根据本发明的一个方面，在步骤634(图9)中，紧挨着所获取出口的名称标以符号"@"500(图5)。以后，在步骤635(图9)中，从所选源子程序或域来的所有连接的列表在子窗口400(图5)中显示出来。如步骤630(图9)的测试结果是*NO*，或在执行步骤635后，控制交给步骤636，其中对于用来从目

的子程序中获取附加出口的所显示的出口，不论它是否连接了，都紧挨着它们的名称标以符号“＋”406(图 5)。其次，在步骤 638(图 9)中第一个由步骤 604 或 632 列举的、可用于建立连接的出口得到增亮。根据本发明的一个方面，在步骤 637，由子窗口 400(图 5)中增亮的出口所支持的连接类型的列表在子窗口 419 中显示出来。

如图 10*A* 所示，控制交给步骤 607，其中测试确定鼠标按钮是否按下。如在步骤 607 处(图 10*A*)鼠标按钮没有按下，系统在步骤 607 处继续等待按鼠标。如步骤 607 处测试结果是 *YES*，这标志着鼠标按钮已按下过，控制交给条件转移点 608，其中测试确定箭头 20(图 5)是否位于检查员窗口 30 内。如箭头 20 不在检查员窗口 30 内，步骤 608(图 10*A*)的测试结果是 *NO*，并在步骤 609 将所有连接线从屏幕上取消。然后在步骤 610 主连接程序结束并退出。

如步骤 608 的测试结果是 *YES*，这标志着箭头 20(图 5)位于检查员窗口 30 内，则条件转移点 611(图 10*A*)测试确定箭头 20(图 5)是否位于检查员窗口 30 的右列 401 中目的动作名称上方。如步骤 611(图 10*A*)的测试结果是 *YES*，这标志着箭头位于检查员窗口的右列中一个动作名称的上方，则控制即交给步骤 612，其中箭头 20(图 5)下的动作被增亮，而任何以前增亮的动作都被减亮。

如步骤 611(图 10*A*)的测试结果是 *NO*，这标志着箭头 20 不位于检查员窗口 30 的右列 401(图 5)中动作名称上方，则控制即交给条件转移点 645(图 10*A*) ，它测试确定箭头 20(图 5)是否位于连接

类型子窗口 420(图 5) 的连接类型名称上方。如步骤 645(图 10*A*) 的测试结果是 *YES*,这标志着箭头 20(图 5)位于连接类型名称 510 (图 5)的上方,则控制交给步骤 643(图 10*A*),其中系统将子窗口 420(图 5)中的连接类型增亮,而将任何其他可能已被增亮的连接类型减亮。

如步骤 645(图 10*A*)的测试结果是 *NO*,这标志着箭头 20(图 5) 不是位于子窗口 420 中连接类型名称上方,则控制交给条件转移点 670(图 10*A*) ,它测试确定箭头 20(图 5)是否位于检查员模式按钮 427 上方。如步骤 670(图 10*A*)的测试结果是 *YES*,这标志着箭头 20 (图 5)位于检查员模式按钮 427 上方,则控制交给步骤 700(图 10*B*),其中将模式加以改变。

如步骤 670(图 10*A*)的测试结果是 *NO*,则控制交给条件转移点 646,它测试确定箭头是否位于左列子窗口 400(图 5)中出口名称上方。如步骤 646(图 10*A*)的测试结果是 *YES*,这标志着箭头 20(图 5)位于出口名称上方,则控制交给步骤 644(图 10*A*),它将出品增亮,并将其他任何增亮出口加以减亮。接着在步骤 642,该出口的有效连接类型和已增亮的目的均列举出来。

在步骤 642 列举有效出口后,或在步骤 643 将连接增亮后,控制交给步骤 641,其中所增亮出口的任何有效目的动作和连接类型均在子窗口 420(图 5)中列举出来。在步骤 641(图 10*A*)中列举有效目的动作之后,或在步骤 612(图 10*A*)中将箭头 20(图 5)下的动作增

亮之后，控制交给步骤640，它将按钮423(图5)上的标号设为“连接”，并将子窗口419中可能已被增亮的任何连接加以减亮。步骤640之后(图10*A*)，控制交回给步骤607，以等待再次按鼠标按钮。

如步骤670(图10*A*)的测试结果是*YES*，这标志着当箭头20(图5)位于检查员模式按钮427(图5)上方时按了鼠标按钮，则控制交给步骤700(图10*B*)，其中如图16所示，检查员模式切换至“属性”模式。

在步骤700(图10*B*)中将检查员模式由“连接”模式切换至“属性”模式后，控制交给步骤701(图10*B*)，控制交给步骤701(图10*B*)，其中为所增亮对象显示属性检查员模式，以代替连接模式检查员30(图5)。其次，条件转移点702(图10*B*)测试确定增亮的对象是否已获取了出口。如步骤702(图10*B*)的测试结果是*YES*，这标志着该对象已获取了出口，则在步骤703中这些出口在检查员窗口的出口窗口920(图16)中列举出来。

在步骤703之后，或如步骤702的测试结果是*NO*，则条件转移点704测试确定当箭头20位于属性检查员上方时鼠标按钮是否按下过。如步骤704的测试结果是*NO*，鼠标按钮并未按下过，则控制仍停留在步骤704，同时所列举出口继续得到显示，直至鼠标按钮按下时止。如步骤704的测试结果是*YES*，这标志着当箭头20位于检查员模式按钮927(图16)上方时，鼠标按钮按下了，则控制交给条件转移点602(图9)，其中检查员模式切换回至连接模式。如步骤

704(图 10*B*)的测试结果是 *NO*，这标志着当按鼠标按钮时箭头 20 不位于检查员模式按钮上方，控制交给步骤 706，它测试确定箭头 20 是否位于所获取出口 921(图 16)的列表中空白区上方。如步骤 706(图 10*B*)的测试结果是 *YES*，这标志着当箭头 20 位于列表中空白区上方时按下鼠标按钮，则控制交给步骤 707，其中将空白区增亮。如步骤 706 的测试结果是 *NO*，控制交给条件转移点 711，它测试确定光标是否位于图 16 所示检查员窗口上所获取出口 921 的列表中一个所列举的出口名称 920(图 16)的上方。如步骤 711(图 10*B*)的测试结果是 *NO*，这标志着箭头 20 不是位于所列举出口名称上方，则控制交回给步骤 701，以等待再一次按鼠标按钮。如步骤 711 测试结果是 *YES*，这标志着箭头 20 位于所列举出口名称上方，则控制交给步骤 712，其中所选出口名称得到增亮，并在步骤 713 去检测所增亮出口。在步骤 713 或步骤 707 之后，在步骤 708 中接收一个用户提供的输入，例如键盘输入，用于定义一个新的出口名称。在条件转移点 709 中所输入出口名称的语义是要测试是否有效，也即要检查它是否为一有效名称(例如并不是全空白)。如步骤 709 的测试结果是 *NO*，名称无效，于是控制交回到步骤 704，去等待再次按鼠标按钮。如步骤 709 的测试结果是 *YES*，则步骤 708 中由键盘输入的出口名称有效，于是在步骤 710 中送信息至增亮的对象以获得取所命名的出口。在步骤 710 中增亮的对象获取该出口后，控制交回至步骤 701，去等待再次按鼠标按钮。

如步骤646(图10A)的测试结果是*NO*,这标志着箭头20(图5)不是位于出口名称上方，则控制交给条件转移点647(图11),它测试确定箭头20(图5)是否位于连接显示子窗口419中一个连接的上方。如步骤647(图11)的测试结果是*YES*,这标志着箭头20(图5)位于连接显示子窗口419中一个连接的上方,则控制交给步骤650(图11),其中系统将所有连接线都消隐掉,并重画所选连接的线条,以此来只是增亮所连接子程序的源和目的图形表示。接着在步骤651中，系统将子窗口400(图5)中已增亮连接的出口加以增亮,而将所有其他已增亮的出口加以减亮。

步骤652(图11)中,所有被选出口的允许的连接类型都在子窗口420(图5)中列举出来,并在步骤653(图11)中,所选连接的连接类型得到增亮，而所有其他连接类型加以减亮。其次在步骤654中，在子窗口401(图5)中显示出由所选出口至所选目的连接的类型所允许的动作。此后,在步骤655(图11)中,所增亮连接的所选动作得到增亮,而显示中所有其他以前增亮过的动作则都减亮。接着在步骤656中,*CONNECT*按钮423(图5)上的标号设置为“断开”(“*DISCONNECT*”)。接着控制交还给步骤607(图10*A*),去等待另一次按鼠标。

如步骤647的测试结果是*NO*,这标志着箭头20不是位于连接显示中的任一连接上方，则控制交给条件转移点616,它测试确定箭头20是否位于连接按钮423(图5)上方。如步骤616的测试结果是

NO,这标志着箭头20不是位于连接按钮423(图5)上方,则控制交还给步骤607(图10*A*),去等待按鼠标。如步骤616的测试结果是*YES*,则控制交给步骤617,它分别引用图12和13所示的连接删除/建立过程。

图12和13一起组成了图11中步骤617所引用的连接删除/建立过程的流程图。当箭头20位于检查员窗口30的"连接"("*CONNECT*")按钮423(图5)上方,并且按鼠标按钮时,在步骤80(图12)处即进入该过程。接着条件转移点81(图12)测试确定在连接显示列表419(图5)中是否存在一个所选连接。

如步骤81(图12)的测试结果是*NO*,这标志着不存在增亮的已选连接,则控制交给条件转移点82(图13),它测试确定是否有一个增亮的出口将要连接。如步骤82的测试结果是*NO*,这标志着没有增亮的出口要连接,系统回至步骤607(图10*A*),去等待另一次按鼠标。如步骤82的测试结果是*YES*,这标志着存在一个增亮的出口,则控制交给条件转移点83,它测试确定对某个存在的连接是否早已定义了*a*)增亮的出口,*b*)增亮的连接类型,*c*)增亮的目的,和*d*)增亮的动作这四项的组合。如步骤83的测试结果是*YES*,这标志着早已存在着这样一个连接,则控制交给步骤84,其中有信息提醒用户,说明在请求一条冗余连接。控制即交还给步骤607(图10*A*),去等待另一次按鼠标。

如步骤83(图13)的测试结果是*NO*,这标志着没有冗余连接,

则控制交给条件转移点850，它测试确定和增亮的出口相关联的参数是否允许对增亮的目的作一连接。本实施例中，在条件转移点850，和每个出口相关联的参数不准程序配置系统为不恰当目的建立连接。特别不准为没有所需操作的目的对象建立连接。如步骤850的测试结果是*NO*，这标志着这些出口参数排除对所选目的建立所选类型的连接，则控制交给步骤85，其中向用户显示一条约束破坏信息，说明在出口参数、连接类型、和目的对象操作之间蕴含着矛盾。控制接着交还给步骤607(图10*A*)，去等待另一次按鼠标。

