



(12) 发明专利申请

(10) 申请公布号 CN 103616868 A

(43) 申请公布日 2014. 03. 05

(21) 申请号 201310612927.6

(74) 专利代理机构 北京市柳沈律师事务所

(22) 申请日 2004. 08. 18

11105

(30) 优先权数据

10/853,655 2004. 05. 25 US

代理人 周少杰

(51) Int. Cl.

G05B 19/418 (2006. 01)

(62) 分案原申请数据

200410103817.8 2004. 08. 18

(71) 申请人 费舍 - 柔斯芒特系统股份有限公司

地址 美国得克萨斯州

(72) 发明人 J. 迈克尔 . 卢卡斯 阿瑟 . 韦布

马克 . J. 尼克松 拉里 . O. 琼特

李健 丹尼斯 . L. 史蒂文森

迈克尔 . G. 奥特

赫谢尔 . O. 科斯卡第二

罗伯特 . B. 哈夫科斯特

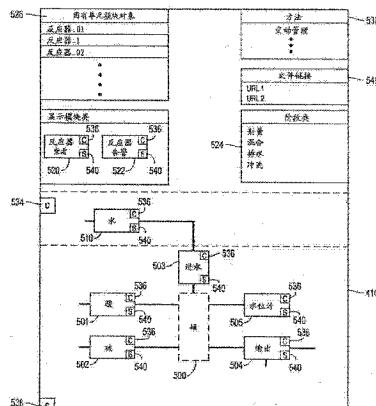
权利要求书7页 说明书36页 附图30页

(54) 发明名称

配置加工厂的方法和系统

(57) 摘要

本发明提供一种配置加工厂的方法和系统。加工厂项目可以表示或能够表示加工厂中的实体以帮助配置、组织和改变加工厂中的控制和显示行为。可通过将访问控制数据与所述项目相关联来控制对所述项目的访问。所述配置系统也可以使用对象，该对象表示或能够表示一个或多个由加工厂中的实体所执行的步骤。可通过将访问控制数据与所述对象相关联来控制对所述对象的访问。所述访问控制数据可以指示用户或特定用户是否能够，例如查看或修改与加工厂项目或所述对象相关联的全部或部分数据。所述加工厂项目可以包括例如，能够类属地表示加工厂的过程实体的模块类对象、能够明确表示处理工厂的过程实体的模块对象、复合模板、模块模板等。



1. 一种配置加工厂的方法,包括:

在耦接到加工厂的至少一个控制器和耦接到存储对应于加工厂的当前配置的数据的配置数据库的计算机上,提供用户接口以使得用户能够创建或改变面向对象编程语言中的对象,所述面向对象编程语言中的对象表示加工厂的过程实体或者表示要由加工厂中的过程实体执行的一个或更多步骤,并且所述对象用于配置加工厂的元件;

经由所述用户接口接收指示所述对象的一个或更多子元件要受到保护、以便对所述一个或更多子元件的访问要受到限制的输入;

设置指示所述对象的一个或更多子元件受到保护的、所述对象中包括的一个或更多安全参数;

在可访问所述计算机的模板对象的库中,存储包括一个或更多安全参数的对象;

经由所述用户接口接收访问所述对象的请求;

基于所述对象中包括的一个或更多安全参数的检查,确定对所述对象的子元件的适当的访问等级;以及

通过以下提供对所述对象的子元件的确定的适当的访问等级:

(i) 限制对所述对象的一个或更多受保护子元件的访问,以防止所述对象的一个或更多受保护子元件被修改;以及

(ii) 允许对所述对象的未受保护子元件的访问;

基于提供的、对所述对象的子元件的确定的适当的访问等级,接收对所述对象的至少一个未受保护子元件的修改的指示;

创建包括对至少一个未受保护子元件的所述修改的对象的实例;

使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的对象的实例配置,包括将包括所述修改的对象的实例下载到所述一个或更多特定元件的每个中;以及

使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视,所述一个或更多特定元件每个利用包括对未受保护子元件的至少一个的修改的对象的下载的实例配置和根据其操作。

2. 如权利要求1所述的方法,其中所述加工厂的至少部分包括所述加工厂的控制器或现场设备的至少一个,并且所述对象用于配置所述控制器或现场设备的至少一个。

3. 如权利要求1所述的方法,其中提供用户接口以使得用户能够创建或改变所述对象包括提供用户接口以使得用户能够创建或改变类对象,所述类对象能够一般地表示加工厂内的过程实体。

4. 如权利要求1所述的方法,其中提供用户接口以使得用户能够创建或改变所述对象包括提供用户接口以创建或改变对象,所述对象能够具体地表示加工厂内的一个或更多过程实体中包括的特定过程实体。

5. 如权利要求1所述的方法,其中提供用户接口以使得用户能够创建或改变所述对象包括提供用户接口以创建或改变对象,所述对象表示所述加工厂中的复合模板、模块模板、阶段类或单元阶段中的至少一个。

6. 如权利要求1所述的方法,进一步包括经由所述用户接口接收指示与所述对象的一个或更多子元件相关联的安全等级的用户输入数据,以及其中设置一个或更多安全参数包括基于与所述对象的一个或更多子元件相关联的安全等级设置一个或更多安全参数。

7. 如权利要求 6 所述的方法,其中指示安全等级的数据包括 :指示不允许访问的数据、指示不允许特定人的访问的数据、指示不允许特定类型的用户访问的数据、指示只读访问的数据、指示特定人的只读访问的数据或指示特定类型用户的只读访问的数据中的至少一个。

