



(12) 发明专利

(10) 授权公告号 CN 1950761 B

(45) 授权公告日 2012. 07. 04

(21) 申请号 200580014273. 0

(22) 申请日 2005. 05. 04

(30) 优先权数据

60/567, 980 2004. 05. 04 US

(85) PCT申请进入国家阶段日

2006. 11. 03

(86) PCT申请的申请数据

PCT/US2005/015942 2005. 05. 04

(87) PCT申请的公布数据

W02005/109130 EN 2005. 11. 17

(73) 专利权人 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 斯蒂芬·吉尔伯特 坦尼森·郝

弗朗西斯·德卡兹曼

肯·J·贝欧格特 布鲁斯·坎普尼

马克·J·尼克松

(74) 专利代理机构 北京德琦知识产权代理有限公司 11018

代理人 陆弋 朱登河

(51) Int. Cl.

G05B 19/042(2006. 01)

(56) 对比文件

US 5841654 A, 1998. 11. 24, 说明书第 5 栏第 8~11 行, 第 35~37 行, 第 6 栏第 24~34 行, 第 61~64 行, 第 7 栏第 52~60 行, 第 8 栏第 62~63 行, 第 10 栏第 37~67 行, 第 11 栏第 40~43 行, 第 12 栏第 15 行至第 13 栏第 15 行, 图 6J-6N.

US 5706455 A, 说明书第 3 栏第 43-45 行, 第 8 栏第 36-39 行, 图 5.

US 2002/0077711 A1, 2002. 06. 20, 说明书第 21 页第 136 段, 图 9.

审查员 魏桂芬

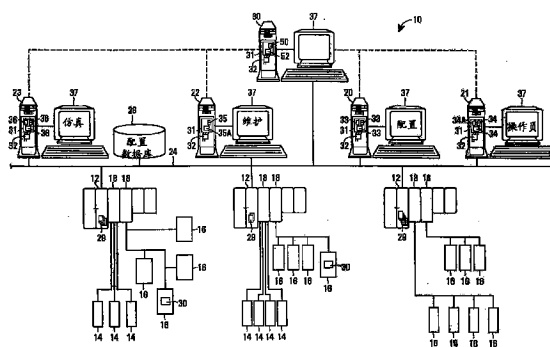
权利要求书 2 页 说明书 40 页 附图 24 页

(54) 发明名称

过程环境中的脚本图形

(57) 摘要

图形元素和图形显示被应用于过程环境中、为一个或多个用户显示与过程环境相关的信息, 例如加工厂内设备的当前状态。图形元素和显示包括一个或多个对象, 每个对象包括一个描述过程环境内实体的视像, 一个与所描述过程实体相关联的属性以及一个例行程序, 所述例行程序与视像和属性共同作用从而基于属性变化来引发视像的变化, 或者基于用户通过视像的输入来引发过程环境的变化。这样, 图形对象可以用于建立显示, 所述显示能够通过动画和其他视像操作来图形化地描述过程的变化状态以及过程中的被检测事件。这些相同的图形元素可以用于使用户能够在用户界面屏幕上与一个视像进行交互, 例如通过鼠标或键盘设备, 从而引发对象属性的变化, 然后所述属性变化可以为加工厂内的设备提供输入。



1. 一种图形显示编辑器,其在加工厂中使用以创建图形显示,所述图形显示呈现了所述加工厂内的一个或多个过程实体的操作的视像,所述过程实体是在用于控制所述加工厂内的过程的过程控制系统内的物理设备,所述图形显示编辑器包括:

图形对象的库,每个图形对象包括在所述加工厂内的过程实体的视觉表现;

基于图形的编辑画布例行程序,其通过将来自所述图形对象的库的所述图形对象的一个或多个视觉表现放置在编辑画布上以定义所述图形对象的一个或多个视觉表现在所述图形显示的执行期间将要在显示设备上向用户显示的方式,使用户能够定义可执行图形显示;

属性定义画布例行程序,其适于使用户能够定义与所述图形对象中的至少一个相关的属性;

绑定定义例行程序,其适于使用户能够指定在与所述图形对象相关的属性和运行时间环境中由所述图形对象所表现的过程实体之间的绑定;和

定义例行程序,其适于使用户能够定义显示例行程序,所述显示例行程序与所述图形对象的视觉表现以及与所述图形对象相关的属性协同操作,以基于所述属性的值来检测加工厂环境内的过程实体的情况,并且当所述图形对象的视觉表现被描述在显示屏幕上时,在所述图形显示的执行期间,基于所检测到的情况或由所述图形对象所表现的过程实体的属性改变所述图形对象中的所述过程实体的视觉表现。

2. 根据权利要求 1 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为动画例行程序,该动画例行程序将所述图形对象中的至少一个的视觉表现做成动画。

3. 根据权利要求 2 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为动画例行程序,该动画例行程序以连续方式将所述图形对象中的至少一个的视觉表现做成动画。

4. 根据权利要求 2 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为动画例行程序,该动画例行程序通过对所述图形对象中的至少一个的视觉表现应用弯斜、旋转、平移和调整大小中的至少一种,来将所述图形对象中的至少一个的视觉表现做成动画。

5. 根据权利要求 2 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为动画例行程序,该动画例行程序通过改变与所述图形对象中的至少一个的视觉表现相关的颜色,来将所述图形对象中的至少一个的视觉表现做成动画。

6. 根据权利要求 1 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为可执行转换例行程序,其将从所述加工厂中的数据源接收的所述属性的属性值进行转换。

7. 根据权利要求 6 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述可执行转换例行程序定义为一种转换,该转换将所述属性值转换为颜色、或计数列表值中的一个、或长度、或字体名称、或局部字符串、或持续时间、或旋转。

8. 根据权利要求 1 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为可执行例行程序,该可执行例行程序检测与所述过程实体相关的情况,向所述图形对象中的至少一个的视觉表现指示所检测到的变化情况。

9. 根据权利要求 1 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为一种使用基于所述属性的值而选出的不同动画来改变所述图形对象中的至少一个的视觉表现的例行程序。

10. 根据权利要求 1 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为一种通过所述图形视像接收来自用户的输入的例行程序。

11. 根据权利要求 10 所述的图形显示编辑器,其中所述定义例行程序使用户能够将所述显示例行程序定义为可执行例行程序,该可执行例行程序使用所述输入来引起所述属性的变化,从而形成所述显示设备外部的运行时间环境中的变量。

## 过程环境中的脚本图形

[0001] 相关申请

[0002] 本申请是常规提交的申请,并要求对于 2004 年 5 月 4 日提交的、名称为“用于对过程控制系统进行表现、监测和交互的图像用户界面 (Graphical UserInterface for Representing, Monitoring, and Interacting with Process ControlSystems)”、序列号为 60/567,980 的美国临时专利申请的优先权,该申请被全文合并在此作为引用参考。本申请还涉及名称为“加工工厂中的图像显示元素、过程模块和控制模块的集成 (Integration of Graphic Display Elements, ProcessModules and Control Modules in Process Plants)”的序列号为 10/625,481 的美国专利申请,该申请于 2003 年 7 月 21 日提交,并于 2004 年 8 月 5 日被公开为美国公开 2004/0153804,该公开接着作为以下专利申请的部分连续申请,即,名称为“加工工厂中的智能过程模块和对象 (Smart Process Modules and Objects inProcess Plants)”、序列号为 10/278,469 的美国专利申请,该申请于 2002 年 10 月 22 日提交,并于 2004 年 4 月 22 日被公开为美国公开 2004/0075689,该申请的公开内容被全文合并在此作为引用参考。本申请还涉及名称为“在加工工厂配置系统中的模块类对象 (Module Class Objects in a Process Plant ConfigurationSystem)”、序列号为 10/368,151 的美国专利申请,该申请于 2003 年 2 月 18 日提交并且在 2004 年 10 月 7 日公开为美国公开 2004/0199925,该申请的公开内容被全文合并在此作为引用参考。本申请还涉及以下专利申请,这些申请作为国际 (PCT) 申请与本申请同一日期被提交,本申请将这些申请全文合并在此作为引用参考:“加工环境中的相关图形显示 (Associated Graphic Displays inProcess Environment)” (Atty. Docket No. 06005/41111);“用于过程控制系统的用户可配置警报和警报趋势 (User Configurable Alarms and Alarm Trendingfor Process Control Systems)” (Atty. Docket No. 06005/41112);“在加工工厂中过程模块和专家系统的集成 (Integration of Process Modules and ExpertSystems in Process Plant)” (Atty. Docket No. 06005/41113);“集成环境中具有用户化过程图像显示层的加工工厂用户界面系统 (A Process Plant User InterfaceSystem Having Customized Process Graphic Display Layers in an IntegratedEnvironment)” (06005/41114);“用于过程配置和控制环境的图像集成 (GraphicsIntegration into a Process Configuration and Control Environment)” (Atty. DocketNo. 06005/41116);“在过程环境中具有多视像的图像元件 (Graphic Elementwith Multiple Visualizations in a Process Environment)” (Atty. Docket No. 06005/41117);“用于在加工工厂中配置图像显示元素和过程模块的系统 (Systemfor Configuring Graphic Display Elements and Process Modules in Process Plants)” (Atty. Docket No. 06005/41118);“用于统一过程控制系统界面的图像显示配置框架 (Graphic Display Configuration Framework for Unified Process ControlSystem Interface)” (Atty. Docket No. 06005/41124);“在加工工厂用户界面中基于标记语言的动态过程图像 (Markup Language-Based, Dynamic ProcessGraphics in a Process Plant User Interface)” (Atty. Docket No. 06005/41127);“用于修改过程控制数据的方法和装置 (Methods and Apparatus for ModifyingProcess Control

Data)” (Atty. Docket Nos. 06005/591622 和 20040/59-11622) ;“用于访问过程控制数据的方法和设备 (Methods and Apparatus for Accessing Process Control Data)” (Atty. Docket Nos. 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 (Integrated Graphical Runtime Interface for Process Control System)” (Atty. Docket Nos. 06006/591628 和 20040/59-11628) ;以及“用于过程控制系统的面向服务的架构 (Service-Oriented Architecture for Process Control Systems)” (Atty. Docket Nos. 06005/591629 和 20040/59-11629) 。

### 技术领域

[0003] 本发明一般涉及加工厂,更具体地,涉及处于过程控制和仿真系统的系统级的图形显示编辑器和图形显示对象的集成和使用,进而使得普通图形显示元素能够在各种与工厂配置、控制、维护和仿真有关的活动中创建和使用。

### 背景技术

[0004] 分布式过程控制系统,例如应用在化学、石油或其他过程中的分布式过程系统,通常包括一个或多个过程控制器,这些控制器通过模拟、数字或模数结合总线,连接通讯连接到一个或多个现场设备。例如可以是阀门、阀门定位器、交换机和发射机(例如,温度、压力、水平面和流量传感器)的现场设备,位于过程环境内,并执行诸如开/关阀门、测量过程参数等功能。智能现场设备,例如根据诸如 FOUNDATION™ Fieldbus 协议的众所周知的 Fieldbus 协议的现场设备,也可以执行控制计算、报警功能以及其他在控制器中通常所进行的控制功能。过程控制器通常也位于工厂环境内,接收指示现场设备所做过程测量的信号,和/或关于现场设备的其它信息,并执行控制器应用,该应用例如运行不同的控制模块,这些模块基于所收到的信息做出过程控制决定、产生控制信号并配合诸如 HART、现场总线现场设备的现场设备中所执行的控制模块或者块。控制器中的控制模块沿通讯线路将控制信号发送到所述现场设备,进而控制过程操作。

[0005] 数据高速通道来自现场设备和控制器的信息通常可用于沿数据高速通道通向一个或多个其它硬件设备例如,操作员工作站、个人电脑、数据历史记录器、报告文件处理器、中央数据库等等,所述硬件设备通常位于远离较为恶劣的工厂环境的控制室或其他地点。这些硬件设备上所运行的应用程序,举例而言,允许操作员执行关于所述过程的功能,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,观察过程的当前状态,观察现场设备和控制器所产生的报警,仿真过程操作,以达到下列目的,即,培训人员或测试过程控制软件、维护和更新配置数据库,等等。

[0006] 举例而言,由爱默生过程管理公司 (Emerson Process Management) 所出售的 DeltaV™ 控制系统包含多种应用程序,这些应用程序保存在该控制系统中,并由加工厂内位于多个地方的不同设备来执行。位于一个或多个操作员工作站中的配置应用程序操作员工作站,允许用户创建或更改过程控制模块,并通过数据高速通道将这些过程控制模块下载到专用的分布式控制器。具体而言,这些控制模块由通讯互连的功能模块组成,这些功能模块是遵照面向对象程序协议的对象,所述对象基于当前输入执行控制方案内的功能,并且将输出提供给控制方案内的其他功能块。配置应用程序还允许设计人员创建或改变操作界

面,所述该操作界面由浏览应用程序使用,来为操作员显示数据并使得操作员能够在过程控制例行程序内更改设置,例如设置多个点。每个专用控制器以及某些情况下的现场设备,存储并执行控制器应用程序,该程序运行向其分配和下载的控制模块来实现实际的过程控制功能。可以在一个或多个操作员工作站上运行的浏览应用程序,可以通过数据高速通道接收来自控制器应用程序的数据,并将所接收数据显示给过程控制系统设计者、操作员或者使用用户界面的用户,还可以提供任意多个不同视图,例如操作员视图、工程师视图、技术员视图等等。数据历史应用程序通常存储于一个数据历史设备并由该设备来运行,所述数据历史设备收集和存储经数据高速通道提供的部分或全部数据,同时可以在与数据高速通道连接的电脑上运行配置数据库应用程序,来储存当前过程控制例行程序配置和与之相关联的数据。可选地,配置数据库也可以与配置应用程序放在同一个工作站中。

[0007] 鉴于用在过程控制环境中的控制和支持应用程序的数量和种类都有所提高,为了使用户能够有效地配置和使用这些应用程序,已经有多种不同的图形显示应用程序问世。例如,为了使配置工程师能够通过图形来创建控制程序,且这些程序能够下载到加工工厂中的控制设备,图形显示应用程序已经被用来支持控制配置应用程序。另外,图形显示应用程序也已经被用来达到以下目的,即使控制操作员能够观察加工工厂当前的运行情况或者加工工厂的各个区域,使维护人员能够观察加工工厂中的硬件设备的状态,启动加工工厂的仿真,等等。然而一直以来,这些图形显示应用程序是作为其所关联具体应用程序一部分被创建的,或者是为了支持其所关联具体应用程序而创建的,因此通常在有效性方面局限于创建所述图形显示应用程序所针对的具体过程功能。例如,如果并非不可能,也很难使用所创建的图形程序来支持控制或其他操作员进行维护、配置或者仿真。

[0008] 举一个具体的例子,一些过程控制配置应用程序目前包含模板对象库,例如功能块模板对象,并且在某些情况下,包含控制模块模板对象,这些对象用于为加工工厂创建控制策略。模板对象具有默认的属性、设置和与其关联的方法,而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,然后必须将所选模板对象的复制放置在配置屏幕上,以开发控制模块。在选择和放置模板对象于配置屏幕的过程中,工程师将这些对象的输入和输出相互连接,并更改这些对象的参数、名称、标签和其他属性,以在加工工厂中创建特定控制模型以特别使用。在创建一个或多个这样的控制模块后,工程师接着可以例示该控制模块,并将该控制模块下载到合适的一个或多个控制器以及现场设备,用于在加工工厂运行时间间执行。

[0009] 其后,工程师可以使用不同的图形显示创建应用程序,通过在显示创建应用程序中选择和建立创建对象,为加工工厂中的操作员、维护人员等创建一个或多个显示。这些显示通常是在一个或多个工作站的宽基础系统中实现的,并能够为操作者或维护人员提供预配置显示,所述预配置显示针对工厂中的控制系统或设备的运行状态。这些显示通常出现的形式包括:报警显示,所述报警显示接收并显示加工工厂中的控制器或设备所产生的警报;控制显示,指示加工工厂中的控制器或设备的运行状态;维护显示,指示加工工厂中的设备的功能状态,等等。然而,这些显示一般会预先配置,从而以公知的方式来显示从工厂中的控制模块或设备所接收到的信息或数据。在有些系统中,通过图形描绘来创建显示,该图形描绘代表物理或逻辑元素,并与所述物理或逻辑元素之间存在通讯联系,从而可以接收关于物理或逻辑元素的数据。显示屏上的图形可以响应某些事件而改变,所述事件,举例而言,是

所接收的数据,其例如描述一个半满的罐 (tank),通过流量传感器等等来描述所测量的流量。但是,用于配置、操作员控制、维护和仿真活动的图形显示一般都是使用不同的图形编辑器彼此分开创建的。另外,这些显示的有限的图形能力作为任何图形对象的某一部分难于执行也不能实现。

[0010] 鉴于此,类似于控制配置应用程序,显示创建应用程序可以具有模板图形显示项目,例如罐、阀门、传感器、滑动条之类的操作员控制按钮、开关电闸等等,这些项目可以任何期望配置而位于屏幕上来创建操作员显示、维护显示以及其他类似显示。当把个体图形项目放在屏幕上时,这些项目可以采用一种方式在屏幕上相互连接,该方式为用户提供加工厂的内部工作的一些信息或显示。但是,为了将图形显示做成动画,显示创建者必须通过指定在图形项目和加工厂内的相关数据源之间的通讯连接,来手工地将每个图形项目与加工厂内生成的数据结合在一起,所述加工厂内生成的数据例如是传感器所测量的数据或者阀门定位器指示的数据,等等。这种过程非常冗长耗时,而且可能会充满错误。

[0011] 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比较方便,因为它们可以被复制和用于创建多种不同的控制模块和图形显示,但是经常需要为加工厂内的许多不同设备创建大量同样的控制模块和图形显示。例如,许多中型到大型的加工厂都有同样的或相似的设备的实例,这些设备可以用同样的基本通用控制模块和显示来控制 and 观察。然而,为了创建这些大量的控制模块和显示,通用控制模块或显示模块被创建,然后,这种通用控制模块或显示模块针对其所用于的设备的每个不同部分进行复制。当然,在复制之后,必须在配置应用程序中手工更改每个新的控制或显示模块,从而指定其所连接的具体设备,并且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。

[0012] 遗憾的是,以上讨论的控制模块和显示项目并非在任何形式下都是模块化的。因此,在复制操作后,必须使用合适的配置应用程序来手工地并单独地更改所述控制模块和显示,从而指定与它们相关的工厂中的设备。对于一个工厂,如果它具有同一类型设备的很多复制(例如被复制的设备),这个过程将会冗长耗时,并充满着操作员带来的错误。而且,这些不同的控制模块和显示一旦被编程,这些控制模块和显示将互不察觉。因此,为了一旦创建就对控制模块进行改变,工程师或操作员必须针对不同复制的设备手工地对每个不同的控制模块进行同样的改变,这种操作同样也是冗长耗时的。同样的问题也存在于针对工厂中的不同批的复制设备所创建的图形视图。换句话说,一旦创建了具体控制模块或者具体的图形视图(单独建立或者从一个模板对象进行复制),并接着将所建控制模块或图形视图连接到工厂中的特定的一套设备,那么这种控制模块或图形视图作为系统内的独立实体或对象而存在,而不会被与其相同的或类似的其他控制模块或图形视图所自动察觉。结果,用于每个特定类型的控制模块和图形显示的变化,必须在这些模块和显示上单独进行。当图形视图是为工厂中的不同功能背景中的相同设备创建时,这个问题更加明显,所述不同功能背景,例如可以是控制视图、维护视图和仿真视图。在这种情况下,就要分开创建图形视图,以使这些视图相互没有联系或影响。

[0013] 因此,当图形视图已经被提供给或者联系到用于加工厂内实施的不同常规活动的不同应用程序时,这些图形显示和关联的图形显示编辑器通常被添加到功能级别的应用程序,而所述图形显示和编辑器正是用于支持这种应用程序的。结果,图形编辑器在一定程度上只是使用户能够建立支持某个应用程序所需特定功能的图形。以前的加工厂没有提供图

形显示编辑器,该编辑器可以用于或者支持在工厂配置和支持背景中所执行的各种或多种活动的图形需要。从而,举例来说,一个用于支持或者启动控制配置活动的图形显示编辑器只能使用户可以创建控制程序,而并不支持操作员或者维护显示的功能或需求。类似地,用于创建在工厂操作过程期间向控制操作员或者维护技师提供的操作员视图、维护视图等的图形显示编辑器,不支持与配置活动、仿真活动等关联的功能。由于在加工厂的独立功能级别被支持的图形显示需求,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别,由各种这类编辑器所创建的不同显示仅限于模拟和描述工厂内的相同部件,结果需要各种不同的工作人员在加工厂的图形显示方面加倍付出努力。这种加倍努力不仅表现在需要建立描述用于不同用途的相同过程元素的不同图形显示,而且表现在需要将用于不同显示应用程序的图形元素联系到与它们相关的加工厂内的实际硬件或软件元素。

[0014] 由于之后已经提供了针对各种加工厂活动的图形支持,并且作为实际活动的一部分而被执行,图形支持没有被集成在工厂环境中来允许在工厂中以各种不同功能级别进行创建和使用。这种图形的非集成化引发了针对功能与功能之间或者应用与应用之间不同的多种不同功能而实际建立的图形,从而导致用户一方的疑惑,虽然用户一种熟悉特定类型的图形显示,但是他可能在有时候需要观察与工厂中的不同操作或功能相关联的不同显示。同样地,正如上面所提到的,对工厂的各种不同功能级别的图形显示支持的提供,导致图形支持的加倍,无论是在创建显示方面,还是在将显示范围内的元素合理地连接到工厂中的实际硬件和软件元素方面。

[0015] 另外,差错检测和其他编程对于检测与不同控制器上的控制环路运行相关的情况、错误、警报等、以及个体器件内的问题都很有用。传统上来说,这种差错检测已经在加工厂的不同功能级别上执行,并已经被显示在针对那些不同功能活动而创建的图形显示上。因此,对过程控制系统编程,从而识别系统级别级别的情况和错误一直都很困难,这些情况和错误必须通过分析广泛分布在加工厂内的不同器件的数据才能检测到,而且,将这些类型的错误显示在操作员显示上更加困难,所述操作员显示还没有被创建来向操作员或维护人员指示或展示这种系统级别级别的情况信息。而且,采用针对显示中的不同元素的这些交替的信息资源或数据而将操作员显示中的对象做成动画也很难。

## 发明内容

[0016] 图形元素和图形显示被提供用于过程环境,以向一个或多个用户显示过程环境的有关信息,例如加工厂内的设备的当前状态。图形元素和显示包括一个或多个对象,其中的每一个包括:视像,该视像描述了过程环境中的实体;属性,其关联到所描述的过程实体;和例行程序,其结合所述视像和属性运行从而基于属性的变化引起视像的变化,或者基于用户通过视像的输入来因其过程环境中的变化。采用这种方式,图形对象可以被用于创建显示,该显示通过动画和其他视像操作图形化地描述了过程的变化状态或过程中的检测事件。这些相同的图形元素可以被用来使用户能够例如通过鼠标或键盘设备与用户界面屏幕上的视像互动,从而引起对象属性的变化,该变化然后可以向加工厂中的设备提供输入。

## 附图说明

[0017] 图 1 是位于加工厂内的一个分布式过程控制网络的结构示意图,所述加工厂包括



操作员工作站,该工作站执行显示例行程序和与加工厂内各种功能相关的其他应用程序,还包括一个工作站,其提供系统级图形支持,该系统级图形支持可用于为工厂内每个不同功能区域创建图形显示元素和图形显示;

[0018] 图 2 是描述加工厂控制、查看和仿真系统内的系统级图形支持的集成的逻辑结构示意图;

[0019] 图 3 是描述配置环境和运行时间环境的逻辑结构示意图,其中,图形元素和显示在所述配置环境中创建,并且可以在所述运行时间环境中被执行;

[0020] 图 4 是由图形编辑器创建的一个简化的显示屏幕,从而使用户可以创建或编辑一个形式为泵元素的图形元素;

[0021] 图 5 是进一步的显示屏幕,其由图形编辑器生成同时生成形式为反应堆的图形元素;

[0022] 图 6 是描述将各种显像与图形元素绑定或关联的方式的结构示意图;

[0023] 图 7 描述的是屏幕显示的一部分,该部分示出了与一个图形元素关联的第一属性视图;

[0024] 图 8 描述的是屏幕显示的一部分,该部分示出了与一个图形元素关联的第二属性视图;

[0025] 图 9 描述的是显示屏幕的动作/动画部分,该部分示出了与一个图形元素属性相关联的动作;

[0026] 图 10 所示结构示意图描述了将视像触发与图形元素的属性和显像集成的方式;

[0027] 图 11 描述可以用于提供或定义一种转换动画来将图形元素可视化的对话框;

[0028] 图 12 描述可以用于提供或定义一种属性动画(包括颜色动画)来将图形元素可视化的对话框;

[0029] 图 13 所示示意图描述了通过一个判决表将图形元素和运行时间环境进行连接的方式;

[0030] 图 14 描述的是与过程环境中多个不同数据源绑定的显示元素的结构示意图;

