

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200580014272.6

[43] 公开日 2007 年 4 月 18 日

[51] Int. Cl.

G05B 19/418 (2006.01)

G05B 23/02 (2006.01)

G06F 9/44 (2006.01)

[11] 公开号 CN 1950771A

[22] 申请日 2005.5.4

[21] 申请号 200580014272.6

[30] 优先权

[32] 2004.5.4 [33] US [31] 60/567,980

[86] 国际申请 PCT/US2005/015393 2005.5.4

[87] 国际公布 WO2005/107409 英 2005.11.17

[85] 进入国家阶段日期 2006.11.3

[71] 申请人 费舍 - 柔斯芒特系统股份有限公司  
地址 美国德克萨斯州

[72] 发明人 马克·J·尼克松  
斯蒂芬·G·汉莫克  
布鲁斯·坎普尼 肯·J·贝欧格特  
斯蒂芬·吉尔伯特  
迈克尔·J·卢卡斯

[74] 专利代理机构 北京德琦知识产权代理有限公司

代理人 周艳玲 宋志强

权利要求书 3 页 说明书 25 页 附图 10 页

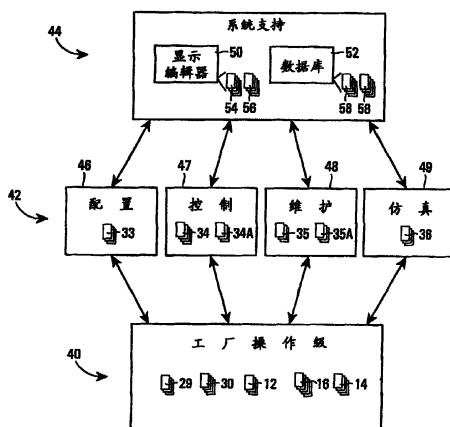
[54] 发明名称

用于过程控制的集成图形用户界面

[57] 摘要

本发明公开一种用于工厂操作员、工厂维护人员和管理人员的公共过程控制图形用户界面，其提供对该过程和工厂的实时界面。该公共界面在设计上是模块化的，并且能够支持用于每种用户类型的各种专属性。操作员控制台专用于工厂的各个部分，并且包括例如维护、配置、仿真和监督信息的附加功能。统一的公共图形界面取代了用单范例模拟控制器、仪表和数字指示器填充的控制室显示。公共界面提供了之前由面板电动机启动/停止按钮以及状态指示、图表记录器、信号器面板和子系统界面提供的功能。操作员利用控制台管理警报、通过输入新的设定点或其它参数来调整过程、“放大”该过程的特定部分以查看细节，并且使用其它专门的应用程序来处理其批处理、先进控制或商务应用。该界面能运行于专用或非专用模式下，能作

为富客户端或作为使用网络服务的浏览器型界面的一部分运行，并且能在工作站、膝上型电脑、平板PC、手持设备和智能电话中运行。



1、一种用于过程控制环境的集成图形用户界面，该界面包括：

用于加工厂的操作、维护、配置、仿真和管理功能的界面图形和各自的功能，

用于操作、维护、配置和仿真功能的实时界面连接功能，

该界面提供操作、维护、配置和仿真功能中的不只一种功能的实时显示。

2、根据权利要求 1 所述的界面，进一步包括原始数据向操作动画和操作趋势的转换以及所述操作动画和操作趋势的图形显示。

3、根据权利要求 2 所述的界面，进一步包括维护需求、维护报告和监督报告的图形表示。

4、根据权利要求 1 所述的界面，进一步包括下列功能中的一个或更多个：面板电动机启动/停止按钮、状态指示、图表记录器、信号器面板、子系统界面及其组合。

5、根据权利要求 1 所述的界面，进一步包括下列功能中的一个或更多个：警报管理、通过输入过程参数调整过程参数、放大过程的一部分视图以获得放大的细节视图、使用与过程有关的专门应用程序。

6、根据权利要求 1 所述的界面，其中，该界面可运行于专用和非专用模式下。

7、根据权利要求 1 所述的界面，其中，该界面可作为富客户端或作为采用网络服务的浏览器型界面的一部分运行。

8、根据权利要求 1 所述的界面，其中，该界面可在工作站、膝上型电脑、PDA 和智能电话中的一个或更多个上运行。

9、根据权利要求 1 所述的界面，其中，该界面支持用于固定显示设置的专用模式。

10、根据权利要求 1 所述的界面，其中，该界面支持由配置人员使用的非专用模式。

11、根据权利要求 1 所述的界面，其中，该界面支持多种用户界面设备。

12、根据权利要求 1 所述的界面，其中，该界面支持下列特征中的一个或更多个：集成声音和视频；实时数据服务；外部数据服务；XML 文件；访问其它服务界面；复合结构过程图形；基于类的控制层次；集成控制、告警以及异常状况的管理和预防；集成批处理操作员界面；集成先进控制操作员界面；路由管理；效率计算；最优化；质量和能量平衡；集成第三方应用程序；多种数据采集系统（DCS）及其组合。

13、根据权利要求 1 所述的界面，其中，该界面接收将被加载入运行期工作空间的显示和其它应用程序。

14、一种集成图形界面，其在控制系统的多个工作站中提供用于操作、维护、配置和仿真功能的无缝图形显示，该界面对包括：

追踪功能，用于追踪该系统中的哪些工作站中使用了哪些显示，并且该界面提供操作、维护、配置和仿真功能中不只一种功能的实时显示。

15、根据权利要求 14 所述的图形界面，进一步包括用于链接来自外部源的数据的链接功能。

16、根据权利要求 14 所述的图形界面，进一步包括用于使用来自外部源的数据的装置。

17、根据权利要求 14 所述的图形界面，进一步包括用于定义工厂标准的装置、用于锁定所述工厂标准的装置、以及用于在所述系统的多个工作站中分配所述工厂标准的装置。

18、根据权利要求 14 所述的图形界面，进一步包括可缩放的图形。

19、一种机器可读介质，其上存储有指令，一旦执行所述指令，将使机器：在控制系统的多个工作站中，提供用于操作、维护、配置和仿真功能的无缝图形显示，并且提供操作、维护、配置和仿真功能中的不只一种功能的实时显示，提供追踪功能，用于追踪该系统中的哪些工作站中使用了哪些显示，并且为操作控制和维护人员提供实时界面功能。

20、根据权利要求 19 所述的机器可读介质，其上存储有指令，一旦执行所

述指令,将使该机器提供完全可缩放的且具有旋转图形和使部分图形变透明的能力的图形。

## 用于过程控制的集成图形用户界面

### 相关申请

本申请是 2004 年 5 月 4 日递交的顺序号为 60/567,980、题为“用于表示、监测和与过程控制系统交互的图形用户界面”的美国临时专利申请的正式递交申请，并出于优先权的目的要求其权益，在此本申请明确地将其全部内容合并作为参考。本申请还与 2003 年 7 月 21 日递交的、题为“加工厂中图形显示元素、过程模块和控制模块的集成”、并在 2004 年 8 月 5 日以美国出版号 2004/0153804 出版的、顺序号为 10/625,481 的美国专利申请有关，该申请为 2002 年 10 月 22 日递交的、在 2004 年 4 月 22 日以美国出版号 2004/0075689 出版的、题为“加工厂中的智能过程模块和对象”、顺序号为 10/278,469 的美国专利申请的部分继续申请，在此明确地将这两个公开的全部内容合并作为参考。本申请还与 2003 年 2 月 18 日递交的、题为“加工厂配置系统中的模块类对象”、并在 2004 年 10 月 7 日以美国出版号 2004/0199925 出版的、顺序号为 10/368,151 的美国专利申请有关，在此明确地将其全部内容合并作为参考。本申请还和与本申请在同一天递交的正在作为国际（PCT）申请的下列专利申请有关，在此明确地将这些专利申请全部内容合并作为参考：“过程环境中的相关图形显示”（代理备案 No.06005/41111）；“用于过程控制系统的用户可配置警报和警报趋势”（代理备案 No.06005/41112）；“加工厂中过程模块和专家系统的集成”（代理备案 No.06005/41113）；“集成环境中具有定制的过程图形显示层的加工厂用户界面系统”（代理备案 No.06005/41114）；“过程环境中的脚本图形”（代理备案 No.06005/41115）；“集成到过程配置和控制环境中的图形”（代理备案 No.06005/41116）；“过程中具有多种视像的图形元素”（代理备案 No.06005/41117）；“加工厂中用于配置图形显示元素和过程模块的系统”（代理备案 No.06005/41118）；“加

工厂用户界面中基于 Markup 语言的动态过程图形”（代理备案 No.06005/41127）；“用于修改过程控制数据的方法和装置”（代理备案 No.06005/591622 和 20040/59-11622）；“用于存取过程控制数据的方法和装置”（代理备案 No.06005/591623 和 20040/59-11623）；“用于过程控制系统的集成图形运行时间界面”（代理备案 No.06005/591628 和 20040/59-11628）；“面向服务的过程控制系统架构”（代理备案 No.06005/591629 和 20040/59-11629）。

## 技术领域

本发明公开一种用于操作员、维护人员、管理人员和其他人员的公共过程控制用户界面。该用户界面能够支持用于操作员、维护工人、管理人员、仿真工程师、配置工程师和其它人员的专门图形，即支持所有用户类型的专门图形，而不是用于每种用户类型的单独界面。该公共界面使用原始数据到有用的动画、趋势和图案的快速且精确的转换，其中这些动画、趋势和图案能有助于确定改进过程控制所需要的动作。该公共界面可由用户操作，但是包括例如警报通知等不可遮盖的部分。

## 背景技术

过程控制系统在制造产品或控制过程（例如化工制造、电厂控制等）的工厂和/或加工厂中有着非常广泛的应用。过程控制系统还被应用在自然资源的开采中，例如油气钻井和加工处理等。事实上，任何制造过程、资源开采过程，包括农业，都能够通过应用一个或多个过程控制系统实现自动化。

例如用于化学、石油或其它过程中的“分布式”过程控制系统，通常包括一个或更多个过程控制器，这些过程控制器经由模拟、数字或模拟/数字混合总线通信连接到一个或更多个现场设备上。现场设备可以是例如阀、阀门定位器、开关、变送器（例如，温度、压力、电平以及流速传感器），它们位于过程环境中，并且执行各种过程功能，例如打开或关闭阀、测量过程参

---

数等。智能现场设备，例如符合诸如 FOUNDATION™ Fieldbus 协议之类的公知 Fieldbus 协议的现场设备，还可以执行控制计算、警报功能以及通常在控制器中实现的其它控制功能。

过程控制器接收表示由现场设备产生的过程测量值的信号和/或属于现场设备的其它信息，并执行控制器应用程序，举例来说，该控制器应用程序运行不同的控制模块，这些控制模块根据接收到的信息作出过程控制决策、产生控制信号，并与正在诸如 HART 和 Fieldbus 现场设备之类的现场设备中执行的控制模块或块（block）协调工作。控制器中的控制模块经由通信线路发送控制信号到现场设备，从而控制过程的操作。