8. 如权利要求 1 所述的方法,其中经由所述用户接口接收指示所述对象的一个或更多子元件要受到保护的输入包括 :接收指示作为整体的对象要受到保护的数据。

9. 如权利要求 1 所述的方法,

其中接收访问所述对象的请求包括接收查看所述对象的子元件的请求 ;

其中确定适当的访问等级包括确定是否要隐藏任意子元件 ;以及

其中提供确定的适当的访问等级包括允许所述对象的未受保护子元件被查看,并且防止所述对象的受保护子元件被查看。

10. 如权利要求 1 所述的方法,其中确定适当的访问等级包括进一步基于与发起访问所述对象的请求的用户相关联的信息,确定适当的访问等级。

11. 如权利要求 1 所述的方法,其中确定适当的访问等级包括进一步基于用户的标识确定适当的访问等级。

12. 如权利要求 1 所述的方法,其中确定适当的访问等级包括进一步基于与所述用户相关联的安全访问等级确定适当的访问等级。

13. 如权利要求 1 所述的方法,进一步包括 :

提示用户输入密码 ;

接收由用户输入的密码 ;以及

比较由用户输入的密码和所存储的密码数据 ;

其中确定适当访问等级的步骤包括 :基于由用户输入的密码和所存储的密码数据的比较来确定适当的访问等级。

14. 如权利要求 1 所述的方法,其中经由所述用户接口接收访问所述对象的请求包括经由所述用户接口接收访问类对象的请求,所述类对象一般表示加工厂中的过程实体。

15. 如权利要求 1 所述的方法,其中经由所述用户接口接收访问所述对象的请求包括经由所述用户接口接收访问类对象的请求,所述类对象具体表示加工厂中的一个或更多过程实体中包括的特定过程实体。

16. 如权利要求 1 所述的方法,其中经由所述用户接口接收访问所述对象的请求包括经由所述用户接口接收访问对象的请求,所述对象表示所述加工厂中的复合模板、模块模板、阶段类或单元阶段中的至少一个。

17. 一种配置加工厂的方法,包括 :

在耦接到加工厂的至少一个控制器和耦接到存储对应于加工厂的当前配置的数据的配置数据库的计算机上,提供用户接口以使得用户能够创建或改变项目组,项目组中的每个项目对应于面向对象编程语言中的各自对象,各自对象表示以下的至少一个 :加工厂中的一个或更多过程实体,或者要由加工厂中的一个或更多过程实体执行的一个或更多步骤,并且各自对象用于配置加工厂的元件 ;

经由所述用户接口接收指示所述项目组的一个或更多项目要受到保护的输入 ;

设置指示所述项目组的一个或更多项目受到保护的、对应于所述项目组的一个或更多

项目的各自对象中包括的一个或更多安全参数；

在可访问所述计算机并且可用于创建所述加工厂的期望配置的模板对象的库中，存储对应于项目组的各自对象；

经由所述用户接口接收修改所述项目组的请求；

基于对应于项目组的各自对象中包括的一个或更多安全参数的检查，确定是否允许对所述项目组的修改；以及

只有当基于所述确定，确定修改所述项目组的请求对应于所述项目组的未受保护项目时，允许所述项目组被修改；

基于修改所述项目组的请求对应于所述项目组的未受保护项目的确定，接收对所述项目组的至少一个未受保护项目的修改的指示；

对于对应于所述至少一个未受保护项目的至少一个中包括的每个项目的每个各自对象，创建每个各自对象的各自实例，所述各自实例包括所述修改；

使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的各自实例配置，包括将包括修改的各自实例下载到所述一个或更多特定元件的每个中；以及

使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视，所述一个或更多特定元件每个利用包括所述修改的下载的各自实例配置和根据其操作。

18. 如权利要求 17 所述的方法，其中对应于所述项目组的至少一个各自对象包括类对象，所述类对象能够一般地表示加工厂内的过程实体。

19. 如权利要求 17 所述的方法，其中对应于所述项目组的至少一个各自对象包括对象，所述对象能够具体地表示加工厂内的一个或更多过程实体中包括的特定过程实体。

20. 如权利要求 17 所述的方法，其中对应于所述项目组的至少一个各自对象表示所述加工厂中的复合模板、模块模板、阶段类或单元阶段中的至少一个。

21. 如权利要求 17 所述的方法，其中指示所述项目组的一个或更多项目要受到保护的输入包括：指示只读访问要提供给所述项目组中的每个项目的数据。

22. 如权利要求 17 所述的方法，其中指示所述项目组的一个或更多项目要受到保护的输入包括：指示一个或更多项目不能从所述项目组移除的数据。

23. 如权利要求 17 所述的方法，其中指示所述项目组的一个或更多项目要受到保护的输入包括：指示一个或更多项目不能增加到所述项目组的数据。

24. 如权利要求 17 所述的方法，进一步包括经由所述用户接口接收指示与所述项目组相关联的安全等级的用户输入数据。

25. 如权利要求 24 所述的方法，其中指示安全等级的用户输入数据包括以下的至少一个：指示对特定人仅允许只读访问的数据、指示对特定类型用户只允许只读访问的数据、或者指示对与指示仅允许只读访问的用户安全数据相关联的用户仅允许只读访问的数据。