[0031] 图 15 所示为一个显示屏幕,该屏幕是由图形编辑器在从多个显示元素和连接器生成图形显示的过程中所生成的;

[0032] 图 16 所示为用于创建图形显示的显示器屏幕,该图形显示描述了与图形显示内的图形元素相关联的各种显像;

[0033] 图 17 所示为一个与图形显示相关联的显示器屏幕,所述图形显示由多个互相连接的图形元素和连接器组成;

[0034] 图 18 所示为一套图形显示屏幕,所述显示屏幕可以通过图 17 所示图形显示来访问,从而为图 17 所示显示器内的元素显示控制仪表或面板;

[0035] 图 19 所示图形显示的显示屏幕描述了一个由各种显示元素组成的石灰窑,并提供了来自加工厂内各种其他应用程序和数据源的数据;

[0036] 图 20A-20E 所示图形显示屏幕描述了具有相同外观和感觉的石灰窑的不同视图,包括操作员视图、工程视图、仿真视图和维护视图;

[0037] 图 21A 和 21B 所示为与一个控制例行程序相关联的显示屏幕,用于运行图 20A-20E 所示的石灰窑;

[0038] 图 22 所示结构示意图描述了不同级别的各种不同图形显示之间的联系,包括物理级别和功能级别;

[0039] 图 23 所示为第一配置屏幕,它描述的是一个与配置系统关联的图形配置分层;

[0040] 图 24 所示为描述一个图形配置分层的第二配置屏幕,该图形配置分层介绍了将图形元素和图形显示分配给或者集成到配置系统中其他元素中的方式;和

[0041] 图 25 描述的对话框可以为用户概述将一个或多个图形显示分配到加工厂配置系统当中的方式。

### 具体实施方式

[0042] 图 1 描述了示例加工厂 10,其中系统级别图形支持被提供给工厂 10 内的各种功能区域系统级别级别。具体而言,加工厂 10 包括分布式过程控制系统,该系统具有一个或多个控制器 12,每个控制器通过输入/输出(I/O)设备或卡 18 连接到一个或多个现场设备 14 和 16,输入/输出(I/O)设备或卡 18 例如可以是 Fieldbus 接口、Profibus 接口、HART 接口、标准 4-20ma 接口等等。控制器 12 还通过数据高速通道 24 被连接到一个或多个主工作站或操作员工作站 20-23,所述数据高速通道 24 例如可以是以太网线路。数据库 28 可以被连接到数据高速通道 24,并作为数据历史记录器进行操作以采集和存储与工厂 10 内的控制器和现场设备相关联的参数、状态和其它数据,和/或作为配置数据库来存储工厂 10 内过程控制系统的当前配置,该当前配置也被下载到以及储存在控制器 12 和现场设备 14 和 16 中。数据库 28 还可以存储用这里所述的方式创建的图形对象,从而在加工厂 10 中提供图形支持。虽然控制器 12、I/O 卡 18 和现场设备 14 及 16 通常位于有时候较为恶劣的工厂环境中并且遍及该工厂环境分布,操作员工作站 20-23 和数据库 28 通常位于控制室或其它不太恶劣的环境中,以便于控制器或维护人员接触。但是,在某些情况下,可以使用手持设备来实现这些功能,这些手持设备通常可以被携带到工厂中的不同位置。

[0043] 公知的是,每个控制器 12,例如由爱默生过程管理公司出售的 DeltaV™ 控制器,使用任意数量的独立运行的不同的控制模块或块 29,储存和执行一种能够实现控制策略的控制器应用程序。控制模块 29 中的每一个可以由通常所称的功能块组成,其中每个功能块都是一个总体控制例行程序的一部分或者子程序,并与其它功能块协同工作(通过通讯呼叫线路),从而在加工厂 10 内实现过程控制环路。众所周知,功能块,可以是依照面向对象编程协议的对象,其通常执行输入功能、控制功能和输出功能的其中之一,从而在加工厂 10 内实现某些物理功能,所述输入功能,例如可以与发射机、传感器或者其它过程参数测量器件相关;所述控制功能,例如可以与实现 PID、模糊逻辑等的控制相关;所述输出功能控制某些诸如阀门的设备的运行。当然,也存在混合的和其它类型的复杂功能块,例如模型预示控制器(Model Predictive Controllers, MPCs)、优化器等等。虽然 Fieldbus 协议和 DeltaV 系统协议使用的控制模块和功能块是以面向对象编程协议的来设计和实现的,但是所设计的控制模块可以使用任何希望的控制编程方案,例如包括顺序功能块、梯逻辑等等,而且所设计和实现的控制模块不限于使用功能块或使用任何其它特定编程技术。

[0044] 在图 1 所示的工厂 10 中,连接到控制器 12 的现场设备 14 和 16 可以是标准 4-20ma 设备,也可以是包括处理器和存储器的智能现场设备,例如 HART、Profibus 或者 FOUNDATION™ Fieldbus 现场设备,或者可以是任何其它所希望类型的设备。这些设备中的

有些设备,例如 Fieldbus 现场设备(在图 1 中标号为 16),可以存储和执行与控制器 12 中实行的控制策略相关联的模块或者诸如功能块的子模块。众所周知的是,功能块 30,也就是图 1 所示的布置在两个不同的 Fieldbus 现场设备 16 中的功能块,可以与控制器 12 中的控制模块 29 协同工作来实现过程控制。当然,现场设备 14 和 16 可以是任意类型的设备,例如传感器、阀门、发射机、定位器等等,并且 I/O 设备 18 可以是依照诸如 HART、Fieldbus、Profibus 等等的任何所希望的通讯或控制器协议的任意类型的 I/O 设备。

[0045] 在图 1 所示的加工厂 10 中,工作站 20-23 可以包括针对各种不同功能而使用的各种应用程序,所述各种不同功能由工厂 10 中的相同或不同工作人员执行。工作站 20-23 中的每一个都包括:存储器 31,其存储各种应用程序、程序、数据结构等;以及处理器 32,其可以被用来执行存储在存储器 31 中的任何应用程序。在图 1 所述的示例中,工作站 20 被指定为配置工作站,并包括一个或多个配置应用程序 33,配置应用程序 33 例如可以包括控制模块创建应用程序、操作员接口应用程序以及其它数据结构,任何被授权的配置工程师都可以访问这些程序和数据结构来创建和下载诸如控制模块 29 和 30 的控制例行程序或模块,将其下载到工厂 10 的多种控制器 12 和设备 16。图 1 所述工作站 21 是控制操作员视图工作站,并且包括许多显示应用程序 34,显示应用程序 34 可以在加工厂 10 运行时间间配备带有多个不同显示的控制操作员,从而使该操作员可以观察和控制加工厂 10 内或者工厂的不同部分正在发生的情况。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、调谐应用程序、报告生成应用程序或者其它任何能够用于帮助控制操作员实现控制功能的控制支持应用程序。类似地,工作站 22 被描述为维护观察工作站,并且包括多个维护应用程序 35,多个不同的维护人员都可以使用维护应用程序 35 来观察工厂 10 的维护需求,从而查看多个器件 12、14、16 等的运行或工作情况。当然,应用程序 35 还可以包括支持应用程序 35a,例如维护诊断应用程序、标度应用程序、波动分析应用程序,报告生成应用程序或者可以用于帮助维护人员在工厂 10 内实现维护功能的任何其它维护支持应用程序。另外,工作站 23 是仿真工作站,其包括多个可用于模拟工厂 10 或者工厂 10 内多个部分的运行情况的仿真应用程序 36,从而达到多种目的,包括培训目的、有助于工厂维护和控制的工厂仿真目的,等等。具体而言,工作站 20-23 中的每一个都包括一个显示屏 37 以及其它的标准外部设备,例如键盘、鼠标等。

[0046] 当然,虽然图 1 所述各种配置、控制、维护 and 仿真应用程序 33-36 位于不同的工作站中,其中的每个工作站专门针对所述的一种功能,但是可以理解的是,与这些或其它工厂功能相关的多种应用程序 33-36 可以位于或运行于工厂 10 内的相同或不同的工作站或电脑中,这取决于工厂 10 的需要和设置。因此,例如,一个或多个仿真应用程序 36 和控制应用程序 33 可以在相同的工作站 20-23 中被执行,而不同的独立仿真应用程序 36 或不同的独立控制应用程序 33 可以在工作站 20-23 中的不同工作站中被执行。

[0047] 在过去,对用于工厂 10 不同功能区域的各种不同的应用程序的开发被相对比较独立地执行。因此,对配置应用程序 33 的开发并未与仿真应用程序 36、维护应用程序 35 或操作员控制应用程序 34 集成。事实上,在很多实例中,工厂可以包括不同公司或软件提供商所开发的针对不同功能区域的应用程序,而且这些应用程序在工厂 10 内相互独立运行。这种对工厂 10 内各个功能区域相关的不同应用程序的独立开发和操作导致的结果是,工厂人员通常被要求在配置、操作员控制、维护和仿真功能级别上分别对工厂进行独立配置

和设置。特别地,相同或不同工厂人员通常必须使用不同的程序来以每种功能级别建立新的数据结构和图形显示。因此,引用图 1,实施配置、控制、维护和仿真功能的多个应用程序 33-36 中的每一个通常都包括或者使用不同的图形显示编辑器和数据库结构来帮助工厂人员实现这些配置、操作员控制、维护和仿真功能。在很多情况下,这些不同的图形显示编辑器和数据库被用来创建不同的图形显示,以描述或模拟工厂 10 内的相同部分或区域,或者描述或模拟工厂 10 内的相同硬件,进而帮助不同的工厂人员观察和了解在配置、操作员控制、维护和仿真活动背景下所进行的加工工厂内正在发生的情况。

[0048] 由于对应用程序 33-36 和工厂 10 内针对多种功能的相关显示的开发和实现通常都相互独立,有时由不同的人甚至不同的公司开发和实现,因此在加工工厂不同功能区域创建并应用的图形显示并未从提供图形视像的角度以任何一致或容易理解的方式被集成。因此,在工厂的各种不同功能级别上对图形显示的独立创建和实行导致的结果是,图形显示看上去因功能的不同而不同,从而导致对全部功能区域的图形显示没有一致的观感和感觉。另外,这种独立创建导致,在图形显示创建于工厂的相同部分或区域然而却用于不同功能使用时,需要付出加倍努力,而且要求如此创建的图形显示可以联系到工厂 10 内的各种设备,并且从工厂 10 内的各种设备接收数据,例如控制器 12 以及现场设备 14 和 16,所述各种设备的功能在工厂 10 的功能级别上被分开。这种情况接着要求对数据库结构进行复制,来为不同显示跟踪同样的硬件元素。因此,举例而言,在过去,第一应用程序(例如应用程序 35 中的一个)被用于创建维护显示,所述维护显示描述工厂 10 的一个部分以达到维护的目的,而第二应用程序(例如应用程序 34 中的一个)被用于创建控制操作员显示,所述操作员显示描述工厂 10 内的相同部分以达到控制的目的。由这些不同显示编辑器分开创建的显示在外观和感觉上可以很不一样,从而使得用户很难在不易混淆或者不需要接受针对每种显示的培训的情况下,维护显示和操作员显示之间来回切换。同样,在独立创建不同的应用程序 34 和 35 的显示时同样需要付出加倍努力,而且还要付出额外努力来创建数据库结构,以分别将维护显示和控制操作员显示联系或者连接到工厂 10 内相同的硬件元素上,以接收来自这些硬件元素的有时候相同或相似的数据。

[0049] 为了减少这些无效操作并在工厂 10 内提供应用更加广泛和更易理解的图形,一种图形支持层在加工工厂 10 的系统级别级别上被提出,以支持图形显示和工厂 10 内每个不同功能区域的数据结构需求,所述每个不同功能区域包括配置、操作员观察、维护观察、仿真和工厂 10 的其它功能区域。对这个系统级别支持的图释如图 2 所示,图 2 所述为工厂操作级别 40、工厂功能级别 42 和系统级别 44。从图 2 可以看到,工厂操作级别 40 包括控制器 12、现场设备 14 和 16 等用于执行控制程序和模块 29 和 30 的部件,以及在工厂运行时间期间运行在工厂 10 内的用于实现工厂操作的其它软件。工厂功能级别 42 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,不过其它的或不同的功能,例如工程功能和商业功能,也可以被提供。配置功能块 46 执行配置例行程序 33,其中配置例行程序 33 与工厂操作级别 40 内的元素接口或通讯,从而向那里提供控制策略或控制模块。控制功能块 47 包含控制观察和其它应用程序 34 和 34a,这些也通常与工厂操作级别 40 内的各种物理或逻辑元素直接接口或通信,以实现工厂 10 内的操作员初始更改、通过控制显示 34 为操作员提供信息、为控制应用程序 34a 获取数据等等。维护功能块 48 包括维护例行程序以及应用程序 35 和 35a,这些与工厂操作级别 40 内的各种物理或逻辑部件接口或通信,以实现

维护过程、采集维护数据、通过维护显示 35 为维护人员提供维护数据或信息、运行诊断应用程序 35a 等等。同样,仿真功能块 49 包括仿真例程序 36,仿真例程序 36 实现对工厂 10 的仿真,并可以在通讯连接到工厂操作级别 40 内的元素,以得到关于工厂 10 的数据。

[0050] 如图 2 所述,系统级别支持层 44 联系进入并支持工厂功能级别 42 内的功能块 46-49 中的每一个,从而能够例如建立和维护应用于各个功能区域 46-49 的普通数据库和显示结构,例如软件对象、图形元素和图形显示等。特别地,系统级别支持层 44 包括应用程序、数据库和图形支持元素,从而使得功能块 46-49 中所进行的画面活动被集成,或者使用普通数据库结构和系统级别支持层 44 内所创建的图形元素来开发功能块 46-49 中所进行的画面活动。为了提供这种系统级别支持,系统级别支持层 44 可以包括图形编辑器 50 和图形对象数据库 52。图形编辑器 50 可用于创建图形元素 54 和图形显示 56,而图形对象数据库 52 则将元素 54 和显示器 56 存储在编辑器 52 和块 46-49 中的各种应用程序所访问的存储器中。数据库 52 也可以储存其它对象 58 和数据结构,所述数据结构将图形元素 54 连接到工厂操作级别 40 内的个体硬件和软件元素。另外,数据库 52 可以储存图形元素或显示模版或能够用于创建其它图形元素和显示的单元。从图 2 可以看到,图形显示元素 54、显示 56 和其它数据库结构 58 可以被用于任何一个和所有的功能块 46-49 中,来创建和使用与这些功能块相关的图形。

[0051] 一般来说,系统级别支持块 44 提供了一种方式,该方式将图 1 所示加工厂 10 内所使用图形集成在所有功能区域 46-49,从而降低或消除为不同功能环境中的相同工厂设备重复创建不同图形元素的需要,并使得每个功能区域 46-49 的用户能够容易地访问与设备相关的数据,其中所述数据被显示在与那些功能区域相关联的图形视图中。可以理解的是,系统级别支持层 44 可以被用于为每个功能区域 46-49 中的多个应用程序提供图形和数据库支持,也可以为不同功能区域 46-49 中的不同应用程序提供图形和数据库支持,等等。

[0052] 再次参见图 1,可以使用一个附加工作站或用户接口 60 来实现系统级别支持块 44,所述附加工作站或用户接口 60 可被连接到其它工作站 20-23 中的每一个。工作站 60 一般可以储存图形编辑器 50 和数据库 52,如果需要的话,也可以储存其它元素 54、56 和 58。另外,工作站 60 可以通过数据总线 24 通讯连接到工作站 20-23 以进行通信,也可以通过独立的有线或无线通讯连接(如图 1 中的虚线所示)或者通过其它任何所希望的方式。在图 1 所述配置中,工作站 60 储存并执行显示编辑器 50 以使得用户能够创建图形元素,并对所述图形元素分组成一个或多个图形显示,具有以上两种功能的模块在此都被称为显示模块。然后这些显示模块可以被存储在数据库 52 中,供图 2 所示的各个功能块 46-49 访问和使用,并可以在多种工作站 20-23 中被执行。虽然,出于描述的原因,系统级别块 44 和功能级别模块 46-49 的功能在图 1 所示不同或单独的工作站 20-23 和 60 中实现,但是可以理解的是,与这些不同模块中任何一个相关联的任意或所有应用程序可以在同样或不同的工作站上执行,或是在与加工厂 10 内的或与加工厂 10 关联的其它电脑中执行。因此,图形编辑器 50 可以保存在或执行于其它工作站 20-23 中的任何一个当中,或者保存在或执行于与工厂 10 关联的其它电脑上,而无需是在一个单机或单独电脑上执行。

[0053] 如上所述,图 2 所示系统级别支持层 44 实现了系统级别显示和数据库对象,所述系统级别显示和数据库对象可用于多种功能环境中,并且提供更高级别显示能力。一般来说,在图 2 所示系统级别支持层 44 中建立的显示对象可以分为图形元素和图形显示。图形

元素一般是与工厂内的特定物理实体关联的图元原始的或低级别显示对象,例如阀门、传感器、泵、控制器之类的硬件设备。图形显示一般由一套互相连接的图形元素组成,用于呈现和模拟工厂内较为复杂的硬件,例如单元、区域等等,而且所述图形显示包括不同硬件单元之间的互联。另外,图形显示可以包括图形、图表和工厂所提供的自其它应用程序的其它数据,例如运行在工作站 20-23 和 60 的诊断和商业应用程序,等等。

[0054] 图 3 通常描述两种环境中的图形元素和图形显示的开发和使用,这些元素和显示可能存在于所述两个环境中,具体而言,所述两个环境指的是配置环境 70 和运行时间环境 72。一般来说,使用诸如显示编辑器 50 可以在配置环境 70 中创建形式为图形元素 74(所示出的单独的元素对象 74a、74b 等等)和图形显示 76(所示出的单独的显示对象 76a、76b 等等)的显示对象。创建之后,对象 74 和 76 可以存储在数据库 52 中。对象 74 和 76 可以创建为类对象,在此称为显示模块类对象,其定义了一种通用对象,该对象并未绑定或联系于加工厂 10 内的特定硬件或逻辑元素。然而,类对象可用于创建与类对象具有相同基本属性的运行时间图形对象,但是所述运行时间图形对象限制于或绑定在加工厂 10 内的特定硬件上。然而,一般来说,类对象保持限制于从其中实例化的子对象,这样类对象的变化可以自动传播给子对象,即使这些子对象在运行时间环境中被实例化。

[0055] 如图 3 所示,图形元素对象 74 中的每一个都包括多个元素,这些元素使得图形元素在很多不同的环境中都适用。具体地,每个图形元素 74 都包括一个或多个视像 (visualization) 77、任意数量的参数或属性 78、任意数量的动作或动画 79 和绑定 80,其中所述任意数量的动作或动画 79 通过使用脚本或触发来实现。一般来说,当图形元素 74 在运行时间环境 72 中被实现时,每个视像 77 定义了显示屏幕上将实际显示的视像属性和元素。具体而言,虽然视像可以呈现其它实体,但视像定义物理或逻辑设备的呈现或者一组设备的表现。可以在运行时间环境 72 中使用任何所希望的描述或者定义了实体的图形描述细节的编程范例来实现视像 77。在一个实施例中,视像 77 可以用 PGXML 或者 Avalon 来实现,所述著名的 PGXML 或者 Avalon 控制是由微软公司 (Microsoft) 提供的众所周知的控制,并且由于它们是基于对象的,因此可以提供一种方式能够很容易在标准 Windows 类型显示上实现视像并且使所述视像可以在显示环境之间移动。这种特性将在下文中进一步详细讨论,并且在另一个名称为“加工厂用户接口中基于标记语言的动态过程图形 (Makeup Language-Based, Dynamic Process Graphics in a Process Plant User Interface)” (Atty. Docket No. 06005/41127) 的共同提交的申请中也对此进行了解释,该申请在此通过引用并入。

[0056] 一般来说参数和属性 78 定义视像所描述的实体相关联的变量或其它属性,例如静态的或可变的固有属性,而且这些属性可以由元素 74 的创建者来定义。同样,动作和动画 79 定义例行程序或程序(其可以作为进行属性转换的脚本来执行,并且基于属性值等等来检测过程实体的情况),并且定义动画例行程序,所述动画例行程序可以包括任何改变待执行或者当视像 77 被描述在显示屏幕上时使用视像 77 的图形视像或行为的例行程序,或者使用户能够使用视像 77 或者与视像 77 互动从而导致诸如向过程的输入变化的过程中的变化的例行程序。这些动作和动画为视像 77 提供了更多有趣的、可理解的或有用的图形属性,并允许用户与视像 77 进行互动。在一种情况中,这些动作或动画可能以颜色变化、各种视像部分的尺寸变化(例如长和宽、行尺寸、字体等)、颜色填充、动画(诸如颜色变化、旋

转、尺寸变化、放缩和滞后)等形式出现。这些动作和动画为图形元素 74 提供了图形属性以及用户互动属性。绑定 80 可以是静态或固定绑定,也可以是使用别名的绑定,绑定 80 定义了参数和属性 78 被绑定到运行时间环境 72 中的数据、标签和其它实体的方式,此时图形元素 74 作为运行时间环境 72 显示的一部分被实现。基本上,针对每个图形元素 74 的绑定 80 建立了图形元素 74 被绑定到工厂环境中的其他地方所定义的一个或多个实体或数据元素的方式,从而也在实际的运行时间环境 72 和图形元素 74 之间定义了一个接口。

[0057] 如图 3 所示,每个图形显示对象 76 包含多个部分,例如一个或多个图形元素 81、连接器元素 82、动作和动画 83、属性 84 和绑定 85 的引用或复制。一般来说,图形显示 76 可以是一种描述各种图形元素 81 之间的互动的显示,各种图形元素 81 可能在视像上由连接器元素 82 连接在一起,所述连接器元素 82 表现为管道、线、传送带等等。所述连接器对象在美国公开 No. 2004/0153804 中有描述。图 3 中的虚线描述了图形元素 74 的图形显示对象 76a 的引用。可以理解的是,引用图形元素 74 的图形显示 76 包括图形元素 74 的所有属性、参数、动作和动画等。类似于图形元素 74,每个图形显示 76 可以包含一个或多个附加的与之关联的动作或动画,例如在显示器、用户接口交互、数据过程上的动画。同样,每个图形显示 76 都可以包含任意数目的与显示相关的属性,而且具体地,这些属性定义了显示中所描述的单元属性、区域属性或其它元素组合的属性。当然,绑定 85 定义图形显示 76 被绑定到工厂环境中的其他地方定义的一个或多个实体或数据元素的发那个时,因此也定义了运行时间环境 72 和图形显示 76 之间的一个接口。

[0058] 一旦创建了图形元素 74 和图形显示 76,它们就可以绑定到运行时间环境 72 并在运行时间环境 72 中执行,例如在图 1 所示工作站 20-23 中的任何一个上。具体地,在作为类对象创建了图形元素 74 和图形显示 76、并将其储存在数据库 52 中之后,所述元素或显示可以被实例化为一个实际的运行时间对象,并可以在运行时间环境 72 中执行。如模块 86 所示,示例过程在对象 74 和 76 中定义的绑定进行填充,该过程可以通过使用加工厂内或者过程控制系统内的一个或多个具有合理变量名、标签、别名等的判决表来实现,从而在加工厂内的实际实体和在工厂 10 内显示设备上运行的图形对象之间提供一个特定的连接。作为绑定的一部分,对象 74 和 76 连接到加工厂内的由判决表定义的数据源,从而能够访问该工厂以在逻辑和通信方面连接到加工厂 10。

[0059] 如模块 87 所示,显示元素 74 或图形显示 76 可以运行于运行时间环境 72 中的多种不同功能,或者作为其一部分,包括配置显示、控制操作员显示、维护显示和仿真显示,但命名的只有少数。另外,显示对象 74 和 76 也可以用于实现系统级别功能,举例而言使用来自图 2 所述各种功能级别的数据的功能,包括例如图元预测控制或图元预测维护功能、系统级别级别错误检测、诊断等。事实上,在配置环境中创建并存储于数据库 52 的显示 76,可被用于多种不同活动。而且,显示对象 74 和 76 可以在任何所希望的显示或电脑硬件上执行,例如工作站 90、便携式电脑 91、诸如个人数字助理 (PDA) 的手持设备 92、电话设备等,或者在任何其它特定显示 93 上执行,例如具有多个监视器的大屏幕显示,等。如果希望,图形显示 76 可分层包含一个或多个视图,例如配置视图、操作员视图、维护视图和仿真视图。可选地,可以配置独立的图形显示 76 从而使用相同或相似的图形元素 81 来提供这些独立视图,从而为各种功能的显示提供一致的外观和感觉。