通常，通过数据总线使得来自现场设备和控制器的信息可用于一个或更多个其它硬件设备，例如操作员工作站、个人计算机、数据历史记录器、报告发生器、集中式数据库等，这些硬件设备通常被放置在远离恶劣的加工厂环境的控制室中或其它地方。这些硬件设备运行各种应用程序，例如使操作员能够执行与过程有关的功能，例如改变过程控制例行程序的设置、修改控制器或现场设备中的控制模块的操作、查看过程的当前状态、查看由现场设备和控制器产生的警报、仿真过程的操作以培训人员或测试过程控制软件、维护和更新配置数据库等。

例如，由爱默生过程管理公司出售的 DeltaV™ 控制系统，包括存储在加工厂的不同地方的不同设备中并由这些设备执行的多个应用程序。保存在一个或更多个操作员工作站中的配置应用程序使用户能够创建或改变过程控制模块，并通过数据总线将这些过程控制模块下载到专用的分布式控制器中。

通常，这些过程控制模块由通信互连的功能块组成，这些功能块是面向对象编程协议中的对象，它们基于其输入来执行控制方案内的功能，并把输出提供给控制方案中的其它功能块。配置应用程序还可以允许设计者创建或改变操作员界面，该界面由查看应用程序使用，以向操作员显示数据并使操作员能够改变过程控制例行程序中的设置，例如设定点。每个专用控制器以

---

及某些情况下的现场设备，都存储并执行控制器应用程序，该控制器应用程序运行被分配和下载到其中的控制模块以实现实际的过程控制功能。

可以运行在一个或更多个操作员工作站中的查看应用程序通过数据总线从控制器应用程序接收数据，并把此数据显示给过程控制系统的设计者、操作员或使用用户界面的用户，并且可以提供多个不同视图中的任意一个，例如操作员视图、工程师视图和技术员视图等。数据历史记录器应用程序通常被存储在数据历史记录设备中并由其执行，其中数据历史记录设备采集和存储经由数据总线提供的全部数据或其中的一些数据，而配置数据库应用程序可以运行在另一台连接到数据总线上以存储当前过程控制例行程序的配置及其相关数据的计算机中。可替代地，配置数据库可以与配置应用程序设置在同一工作站中。

随着用在过程控制环境中的控制和支持应用程序的数量和类型逐渐增加，已经提供了不同的图形显示应用程序来使用户能够有效地配置和使用这些应用程序。例如，已经使用图形显示应用程序来支持控制配置应用程序，以使配置工程师能够图形化地创建将被下载到加工厂的控制设备上的控制程序。

另外，已经使用图形显示应用程序，以便使控制操作员能够查看加工厂或加工厂的一些区域的当前功能，使维护人员能够查看加工厂中的硬件设备的状态，使加工厂的仿真能够进行等。然而，过去，这些图形显示应用程序是作为与它们有关的特定应用程序的一部分或用来支持这些应用程序而创建的，因而，通常局限于对创建了它们的特定过程功能有用。例如，如果不是不可能，那么也很难在维护、配置或仿真功能中使用一个当前可用的、被创建用来支持控制操作员的图形程序。相反，如果不是不可能，也很难使用一个当前可用的、为维护技术员设计的图形程序来支持控制操作员、配置工程师，或在仿真功能中使用该图形程序。总之，当前可用的图形程序是专门而非通用的。

举一个特定的示例，某些过程配置应用程序目前包括模板对象库，例如

---

功能块模板对象，以及某些情况下用来为加工厂创建控制策略的控制模块模板对象。这些模板对象具有默认的属性、设置和与其相关的方法，并且使用图形配置应用程序的工程师，可以选择这些模板对象，并主要把所选择的模板对象的拷贝放进配置屏幕中来开发控制模块。

在选择和把模板对象放进配置屏幕的过程中，工程师将这些对象的输入和输出相连并改变它们的参数、名称、标签以及其它属性，以创建加工厂中具有特定用途的特定控制模块。在创建一个或更多个这种控制模块后，工程师随后可以实例化该控制模块，并把它下载到一个或更多个合适的控制器以及现场设备中，用于在加工厂运行期间加以执行。

此后，工程师可以使用不同的图形显示创建应用程序，通过在该显示创建应用程序中选择并构造显示对象来创建一个或更多个用于加工厂内的操作员、维护人员等的显示。这些显示通常以系统范围为基础在一个或更多个工作站中实施，并向与加工厂中控制系统或设备的运行状态有关的操作员或维护人员提供预先配置的显示。这些显示通常采取警报显示、控制显示以及维护显示等形式，其中警报显示接收并显示由加工厂中的控制器或设备产生的警报；控制显示指示加工厂中的控制器和其它设备的运行状态；维护显示指示加工厂中设备的机能状态。

然而，这些显示通常被预先配置来显示从加工厂中的过程控制模块或设备接收到的信息或数据。在一些公知的系统中，通过使用具有图形的对象来创建显示，其中图形与物理或逻辑元件有关，并且被通信连接到该物理或逻辑元件上以接收关于该物理或逻辑元件的数据。该对象可以基于接收到的数据来改变显示屏幕上的图形，以便示出诸如某个罐是半满的、流速传感器所测量的流速等。然而，由于单独的图形编辑器不能在操作员、维护人员、配置人员、仿真人员和管理人员的需要之间转换，因此目前需要使用不同的图形编辑器来彼此单独地创建用于配置、操作员控制、维护和仿真行为的图形显示。

如此，类似于控制配置应用程序，显示创建应用程序也可以具有模板图

形显示项目，例如罐、阀、传感器、例如滑动条的操作员控制按钮、开/关转换器等，它们可以以任何想要的配置被放置在屏幕上，以创建操作员显示、维护显示等。在把图形项目放置到屏幕上时，可以向用户提供加工厂内部工作的某些信息或显示的方式，在屏幕上互连各个图形项目。然而，为了激活图形显示，显示的创建者必须通过指定图形项目和加工厂中的相关数据源之间的通信链接来手工地把各个图形项目连接到加工厂中所产生的数据上，例如由传感器测量的数据或表示阀位置的数据等。这一过程单调、耗时，可能充满错误，并且必须被分别执行以用于过程控制人员、维护人员、仿真人员、配置人员和监督人员。

虽然控制配置应用程序中的控制模板对象和显示创建应用程序中的显示项目因为它们可以被拷贝并用来创建很多不同的控制模块和图形显示，所以显得比较方便，但是经常需要为加工厂中的不同设备创建许多相同的控制模块和图形显示。例如，大型加工厂的很多机构都具有相同或类似设备的许多实例，它们能够使用同一个基本通用的控制模块和显示来控制和查看。

然而，为创建这许多的控制模块和显示，先创建一个通用的控制模块或显示模块，然后把此通用控制或显示模块拷贝到它能够应用的每一个不同设备。当然，在拷贝后，每个新的控制或显示模块必须在配置应用程序中对其进行手工修改，以指定其所连接的特定设备，然后所有这些控制和显示模块必须被实例化并下载到过程控制系统中去。

不幸的是，上述控制模块和显示项目无论如何也不是模块化的。因而，在拷贝后，必须使用合适的配置应用程序来手工地、个别地修改各个控制模块和显示，以指定其将被关联的加工厂中的设备。在具有同一类型设备的许多拷贝（即重复的设备）的加工厂中，这一过程是单调、耗时且充满操作员所引入的错误。此外，一旦编好程序，这些不同的控制模块和显示彼此互不知晓。

因此，为了修改创建好的控制模块，工程师和操作员必须手工地对不同的重复设备的不同控制模块进行相同的修改，这仍是耗时和单调的。为加工

厂中重复设备的不同集合创建图形视图也存在相同的问题。换句话说，一旦创建了（个别地和从模板对象拷贝）特定的控制模块和特定的图形视图，并把它们连接到加工厂的特定设备集合上，该控制模块和图形视图就作为系统中独立的实体和对象存在，并不自动地知晓与它相同和类似的其它控制模块和图形显示。

结果，可应用于特定类型的控制模块和图形显示中的每一个的改变，却只能在它们的模块和显示上个别地完成。在为相同设备但是以加工厂中不同的功能关系创建图形视图，诸如为控制视图、维护视图和仿真功能创建图形视图时，这一问题更加明显。在此情况下，这些图形视图被单独地创建，彼此并不知道或知晓。

因而，当把这些图形显示提供给在加工厂中所进行的不同普通活动所使用的不同应用程序并与之关联时，通常在创建它们来支持应用程序的功能级处，添加这些图形显示以及有关的图形显示编辑器。结果，就图形编辑器目前的存在程度来说，它们仅使用户能够创建支持特定应用程序所需的特定功能的图形。之前的加工厂没有提供这样的图形显示编辑器：即能够由加工厂配置和支持的关系中所执行的各种或多种行为的图形需求来使用或者支持这种需求。因此，例如，用来支持控制配置行为或使该行为能够进行的图形显示编辑器仅使用户能够创建控制程序，但不能支持操作员或维护显示的需要或功能。类似地，在加工厂的运行期间，将被提供给控制操作员或维护技术员的、用于创建操作员视图和维护视图等的图形显示编辑器，也不支持与配置行为、仿真行为等有关的功能。

由于需要在加工厂的各个功能级，例如在控制配置、维护支持、控制操作员支持和仿真支持功能级，支持图形显示，因此这些不同编辑器所创建的不同显示无法对加工厂中同一组成部分进行建模和描述，这将导致加工厂中各种不同人员对图形显示的重复努力。这种重复努力不仅在创建用于描述用途不同的同一过程元件的不同图形显示所需的努力中比较明显，而且在把不同的显示应用程序中所使用的图形元素结合到与之关联的加工厂中的实际

硬件或软件元件上所需的努力中也比较明显。

因为有了这样的事实，已经提供了各种加工厂行为的图形支持，并且由于已经进行了部分实际行为，所以不能以如下方式把图形支持集成到加工厂环境中：即能够在加工厂中，在加工厂的各种不同功能级创建和使用公共图形。没有集成图形导致了实际上为不同功能所创建的图形在功能上，或在应用程序上彼此不同，这将导致一部分用户的混淆，这部分用户虽然熟悉一种特定类型的图形显示，但是可能偶尔需要查看与加工厂中的不同操作或功能有关的不同显示。

例如，与操作和维护有关的操作、维护和监督功能的图形支持通常是全然不同的，没有集成到一起的，并且涉及不同的技术且需要训练。另一方面，在加工厂的各个不同功能级（例如操作、维护、管理、安装和建立等）提供图形显示支持，导致在创建显示和在显示中把元素适当地连接到加工厂中实际的硬件或软件元素上时，需要复制各种图形支持元素。

因此，需要一种过程控制系统的公共图形用户界面，该图形用户界面中，支持诸如操作、维护、监督等过程或加工厂的各种功能级被集成到单个界面平台，以消除在使用非集成程序中所涉及的复制，并且更好地帮助所有的过程控制人员熟悉他们同事所使用的界面。这样的公共或通用图形界面应该容易地被终端用户修改或操作，同时使得诸如警报或其它紧急显示之类的一些显示不可遮盖。

## 发明内容

公开了一种用于加工厂环境中操作员、维护人员和执行其它各种功能的人员的图形界面，该图形界面向过程和工厂提供实时界面。该界面能够支持每种用户类型的各种专属性和功能。用户通过使用配置数据库，可以相对自由地利用多种浮动面板显示来操作他们的工作站。该配置数据库把数据源和事件绑定成低级形式。

