



(12) 发明专利

(10) 授权公告号 CN 101589357 B

(45) 授权公告日 2012. 08. 08

(21) 申请号 200780045267. 0

(22) 申请日 2007. 10. 12

(30) 优先权数据

11/549801 2006. 10. 16 US

(85) PCT申请进入国家阶段日

2009. 06. 08

(86) PCT申请的申请数据

PCT/US2007/081233 2007. 10. 12

(87) PCT申请的公布数据

W02008/048892 EN 2008. 04. 24

(73) 专利权人 因文西斯系统公司

地址 美国马萨诸塞州

(72) 发明人 J·J·克拉耶夫斯基三世

F·A·G·弗朗索瓦

J·P·麦金太尔 小 J·R·安德森

(74) 专利代理机构 中国国际贸易促进委员会专  
利商标事务所 11038

代理人 李向英

(51) Int. Cl.

G06F 3/00(2006. 01)

(56) 对比文件

US 2003090453 A1, 2003. 05. 15, 全文.

US 5799298 A, 1998. 08. 25, 全文.

US 6330007 B1, 2001. 12. 11, 全文.

审查员 杨华

权利要求书 2 页 说明书 14 页 附图 6 页

(54) 发明名称

用于人机界面的查看状态系统

(57) 摘要

所公开的工业控制与自动化人机界面 (HMI) 技术包括图形元素所用数据质量和状态行为的集中定义。所述集中定义随后应用于链接到对其保持 / 提供状态的数据值的每一个图形元素。在全部 HMI 应用程序上,甚至在系统中 HMI 应用程序的若干组上,对集中配置的数据质量和状态指示行为进行合并,以便通过全局定义的数据状态动画行为通知操作员读 / 写数据的数据质量和 / 或状态。集中定义的行为分布在系统中一切节点上而且合并在活动应用程序中,更新其行为定义时无须关闭这些应用程序。状态图形元素类型。状态图形元素检查某指定的数据变量并显示图片或图标,指明数据的质量或状态。

状态	图标
不合格	
不确定	
初始化	
通信错误	
配置错误	
操作错误	
软件错误	
安全错误	
警告	
挂起	

1. 一种联网的计算机系统,用于便于可分布在所述系统的多个节点上的人机界面应用内的图形元素的数据质量和状态行为的配置和传播的集中管理,所述联网的计算机系统包括:

包括配置数据库的第一计算机节点,所述配置数据库在中心配置存储中保持全局数据质量和状态行为定义;

包括具有用户界面的定义编辑器的第二计算机节点,所述定义编辑器包括用于支持所述全局数据质量和状态行为定义中所定义的数据质量和状态行为的指定的装置;以及

其中所述第一计算机节点还包括用于定义分布机制的装置,所述定义分布机制用于将对所述全局数据质量和状态行为定义的改变传播到网络上执行人机界面应用的一组工作站,其中所述人机界面应用包括与具有相关联的质量和状态的数据相关联的图形。

2. 根据权利要求1所述的联网的计算机系统,进一步包括状态图形元素数据质量和状态显示特性,其中所述定义编辑器包括用于为指定的数据源配置状态图形元素的装置。

3. 根据权利要求2所述的联网的计算机系统,其中所述状态图形元素包括对应于一组所支持的数据质量和状态的一组图标。

4. 根据权利要求1所述的联网的计算机系统,其中,对全局数据质量和状态行为定义的改变被在没有来自受所述改变影响的工作站的请求的情况下传递。

5. 根据权利要求1所述的联网的计算机系统,其中,工作站包括用于响应收到对全局数据质量和状态行为定义的改变,重画受影响的图形的装置。

6. 根据权利要求1所述的联网的计算机系统,其中,所述定义编辑器包含这样的装置,该装置用于在工作站上呈现包括全局启用控制的编辑器对话框图形用户界面,在选定该全局启用控制时,使所述全局数据质量和状态行为定义替换非全局数据质量和状态行为。

7. 根据权利要求1的联网的计算机系统,其中,所述定义编辑器包含这样的装置,该装置用于在工作站上呈现包括所支持的数据质量和状态类型的列表的编辑器对话框图形用户界面。

8. 根据权利要求1的联网的计算机系统,其中,所述定义编辑器包含这样的装置,该装置用于在工作站上呈现编辑器对话框图形用户界面,包括用于传达数据质量或状态的可配置的显示特性列表。

9. 根据权利要求8的联网的计算机系统,其中,所述编辑器对话框图形用户界面包括显示特性编辑器框,用于配置来自所述可配置的显示特性列表的具体显示特性。

10. 一种用于配置和传播可分布在联网的计算机系统的多个节点上的人机界面应用内的图形元素的数据状态行为的方法,所述方法包括:

利用第一计算机节点上的配置数据库在中心配置存储中保持全局数据质量和状态行为定义;

在第二计算机节点上提供具有用户界面的定义编辑器,所述定义编辑器支持所述全局数据质量和状态行为定义内所定义的数据质量和状态行为的指定;以及

其中,所述第一计算机节点进一步包含用于定义分布机制的装置,并且所述方法包括利用所述定义分布机制将对所述全局数据质量和状态行为定义的改变传播到网络上执行人机界面应用的一组工作站的进一步的步骤,其中所述人机界面应用包括与具有相关联质量和状态的数据相关联的图形。

11. 根据权利要求 10 的方法,进一步包括通过所述定义编辑器为指定的数据源配置状态图形元素数据质量和状态显示特性。

12. 根据权利要求 11 的方法,其中,所述状态图形元素包括对应于一组所支持的数据质量和状态的一组图标。

13. 根据权利要求 10 的方法,其中,所述传播步骤包括传递对所述全局数据质量和状态行为为定义的改变,而不需要来自受所述改变影响的工作站的请求。

14. 根据权利要求 10 的方法,进一步包括响应收到对所述全局数据质量和状态行为为定义的改变而重画受影响的图形。

15. 根据权利要求 10 的方法,进一步包括以下步骤:通过所述定义编辑器,在工作站上呈现包括全局启用控制的编辑器对话框图形用户界面,在选定该全局启用控制时,使所述全局数据质量和状态行为定义替换非全局数据质量和状态行为。

16. 根据权利要求 10 的方法,进一步包括以下步骤:通过所述定义编辑器,在工作站上呈现包括所支持的数据质量和状态类型的列表的编辑器对话框图形用户界面。

17. 根据权利要求 10 的方法,进一步包括以下步骤:通过所述定义编辑器,在工作站上呈现包括用于传达数据质量或状态的可配置显示特性的列表的编辑器对话框图形用户界面。

18. 根据权利要求 17 的方法,其中,所述编辑器对话框图形用户界面包括用于配置来自所述可配置的显示特性列表的特定显示特性的显示特性编辑器框。

## 用于人机界面的查看状态系统

### 技术领域

[0001] 一般来说,本发明涉及网络化计算机化工业控制与自动化系统的领域。更确切地说,本发明涉及与工业控制与自动化系统关联采用的人机界面(HMI),用于便利在监管层次观察物理装备的状态和运行以及相关的信息。所述HMI通常连接到更低层次的控制单元,比如可编程逻辑控制器或分布式控制系统(DCS)。采用这样的系统也是为了采集和管理与这样的过程及其相关联输出有关的历史信息。

### [0002] 背景技术

[0003] 工业日益增长地依赖于高度自动化的数据采集和控制系统,以确保工业过程运行高效且可靠,同时降低其整体生产成本。数据采集开始于许多传感器测量工业过程的若干情况并向数据收集和控制系统往回报告其测量结果之时。这样的测量结果表现为种类繁多的形式。例如传感器/记录器产生的测量结果包括:温度、压力、pH、材料的质量/体积流、经过特定机器/过程的物品计数、航海线上等待的包装的记载存货、循环完成数等。复杂的过程管理和控制软件往往考察与工业过程相关联的输入数据,产生状态报告和运行概要,并且在许多情况下响应事件/操作员指令,方式为发送命令到修改工业过程的至少一部分的运行的致动器/控制器。传感器所产生的数据也允许操作员执行许多监管任务,包括:响应变化的外部条件(包括原材料的成本)而调整所述过程(如指定新的设定值),检测低效/非最优的运行条件和/或迫在眉睫的装备故障,以及采取补救措施,比如在需要时使装备投入和退出使用。

[0004] 典型的工业过程极为复杂,并且收到的信息量显著大于任何人可能以其原始形式消化的量。例如,在工厂内有数千个(模拟/数字)传感器和控制单元(如阀致动器、电机等)监视/控制多阶段过程的情况并非闻所未闻。传感器有多种类型并且报告过程的多种特征。在它们的测量结果的意义、每项测量结果所发送的数据量以及它们的测量结果的频率的方面,它们的输出同样有变化。至于后者,为了准确并能够快速响应,这些传感器/控制单元的某些每秒进行一次或多次测量。与数千个传感器/控制单元相乘时,大量的定期读数导致如此多的数据流入控制和制造信息管理系统,以至于需要复杂的数据管理和过程可视化技术/应用程序。