[0060] 如模块 95 所示,为了转移到运行时间环境 72,图形元素 74 或图形显示 76 可以被

复制或实例化,然后装载到运行时间机器。一般来说,只有当显示对象 74 或 76 被调用或在运行时间机器上实际运行时,才希望显示对象 74 或 76 被绑定到运行时间环境 72,这个过程在此称之为运行时间绑定。也就是说,当显示对象在实际运行或者在运行时间机器上被执行时,每个实例化对象的判决表才会被填写或者被绑定到运行时间环境。这个过程可以保证显示对象,包括显示对象的视像、控制、脚本等,只有当对象的视像正在被实际呈现在显示屏幕上时,才执行或进而使用过程能力。因此,当对象正实际运行在运行时间电脑上时,显示对象才被优选地绑定到运行时间环境 72,这也就意味着显示对象 74 和 76 可以被间歇地连接到运行时间环境 72,所采用的连接方式通过用户的行为而观察视像的活动所定义。具体地,当这些对象被要求查看时,可以被绑定到运行时间环境 72,当这些对象未被用户查看时,例如当一个用户最小化或关闭这些对象正在提供一个视像的屏幕时,这些对象可以被解除绑定或者释放。

[0061] 因此,显示对象 74 和 76 是可以在单机环境下创建的对象,即,配置环境 70,但也可以绑定或连接到加工厂环境内定义的其他对象或数据结构,或者加工厂环境内运行的任何应用程序,例如包括定义在任何控制、仿真、维护或配置环境中的对象、数据结构、应用程序等。进一步地,显示对象 74 和 76 一旦被创建,就可以直接通过判决表定义的直接查询、变量或标签,直接绑定到物理或逻辑过程实体,或者通过使用别名、变量和参数间接绑定到物理或逻辑过程实体,以上过程或者在显示对象 74 或 76 在运行时间环境 72 内被下载或实例化时,或者在某些情况下,当显示对象 74 和 76 在运行时间环境 72 中实际运行时得到解决。

[0062] 图 3 所示的显示编辑器 50 使得显示对象 74 和 76 可以在各种细节级别上被创建,从而增强了显示对象 74 和 76 的使用舒适性和多功能性。例如,图形元素 74 可以被首先创建,从而定义更多基本的物理和逻辑实体的属性和操作,然后通过将一个或多个图形元素 74 互相连接在一起,来创建图形显示 76,从而创建更高级别或更加复杂的显示,所述更加复杂的显示描述更加复杂的物理或逻辑实体,或者描述分组的物理或逻辑。当然,图形元素 74 或图形显示 76 都可以在各种类型中被储存和访问,从而使得更高级别显示对象的创建对于用户来说更加简单。

[0063] 图 4 所述为示例屏幕显示 100,其可以由显示编辑器 50 所创建。在为泵而创建图形元素的过程中用来显示的屏幕 100 包括主要编辑部分 102、元素层部分 104、属性定义部分 106 和视像部分 108。主要编辑部分 102 为用户或设计者提供工作空间,从而为图形元素定义或创建视像,并进而定义图形元素的视像属性,例如在此的视像 109 所述的泵。一般来说,元素层部分 104 使用分层视图或树结构来提供与主要编辑部分 102 内的视像 109 相关联的部件。在图 4 的实例中,层部分 104 示出,在主要编辑部分 102 内定义的视像 109 包括一个圆(命名为 Circle1)以及两个长方形(命名为 Rect1 和 Rect2)的图元或子元素。虽然图 4 没有示出,但层部分 104 可能还包括为视像 109 定义的动画、动作和诸如脚本、视像触发等的其他显示特色的指示。

[0064] 属性定义部分 106 描述了当前为与编辑部分 102 中示出的视像化 109 相关联的图形元素所定义的所有的属性,包括固有属性。图 4 所述示例屏幕 100 描述了两个属性,其中一个为开启(Is on)属性,该属性定义与视像化 109 相关联的泵是开启还是关闭,另外一个为速度(Speed)属性,该属性定义了与视像化 109 相关联的泵的速度。用户或设计者可以通



过在属性定义部分 106 内定义名称、其他变量的种类和绑定、属性等,为图形元素增加其他属性和参数,从而定义图形元素的其他方面。图 4 中的部分 106 所示两个属性作为布尔型 (Boolean) 变量和浮点变量而列出。然而,其他类型的变量也可以替代或加入。因此,例如,部分 106 中所定义的属性可以是数列、表格、枚举表或任何其他类型的变量或数据结构。

[0065] 如果需要,在屏幕 100 内定义的图形元素可以具有多个关联的视像。这些不同的视像可以在视像部分 108 中进行描述,也可以被单独选择而被放于主要编辑部分 102。例如,图 4 中的视像部分 108 所示为两个视像 110A 和 110B,但是任何其它数量和类型的视像可被定义用于所创建的图形元素。在创建图形元素的过程中,可以将每个视像都放于主要编辑部分 102,例如通过右键单击或者双击部分 108 中的视像的指示,将视像指示拖放到编辑部分 102,等。一旦将视像放置于主要编辑部分 102 中,可以编辑该视像从而定义或重定义其显示属性。一般来说,其中一个视像将作为默认视像而针对正在创建的图形元素而设置或定义。可以通过某种方式来表示这个默认设置,例如对默认视像施加高光、如图 4 对视像 110A 所示在视像部分 108 中围绕以虚线,或者通过其他方式。

[0066] 一般来说,由元素编辑器 50 所创建的视像可以包含一个或多个具有各种形状的合成,这些部件以一种特定的方式位于或组放在一起。因此,这些视像被称为形状合成。例如,形状合成可以包括圆、线、点、多边形、正方形、长方形、三角形或其他简单图形形状。在这样的定义方式下,独立的动作或动画可以被用于或者关联于组成形状合成的每个不同形状形状。当然,形状合成可能包含更加精巧的艺术再现元素。为了定义或建立形状合成,用户或设计者可以往主要编辑部分 102 中添加任意数目的图元图形元素,并以希望的方式对这些图元分组。一旦创建,形状合成就为实际对象定义了视像,所述实际对象可以在运行时间内作为 XAML 对象而实现,当图形元素被用在运行时间环境中时,该对象将被作为视像显示在屏幕或显示上。可以在层部分 104 中的合成层中来描述组成形状合成的形状或图元。

[0067] 为单个图形元素创建多个视像可以使不同的视像用在不同的环境中或用作不同的应用,从而在显示一个普通元素的不同视像时,可以从同一个图形元素(例如从同一个图形类对象)来创建工厂内为不同目的而建立的显示。例如,不同的视像 110A、110B 等可以用在不同的功能环境中,从而当图形元素用作控制操作员显示的一部分时可以使用第一视像 110A,当图形元素用作维护显示的一部分时可以使用第二视像 110B,当泵元素用于仿真显示时可以使用第三视像(未示出)。另一方面,可以使用不同的视像用于不同类型显示设备的显示目的。例如,图形元素的第一视像可能适用于典型计算机或工作站,另一种视像可能更适用于手持设备,例如 PDA 或电话设备,再有一种视像可以被用于大型屏幕显示或多屏幕监视器。可以理解的是,可以基于用于显示视像的显示器大小,来修改显示元素的不同视像,从而使第一视像可适于创建较大型屏幕,例如,例如具体的计算机屏幕,并且第二视像可以被创建以更加适用于微小屏幕,例如手持设备的显示屏幕,像 PDA 或无线手机设备。因此,同一图形元素的不同视像可以用于不同的运行时间设备中。

[0068] 另一方面,可以在图形元素开发中的不同时间为图形元素添加不同的视像。例如,一个可能并不擅长绘制或创作令人满意的视像的配置工程师,在起初创建图形元素的时候可以做出第一基本视像(例如长条图类的图画),并在此时可以将该视像作为图形元素的一部分加以保存。在之后的日期或时间中,一个熟练的绘图员或艺术家可能创作出更加精美的、在艺术上令人满意的视像,并可以将该第二视像添加到图形元素中作为可选的视像。

这时,如果需要,可以将第二视像设置为默认视像,并可以将该第二视像传播到所有已经被主台或类图形元素例示过的运行时间图形显示,从而第二视像可以在运行时间显示中出现或可以用于运行时间显示。

[0069] 在另一个实例中,可以为同一图形元素提供不同的视像来支持不同的主题、绘图标准、规范或显示风格。众所周知,不同产业通常使用不同的图形规范或绘图标准来描述泵、阀门、传感器和其他加工厂实体。因此,石油和天然气行业使用的绘图标准与医药行业使用的绘图标准是不一样的。使用元素编辑器 50 可以为每个图形元素提供不同的视像以支持多个绘图标准或规范,例如石油和天然气规范以及医药规范。采用这种方式,同一图形元素可能被用在所创建的图形显示中来支持不同显示标准或显示规范或显示主题,例如不同行业可以接收的显示或对不同行业有用的显示。另外,不同的绘图风格,例如艺术风格,可以被用在不同的视像中,从而使得用户可以创建各种艺术类型的显示。当然,这些只是有益于为同一图形元素提供多个视像的几个实例,所述多个视像还具有其他用途。

[0070] 如果希望,图形元素具有与其关联的一个或多个绘图行为。具体地,当视像显示在屏幕上时,设计者或创作者可以为图形元素的每个视像定义动画,例如旋转、线性变换、背景变化、颜色变化、缩放、颜色梯度特效、不透明度特效、字体特点特效,以及视频和视频特色,例如开始/停止特色、二维或三维变化等等。为添加所述动态行为,用户可以选择图形元素并选择特效(或者叫做特效例行程序)添加到视像中。这时,用户可以输入与所选特效有关的配置信息,或者可以使用脚本来定义需要的行为,可以通过例如对话框、表达编辑器等来输入所述脚本。一旦定义了脚本,所述脚本可以在层部分 104 的层中显示。基本上,脚本是一个程序或例行程序,当显示元素的视像在某个运行时间活动中被查看或显示在屏幕上时,该程序将作为显示元素一部分来运行或执行。虽然没有必要,但是这些行为或脚本可以连接到并操作一个或多个为图形元素定义的属性或参数。例如,脚本可以跟一个图形元素关联到一起,从而可以基于图形元素的属性部分 106 所定义的某个属性值来改变所述图形元素的视图颜色。例如,当开启属性(该属性绑定到运行时间环境)为“真”(True)时,即,泵被开启时,泵视像 109 的颜色可以由黄色变为绿色。作为另外一个实例,可以为泵元素定义一个脚本,该脚本对比泵的速度属性值和一个已知的点,如果速度属性的值大于特定水平,该脚本就在视像 109 内引起某种图形动画。所述动画可以包括例如将泵的颜色变为红色、旋转泵、在旋转的泵内显示一个马达、造成泵视像的跳动或振动,等等。

[0071] 为视像定义的脚本,例如为视像提供管理或其他行为,可以被设计来运行多个形状合成中的形状或图元,所述形状或基于用于组成视图,或者运行在形状合成内的多个形状上。由于脚本操作或使用固有属性,所述固有属性被绑定到运行时间环境内的实际物理元素上,这些脚本使得视像能够基于实际工厂操作来变换,或者基于来自加工厂其他区域的数据,该数据能够反映与图形元素关联的实际实体的属性。

[0072] 可以理解的是,不同类型的图形元素可以具有不同的图形,所述不同图形适合用于这些图形元素。因此,可以提供给图形元素的行为并不局限于在此所提供的例子。例如,用于旋转设备的图形元素包含的脚本提供振动图形、运动、颜色变化等,而针对诸如传感器的设备的图形元素包含的脚本能够描述高或低的限制条件、校准条件需求等。当然,可以为图形元素使用或定义任何属性,而且这些属性基本上是基于正在显示的实体的类型。还可以理解的是,可以为每个不同的视像提供多种脚本,从而为不同的视像提供多种行为。另一

方面,有些脚本可以用来为每个与图形元素关联的视像提供行为。

[0073] 另外,可以将脚本或其它例行程序与图形元素关联在一起,从而可以基于一个或多个针对图形元素的参数来检测关于所关联物理实体的特定情况。这些情况可以包括检测实体状态,包括物理状态、如通信状态的某个状态、设备状态、数值状态等等。可以使用一个与脚本生成的状态或数值相关的动画或其他动作、行为,将所有的被检测的情况或状态反映在图形上。例如,将脚本提供给泵图形元素,该图形元素基于针对该元素的一个或多个参数来检测所关联泵的过热情况,也就是说,基于图 4 所示属性部分 106 所定义或显露的一个或多个参数来检测。作为另一个实例,脚本可以被提供给图形元素,以检测过度振动或泵的任何其它情况。如果检测到有过热或过度振动这种情况,将针对该情况进行动画或其它与图形元素关联的动作,从而在图形元素的视像内或视像上提供针对该情况的图形指示。

[0074] 图 5 描述了另外一个示例屏幕显示 112,其中示例屏幕显示 112 可以由显示编辑器 50 来生成。与图 4 所示的屏幕 100 类似,屏幕显示 112 包括主要编辑部分 114、平板视图 116、层视图 118 和属性视图 120。电抗器元素 122 的第一个视像由主要编辑部分 114 来描述。如分层视图 118 所示,该元素的标题为 Reactor1,并且该元素包括三个视像,分别名为 Visual1(默认视像)、Visual2 和 Visual3。在分层视图 118 中的顶部 Visual1 下,第一视像由包括长方形元素和椭圆形元素的画布背景组成。当前定义的属性,在本例中为视像的名称、高度和宽度,列在属性视图 120 中。当一个视像被选择在分层视图 118 中时,与该视像关联的任何子图元或元素被显示在编辑视图 114 中,而当前所选元素的属性显示在属性视图 120 中。

[0075] 在屏幕 112 中,平板视图 116 包括多个基本元素,所述基本元素可以用来建立视像。例如,平板视图 116 包括一套诸如按钮、文本框、滑动条、旋钮等等的基本用户接口(User Interface, UI)元素、一套基本面板以及一套基本形状。所定义的面板可以包括画布面板,该面板定义的区域中,用户可以通过与画布区域有关的坐标来准确地定位元素,还可以包括一个对接面板,该面板定义的区域中,用户可以互相水平或者垂直地排列元素,还包括一个流面板,该面板可以用于在流面板区域内沿着一个流指示方向破坏、包装以及排列该面板的内容。例如,流方向可以是上下左右的任意组合,例如从左到右并且从上到下,或者从右到左、从上到下,等等。而且,平板视图 116 中的基本形状可以包括美国仪表协会(Instrument Society of America, ISA)符号、发射器符号、阀门符号、PI&D 示意图符号或者其它控制符号,等等,或者是任何其它希望的形状,所有这些可以被用于建立图形元素。

[0076] 当然,其他基本元素,例如基本控制元素、设备等,也可以作为图元提供给平板视图 116,以用于为正在定义的图形元素创建视像。所述平板视图 116 也可以提供列表的用户定义的类型和元素,从而使用户可以将任何其它有用的形状从平板视图 116 拖拽到编辑视图 114。如果需要的话,可以限制只有建立这些类型的用户才可以访问这些用户定义的类型,而且这些类型和元素可以存储在一个与某个特定用户关联的用户优选文件中。然而,内置的类型和平板项目可以存储在数据库中,并可以通用地供所有用户使用。在任何情况下,平板视图 116 都可以用于显示或提供向图元元素库的访问,这些图元元素可用于组建图形元素,而且,这些库可以根据需要被锁定、按版本分类、限制特定用户使用等等。

[0077] 如上所述,图形元素的任何视像都可以具有与之关联的动画和/或动作,所述动画或动作可以显示在屏幕 112 的动作/动画视图 123 中。当视像包括动画或动作时,这些

动画或动作也可以采用诸如星号的特定符号被指示于分层中。当在分层视图 118 中进行选择的时候,任何由视像或视像的子元素定义的动作或动画都将被显示在动作 / 动画视图 123 中。通过在视图 123 中定义这样的动作或动画,或者通过将这些动作或动画添加到分层视图 118 中,动作或动画可以进行创建或分配。当用户希望创建或编辑动作或动画时,编辑器 50 可以提供对话框或编辑框,从而允许这个特点能够被全面指定或定义。当然,通过对脚本、视像触发器或其它程序可以定义动作或动画。

[0078] 在使用屏幕 112 期间,用户或设计者可以通过拖动和释放或者选择平板视图 116 内的不同项目,来创建元素的视像,并通过在编辑视图 114 中布置这些项目来创建所希望的视像。可以使用一个或多个工具栏 124 来提供任何标准编辑功能,例如添加新的视像或动画、删除、移动、编辑动画、例如从后向前地布置图元或元素、为连接元素提供显示或制成正在建立的视像、对不同图元进行分组从而使不同图元之间保持某种相对位置、添加线条或文本之类的静态元素。

[0079] 如图 6 所示,图形元素 130 可以具有多个视像 132 或与之关联的视觉表现,可以理解的是,不同的视像 132 可以在不同时间段内或者在图形元素 130 的不同显示中得到使用。如上文所解释的,任何视像 132 都可以由任意数量的图元 134 以及动画和动作 136 组成。而且,图形元素 130 可以包括任意数量的属性或参数 138,所述属性或参数可以被链接到动作和动画 136 或者被动作和动画 136 所使用,以对视像 132 执行变化,并操纵与视像 132 关联的视像触发器。而且,组成视像 132 的视像 132 或个体图元可以包含为预定事件所定义的动作,例如鼠标覆盖事件、鼠标点击事件,等。这些动作(也称作例行程序)使得事件句柄能够被设置或定义,从而进一步将图形元素 130 的行为用户化,并允许用户与视像 132 进行互动,例如,从而引起运行时间环境内的变化。具体地,用户可以通过输入数值或其他数字或信息,与视像 132 进行互动,将某个元素(例如滑动条)移动到视像上,或者通过采取其他动作来施加变化,例如视像内的一个属性。所述视像属性变化可以通过脚本或直接链接到过程输入,例如过程运行时间变量,从而使得该变量进行变化。具体地,视像属性可以被连接到为图形元素定义的属性,所述图形元素接着被绑定到过程输入。在这种方式下,用户可以通过动作或动画例行程序与视像进行互动,从而导致过程或其他运行时间环境发生变化或为过程或其他运行时间环境提供输入,所述其他运行时间环境,例如仿真环境。

[0080] 如果需要,可以通过用户指定或程序指定,将视像 132 连接到图形元素 130。具体地,图形元素 130 可以提出列举属性,从而基于例如图形元素所表示元素的制造或诸如与图形元素关联的设备的状态与图形元素 130 关联的其他参数,允许视像通过编程而变化元素元素元素,例如元素。

[0081] 如上所述,图 5 所示的属性视图 120 提供或展示了为分层视图 118 中所选项目以及因此为主要编辑视图 114 所描述项目所定义的属性和事件。用户可以使用例如工具栏按钮,在属性视图 120 内的属性、固有属性和事件之间进行切换。图 7 示出了属性视图 120A,在该视图中显示了固有属性,在这种情况下包括开启和速度属性。在这种情况下,属性视图 120A 示出了这些变量的数据类型和针对这些变量的任何默认设置。如上所述,用户可以添加、删除或编辑这种属性列表,来定义任何所需数据类型的属性,包括列举、表结构等。如果希望,属性视图 120 还可以示出为这些属性所定义的任何运行时间绑定,而且这些运行时间绑定可以是固定变量或标签,也可以是使用别名的标签,所述使用别名的标签在运行时

间或图形元素的下载过程中被填写到运行时间机器。

[0082] 为图形元素定义的属性值可以用做针对动画、动作等的触发条件，而且这些行为可以由与图形元素相关的一个或多个脚本来定义。例如，如果一个马达具有的 On 属性是真，运行中的马达的动画可以在图形显示上触发。而且，图元属性，例如长方形形式图元的长方形填充属性，可以绑定到图形元素属性，这样图形元素属性的变化对图元属性起作用。同样，可以将视像的变化链接到一种属性，从而视像的变化可以倒置属性值的变化。

[0083] 如果需要，编辑器 50 可以允许用户指定一个或多个转换器功能，从而在图形元素属性和动画或动作之间提供更为需要的绑定。例如，用户可能希望将 TankLevel（定义了罐的流体水平面）的图形元素属性绑定到图元属性长方形填充，这样，由于所述图元被定义为视像的一部分，就可以采用填充颜色来图形化地描述罐水平图元图元。然而，在这种情况下，用户可以定义转换器功能，该功能可以将属性（TankLevel）转换为一种列举组或条件，这样，如果罐水平面介于第一和第二水平面之间，长方形填充被设置为绿色，如果罐水平面介于第二和第三水平面之间，长方形填充被设置为黄色，如果罐水平面高于第三水平面，长方形填充被设置为红色。这种转换器功能可以定义为脚本或任何其他程序，所述其他程序采用图形元素来运行，而且可以用来引起任何希望的属性变换，例如将属性值变成长度、字体名称、局部字符串、持续时间、旋转、颜色渐变、不透明度、刷子模式等等。同样，变换例行程序可以将来自用户的输入通过视像转换为任何希望的属性值。

[0084] 在另外一个示例中，可以使用转换器功能将图形元素属性或参数绑定到外部引用。在此，编辑器 50 可以允许用户来指定一个或多个转换器功能，当图形元素在运行时间环境中被使用时，该转换器功能将自动被应用来将源数值（运行时间变量）转换为目的数值（图形变量），或者反之亦然。例如，可以被绑定到提供速度数值的数据源的速度目的变量，可以基于源变量的值，被转换成一个字符串类型变量，例如表示为“slow（慢）”、“moderate（中速）”、“fast（快）”的字符串。这样的转换器功能也可以用于转换单元或提供其他转换。无论如何，这些转换功能都可以作为脚本或其他程序来实现，并可以出于任何目的而被使用来提供动作或动画，或者影响针对图形元素的显示属性。

[0085] 当然，可以为不同类型的图形元素定义任何需要的属性，所述不同类型的图形元素是针对过程控制内的不同物理元素创建的。例如，一个泵元素可以包含泵状态（例如开或关）、内压属性、外压属性、流入属性和流出属性。同样，与执行元素一同使用的调节阀门可以包括，例如，名称属性、内密度属性、外密度属性、流入属性、流出属性、内压属性、外压属性、内温度属性、外温度属性、阀门位置属性、阀门开启和阀门关闭属性（该属性可定义阀门是否一直处于开启或关闭状态）、设置点、过程数值和诸如线性、快速开启、等同百分比等等的阀门类型例如。当然，所述列表并非意在穷举。而且，任何这样的属性都可以被连接到针对图形元素的动画或动作。

[0086] 图 8 描述了第二属性视图 120B 元素，该视图可用于观察与图形元素属性关联的动画和绑定，所述图形元素属性针对的是分层视图 118（图 5）内的某个选定元素。图 8 所示属性视图 120B 描述了开启属性与针对图形元素的视像的各种部件联系的方式。具体地，如表项目 140 所示，存在一个动画，其涉及视像的背景，而且该动画基于开启属性来操作。表项目 142 展示了开启属性之间的绑定关系，且可以被用于访问动画、脚本以及开启属性和背景元素或视像图元之间的转换。这样，用户可以通过选择表项目框 140 和 142 中的按钮

来获得更多的信息。例如,通过选择表项目 140 中的按钮,用户可以访问能够引起视像背景颜色基于开启属性的值变化的动画。点击这样的动画按钮也可以引发编辑器 50 打开属性动画对话框,该对话框可以允许对动画进行管理和配置。而且,可以通过点击框 42 中的绑定按钮来实施和管理绑定,从而引发编辑器 50 开启一个对话框来创建、查看和编辑所述绑定。当然,也可以为图形元素的其他属性提供类似的屏幕,而且属性屏幕 120B 所示的属性列表将依赖于分层视图 118 中当前选择的项目。

[0087] 再次引用图 5,当视像被选在分层视图 118 中时,属性视图 120 将显示视像属性,例如名称、提供视像唯一标识的 ID、描述以及该视像是否被设为图形元素的默认视像。分层视图 118 或者属性视图 120 也可以指示视像是否被锁定,例如该视像是否可以被修改。

[0088] 另外,当从分层视图 118 中选择了—个连接器元素时,属性视图 120 将展示该连接器元素的属性,所述属性可以包括:该连接器元素种类(例如流体线路、电子线路、天然气线路、圆形或方形管道、传送带等)的指示;是向设备内部连接还是向外部连接(即,材料相对图形元素呈现的实体的流动方向);需要的或允许的最大连接数目;连接的宽度和高度;相对图形元素的连接位置例如是上面还是左边,等。

[0089] 当在分层视图 118 中选择了—个图元时,属性视图 120 将显示该图元的属性,所述属性可以包括例如 ID、类型描述,该图元是否可选和 / 或可见,该图元形状或种类的尺寸、长度或其他定义,图元的背景颜色和填充空间,等。当然,在分层视图 118 中任何其他选择项目也将使得该项目的属性被显示于属性视图 120 中,并且属性的类型和特点将依赖于所选项目的特点。

[0090] 如果需要,可以提供事件单来展示与分层视图 118 内的每个或所有选择的分层项目相关的事件。这样的—个事件单可以包括当用户针对某个图形元素施加动作时发生的事件,例如“鼠标单击”事件和“鼠标移动”事件,所述事件定义了用户点击图形元素的视像时所发生的事情,以及用户把鼠标放在图形元素的视图上或放在其一部分上所发生的事情。作为该事件单的一部分,用户可以访问脚本编辑器来定义或访问一旦事件发生就将要运行的脚本(在此定义为例行程序)。