26. 如权利要求 25 所述的方法，其中用户安全数据包括用户输入的密码。

27. 如权利要求 17 所述的方法，其中接收修改所述项目组的请求包括接收修改对应于所述项目组的对象组的请求。

28. 如权利要求 27 所述的方法，其中所述对象组中的至少一个对象一般表示加工厂中的对应过程实体。

29. 如权利要求 27 所述的方法，其中所述对象组中的至少一个对象具体表示加工厂中

的对应过程实体。

30. 如权利要求 27 所述的方法,其中所述对象组中的至少一个对象表示加工厂中的复合模板、模块模板、阶段类或单元阶段中的至少一个。

31. 如权利要求 27 所述的方法,其中接收修改所述对象组的请求包括以下至少一个 :
接收修改所述对象组中包括的一个或更多对象的请求 ;
接收从所述对象组移除一个或更多对象的请求 ;或者
接收向所述对象组增加一个或更多附加对象的请求。

32. 如权利要求 17 所述的方法,其中确定是否允许所述项目组的修改包括 :进一步基于与发起修改所述项目组的请求的用户相关联的信息,确定是否允许所述项目组的修改。

33. 如权利要求 32 所述的方法,其中与用户相关联的信息包括指示发起修改所述项目组的请求的用户的标识的数据。

34. 如权利要求 32 所述的方法,其中与发起修改所述项目组的请求的用户相关联的信息包括 :指示与发起修改所述项目组的请求的用户相关联的安全访问等级的数据。

35. 如权利要求 17 所述的方法,进一步包括 :

提示用户输入密码 ;
接收由用户输入的密码 ;
比较由用户输入的密码和所存储的密码数据 ;
其中确定是否允许所述项目组的修改包括 :基于由用户输入的密码和所存储的密码数据的比较来确定是否允许所述项目组的修改。

36. 如权利要求 17 所述的方法,其中确定是否允许所述项目组的修改包括确定以下至少一个 :是否不允许用户修改所述项目组、是否允许用户修改所述项目组、是否允许用户仅修改所述项目组的某些项目、是否对用户隐藏所述项目组、或者是否对用户仅隐藏所述项目组的某些项目。

37. 如权利要求 17 所述的方法,进一步包括 :

经由所述用户接口接收输出一个或更多对象的请求,输出请求中指示的每个对象表示所述项目组中的至少一个项目 ;

对于请求输出的每个对象,检查与所述对象相关联的各自安全参数,以确定是否要保护对所述对象的数据的访问 ;

对于请求输出的每个对象,如果要保护对所述对象的数据的访问,则加密每个对象的所述数据以生成加密数据,并且用加密数据替代每个对象的数据 ;以及

输出一个或更多对象。

38. 如权利要求 37 所述方法,其中请求输出的一个或者更多对象中的至少一个对象一般地表示加工厂中的对应过程实体。

39. 如权利要求 37 所述的方法,其中请求输出的一个或者更多对象中的至少一个对象具体表示加工厂中的对应过程实体。

40. 如权利要求 37 所述的方法,其中要输出的一个或者更多对象中的至少一个对象表示加工厂中的复合模板。

41. 如权利要求 37 所述的方法,进一步包括 :

对于请求输出的每个对象,如果要保护对所述每个对象的数据的访问,则加密与所述

每个对象相关联的访问安全数据，并且用加密的访问安全数据替代访问安全数据。

42. 一种用于方便配置加工厂的配置系统，包括：

一个或者更多计算机可读介质；

与一个或者更多计算机可读介质相通信的一个或者更多处理器；

用于存储面向对象编程语言中的对象的数据库，所述对象对应于表示以下至少一个的功能块：加工厂内的一个或更多过程实体，或者要由加工厂内的一个或多个过程实体所执行的一个或多个步骤；以及

存储在一个或者更多计算机可读介质上的一个或者更多配置例程，被配置以由所述一个或者更多处理器执行以：

提供用户接口以使得用户能够创建或改变所述对象；

经由所述用户接口接收指示所述对象的一个或更多子元件要受到保护、以便对所述一个或更多子元件的访问要受到限制的输入；

设置指示所述对象的一个或更多子元件受到保护的、所述对象中包括的一个或更多安全参数；

在可访问所述一个或者更多配置例程的模板对象的库中，存储包括一个或更多安全参数的对象；

接收访问所述对象的请求；

基于与所述对象相关联的一个或更多安全参数的检查，确定对所述对象的子元件的适当的访问等级；以及

通过以下提供对所述对象的子元件的确定的适当的访问等级：

(i) 限制对所述对象的一个或更多受保护子元件的访问，以防止所述对象的一个或更多受保护子元件被修改；以及

(ii) 允许对所述对象的未受保护子元件的访问；

基于提供的、对所述对象的子元件的确定的适当的访问等级，接收对所述对象的至少一个未受保护子元件的修改的指示；

创建包括对至少一个未受保护子元件的所述修改的对象的实例；

使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的对象的实例配置，包括将包括所述修改的对象的实例下载到所述一个或更多特定元件的每个中；以及

使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视，所述一个或更多特定元件利用包括对未受保护子元件的至少一个的修改的对象的下载的实例配置和根据其操作。