然而，如步骤850(图13)的测试结果是*YES*，这标志着增亮的出口和增亮的连接类型的参数允许对增亮的目的对象建立一个连接，则控制交给步骤851，其中建立一个与增亮动作关联的从增亮出口至增亮目的的所选类型的新连接。还有，在子窗口400中紧挨着出口名称显示一个星号404(图5)。其次，在步骤852(图13)中，新连接的细节附加到子窗口419(图5)中的连接显示中去。

此后，条件转移点853(图13)测试确定增亮的新连接是否用于获取出口。如步骤853(图13)的测试结果是*YES*，这标志着新连接用于从目的子程序或域获取新出口，则控制交给步骤854，在那里系统使所选源对象去获取附加出口，并在检查员窗口中将这些出口连同紧挨着每个这种出口所显示的符号"@"一起列举显示出来，以标志每一个这样的出口是一个获取的出口。

如步骤853的测试结果是*NO*，这标志着系统发现新连接不是

用于任何获取的出口，或如在执行步骤 854 后，则控制交给步骤 855，其中如图 5 所示，按钮 423（图 4）的标号改变为“断开”（“*DISCONNECT*”）423。于是控制交还给步骤 607（图 10*A*），去等待一次按鼠标。

如步骤 81（图 12）的测试结果是 *YES*，这标志着在连接显示列表 419（图 5）中存在着增亮的连接，则控制交给步骤 810（图 12），其中开始断开所选连接和更新显示的过程。在步骤 810 中该过程继续进行，其中所选连接从用于所选出口的连接列表中被清除掉。其次在步骤 811 中，检查员窗口 30（图 5）中的子窗口 400（图 5）被清除。在步骤 812（图 12）中，由所选源对象提供的出口列表在子窗口 400（图 5）中重新加以显示。

条件转移点 830（图 12）测试确定步骤 812 中显示的任何出口是否早已连接。如存在这种早已连接的出口，步骤 830 的测试结果是 *YES*，同时控制交给条件转移点 831，它测试确定是否所有可能的获取的出口都在检查员子窗口 400（图 5）中列举出来。在步骤 832 中所有获取的而以前 又没有列举的出口都附加到列表中。接着控制交还给步骤 831 去测试是否所有可能的获取出口都已列举出来。

在经过测试 831 和步骤 832 之间可能的几个循环后，所有出口和所获取出口都得到标识和列举，则控制交给步骤 833，其中如果任何列举的出口用在连至一个或更多个目的子程序或域的一个或更多个连接中，它们都用一个星号 404（图 5）加以标注。在步骤 834（图

9)中，所有获取的出口都用一个符号“@”500(图5)加以标注，该符号500是紧挨着出口名称的，以及在步骤835(图9)中，从所选源子程序或域来的所有连接的列表在子窗口400(图5)中显示出来，其次在步骤836(图12)中，用于从目的子程序获取附加出口的出口，不论它们是否连接，都用符号“+”406(图5)标注，该符号406是紧挨着出口名称的。此后，在步骤837(图12)中，由子窗口400(图5)中增亮的出口所支持的连接类型列表在子窗口420(图5)中显示出来。接着控制交还步骤607(图10*A*)，去等待另一次按鼠标。

根据本发明的原理，本系统通过发送一条信息给程序存储器中的由所选图形单元所代表的对象，并向对象查询它的出口清单，来激励“源”对象程序逻辑去获取出口。当一个恰当设计的源对象这样被查询时，它就接下来向所连接(和增亮的)的目的对象查询可能用于命名新出口变量或对象的性质名称。接着源对象为用于它自己的程序逻辑而建立新出口，此时使用对查询作出响应的目的对象所提供的名字。由源对象功能这样所建立的新出口称为“获取的”出口，并且在检查员显示中紧挨着符号“@”列举出来，在最佳实施例中，被查询过名字的特定目的对象是用于产生出口名称的那种类型的早已存在连接的一部分(例如，在检查员窗口中其出口用“+”标注的连接)。

在举例的程序中，当检查员窗口第一次显示时，系统向源对象“*CONNECTIONS*—1”查询它的出口列表，以便将它们在检查员窗

口 30 中列举出来。当存储器内和“*CONNECTIONS*—1”相关联的对象如此被查询时，建立到 *FUNCTION*—1 的一个连接连同每个子程序的功能一起将 *CONNECTIONS*—1 加以激励，向 *FUNCTION*—1 查询名称列表，后者可能用作出口变量名称或将被 *CONNECTIONS*—1 所获取的对象。

在最佳操作模式中，有一些对象包含了功能，该功能是通过查询所连接目的对象来获取出口变量名称或出口对象时所必需的，系统可操纵包含该功能的对象，使系统用户使用系统键盘直接敲入所需出口变量名称。系统从连接检查员 30 切换到属性检查员，可供用户利用键盘手动地增加、删除、和改变所获取出口的名称。为了切换到属性检查员，用户可将鼠标放置于检查员模式按钮上方并按鼠标按钮。利用同样过程，用户可退回至连接检查员。

本发明可以很好地在差不多任何常规计算机系统中实现，实现本发明的示范性计算机系统 900 示于图 15。系统 900 包括 *a*) *CPU*901；*b*)主存储器 902；*c*)视频存储器 903；*d*)供用户输入用的键盘 904；*e*)用于根据本发明操纵图形图象的鼠标 905；以及 *f*)可能包括固定和可换媒体的海量存储器 906，其中使用了一种或更多种磁、光或光磁存储技术或任何其他可用的海量存储技术。这些部件通过常规双向系统总线 907 互相连接。总线 907 包括 32 条地址线，用于对存储 902 和 903 的任何部分进行寻址。系统总线 907 还包括一条 32 位数据总线，用于在下列部件间传送数据：*a*)*CPU*901，*b*)主存储

器 902，c)视频存储器 903，和 d) 海量存储器 906。在所示实施例中，*CPU*901 是一个 *Motorola* 68030 32 位微处理器，但任何其他合适的微处理器或微机也可作为替代方案加以使用。*Motorola* 公司(*Arizona* 的 *Phoenix* 市)印刷的 *MC*68030 用户手册提供了有关 68030 微处理器的详细信息，特别是有关指令系统、总线结构和控制线的详细信息。

系统 900 的主存储器 902 包括 8 兆字节常规动态随机存取存储器，当然更多或更少的存储器容量也是合适的。视频存储器 903 包括 256*K* 字节常规双口视频随机存取存储器。再者，视所需分辨率而定，可使用更多或更少这样的存储器。视频存储器 903 的一个端口连接了一个视频多路分配(*MUX*)和移位器电路 908，它又接到视频放大器 909。视频放大器 909 驱动阴极射线管(*CRT*)光栅显示器 910。常规的视频多路分配 和移位器电路 908 和视频放大器 909 将视频存储器 903 中存储的象素数据转换成适用于显示器 910 的光栅信号。显示器 910 的类型适合于显示图形图象，具有 1120 象素和 832 象素高的分辨率。

以上所述只是解释了本发明的原理。可以理解，熟悉技术的人能够设计不同的虽然在这里未曾明显地解释或示出过但却能体现本发明原理因而不越出本发明实质和范围的设备和装置。

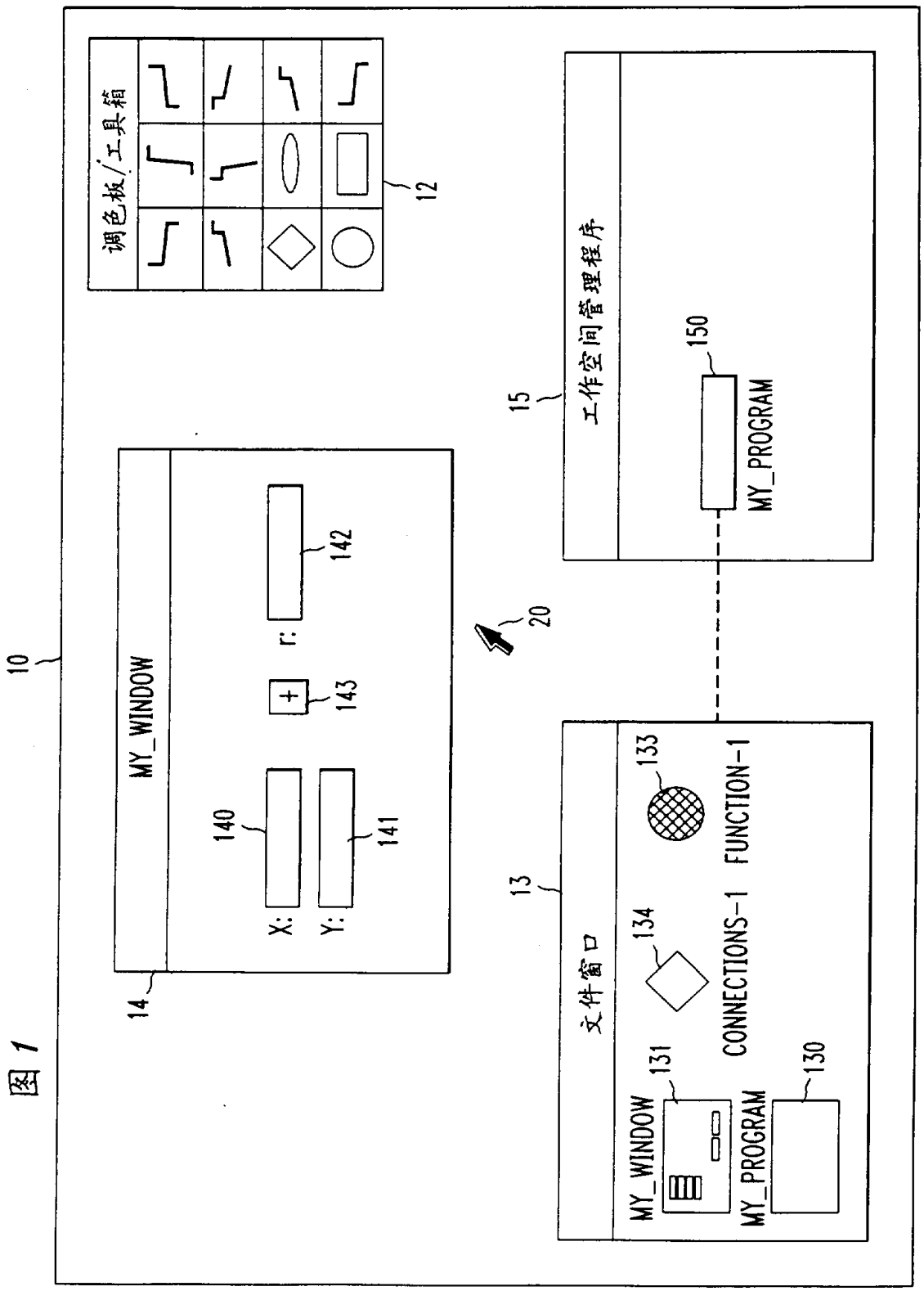

10
调色板/工具箱
12
MY_WINDOW
140
X:
141
Y:
142
r:
143
+
14
20
15
工作空间管理程序
150
MY_PROGRAM
13
文件窗口
MY_WINDOW
131
MY_PROGRAM
130
134
CONNECTIONS-1
133
FUNCTION-1