在实施例中，该界面的配置数据库提供可下载的显示，该显示允许操作

员通过利用原始数据向有用动画、趋势和图案的快速且精确的转换，跟踪待控制的过程，这能够有助于确定该操作员所需的部分动作。然而，该同一界面还为维护人员提供关于由操作员正在控制的过程的适当显示，并且提供各种类型的报告数据以供管理人员使用。从该配置数据库还可以得到仿真和配置显示。所有这些功能都由一个界面平台支持。

在实施例中，操作员控制台的该界面图形被扩展为包括诸如维护和监督报告的附加功能。该界面可以被用在工作站中，该工作站可以包括一个或更多个监视器、标准或定制的键盘、声卡、扬声器和诸如鼠标、跟踪球和/或触摸屏之类的指示设备。所公开的操作员图形界面取代了塞满有单范例模拟控制器、仪表、量具和数字指示器的控制室面板。

在实施例中，所公开的界面包括下列功能中的一个或更多个：面板电动机启动/停止按钮、状态指示、图表记录器、信号器面板和子系统界面。

在实施例中，所公开的用户界面使操作员能够执行下列动作中的一个或更多个：管理警报、通过输入新的设定点或其它参数调整过程、“放大”过程的特定部分来查看进一步的细节以及使用与正在被执行和被控制的过程有关的其它专门的应用程序。

在实施例中，所公开的图形界面能在专用和非专用模式下运行。

在实施例中，所公开的图形界面能作为富客户端或作为使用网络服务的浏览器型界面的一部分运行。

在实施例中，所公开的图形界面能在工作站、膝上型电脑、平板 PC、手持设备和智能电话中运行。

在实施例中，所公开的界面支持用在控制室中或用在需要固定显示设置和/或严格控制存取的任何地方（即亭子型应用程序环境）的专用模式。

在实施例中，所公开的图形界面的配置能力支持由配置人员、工程师、加工厂管理者和其他人员使用的非专用模式，其中，界面的灵活性和丰富性是最主要的特征。

在实施例中，所公开的图形界面的配置能力支持多种用户界面设备，包

括但不限于富客户端、网络浏览器、手持设备和智能电话。

在实施例中，所公开的图形界面支持下列特征中的一个或更多个：集成声音和视频；诸如那些用在平板 PC 中的输入技术；包括所嵌入的历史数据、警报、事件等的实时数据服务；外部数据服务；XAML 文件；访问其它服务界面；处理包括复合结构的图形；基于类的控制分层；集成控制、告警和异常情况管理和预防；集成批处理操作员界面；集成先进控制操作员界面；路由管理；效率计算；最优化；质量和能量平衡；集成第三方应用程序；从 DeltaV®核心子系统分离应用程序；支持多重数据采集系统（DCS）；支持来自诸如 DeltaV® 和 PROVOX® 的多种系统的实时数据；支持来自诸如 DeltaV® 和 PROVOX® 的多种系统的告警；支持诸如警报栏之类的不能被遮盖的显示区域。

在实施例中，所公开的图形界面允许显示和其它应用程序被下载到运行期工作空间中。该运行期工作空间提供可以执行各种应用程序的安全环境。可以为包括但不限于智能电话、手持设备、平板 PC、膝上型电脑、工作站和多面板工作站的各种目标环境定制运行期工作空间。

在实施例中，所公开的图形界面提供建立完全集成的显示系统的能力。

在实施例中，所公开的图形界面提供绑定到数据值和事件上的全局跟踪。

在实施例中，所公开的图形界面提供跟踪到系统中使用显示地方的能力。

在实施例中，所公开的图形界面提供所述一个或更多个显示的完全版本跟踪。

在实施例中，所公开的图形界面提供链接来自其它源的数据的能力。

在实施例中，所公开的图形界面提供使用来自其它源的数据的能力。

在实施例中，所公开的图形界面提供定义可重用显示及显示组成部分库的能力。

在实施例中，所公开的图形界面提供定义工厂标准、锁定所述标准、翻译所述标准以及在不同功能级的多个用户之间容易地共享所述标准的能力。

在实施例中，所公开的图形界面提供用于开发加工厂人员在他们的各种各样设备上所使用的安全视图的装置。

在实施例中，所公开的图形界面提供给不同功能级的多个用户存取的、具有旋转和透明等特殊特征的全缩放图形。

对本领域技术人员来说，其它特征和优点将从下面参照附图的详细说明和所附权利要求书中变得明白。

#### 附图说明

所公开的统一的图形显示配置的各种特征都示出在附图中，其中：

图 1 为位于加工厂中的分布式过程控制网络的框图，其中该加工厂包括实现显示例行程序以及与该加工厂中的各种功能相关的其它应用程序的操作员工作站，还包括提供可用来为该加工厂的各种功能区域中的每一个创建显示元素和显示的系统级图形支持的工作站。

图 2 为集成加工厂控制中的系统级图形支持、查看和仿真系统的逻辑框图；

图 3 为示出创建图形元素和显示的配置环境以及执行图形元素和显示的运行期环境的逻辑图；

图 4 为运行期图形界面工作空间的示例，其中该工作空间能够被定制，以用于包括工作站、多面板工作站、电话、手持设备、平板电脑和膝上型电脑的各种目标环境；

图 5 为图 1 所示的图形界面的局部视图，其示意性地示出如下步骤：创建低级形状和形状合成、把数据源和事件绑定到数据源上、创建完全的显示、在配置数据库中存储显示、从配置数据库下载显示以及产生适于载入到富用户界面或网络浏览器中的配置数据库的运行时间可载入视图；

图 6 为支持包括过程控制模块示意图、数据趋势的图形示意图以及电子数据表或网格点数据表示的多种数据源的图形界面的示意图；

图 7 为支持来自多种数据源的数据的所公开图形界面的示意图；

图 8 为根据本公开所设计的图形编辑器工作空间的示意图；

图 9 示出一种启动图 8 所示编辑器程序的方法的工作空间视图；

图 10 示出图 9 中所描述的编辑器的浏览 (explore) 功能；

图 11 示出在诸如 PDA、智能电话或平板 PC 上的手持设备上的所公开显示；和

图 12 和图 13 示出所公开的图形显示编辑器到多面板或多屏幕工作空间的适用性。

应该理解，附图没有按照一定的比例绘制，并且所公开的实施例使用图解符号、概略说明和片断视图示出。可能省略了对于理解所公开的实施例不必要的细节，或是使另外的细节难以发现的细节。当然，还应该理解，本公开不限于这里所示出的特定实施例。

#### 具体实施方式

图 1 示出了示例性加工厂 10，在加工厂 10 中，给各种功能区域提供了系统级图形支持。正如通常的那样，加工厂 10 包括具有一个或更多个控制器 12 的分布式过程控制系统，每个控制器 12 都通过输入/输出 (I/O) 设备或卡 18 连接到一个或更多个现场设备 14 和 16 上，输入/输出设备或卡 18 例如可以为 Fieldbus 接口、Profibus 接口、HART 接口、标准 4-20 ma 接口等。控制器 12 也通过例如 Ethernet 链接的数据总线 24 连接到一个或更多个主机或操作员工作站 20-23 上。

数据库 28 可以连接到数据总线 24 上，并作为数据历史记录器运行，和/或作为配置数据库运行，其中数据历史记录器采集和存储与加工厂 10 中的控制器和现场设备相关的参数、状态和其它数据；配置数据库存储被下载到并存储在控制器 12 以及现场设备 14 和 16 中的工厂 10 中的过程控制系统的当前配置。另外，数据库 28 可以存储以这里所描述的方式创建的图形对象，以便提供加工厂 10 中的图形支持。控制器 12、I/O 卡 18 以及现场设备 14 和 16 通常向下设置在有时比较恶劣的加工厂环境中且分布于整个环境中，

而操作员工作站 20-23 和数据库 28 通常被设置在控制人员或维护人员容易接近的控制室或其它较不恶劣的环境中。

正如所公知的一样，每一个控制器 12，例如可以是由爱默生过程管理出售的 DeltaV™ 控制器，都存储并运行控制器应用程序，该控制器应用程序利用任意数目的、不同的、独立执行的控制模块或块 29 来实现控制策略。每个控制模块 29 都可以由通常所称作的功能块组成，其中每个功能块是整个控制例行程序的一部分或者子例行程序，并与其它功能块合作（经由被称作链接的通信）来实现加工厂 10 中的过程控制回路。

正如所公知的一样，可以是面向对象编程协议中的对象的功能块，通常执行输入功能，控制功能或输出功能之一，其中输入功能例如与变送器、传感器或其它过程参数测量设备相关；控制功能例如与执行 PID、模糊逻辑等控制的控制例行程序相关；输出功能控制诸如阀的某些设备的操作来执行某些加工厂 10 中的物理功能。当然，也存在诸如模型预测控制器（MPC）、优化器等的混合和其它类型的复杂功能块。虽然 Fieldbus 协议和 DeltaV 系统协议使用以面向对象编程协议设计并实现的控制模块和功能块，但是也可以使用例如包括顺序功能图、梯形逻辑等的任何想要的控制编程方案来设计控制模块，并且不限于使用功能块或任何其它特定的编程技术来设计和实现控制模块。

在图 1 所示的加工厂 10 中，被连接到控制器 12 上的现场设备 14 和 16 可以为标准 4-20 ma 设备，也可以为包括处理器和存储器的诸如 HART、Profibus 或 FOUNDATION™ Fieldbus 现场设备的智能现场设备，或者也可以为其它任何想要类型的设备。这些设备中的某些设备，例如 Fieldbus 现场设备（图 1 中以附图标记 16 标记），可以存储并执行与控制器 12 中所实现的控制策略相关的模块或诸如功能块的子模块。正如所公知的一样，图 1 中所示的作为设置在两个不同的 Fieldbus 现场设备 16 中的功能块 30，可以与控制器 12 中的控制模块 29 合作运行来实现过程控制。当然，现场设备 14 和 16 可以为任何类型的设备，例如传感器、阀、变送器、定位器等，I/O 设备 18 也可以为符合任何想要的诸如 HART、Fieldbus、Profibus 等通信或控制器协议的任何类型的

I/O 设备。

在图 1 的加工厂 10 中，工作站 20-23 可以包括用于由加工厂 10 中相同或不同人员执行的各种不同功能的各种应用程序。工作站 20-23 中的每一个都包括存储各种应用程序、程序、数据结构等的存储器 31，以及可以用来执行存储器 31 中所存储的任何应用程序的处理器 32。在图 1 所示的示例中，工作站 20 被设计为配置工作站，并包括一个或更多个配置应用程序 33，其中配置应用程序 33 可以包括例如控制模块创建应用程序、操作员界面创建应用程序和其它数据结构，这些数据结构可由任何被授权的配置工程师访问，以将控制例行程序或诸如控制模块 29 和 30 的控制模块创建并下载到加工厂 10 的各种控制器 12 和设备 16。