[0091] 另外,如果需要,图形元素可以包含与之关联的常规事件。常规事件一般是被定义为某个外围或外部事件的结果,或者是由于用户对图形元素施加的某个动作,要求与外部应用程序或数据源进行通讯的事件。基本上,图形事件就是图形元素向信号发送的一个消息,该消息用于表示发生了针对图形元素视像的动作。实现图形元素的包含或运行时间应用程序可以使用例如—个 C# 句法或以某种已知方式显示事件通知,来捕捉这些使用的事件。在—个实例中,包含应用程序可以显示—个泵过热常规事件,而且当该事件被触发时,提供—个“我的句柄”功能,该功能将运行脚本或其他例行应用程序来使得用户能够操纵所述事件。当图形元素作为控制操作员屏幕的一部分被执行时,这些常规事件尤其有用。在这种情况下,元素例如通过对泵热属性(该属性与—个外部测量绑定在—起)和设置点或其他限制进行比较,图形元素本身可以确定事件情况,还可以通过将事件通知给用户并执行应用程序、脚本、对话等,来触发—个事件响应,从而使得用户能够操纵或处理该事件,在此所述事件为过热器。为了启动这个事件过程程序,元素编辑器 50 将允许用户定义事件自变量,即,什么能被认为是事件以及某个事件发生后的结果如何。

[0092] 当然,如果需要,图形元素可以提供从 Avalon 控制类继承来的标准事件。公知

的是, Avalon 对象或控制是微软为 Avalon 用户界面结构定义的图形, 所述结构用于微软 Longhorn 操作系统, 该系统支持在用户界面上显示向量图形。这些标准事件可以包括, 例如, 将查看面板显示或设备细节显示与鼠标单击或双击事件联系在一起。例如, 鼠标单击或双击事件用于报告图形元素视像范围内的鼠标单击或双击事件, 可以触发面板显示或视像呈现的设备或实体的细节显示, 并且为用户提供关于所显示实体的状态、设计、生产等的更加完整或全面的信息。当然, 其他事件, 包括标准事件, 可以针对键落下 / 弹起事件而定义, 该事件报告某个键被按下, 也可以针对鼠标事件而定义, 该事件包括鼠标输入、鼠标停悬、鼠标移动、鼠标盘旋等动作, 或者是其他任何用户初始化事件。

[0093] 因此, 如果需要, 可以在事件句柄脚本内触发图形元素事件, 例如在针对图元事件的事件句柄中, 或者是当图形元素的某个属性发生变化时, 在转换器功能内触发图形元素事件。而且, 图形元素的图元可以表示那些由用户与图元交互而触发的事件, 例如通过鼠标和键盘。这些事件可以为用户提供意见使得用户可以在形状和图元水平上与图形元素进行交互。开发人员可以通过使用诸如 C# 方法的方法指定事件句柄, 从而在图形元素内部操纵这些事件。

[0094] 再次参见图 5, 动作 / 动画视图 123 可以提供或显示一个转换动画和属性动画列表, 该列表是为当前显示在编辑视图 114 中的视像定义的。用户可以, 例如, 双击视图 123 内的一行, 从而引发编辑器 50 呈现对话框, 该对话框可以是转换器动画对话框或者属性动画对话框, 所述对话框允许用户编辑所选动画。具体地, 只有当前选定视图的动画才会被显示, 但是选中了显示全部框 (Show-All) 144 将显示所有视像的动作和动画。假设为动画提供了有意义的名称或描述, 例如, “驱动杠杆”、“旋转马达”等, 动作 / 动画视图 123 可使用户容易地查看和访问为视像定义的动画行为。图 9 描述了示例动作 / 动画视图 123A, 该视图描述了为开启属性定义的动作。在该例中, 当开启属性为“真 (True)”时, 被称为 Visual1 的视像的被称为 Rectangle1 的图元执行填充动作, 填充值为“red”。同样, 当开启属性为“真 (True)”时, 被称为 Visual1 的视像的被称为 Ellipse1 的图元执行可视动作, 值为“假 (False)” (例如使该图元不可见)。当然, 可以分别为视像的独立图元定义动作和动画, 但是这些动作和动画可以同时实施在同一事件、触发或属性变化上, 从而展示一个更加复杂而令人满意的动画。而且, 动作和动画可以包括实施单独的操作, 例如增大尺寸、填充颜色等, 或者包括连续发生多次直到关闭的操作。

[0095] 图 10 描述了一种在图形元素 152 的视像 150 上基于图形元素 152 的属性 (例如固有属性) 来实现视像触发器 148 的方式。具体地, 通过针对视像触发器 148 定义的一个或多个属性触发器 154 (图 10 中示出三个) 可以监视图形元素 152 的一个或多个固有属性。可以使用脚本来实现每个属性触发器, 而所述属性触发器可以监视特定的图形元素属性的值, 如虚线 155 所示。之后, 当被监视的图形元素属性值令人满意或满足特定条件时, 每个属性触发器可以设置视像 150 的一个或多个图元属性值, 如虚线 156 所示。因此, 例如, 某一个属性触发器 154 可以监视一个或多个图形元素属性, 从而确定所述一个或多个属性的值何时落在某个特定范围内。当满足这个条件的时候, 所述某个属性触发器 154 可能引发一个动画或其它视像脚本运行, 例如, 视像 150 的图元或其它元素, 从而提供视像触发器 148。当然, 多个属性触发器 154 可以共同运行以提供多个同时发生的变化或动画, 作为视像触发器 148 的一部分, 或者不同的属性触发器 154 可以独立运行, 在例如不同的固有属性

或者基于同一固有属性的不同值上,从而在不同时段或响应不同的过程条件提供视像触发器 148 的不同操作。这样,可以基于图形元素 152 的固有属性,在视像 150 中提供颜色变化 / 动画等。

[0096] 如果需要,编辑器 50 可以提供或显示视像触发器面板,该面板列出了针对当前所选视像的所有属性触发器。这样的面板如图 9 所示,在图 9 中,元素属性栏列出了图形元素属性的名称,第一数值栏指出了正在观察的图形元素属性值,目标栏提供了即将改变的视像或图元识别,路径栏是正在改变的图元属性,第二数值栏是当被观察的图形元素属性值满足条件时要应用的图元属性。当然,可以使用这个结构来提供动画和其他视像变化。

[0097] 图 11 描述了变换动画对话框 160,该对话框可以用于帮助或使得用户可以指定变换动画。如图所示,变换动画对话框 160 包括:移动部分 162,该部分允许用户根据在动画内定义运动的像素来指定方向和距离;旋转部分 164,该部分允许用户指定动画的旋转方向和角度;和缩放部分 166,该部分允许用户指定动画在水平和垂直方向上的缩放程度,并指定是否其纵横比例是否被锁定。对话框 160 还包括倾斜部分 168,该部分允许用户指定动画运行时间在水平方向和垂直方向上的待应用的倾斜。设置部分 170 允许用户定义该动画是否为连续的,并启动其他每个动作、旋转、缩放和倾斜动作。而且,预览框 172 可以描述动画预览。

[0098] 可以理解的是,动画是一个其值在某个时间段内变化的对象。通过将动画和图元属性关联在一起,可以获得属性动画。属性动画可以做的比较精妙,例如改变文本颜色或者使诸如线的元素闪烁。另一方面,属性动画也可以更加复杂,例如将多条线的点做成动画,等等。当然,这些只是少数的动画实例,其他动画实例还包括添加或改变颜色、改变图元尺寸(例如宽度、长度或点磅值)、移动、旋转、倾斜、缩放图元等等。而且,可以在视像的任何级别上提供其他动画,例如在图元级别或作为整个视像的一部分。当然,如果需要的话,可以将多个动画提供给或应用在任何特定视像上或视像的任何特定图元上。而且,与某个特定视像或视像的图元关联的多个动画图元可以同时运行,或者基于属性值的变化或基于诸如鼠标事件的用户初始化的触发事件而在不同时间运行运行。

[0099] 图 12 描述了属性动画对话框 180,该对话框可以由编辑器 50 来生成,从而使用户可以定义或改变动画的属性,从而定义该动画。对话框 180 包括范围定义部分 182 和时间线定义部分 184。范围定义部分 182 提供或定义动画的范围。具体地,属性“始于(from)”定义了动画的起始值而属性“到达(to)”定义了动画终止值。时间线定义部分 184 定义了动画的持续时间、动画的开始时间和动画的结束时间。持续时间属性定义了动画完成的时间长度,开始时间属性定义了相对动画开始时间的的时间偏移量,而结束时间属性定义了相对开始时间的动画结束时间。速度定义部分 186 允许用户指定使用例如滑动条的工具来指定速度、加速和减速。明显的是,速度属性定义动画的速度,加速属性使动画相对渐增的时间播放更快,而减速属性使动画对于渐增的时间播放更慢。重复定义部分 188 允许用户定义一种重复动画的方式,例如,重复一定次数、重复指定的计数或持续时间或一直重复直到被关闭。

[0100] 可以通过对话框 180 访问的进一步的动画对话框 190,当使用范围定义部分 182 时,可被用来指定动画范围所使用的不同颜色。可以使用类似对话框来使得用户能够选择动画属性的其他非数字的量。虽然未被示出,但是其他动画属性也可以被选择或指定来使



用这个或其他对话框。例如,可以使用自反属性来定义一个布尔值,该布尔值表示在动画在前进方向完成后是否沿着反方向继续播举,还可以使用“通过 (by)”属性来定义总的数量,通过“通过 (by)”属性动画可以改变其初始值。相对速度属性可以用于定义相对速度,在某一时间点,所述动画以相对于父动画(即,与视像的父元素相关的动画)的相对速度推移。例如,如果相对速度的值为 1,指的是动画推移的速度与父动画的速度相同,如果值为 2,是指动画推移的速度是父动画的两倍,以此类推。

[0101] 虽然未显示,但对话框还可以用于设置动画的字体属性,例如动画的风格、大小、字体、文本颜色以及其中的变化。另外,可以使用笔画属性对话框来设置边缘或线型、线条的厚度和颜色,或者改变这些属性。

[0102] 图形对话框也可以用于编辑图形元素的其他特征。例如,图形对话框可以用于创建新的过程图形,所述过程图形包括将一种图形元素属性添加到某一图形元素、以及将图形元素属性绑定到运行时间环境。这样的—个绑定对话框可以提供—种浏览器,该浏览器使得用户可以浏览定义在控制系统或其他运行时间环境中的不同标签或变量,从而定位所需的标签、变量、名称等等来实现绑定。同样,也可以使用其他对话框来添加视像、添加视像触发器、通过图形元素或显示添加浏览器以及添加事件。

[0103] 因此,可以理解的是,图形编辑器 50 提供对定义图形元素固有属性的支持、对图形元素视像的支持,包括创建和操作图形图元或形状、添加动态行为,尤其是添加转换动画(旋转、平移、缩放、弯斜)和属性动画(长度动画、颜色动画等),还包括定义执行动态行为的触发条件。而且,图形编辑器 50 提供对保存和调用数据库中图形元素的支持,包括对图形元素序列化成 xml blob 的支持,以及对图形元素库中的图形元素进行分类的支持。进一步,系统提供将图形元素保存在用户定义类或组中、锁定或提供针对这种图形元素的安全性、将图形元素储存在用户定义的或其他有用的类型、对图形元素进行版本化,等。

[0104] 在任何事件中,一旦建立,图形元素都保存在图形元素数据库中,而且并未绑定到加工厂的运行时间环境内的过程变量或其他数据。这种存储的图形元素不一定用于任何显示,但是当需要使用时,可以将所需元素下载到运行时间环境,或者绑定到加工厂内或过程控制系统内的特定真实或模拟的物理元素。当元素被绑定时,该图形元素的固有属性,例如过程变量、设置点、当前速度等,也被关联到运行时间环境的数据参考中。

[0105] 图 13 描述了图形元素被联系到或被绑定到加工厂内或使用于加工厂内的过程控制系统中的实际物理部分或元素的方式。具体地,每个图形元素 192 包括 XAML194,该 XAML194 主要定义了元素的视像,还包括关联脚本 196、动画 198、触发器 200、动作 202 和事件 204。图形元素 192 还包括引用表 206,该表列出或包括所有与 XAML194 关联的引用,这些引用在 XAML 中可用或公开。引用表 206 主要由以下部分组成或包括以下部分,即,图形元素的属性和参数元素以及由脚本 196、动画 198、触发器 200、动作 202 和事件 204 所使用的其他任何变量。引用表 206 内的变量或实体可以涉及或绑定到其他程序中的变量、表格、表项或者是过程控制系统中定义的其他任何种类或类型的数据。

[0106] 如图 13 所示,判决表 208 用于将引用表 206 中的引用或变量绑定到实际的过程控制环境或其他运行时间环境。一般来说,判决表 208 可以或者直接或者通过别名来定义引用,当图形元素 192 被配置用于运行时间环境中的某个特定显示并且被下载到运行时间机器时,所创建的判决表 208 将被提供给图形元素 192 使用。在显示中的图形元素 192 运行

之前或运行过程中,判决表 208 判决别名和其它参数,然后将这些判定的数据绑定连接到引用表格 206,从而在引用表 206 的变量和过程控制系统或其它运行时间环境内的数据源之间提供了绑定。

[0107] 为了避免运行时间期间过程控制系统中处理能量的不必要使用,当图形元素和图形显示(过程能力在二者当中使用)实际中并未在任何显示屏幕上被显示或使用时,不需要同运行时间环境保持绑定。取而代之的是,只有当图形元素 192 正被运行或显示在在运行时间环境内的屏幕上时,才将判决表 208 绑定到图形元素 192 的引用表 206。

[0108] 由于每个图形元素都通过使用引用表 206 和判决表 208 被联系到过程控制系统或运行环境,而且由于绑定发生在图形元素被创建在系统级别以及被复制和载入到运行时间环境或机器之后,图形元素 192 可以被分别绑定到不同的数据源,从而在不同时间用于不同用途。而且,图形元素 192 可以被绑定到由任意数量的不同源生成的数据,包括图 2 的不同功能区域 42 中的数据源,例如与控制活动、维护活动、模拟活动、仿真活动、配置活动等关联的数据源。

[0109] 例如,如图 14 所示,显示元素 192 可以被联系到、或者可以被用于显示或过程数据,所述数据由大量不同种类的数据源产生或可以从所述大量不同种类的数据源获得。这些数据源可以包括:控制环境数据源 210,例如像众所周知的 DeltaV 控制系统;OPC 数据源 212,其提供通过著名的 OPC 连接接口到其它系统的连接;维护数据源 214,例如有名的 AMS 系统;高级别或商务系统 216,例如著名的 Ovation 系统;甚至还有使用竞争系统 218 的数据源,例如竞争控制应用程序。这样,使用来自系统内任意应用程序的数据,图形元素 192 可用于显示系统任意级别的数据或 / 和物理元素视像,即使所述数据来自或源自大量不同种类的应用程序,包括与竞争系统关联的、从未有意于一起操作的应用程序。因此,由于图形元素和从图形元素所创建的元素图形显示以系统级别构建,它们可以被用于提供工厂内正在发生的、针对任何目的的情况的视像,即使不同种类的软件正在访问并运行工厂内不同的硬件和软件部件。

[0110] 由于图形元素具有模块特性,可以从用著名的和备受推崇的设备描述语言(Device Description Language, DDL)写成的设备描述(Device Descriptions, DD)来自动或半自动地创建图形元素。特别地,设备生产商通常都为其生产的每个设备提供 DD,在 DDL 中,DD 定义了与设备关联的参数、怎样与设备通讯、设备的限度,等等。结果,元素图形元素创建应用程序可以针对设备在 DDL 中读出 DD,从而确定设备的种类和与设备关联的重要参数、限度等,然后将这些参数定义成设备图形元素的固有属性或参数。基于来自 DD 的信息或者基于为该设备类型的设备所保存的模板,程序也可以选择或定义基本形状合成作为针对设备的视像,也可以选择一个或多个通用脚本用来为设备提供基本的动作和动画,所属设备类型由设备的 DD 来定义。如果需要,在这种过程中,程序可以请求用户提供关于设备的信息或者决定为图形元素使用那些动画、视像、图元等。

[0111] 对于一个较完整和定义完善的元素图形元素,程序可以为不同类型的设备存储各种通用图形模板,所述设备例如传感器、阀门、马达、罐等等。然后程序可以确定模板图形元素,从而基于通过 DD 为设备定义的设备类型来使用元素。如果需要,模板可以提供或具有可用于图形元素的各种选择或选项,而且这些选择可以基于针对设备的 DD 内的信息或基于用户输入来确定。这样,例如,模板可以提供与诸如传感器的设备的各种子类型相关联的

各种固有参数,而且程序可以确定固有参数,从而基于 DD 的信息来定义图形元素。

[0112] 各种基本的脚本可以被提供作为模板的一部分,也可以被使用来为视像提供行为。而且,可以基于例如设备类型等的 DD 中的信息,或者,如果需要,可以基于针对用户的问题,来自动选择用于图形元素的脚本。另外,可以根据 DD 的信息来确定用于脚本的各种限制或变量。从而,例如,如果 DD 表明该设备是旋转设备,脚本就可以被选择或使用,例如一个提供旋转图形的脚本,用于或提供给图形元素,且旋转图形的某些方面(例如速度、图形改变颜色的时间,等)可以是基于与 DD 所定义设备关联的限制。这些限制可以是诸如正常或额定的运行速度、定义的超速或低速条件或限度,等。举另外一个实例,如果设备是传感器,传感器的高值和低值可以被用于提供与传感器的当前读取相关的图形,描述传感器是否出故障的图形,等等。

[0113] 在这种方式下,当设备被放在或识别位于一个加工厂内时,可以通过为图形元素定义特定的基本脚本、图形动画、视像和固有参数,基于 DD 为设备定义的设备类型和设备的已知特点,从针对设备的 DD 来自动创建基本的图形元素。所述图形元素的自动创建使得用户可以将一个新加入的设备自动集成到图形中,而且如果要对设备的图形进行编辑,则不必进行大量的执行操作,这样,当针对那些设备的 DD 被加载到系统中时,可以在图形显示中自动支持设备,至少在基本级别上可以支持。换句话说,通过为系统设备提供 DD 然后运行程序以针对来自 DD 的设备创建图形元素,用户可以自动从 DD 为设备创建一个图形元素。之后,这个图形元素可以被用于一个或多个图形显示来模拟或提供设备的视像,所述视像使得用户可以模拟该设备或描述图形显示内的设备,而无需手动为设备创建图形元素。

[0114] 如上所述,一旦已经创建了图 3 所示的多个图形元素 74,图形编辑器 50 可以被用于创建一个或多个图形显示,例如图 3 所示的显示 76。事实上,如果需要,显示编辑器 50 可以为用户或购买者提供多个预配置的图形元素,这些图形元素可以具有与之相关的各种视像,例如针对不同产业的视像、针对不同功用的视像等。然后绘图编辑器 50 允许用户创建定制元素以及创建或建立图形显示 76。

[0115] 一般来说,要创建显示 76,用户要从图形元素 74 和其他视像元素的库中选择所需元素,然后将元素它们放在一起以建立一个显示。在完成某个图形显示后,作为结果的数据结构或对象,可以作为具有所有不同元素的现实类对象、固有属性和定义为单个定义实体的视像,被存储在一个诸如配置数据库的数据库中元素。然而,所述显示类对象将不再绑定于过程变量,也不一定用于任何运行时间显示。如果需要,可以将所述类对象作为 XML blob 存储在数据库中,所述 XML blob 具有视像、脚本等等,所具有的这些都存储并连接在一起组成 XML 实体。之后,可以从这个类对象来创建单独的图形显示,且这些单独的图形显示可以被分配和下载到操作员工作站或其他运行时间环境。

[0116] 当某个显示被下载时,这里的图形元素定义将被转化为 Avalon 控制、编译成汇编语言并被配置到目标机器,例如,配置到运行时间机器。显示本身是一个 Avalon 控制且涉及编译过的控制汇编,另外,可能产生一个定制数据源,该数据源作为数据适配器将 Avalon 控制连接到后端数据源,例如控制或维护运行时间应用程序。在这种情况下,可以用一种语言创建和编辑图形显示(需要的话也可以是个体图形元素),用另外一种语言或形式保存所述图形显示,然后用第三种语言或形式运行所述图形显示(例如,绑定于用任何可执行语言所写脚本的 Avalon 控制)。

[0117] 现在参考图 15,可以由绘图编辑器来生成屏幕 220,从而使得用户可以创建一个或多个图形显示。一般来说,图像显示由代表工厂内的物理设备的互连元素组成,所述图形显示还可以包括或显示与那些设备相关的其他信息。图形显示中的每个图形元素包括一套固有属性,所述固有属性相当于过程变量、常量或其他外部值,而且,如上所述,每个图形元素可以具有几个视觉表现,包括动态行为,例如颜色变化或动画。另外,图像显示可以包括例如文本、框的静态元素,允许用户与图形显示以某种方式交互的用户发电机、可以为用户显示过程或其他信息的变量框,等等。

[0118] 与图 5 所示针对图形元素的编辑屏幕 112 类似,图 15 所示示例图形显示编辑器屏幕 220 包括:主要编辑画布 224,在该画布中建立图形显示;和调色板部分 226,在此可以显示模板图形元素、图元或其他库元素,而且可以从中选择各种元素,并将各种元素拖放到主要编辑画布 224 中。主要编辑画布 224 提供了图表视图,在此用户通过互动来建立和编辑图形(或者是图元或者是合成),并且提供向量图形编辑和查看特性。编辑器 50 使得用户可以将诸如图形元素的图形对象放置在不界空间的任何地方,所述不界空间可以在当前显示在编辑画布 224 的显示框的内部或外部。因此,主要编辑器 224 只可以描述当前创建的显示的一部分,而用户可以整理视图以显示其他部分,可以放大或缩小视图以改变视图的放大倍数,还可以旋转视图以得到最方便的朝向,从而在给定时刻修改显示。然而,可以理解的是,所有图画和对图形对象的操纵都在主要编辑画布 224 内完成。为了方便绘图,可以从标尺 227 读取视图的 X-Y 位置,放大倍数可以显示在工具栏 228 的一个下拉列表,并可以通过选择缩放弹出项来改变,等。

[0119] 屏幕 220 还包括:分层部分 230,该部分描述了主要编辑画布 224 中的分层显示或元素列表;属性部分 232,该部分列出了与画布 224 中建立的显示或画布 224 内高亮度元素相关联的属性或参数;和绑定部分 234,该部分描述或列出了不同参数或特定被绑定到过程控制系统或其他任何运行时间环境中的元素。另外,屏幕 220 包括工具栏 236,该工具栏列出了与主要编辑画布 224 中创建的图形显示相关联的各种视图或层。更具体来说,任何特定显示都可以具有能被不同人群使用的、用于不同环境的各种层或视图,例如操作符视图、维护视图、仿真视图、模拟显示、工程视图、业务视图等等。在图 15 所示的示例屏幕 220 中,工具栏 236 包括三个视图和级别,其被描述为:操作视图(当前显示在主要编辑画布 102 中),该操作视图提供典型的控制操作符视图;Eng/维护视图,其提供典型维护或工程视图;和训练视图,该视图提供用于仿真在创建的显示中所描述的加工厂或加工厂某部分的仿真视图。由于在图 15 所示主要编辑画布 224 内没有选择任何元素,属性部分 232 显示了正在创建的显示名称以及显示特点,例如宽和高、创建者提供的描述、名称、背景描述和创建者希望为显示保存的任何其它信息。类似的是,由于在图 15 所示主要编辑画布 224 内没有放置实际元素,分层部分 230 只展示了显示的名称,并且没有在绑定部分 234 显示绑定关系。

[0120] 在建立显示的过程中,用户可以通过例如定义于平面部分 226 的一套类别的元素中的一个来访问某个元素,从而将诸如图形元素的元素放置在主要编辑画布 224 中。在图 15 所示的例子中,平面部分 226 描述了不同种类的元素,所述元素可以放置在主要编辑画布 224 并可以连接到一起来创建一个完整显示,包括执行元素、计算和控制元素、过程元素、属性和测量、形状、用户接口控制和用户定义的元素。当然,任何其它种类和类别或子类

别的预定义元素可以提供给平面视图 226 或者被访问。在这个例子中,执行元素可以包括阀门和其它执行元素,而计算和控制元素可以包括任何与控制相关的元素,例如控制器的指示、诸如 PID 控制环或其它种类的控制环、功能块、控制模块等。图 15 所示的过程元素可以包含罐、反应堆、混合器或其它采用某种方式的过程材料的元素,以及任何其它种类的设备、单元等。属性和测量可以包括:被设计用于显示属性、测量的框或显示元素;或运行时间环境中的其它数据,例如过程变量、报警等。形状可以是图元或其它预定义形状,而 UI 控制可能包括各种用户接口控制元素,例如按钮、滑动条、旋钮、工具箱等,用户可以在显示屏幕上操纵所述用户接口控制元素来实现显示的输入。当然,用户定义的元素可以包括任何预定义的元素,例如由图形元素制作的任何其它图形元素或较高级别元素。在某种情况下用户定义的元素可以包括过程单元、加工厂区域或其它高级别过程实体。如上所述,可以使用编辑器基于用户身份来对用户定义元素的访问加以限制,或者用户定义元件也可以是全局可访问的。当然,可以理解的是,任何其它元素、图形等都可以被放置在平面部分 226 中并配以适当的标题,从而可以合理组织这些元素,也便于用户访问。