图 1

图 2

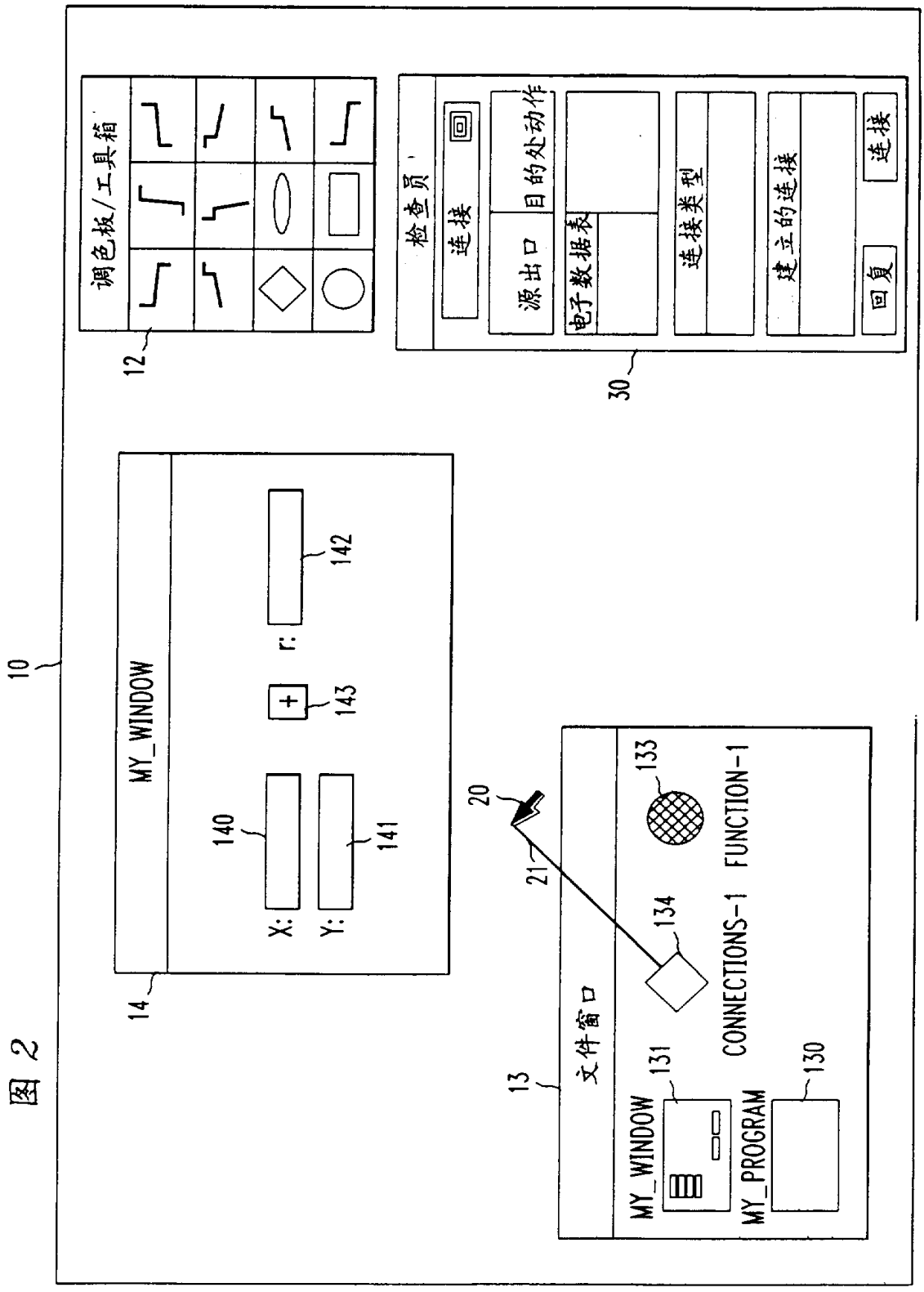

10
调色板/工具箱
12
检查员
连接
源出口
目的处动作
电子数据表
连接类型
建立的连接
回复
连接
30
MY_WINDOW
140
X:
141
Y:
143
r:
142
14
20
21
文件窗口
13
MY_WINDOW
131
MY_PROGRAM
130
134
133
CONNECTIONS-1
FUNCTION-1

图3

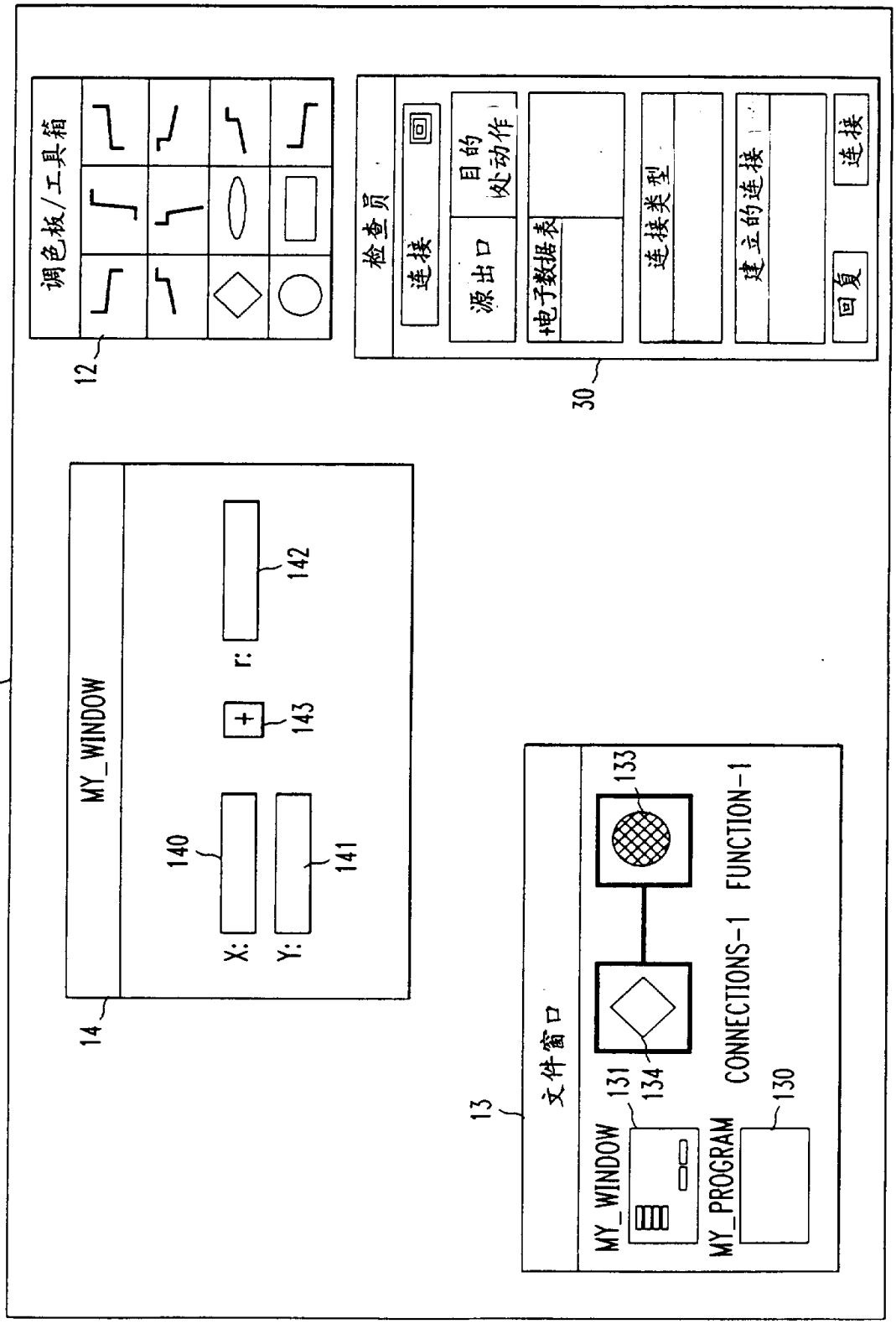

10
调色板/工具箱
12
检查员
连接
源出口
目的
处动作
电子数据表
连接类型
建立的连接
回复
连接
30
MY_WINDOW
140
142
r:
143
X:
141
Y:
14
文件窗口
13
MY_WINDOW
131
134
133
MY_PROGRAM
CONNECTIONS-1
FUNCTION-1
130

图 4

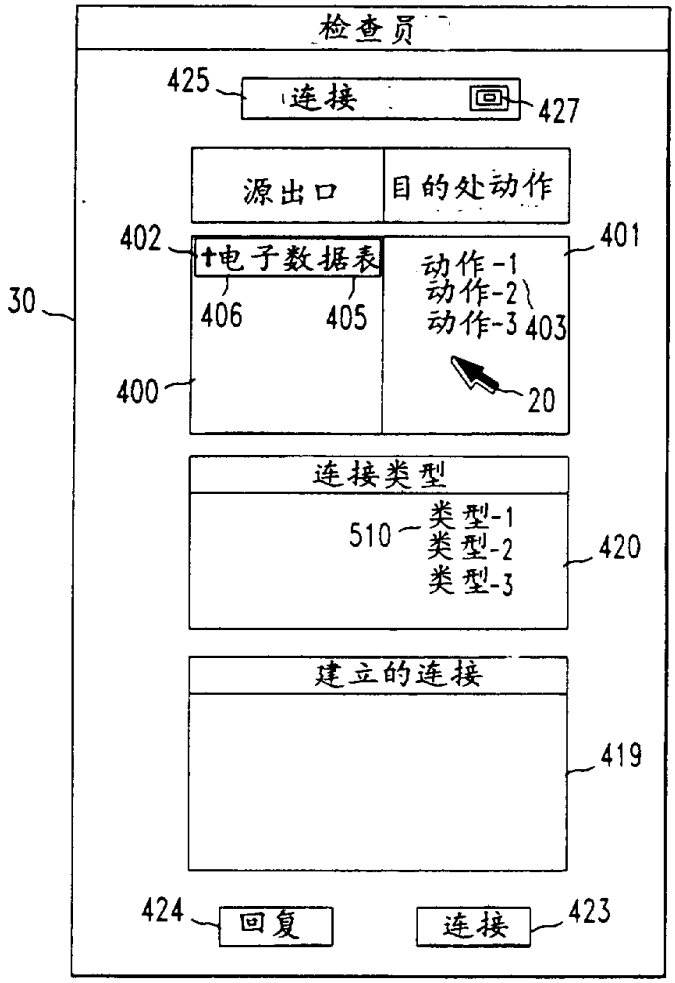

检查员
425
连接
427
源出口
目的处动作
402
电子数据表
406
405
401
动作-1
动作-2
动作-3
403
30
400
20
连接类型
510
类型-1
类型-2
类型-3
420
建立的连接
419
424
回复
连接
423