[0005] 现在有高度先进的人机界面/过程可视化系统,它们链接到若干数据源,比如以上介绍的传感器和控制器。这样的系统采集并消化(如过滤)以上介绍的过程数据。消化后的过程数据又驱动可视化应用程序,表示/呈现过程的图形视图以便操作人员观测。这样的系统的实例是众所周知的Wonderware IN-**TOUCH**®人机界面(HMI)软件系统,用于可视化和控制种类繁多的工业过程和制造信息。IN-**TOUCH**®HMI过程可视化应用程序包括具体过程及其物理输出的一组图形视图。每个视图又包括一个或多个图形元素。在这些图形元素的显示状态响应所关联/链接的数据源而随时间改变的意义上,它们潜在地是“动画的”。例如,精炼过程的视图潜在地包括油罐图形元素。油罐图形元素具有显示油罐内液面的可视指示器,并且图形元素的液面指示器响应油罐液面传感器供应的数据流而升降,指示油罐内液面。对于观测人,由数据流内不断改变的过程数据值驱动的动画的图

形影像——其中油罐液面指示器仅仅是一个实例——比数字流容易理解得多。由 HMI 应用程序提供的图形影像也用于描绘并便于修改当前的过程设定值。由于这个原因,过程可视化系统比如 IN-TOUCH 已经变为监管过程控制和制造信息系统的基本组件。

[0006] 在 Wonderware 的 ARCHESTRA HMI 和监管控制环境中,真实工厂地面设备由通常称为 Automation Object(自动化对象)的软件单元来建模。在已知的基于 ARCHESTRA 的系统中,执行具体数据采集和过程表示任务的 Automation Object 被定义为应用程序引擎上运行的变量、脚本、警报和历史行为。Automation Object 的这些部分中的每一个都称为“方面(facet)”。应用程序引擎循环地执行所保持的 Automation Object 的若干方面。ARCHESTRA 中更新近的发展涉及视图引擎中保持的自动化对象中图形方面的引入(得到 HMI 对象)。图形方面支持可配置的动画,它们链接到与系统中全局配置数据库中保持的对象相关联的可读的/可写的数据。

[0007] ARCHESTRA 依赖于 Message Exchange(消息交换)以支持对象间的通信,既包括本机内也包括网络上的节点之间的对象间的通信。Message Exchange 支持/提供它传送的一切信息的数据质量、读取状态、写入状态有关的信息。重要的是用户知晓驱动图形元素的信息的质量/状态。例如,未改变的图形状态可能是被中断的连接的结果或者冻结的过程状态。

[0008] 就图形中过去支持的可配置的数据质量和状态而言,它们是按个体配置/定义的。在其他产品中,支持全部数据质量和状态信息的固定子集。不过,应用程序开发者不能按全局改变/增加数据质量和状态的预定义的图形显示。

## 发明内容

[0009] 本发明解决了为工业自动化与控制系统提供更好的在 HMI 上实施数据质量和状态表示的方式的需要,方式为提供数据质量和状态行为的集中定义。所述集中定义随后应用于链接到对其保持/提供状态的数据值的每个图形元素。

[0010] 本发明的第一方面涉及为 HMI 图形组以集中方式定义所述数据质量和状态行为。图形或者直接地(自动化对象的图形方面)或者通过与图形工具箱提供的图形相关联来与多种对象相关联。HMI 应用程序采用所述图形表示例如自动化过程。

[0011] 在整个 HMI 应用程序上,甚至在系统中 HMI 应用程序的若干组上,将集中配置的数据质量和状态指示器合并,以通知所读取/写入数据的数据质量和/或状态的运算器。通过配置界面,对于系统(如 Galaxy)的集中配置数据库中保持的一切对象,管理人员能够访问数据质量和状态行为的集中定义。

[0012] 本发明的另一个特点是在运行时所述集中配置的数据质量和状态行为自动合并并在所述图形元素中。不仅如此,在所述集中定义被改变的情况下,所述修改通过自动更新机构传播到一切受影响的图形元素,无论它们在网络上的什么位置。这些新行为被合并而不中断已经影响了图形元素的任何正在运行的 HMI 应用程序(仅仅重画被改变的图形元素)。

[0013] 不仅如此,另一个方面涉及状态图形元素类型的引入。为了考察指定的数据变量并显示指示所述数据的所述质量或状态的图像或图标,定义了所述状态图形元素,其首要行为在所述集中行为定义中配置。

## 附图说明

[0014] 虽然附带的权利要求书具体地阐明了本发明的特点,不过,连同附图参考以下详细说明,可以最好地理解发明以及其目的和优点,其中:

[0015] 图 1 是示意图,描绘了示范监管过程控制网络,包括多层监管过程控制和制造信息应用程序,包括具有视图引擎并与人机界面 (HMI) 应用程序对象相关联的一组个人计算机;

[0016] 图 2 描绘了多层的对象运行 (host) 布局,用于在实施本发明的示范系统内的若干平台和引擎上运行若干应用程序;

[0017] 图 3 描绘了视图引擎对象定制图元属性的示范组;

[0018] 图 4 描绘了 HMI 应用程序对象定制图元属性的示范组;

[0019] 图 5a- 图 5e 描绘了对于所显示的 HMI 图形元素,与对话框相关联的布局和屏幕显示组,用于配置数据质量和状态行为的集中定义;

[0020] 图 6 提供了与状态图形绘制元素相关联图标的示范默认组及其对应的质量 / 状态触发器。

## 具体实施方式

[0021] 以下说明基于本发明的若干实施例,关于本文未明显介绍的替代实施例,不应当视为限制本发明。例如,本发明合并并在监管过程控制和制造信息应用程序开发和运行时环境内,其中由应用程序对象代表各个数据源(如过程设备和相关联的逻辑)。在 Resnick 等人 2002 年 6 月 24 提交的美国专利申请序列号 10/179,668 SUPERVISORY PROCESS CONTROL AND MANUFACTURING INFORMATION SYSTEM APPLICATION HAVING A LAYERED ARCHITECTURE 中详细介绍了这样的系统的实例,其全部内容在此引用作为参考,包括其中标识的 / 包含的任何参考文献的内容和教导。不过,正如本领域的技术人员将认识到,鉴于所公开的示范实施例,本发明潜在地可应用于多种替代监管过程控制和制造信息应用程序开发和运行时环境。

[0022] 本文的公开主要针对用于在潜在地包括运行着先前定义的 HMI 应用程序的分开实例的许多联网的 HMI 节点的监管过程控制和制造信息应用程序环境内对 HMI 应用程序(如 IN-TOUCH 应用程序)进行集中管理的基础设施和相关的方法。本公开包括可重用的 HMI 应用程序模板内封装的 HMI 应用程序的描述。随后,从 HMI 应用程序模板实例化 (instantiate) HMI 应用程序对象并将其安装在指定的联网 HMI 节点上。

[0023] 本文公开的集中管理 HMI 应用程序的第二方面涉及将组成 HMI 应用程序模板的部分图形的符号的变化传播到一组 HMI 应用程序对象模板中。例如,符号模板全局地定义在 HMI 应用程序之外。符号图形通过引用集中管理的符号模板而合并并在 HMI 应用程序模板中。使用符号模板为 HMI 应用程序定义符号图形使得便于(使用上述交叉引用列表)将符号模板的改变向下传播到一切子符号模板以及通过引用改变的原始的和导出的子符号模板而合并的一切 HMI 应用程序模板。本文以下参考图 5 进一步介绍这样的关系和传播路径。

[0024] 本文公开的集中管理 HMI 应用程序的第三方面涉及通过 IDE126 以配置数据库 124 中内容的多种视图(如部署、派生、模型等)保持和图形地呈现 HMI 对象的状态。当前状态的实例包括:检入 / 检出、已部署 / 解除部署 (undeploy) 以及已改变。这些示范状态中的

每一种都让用户能够关于 HMI 应用程序的分布实例做出决定。

[0025] 所公开的集中管理布局的再一方面是从用户能够从远程部署的配置工具比如集成开发环境 (IDE) 设施编辑现有的 HMI 应用程序定义 (模板)。

[0026] 参考图 1, 示意图描绘了示范分布式 / 联网式监管过程控制环境内组件的容纳 / 层次关系。在该示范网络中, 多个计算硬件节点 (PC 100、120、130、132、134) 中每一个都运行引导程序软件, 它用作后来加载的平台对象的主机和本文称为 IDE 设施的开发工具。随后, 在这些 PC 上安装平台对象实例。在每台 PC 上只能安装一个平台对象。这些平台对象容纳后来安装的引擎对象并向其提供服务。这些引擎对象又潜在地用作容纳后来安装的 HMI、设备集成和应用程序对象。这些引擎对象由其不同的服务 / 容纳能力以及其容纳的对象类型来区分。例如, 视图引擎容纳 HMI 对象实例, 而应用程序引擎容纳设备集成对象和应用程序对象。本文以下进一步介绍以上提及的多种类型的对象。