[0121] 当用户将一个元素,例如混合罐,放置在主要编辑画布 224 中,可以使用该元素的默认视像使该元素呈现在画布 224 中。在这种情况下,图形分层部分 230 将会以类似图 5 所述的方式显示该元素,并为该元素提供子元素分层,例如与图形元素关联的视像、动画、图元等。而且,当图形元素显示在的画布 224 中时,绑定预定部分 234 将描述当前定义的绑定,画布 224 包含可能绑定到运行时间环境的视像或动画。

[0122] 如果需要,用户可以通过以任何希望的方式选择不同的视像,来选择或改变主要编辑画布 224 或平面部分 226 中的图形元素的视图。用户可以使用一个下拉列表或框来实现这种视像选择功能,可以例如通过在元素上采用鼠标光标单击右键、或在分层部分 230 内选择一个不同的视像或任何其它希望的方式来访问所述下拉列表或框。图 16 描述了屏幕 220 的主要编辑画布 224 的一部分和平面部分 226,在屏幕 220 中,所示的垂直罐元素 240 被从平面元素 226 放置到主要编辑画布 224 中。垂直罐元素 240 是一个模板或平面部分 226 所示类垂直罐元素 241 的复制或示例。然而,当垂直罐元素 240 位于主要编辑画布 224 中正在创建的显示中时,垂直罐 240 的附加视像,包括侧条 242 所示的视像 1 到 8(当鼠标光标停留在视像 240 上时,通过单击右键可以访问该视像),可以被选用作为针对所述垂直罐元素 240 而使用的视像。如平面部分 226 所示,通过右键单击平面部分 226 的模板垂直罐元素 241,可以得到相似种类的侧条显示 243,从而获得或查看模板罐元素 241 的可能视像。在侧条 242 中选择一个不同的视像将会改变用于画布 224 中的罐元素 240 的视像,在侧条 243 中选择一个不同的视像将会改变存储于平面部分 226 的模板罐元素 241 的默认设置或视像。

[0123] 除了从平面部分 226 选择图形元素和其它元素以创建显示,用户还可以使用或选择图 15 所示的工具栏 228 中的项目,从而获得基本绘图工具,例如线条和正方形、长方形、圆、五角星等形状,以及文本等,还可以使用这些简单绘图工具或元素来在显示中绘制线条或添加文本。另外,用户可以使用工具栏 228 所示工具栏连接器元素 245 来向画布 224 中的显示添加连接器元素。当用户选择了连接器元素 245 时,该元素可以使用下拉菜单或对话框等,为用户提供一系列连接器。可能的连接器元素包括管道、传送带、电子线路、流体线路或其它种类的连接,所述连接器实际上将一个硬件元素(如罐、混合器、泵等)连接到另

一个硬件元素（如阀门、传感器等）。在美国公开 No. 2004/0153804 中，对这样的连接器元素以及在不同物理设备表现之间提供连接器元素进行了更加详细的描述。连接器元素可以允许用户把创建中的显示的不同元素进行互联，以及与为图形元素的上述不同元素定义的连接点紧密配合。例如，管道连接器可用于连接具有管道连接点的不同元素，而输送管道连接器可用于连接定义了输送管道连接点的元素。如果需要，编辑器 50 可以实现只允许用户所进欧泡蘑菇合适类型的连接器连接诸如阀门和罐的不同图形元素的连接规则，正如阀门和罐的连接点所定义的罐罐。当然，连接器元素的外观可以基于连接类型而不同，从而为创建中的、具有连接器元素的显示提供更加完善的外观和感觉。

[0124] 当然，工具栏 228 可以包含其它向量绘图工具和标准命令或函数，例如典型的文件选项（新建、打开、保存、另存为、关闭、退出等），编辑选项（例如撤销、重复、剪切、复制、粘贴、删除），查找选项，等等。另外，工具条 228 还可以包括提供图形相关命令的菜单项目，例如用于添加图形元素固有属性的“添加属性”功能，用于添加绘图事件的“添加事件”功能，允许用户添加视像的“添加视像”功能，“添加视像触发器”功能，该功能可用于打开对话框以允许用户输入与待建触发器有关的信息，等等。而且，菜单可允许具有格式功能（例如设置文本、线条和二维 / 三维图形的有关文本、线条和填充属性）和形状功能，该功能可以包括将对象分组或解除分组、从前向后选择对象的顺序、在垂直和水平方向上对齐等等。工具栏菜单 228 也可以提供其它功能来使得用户可以把动画应用在对象上，这样的动画也可以应用在一个元素的图元或对象的另一元素上，例如通过对元素的宽、颜色、形状等施加动画效果来对形状进行弯斜、旋转或缩放，来旋转或翻动元素，等。当然，用户可以访问与显示与每个元素关联的动画和动作，并可以使用合适的对话框或其它编程工具来改变这些动画和动作。

[0125] 可以理解的是，用户或设计者可以使用显示编辑器 50，特别是可以通过界面访问图 15 所示屏幕 220，来快速而简便地创建图形显示，该图形显示由可以动态显示测量、执行元素和过程设备的标准二维甚至三维图形元素组成。而且，可以在显示中支持和提供与控制相关的静态元素和用户动态图标。另外，通过使用具有略微不同视图或环境信息的相同元素来创建操作符、维护和仿真显示，用户可以定义层来提出不同用户的图形显示界面要求。

[0126] 图 17 描述了一个示例图形显示的显示屏幕 300，该显示的形式是控制操作符显示，并可以使用图形编辑器 50 来创建。显示屏幕 300 包括一个反应堆元素 310，其通过连接器元素 316 连接到阀门元素 312 和一个或多个泵元素 314，在此的连接元素是流体线路。流体或水流元素 318 为进入和离开屏幕所示工厂部分的材料流提供引用。可以理解的是，可以通过选择和将各种图形元素互连在一起来创建显示屏幕 300 中的图形，因此显示屏幕 300 可以包含高分辨率的、以针对个体图形元素描述的方式所开发的图形、动画、动作、视像触发器等。当然，附加图形，包括动画、动作、视像触发器、流体元素等，可以被添加到图形显示级别。这样，动画和用户动作可以合并到图形显示中。例如，使用这个功能，针对过程设备的静态图形部件可以被修改以指示设备状态，例如，某个马达是否开启或关闭、某个阀门的运行状况或位置，等等。另外，动画可以用于展现与设备相关联的动态数据，例如说明罐的填充水平（例如，使用填充技术）或者通过表示动作的显示变化（动画）来说明搅拌器的状态。同样，可以将数据显示元素 319 放置在屏幕 300 来显示过程数据或者在屏幕 300

外面开发的、但与显示器 300 的部件操作有关的其它数据。

[0127] 如果需要,图形显示器 300 也可以包括动态图标 320A 或用户界面按钮 320B,其中动态图标 320A 和用户界面按钮 320B 使得用户能够通过界面访问显示器 300,所用方式允许用户通过与显示屏幕 300 进行互动来查看附加信息或针对运行时间环境采取动作。某些情况下,这些互动可以用针对图形元素的上述视像触发器或动作来实现。例如,按钮 320A 可以为用户提供关于冷却塔结构的深入信息,而按钮 320B 可以为反应堆 310 提供面板显示。因此,显示 300 包括一套元素,这些元素描述了使用三维部件组成过程或过程部分的各个设备,所述三维部件过去是基于从多个不同源引进的图形而做成的,所述的源包括例如 In-Tools, Auto-Cad, Windows 图元文件(如 Visio), 向量绘图, JPEG 和位图图像格式, 屏幕 300 现在可以包含将要以元素级别执行例如旋转、改变大小、缩放、弯斜、改变颜色等等的动画,从而提供更多有趣而实用的动画,并因此提供更加容易理解的显示。

[0128] 而且,基本用户接口部件,如旋钮、刻度盘、滑动条和按钮,可以在显示 300 中显示并可以自动链接到控制系统或其它运行时间环境中的信息或控制。数据视图元素或动态图标也可以提供或描述与控制功能、报警、事件等测量有关的关键参数。例如,动态图标可用于显示一个屏幕,该屏幕具有与控制环参数有关的信息,以及显示工程单元,所述屏幕可以显示在动态图标旁边来提供动态图标所显示数值的上下文。可以在动态图标中反映与控制环关联的过程报警的状态,例如,控制参数值背景颜色的改变。而且,为了消除显示混乱,某个环没有在设计的正常操作模式这个事实可以通过颜色变化来指出。当然,可以使这样的动态图标符合任何标准。

[0129] 举一个例子,当一个操作员访问用户交互部件或动态图标时,脚本或其它程序可以调出另外一个屏幕或显示,例如一个面板显示或控制面板显示,这些显示的例子如图 18 所示。例如,当用户界面按钮,例如屏幕 300 所示的按钮 320B,被访问时,反应堆 310 的面板被展示给操作员,然后操作员可以使用这个面板来修改或查看这个反应堆 310 的细节。在图 18 所示的例子中,面板信息 350 与反应堆 310 的控制环(称为 FIG2\_28/TC2-1)关联在一起,用户可以通过屏幕 300 中的按钮 320A 来访问该反应堆 310。使用面板 350 的部分 350A,用户可以使用按钮 352(该按钮的操作由动作例行程序来定义)改变模式(例如从层叠模式改为手动或自动模式)、可以查看滑动显示 354、可以使用箭头 355 来改变与控制环连接的设置点,等等。另外,可以为用户提供显示部分 350B 中控制环的限度和调节参数,还可以为部分 356 赋予仿真能力。如果需要,可以通过选择部分 350A 中的一个按钮 358,从部分 350A 的视图获取部分 350B。同样,用户可以进一步访问关于控制环的信息,例如倾向数据、诊断数据等等,或者可以通过其他按钮 358 来访问和运行控制和诊断程序,例如环调整程序。因此,任何其他活动、屏幕和动作可以响应在屏幕 300 上被允许的或被采取的用户动作,而通过显示 300 被访问。

[0130] 控制系统支持使用别名来定义相似设备的地方,也可以基于从显示屏幕 300 中选择的设备,设计动态显示部件以支持动态参考。在这种情况下,预配置的别名和属性可以代替正常定义的对象标签或图形属性来使用,所述对象标签或图形属性定义为显示对象的一部分。所述别名性能支持高度的灵活性和重复使用性,因为相似的显示对象可以连接到不同的输入/输出点,并代表不同的图形属性、外观和安全。这样的性能可以消除为工厂内的不同复制设备重建相似显示对象或相似显示的必要。这样,可以使用相同的图像显示来查

看不同的硬件单元,所述硬件单元在工厂内具有相同的结构和用途。

[0131] 当然,可以设计图 17 所示的显示屏幕 300 以支持工具栏,除了或代替屏幕 300 上方描述的工具栏外,还包括水平(显示下方)和垂直(显示右方)工具栏。如果需要,可以提供默认工具栏以支持时间和日期显示、直接访问要求确认报警或关闭报警的报警显示以查看报警列表、对报警摘要显示或菜单、主菜单或其他标准菜单或显示、系统状态显示等的导航。

[0132] 图 19 描述了与又一个高级图形显示相关的显示屏幕 400,即,具有更多元素、连接、用户界面动作、动态图标和其他数据参考。特别地,屏幕 400 描述了石灰窑单元的操作,在所述石灰窑单元,空气和其他燃烧燃料产物通过泵 412、阀门 404 和相关的连接器元素被抽吸到或反馈到窑桶干燥器 408 的输入 406。同样,罐 409 内的过程材料从平板传送设备 410 进入,所述平板传送设备 410 通过桶干燥器 408 来运输石灰。当然,屏幕 400 中所述每个元素和其他设备都可以是上述可以单独建立并可以放在屏幕 400 内的图形元素。桶干燥器 408 的输入 406 可以包括动画图形元素,当干燥器 408 正在运行时,所述动画图形元素以动画形式来显示火或者火焰 415,从而向查看屏幕 400 的人清楚地指出窑单元的操作。另外,可以通过动态图标或温度显示框 416 来指示干燥器 408 内各个点的温度,如果希望,可以通过火焰动画的颜色,干燥器 408 或其他任何方式来指示所述各个点的温度。同样,可以用图 19 所示的参数框来描述其他过程参数,例如机罩压力、窑速度、空气总量、甲醇、输入、初级和二级空气输入、油气输入等,所述参数框具有框内变量的值,这些值被绑定到或来自于过程控制系统内的特定引用。当然,屏幕 400 还描述了石灰窑单元的其他部件。

[0133] 然而,屏幕 400 也可以提供各种其他信息,所述信息可以来自运行时间环境的其他数据源,例如处理来自控制系统或控制系统内设备的数据的其他应用程序。这样的应用程序可以包括例如控制应用程序、维护应用程序、诊断应用程序、业务应用程序等。例如,显示系统温度(单位为度)随时间变化情况的趋势图形 420 被显示在显示屏幕 400 的中上方,并被绑定到一个历史数据记录器或趋势应用程序,并自动绘图所述数据。同样,提供窑 412 操作内容的图表 422 被显示在屏幕 400 的左上方,其中所述操作内容可以通过跟踪各种业务方面的业务应用程序来提供,所述各种业务方面包括窑的利润、能量使用、生产率等。而且,可能对用户有用的其他信息,例如历史趋势、帮助信息、窑信息、控制约束、窑能量和其他信息,可以通过选择位于显示屏幕 400 上面部分的用户界面按钮 424 来访问。而且,可以在屏幕 400 的底部提供一个报警标语 430 或其他标语,还可以将所述标语联系到报警程序。当然,可以使用上述显示编辑器 50 的特点,将显示屏幕 400 的所有这些元素和特点提供给或编程到显示中。另外,可以将这些和其他特点以任何希望的方式联合到图形显示,从而创建任何期望类型的显示。而且,可以在屏幕 400 内提供任何希望的动画和绘图动作,从而为用户提供更加有用或可理解的视像信息,并允许用户更直接地查看来自屏幕 400 的其他有关信息,等等。

[0134] 因此,如图 19 所示,显示在显示屏幕 400 上的可以是来自各种不同数据源的信息,包括:控制器子系统访问的过程元素;以维护、控制、诊断、调整形式存在的应用程序;业务应用程序;来自诸如历史数据库或其他任何数据源的数据库的历史数据或趋势数据。而且,可以通过用户到所述显示的链接,直接访问来自显示屏幕 400 的其他信息或显示。这样,显示屏幕 400 描述的图形显示就以更加有用的方式为用户提供了更多的信息。



[0135] 由于编辑器 50 可以使用一套公有图形元素、连接元素等而被用来创建任何图形显示,因此编辑器 50 也可用于很容易地创建类似或相关的显示。这样的显示可以涉及或显示同一套工厂硬件,但是描述了该硬件的针对不同目的的不同信息,例如针对控制操作目的、仿真目的、维护目的等等。这种情况下,可以使用一个公有或基础显示描述与工厂或工厂某一部分相关的硬件元素,并可以使用这个基础显示来创建不同的显示,但是所述不同显示为不同用户或不同类型用户提供不同的信息。

[0136] 举一个例子,图 20A-20E 示出了几个示例显示,包括为石灰窖创建的、显示相同基本硬件元素的操作员视图、工程师视图、管理者视图、仿真视图和维护视图,这些视图具有相同的配置并使用相同的视像,但是具有针对不同功能目的而添加的不同信息。因此,图 20A-20E 中的每个显示都具有相同的外观和感觉,因为这些显示是由同一显示编辑器、使用相同基本元素做成的,因此,当保持对展示信息的理解以及理解该显示怎样与工厂内硬件发生联系时,很容易在两个显示之间进行切换。

[0137] 具体而言,图 20A 描述了一个石灰窖单元的操作员视图 500,该石灰窖单元具有一个石灰和泥浆进料源 502 (502 可以是水流元素),用于将原材料通过泵 504 添加到窖干燥器 508 的冷端 512。运输件 510 从干燥器 508 的热端 512 将加工后的石灰移除,然后将石灰沉积罐 514 中。产品线或水流元素 516 指出了退出罐 514 的产品(石灰)量。同样,通过阀门 522 可以提供来自燃料源流量指示器 520 的燃料,所述阀门的颜色可以以图形形式描述阀门的操作,然后将所述燃料添加到窖干燥器 508 的热端 512。风扇 524 通过风道连接器元素 526 将空气压入窖干燥器 508 的热端 512,在窖干燥器 508 将空气与燃料混合在一起。诸如火或火焰 528 这样的动画可以显示在窖干燥器 508 的热端来说明窖干燥器 508 的运行状态。同样,感应风扇 530 通过附加风道 532 从窖干燥器 508 的冷端 506 抽出空气,然后将所抽空气送到一个堆栈,该堆栈如堆栈流元素 534 所示。而且,在此描述了各种变量或参数框来显示各种过程参数值,例如系统各部分的温度、窖干燥器 508 的桶速度、空气和燃料流量,等等。可以看到,所述硬件和变量框在图 20A-20E 所示的屏幕上通用的,并提供了显示的基本元素,这些基本元素使得这些显示具有相同的外观和感觉。

[0138] 然而,图 20A-20E 所示的每个屏幕都包括附加信息、用户界面按钮和工厂内配合不同功能的动作。例如,图 20A 所示的屏幕 500 是一个操作员视图,并在框 540 中描述了一个实验室测验所测量、估算和提供的残留碳酸盐,在框 542 中描述了窖的总能量和具体能量,以及在趋势图 544 中描述了一个模型预示控制程序提供的过去和未来的温度趋势,所述模型预示控制程序是获得最优操作的关键所在。另外,屏幕 500 提供各种用户界面按钮或动态图标 548,其允许用户查看相关项目的附加信息,例如控制环信息、元素的深入操作信息等等。

[0139] 图 20B 所示的屏幕 550 描述了一个非常类似于图 20A 所示的工程视图,但是图 20B 所示的工程视图通过虚线 552 示出了对于控制和信号路径很重要的测量位置,从而显示基本控制系统和模型预示控制系统怎样共同工作来使操作最优化。图 20C 所示的屏幕 560 提供了一个管理视图,该视图在显示窖单元基本操作的同时,并不提供允许在系统的个体部件或控制环上得到附加信息的用户界面按钮。而且,管理视图 560 包括显示总能量、具体能量和残留碳酸盐(未转换的进料)的图表 562,所述显示内容来自于前一天或上个月的试验测量。例如可以从业务计算机中运行的业务应用程序来得到这些信息。

[0140] 而且,图 20D 提供了一个仿真视图 570,该视图允许仿真员改变被仿真系统内的参数以及查看仿真结果。视图 570 可用于,例如,培训操作员、测试不同未来操作模式等等。如图 20D 所示,仿真员可以通过一个或多个对话框 570 来改变参数,所述对话框通过屏幕 570 的一个或多个用户界面按钮提供给仿真操作员。过程仿真视图 570 可用于离线状况来进行培训或用于在线状况来提供附加信息,其对于检测将来的问题很有用。过程仿真可以产生于图形,因为设备类型和设备连接是已知的,这是在构造操作员图形过程中使用智能对象或具有仿真能力图形元素的结果。美国公开 No. 2004/0153804 对所述智能对象进行了详细描述。

[0141] 同样,图 20E 描述了一个维护视图 580,该视图提供了窖单元内设备的状态信息。在视图 580 中,可以使用设备健康指示 582 和 584 来显示设备失败。在这种情况下,指示 582 和 584(二者并未都标注在图 20E 中)是半圆,所具有的填充颜色表示或说明设备被监视的当前健康状态。当然,可以从维护或诊断应用程序来提供所述指示 582 和 584。而且,可以理解的是,当检测到一个过程问题时,维护技工可以使用屏幕 580 来滚读找到修复问题的推荐过程。例如,堵塞的泥浆过滤器有必要进行清洗,这就需要切断进料并将窖设置在停顿模式,从而可以从图 20A 所示的操作员视图 500 来解决问题。也可以采取措施检查进料的质量以防止因过程问题逆流而导致的过多粗沙。因此,如本简单例子所示,不同功能之间的各种互动,例如检测到问题并解决问题,可以简单地通过在不同视图之间进行切换而做到,其中所述不同视图具有相同的外观和感觉,从而很容易操作。当然,如果希望,可以将图 20A-20E 所示各种屏幕的信息加到显示器中,并可以仅基于屏幕的用户来显示。而且,所示每个元素或智能对象的信息可以随着用户或用户身份的改变而改变。

[0142] 而且,附加显示可以提供给并联系到图 20A-20E 的显示。这些显示可以包括,例如,诸如 21A 和 21B 所示的控制配置显示。图 21 所示的控制显示 585 说明了各种控制信号被送到控制器和从该控制器送至如图 20A-20E 所示的窖单元的各种硬件元素的方式,以及与这些控制元素关联的标签或变量名称。图 21B 所示的控制显示 590 描述了一个控制模块,该模块表示实现控制图 20A-20E 所示窖单元的控制例行程序。没有明确示出的是,图 21A 所示的控制显示 585 可以使用与图 20A-20E 相同的元素(具有与这些元素关联的相同的或不同的视像)而形成,或者使用不同元素。同样,由于图 21A 和 21B 所示的显示 585 和 590 可以使用显示编辑器 50 形成,这些显示可以被提供以上述任何图形和动画能力。而且,可以通过例如图 20A 所示的操作员视图 500 并通过操作员视图上所提供的用户界面按钮 584 来访问控制显示 585 和 590,从而允许操作员容易地获得到当前控制的接入,所述当前控制从操作员视图 500 来建立。

[0143] 因此,从上述讨论可以理解,各种有关显示可以由编辑器 50 来创建,而且这些显示可以用多种方式来层叠。具体地,为了提供有关或相似的操作符、业务、仿真、维护和工程视图,这些显示可以如上所述而被层叠,而且这些视图可以容易地相互访问。而且,不同显示可以以某种方式被层叠或连接在一起,所述方式反映了工厂的特定分层、逻辑或物理结构。因此,例如,可以为工厂的不同物理或地理区域制作显示。所以,可以创建单个显示来说明工厂的主要区域,并可创建附加显示来表示工厂的每个区域中诸如基于单元接单元(unit by unit)的基本结构,同时可以为每个单元创建进一步的显示。这种方式下,用户可以通过向下钻取显示以得到与工厂越来越小的部分相关的越来越多的细节。举另一个实

例,可以通过用户界面按钮将针对工厂的不同部件或部分的显示连接或约束在一起,从而操作员可以容易地沿正向和反向方向滚读不同的显示,从而以逻辑方式查看工厂内不同的然而却物理连接在一起的部分或部件。

[0144] 图 22 所示的示意图更加详细地描述了这些类型显示的层叠。具体而言,通用或整体工厂概观显示 600 可以提供一种显示,该显示描述了整个工厂的基本或高级别结构,虽然该显示不提供关于工厂内任何特定部件或部分的细节。从显示 600,用户可以选择(使用面板按钮)或向下钻取工厂的任何特定部分或区域,从而提出工厂内区域 A 的一个或多个显示 602 或提出区域 B 的一个或多个显示 604。如图 22 所示,区域 A 可以具有与之关联的  $n$  个个体显示,所有这  $n$  个个体显示互相逻辑串联连接,以反映工厂区域 A 或与工厂区域 A 关联的其他某个逻辑结构的过程流程。可以使用向前翻页或向后翻页类型的动作来访问区域 A 的所述  $n$  个显示,在此标记为 602a、602b、...、602n,在所述类型的动作中,用户可以从一个显示滚动到下一个。在这种方式下,用户可以通过显示容易地从区域 A 的一个部分滚读到区域 A 的另一个部分,所用方式对于操作员来说比较合理。而且,如图 21 所示,操作员可以获得来自显示 602a 到 602n 的多个预定义信息或其他显示信息。因此,当查看区域 A 的显示 602A 时,操作员能够获得进一步的显示 610,其描述显示 602a 中关键参数预定义趋势。同样,当查看显示 602b 时,操作员能够访问列表或文档 612,其具有启动或关闭过程信息。当然,用户可以正向或方向地在显示 602a、602b 等之间滚动。

[0145] 同样,当查看包含  $m$  个显示 604a 到 604m 的区域 B 时,用户可以使用显示内的正向和反向(下一个和返回)按钮在同一级别内容的显示之间滚动,从而操作员或其他用户可以访问包含过程显示部分的上游或下游信息的显示。另外,可以提供动态图标或其他用户界面按钮以访问另一个显示,从而获得与当前显示内元素相关的其他信息。通过使用这些工具,可以创建显示层,从该显示层可以使用概观显示来访问每个过程区域的关键显示。

[0146] 另外,如图 20A-20E 所示,显示可以被层叠从而使多个显示针对工厂的同一部位或部分而存在,但是这些显示用于不同的功能用途,例如用于操作员动作、维护动作、仿真动作、业务动作、工程动作等等。如图 22 所示的这些独立的功能显示被层叠在显示 602a、602b 等的下方,并且需要的话可以互相访问。因此,当操作员或用户正在查看显示 602a 时,用户可以在针对工厂某部分的其他功能显示之间切换,或者如果允许访问的话,可以访问所述显示,例如维护视图、仿真视图、业务视图等。当然,使用相同的基本显示元素来层叠维护视图、操作员视图、仿真视图等,可以提供这些视图之间的简便切换,并且针对工厂内的不同功能,可以提供对工厂内正在发生情况的更好的理解。