图 5

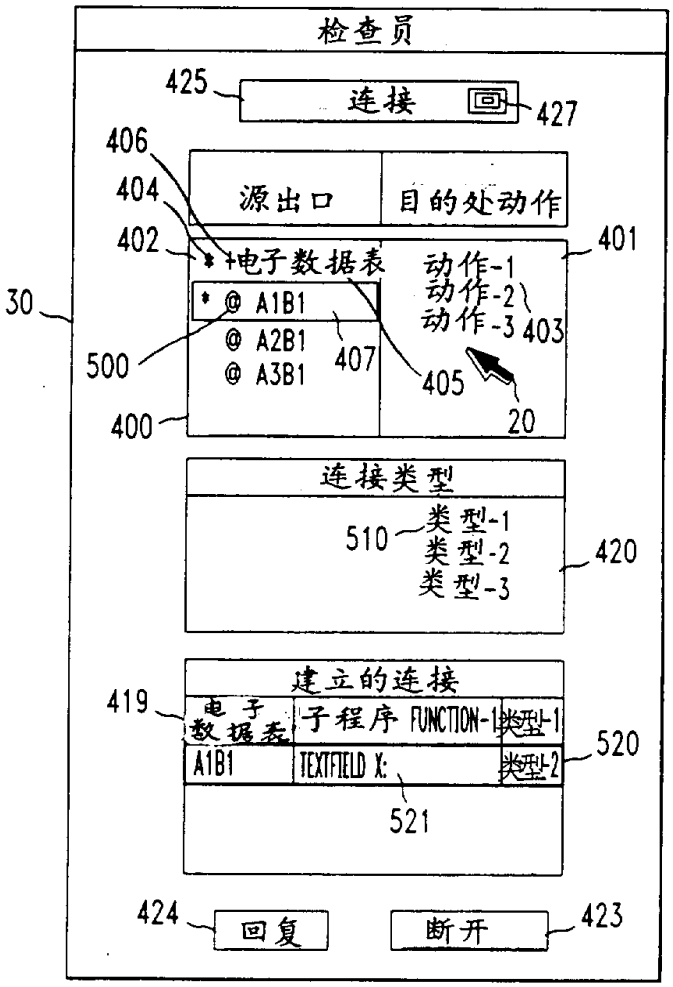

检查员
425
连接
427
406
404
源出口
目的处动作
402
401
+电子数据表
动作-1
动作-2
动作-3
403
30
* @ A1B1
@ A2B1
@ A3B1
500
407
405
20
400
连接类型
510
类型-1
类型-2
类型-3
420
建立的连接
419
电子数据表
子程序 FUNCTION-1
类型-1
A1B1
TEXTFIELD X:
类型-2
520
521
424
回复
断开
423

图 6

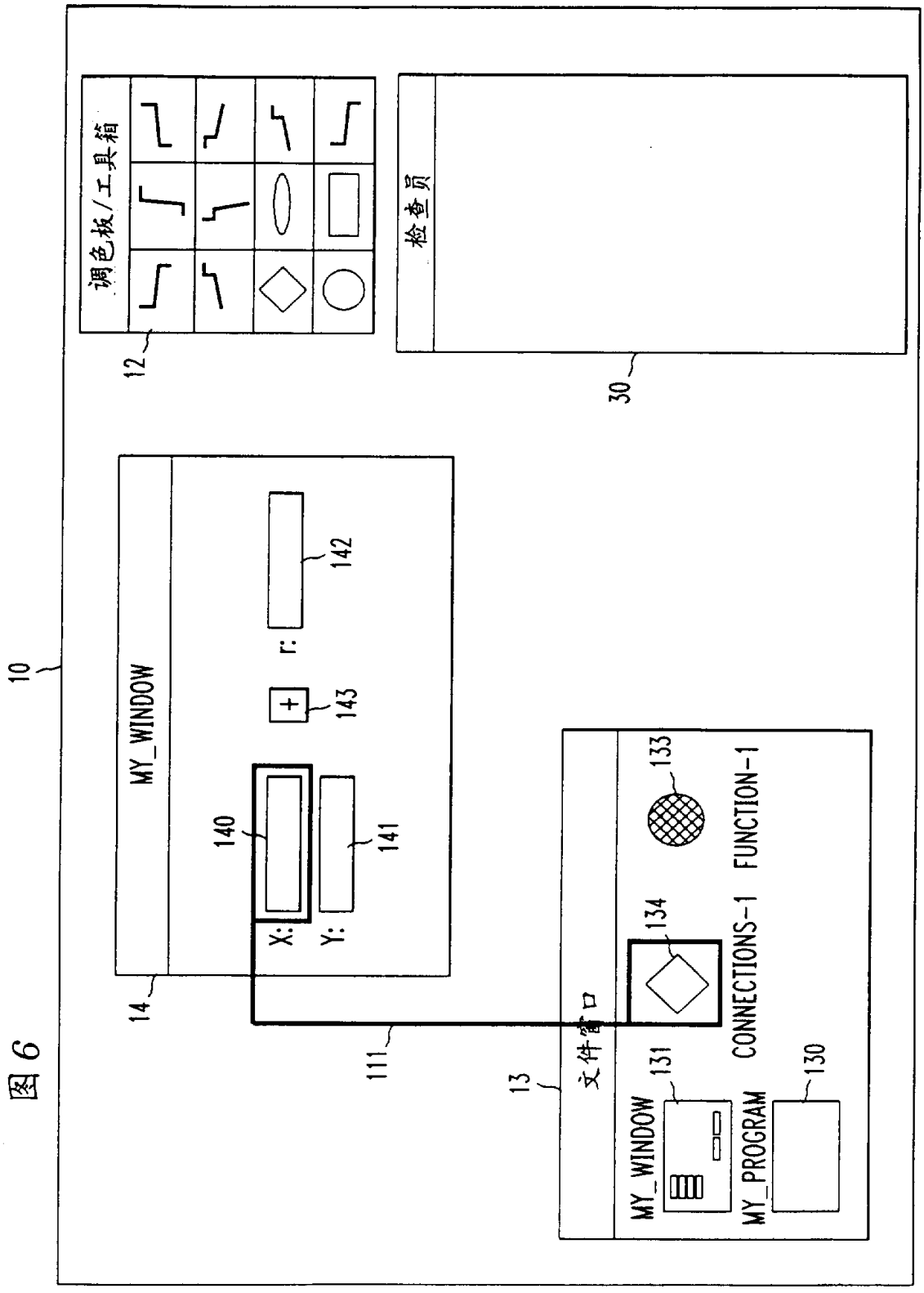

10
调色板/工具箱
12
检查员
30
MY_WINDOW
14
140
X:
141
Y:
142
r:
143
+
111
13
文件窗口
MY_WINDOW
131
MY_PROGRAM
130
134
CONNECTIONS-1
133
FUNCTION-1

图7

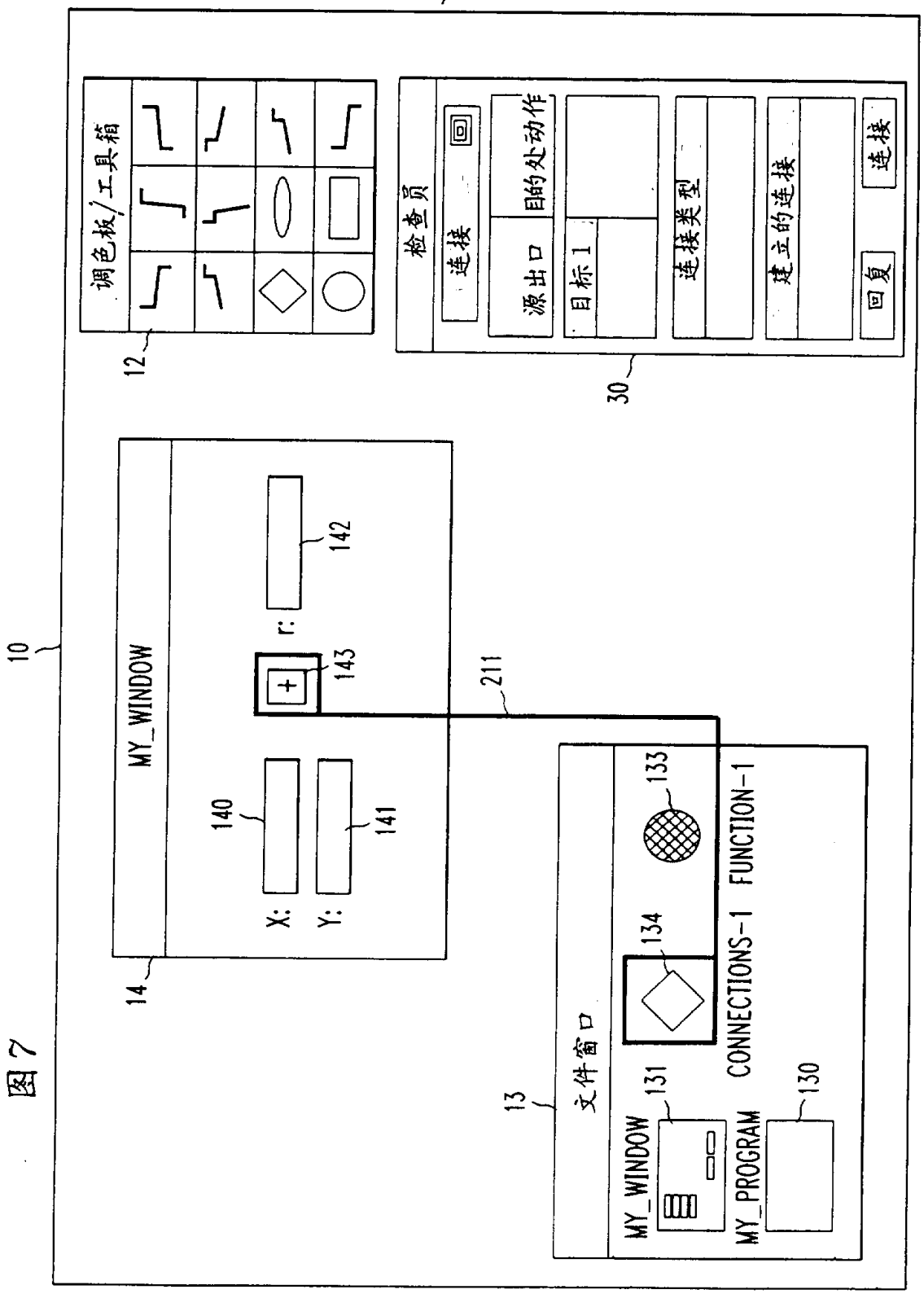

10
调色板/工具箱
12
检查员
连接
源出口
目标1
目的处动作
连接类型
建立的连接
回复
连接
30
MY_WINDOW
r:
142
143
140
141
X:
Y:
14
211
文件窗口
13
MY_WINDOW
131
133
134
CONNECTIONS-1
FUNCTION-1
MY_PROGRAM
130