[0027] 继续参考图 1, 多台 PC 120、130 和 134 运行集成的设计和开发工具 (IDE 126a-c)。开发人员采用 IDE 126 将监管过程控制和制造信息系统的若干组件, 包括应用程序对象, 配置和部署到连接至工程网络 119 的指定 PC 节点。IDE 126 是实用程序 (潜在地包括多个组件), 过程控制和制造信息应用程序, 包括应用程序对象和引擎, 从它定义、创建和部署到多种平台 / 引擎, 包括例如应用程序服务器 PC100。监管过程控制和制造信息应用程序的开发人员通过 IDE 126 实现范围广泛的应用程序设计功能, 包括: 输入新的对象和模板类型, 从现有模板配置新的模板, 定义新的应用程序对象以及将应用程序对象部署到主机应用程序引擎 (如应用程序服务器 PC 100 上的 AppEngine1)。IDE 126 也在合并了先前开发的 HMI 应用程序的 HMI 模板被定义以及最终的 HMI 对象得到实例化并部署到具有先前安装的视图引擎 (如视图引擎 129a 和 129b) 的目标 PC 之处。

[0028] IDE 126 的副本对配置数据库 124 (如星系数据库) 中存储的一组对象模板进行操作, 其中已定义对象模板的名称保持在全局名称表 125 中。全局名称表 125 便于将位置独立的对象名称绑定到位置导出的句柄, 便于在图 1 所示系统内的对象之间对消息进行择路 (route)。配置数据库 124 为已配置的应用程序组件存储对象数据以及与已配置对象相关联的任何代码或文档。配置数据库 124 为图 1 所示的多种对象 (如应用程序引擎、应用程序对象、视图引擎和 HMI 对象) 既存储基本对象模板又存储导出的模板。本文以下参考图 5 描述了示范可视化 HMI 应用程序对象导出和实例创建模式。在示范实施例中, 配置数据库 124 包括微软的 SQL 服务器。

[0029] 经由配置数据库引擎 122, 也称为星系库, 访问配置数据库 124 的内容。对于配置数据库 124 中每个已定义的对象, 通过可图形地呈现的检入 / 检出状态描述符, 配置数据库引擎 122 支持经由 IDE 126 副本的远程多用户访问。配置数据库引擎 122 也支持从集中源向系统中其他节点部署对象和软件。

[0030] 根据展示性实施例, 数据质量和状态行为定义 123 存储在配置数据库 124 内。从这个集中位置, 全局数据分布机制自动地向一切运行时节点传递新版本的定义 123, 无须用户进一步干预。响应数据状态而规定了动画图形行为的定义 123 在落入配置数据库 124 的范围内的节点和 HMI 应用程序的整个组内共享, 而不是专用于任何个体节点或 HMI 应用程序。

[0031] 在此展示性实施例中, 配置数据库引擎 122 由配置数据库平台 127 容纳。配置数据

库平台 127 一般与该系统中 PC 上安装的其他平台相同。不过,向配置数据库平台 127 分配了该系统内的唯一状态(和相应的名称)作为与单一活动配置数据库 124 相关联的平台。因此,所公开的系统包括单一的集中管理的配置数据库。在替代实施例中,数据库 124 中内容的多个副本(如数据库 124 中内容的只读或备份副本)保持在该系统中的多个节点上。在此展示性实施例中,配置数据库平台 127 和所容纳的配置数据库引擎 122 执行以下专用功能:数据/软件分发、保持全局名称表 125、(使用名称表 125)将全局唯一的位置无关的引用字符串解析到位置导出的句柄(用于消息交换)、管理对多用户环境中若干资源的安全/受限的访问、版本管理、集中许可管理以及输入/输出对象模板和实例。

[0032] IDE 126 支持涉及配置数据库 124 的多种配置操作。例如,工程师采用 IDE 126(经由配置数据库引擎 122)将新的对象模板输入配置数据库 124 中、配置新的对象模板以及将对象部署在工程网络 119 上的指定 PC 上。如上所述,完全不同的网络节点上驻留的 IDE 126 的多个副本能够访问和编辑对象定义,包括 HMI 应用程序定义和潜在地合并 HMI 应用程序定义(模板)中的符号定义。

[0033] 在此展示性实施例中,在多个硬件节点(PC 130 和 132)上部署了多个 HMI 对象实例 128a-b。本文以下参考图 4 进一步介绍的 HMI 对象实例 128a-b 提供了图形视图/窗口,其根据经由设备集成和应用程序对象从工厂地面网络 115 上驻留的设备/控制器获得的信息表示过程/工厂或者其一部分的当前状态。单一的视图引擎容纳了由例如已连接的场地设备或 PLC(如 PLC 112)提供的信息驱动的多种配置的过程/工厂视图对应的多个完全不同的 HMI 对象实例。在示范实施例中,多层监管过程控制和制造信息系统架构中(本文以下参考图 3 介绍的)视图引擎 129a-b 容纳 HMI 对象实例 128a-b。虽然在图 1 中对每个视图引擎仅仅显示了单一的 HMI 对象实例,但是每个视图引擎都能够同时容纳多个 HMI 对象实例。

[0034] HMI 对象实例 128 与对应视图引擎 129 之间的容纳关系使得便于访问视图引擎 129 支持的某些服务。例如视图引擎 129 支持独立地更新所容纳的 HMI 对象实例 128(对应的模板更新时的自动的变化传播)。同时,视图引擎 129(在相关联的网络节点上)高速缓存与 HMI 对象实例 128 相关联的显示。

[0035] 转向工程网络 119 上的应用程序服务器 PC 100,在此展示性实施例中,例如以应用程序对象 105 的形式呈现数据源。应用程序对象 105 实现多种功能,包括表示过程装备和相关联的应用程序逻辑的状态。应用程序对象实现多种监视/控制功能中的任何功能,同时位于所展示分布式多层次的监管过程控制和制造应用程序架构的应用程序层次。位于也在该层次中的应用程序层次的设备集成对象 106a 和 106b 表示工厂地面网络上的数据源,比如 PLC(PLC1)、智能场地设备以及相关 I/O 网络(如 PLC1 网络)。

[0036] 应用程序对象与设备集成对象彼此通信,既有本机地(在单台个人计算机之内)也有利用连接到工程网络 119 的个人计算机上容纳的对象通过非本机的通信。

[0037] 例如,在配置数据库 124(如 Wonderware 星系库)保持的全局名称表 125 内标识应用程序对象 105,经由例如 IDE 126a-c 和 HMI 对象实例 128a-b(它们例如合并了 IN-TOUCH 应用程序及其相关联的显示)使其内容对开发人员可用。因此,根据本发明的实施例,使用例如 WINDOWMAKER 实用程序初始地创建工厂/过程的 IN-TOUCH 应用程序形式的动态图形视图。整个 IN-TOUCH 应用程序随后合并 HMI 对象模板中,包括本文介绍的多层应用程序



执行环境中使用的必要组件。所得 HMI 对象模板在配置数据库 124 中存储 / 维护 / 管理。随后,基本模板的后续导出版本保持为子模板,并且与父 HMI 对象模板保留继承关系。原始的和导出的模板可用于经由 IDE 126 分布到网络 119 上包含先前安装的视图引擎(如视图引擎 129a)的适当节点。

[0038] 继续参考图 1,应用程序服务器 PC 100 执行多层的监管过程控制和制造信息应用程序,包括第一部分 104。应用程序部分 104 包括应用程序对象 105 和设备集成对象 PLC1Network 106a 和 PLC1 106b。设备集成对象 PLC1Network 106a 使得便于配置数据访问服务器(如 OPC DAServer 116)。设备集成对象 PLC1 106b 用作 OPC 客户机,访问 OPC DAServer 116 的缓冲区内的数据位置。数据访问服务器 116 与若干设备集成对象合作地输入并缓冲来自外部过程控制组件(比如 PLC(如 PLC1 112)或工厂地面网络 115 上的其他场地设备(未描绘))的数据。应用程序引擎 107 容纳应用程序对象 105 和设备集成对象 106a 和 106b。应用程序引擎 107 作为主机管理着所容纳的应用程序和 设备集成对象的定期的 / 事件驱动的执行。本文以下参考图 2 介绍 PC100 上分层容纳布局的前述组件。

[0039] 在此展示性实例中,经由数据访问服务器 116 提交对数据的请求,以从 PLC1112 检索数据。检索出的数据随后由 HMI 对象实例 128a 和 128b 用于驱动表示例如工厂地面装备的状态的图形显示。数据访问服务器 116 的数据缓冲区由个人计算机 100 上执行的多种应用程序层次的对象(如应用程序对象 105、PLC1Network 106a、PLC1 106b 等)(直接 / 间接)访问。应用程序对象的实例表示数据源和逻辑,包括例如离散设备、模拟设备、场地基准、事件 / 触发器、生产事件等。在示范实施例中,由应用程序层次的对象 105、106a 和 106b 获得 / 提供的信息存储在运行时(Historian)过程信息数据库(未显示)。此数据随后由 HMI 对象实例 128a-b 获得以驱动动画过程图形的显示状态。

[0040] 数据访问服务器 116 是例如 OPC 服务器。不过,本领域的技术人员将很容易认识到由数据访问服务器 116 潜在地执行的定制和标准化数据格式 / 协议的范围广泛。不仅如此,示范应用程序层次的设备集成对象 106a 和 106b 通过连接到数据访问服务器 116,表示 PLC 网络和 PLC 本身的操作。不过,应用程序引擎 107 容纳的应用程序层次的对象(如设备集成和应用程序对象)包括在监管过程控制和制造信息应用程序的环境中执行所需监管控制和数据采集 / 集成功能的可执行对象类的实际上无限范围。