[0147] 为了实现该功能,所创建的图形显示可以包含针对图形显示的计划中任务(或功能)的指示。所述任务可以包括,例如,面板显示、细节显示、主控制显示、原理显示、维护显示、业务显示、仿真显示或任何其他用户定义的任务。所述任务是图形显示的一部分,并且当图形显示被分配到控制模块或硬件设备时,所述任务可以用于定义运行时间期间内对显示的使用和访问。向图形显示所分配任务的指示可用于指出:某个特定显示可以被分配到配置系统或工厂的哪里或哪个部分。另外,这个任务信息可用于:基于某个特定用户正在进行的工作来确定可以访问哪些显示。例如,控制操作员不能查看或访问定义为仿真或业务显示的图形显示。

[0148] 上述图形显示能力也可以用于创建特殊显示,从而能够容易地对关键设备的状态

进行监视。这些类型显示的一些实例包括过程关闭、振动监视、火炉管理、吹火器操作和安全系统状态的先出指示。当然，相关联的显示可以被构造从而使得其能总结信息，并且在诸如吹火器的移动设备包含动画例如的情况下，可以有效地将动画用于允许操作员快速访问或理解系统的操作。另外，大多数控制系统的计算能力可用于实现在线操作费用计算、效率计算等，而且这种信息可以很容易地合并到操作员的图形显示中，从而操作员可以使用该信息来改善过程操作。另外，可以使用多种技术将控制系统的子系统信息集成，这样，标准图形和动态图标可以用于创建操作员显示，以允许操作员访问来自高级别显示的子系统信息。在某些情况下，可以实现矩阵数值的三维绘图来显示信息（例如板规信息（sheet gauging information））。

[0149] 另外，需要注意的是，可以使用向量图形来方便地实现图形元素和显示，例如微软 Avalon 控制提供的那些向量图形，从而提供灵活性和高速性。向量图形的使用使得用户可以使用可升级的图形，从而使得用户能够创建可缩放和改变尺寸的显示，以配合某个特定显示及提供使用缩放功能的动画。

[0150] 一般来说，如标题为“每个过程图形显示的创建（the creation of each process graphic display）”的同样待审的申请的详细描述，其中所包含图形显示被记录在相应的以第一说明格式阐述的文本描述中。每个描述中的脚本命令提供一种有效的、非记忆性集中机制，用于定义显示，而不论待汇报的图形有多复杂。说明格式和脚本命令可以基于多个不同标记语言中的任何一个。具体而言，可以依赖基于 XML 的标记语言来为每个显示和显示元素提出一个描绘定义（也叫做 XML blob），且这个 XML blob 可以用于在这些显示和元素被下载到过程运行时间环境之前，将所述显示和元素保存在一个配置库或数据库中。为了支持高级图形，例如动画，标记语言也可以定义针对向量图形方案的图形。

[0151] 如上所述，特别地如共同待审的、通过参考并入的、标题为“加工厂用户界面中的基于标记语言的动态过程图形（Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface）”的申请所述的，过程图形的动态特性被设计来反映当前的诸如在线或仿真条件变化的加工厂参数值。至此，过程图形可以被链接到反映这种变化的数据源。每个基于 XML 的描述可以相应地包括一个或多个数据源引用，所述数据源引用一般为每个动态图标参数（例如罐内部变化颜色）确定一个相应的数据源位置，所述动态图标参数将根据数据被修改。数据源位置也可以保留为未定，以便于在通过编辑器配置期间进行后续指定，这样，脚本识别别名或占位符，以便参考今后待指定的数据源或路径信息。由于过程图形显示的数据源信息和其它特点（诸如事件处理的行为）通过基于 XML 的描述得到详细说明，因此基于 XML 的语言可以被称为 PGXML，或者叫过程图形 XML。

[0152] 完成对过程图形显示及其组成元素进行定义的配置和设计工作后，配置工程师或其他用户可以选择处理 PGXML 描述，为将过程图形下载到操作员工作站或其他用户显示设备做准备。一般来说，图形显示和显示元素的每个 PGXML 描述都被处理来生成 (i) 向量图形格式的脚本，该脚本与将要使用的图形呈现引擎互相兼容，和 (ii) 含指令的代码，该指令可以详细描述数据源引用和显示的其它任何非图形功能（例如行为）。脚本的向量图形格式也可以用于说明，或者是基于 XML 的语言。在使用微软 Avalon 用户界面结构的实施例中，可以使用微软 XAML 来阐明向量图形脚本。其他实施例可以使用开源格式，可缩放向量图形（Scalable Vector Graphics, SVG）。可以使用 C# 语言或任何其他合适的编程语言

来阐述所述代码。

[0153] 在某些实施例中, 向量图形脚本和相关代码随后被结合并被编译成文件, 所述文件为操作员工作站或其他用户显示设备阐明了可执行命令。至此, 可以为每个过程图形显示和图形显示元素创建各自的动态链接库 (Dynamic LinkLibrary, DLL) 文件。在任何情况下, 可以在下载之前对所述向量图形脚本和相关代码进行这样的编译, 以将网络数据传输的要求最小化。

[0154] 一旦创建, 图形元素和图形显示可以作为通用或模板对象被存储在一个配置数据库中, 也可以在用于运行时间环境之前作为基于类的或基于非类的对象或元素被存储。一般来说, 在此所讨论的图形元素和图形显示可以是显示模块形式的模块, 如在此通过参考并入的美国公开 No. 2004/0199925 的所述。类模块 (对象) 是一个没有绑定到或约束到加工工厂内或过程控制系统内任意特定硬件或设备的模块, 但是该对象可以是这样一个对象, 其他绑定到加工工厂或过程控制系统的对象由该对象实例化。一般来说, 要配置一个加工工厂, 尤其是配置加工工厂的运行时间环境, 可以使用诸如图 1 中配置工作站 20 所示的配置应用程序 33 其中之一的配置引擎, 来将图形对象 (包括图形元素和图形显示) 与加工工厂内的其他逻辑和物理实体联系在一起, 所述其他逻辑和物理实体包括控制模块、设备模块 (例如单元模块)、过程模块等这样的逻辑实体, 或者操作员工作站或其它显示设备这样的物理实体。在某些情况下, 图形对象可以是类对象, 并可以与其他类对象联系在一起, 例如设备模块类对象或控制模块类对象, 或所述图形对象可以是个体对象并可以与实例化对象联系在一起, 所述实例化对象, 例如是已经被绑定到加工工厂内设备并被下载的对象, 例如, 加工工厂内的控制器、工作站或其他设备。

[0155] 因此, 上面提到的图形元素和图形显示可以在加工工厂内被配置, 并可以被下载到加工工厂运行时间环境内的硬件上, 所用方式类似于如美国公开 No. 2004/0199925 所描述的方式, 即其他类对象被存储在库当中, 然后被下载到过程控制系统的各个部分或子部分。尤其是, 在诸如图 1 所示的数据库 28 的配置数据库中操作数据的配置引擎, 使用诸如图 23 和 24 所示的配置数据库分层来使用户能够查看和改变加工工厂的配置。图 23 所示的配置分层屏幕 700 显示了, 针对控制系统 702 (标记为 DeltaV 控制系统) 的配置数据库包含库部分 704、系统配置部分 706 和搜索结果部分 708。

[0156] 库部分 704 包括类对象和其他对象定义, 所述类对象和其他对象定义尚未被分配到或下载到运行时间环境, 但是被作为模板或通用和未绑定的对象存储。如图 23 所示, 库部分 704 包括控制模块部分 710、设备部分 712、过程模块部分 714、操作员界面部分 716 以及批量部分和安全指令系统部分 (没有数字标示)。

[0157] 由于控制模块部分 710 没有被扩展, 配置数据库的这一部分一般包括不同的控制模块, 例如控制模块模板和控制模块类对象, 所述控制模块模板和控制模块类对象已经为加工工厂创建而被使用在加工工厂内的各种控制器和其他控制设备中。而且, 设备部分 712 包括设备对象说明, 所述说明可以包括对用于加工工厂内的实际设备和设备类型的说明, 例如对阀门、传感器、控制器、工作站等的说明, 还可以包括对设备类对象、装备类对象等的说明, 所述说明定义了与加工工厂内不同设备或硬件关联的逻辑元素。而且, 过程模块部分 714 包括为加工工厂创建的各种过程模块, 包括过程模块模板和为这些过程模块定义的过程模块定义。美国公开 No. 2004/0199925 对这样的过程模块进行了详细描述, 在此不再赘述。

[0158] 然而,如图 23 所示,配置分层 700 的操作员界面部分 716 存储与上述图形对象关联的信息。尤其是,图像部分 720 包括各种合成过程图形定义、图形类对象和图形模板,所述包含的内容定义了已经为系统创建的各种图形元素和图形显示。可以看到,合成过程图形定义部分通常包括为系统创建的图形元素,过程图形类部分包括以上述方式创建作为类对象的图形显示,而过程图形模板可以包括基于非类的图形显示和其他模板对象。存储在配置数据库中的图形对象通常是非绑定的对象,所述非绑定的对象可以用于创建其他对象,并可以在配置过程期间用于定义图形元素和图形显示被绑定到加工厂内各种硬件和软件的方式。而且,在操作员界面部分 716 下方的版面设计部分 722 针对操作员界面定义了各种界面设置,而角色部分 724 定义了可以在操作者界面执行的各种角色,还定义了由各种图形显示和元素完成的角色。

[0159] 因此,由配置分层 700 描述的配置数据库可以包括一个存储图形对象的部分,所述图形对象包括作为通用模板、类对象或特定 / 个体元素或显示的图形元素和图形显示,其虽然没有被绑定到具体的过程实体,但是可以用于定义将要绑定到具体过程实体的元素或显示。一般来说,配置工程师或其他用户可以使用分层 700 来以某种方式配置一个加工厂,从而具有图形显示能力,所述方式将在下文进行详细讨论。

[0160] 图 24 描述了分层屏幕 730,其中,系统配置部分 706 已经被扩展以说明怎样将过程图形元素和图形显示作为系统配置的一部分联系到加工厂的逻辑和物理实体。尤其是,系统配置部分 706 下方的过程图形部分 734 定义了各种图形显示,所述各种图形显示通常已经被定义来被下载到加工厂在运行时间环境内使用。一般来说,过程图形部分 734 内的图形显示和元素可以包括例如面板显示、操作员显示、细节显示、维护显示、仿真显示等,所述图形显示和元素将被下载到加工厂内的每个显示单元(例如每个工作站或其他显示设备),而作为应用于整个系统配置的过程图形部分 734。然而,图形元素和显示也可以被联系到系统配置的特定部分,包括控制策略部分 736 和物理网络部分 740,二者可以限制这些图形元素和显示可用于的显示设备。

[0161] 一般来说,控制策略部分 736 定义了分配给加工厂的不同物理和逻辑部分的各种控制程序或者针对加工厂的不同物理和逻辑部分所实现的策略。各种图形显示(可以是任何种类的显示,例如面板显示、控制显示、维护显示等)可以被分配到控制策略部分 736 的特定子部分或子类别,包括例如分配到区域(例如名为 Area\_A 742 和 Area\_B 744 的区域)、与区域联系的诸如控制模块部分 746 的控制模块、和分配到区域的诸如过程模块 748 的模块。因此,如图 24 所示,显示部分 750 与 Area\_A 部分 742 联系在一起,所述 Area\_A 部分 742 定义了用于工厂 Area\_A 内的或在其中的所有显示。在没有突发的情况下,文件夹 750 下的显示可以包括与区域 A(Area\_A) 的不同硬件关联的各种显示,例如图 22 所示的不同显示 602a-602n,文件夹 750 下的显示还可以包括从这些显示可以访问的各种子显示,例如图 22 所示的显示 610 和 612。为显示定义的角色可以指出这些显示可以相互访问或者可以向使用显示设备的操作员访问的方式。因此,如图 22 所述,用户可以浏览部分 750 的全部显示以查看工厂内区域 A 的不同部分或者查看针对任何特定区域的更多细节,包括在高级别显示中没有详细示出的高级别显示特定部分的详细显示。

[0162] 而且,显示可以被分配到区域 A 部分 742 的特定控制模块,如控制模块部分 746 下面的显示图标 752 所示。这种情况下,显示 752 与一个控制模块环(称为环(LOOP))联系

在一起,而且显示 752 可以是与该特定控制环联系在一起的控制操作员显示。再次,作为控制显示的一部分,为显示所定义的角色可以说明该显示在操作员工作站中的角色。而且,如区域 B(Asea\_B) 部分 744 所述,显示 754 通常可以和区域联系在一起,而显示 756 可以与分配给区域的特定过程模块联系在一起。在这种情况下,显示图标 756 就和一个叫做 PMOD2 的过程模块联系在一起,所述 PMOD2 是一个与区域 B 内设备联系的过程模块。PMOD2 例如可以是一个单元模块或任何其他类型的过程模块。

[0163] 因此,如控制策略部分 736 下的显示大致所述,显示可以与特定控制策略或控制定义联系在一起,因为这些控制定义是在配置系统内创建的。如图 24 的例子所示,显示可以与控制定义联系在一起,所述控制定义是在区域接区域 (area-by-area) 的基础上定义的,例如基于区域部分 742 和 744,也可以是基于控制模块接控制模块 (control module-by-control module) 的基础上定义的,例如基于控制模块 750,还可以是基于一个过程模块接一个过程模块的基础上定义的,如显示部分 756 所示。另外,如果希望,设备项目可以与角色联系在一起,该角色允许与这些角色相关的显示从设备元素动态访问。可以将这样的设备角色和显示存储在过程图形部分 734 中。

[0164] 另外,也可以将显示分配给或关联到加工工厂内的实际硬件元素,包括操作员工作站或其他显示设备。因此,如物理网络部分 740 下面所述,特定的显示可以被分配到特定的操作员工作站或其他具有显示的运行时间机器。在图 24 所示的特别示例中,物理网络部分 740 包括一个控制网络 760,其中控制网络 760 具有关联的硬件,例如名为 CTRL1 的控制器、可以是一个配置站的 ProPlus 站以及一个叫做操作员站 762 的操作员工作站。操作员站 762 定义了或关联到工厂内的一个特定工作站或用户界面,所述操作员站 762 包括相关的报警和事件功能、操作员活动、连续历史功能和仿真活动,虽然其他活动或功能,例如维护活动、业务活动等,也可以被关联到任何特定用户界面或者在其上被执行。在这种方式下,特定用户界面可以被分配到加工工厂内的角色,并具有针对存储或下载到显示设备上的这些角色的显示。

[0165] 如图 24 所示,显示 764 和 766 被分配到工作站 762 所实施的操作员功能或活动以及仿真功能或活动。没有在图 24 中示出的是,显示还可以被分配到其他显示设备,例如那些与加工工厂内其他节点联系在一起的显示设备,从而在这些显示设备上实现显示。而且,把显示分配到操作员工作站内或其他运行时间显示机器内的特定功能上可以确定这些显示将在运行时间环境内实施的角色或功能,例如操作者功能、仿真功能、维护功能等。当然,可以用配置系统设定的方式,在同一显示设备上或者不同显示设备上实施这些各种功能。同样,虽然配置分层 730 描述了正在分配给某个操作员工作站的显示,所述显示也可以被分配给其他种类的显示设备,包括手持计算机显示设备,例如 PDA、电话设备、商务工作站或其他所需种类的显示设备。

[0166] 配置引擎使得诸如配置工程师的用户可以使用配置分层来定义和管理过程图形显示被绑定到和下载到加工工厂内的方式。一般来说,诸如配置工程师的用户可以创建、删除、重命名、分配和下载配置系统内的显示对象,所用方式与该用户针对其他对象进行操作的方式相同。特别地,配置引擎可以通过定义将这些执行显示和过程实体(逻辑的和物理的)分配到哪里(哪些设备),使得图形显示可以被集成到加工工厂的配置中,而且这些显示在运行时间内被绑定到所述执行显示和逻辑/物理过程实体。如果希望,配置分层屏幕,例

如图 23 和 24 所示的屏幕,可以被用于指示所需下载,并用于实现显示的版本控制。

[0167] 可以在配置系统的多个级别上进行配置。首先,配置工程师可以将库部分 704 内的一个显示类对象和一个模块类对象联系在一起,所述库部分 704 将一个特定显示类对象绑定到一个特定模块类对象,例如装置模块类对象、区域模块类对象等。然后,当模块类对象被实例化并分配到(或通过其他方法配置于)加工厂的某个特定部分时,与实例化的模块类对象联系在一起的显示对象(例如图形显示)被绑定到模块对象所绑定到的同样的硬件,从而没有必要将图形显示分配到工厂内合适的硬件或将该图形显示对象下载到合适的硬件。取而代之的是,在这种情况下,图形显示对象遵循其绑定的模块对象的配置。而且,当显示类对象被联系到一个模块类对象,如上所述,对模块类对象所作的变化可以促使这些变化自动传播到相关显示类对象内的那些相同元素,使所述元素也发生相同变化,然后这些变化可以向下传播到图形显示的实际实例化版本。举一个例子,如果图形显示包括一个混合器(设备类对象),该混合器包含一个叫做环的控制模块,并且该控制模块被选择和重命名,那么配置系统可以自动将图形显示内对该环的引用改变为新的名字,从而保证图形显示的绑定也更新为新的名字。然而,可能仍然需要将实际更改后的新显示下载到运行时间机器内以在运行时间系统中做相应改变。如果希望,可以将下载需求指示放置在配置分层内所保存显示的后面,例如通过在合适的图形显示图标后面放置一个蓝色的三角形。

[0168] 另一方面,可以单独地或直接地将图形显示分配到加工厂或配置系统的不同部件。尤其是,如图 24 所示,可以将个体图形显示放置在系统配置部分 734 下方的过程图形部分 734 中。例如,可以通过在库部分 702 中选择这些显示,并将其拖拽到部分 734 上来将这些显示放置在该部分。当然,也可以使用其他方法把图形显示移动到分层 730 的特定部分。无论如何,部分 734 中的图形显示包括例如面板显示或其他基于类的或基于非类的显示,而且通常可以被下载到工厂内的所有工作站或者工厂内由系统配置部分 706 所涵盖的部分。

[0169] 然而,用户可以将显示移出过程图形部分 734 并移入特定子部分,以控制过程控制系统内工作站(或其他显示设备)的身份,所述工作站将可以访问这些显示,从而控制可以使用这些显示的设备的身份。为了向一个逻辑或物理工厂部分分配某个特定显示,用户可以,例如,选择一个图形显示,将该图形显示拖放到配置分层 730 的特定部分,从而定义图形显示和逻辑/物理实体之间的联系,这样就可以把该显示联系到所述逻辑/物理实体。当这个配置步骤发生时,基于显示被释放到或联系到的配置部分,可以自动填写针对显示的判决表。

[0170] 例如,可以将图形显示从库部分 702 拖动到过程图形部分 706,从而将显示和运行时间环境联系在一起。同样,例如过程图形部分 734 内的图形显示可以被移动到其他部分,例如区域 A 显示部分 750(使该显示成为特别针对区域 A 实体的显示和功能,但是整体上对任何区域 A 显示设备都可用)、区域 B 显示部分 754,等等。同样,可以将图形显示和逻辑实体的子部分联系在一起,例如与名称为“环”(如显示文件夹 752 所示)的控制模块联系在一起,所述环使这些显示特别针对这个逻辑实体,并且只有在实施这个控制环功能或定义为与该环相关联的机器上才可用。另外,可以将显示分配到特定显示设备或显示设备的功能子部分,例如由操作员站 762 内的操作员功能所定义的显示设备,或者操作员站 762 内的仿真功能 770。没有详细说明的是,用户还可以将图形显示分配到一个区域、小区、单元、设



备模块,以及分配到不同的控制部分,从而定义图形显示被联系到一起的方式并且从而被使用于加工厂内。

[0171] 当一个图形显示被分配到一个逻辑实体时,例如分配到区域名称像区域 A 或区域 B 所定义的控制策略,图形显示被分配到物理界面(例如,工作站),其中所述逻辑实体被分配给这些物理界面。类似地,当一个显示被分配到一个单元、小区或设备模块时,显示被分配到所述单元、小区或设备被分配到的工作站或其他界面设备。

[0172] 当模块包含多个图形显示时,该模块可以定义针对这些显示的角色,或者定义每个显示定义待完成的角色,所用方式是在创建显示时为这些图形显示分配角色标识。对任何显示设备内图形显示的访问就可以与为显示定义的角色联系在一起,或者受到为显示所定义的角色限制。而且,如果一个图形显示被分配到某个区域或模块,对该图形显示的引用可以表现为该区域显示文件夹内的一个图形显示或者是该图形显示所归属的模块。一旦将图形显示分配给某个区域或模块,该区域或模块所拥有的显示引用,以及进而该显示,将会在该区域或模块的任何分配或移动过程中跟随所述区域或模块。另一方面,当一个区域或模块被分配到一个工作站时,属于该区域或模块的显示就会自动被分配到显示角色所定义的操作员子系统或该工作站的功能。当然,可以将个体图形显示直接分配给某个显示设备的操作员子系统,所用方法是将个体图形显示直接放置在特定界面设备之内或之下,或者是放在配置分层的物理网络部分 740 内界面设备的子系统之内或之下。

[0173] 因此,可以理解的是,可以将图形显示分配给个体节点、控制区域、控制模块、过程模块或其他逻辑控制实体,也可以分配到个体显示设备和其中的子系统,方法是将图形显示拖拽到配置屏幕 730 内的每个所述实体内。当然,如果需要,可以将同样的图形显示分配到多个站,并且可以用图标来指示每个(下载的或者刚分配的)显示的状态,所述图标是为分层 730 内各种文件夹下的每个显示名称分配的。如果需要,也可以使用分层 730 内的其他图标来指示图形显示何时被锁定在内存中,或指示图形显示是否可以被保留在盘上的虚拟内存中。

[0174] 总的来说,图形显示包括:显示内容(例如 XAML 脚本,该脚本定义了显示以视像形式出现在显示屏上的方法);局部表格和引用,其可以是对其他显示的引用或对运行时间参数的引用,例如对控制参数的引用等,还可以是对显示控制、运行时间别名(有些可能在运行时间内绑定)的引用;以及局部表格引用和全局表格引用。一般来说,配置引擎对图形元素(XAML)的图形部分和引用分开处理,从而可以更加容易地处理和使用图形显示。尤其是,配置系统可以在配置过程中,通过填充不使用动态别名的绑定、基于配置分层内进行的分配操作,绑定一个图形显示。所以,例如,当用户把一个图形显示从过程图形部分 734 或配置层 730 的库部分 704 拖动到某个特定区域或控制模块时,图形内会自动进行绑定操作,从而将图形显示内的引用绑定到与该区域或模块关联的特定元素。在这种方式下,下载绑定可以基于用户实施的配置自动进行,从而可以减少用户手动指定这些绑定的工作量。

[0175] 而且,如果在配置步骤中存在未判定的绑定,配置引擎可以要求用户提供判定这些绑定的信息,例如改变绑定或引用。而且,如果需要,可以使用批量编辑工具,例如 Microsoft Excel 这样的扩展片程序,来帮助配置工程师在配置过程期间填写或定义判决或绑定参数。在任何情况下,这些配置步骤填满图 13 所讨论的判决表 208,其中判决表 208 使图形显示可以在运行时间环境内被使用。因此,用户可以对图形显示(图形元素)进行

绑定和下载,所用方法与美国公开 No. 2004/0199925 中描述的用户针对过程模块所用的方法大体相同。

[0176] 当然,用户或配置工程师可以通过点击图形显示以及将图形显示从分层 730 中移走或者通过预定一个对话框来解除对图形显示的分配,从而解除了对图形显示的分配。而且,用户可以查看分配特定图形显示或某套图形显示被分配的方法,并且这样的信息可以用图 25 所示的对话框 800 来提供。图 25 所示的对话框 800 指示在加工工厂内配置显示 1 和显示 2 中的每一个的方式。尤其是,对话框 800 指示每个显示在哪里被使用(例如,显示 1 在叫做区域 A 的工厂区域内和叫做 Mod1 的控制模块内被使用),还指示每个显示被分配给哪个物理站(例如,显示 1 被分配到叫做 Oper1 的操作员站),以及指示每个显示所完成的角色(例如显示 1 在其每个用途中是主要显示)。

[0177] 可以理解的是,也可以以任何希望的方式改变显示角色,而且所述角色可以用于指示图形显示可以被分配给配置系统的某个部分,或者谁可以使用图形显示,用于何种功能。如果需要,图形显示也可用于实现多个角色。当然,如上所述,当用户将区域或某种模块分配给一个工作站或其他显示设备时,也将与该区域或模块关联的所有显示分配给工作站,以保证在该工作站上为该区域或其他模块提供图形支持。同样,当图形显示被分配到一个工作站或其他显示设备时,可以基于所述图形显示的角色把该图形显示分配给工作站的子系统,所述图形显示的角色例如该图形显示是否一个仿真显示、操作者显示等。另外,用户可以将图形显示拖放到工作站或其他显示设备的特定子系统中,而且如果分配的角色是错误的,配置系统可以呈现一个对话框来告诉用户刚才做出了错误的分配或者允许用户改变图形显示的角色。