工作站 21 在图 1 中被总地示出为控制操作员查看工作站，并且包括多个显示应用程序 34。显示应用程序 34 在加工厂 10 的运行期间，可以把各种显示提供给控制操作员，以使该操作员能够查看并控制在加工厂 10 中或是在加工厂的各个部分中正在发生的情况。应用程序 34 可以包括支持应用程序 34a，例如控制诊断应用程序、调整应用程序、报告产生应用程序或可以用来帮助控制操作员执行控制功能的任何其它控制支持应用程序。

类似地，工作站 22 被示出为维护查看工作站，并包括可由各种维护人员来查看加工厂 10 的维护需要的多个维护应用程序 35，以便查看各种设备 12、14 和 16 等的运行和工作状况。当然，应用程序 35 可以包括支持应用程序 35a，例如维护诊断应用程序、校准应用程序、振动分析应用程序、报告产生应用程序或可以用来帮助维护人员执行加工厂 10 中的维护功能的任何其它维护支持应用程序。

另外，工作站 23 被示出为仿真工作站，其包括多个仿真应用程序 35，以帮助加工厂的维护和控制等，仿真应用程序 35 可以出于包括训练目的和加工厂建模目的的任意数目的目的，用于仿真加工厂 10 或加工厂 10 的各个部分的操作。正如通常那样，工作站 20-23 中的每一个都包括显示屏 37 以及例如键盘、鼠标等的其它标准外围设备。

当然，尽管各种配置、控制、维护和仿真应用程序 33-36 在图 1 中被示出为设置在专用于这些功能之一的不同工作站中，但是应该理解，根据加工厂 10 的需要和设置，与这些或其它加工厂功能相关的各种应用程序 33-36 可以设置在加工厂 10 的不同工作站或计算机中，并在这些工作站或计算机中执行。因此，例如，一个或更多个仿真应用程序 36 和控制应用程序 33 可以在工作站 20-23 的同一个工作站中执行，而不同的各个仿真应用程序 36 或不同的各个控制应用程序 33 也可以在工作站 20-23 中的不同工作站中执行。

过去，用在加工厂 10 的不同功能区域中的不同应用程序的开发是相当独立地进行的。因此，配置应用程序 33 的开发就没有与仿真应用程序 36、维护应用程序 35 或操作员控制应用程序 34 集成起来。实际上，在很多实例中，一个加工厂可能包括由不同的公司或软件厂商开发的、用于不同功能区域的应用程序，并且这些应用程序被实际上开发得独立于加工厂 10 内的其它软件运行。

由于这种与加工厂 10 的各种功能区域相关的不同应用程序的独立开发和操作，因此通常要求加工厂人员独立地在各个配置、操作员控制、维护和仿真功能级配置或设置加工厂。特别地，相同或不同的加工厂人员通常不得不使用不同的程序在各个功能级建立新的数据结构和图形显示。因此，对于图 1，执行配置、控制、维护和仿真功能的各种应用程序 33-36 中的每一个通常都包括或使用不同的图形显示编辑器和数据结构，以帮助加工厂人员进行这些配置、操作员控制、维护和仿真功能。

在很多实例中，使用这些不同的图形显示编辑器和数据库来创建不同的图形显示，以描述或建模加工厂 10 的相同部分或区域，或加工厂 10 中相同的硬件，并且帮助不同的加工厂人员在配置、操作员控制、维护或仿真行为的关系中可视化并理解加工厂中正在发生的情况。

因为应用程序 33-36 以及用于加工厂 10 中各个功能的每一个的相关显示通常是彼此独立地、有时是由不同的人员并且甚至是由不同的公司开发和实现的，所以从提供图形视像的观点来看，加工厂的不同功能区域中创建或

使用的图形显示没有以任何一致或容易理解的方式集成起来。结果，加工厂的各个不同功能级的图形显示的独立创建和执行，导致了功能与功能之间的图形显示看起来不同，以致在整个功能区域中，图形显示没有一致的外形和感觉。

另外，这种独立的创建还导致为了加工厂的相同部分或区域但为了不同的使用功能而创建图形显示中的重复努力，并且还要求在加工厂 10 的各个功能级独立地把这么创建的图形显示结合到诸如控制器 12 以及现场设备 14 和 16 的加工厂 10 中的各种设备上，并从这些设备接收数据。这一事实随之要求重复的数据库结构去追踪不同显示的相同硬件元素。因此例如，在过去，为了维护目的，使用第一应用程序（例如应用程序 35 中的一个）创建了一个示出加工厂 10 一部分的维护显示，而为了控制目的，使用第二应用程序（例如应用程序 34 中的一个）创建了一个示出该加工厂 10 同一部分的控制操作员显示。

用这些不同的显示编辑器独立创建的显示在外形和感觉上可能是完全不同的，这使得用户难以在不容易混淆的情况下或不需要在各种类型的显示上进行训练的情况下，在维护显示和操作员显示之间来回转换。同样地，独立地在不同应用程序 34 和 35 中创建显示，也存在重复努力，并且还需要另外的努力去创建数据库结构，以独立地把维护显示和控制操作员显示结合或连接到加工厂 10 中的同一硬件元素上，以便不时从这些硬件元素接收相同或类似的数据。

为了缓解这种效率低下，也为了提供更广泛的可用且可理解的加工厂 10 中的图形，在加工厂 10 的系统级提供一种图形支持层，以支持包括加工厂 10 的配置、操作员查看、维护查看、仿真和其它功能区域的各个加工厂功能区域的图形显示和数据结构需要。

这种系统级的支持被概略地描述在示出加工厂操作级 40、加工厂功能级 42 和系统级 44 的图 2 中。如同将要从图 2 中理解到的一样，加工厂操作级 40 包括控制器 12、现场设备 14 和 16 等，它们执行控制例行程序或模块

29 和 30，以及在加工厂 10 中运行来实现日常加工厂操作的其它软件。尽管还能够提供其它或不同的功能块，但是加工厂功能级 42 被描述为包括配置功能块 46、控制功能块 47、维护功能块 48 和仿真块 49。

配置功能块 46 实现配置例行程序 33，其中，配置例行程序 33 与加工厂操作级 40 的组成部分交互或通信，以向其提供控制策略或控制模块。控制功能块 47 包括控制查看和其它应用程序 34 和 34a，其通常也与加工厂操作级 40 中的各种物理和逻辑组成部分直接交互或通信，以在加工厂中实现操作员发起的改变、经由控制显示 34 向操作员提供信息、为控制应用程序 34a 获取数据等。维护功能块 48 包括维护例行程序和应用程序 35 和 35a，其与加工厂操作级 40 中的各个物理和逻辑组成部分交互或通信，以实现维护例行程序、采集维护数据、经由维护显示 35 向维护人员提供维护数据或信息、运行诊断应用程序 35a 等。同样地，仿真功能块 49 包括仿真例行程序 36，其实现加工厂 10 的仿真，并可以被通信连接到加工厂操作级 40 中的组成部分上以获取关于加工厂的数据。

如图 2 中所示，系统级支持层 44 被结合到加工厂功能层 42 中的各个功能块 46-49 中，并支持它们，以便例如使得能够创建并维护诸如软件对象、图形元素和用于各个功能区域 46-49 中的图形显示之类的公共数据库和显示结构。更具体地，系统级支持层 44 包括应用程序、数据库和图形支持元素，它们使各个功能块 46-49 中所执行的图形活动能够集成在一起，或者利用在系统支持层 44 所创建的公共数据库结构和图形元素被开发。

为了提供这一系统级支持，系统支持层 42 可以包括图形显示编辑器 50 和图形对象数据库 52。图形编辑器 50 创建图形元素 54 和图形显示 56，而图形对象数据库 52 把元素 54 和显示 56 存储到可由编辑器 52 和块 46-49 中的各个应用程序访问的存储器中。数据库 52 还可以存储其它对象 58 和把图形元素 54 连接到加工厂操作级 40 中的各个硬件和软件元素上的数据结构。另外，数据库 52 还可以存储图形元素或图形模板，或可以用来进一步创建图形元素或显示的图元。如同将从图 2 所理解到的一样，功能块 46-49 中的所有块

或任意块可以使用图形显示元素 54、显示 56 和其它数据库结构 58，来创建并使用与这些功能块有关的图形。

如同所理解的一样，系统级支持块 44 提供一种在所有功能区域中集成图 1 的加工厂 10 中所使用的图形的方式，从而减少或消除必须要为在不同的功能关系中的同一加工厂设备重复地创建不同图形元素的必要性，并且使得功能区域 46-49 中的每一区域中的用户都能够容易地结合进与正在与这些功能区域有关的图形视图中所显示的设备相关的数据。如同所理解的一样，系统级支持层 44 可以用来为功能区域 46-49 中的每一区域中的多个应用程序，并且为功能区域 46-49 中的不同区域中的不同应用程序等提供图形和数据库支持。

再次参见图 1，可以采用另外的工作站或用户界面 60 来实现系统级支持块 44，其中工作站或用户界面 60 可以被连接到其它工作站 20-23 中的每一个上。工作站 60 通常可以存储图形编辑器 50 和数据库 52，并且如果需要的话，也可以存储其它元素 54、56 和 58。另外，工作站 60 可以经由数据总线 24、经由单独的有线或无线通信连接（图 1 中以虚线示出）或者以其它任何想要的方式通信连接到工作站 20-23 上。

在图 1 所示的配置中，工作站 60 存储并执行显示编辑器 50，以使用户能够创建图形元素，并把这些元素编组成一个或更多个图形显示，在这里，图形元素和图形显示都被称作显示模块。然后，这些显示模块可以被存储到数据库 52 中，以便由图 2 中所示出的功能块 46-49 中的各个块访问并使用，并且在工作站 20-23 上实现。虽然，为了示出，系统级块 44 和功能级块 46-49 的功能被示出为在图 1 中不同的或单独的工作站 20-23 和 60 中实现，但是应该理解，与这些块中的任何一个有关的所有或任何一个应用程序可以在加工厂 10 中或与加工厂 10 有关的相同或不同的工作站或其它计算机中实现。因而，图形编辑器 50 可以被存储到其它工作站 20-23 或与加工厂 10 有关的任何其它计算机中，并在其上执行，并且不需要位于单独的或孤立的计算机中。

如上所述，图 2 的系统级层 44 实现了系统级显示和数据库对象，其可用于多种功能环境中，也能提供更高级的显示能力。一般来说，在图 2 的系统级 44 处所创建的显示对象可以分为图形元素和图形显示。图形元素通常为图元或与加工厂中的特定物理实体有关的低级显示对象，例如象阀、传感器、泵、控制器等的硬件设备。图形显示通常由一套互连的显示元素构成，并且用来表现和建模加工厂中的更加复杂的硬件集合，例如单元、区域等，图形显示还包括不同硬件单元之间的互连关系。另外，图形显示还可以包括曲线图、图表和从加工厂和其它应用程序提供的其它数据，例如在工作站 20-23 和 60 中运行的诊断和商务应用程序等。

图 3 总体上示出了两个环境中的图形元素和图形显示的开发和使用，这些元素和显示可以存在于这两个环境中，具体地，存在于在配置环境 70 和运行期环境 72 中。一般来说，形式为图形元素 74（被描述为单独的元素对象 74a、74b 等）的显示对象和图形显示 76（被描述为单独的显示对象 76a、76b 等）利用诸如显示编辑器 50 在配置环境 70 中创建。对象 74 和 76 在创建后可以被存储到数据库 52 中。