[0041] 监管过程控制和制造信息系统经由多种通信通道潜在地与多种过程 / 工厂信息源集成。包括多层应用程序(其包括部分 104)的示范系统通信地耦合到 PLC1 112。PLC1 又经由工厂地面网络 115 接收工厂装备状态信息。在特定实施例中,PLC 112 包括 PC 100 所连接到的以太网 LAN 上的节点。在其他实施例中,PLC 112 直接链接到 PC 100 上的物理通信端口。在又其他的替代实施例中,PC 100 从场地 I/O 模块接收数据,这些模块从分布式调节控制系统中运行的场地设备接收例如模拟数据。

[0042] 应当注意,图 1 描绘的和上文介绍的系统仅仅是监管过程控制和 制造信息系统所用的包括多层的层次架构的系统的实例。应当进一步注意,图 1 呈现为包括软件和物理计算硬件的已安装组件之间的容纳和 / 或包含相互关系的逻辑视图。本文公开的系统适于实际上任何网络拓扑。例如,本发明可应用于如下系统,即,其中配置实用程序和监管过程控制可视化应用程序都运行在链接到受控过程的单一计算机系统上。

[0043] 转向图 2,类图描绘了分层软件的层次容纳布局,包括计算机可执行的指令,与执

行监管过程控制和制造信息应用程序的至少一部分的计算机（如 PC 100）相关联。此计算机执行操作系统 200，比如微软的视窗，在层次的最低等级。操作系统 200 容纳引导程序对象 202。引导程序对象 202 加载到计算机上并与操作系统 200 执行的启动过程相关联地激活。作为平台类对象 204 的主机，必须先激活引导程序对象 202，再启动平台类对象 204 的操作。引导程序对象 202 启动和停止平台类对象 204。引导程序对象 202 还提供平台类对象 204 采用的若干服务，以启动和停止平台类对象 204 容纳的一个或多个引擎对象 206。

[0044] 平台类对象 204 容纳一个或多个引擎对象 206。在本发明的实施例中，对于这一个或多个引擎对象 206，平台类对象 204 表示执行特定操作系统的计算机。平台类对象 204 保持着平台类对象 204 上部署的引擎对象 206 的列表，启动和停止这些引擎对象 206，并且在这些引擎对象 206 崩溃时重新启动它们。平台类对象 204 监视引擎对象 206 的运行状态并向客户机发布状态信息。平台类对象 204 包括系统管理控制台诊断实用程序，它允许在执行平台类对象 204 的计算机系统上执行诊断和管理任务。平台类对象 204 也对分布式警报子系统提供若干警报。

[0045] 引擎对象 206 容纳一组应用程序对象 210，它们实施与应用程序相关联的监管过程控制和 / 或制造信息采集功能。引擎对象 206 发起一切应用程序对象 210 的启动。引擎对象 206 也利用调度器对象 208 的帮助，调度应用程序对象 210 关于彼此的执行。引擎对象 206 以调度器对象 208 寄存应用程序对象 210 以便执行。调度器对象 208 根据引擎对象 206 中对应的一个指定的配置，相对于其他应用程序对象执行若干应用程序对象。引擎对象 206 监视应用程序对象 210 的运行并将有故障的对象置于被隔离状态。引擎对象 206 支持检查点，方式为将由自动化对象对运行时应用程序的改变保存到配置文件 / 恢复这种改变。引擎对象 206 保持将若干属性参考（如 tank1.value.pv）绑定到应用程序对象 210 中恰当的一个的名称绑定服务。引擎对象 206 关于所容纳的设备集成对象执行类似的功能。

[0046] 引擎对象 206 最终控制着应用程序对象 210 中相关联对象的执行将如何发生。不过，一旦引擎对象 206 对应用程序对象 210 确定了执行调度，其执行的实时调度由调度器 208 控制。调度器 208 支持包含方法 RegisterAutomationObject() 和 UnregisterAutomationObject() 的接口，使引擎对象 206 能够向 / 从调度器 208 的被调度的操作的列表增加 / 去除应用程序对象中的特定对象。

[0047] 应用程序对象 210 包括范围广泛的对象，它们执行的商务逻辑便于在例如工业过程控制系统的环境中执行具体过程控制操作（如启动泵、驱动阀）以及 / 或者信息汇集 / 管理功能（如根据收到的场地设备输出信号值发出警报）。过程控制（自动化）应用程序对象的实例包括模拟输入、离散设备和 PID 循环对象。一类应用程序对象 210 作用在过程控制系统比如 PLC 经由设备集成对象（如 OPC DAsServer 118）提供的数据上。也由引擎对象容纳的设备集成对象的功能是提供过程控制 / 制造信息源与监管过程控制和制造信息应用程序之间的桥路 / 数据通道。

[0048] 在示范实施例中，应用程序对象 210 包括引擎对象 206 和调度器 208 访问的应用程序界面。引擎对象 206 访问应用程序对象界面以便初始化应用程序对象、启动应用程序对象和关闭应用程序对象。调度器 208 使用此应用程序对象界面以便启动对应应用程序对象的已安排执行。

[0049] 已经介绍了示范多层次分层布置的监管过程控制和制造信息应用程序中的引导程序、平台、引擎和应用程序对象之间的关系后,应当注意,对于构成 HMI 应用程序的多层架构的对象,存在类似的关系(参见如图 1 中的 PC2130 上的 HMI 应用程序分层架构)。

[0050] 转向图 3,对为了便于容纳已经部署到某 PC(如 PC 130)的一组可用 HMI 对象实例中指定的一个而增加基本引擎功能的视图引擎对象定制图元,标识属性的示范组。在 Resnick 等人 2002 年 6 月 24 提交的美国专利申请序列号 10/179,668 SUPERVISORY PROCESSCONTROL AND MANUFACTURING INFORMATION SYSTEMAPPLICATION HAVING A LAYERED ARCHITECTURE 中介绍了基本引擎图元的内容/功能,其全部内容在此引用作为参考。视图引擎对象支持基本引擎功能,比如部署、解除部署、启动和关闭。视图引擎对象也支持本文以下进一步介绍的可视化应用程序专用的功能。在展示性实施例中,视图引擎对象是专用的引擎对象类型,它们仅仅容纳 HMI 对象实例——与能够容纳多种应用程序层次对象包括设备集成对象和应用程序对象的应用程序引擎相反。

[0051] 所述视图引擎(如视图引擎 129a)容纳并调度指定 HMI 对象实例的执行。所述视图引擎根据当前占用的视图引擎运行时状态,支持关于所容纳 HMI 对象实例的一组运行时操作。当视图引擎处于启动状态时,所容纳的 HMI 对象为:从检查点初始化的、由所述视图引擎启动的、以 Message Exchange(或其他合适的对象间数据通信服务)注册的以及根据与所述视图引擎相关联的调度器发出的命令执行的。当视图引擎进入扫描开启或扫描关闭状态时,所容纳的 HMI 对象收到视图引擎的新扫描状态的通知。不仅如此,当视图引擎进入关闭状态时,所容纳的 HMI 对象由其容纳引擎关闭。

[0052] 在示范实施例中,视图引擎管理着向其部署的 HMI 对象实例列表。不过,视图引擎不负责调用脚本的执行或者读写与 HMI 对象实例相关联的相关过程数据。而是将执行脚本和管理数据订阅委托给合并(嵌入、包装在)对应的 HMI 对象实例中的 HMI(如 IN-TOUCH)应用程序。因此,在此展示性实施例中,不能在图 1 中描绘的所公开的多层容纳架构中执行的某另外情况下独立的 HMI 应用程序合并到某 HMI 包装对象中以提供这样的能力。因此,独立的传统 HMI(IN-TOUCH)应用程序可以无缝地合并到实施本文以上参考图 1 和图 2 介绍的基于对象的分层架构的系统中。

[0053] 如上所述,视图引擎的定制图元包括与容纳 HMI 应用程序对象有关的一组属性。(本文以下介绍的)在图 3 中标识的这组属性旨在示范并根据本发明的替代实施例而修改。

[0054] 在此展示性实施例中,应当注意,这些对象(如平台、引擎、应用程序对象等)以一组数据点(本文称为“属性”)定义。每个属性又潜在地包括配置和若干运行时空柄,它们根据此属性的当前指定值处理所述对象。在此示范实施例中,这些句柄是受触发的事件并将具有定制代码的功能。配置设定的句柄是在使用配置客户机(比如此 IDE)设置此属性时触发的事件,而运行时设定的句柄在运行时客户机(比如 IN-TOUCH)设定此属性的值时触发。

[0055] 为了部署到某视图引擎而指定某指定 HMI 对象模板时,\_CreateViewApp 属性 300 创建新的 HMI 对象实例。对此新 HMI 对象实例的引用增加到由此视图引擎管理的已部署的 HMI 对象的列表中。

[0056] \_DeleteViewApp 属性 302 从视图引擎上当前部署的一组 HMI 对象中去除某先前部署的 HMI 对象。对此 HMI 对象的对应引用从此视图引擎上已部署 HMI 对象的列表中删除。