43. 一种用于方便配置加工厂的配置系统，包括：

一个或更多计算机可读介质；

与一个或更多计算机可读介质相通信的一个或更多处理器；

用于存储面向对象编程语言中的各对象的数据库，每个对象对应于项目组的各自项目并且表示以下的至少一个：加工厂内的一个或更多过程实体，或者要由加工厂内的一个或更多过程实体执行的一个或多个步骤；以及

存储在一个或更多计算机可读介质上的一个或更多配置例程，被配置以由一个或更多处理器来执行；

提供用户接口以使得用户能够创建或改变所述项目组；

经由所述用户接口接收指示所述项目组的一个或更多项目要受保护、以便对所述一个或更多项目的访问要受到限制的输入；以及

设置指示所述项目组的一个或更多项目受到保护的、对应于所述项目组的一个或更多项目的各自对象中包括的一个或更多安全参数；

经由所述用户接口接收修改所述项目组的请求；

基于对应于项目组的对象中包括的一个或更多安全参数的检查，确定是否允许对所述项目组的修改；以及

只有当确定所述项目组的修改对应于所述项目组的未受保护项目时，允许所述项目组被修改；

基于所述项目组的修改对应于所述项目组的未受保护项目的确定，接收对所述项目组的至少一个未受保护项目的修改的指示；

对于对应于所述至少一个未受保护项目中包括的每个项目的每个对象，创建每个对象的各自实例，所述各自实例包括所述修改；

使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的每个对象的各自实例配置，包括将包括所述修改的每个对象的各自实例下载到所述一个或更多特定元件的每个中；以及

使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视，所述一个或更多特定元件每个利用包括修改的下载的每个对象的各自实例配置和根据其操作。

44. 如权利要求 43 所述的系统，其中经由以下的至少一个接收修改项目组的请求：与配置系统相关联的用户接口、与调试系统相关联的用户接口、与监视系统相关联的用户接口、或者与资产管理系统相关联的用户接口。

45. 如权利要求 44 所述的配置系统，其中所述一个或更多配置例程进一步配置为由所述一个或更多处理器执行：

经由所述用户接口接收输出一个或更多对象的请求，请求输出的一个或更多对象的每个对应于所述项目组中的各自项目；以及

对于请求输出的每个对象：

检查与每个对象相关联的各自安全参数，以确定是否要保护对每个对象的数据的访问；

如果要保护对每个对象的数据的访问，则加密所述每个对象的数据以生成每个对象的加密数据，并且用每个对象的加密数据替代所述每个对象的数据；以及

输出一个或更多对象。

46. 一种用于方便配置加工厂的系统，包括：

用于存储面向对象编程语言中的一个或多个对象的数据库，所述对象对应于表示以下至少一个的功能块：加工厂内的一个或多个过程实体，或者要由加工厂内的一个或多个过程实体所执行的一个或多个步骤，一个或多个对象包括一个或更多子元件；以及

用于以下的部件：

经由用户接口接收指示所述数据库中存储的对象的一个或更多子元件要受到保护、以便对所述一个或更多子元件的访问要受到限制的用户输入；

设置指示所述对象的一个或更多子元件受到保护的、所述对象中包括的一个或更多安全参数；

经由用户接口接收访问所述对象的请求；

基于所述对象中包括的一个或更多安全参数的检查，确定对所述对象的子元件的适当的访问等级；以及

通过以下提供对所述对象的子元件的确定的适当的访问等级：

(i) 限制对所述对象的一个或更多受保护子元件的访问，以防止所述对象的一个或更多受保护子元件被修改；以及

(ii) 允许对所述对象的未受保护子元件的访问；

基于提供的、对所述对象的子元件的确定的适当的访问等级，接收对所述对象的至少一个未受保护子元件的修改的指示；

创建包括对至少一个未受保护子元件的所述修改的对象的实例；

使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的对象的实例配置，包括将包括所述修改的对象的实例下载到所述一个或更多特定元件的每个中；以及

使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视，所述一个或更多特定元件根据包括对未受保护子元件的至少一个的修改的对象的下载的实例配置和操作。

47. 如权利要求 46 所述的系统，其中所述用户接口是以下的至少一个：与配置系统相关联的用户接口、与调试系统相关联的用户接口、与监视系统相关联的用户接口、或者与资产管理系统相关联的用户接口。

48. 如权利要求 46 所述的系统，其中限制对所述对象的受保护子元件的访问包括通过查看或者配置应用限制对所述对象的受保护子元件的访问。

49. 如权利要求 46 所述的系统，其中：

所述对象的一个或多个子元件的保护进一步包括对应于输出所述对象的子元件的限制；以及

当确定在输出所述对象期间要限制对所述对象的受保护子元件的访问时，所述系统进一步包括以下部件：

加密与所述对象的受保护子元件相关联的数据，

用加密数据替代与所述对象的受保护子元件相关联的数据，以及

输出所述对象。

50. 如权利要求 46 所述的系统，其中所述对象是以下的至少一个：

(i) 能够一般地表示加工厂内的过程实体的类对象；

(ii) 能够具体地表示加工厂内的一个或更多过程实体中包括的特定过程实体的对象；

或

(iii) 加工厂中的复合模板、模块模板、阶段类或单元阶段。

配置加工厂的方法和系统

[0001] 本申请是申请日为 2004 年 8 月 18 日、申请号为 200410103817.8、发明名称为“加工厂配置系统中的对象的安全性”的专利申请的分案申请。