图 8

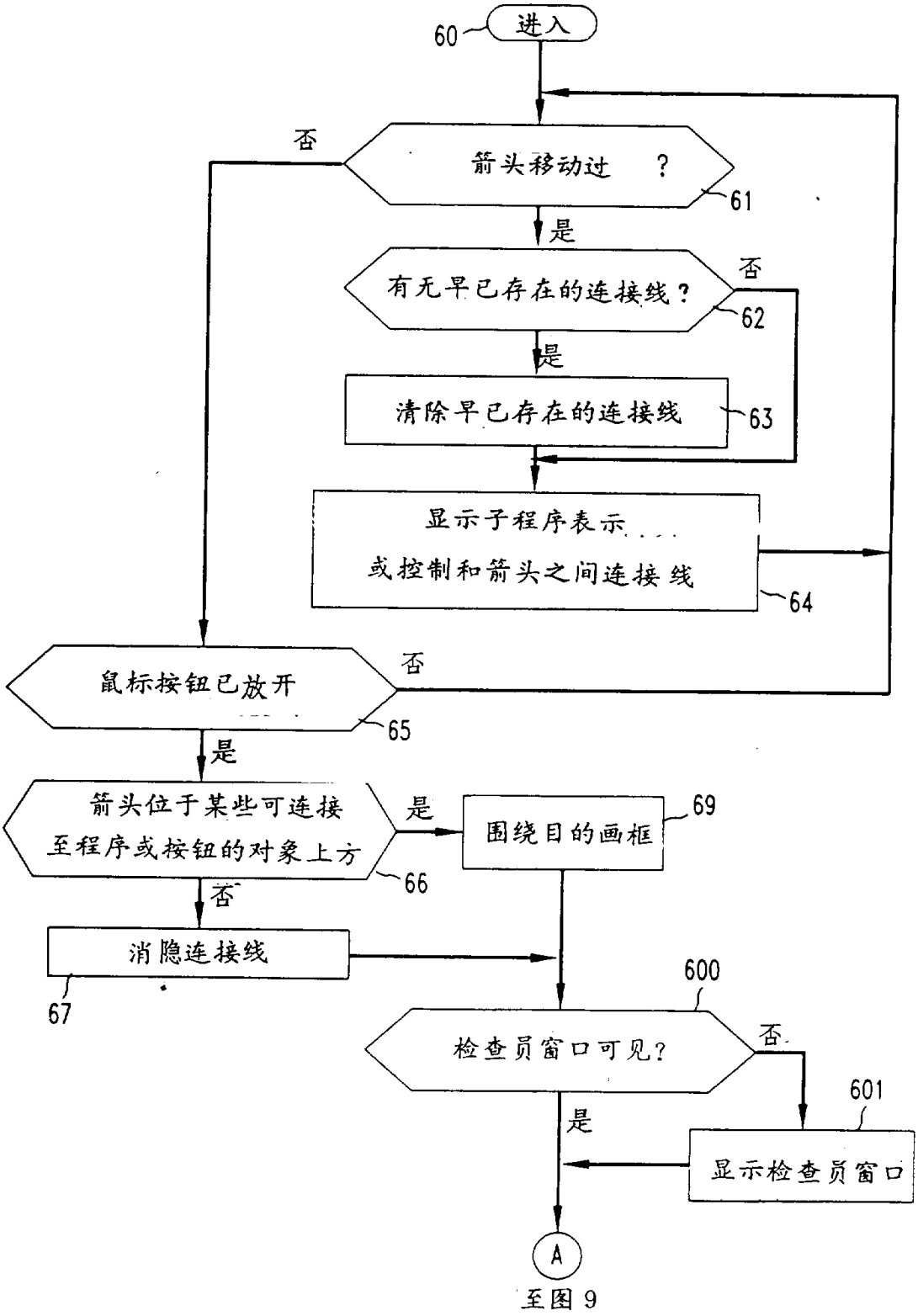
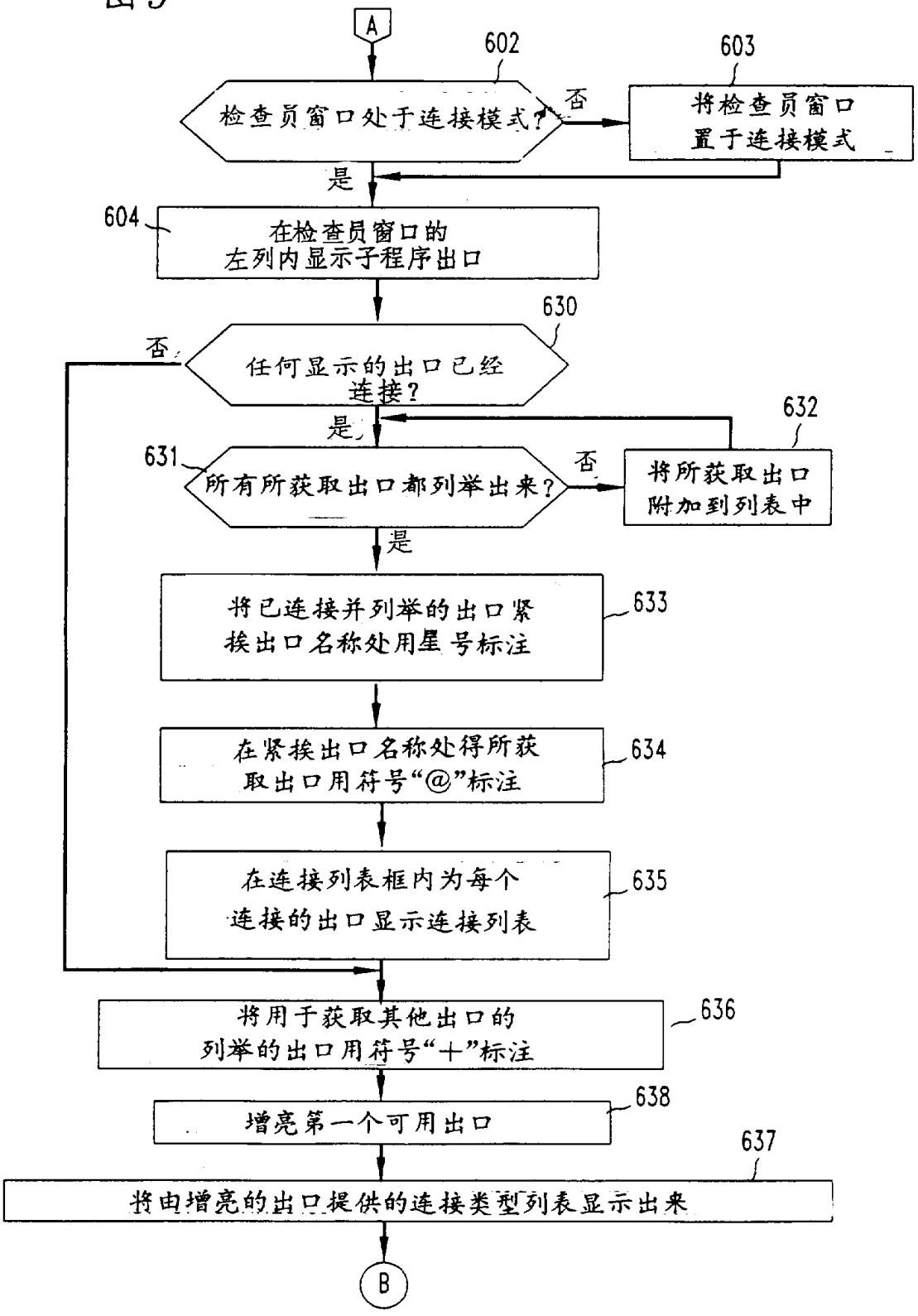

60
进入
否
箭头移动过 ?
61
是
有无早已存在的连接线?
否
62
是
清除早已存在的连接线
63
显示子程序表示
或控制和箭头之间连接线
64
鼠标按钮已放开
否
65
是
箭头位于某些可连接
至程序或按钮的对象上方
是
69
围绕目的画框
66
否
消隐连接线
67
600
检查员窗口可见?
否
601
是
显示检查员窗口
A
至图 9

图 9

自图 8


A
602
检查员窗口处于连接模式?
否
603
将检查员窗口置于连接模式
是
604
在检查员窗口的左列内显示子程序出口
630
否
任何显示的出口已经连接?
是
632
631
所有所获取出口都列举出来?
否
将所获取出口附加到列表中
是
将已连接并列举的出口紧挨出口名称处用星号标注
633
在紧挨出口名称处得所获取出口用符号“@”标注
634
在连接列表框内为每个连接的出口显示连接列表
635
将用于获取其他出口的列举的出口用符号“+”标注
636
增亮第一个可用出口
638
637
将由增亮的出口提供的连接类型列表显示出来
B