[0057] `_StartHostedObjects` 属性 308 开始运行此视图引擎上一切已部署的 HMI 对象。这些 HMI 对象的初始状态基于从检查点永久存储中提取的数值。

[0058] `_StopHostedObjects` 属性 310 开始关闭此视图引擎当前容纳的一切 HMI 对象实例。

[0059] 转向图 4, 注意指向 HMI 应用程序对象定制图元属性的示范组。HMI 应用程序对象实现与提供分布式监管过程控制和制造信息应用程序的图形视图部分相关联的功能。在以上介绍的分层运行时环境中 容纳视图引擎上执行的 HMI 应用程序对象管理着已合并的 HMI (IN-TOUCH) 应用程序的检入 / 出、编辑、部署以及运行时属性监视, 该应用程序又提供工厂 / 过程的动态图形视图。HMI 应用程序的图形状态由例如工厂装备传感器、监视器和控制器提供的实时数据驱动。这样的信息经由 ( 本文以上参考图 1 介绍的 ) 应用程序引擎上执行的设备集成和应用程序对象从工厂地面网络提取。HMI 对象也支持应用程序服务器容纳的应用程序层次对象上的引用标签 (MessageExchange), 动态过程数据通过它传递到其中合并的 HMI 应用程序。

[0060] 在此展示性实例中, 执行脚本和管理数据订阅的 HMI ( 如 IN-TOUCH ) 应用程序合并 ( 嵌入 / 包装在 ) 对应的 HMI 应用程序对象模板和实例中。因此, 在此展示性实例中, 不能在图 1 中描绘的所公开的多层容纳架构中执行的某另外情况下独立的 HMI 应用程序合并 ( 在某 HMI 应用程序包装对象中, 它便于集成 ( 管理、运行等 ) 采用前述所容纳的分层运行时环境的系统内的 HMI 应用程序。因此, 独立的传统 HMI (IN-TOUCH) 应用程序可以无缝地合并到实施本文以上参考图 1 和图 2 介绍的基于对象的分层架构的系统中。

[0061] 前述 HMI 包装对象包括定制图元, 它包括一组属性, 与视图引擎所支持的容纳环境内 HMI 应用程序的执行有关。( 本文以下介绍的 ) 图 4 中标识的这组属性旨在示范并且根据本发明的替代实施例而不同。

[0062] `_VisualElementReferenceList` 属性 400 包含向某 HMI 应用程序对象分配的一切可视元素 ( 如符号 ) 的列表。

[0063] `_VisualElementReferenceStatusList` 属性 402 指定向某 HMI 应用程序对象分配的每个符号的当前状态。此状态可以用于为 HMI 应用程序对象内包含的符号传达多种状态, 包括例如显示某符号何时已经从此 HMI 应用程序对象中删除。

[0064] 当与 HMI 应用程序对象相关联的若干 HMI 应用程序文件与配置数据库 124 同步时, `DeploymentInProgress` 属性 404 设定为真。

[0065] `_UndeployNotify` 属性 406 指定 HMI 应用程序对象是否能够解除 部署。

[0066] `_StartSynchronization` 属性 408 设定为真以通知 HMI 应用程序对象, 应当开始将与 HMI 应用程序对象相关联的应用程序的 HMI 应用程序文件传递到此 HMI 应用程序对象所部署的节点。

[0067] `_SyncStatus` 属性 410 指明 HMI 应用程序向相关联的 HMI 应用程序所部署的节点传递的状态。

[0068] `_Namespace` 属性 412 包含关于参数标签的信息, 这些标签是与 HMI 应用程序对象相关联的 HMI 应用程序的一部分。`_Namespace` 属性 412 用于支持属性浏览器内 HMI 应用程序标签的浏览。

[0069] 刚刚在关闭相关联的 HMI 应用程序编辑器之前, 写入 `_ShutdownNotify` 属性 414,

以确保在编辑过程允许关闭之前进行中的同步方法完成。

[0070] 在编辑会话开始时，\_BeginDBMonitoring 属性 416 写入 HMI 应用程序编辑器何时启动，以确保此 HMI 应用程序对象正确地加载和验证。

[0071] LastModified 属性 418 指定此 HMI 应用程序的版本号修改的最后时间。

[0072] 例如，此 HMI 应用程序对象展现出以下的说明中归纳的运行时行为。当（在容纳视图引擎的指引下）执行此 HMI 应用程序对象时，合并在此 HMI 应用程序对象中的逻辑判断此 HMI 应用程序对象内合并的 HMI 应用程序是否需要从配置数据库 124 传递。如果需要启动传递，那么在此 HMI 对象的下一次扫描时由视图引擎开始此传递。

[0073] 在此 HMI 应用程序对象启动后任何时间都能够发生同步。此 HMI 应用程序对象启动 HMI 应用程序与源应用程序的同步。如果挂起的同步操作完成，那么此 HMI 对象设置配置数据库 124 中的属性以指明此传递已完成。根据本发明的实施例，同步应用程序可以包括更新封装的 HMI 应用程序或配置数据库 124 内已经更新的 HMI 应用程序中合并的各个符号对象。在更新 HMI 应用程序的情况下，从配置数据库 124 仅仅传递配置数据库 124 内与当前在具有合并了 HMI 应用程序的 HMI 应用程序对象实例的节点上文件不同的应用程序文件。

[0074] 根据示范实施例，通过图 5a-e 中的实例提供 IDE 126 支持的全局集中管理界面，以便定义这样的显示合并了这样的状态信息的 HMI 应用程序图形内首要的 HMI 应用程序数据质量和状态行为。所指定的行为活动时将取代所部署的 HMI 应用程序图形上任何本机定义的行为。正如以上参考图 1 所述，质量和状态行为定义 123 与其他集中管理的配置信息一起保持在配置数据库 124 中。

[0075] 在 IDE 126 配置环境内，用户（受控于检入 / 出状态）从运行此 IDE 126 的网络上的任何节点访问数据质量和状态行为定义 123。在示范实施例中，用户请求编辑数据库 124 中存储的数据质量和状态定义 123。如果定义 123 尚未检出并且登录到 IDE 的用户具有适当的许可，那么经由 IDE 126 支持的对话框（例如参见图 5a 中的对话框 500）提供访问。在用户编辑定义 123 时，它的状态为“检出的”。当用户为了编辑定义 123 而关闭对话框时，此定义的检入副本（在需要时）被更新并且状态改变为“检入的”。

[0076] 在此示范实施例中，对话框 500 划分为以下主要区域：启用 / 禁用质量显示 502、质量和状态替换选项 504、配置区域 506 和命令区域 508。启用 / 禁用质量显示 502 开启 / 关闭定义 123 的检入版本中存储的替换行为。质量和状态替换选项 504 显示状态 / 质量的列表以及对每项指定的性质。配置区域 506 包含选项 504 中选定质量 / 状态所用的详细配置选项。命令区域 508 呈现关于对话框访问的定义要执行的措施。本文以下参考图 5b-e 进一步介绍这些区域的每一个。

[0077] 主要参考图 5b，用户通过启用 / 禁用按钮 512 全局地启用 / 禁用已配置的替换行为。

[0078] 参考图 5c，为了指定替换动画行为的目的，向用户呈现可配置数据质量和状态类型的示范组。这个列表与示范系统中合并的 MessageExchange 消息递送协议支持的信息相关联。示范二维网格 514 列出了所支持数据质量 / 状态以及传达所述质量 / 状态情况（如文本 -Ts、填充 -Fs、线 -Ls、状态图元（图标组）-St 以及轮廓 -O1）所用的一组可用显示特性（列）的集合。用户选择任何数目的可用显示特性（无、某些或全部）用于每个列出的

数据质量 / 状态。在此展示实例中,对每个列出的数据质量 / 状态情况都指定基于图标的“状态图元”显示特性。如果对列出的质量 / 状态(如“不合格”)没有指定所支持的替换显示特性,那么在相关联的质量 / 状态为真时,关于显示特性将不采取任何措施。示范实施例中列出的质量 / 状态包括:

[0079] 1. 不合格质量 - 数据无法使用。映射到 OPC 不合格状态。

[0080] 2. 不确定质量 - 数据可疑但是能够使用(例如,手工替换数据)。映射到 OPC 不确定状态。

[0081] 3. 初始化质量 - 数据尚未可用但是很快将可用。映射到具有初始化子状态的 OPC 不合格状态。

[0082] 4. 通信错误读取状态 - 由于与目标 AutomationObject 的通信错误造成请求失败;或者在 DeviceIntegrationObjects 的情况下,由于与目标设备的通信错误造成请求失败。

[0083] 5. 配置错误读取状态 - 由于配置中的错误造成请求失败。在 DeviceIntegrationObjects 的情况下,由于对象或服务器的不合格项名称或其他非法配置造成请求失败。

[0084] 6. 操作错误写入状态 - 由于非法的操作员动作造成请求失败。在 DeviceIntegrationObjects 的情况下,由于非法的操作员动作造成请求失败。例如,在设备当前处于不允许修改它的操作模式时,试图写入此设备中的项,或者试图写入目标设备无法接受的不合格值。