[0178] 在使用图 23 和 24 所示的配置引擎和配置分层 730 进行了合适的或期望的分配操作后,用户操作员可以对图形显示执行实际的下载,从而将图形显示实际下载到配置系统定义的合适的操作员工作站内,以实现配置系统定义的目标或角色。在所述下载过程中,基于针对区域、环、控制模块、过程模块等的分配,以及基于针对操作员工作站或其他显示设备的分配,来判定显示内的绑定。

[0179] 如果需要,配置分层的搜索结果部分 708(如图 23 所示)可以使用户能够搜索在整个加工工厂或配置系统内的图形显示内使用的任何个体图形显示或图形元素。尤其是,配置引擎可以包括一个搜索引擎,在被用户使用,所述搜索引擎可以搜索特定图形显示或图形元素,并可以将搜索结果存储在搜索结果部分 708 中,从而为用户提供了一个完整的列表,该列表包括任何特定图形元素或图形显示的使用。对于进行改变操作的用户来说,这样的搜索功能很有用,因为用户为了实现那些改变,可以查看哪些图形显示需要被改变或者哪些下载操作需要被实施。

[0180] 在此所述的软件被实现后,可以存储在任何计算机可读的存储器中,例如磁盘、光盘或,在电脑或处理器的随机存取存储器(RAM)或只读存储器(ROM)中的其他存储媒介,等等。同样,可以使用任何已知的或期望的传递方法将所述软件传递给用户、加工工厂或操作员工作台,例如使用一个计算机可读硬盘或其他可传输计算机存储机制,或者是通过一个通信通道,例如电话线、因特网、万维网、任何其他局域网或广域网,等等(这种传递方式被看作与通过传输存储媒介提供所述软件的方式相同或可以相互交换)。进一步,可以在将该软件通过通信通道进行传输之前直接提供该软件,无须进行调制或编码,也可以使用任何合

适的调制载波和 / 或编码技术对该软件进行调制和 / 或编码。

[0181] 虽然本发明已经参照特定实例进行了描述,而这些特定实例只是旨在例示而并非意在限制本发明,然而对于本领域的普通技术人员可以理解的是,在不偏离本发明的精神和范围的前提下,可以对所公开的实施例进行改造,添加和删减。

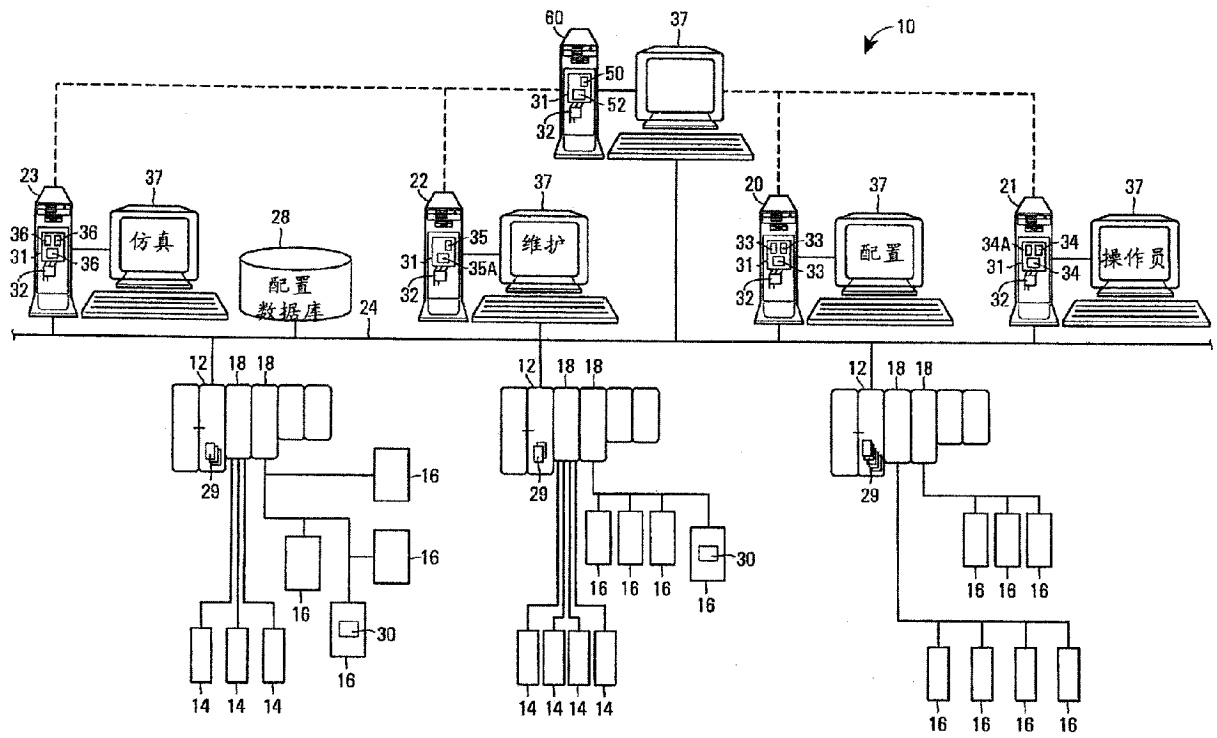


图 1

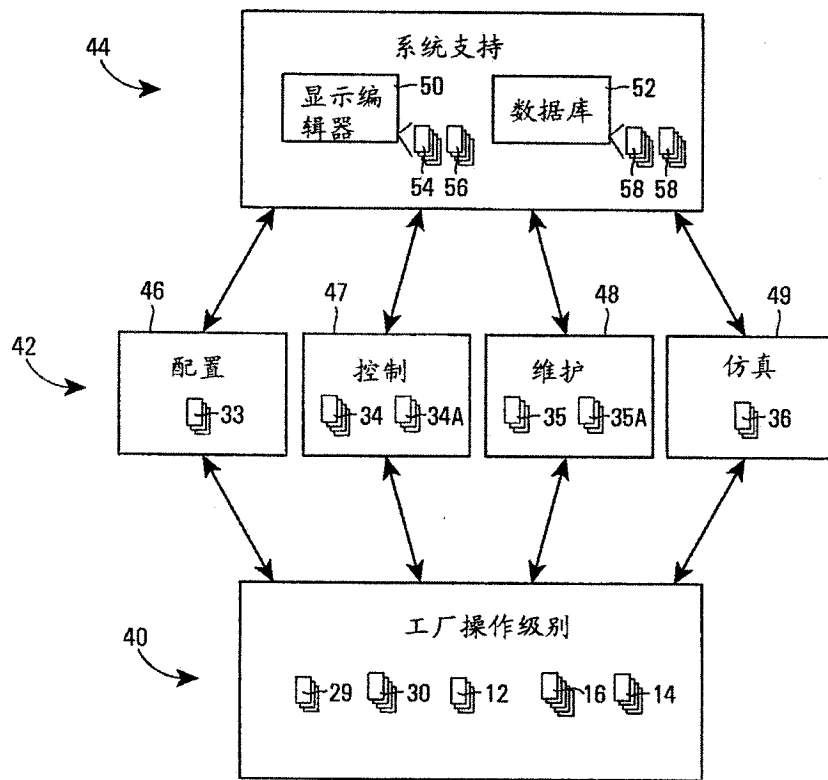
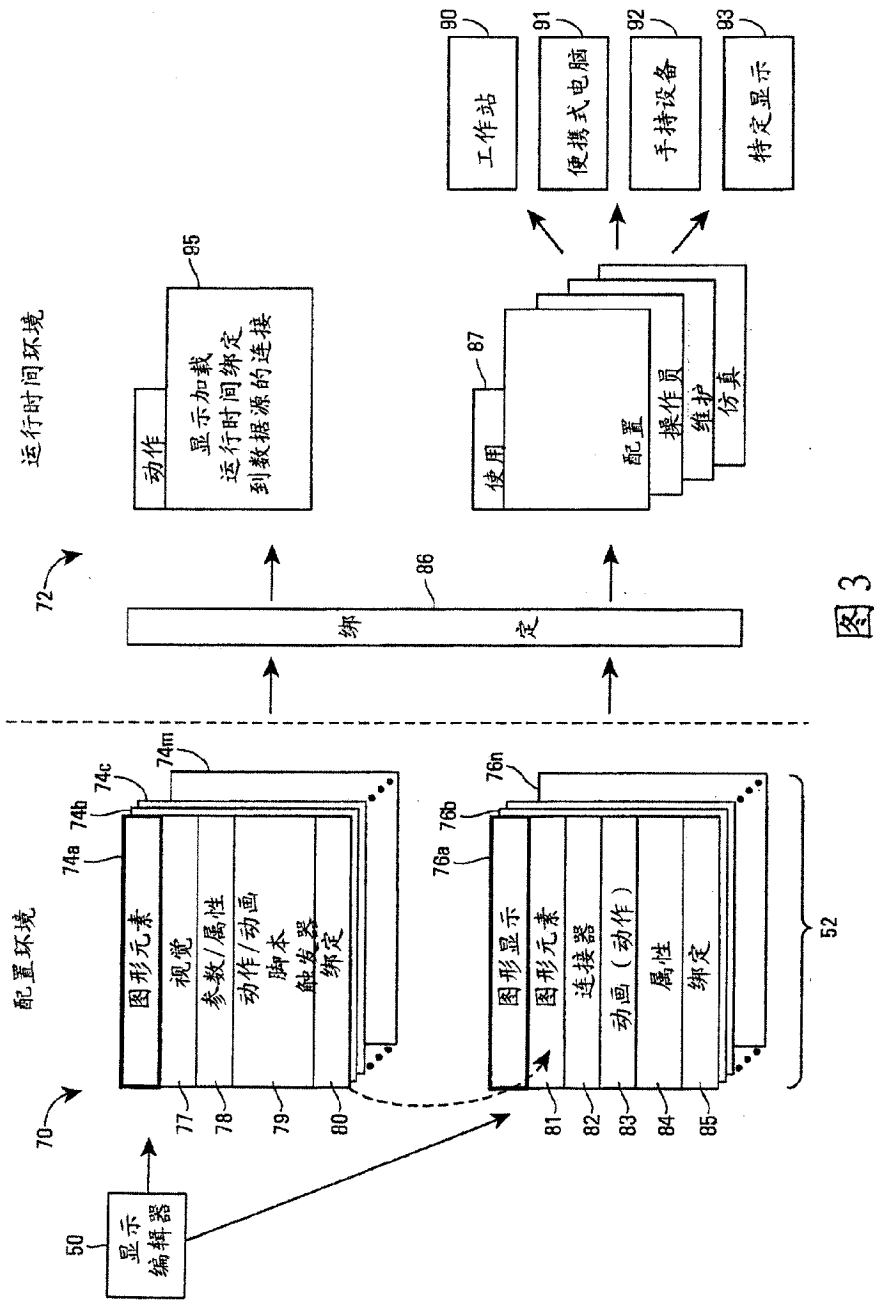


图 2



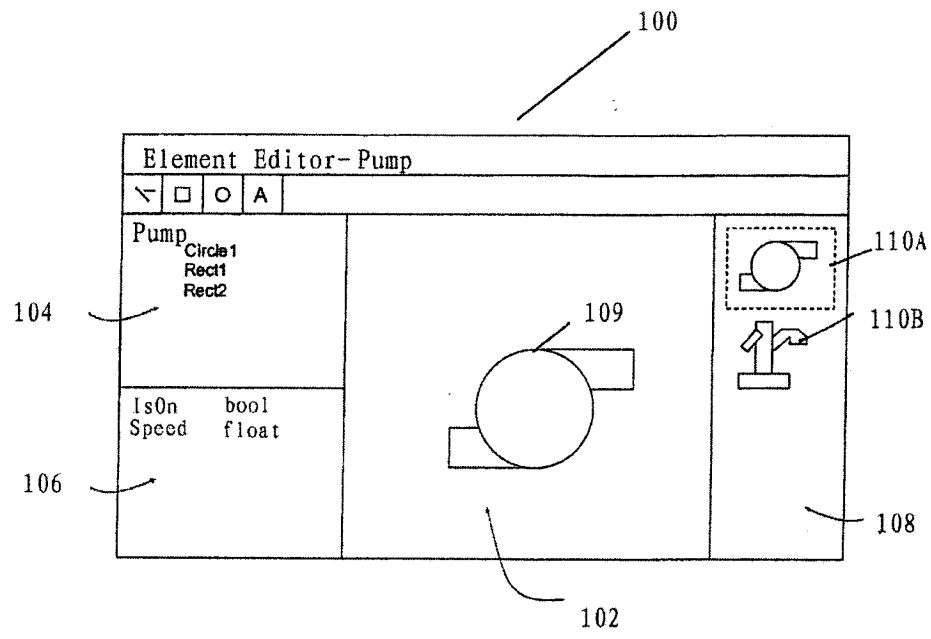


图 4

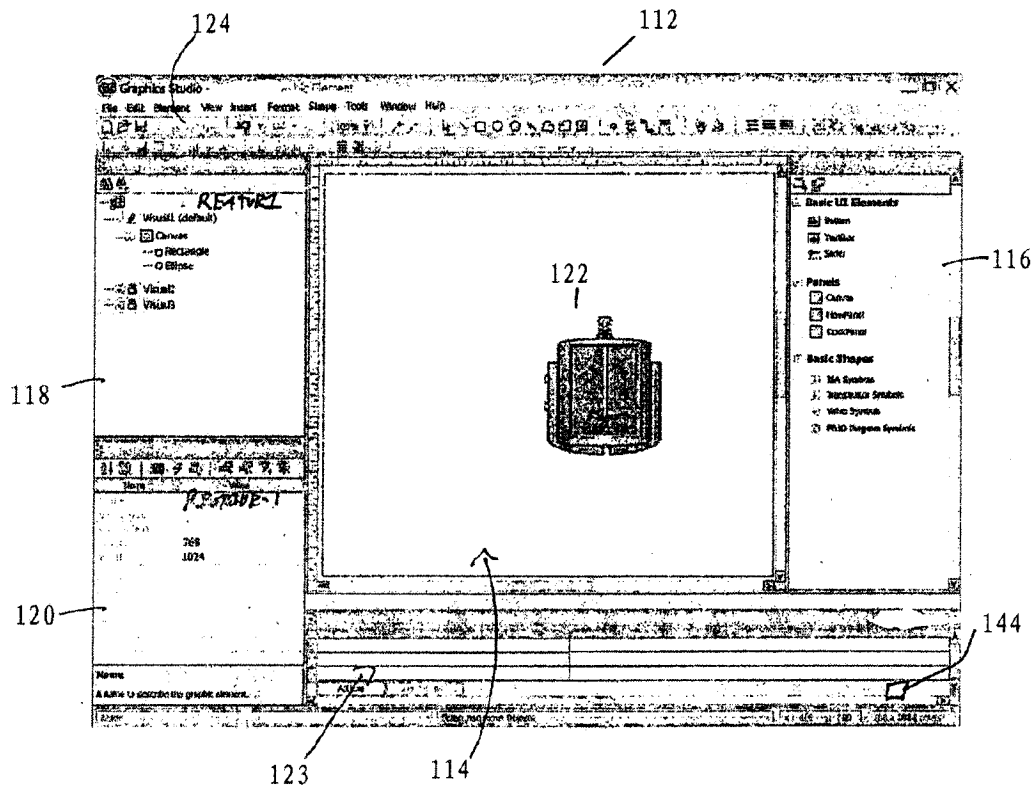


图5



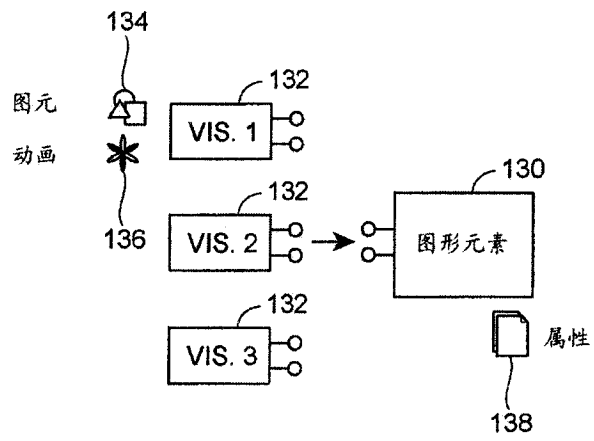


图6

120A

Name	Data Type	Default
IsOn	Bool	True
Speed	String	Fast

**IsOn**  
Indicates whether the pump is on or not.

图 7

120B

Name	Value	Animation	Binding
Name	1		
Background	1	<input type="checkbox"/>	Spe... <input type="checkbox"/>
Description			
Height	768		
Width	1024		

**IsOn**  
Indicates whether the pump is on or not.

图 8

123A

Element Property	Value	Label	Type	Value
IsOn	True	Visual1/Rectangle1	Fill	Red
IsOn	True	Visual1/Ellipse1	Visible	False