对象 74 和 76 可以被创建为在这里被称作显示模块类对象的类对象，显示模块类对象定义了没有与加工厂中的具体硬件或逻辑元件绑定或结合的通用对象。然而，可以使用类对象来创建与该类对象具有相同的基本属性、但却被绑定或结合到加工厂中的具体硬件上的运行期图形对象。然而，一般来说，类对象保持与从其生成的子对象结合，因而，对类对象的改变可以自动地遗传给其子对象，即使这些子对象实例化在运行期环境中也是如此。

如图 3 所示，每个图形元素对象 74 都包括使该图形元素可用于很多不同关系中的多个组成部分。具体地，每个图形元素 74 都包括一个或更多个视像 77、任意数目的参数或属性 78、可以被作为脚本或触发器实现的任意数目的动作或动画 79 以及绑定 80。一般来说，每个视像 77 定义了当在运行期环境 72 中实现该图形元素时，将被在显示屏上实际显示的视觉属性或元素。

尽管视像可以表现其它实体，但是通常，视像定义了物理或逻辑设备或

设备组的表现。视像 77 可以在运行期环境 72 中使用任何想要的描述或定义了实体的图形描述细节的编程范例来实现。在一个实施例中，可以使用 PGXML 或 Avalon 控制来实现视像 77，PGXML 或 Avalon 控制是由 Microsoft® 提供的公知控制，并且由于它们是基于对象的，因此提供了一种使得视像容易在标准的 Windows 型显示器中实现并且在显示环境之间可移动的方式。

一般来说，参数和属性 78 定义了与由视像正在描述的实体有关的变量或诸如静态的或可变的固有属性之类的其它属性。同样地，动作和动画 79 定义了程序（可被作为脚本实现）、例行程序或在视像 77 被描述在显示屏上时，将在视像 77 上执行或利用视像 77 执行的行为，以提供给视像更多有趣的、可理解的或有帮助的图形属性。在一种情况下，这些动作或动画可以采取改变视像的各种组成部分的颜色和大小（例如高度、宽度、线的尺寸、字体等）、颜色填充、诸如颜色改变、旋转、尺寸的改变和缩放、倾斜之类的形式。

这些动作和动画把图形属性以及用户交互属性提供给图形元素 74。绑定 80 可以为静态的或固定的绑定，或可以为使用别名的绑定，绑定 80 定义了这样一种方式：即当图形元素 74 在运行期环境 72 中作为显示的一部分被实现时，参数或属性 78 被绑定到数据、标签和运行期环境 72 中的其它实体上。主要地，用于各个图形元素 74 的绑定 80 建立这样一种方式：即图形元素 74 被绑定到加工厂环境中其它地方所定义的一个或更多个实体或数据元素上，从而定义实际的运行期环境 72 和图形元素 74 之间的界面。

如图 3 所示，每个图形显示对象 76 都包括很多组成部分，例如对一个或更多个图形元素 81、连接器元素 82、动作和动画 83、属性 84 和绑定 85 的参考。一般来说，图形显示 76 可以为描述各种图形元素 81 的交互的显示，其中各种图形元素 81 可以与表示管、线、传送带等的连接器元素 82 视像地连接在一起。图 3 中的虚线示出了图形显示 76a 对图形元素 74 之一的参考。应该理解，参考图形元素的图形显示包括该图形元素的所有属性、参数、动作和动画等。

与图形元素 74 类似，每个图形显示 76 可以包括一个或更多个动作或与执行例如显示上的动画、用户界面交互和数据操作等有关的动画。同样地，每个图形显示 76 可以包括任意数目的与显示有关的属性，通常这些属性定义了单元、区域或显示中所描述的其它元素组的属性。当然，绑定 85 定义了这样一种方式：即图形显示 76 被绑定到在加工厂环境中其它地方所定义的一个或更多个实体或数据元素上，并从而定义了实际的运行期环境 72 与图形显示 76 之间的界面。

图形显示 76 和图形元素 74 一旦被创建，就可以被绑定到图 1 的工作站 20-23 中的任何一个上的运行期环境 72 中，并在其中执行。特别地，在图形元素 74 或图形显示 76 被创建为类对象，并被存储在数据库 52 中后，该元素或显示就可以被实例化为实际的运行期对象，并且可以在运行期环境 72 中执行。

如框 86 所示，实例化过程把对象 74 和 76 中定义的绑定填进可以在加工厂或过程控制系统中以合适的变量名、标签和别名等加载的解析表中，以提供加工厂中的实际实体和加工厂中的显示设备上运行的显示对象或显示模块之间的具体连接。作为该绑定过程的一部分，显示对象 74 和 76 连接到如解析表所定义的加工厂内的数据源上，从而访问加工厂，并把实例化的显示对象逻辑地且通信地连接到加工厂 10 上。

如框 87 所示，显示元素 74 或图形显示 76 可以在运行期环境 72 内的多个不同功能中执行或作为这些功能的一部分执行，包括命名的不只这几个的配置显示、控制操作员显示、维护显示和仿真显示。另外，显示对象 74 和 76 还可以用于执行系统级功能，例如使用来自图 2 中所描述的各个功能级的数据的那些功能，例如包括预测控制或预测维护功能，系统级错误检测等。实际上，显示一旦被创建在配置环境 70 中，并被存储在数据库 52 中，就可以用于多种不同的动作。

另外，显示对象 74 和 76 可以在任何想要的显示器或计算机硬件，或其它任何专用显示器 93 上执行，计算机硬件例如有工作站 90、膝上型电脑 91、诸如个人数字助理（PDA）之类的手持设备 92，专用显示器 93 例如具有多

个监视器的大屏幕显示器等。如果需要的话，单一图形显示 76 可以被分层为包括一个或更多个视图，例如配置视图、操作员视图、维护视图和仿真视图。可替代地，单独的图形显示 76 可以被配置成使用相同或类似的图形元素 81 来提供这些单独的视图，以便通过为这些各种功能创建的显示提供一致的外形和感觉。

如框 95 所示，为了将图形元素 74 或图形显示 76 移动到运行期环境 72 中，图形元素 74 或图形显示 76 可以被拷贝或实例化，并加载到运行期机器中。一般来说，仅当调用显示对象 74 或 76，或在运行期机器上实际执行它们时，才需要把它们绑定到运行期环境 72 上，这在这里被称作运行期绑定。也就是说，只有当显示对象在运行期计算机中实际运行或执行时，才将用于每个实例化对象的解析表填充进或绑定到运行期环境上。这一过程确保仅当显示对象的视像实际需要在显示屏幕上显示时，包括其视像、控制和脚本等的显示对象才执行，并因此使用处理能力。

换句话说，显示对象优先仅在该对象在运行期计算机中实际运行时，才被绑定到运行期环境 72 上。因而，显示对象 74 和 76 可以由查看这些对象所创建的视像的用户的动作所定义的方式，间歇地连接到运行期环境 72 上。特别地，这些对象可以在需要查看它们的时候被绑定到运行期环境 72 上，并且可以在用户不看它们的时候，例如在用户最小化或关闭这些对象正在提供视像的屏幕时，被解绑定或释放。

因而，显示对象 74 和 76 是可以在孤立环境中即配置环境 70 中创建的简单对象，但是它们可以与加工厂内或运行在加工厂环境中的任何应用程序所定义的其它对象或数据结构结合或连接，例如包括任何控制、仿真、维护或配置环境中所定义的对象或数据结构。另外，显示对象 74 和 76 一旦被创建，就可以经由解析表所定义的直接的参考、变量或标签被明确且直接地绑定到那些物理或逻辑过程实体上，或者通过使用别名名称、变量和参数间接地绑定到那些物理或逻辑过程实体上，这可以在显示对象 74 或 76 被下载到运行期环境 72 中或在运行期环境 72 中被实例化时进行，或者在某些情况下，也可以在

显示对象 74 或 76 在运行期环境 72 中实际运行时进行。

显示编辑器 50 可以使得能够以各种细节级别来创建显示对象 74 和 76，以提高易用性和显示对象 74 和 76 的多功能性。例如，可以首先创建图形元素 74 来定义原始的物理和逻辑实体的属性和操作，然后，可以通过互连一个或更多个图形元素 74 来创建图形显示 76，以创建用于描述更复杂的物理或逻辑实体或物理实体组或逻辑实体组的更高级别或更复杂的显示。

图 4 为包括不可遮盖的较低的警报栏 111 的显示 110 的示例。显示 110 还包括可沿显示 110 的顶部边缘延伸的配置栏 112。警报栏 111 和配置栏 112 的位置明显是可互换的，并且可以被设置在显示 110 的其它部分或方位。如图所示，显示 110 包括 3 个主要显示区，即通知区 113、主区 114 和可以包括可加载面板、趋势和其它查看选项的附加区 115。通知区 113 主要用于适用于操作员、维护人员和监督人员的通知。主显示 114 可以包括设备模块或整个系统（例如参见下面的图 3）的图形显示，或者可以包括数字、文字或图形形式的数据。由于过程控制警报及其通知除了对操作人员是重要的，对所有人员也都是重要的，因此警报栏 111 不能被遮盖很重要。在实施例中，警报栏不能缩放或移动。然而，移动警报栏 111 可能对于非操作人员来说是有用的，这些人员比起管理人员、维护和配置工程师来说较少关心警报栏。

显示 110 的配置功能 112 使用户能够把显示 110 配置为操作控制显示、维护显示、配置显示、仿真显示或用于管理人员的显示或上述显示中任意一个或更多个的组合。此外，这里所公开的显示可以被示出在单监视器工作站上、多监视器工作站上、PDA 设备上、电话上或平板 PC 型的设备上。

根据本公开的显示 110 的配置包括低级形状和形状合成 77-80（参见图 3）的创建、如图 3 中 86 处所示的数据源和事件到这些数据源上的绑定、完全显示 110 的创建、显示 81-85 在配置数据库中的存储、从配置数据库下载显示和产生用于加载进富用户界面或网络浏览器的、配置数据库的运行时间可加载的视图。图 5 中示出了显示 110 这种配置的一个示例，由此运行期功能可以从配置栏 112 获取，并且包括显示加载、运行时间加载和到数据源的

绑定或者连接，还可从配置栏 112 或运行期仿真获取作为有支持的对象和预测故障。箭头 117 和 118 表示从配置数据库下载各种脚本，配置数据库可以从配置栏 112 访问。

转到图 6，其中示出了修改的显示 110a，其具有主要或主显示 114a、114b 和 114c 中的不可遮盖的警报栏 111。在 114a 中，示出了包括泵 121、控制阀 122 和积聚罐 123 的设备模块的一部分的示意性示图。随着正在填充罐，数据表 124 指示填充罐 123 的情形。然而，另外的信息可能是必需的，因此显示 114b 包括填充罐 123 的图形描述 125，显示 114c 包括以数字形式示出正在进行数据累加的电子数据表或网格 126。以此方式，可以用逐渐增加细节级别的方式布置显示，从综述显示工作开始，下至集中系统的一个方面的细节显示。利用所公开的界面可以很容易地配置这些和其它显示，并且对于本领域技术人员来说，其它变动也会很清楚，下面将讨论一些示例。