[0085] 7. 软件错误写入状态 - 由于内部软件错误造成请求失败。

[0086] 8. 安全错误写入状态 - 由于安全访问权限不足造成请求失败。

[0087] 9. 警告写入状态 - 这仅仅适用于若干组。成功地完成但是具有某种警告情况的操作,比如箝位该值。

[0088] 10. 挂起写入状态 - 请求已经进入队列但是尚未完成。这不是错误状态而是指明操作在进行中。MxCategoryPending 状态是瞬间状态,不无限期地持续。

[0089] 转向图 5d,为了便于用户(对于从网格 514 中列表选定的数据质量 / 状态)经由配置区域 506 输入全局实施的动画行为的可配置特征,描绘了示范布局。状态区 520 包含着当前选定的状态 / 质量的名称。预览区 522 随着在对话框中在字段之间导航而自动地更新。在配置了替换值时将显示配置的预览。当选定复位至默认按钮 524 时,一切值都复位至当前选定质量 / 状态的默认值。关于配置区 526,点击图像选择将打开标准的文件打开对话框。文件可以是以下标准图像类型之一: BMP、GIF、JPG(jpeg)、TIF(tiff)、PNG、ICO、EMF 和 WFM。一旦选定了图像,此图像便更新。点击颜色选择会打开标准的颜色对话框。一旦选定了某选择,颜色选择将更新。

[0090] 对于选定数据质量 / 状态,以下风格经由配置区域 506 分别可配置为网格 514 中列出的和本文以上介绍的每种状态的运行时替换值。

[0091] 1. 文本 - 这将应用于具有已配置动画的一切文本框、文本标签、单选按钮、复选框、编辑框、组合框和列表框绘制元素。每个以下选项都能够分别启用或禁用。

[0092] a. 字体

[0093] b. 颜色

[0094] c. 闪烁

[0095] 2. 填充 - 这将应用于支持填充颜色并具有已配置动画的一切闭合绘制元素（椭圆、矩形、圆角矩形、多边形、按钮、闭合曲线、饼状图或连枝）。每个以下选项都能够分别启用或禁用。

[0096] a. 颜色

[0097] b. 闪烁

[0098] 3. 线条 - 这将应用于支持线条颜色并具有已配置动画的一切绘制元素（线条、H/V 线条、椭圆、矩形、圆角矩形、折线、多边形、按钮、曲线、闭合曲线、弧、饼状图或连枝）。每个以下选项都能够分别启用或禁用。

[0099] a. 线型

[0100] b. 深浅

[0101] c. 颜色

[0102] d. 闪烁

[0103] 4. 状态图形 - 这将应用于一切状态图形元素（参见如图 6）。每个以下选项都能够分别启用或禁用。

[0104] a. 线条颜色

[0105] b. 线型

[0106] c. 线条深浅

[0107] d. 填充颜色

[0108] e. 图像

[0109] f. 图像透明颜色

[0110] g. 图像风格

[0111] h. 图像对齐

[0112] 5. 轮廓 - 这是将围绕具有已配置动画的一切图形元素绘制的线条。整个“轮廓”功能能够整体禁用。

[0113] a. 轮廓颜色

[0114] b. 轮廓线型

[0115] c. 轮廓深浅

[0116] d. 轮廓闪烁

[0117] 参考图 5e, 命令区域 508 提供了“确认”和“取消”按钮的标准组, 以接受或抛弃与此对话框会话相关联的改变的当前组。预览图例按钮启动新的对话框, 它图形地显示当前已配置替换的外观。

[0118] 状态图形绘制元素

[0119] 转向图 6, 根据示范实施例, 以 HMI 编辑器工具条上的其他图元（如线条、方框、文本框等）提供状态图形绘制元素。不过, 与这些纯粹图形图元不同, 状态图形绘制元素包括用户配置的数据源的多种数据质量 / 状态情况对应的可配置图标组（参见如图 6 的默认图标组）和动画。所显示的图标和 / 或动画表示与状态图形绘制元素（如图标）已经附着的图形相关联数据的质量 / 状态。在示范实施例中, 状态图形 绘制元素与 HMI 编辑器（如 WindowMaker）的其他绘制元素一起出现在工具条上。以鼠标或其他图形界面点击设备选择状态图形绘制元素工具后, 用户点击并在画布上拖出矩形（表示状态图形绘制元素的位置

置)以放置状态图形绘制元素。在配置期间,状态图形将显示普通图形作为位置占用者,因为它出现在工具盒上(由于在运行时之前可能没有状态)。

[0120] 一旦数据质量/状态图形元素已经放置在 HMI 视图编辑器的画布上,用户便通过打开动画编辑器访问状态图形元素的其他可用的可配置特征。状态图形具有的预定义动画具有以下两个标签:

[0121] 1. 图形 - 将显示当前画布上链接到数据的一切图形元素。用户可以不选择、选择所述元素的一个或许多以与状态图形绘制元素相关联。

[0122] 2. 表达式 - 这个标签将允许用户直接输入表达式,它将包含对数据的引用。

[0123] 在运行时评估链接到相关联图形中的或在表达式中显式地调出的数据点,以确定何种状态和质量行为(如果有的话)将应用于此状态图形元素。

[0124] 质量和状态行为的运行时执行

[0125] 因为在运行时查看器(如 WindowViewer)显示了某图形并且启用了质量和状态行为,在相关联的动画中任何数据点假定以上在可配置的状态区域中列出的状态之一时,将假定已经被配置的专用行为。不过,以下动画在活动时,将取消图形上当前可应用的质量和状态行为:

[0126] 1. 可见性动画 - 如果因为可见性动画此元素当前不可见,那么将不显示质量和状态行为。

[0127] 2. 禁用动画 - 如果某元素当前使禁用动画活动,一切用户交互动画将从质量和状态评估中排除(用户输入、水平滑动块、垂直滑动块、按钮或脚本)。

[0128] 3. 动画被禁用 - 每项动画都能够分别被配置为禁用。当某具体动画类型被禁用时,该动画中配置的一切数据点将不加入质量和状态评估。

[0129] 对于某图形,所配置的数据质量和状态行为的运行时执行的另一个方面是应对多个活动状态 - 导致冲突的动画。在示范实施例中,所显示的动画由状态的严重性/优先权确定。例如,矩形图形元素被配置为具有不止一个数据点与其动画相关联。然后将可能对于此矩形图形元素在同一时间有不止一个已配置的状态活动。对于任何单一元素,某时刻仅仅显示已配置的质量和状态行为之一。所以优先权确定应用多个活动状态中的哪一个。在示范实施例中,采用以下优先权次序从多个数据状态中选择(高到低):通信错误、配置错误、挂起、操作错误、软件错误、安全错误、警告、不合格、不确定和初始化。

[0130] 替换列表区域 504 中列出状态的次序确定优先权的次序(不合格质量数据在最高)。例如,假若矩形元素有一个点为不合格质量而另一个点为配置错误,这两种状态都已经被配置到矩形元素,那么仅有配置错误行为活动地显示。

[0131] 不仅如此,所支持的数据质量/状态中的某些表示有关某时间点的信息并且不会无限期地持续。这些状态显示 20 秒的最小保持时段,然后如果此状态不再存在,此值将返回当前状态。这就确保了一切质量/状态故障将向用户显示至少 20 秒。

[0132] 本文介绍的数据质量和状态行为配置模式的另一个方面是在定义 123 改变时已配置行为的更新。在示范实施例中,配置数据库 124 中保持的集中定义的数据质量和状态行为定义 123 传播到系统(如星系)内的一切节点。使用配置数据库 123 和数据库引擎 122 支持的全局数据更新功能,对所配置的数据质量和状态行为定义 123 的改变自动传递,无须从受影响的工作站发出请求。不仅如此,更新定义时无须部署任何对象或关闭运行的



HMI 应用程序。受影响的图形收到这些改变后,将使用更新后的、全局应用的替换数据质量和状态行为定义来进行重画。

[0133] 考虑到所公开的这个系统的原理可以应用的许多可能的实施例,应当理解本文关于附图介绍的实施例仅仅意味着展示,不应当视为限制本发明的范围。例如,本领域的技术人员将理解,计算机可读介质上以计算机可执行指令形式存储的软件中显示的所展示实施例的某些元素可以由硬件实施,反之亦然,所展示的实施例也能够在布局和细节中修改而不脱离本发明的实质。所以,本文介绍的发明设想了可能落入以下的权利要求书及其等效内容范围的一切这样的实施例。