[0002] 相关申请的交叉参考

[0003] 本申请是 2003 年 2 月 18 日提交的申请号为 10/368,151、题为“加工厂配置系统中的模块类对象”的美国专利申请的部分继续，为了所有目的，其中公开的全部内容在此包含作为参考。

[0004] 本申请涉及与本申请同时提交的 No. _____ (代理人卷号为 No. 06005/40129)，题为“加工厂配置系统中的对象的版本控制”的共同未决 (copending) 的美国专利申请，为了所有目的，其中公开的全部内容在此包含引作参考。

技术领域

[0005] 本发明通常涉及加工厂 (process plant)，特别是涉及配置加工厂的方法和系统。

背景技术

[0006] 诸如用于化学、石油或其它过程中的那些分布式过程控制系统，一般包括一个或多个过程控制器，所述控制器通过模拟、数字或组合的模拟 / 数字总线通信地耦合到一个或多个现场设备，所述现场设备可以是例如阀门、阀门定位器、开关和变送器（如温度、压力、水位和流速传感器），它们位于过程环境中，并且执行过程功能，诸如打开或关闭阀门、测量过程参数等。智能现场设备，诸如符合众所周知的 Fieldbus (现场总线) 协议的现场设备，也可以执行控制计算、报警功能以及其它通常在控制器中实现的控制功能。过程控制器典型地位于工厂环境中，它接收表示由现场设备进行的过程测试的信号和 / 或与现场设备有关的其它信息，并且执行运行例如不同的控制模块的控制器应用程序，所述控制模块作出过程控制决定，根据所接收的信息来产生控制信号，并且与在诸如 HART 和 Fieldbus 现场设备的现场设备中被执行的控制模块或部件协调。控制器中的控制模块通过通信线路向现场设备发送控制信号以便控制加工厂的运行。

[0007] 来自现场设备和控制器的信息通常通过数据总线提供给一个或多个其它硬件器件，如操作员工作站、个人计算机、数据历史器 (data historian)、报告产生器、中央数据库等，它们一般位于远离较为恶劣的工厂环境的控制室或其它位置中。这些硬件设备运行应用程序，例如可以使操作员执行与过程相关的功能，诸如改变过程控制程序的设置、修改控制器或现场设备中的控制模块的操作、查看过程的当前状态、查看由现场设备和控制器产生的警报、为了培训人员或测试过程控制软件而模拟过程的操作、保持和更新配置数据库，等等。

[0008] 作为一个实例，由 Fisher-Rosemount Systems 公司出售的 DeltaV™ 控制系统，包括多个存储在不同设备中并由这些不同设备执行的应用程序，所述设备位于加工厂内的不同位置。常驻在一个或多个操作员工作站中的配置应用程序，使得用户能够创建或改变过程控制模块以及经由数据总线向专门的分布式控制器下载这些过程控制模块。典型地，这

些控制模块由通信互连的功能块组成,所述功能块是面向对象的编程协议中的对象,它们根据其输入来执行控制方案图 (control scheme) 中的功能,并且它们为控制模式图中的其他功能块提供输出。所述配置应用程序还允许配置设计者创建或改变操作员界面,其中所述操作员界面由查看应用程序使用以便为操作员显示数据,以及允许操作员改变过程控制程序中的设置,如设定点。每个专用的控制器和在某些情况下的现场设备存储并执行控制器应用程序,所述控制器应用程序运行向其分配和下载的控制模块,以便实现实际的过程控制功能。可以在一个或多个操作工作站上运行的查看应用程序,通过数据总线从控制器应用程序接收数据,并使用用户界面将这些数据显示给过程控制系统的设计者、操作员或用户,并可提供任意数量的不同视图,如操作员视图、工程师视图和技术员视图等。数据历史应用程序一般存储在数据历史设备中并由它执行,其中所述数据历史设备收集和存储通过数据总线提供的数据的部分或全部,而配置数据库应用程序可以在连接到数据总线的另一个计算机中运行,以便存储当前过程控制程序配置及其相关数据。可替换地,所述配置数据库可位于与所述配置应用程序相同的工作站中。

[0009] 目前,配置应用程序可以包括模板对象库,如功能块模板对象以及在某种情况下的控制模块模板对象。这些配置应用程序用于为加工厂配置控制策略 (control strategy)。所述模板对象都具有默认的属性、设置和与此相关的方法,并且使用所述配置程序工程师能够选择这些模板对象,并在实际上将选定模板对象的副本放到配置屏幕中来开发一种控制模块。在选择模板对象并将其放到配置屏幕中的过程中,工程师将这些对象的输入和输出进行互连,并改变它们的参数、名称、标签和其他属性,以便创建特定控制模块,用于加工厂中的特定用途。在创建一个或多个这样的控制模块之后,然后工程师能够例示该控制模块,并将控制模块下载到适当的控制器或用于在加工厂运行过程中执行的控制器和现场设备中。

[0010] 此后,工程师一般通过在显示创建应用程序中选择和构造显示对象,为加工厂内的操作员、维护人员等创建一个或多个显示。典型地,在一个或多个工作站中,以系统范围基础实现这些显示,并将这些预先配置的显示提供给关注工厂内的控制系统或设备的运行状态的操作员、维护人员。典型地,这些显示采取的形式有:报警显示,该报警显示接收和显示加工厂中的控制器或设备所产生的警报;控制显示,用于指示加工厂中控制器和其他设备的运行状态;以及维护显示,指示加工厂内设备的功能状态,等等。通常以公知的方式预先配置这些显示,用于显示从加工厂中的过程控制模块或设备接收到的信息或数据。在一些已知的系统中,可通过使用对象创建这些显示,其中所述对象具有与物理或逻辑元件相关的图形,并且与物理或逻辑元件通信地连接以接收关于物理或逻辑元件的数据。所述对象可根据接收的数据改变显示屏上的图形,以图解例如油罐半满,和图解流量传感器所测量的流量,等等。