转到图 7，本公开的显示图形软件与其它数据源和程序可以互换，包括委托人自己的 DeltaV® 软件、OVATION® 软件，PROVOX® 软件，和开放过程控制（Open Process Control，OPC）软件。这里所公开的图形显示界面还能与如图 7 中所示的竞争数据库和软件通信。

图 8 为操作员设计下的图形显示 110b 的一个示例，操作员的设计由在较低处的工具栏 152 中被激活的按钮 151 来指示。图形分层视图 153 用来示出显示 110b 的图形对象层次。由于显示 110b 正在被设计，所以属性视图 154 最初显示文本属性，然后改为显示新选中的工具或当前所选择的对象的属性。模板视图 155 显示过程元件 156a-156d 的多个模板。显示 110b 中所使用的外部参考列表可以在 157 处所示的绑定视图中找到。对于每个用户创建或修改的显示或元素，将以标题栏中出现的文件名打开一个单独的文档窗口。

图 8 中所示的显示编辑器还包括下列特征。菜单 171 包括可以被应用到特定图形元素上的所有用户命令。工具栏 172 包括到各种菜单命令的快捷方式。图形分层 152 包括 2 个元素，即树状视图 153a 和列表视图 153b。属性视图列出特定图形元素的固有属性和事件。主屏幕 174 是主要编辑区域。模板 155

列出在系统中所定义的各个图形元素 156a-156d。图 8 中所示的界面还包括水平标尺 175 和竖直标尺 176。图 8 中所示的实施例能够支持包括操作层 151、维护层 177、配置层 178、仿真层 179 和监督或管理层 180 在内的 5 层。所有的这些层或多或少地是能够被使用的功能。另外，还可以使用训练层。

图 9 是一种启动图 8 所示的图形程序的方法的示例。图 10 是示出已经被创建并存储到数据库的各种文件中的各种显示的屏幕局部视图。当选择“Display” 158 时，文件 159 和 160 将出现在图 10 所示的主屏幕 161 中。

转到图 11，其中示出了可以在 PDA、平板 PC 或智能电话型设备中使用的样本显示 110b。与图 1-10 有关的上述图形显示编辑器可以在这些类型的手持设备中使用。

此外，如上所述，上述显示编辑器可以在诸如图 12 和 13 所示的那些多屏幕工作站中使用。在 110c-110h 处所示的显示都可以被配置成用于操作或过程控制人员、维护人员、配置人员、仿真人员或监督人员。各个显示 110c-110h 可以分别是过程控制、维护、仿真、配置或监督显示，或者每个单独的显示面板 110c-110h 可以是图形显示编辑器可支持的各种功能中的任意功能的组合。从而，本公开的系统为单显示器型工作站的单显示器系统或手持设备以及多显示器工作站提供了多种功能显示。

尽管只说明了特定实施例，但是对于本领域技术人员来说，各种修改的可替代实施例将从上述说明中变得明白。这些和其它可选择的被认为是等同的，并且处于本公开的精神和范围之内。