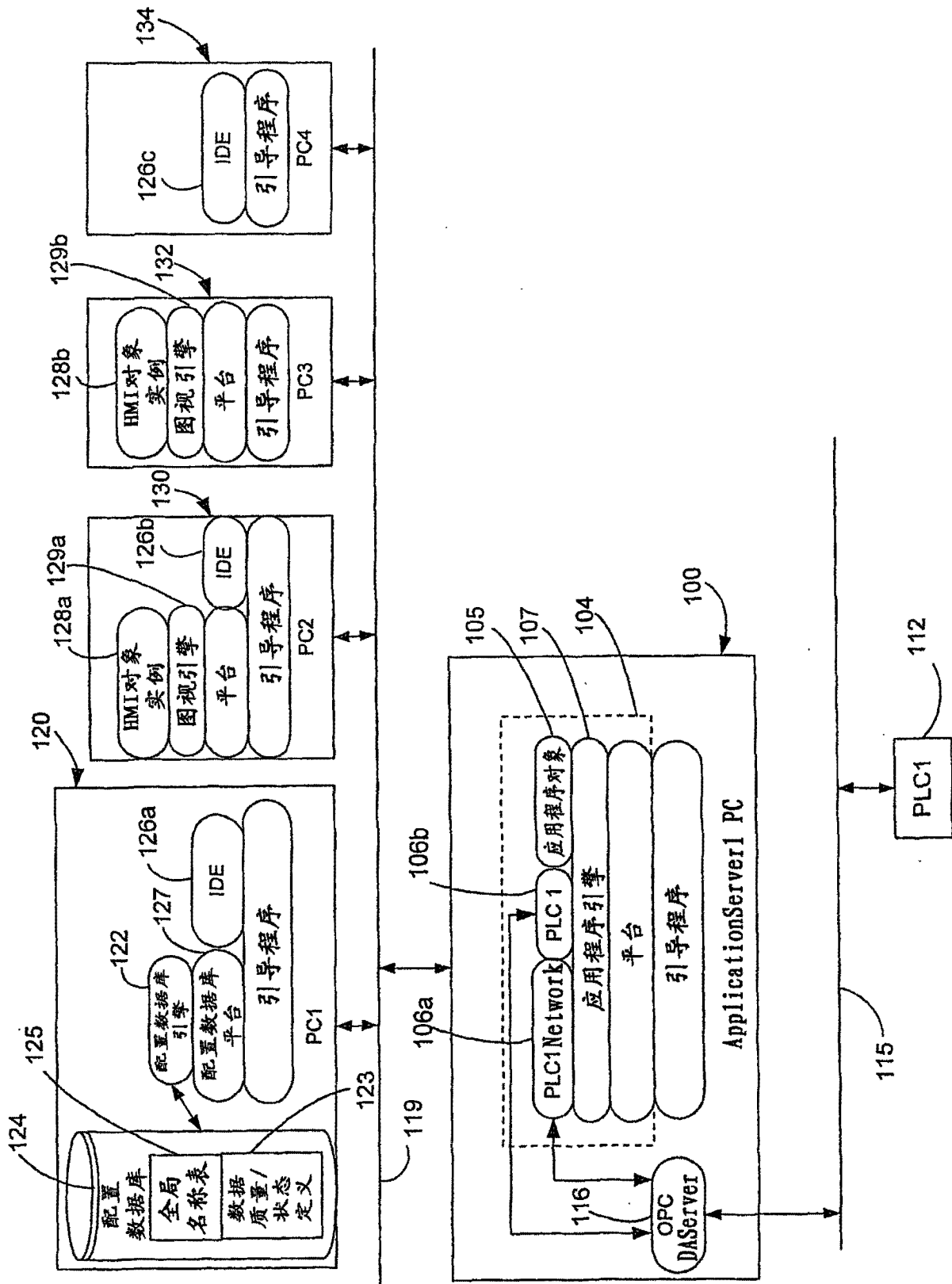


图 1

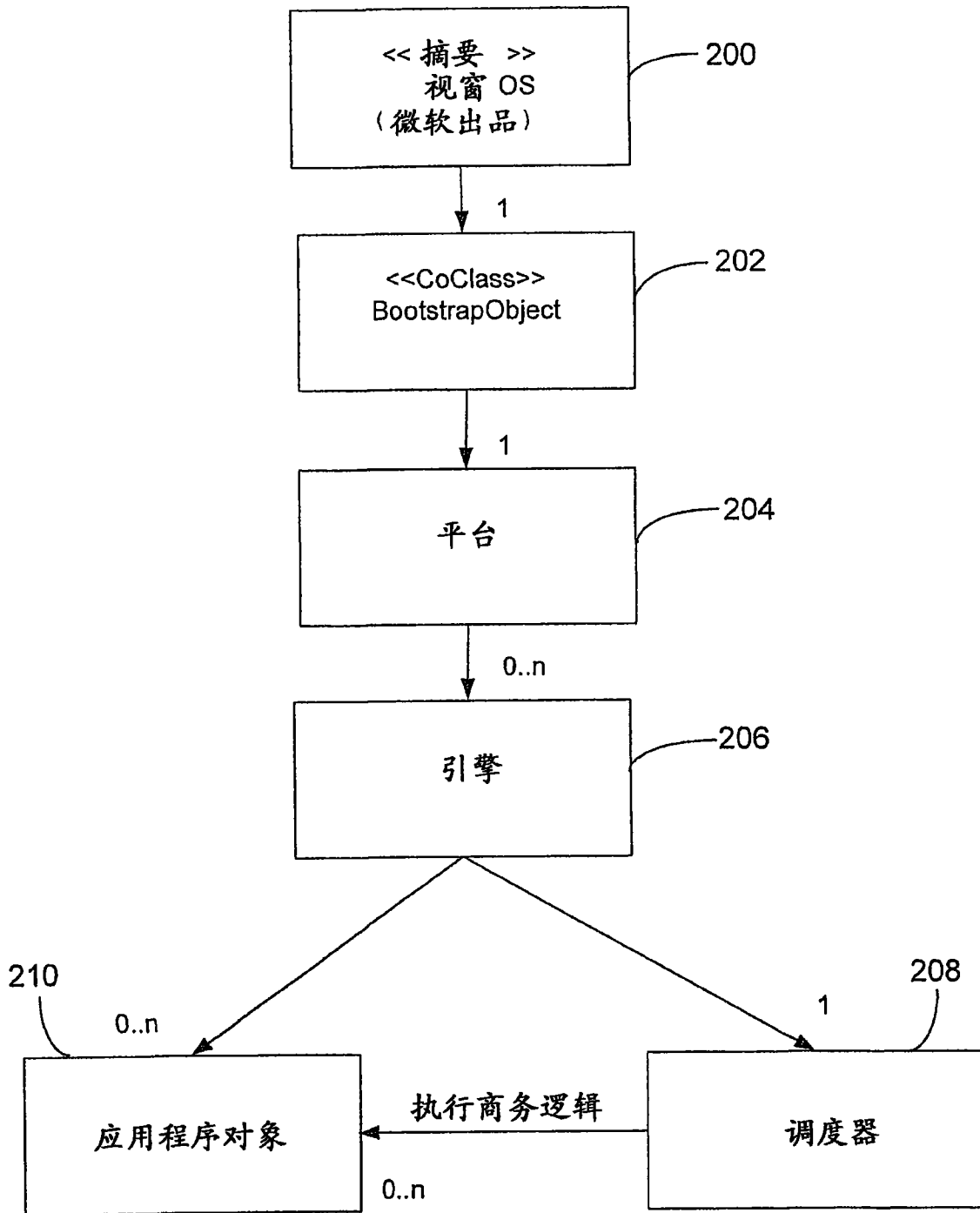


图 2

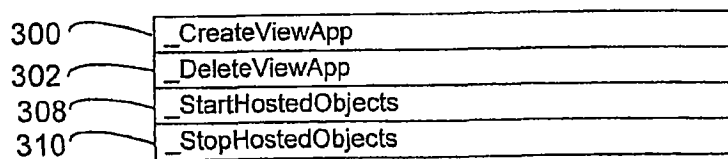


图 3

400	_VisualElementReferenceList
402	_VisualElementReferenceStatusList
404	DeploymentInProgress
406	_UndeployNotify
408	_StartSynchronization
410	_SyncStatus
412	_NameSpace
414	_ShutdownNotify
416	_BeginDBMonitoring
418	_LastModified