[0011] 与所述控制配置应用程序相类似,显示创建应用程序具有模板图形显示项,如油罐、阀门、传感器、类似滑动条的操作员控制按钮、开 / 关开关等,这些显示项可以以任何期望的配置放置在屏幕上,以创建操作员显示、维护显示等。当各个图形项目被置于所述屏幕上时,它们可能以如下方式在屏幕上互连,即所述方式为不同用户提供加工厂内的内部工作的一些信息或显示。但是,为了绘制所述图形显示,显示创建者必须通过规定图形项和加工厂内的相关数据源之间的通信链路,来手动地连接每一个图形项目和加工厂内产生的数

据,诸如传感器所测量的数据或表示阀门位置的数据等。该过程单调乏味、耗时并可能充满差错。

[0012] 当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方便时,这是因为它们能够被复制和用于创建许多不同的控制模块和图形显示,此时通常需要为加工厂内的不同设备创建大量相同的控制模块和图形显示。例如,许多中型到大型的加工厂具有大量的相同或相似设备的情况,其中所述设备可以使用基本常用的控制模块和显示控制和查看。但是,为了创建这些大量的控制模块和显示,要创建一个常规控制模块或显示模块,然后为模块所应用的设备的每个不同部分复制该常规被控制或显示模块。当然,在复制模块之后,必须在应用程序中手动地改变每个新的被控制或显示模块以指定模块所连接的特定设备,并且然后必须例示所有的这些控制和显示模块并将它们下载至所述过程控制系统。

[0013] 令人遗憾的是,没有以任何方式模块化上述的控制模块和显示项目。因此,在复制以后,必须使用适当的配置应用程序来手动并逐个地改变每个控制模块和显示以指定工厂内与它们相关联的设备。在具有许多相同类型的设备副本的设备(也就是,被复制设备)中,该过程乏味、耗时且充满操作员所引入的差错。更进一步,一旦编程这些不同的控制模块和显示,它们将不会彼此知道对方。所以,为了改变以前创建的控制模块,工程师或操作员必须手动地为不同的复制设备对每个不同的控制模块进行相同的改变,而该过程同样也是耗时和乏味的。相同的问题也适用于为工厂内不同组复制设备创建的图形视图。换句话说,一旦创建了特定控制模块或特定图形视图(逐个地或通过复制模板对象),则将它与工厂内特定的一组设备相连接,以及该控制模块或图形视图作为系统内的单独实体或对象而存在,而没有自动认识与其相同或类似的其他控制模块或图形显示。因此,必须对这些模块和显示逐个地进行适用于每个特定类型的控制模块和图形显示的改变。

[0014] 更进一步,由于每个控制模块和显示是单独的对象,因此它必须是开放的,意思就是说它所有的内部参数、视图、功能块以及其他单元对于任何用户来说必须都是可用的以便对它们进行改变、查看等。目前,没有任何方式来控制操作员或其他用户在这些控制模块和显示中所能够查看或访问的内容,因此,也没有能力来隐藏这些控制模块和显示的某些单元,诸如专用软件和方法、来自控制模块和显示的用户的报警动作等。

发明内容

[0015] 一种配置系统,所述系统使用加工厂项目,所述加工厂项目可以表示或能够表示加工厂中的实体以帮助配置、组织和改变加工厂中的控制和显示动作。通过将访问控制数据与所述项目相关联,可以控制对所述项目的访问。所述配置系统也可以使用对象,这些对象表示或能够表示一个或多个由加工厂中的实体所执行的步骤。通过将访问控制数据与所述对象相关联,可以控制对这些对象的访问。所述访问控制数据可以指示用户或某些用户是否能够,例如查看或修改与加工厂项目或所述对象相关联的全部或部分数据。所述加工厂项目可以包括,例如,能够类属地表示加工厂的过程实体的模块类对象、能够具体表示加工厂的过程实体的模块对象、复合模板、模块模板等。表示或能够表示一个或多个加工厂中的实体所执行步骤的对象可包括,例如阶段类(phase class)或单元阶段(unit phase)。

[0016] 根据本发明另一实施例,提供了一种配置加工厂的方法,包括:

- [0017] 在耦接到加工厂的至少一个控制器和耦接到存储对应于加工厂的当前配置的数据的配置数据库的计算机上,提供用户接口以使得用户能够创建或改变面向对象编程语言中的对象,所述面向对象编程语言中的对象表示加工厂的过程实体或者表示要由加工厂中的过程实体执行的一个或更多步骤,并且所述对象用于配置加工厂的元件;
- [0018] 经由所述用户接口接收指示所述对象的一个或更多子元件要受到保护、以便对所述一个或更多子元件的访问要受到限制的输入;
- [0019] 设置指示所述对象的一个或更多子元件受到保护的、所述对象中包括的一个或更多安全参数;
- [0020] 在可访问所述计算机的模板对象的库中,存储包括一个或更多安全参数的对象;
- [0021] 经由所述用户接口接收访问所述对象的请求;
- [0022] 基于所述对象中包括的一个或更多安全参数的检查,确定对所述对象的子元件的适当的访问等级;以及
- [0023] 通过以下提供对所述对象的子元件的确定的适当的访问等级:
- [0024] (i) 限制对所述对象的一个或更多受保护子元件的访问,以防止所述对象的一个或更多受保护子元件被修改;以及
- [0025] (ii) 允许对所述对象的未受保护子元件的访问;
- [0026] 基于提供的、对所述对象的子元件的确定的适当的访问等级,接收对所述对象的至少一个未受保护子元件的修改的指示;
- [0027] 创建包括对至少一个未受保护子元件的所述修改的对象的实例;
- [0028] 使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的对象的实例配置,包括将包括所述修改的对象的实例下载到所述一个或更多特定元件的每个中;以及
- [0029] 使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视,所述一个或更多特定元件每个利用包括对未受保护子元件的至少一个的修改的对象的下载的实例配置和根据其操作。
- [0030] 根据本发明另一实施例,提供了一种配置加工厂的方法,包括:
- [0031] 在耦接到加工厂的至少一个控制器和耦接到存储对应于加工厂的当前配置的数据的配置数据库的计算机上,提供用户接口以使得用户能够创建或改变项目组,项目组中的每个项目对应于面向对象编程语言中的各自对象,各自对象表示以下的至少一个:加工厂中的一个或更多过程实体,或者要由加工厂中的一个或更多过程实体执行的一个或更多步骤,并且各自对象用于配置加工厂的元件;
- [0032] 经由所述用户接口接收指示所述项目组的一个或更多项目要受到保护的输入;
- [0033] 设置指示所述项目组的一个或更多项目受到保护的、对应于所述项目组的一个或更多项目的各自对象中包括的一个或更多安全参数;
- [0034] 在可访问所述计算机并且可用于创建所述加工厂的期望配置的模板对象的库中,存储对应于项目组的各自对象;
- [0035] 经由所述用户接口接收修改所述项目组的请求;
- [0036] 基于对应于项目组的各自对象中包括的一个或更多安全参数的检查,确定是否允许对所述项目组的修改;以及

- [0037] 只有当基于所述确定,确定修改所述项目组的请求对应于所述项目组的未受保护项目时,允许所述项目组被修改;
- [0038] 基于修改所述项目组的请求对应于所述项目组的未受保护项目的确定,接收对所述项目组的至少一个未受保护项目的修改的指示;
- [0039] 对于对应于所述至少一个未受保护项目的至少一个中包括的每个项目的每个各自对象,创建每个各自对象的各自实例,所述各自实例包括所述修改;
- [0040] 使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的各自实例配置,包括将包括修改的各自实例下载到所述一个或更多特定元件的每个中;以及
- [0041] 使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视,所述一个或更多特定元件每个利用包括所述修改的下载的各自实例配置和根据其操作。
- [0042] 根据本发明另一实施例,提供了一种用于方便配置加工厂的配置系统,包括:
- [0043] 一个或者更多计算机可读介质;
- [0044] 与一个或者更多计算机可读介质相通信的一个或者更多处理器;
- [0045] 用于存储面向对象编程语言中的对象的数据库,所述对象对应于表示以下至少一个的功能块:加工厂内的一个或更多过程实体,或者要由加工厂内的一个或多个过程实体所执行的一个或多个步骤;以及
- [0046] 存储在一个或者更多计算机可读介质上的一个或者更多配置例程,被配置以由所述一个或者更多处理器执行以:
- [0047] 提供用户接口以使得用户能够创建或改变所述对象;
- [0048] 经由所述用户接口接收指示所述对象的一个或更多子元件要受到保护、以便对所述一个或更多子元件的访问要受到限制的输入;
- [0049] 设置指示所述对象的一个或更多子元件受到保护的、所述对象中包括的一个或更多安全参数;
- [0050] 在可访问所述一个或者更多配置例程的模板对象的库中,存储包括一个或更多安全参数的对象;
- [0051] 接收访问所述对象的请求;
- [0052] 基于与所述对象相关联的一个或更多安全参数的检查,确定对所述对象的子元件的适当的访问等级;以及
- [0053] 通过以下提供对所述对象的子元件的确定的适当的访问等级:
- [0054] (i) 限制对所述对象的一个或更多受保护子元件的访问,以防止所述对象的一个或更多受保护子元件被修改;以及
- [0055] (ii) 允许对所述对象的未受保护子元件的访问;
- [0056] 基于提供的、对所述对象的子元件的确定的适当的访问等级,接收对所述对象的至少一个未受保护子元件的修改的指示;
- [0057] 创建包括对至少一个未受保护子元件的所述修改的对象的实例;
- [0058] 使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的对象的实例配置,包括将包括所述修改的对象的实例下载到所述一个或更多特定元件的每个中;以及
- [0059] 使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视,所