144

图9

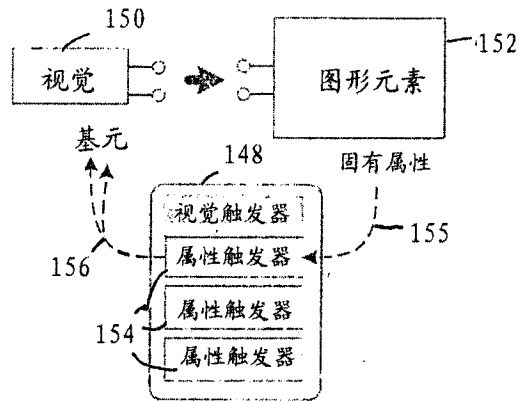


图10

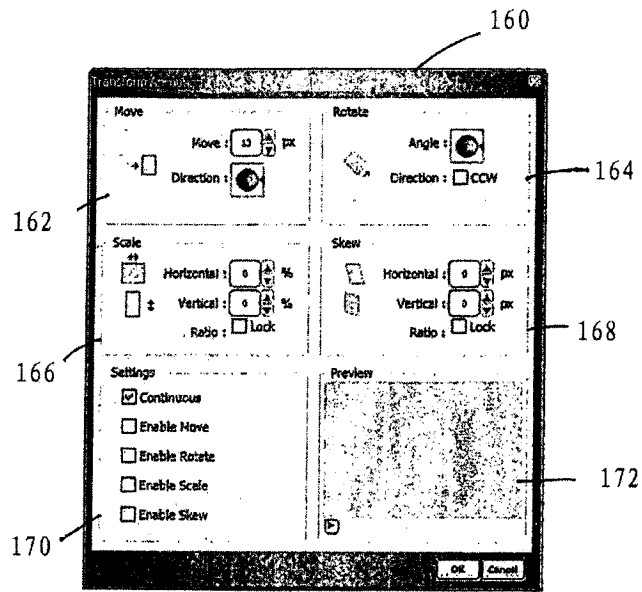


图11

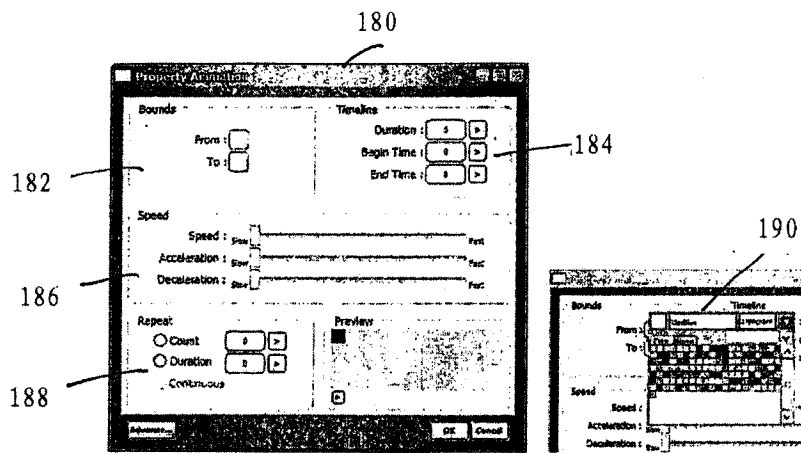


图12

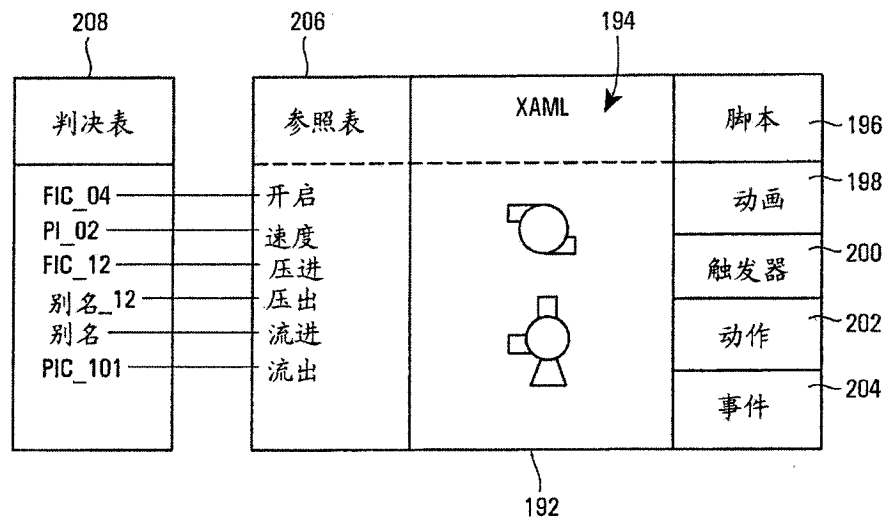


图 13

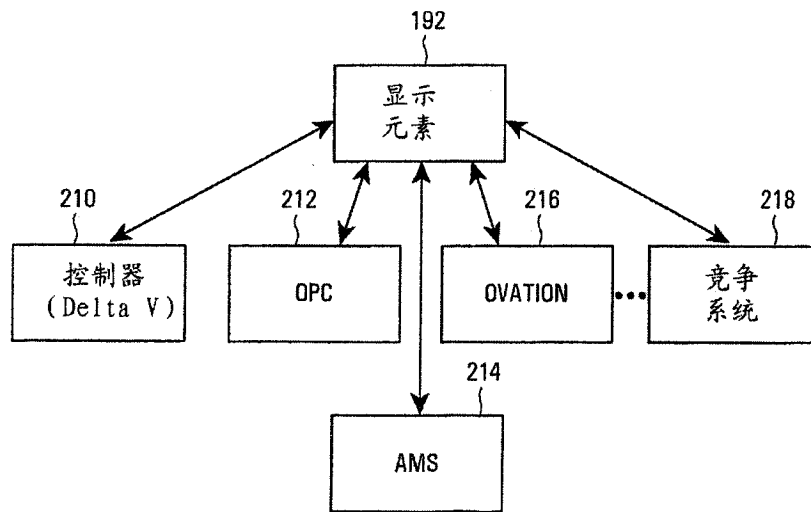


图 14

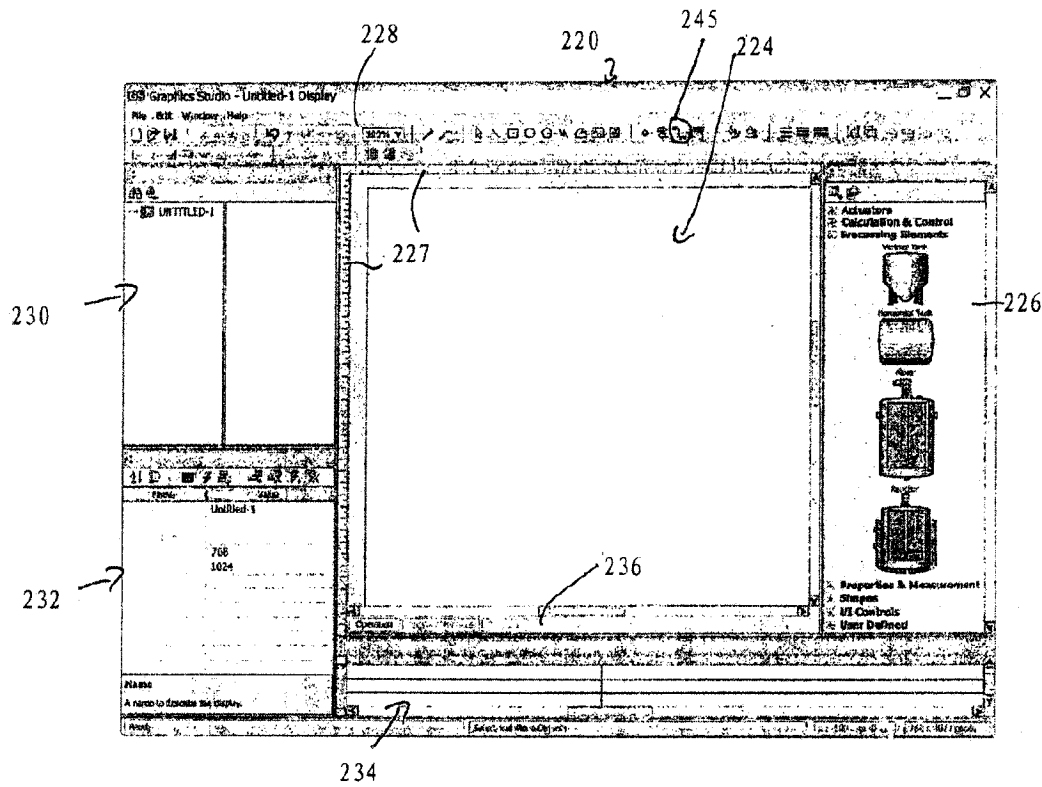


图15

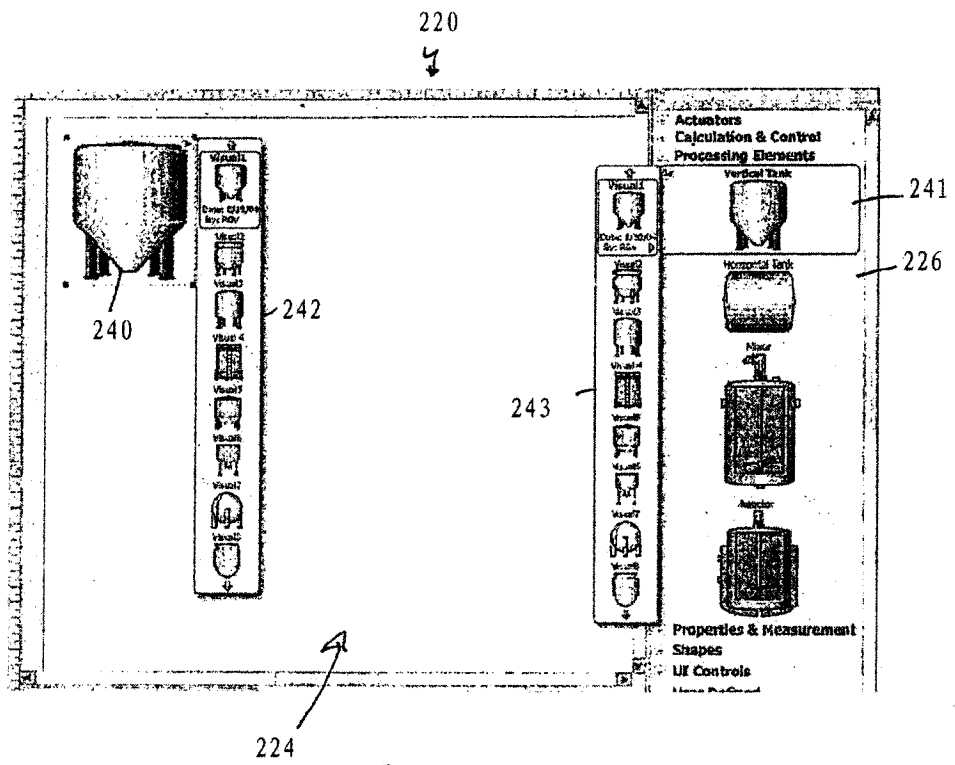


图 16

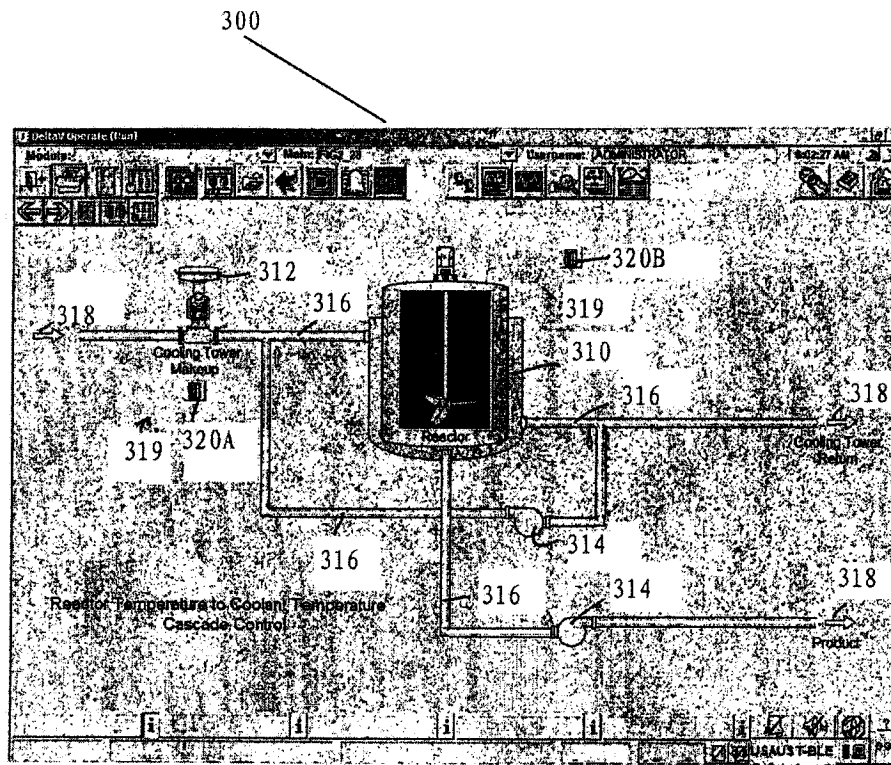


图17





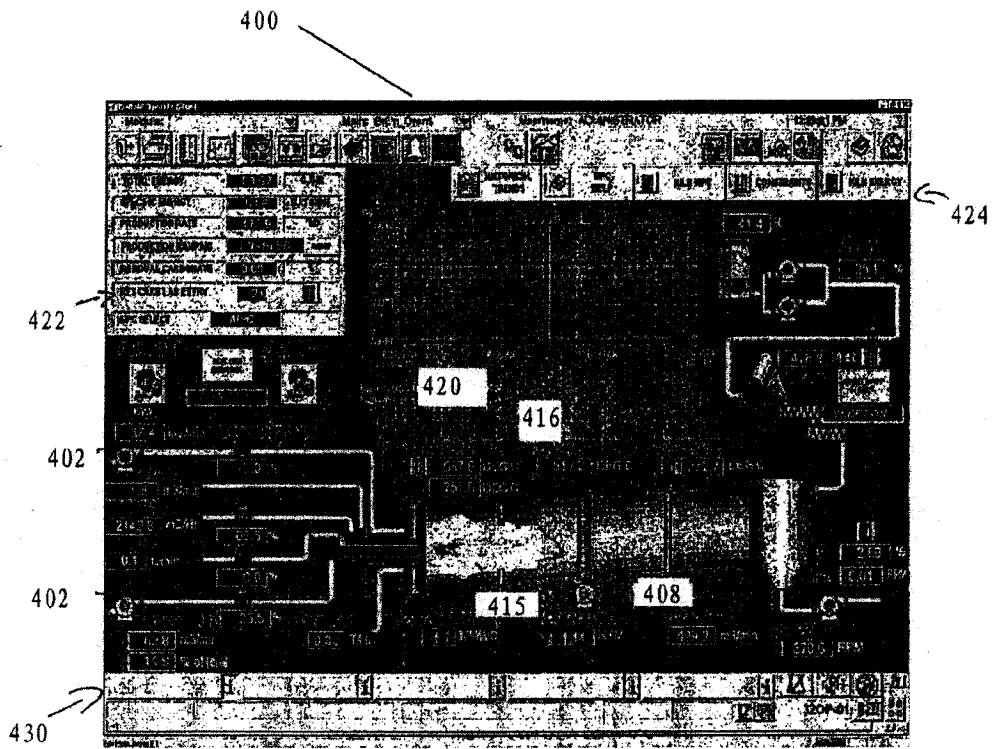


图19



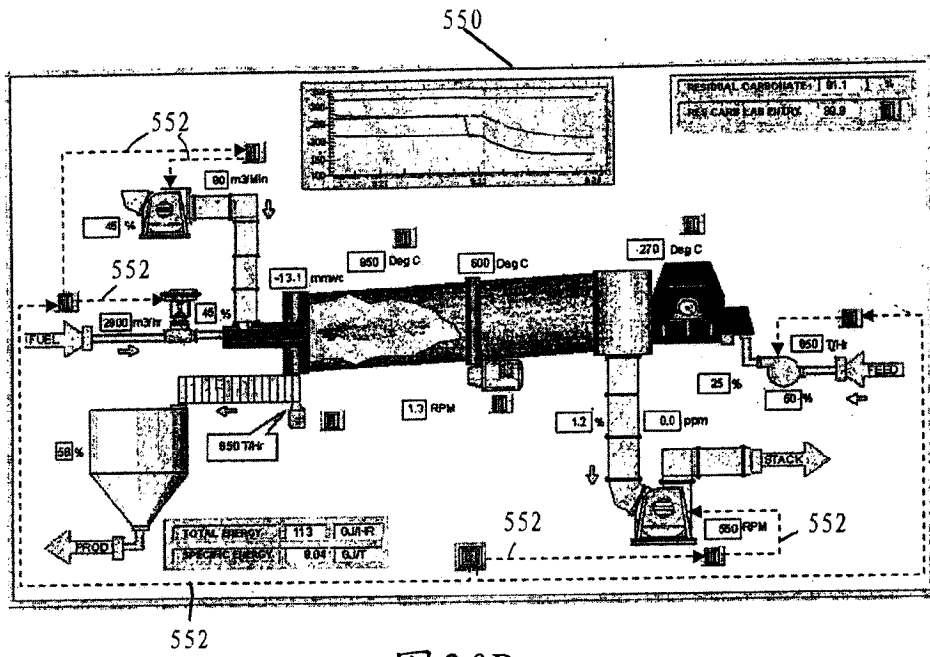


图 20B

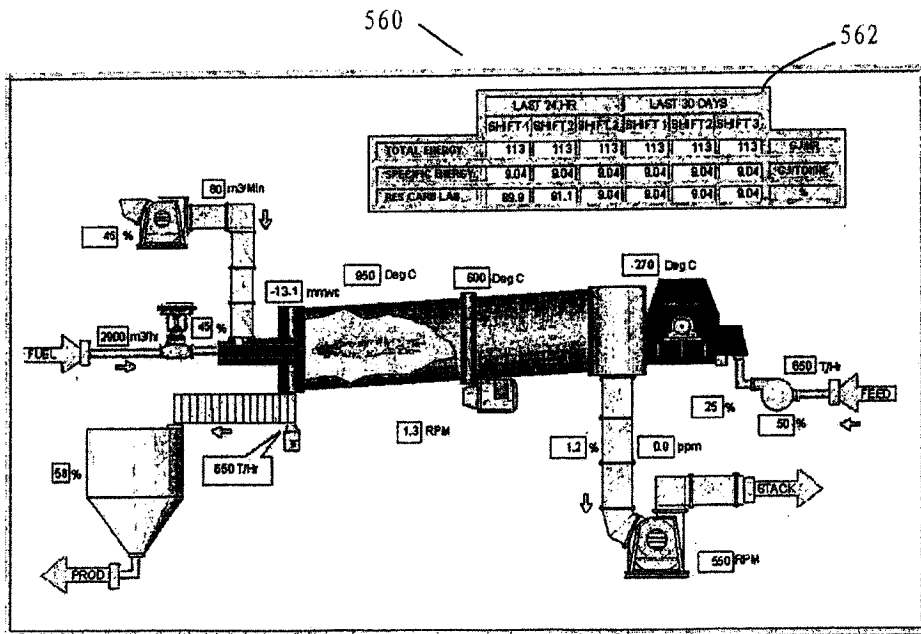


图 20C

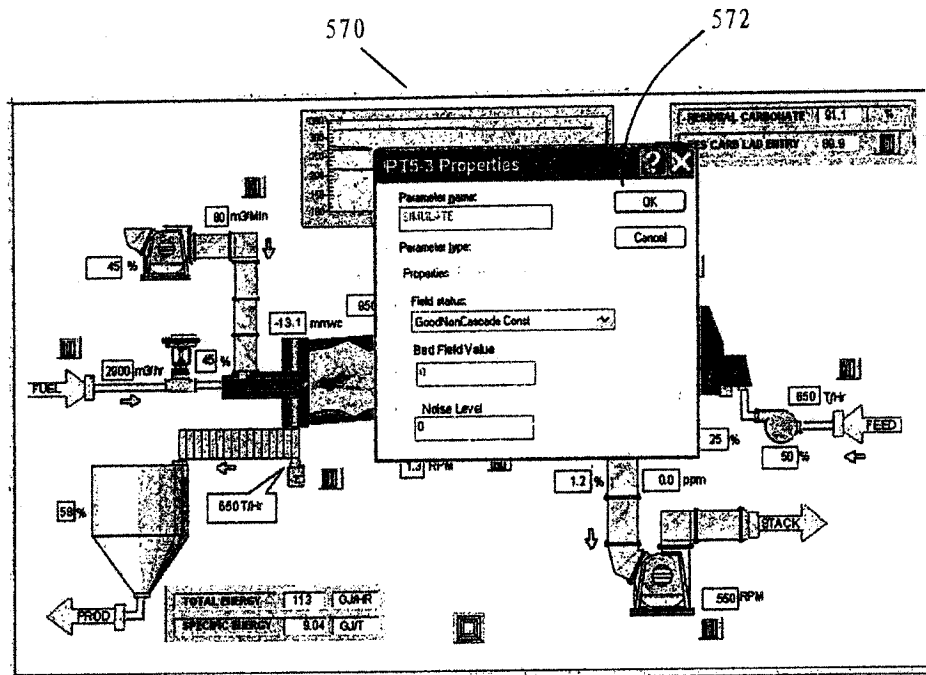


图 20D

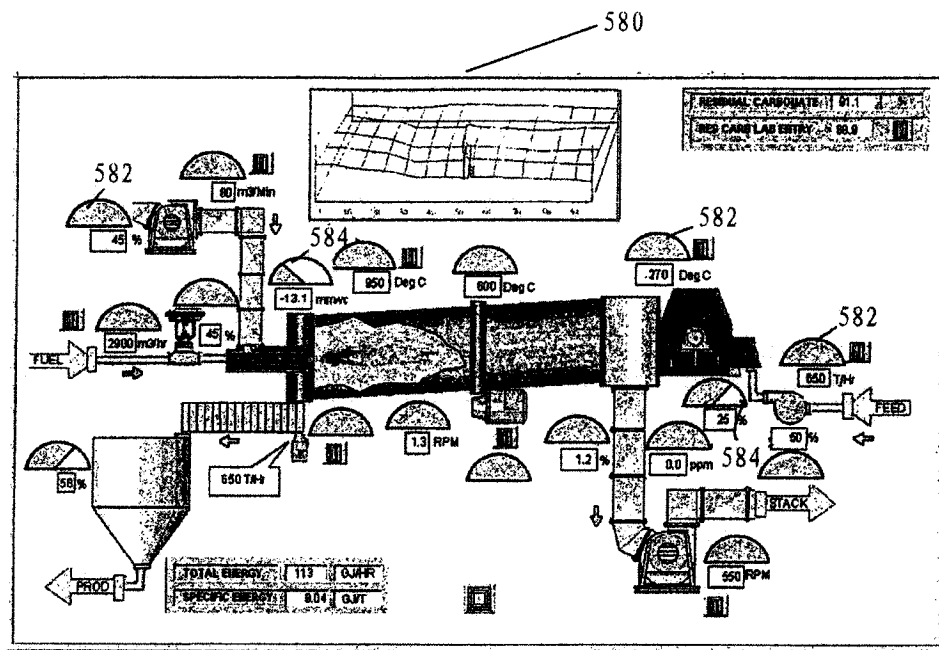


图 20E

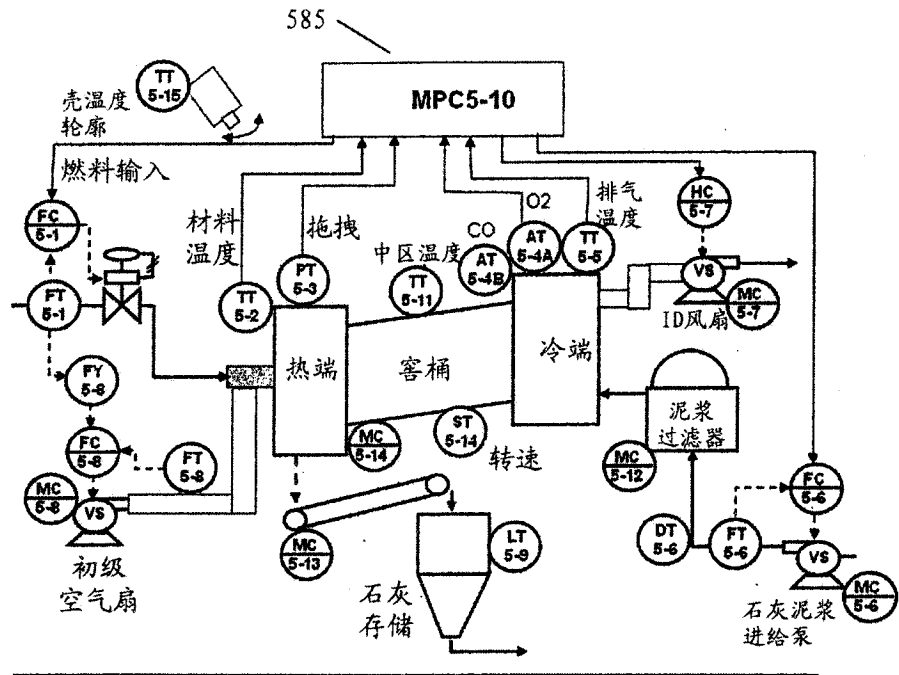


图 21A

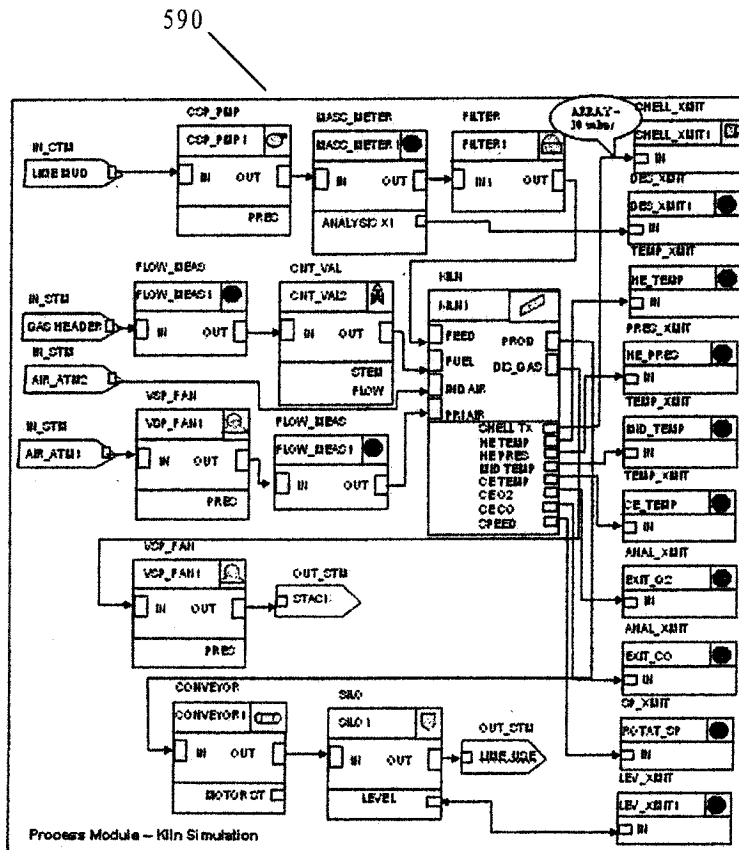


图 21B



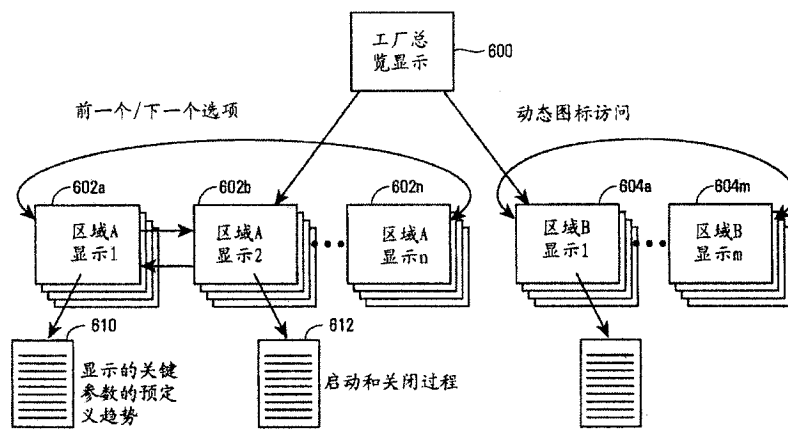


图 22

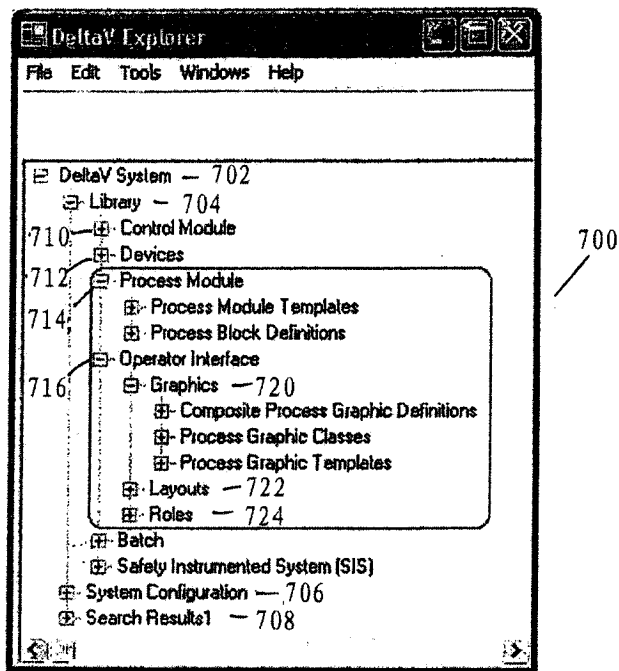


图 23

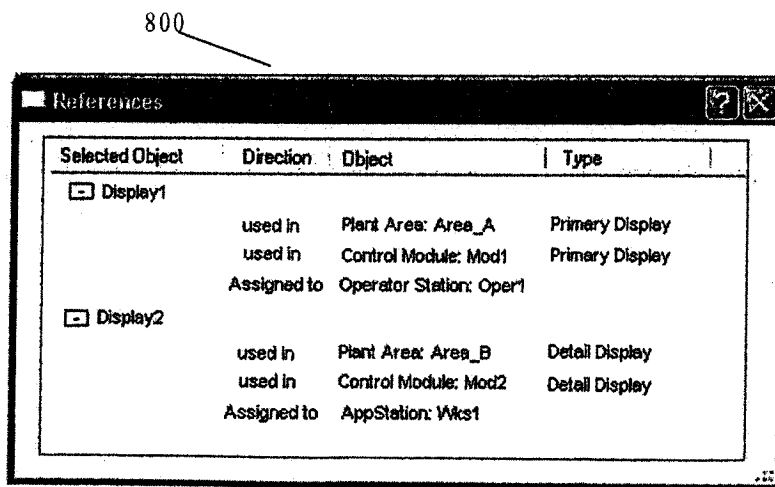


图 25

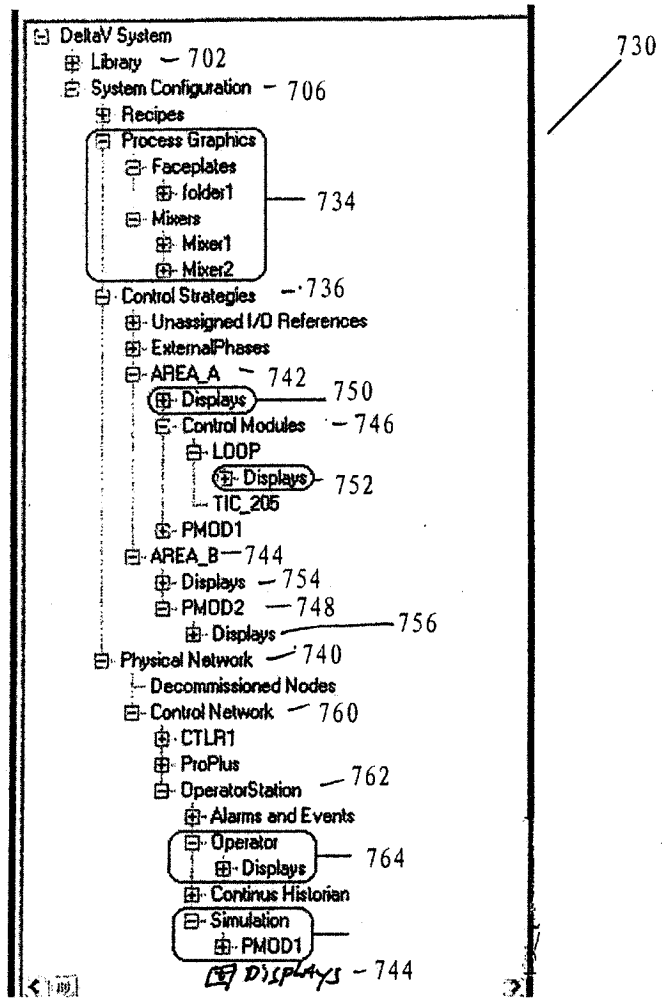


图 24