至图 10

图 10A

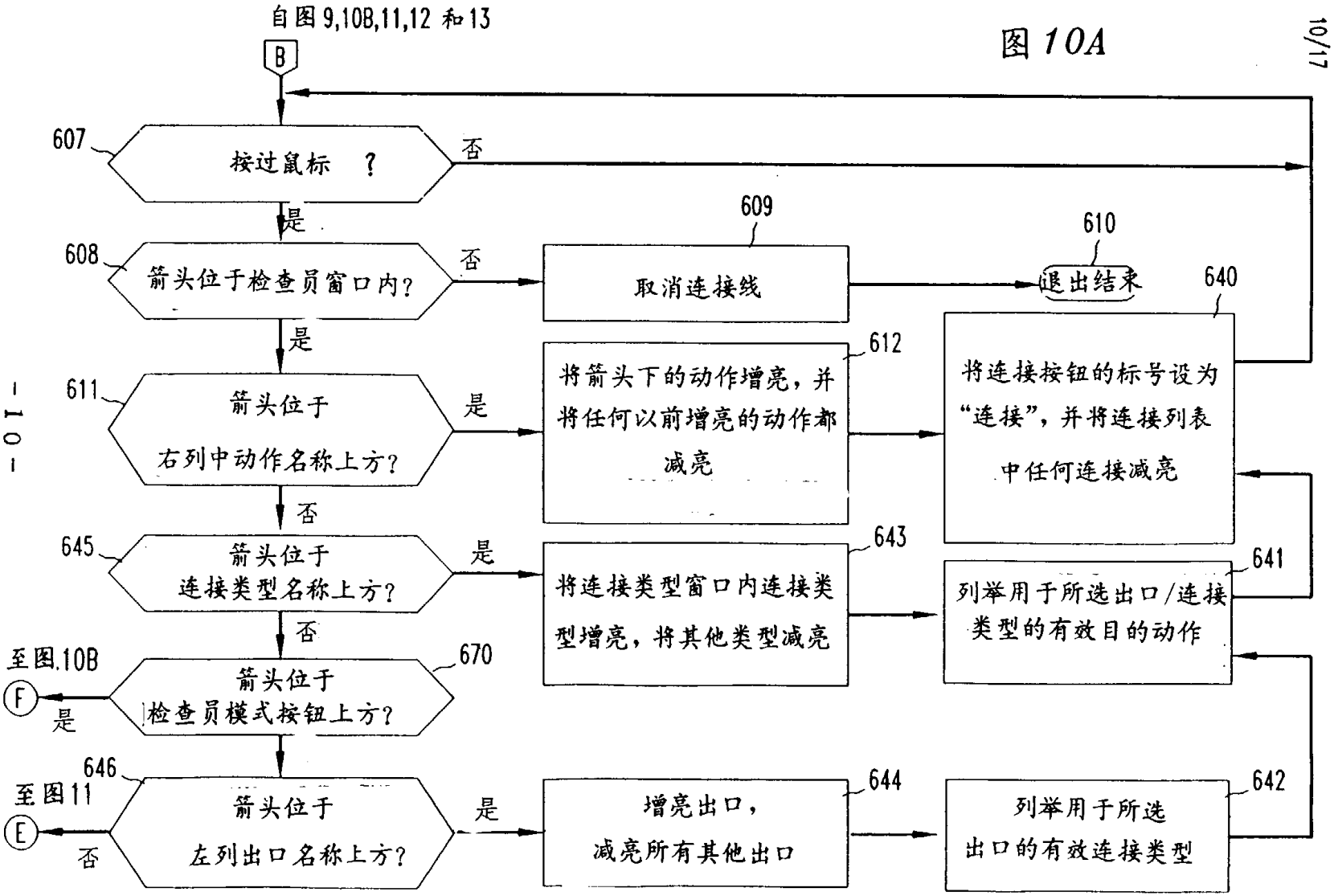

自图 9,10B,11,12 和13
B
607
按过鼠标 ?
否
是
608
箭头位于检查员窗口内?
否
609
取消连接线
610
退出结束
640
是
611
箭头位于
右列中动作名称上方?
是
612
将箭头下的动作增亮，并将任何以前增亮的动作都减亮
将连接按钮的标号设为“连接”，并将连接列表中任何连接减亮
否
645
箭头位于
连接类型名称上方?
是
643
将连接类型窗口内连接类型增亮，将其他类型减亮
641
列举用于所选出口/连接类型的有效目的动作
否
至图.10B
F
是
670
箭头位于
检查员模式按钮上方?
646
至图11
E
否
箭头位于
左列出口名称上方?
是
644
增亮出口，
减亮所有其他出口
642
列举用于所选
出口的有效连接类型

图10B

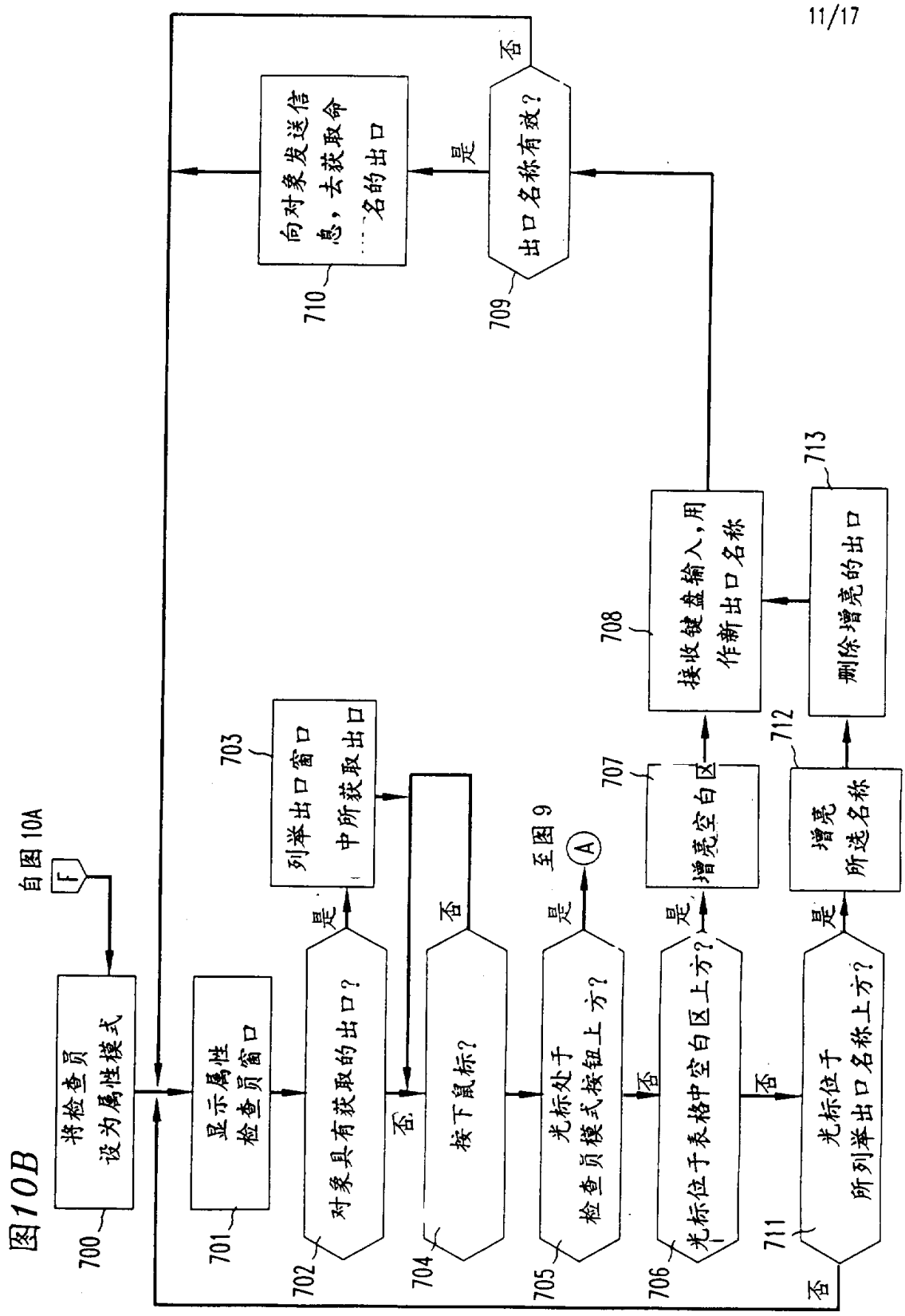

自图 10A
F
将检查员设为属性模式
700
显示属性检查员窗口
701
对象具有获取的出口?
702
是
否
列举出口窗口中所获取出口
703
按下鼠标?
704
光标处于检查员模式按钮上方?
705
是
否
至图 9
A
光标位于表格中空白区上方?
706
是
否
增亮空白区
707
接收键盘输入,用作新出口名称
708
出口名称有效?
709
是
否
向对象发送信息,去获取命名的出口
710
光标位于所列举出口名称上方?
711
是
否
增亮所选名称
712
删除增亮的出口
713