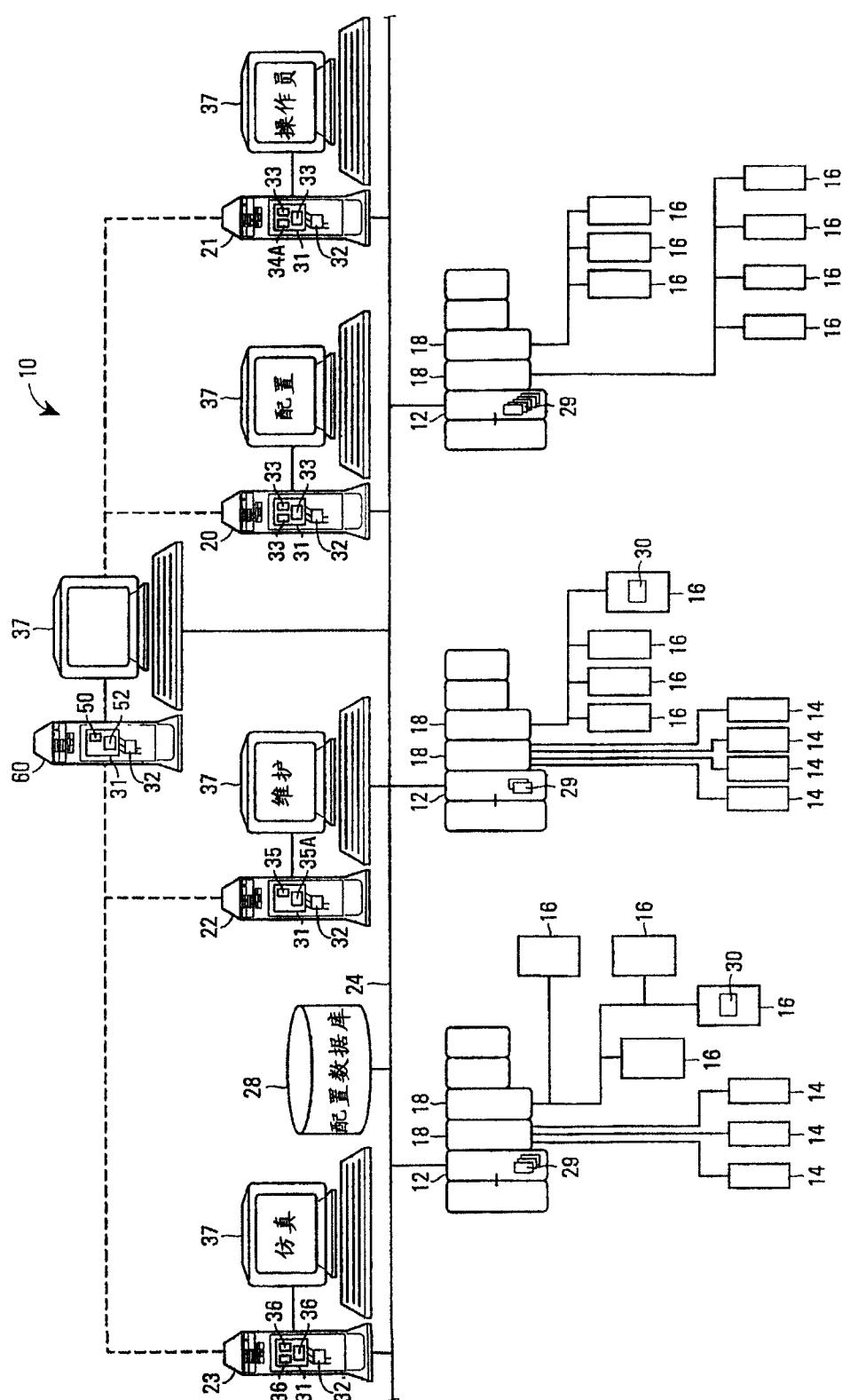


图 1

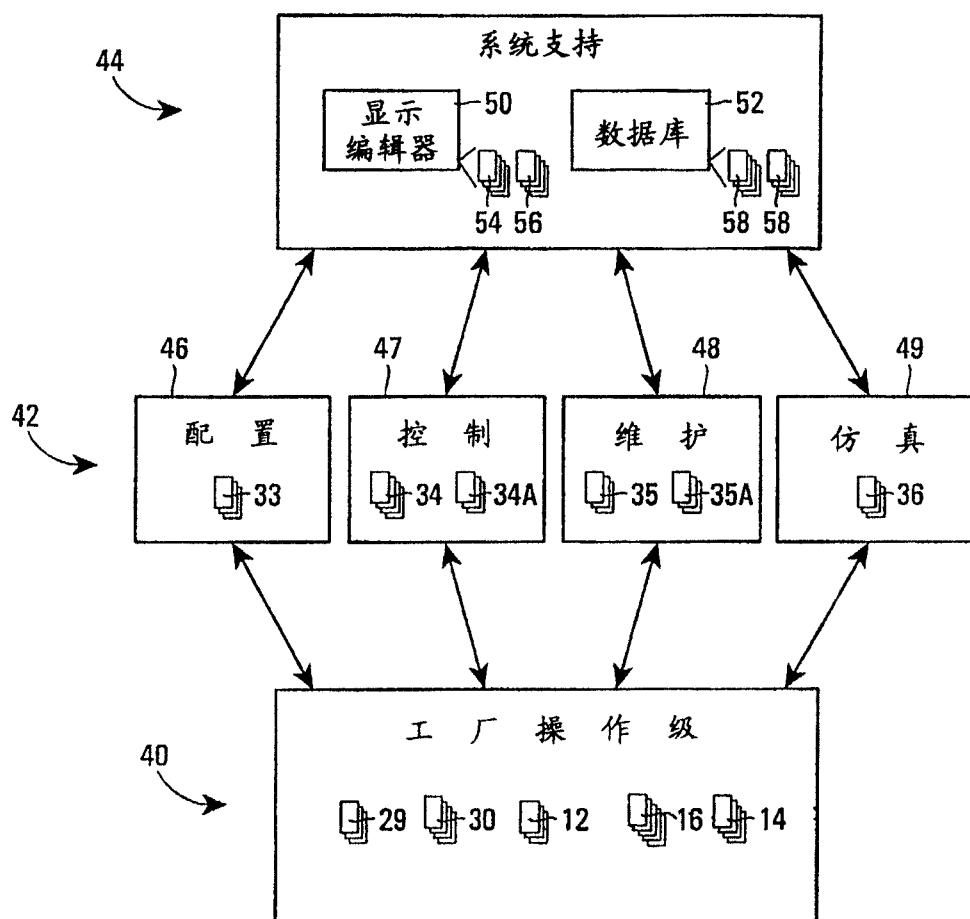


图 2

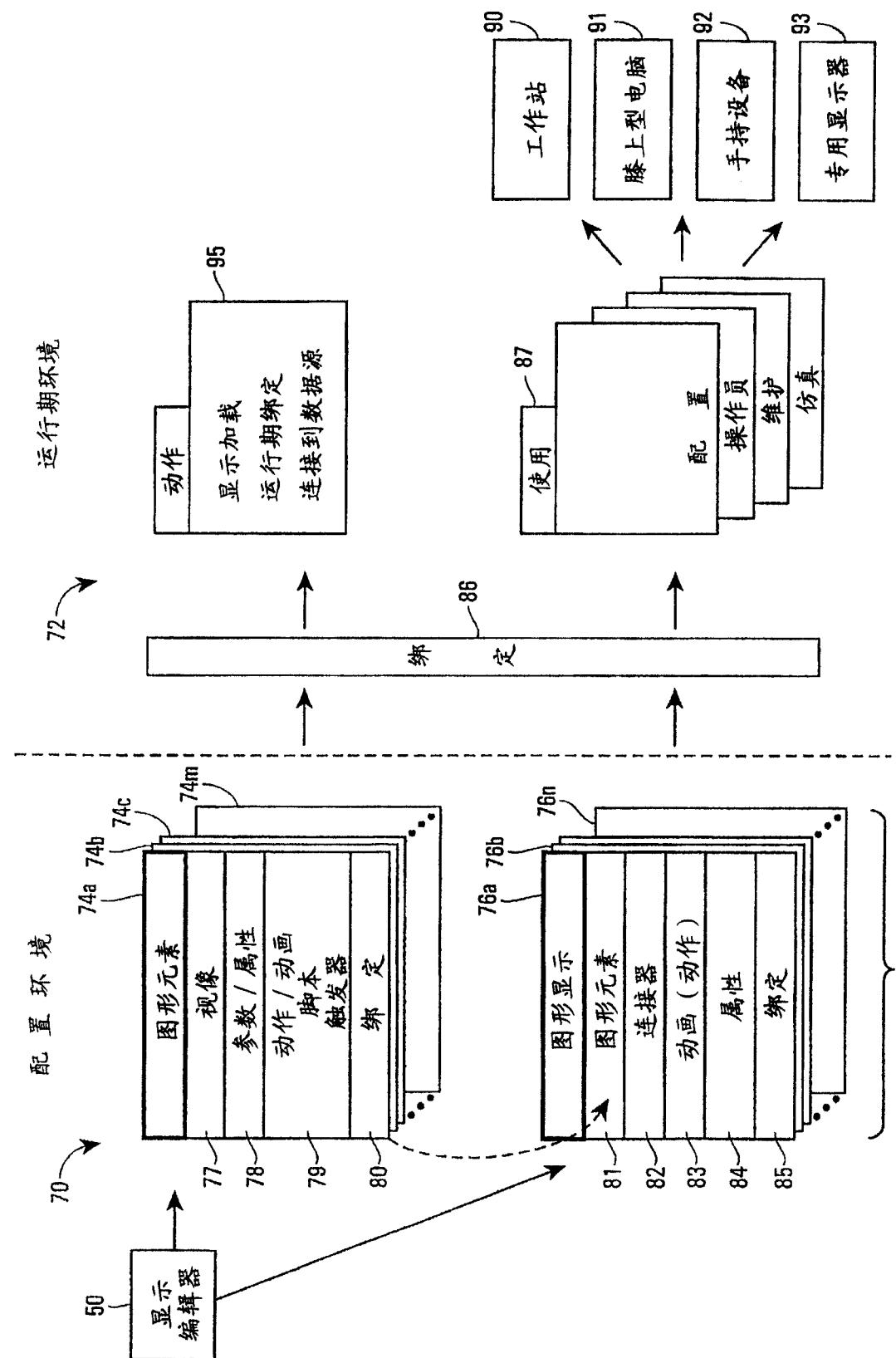


图 3

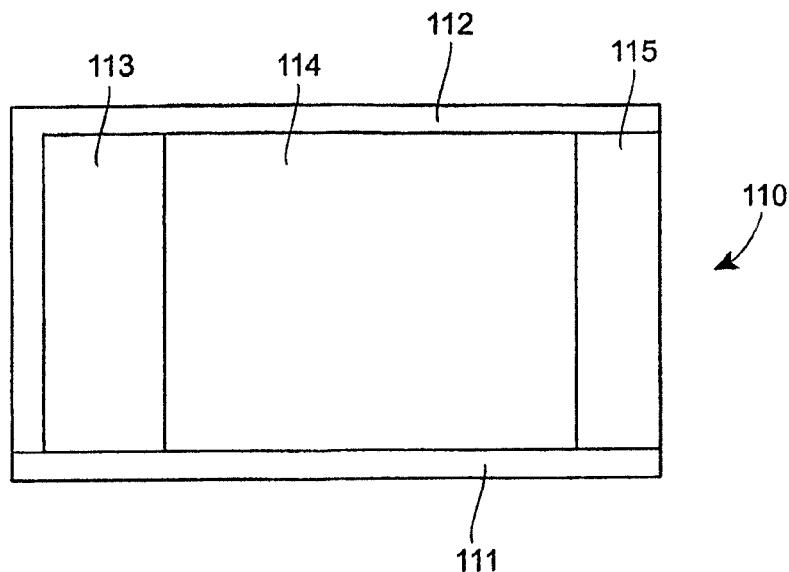


图 4

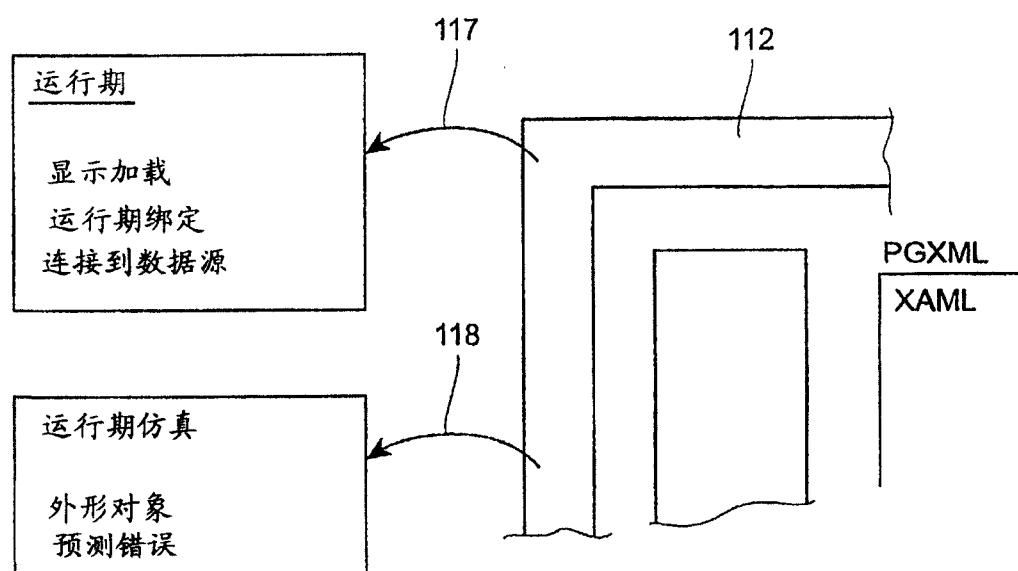


图 5

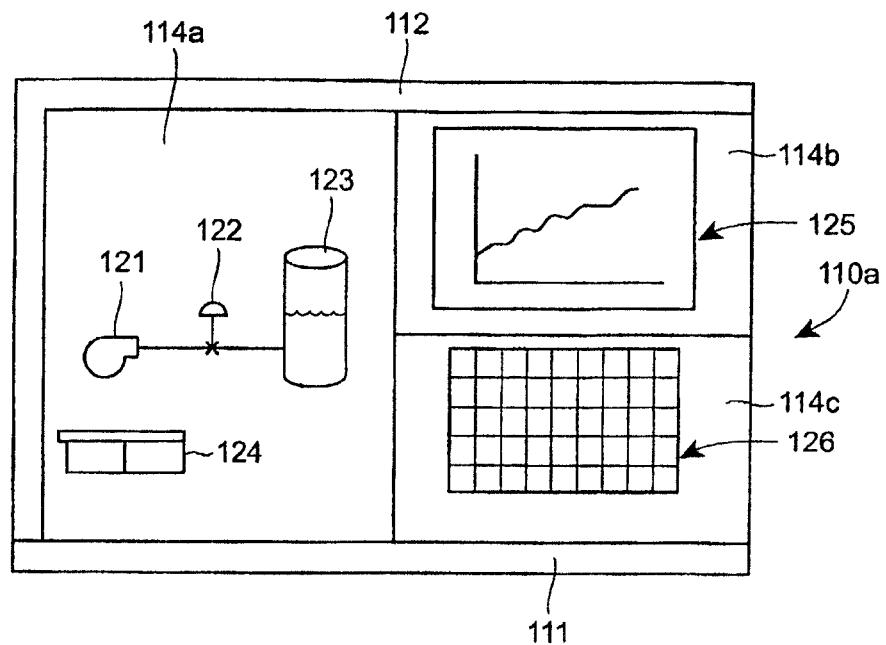


图 6

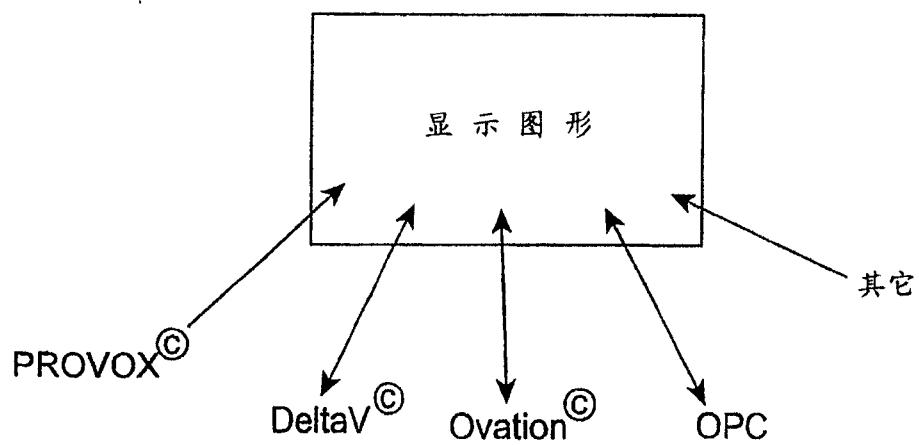