图 4

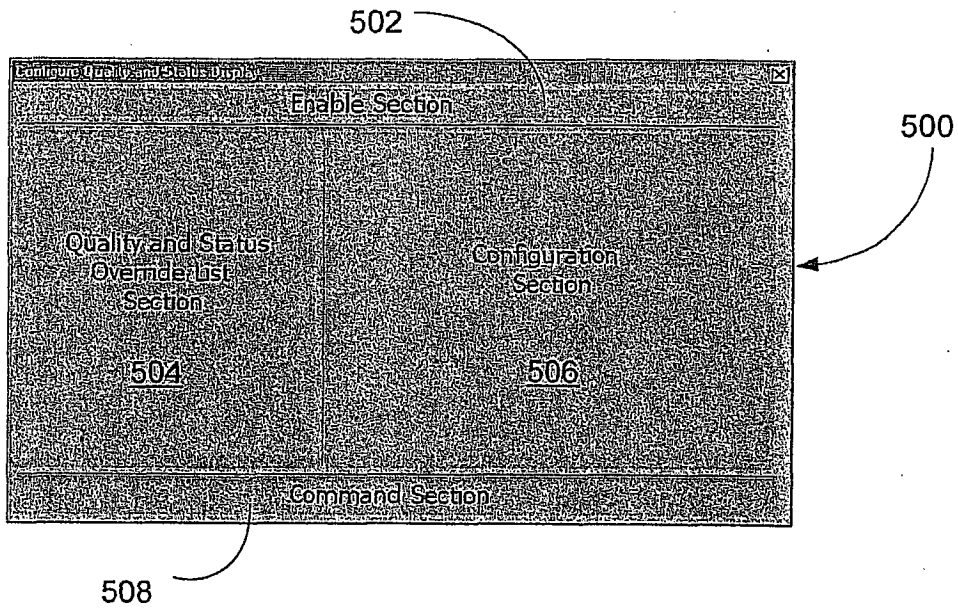


图 5a

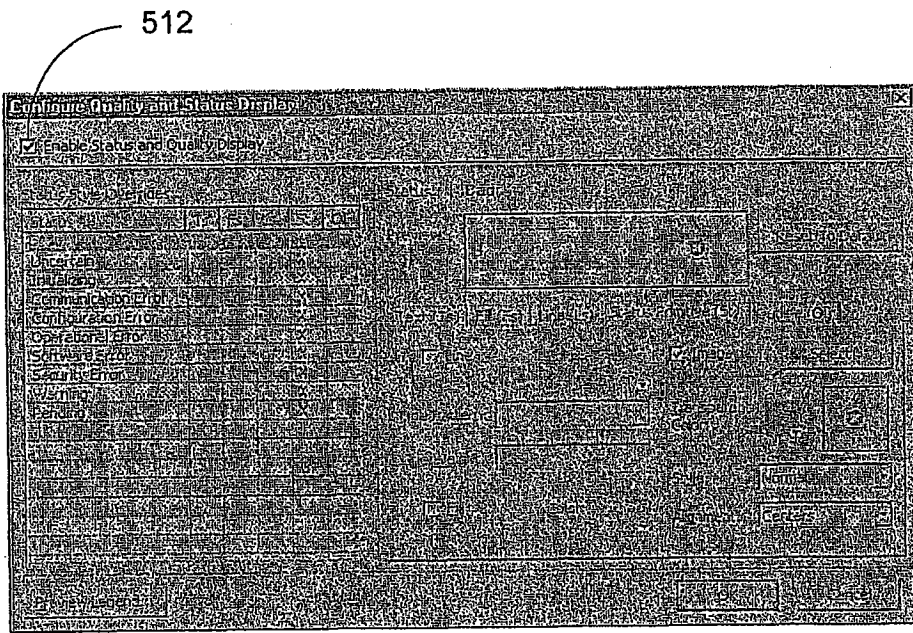


图 5b

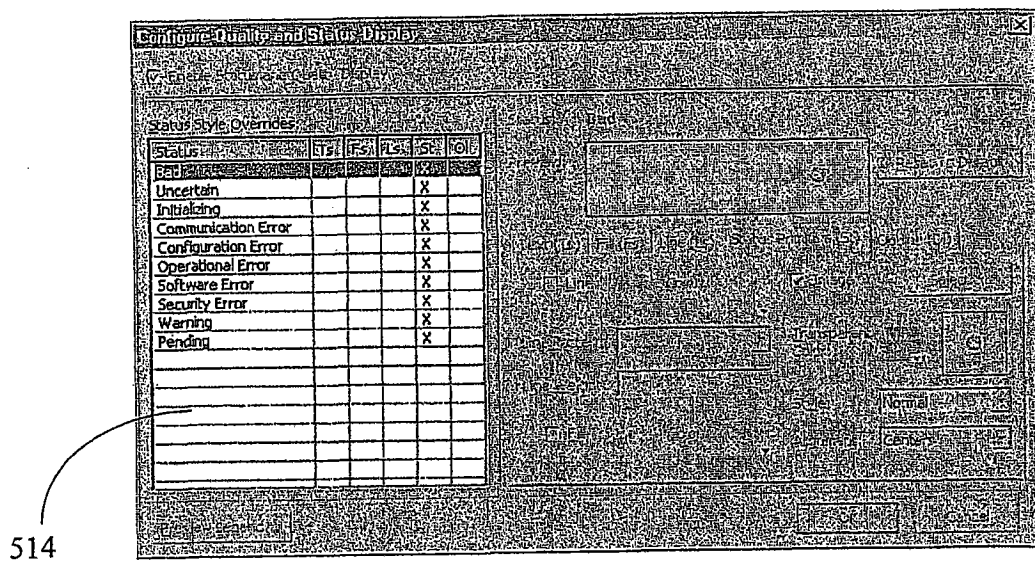


图 5c

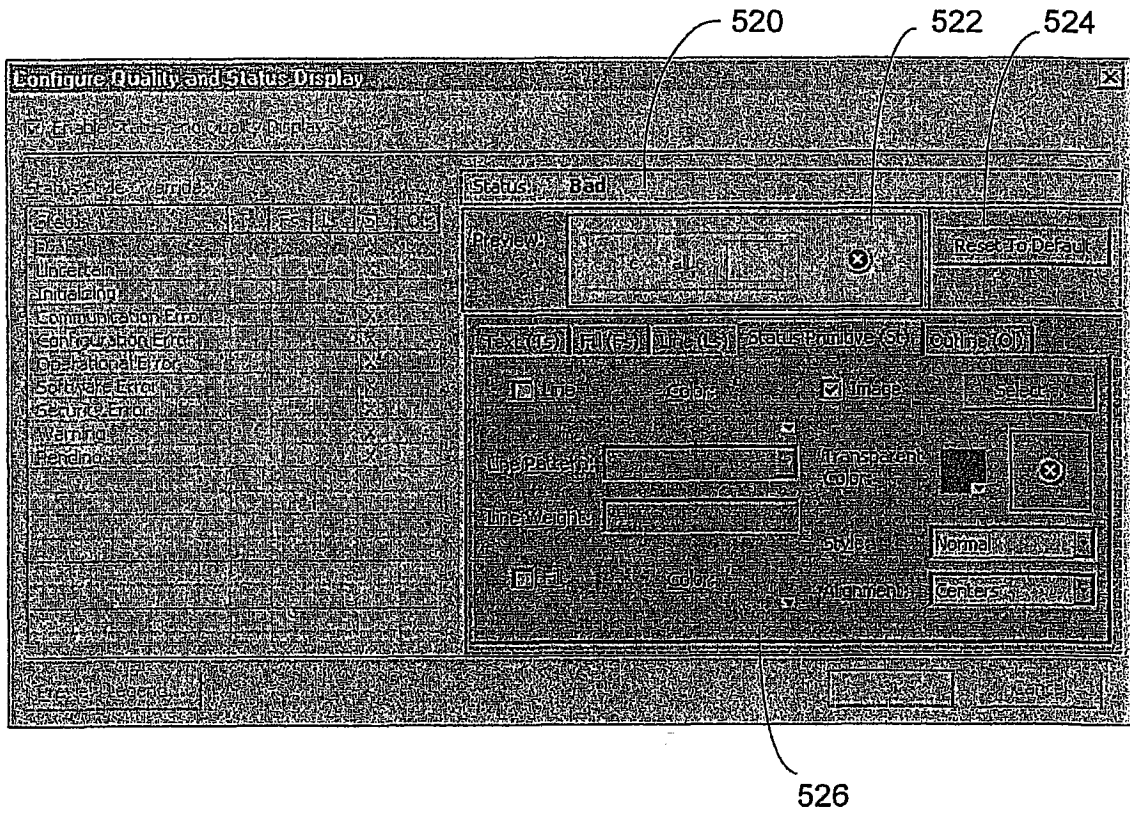


图 5d

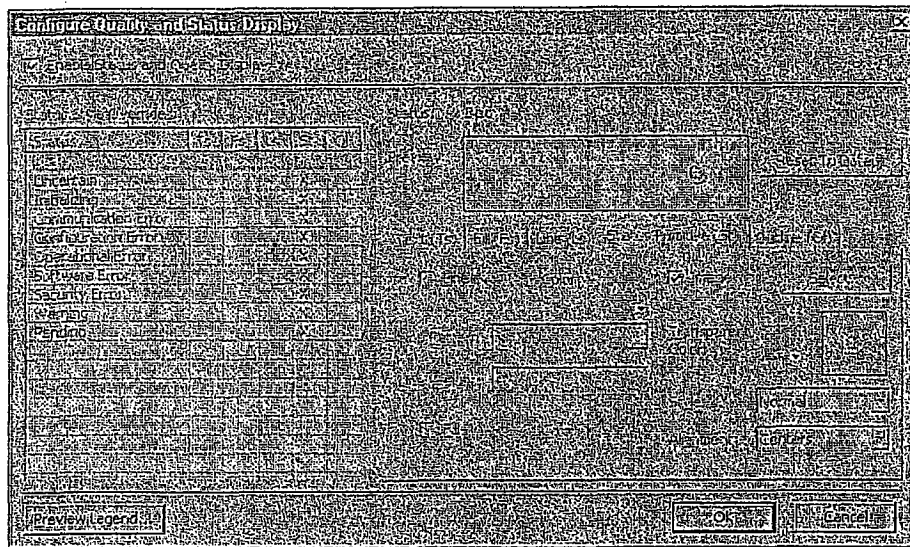


图 5e

状态	图标
不合格	
不确定	
初始化	
通信错误	
配置错误	
操作错误	
软件错误	
安全错误	
警告	
挂起	

图 6