图 11

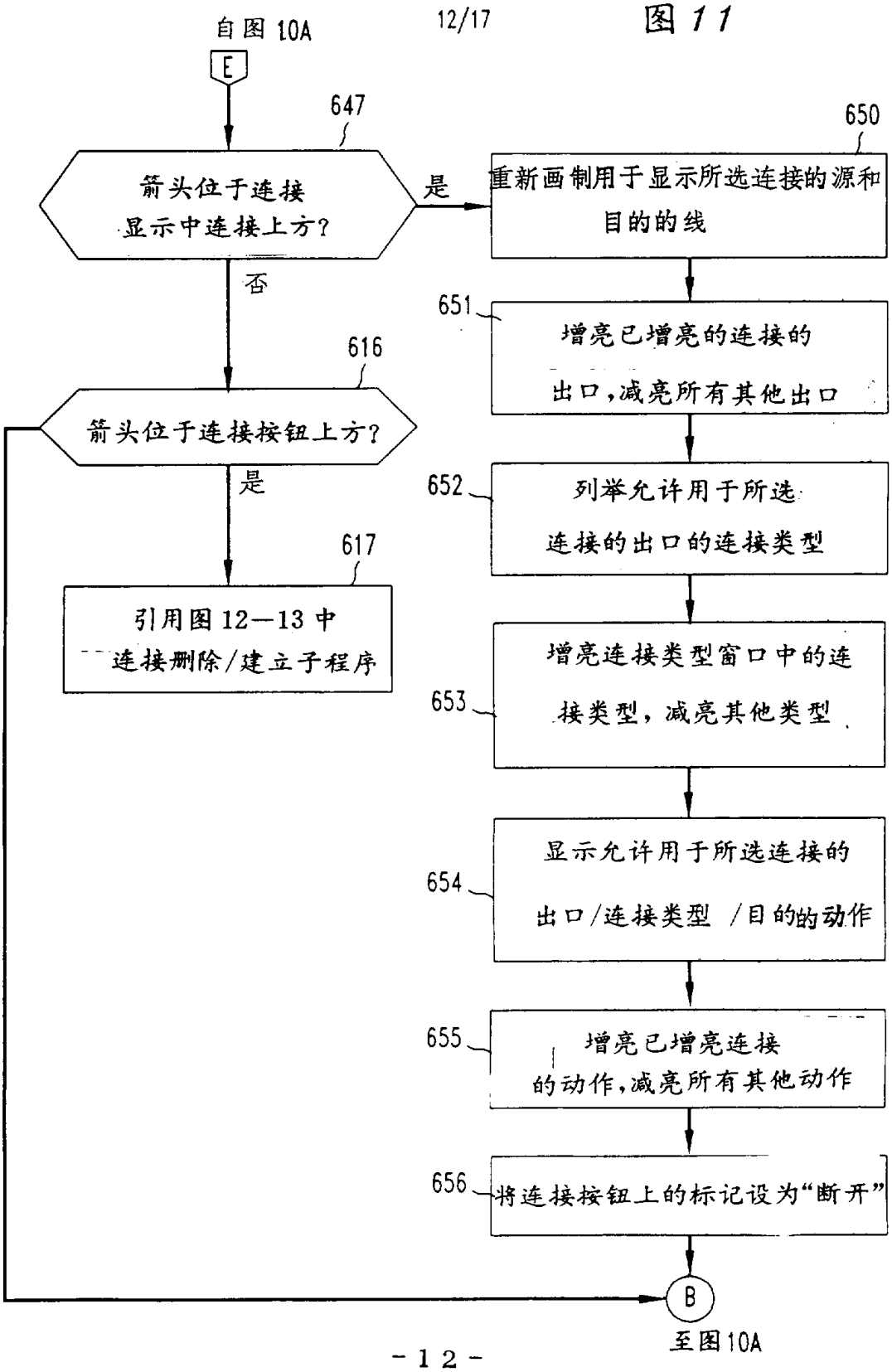

自图 10A
E
647
箭头位于连接显示中连接上方?
是
650
重新画制用于显示所选连接的源和目的的线
否
651
增亮已增亮的连接的出口,减亮所有其他出口
616
箭头位于连接按钮上方?
是
652
列举允许用于所选连接的出口的连接类型
617
引用图 12—13 中连接删除/建立子程序
653
增亮连接类型窗口中的连接类型,减亮其他类型
654
显示允许用于所选连接的出口/连接类型 /目的的动作
655
增亮已增亮连接的动作,减亮所有其他动作
656
将连接按钮上的标记设为"断开"
B
至图10A

图12

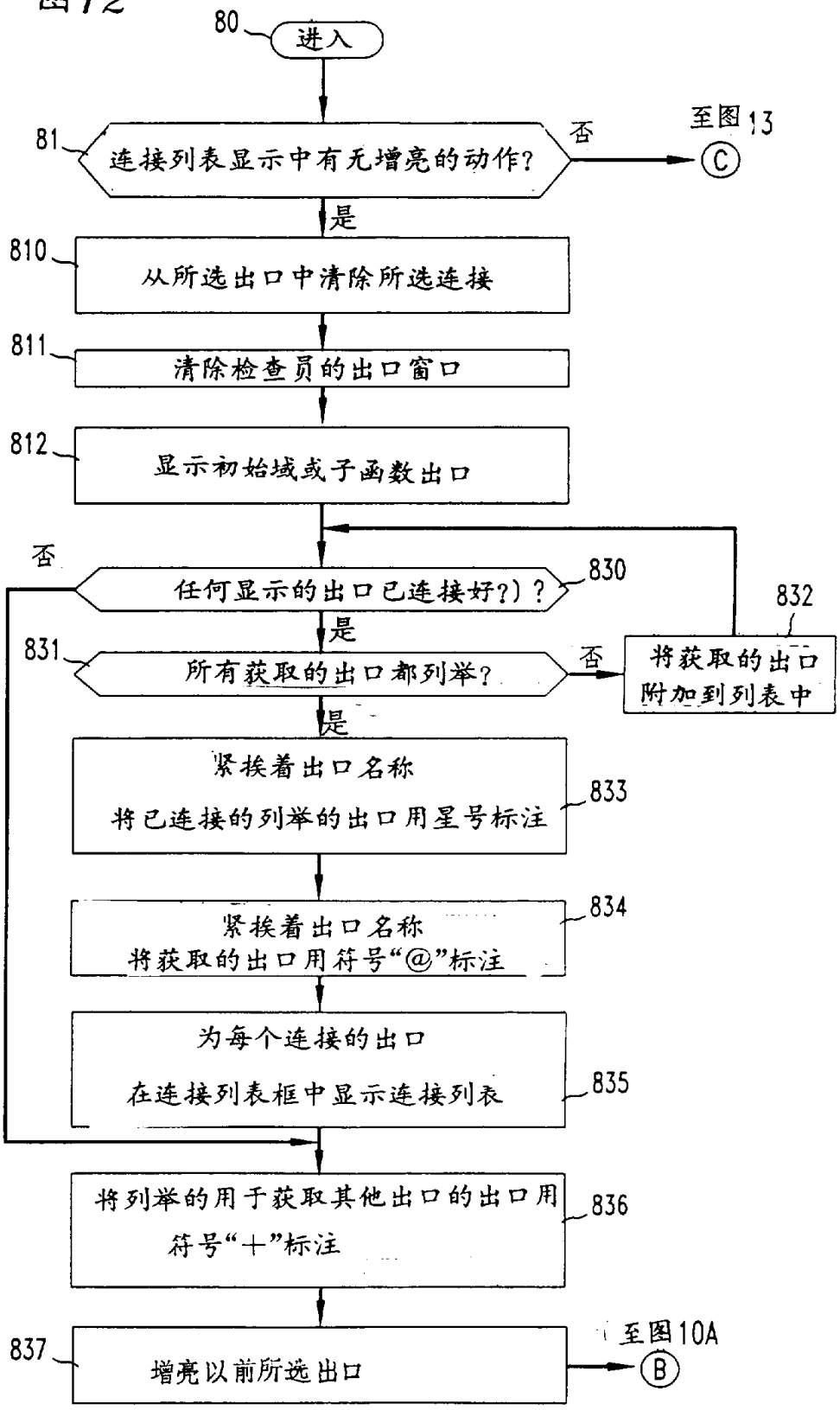

80
进入
81
连接列表显示中有无增亮的动作?
否
至图13
C
是
810
从所选出口中清除所选连接
811
清除检查员的出口窗口
812
显示初始域或子函数出口
否
任何显示的出口已连接好?)?
830
是
831
所有获取的出口都列举?
否
832
将获取的出口附加到列表中
是
紧挨着出口名称
将已连接的列举的出口用星号标注
833
紧挨着出口名称
将获取的出口用符号"@"标注
834
为每个连接的出口
在连接列表框中显示连接列表
835
将列举的用于获取其他出口的出口用符号"+"标注
836
837
增亮以前所选出口
至图10A
B

图13

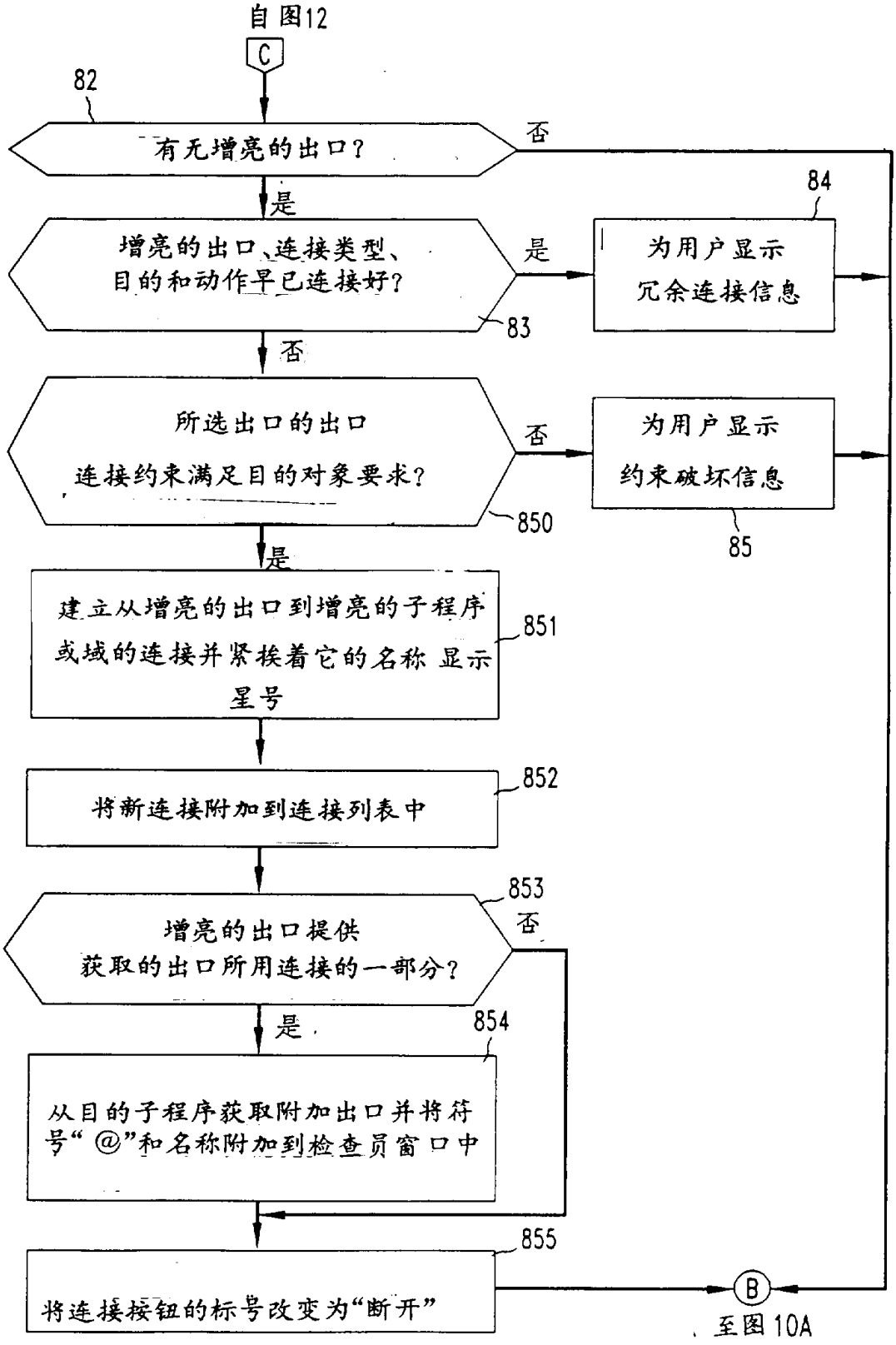

自图12
C
82
有无增亮的出口？
否
是
84
增亮的出口、连接类型、目的和动作早已连接好？
是
为用户显示冗余连接信息
83
否
所选出口的出口连接约束满足目的对象要求？
否
为用户显示约束破坏信息
850
85
是
建立从增亮的出口到增亮的子程序或域的连接并紧接着它的名称显示星号
851
将新连接附加到连接列表中
852
853
增亮的出口提供获取的出口所用连接的一部分？
否
是
854
从目的子程序获取附加出口并将符号“@”和名称附加到检查员窗口中
855
将连接按钮的标号改变为“断开”
B
至图10A