图 7

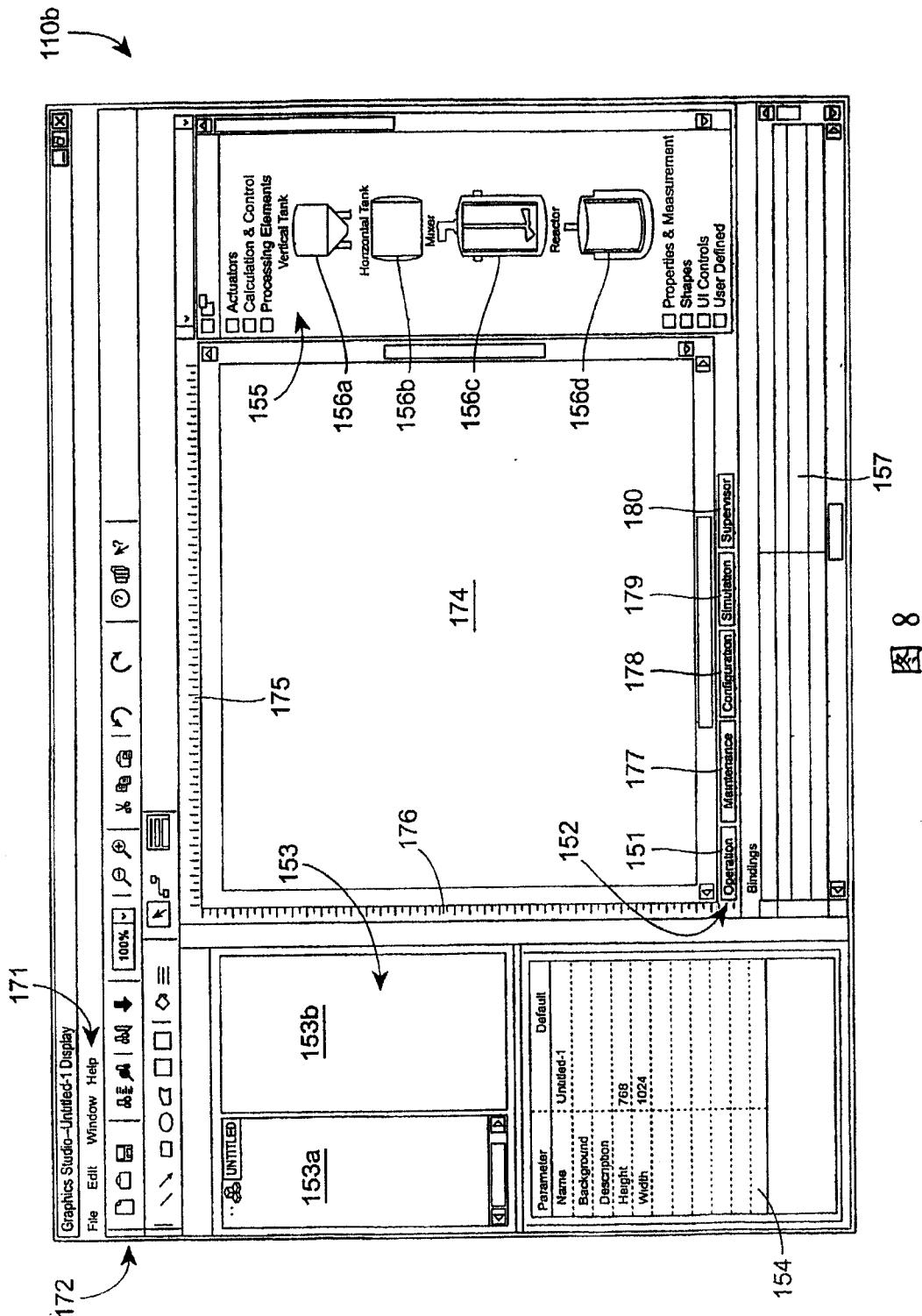


图 8

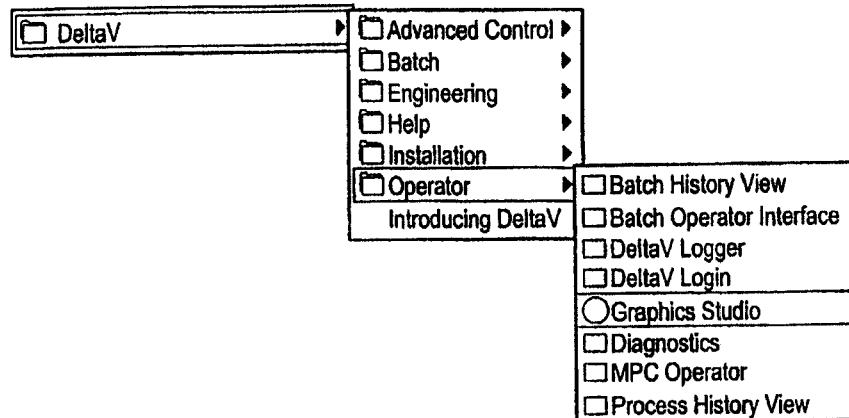


图 9

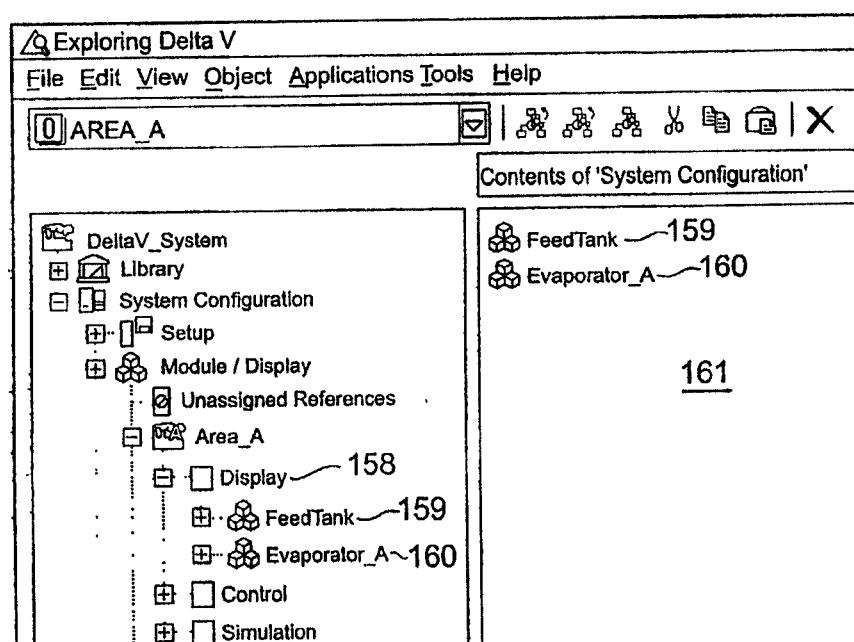


图 10

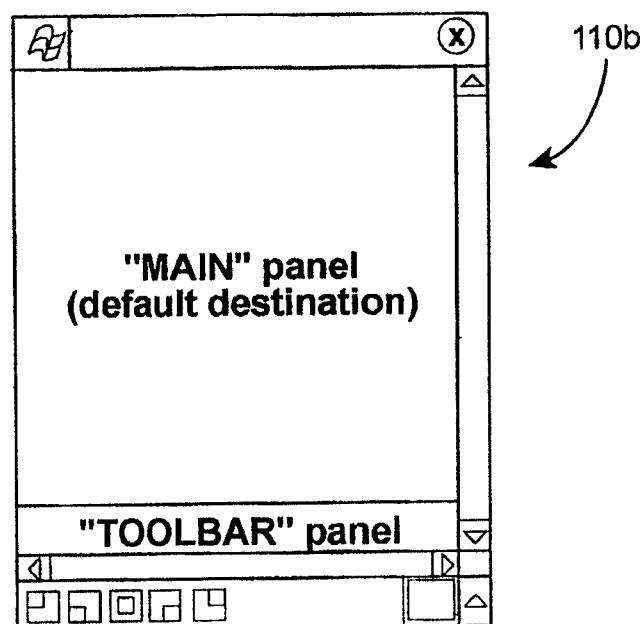


图 11

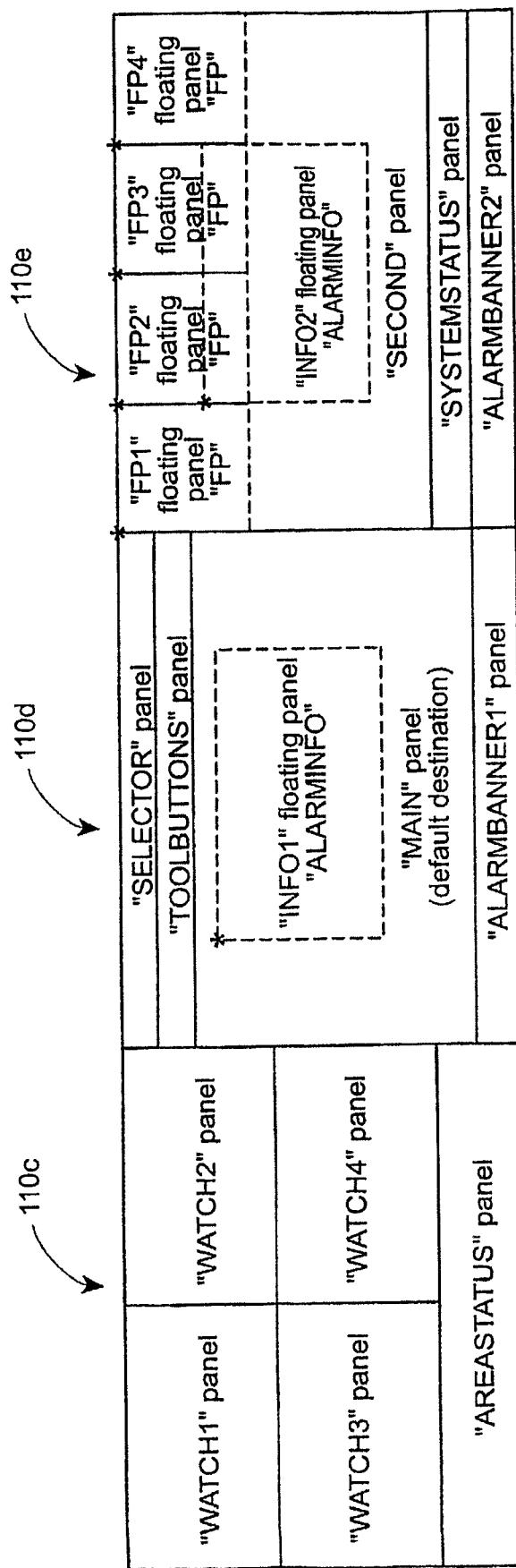


图 12

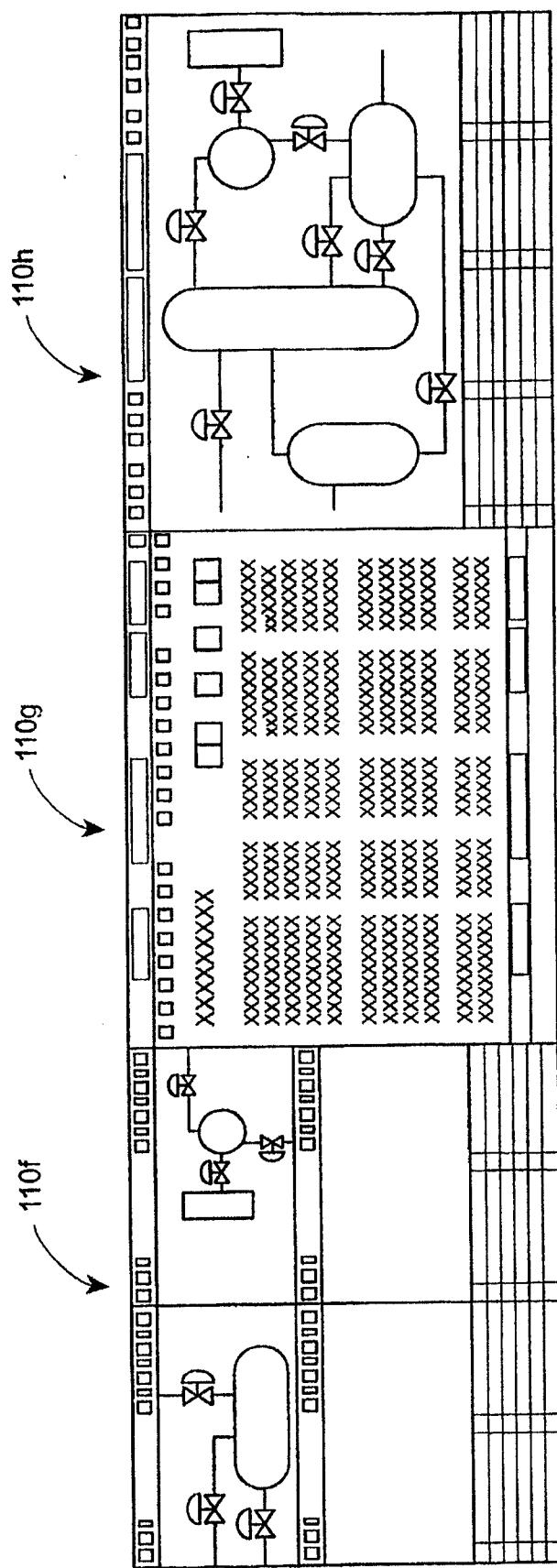


图 13