图 14

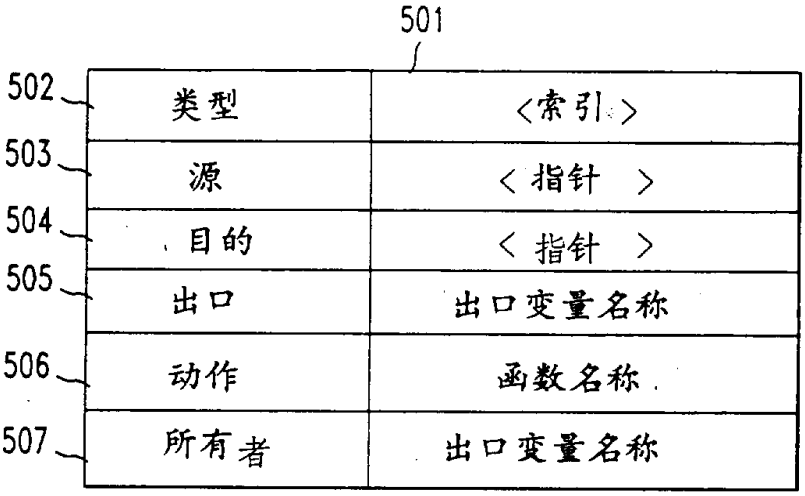

501
502
503
504
505
506
507
类型
<索引>
源
< 指针 >
目的
< 指针 >
出口
出口变量名称
动作
函数名称
所有者
出口变量名称

图15

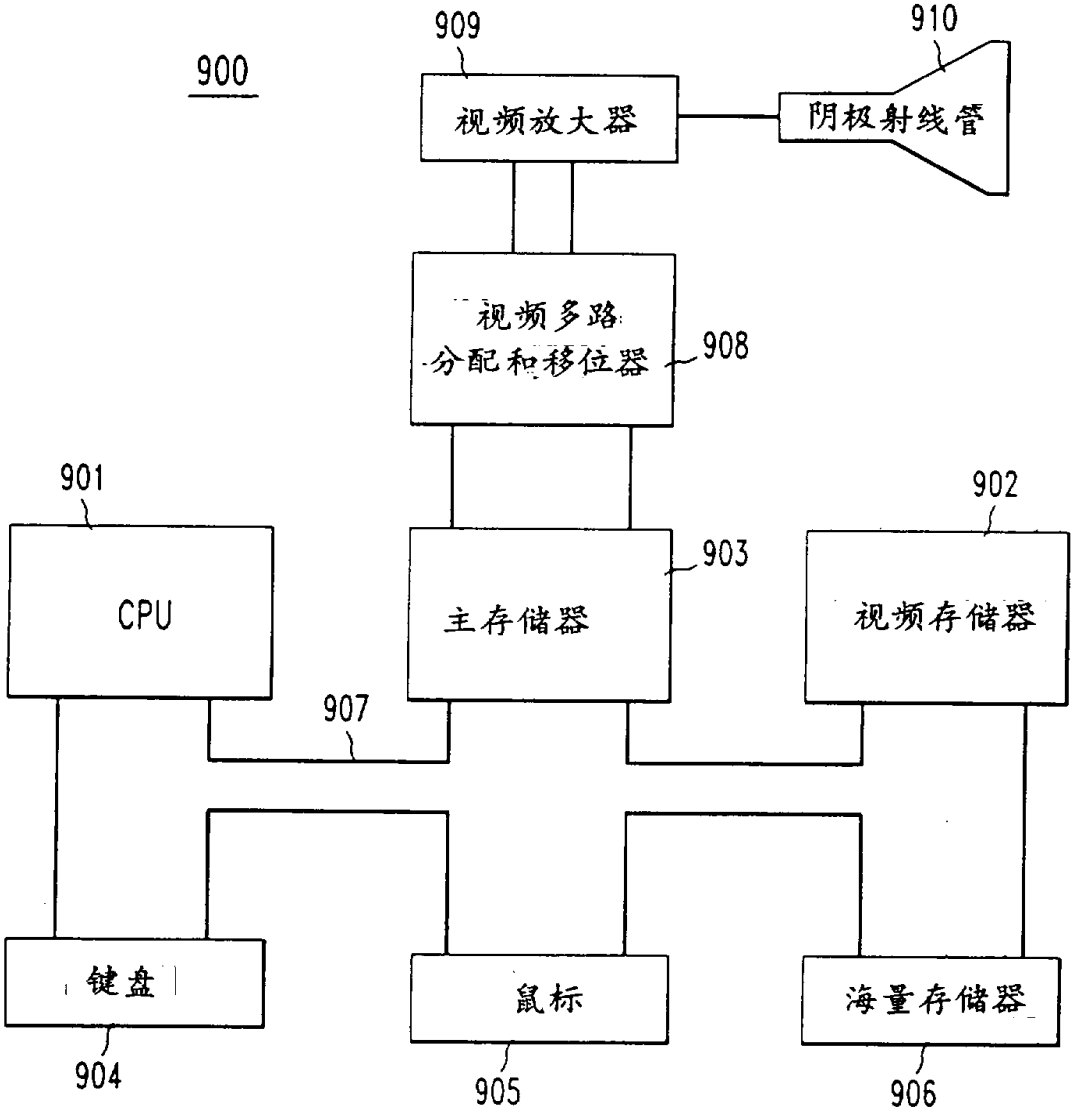

900
909
910
视频放大器
阴极射线管
视频多路
分配和移位器
908
901
902
CPU
903
主存储器
视频存储器
907
键盘
鼠标
海量存储器
904
905
906

图16

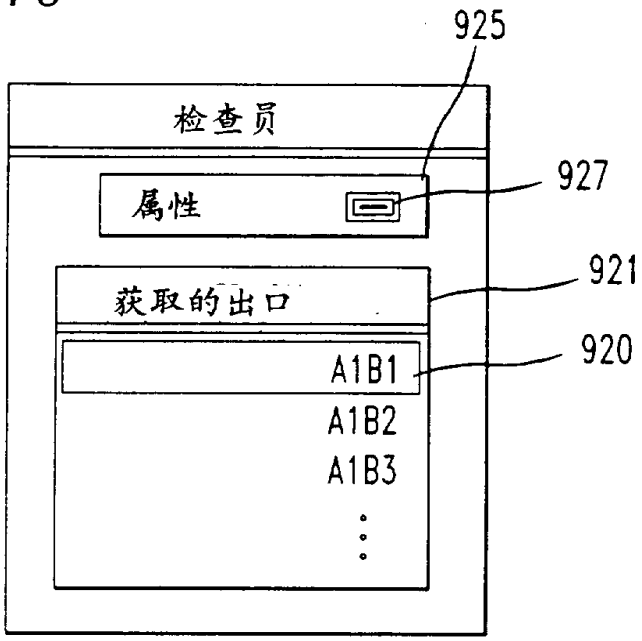
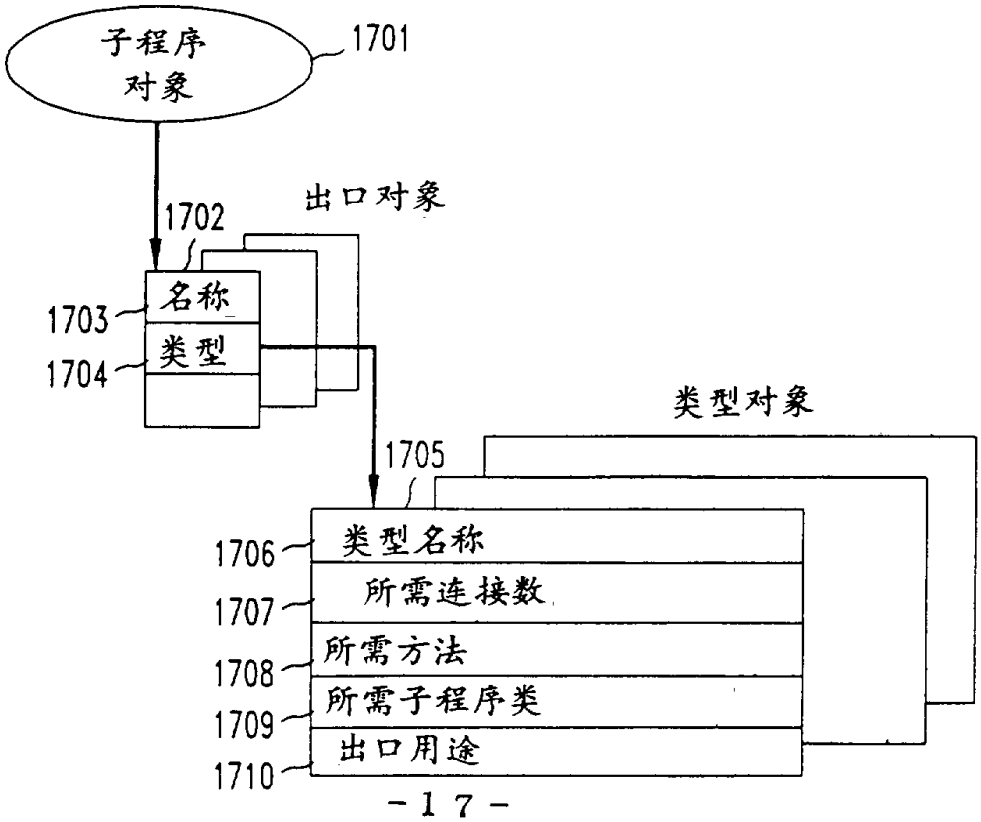

925
检查员
属性
927
获取的出口
921
A1B1
920
A1B2
A1B3

图17


子程序
对象
1701
出口对象
1702
名称
1703
类型
1704
类型对象
1705
类型名称
1706
所需连接数
1707
所需方法
1708
所需子程序类
1709
出口用途
1710