述一个或更多特定元件利用包括对未受保护子元件的至少一个的修改的对象的下载的实例配置和根据其操作。

- [0060] 根据本发明另一实施例，提供了一种用于方便配置加工厂的配置系统，包括：
 - [0061] 一个或更多计算机可读介质；
 - [0062] 与一个或更多计算机可读介质相通信的一个或更多处理器；
 - [0063] 用于存储面向对象编程语言中的各对象的数据库，每个对象对应于项目组的各自项目并且表示以下的至少一个：加工厂内的一个或更多过程实体，或者要由加工厂内的一个或更多过程实体执行的一个或多个步骤；以及
 - [0064] 存储在一个或更多计算机可读介质上的一个或更多配置例程，被配置以由一个或更多处理器来执行；
 - [0065] 提供用户接口以使得用户能够创建或改变所述项目组；
 - [0066] 经由所述用户接口接收指示所述项目组的一个或更多项目要受保护、以便对所述一个或更多项目的访问要受到限制的输入；以及
 - [0067] 设置指示所述项目组的一个或更多项目受到保护的、对应于所述项目组的一个或更多项目的各自对象中包括的一个或更多安全参数；
 - [0068] 经由所述用户接口接收修改所述项目组的请求；
 - [0069] 基于对应于项目组的对象中包括的一个或更多安全参数的检查，确定是否允许对所述项目组的修改；以及
 - [0070] 只有当确定所述项目组的修改对应于所述项目组的未受保护项目时，允许所述项目组被修改；
 - [0071] 基于所述项目组的修改对应于所述项目组的未受保护项目的确定，接收对所述项目组的至少一个未受保护项目的修改的指示；
 - [0072] 对于对应于所述至少一个未受保护项目中包括的每个项目的每个对象，创建每个对象的各自实例，所述各自实例包括所述修改；
 - [0073] 使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的每个对象的各自实例配置，包括将包括所述修改的每个对象的各自实例下载到所述一个或更多特定元件的每个中；以及
 - [0074] 使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视，所述一个或更多特定元件每个利用包括修改的下载的每个对象的各自实例配置和根据其操作。
- [0075] 根据本发明另一实施例，提供了一种用于方便配置加工厂的系统，包括：
 - [0076] 用于存储面向对象编程语言中的一个或多个对象的数据库，所述对象对应于表示以下至少一个的功能块：加工厂内的一个或多个过程实体，或者要由加工厂内的一个或多个过程实体所执行的一个或多个步骤，一个或多个对象包括一个或更多子元件；以及
 - [0077] 用于以下的部件：
 - [0078] 经由用户接口接收指示所述数据库中存储的对象的一个或更多子元件要受到保护、以便对所述一个或更多子元件的访问要受到限制的用户输入；
 - [0079] 设置指示所述对象的一个或更多子元件受到保护的、所述对象中包括的一个或更多安全参数；

- [0080] 经由用户接口接收访问所述对象的请求；
- [0081] 基于所述对象中包括的一个或更多安全参数的检查,确定对所述对象的子元件的适当的访问等级;以及
- [0082] 通过以下提供对所述对象的子元件的确定的适当的访问等级：
 - [0083] (i) 限制对所述对象的一个或更多受保护子元件的访问,以防止所述对象的一个或更多受保护子元件被修改;以及
 - [0084] (ii) 允许对所述对象的未受保护子元件的访问；
- [0085] 基于提供的、对所述对象的子元件的确定的适当的访问等级,接收对所述对象的至少一个未受保护子元件的修改的指示；
- [0086] 创建包括对至少一个未受保护子元件的所述修改的对象的实例；
- [0087] 使得加工厂的至少部分中包括的一个或更多特定元件用包括所述修改的对象的实例配置,包括将包括所述修改的对象的实例下载到所述一个或更多特定元件的每个中;以及
- [0088] 使得所述加工厂的至少部分通过使用所述一个或更多特定元件被控制或监视,所述一个或更多特定元件根据包括对未受保护子元件的至少一个的修改的对象的下载的实例配置和操作。
- [0089] 配置系统、查看系统、调试系统、运行监控系统、资产管理系统等,可以检查与项目、项目组、对象、对象组等相关联的访问控制数据,以便确定是否要限制对与所述项目或对象相关联的数据的访问。而且,能够例如加密被保护的数据以阻止配置系统、查看系统、调试系统、运行监控系统、资产管理系统等查看被保护的数据。

附图说明

- [0090] 图 1 是位于加工厂内的分布式过程控制网络的方框图,该控制网络包括执行配置应用程序的操作员工作站,所述配置应用程序使用模块类对象为所述加工厂配置控制和显示动作；
- [0091] 图 2 是图 1 的反应器单元的图；
- [0092] 图 3 是图 2 的反应器单元中使用的累加器设备实体的图；
- [0093] 图 4 是图 2 的反应器单元中使用的输出阀门系统的图；
- [0094] 图 5 是示出模块类对象和用于模块类对象类型的单元、设备、控制和显示类型的相关联模块对象之间的相互关系的逻辑图；
- [0095] 图 6 是反应器单元模块类对象的逻辑图,它能够用于执行图 1 工厂内的反应器的配置动作；
- [0096] 图 7 是一种累加器设备模块类对象的逻辑图,它能够用于执行图 1 工厂内的累加器的配置动作；
- [0097] 图 8 示出第一配置屏幕,所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；
- [0098] 图 9 示出第二配置屏幕,所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；
- [0099] 图 10 示出第三配置屏幕,所述屏幕可被配置操作员用于通过使用模块类对象来

配置加工厂；

[0100] 图 11 示出第四配置屏幕，所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；

[0101] 图 12 示出第五配置屏幕，所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；

[0102] 图 13 示出第六配置屏幕，所述屏幕可被配置操作员用于通过使用模块类对象来设置加工厂；

[0103] 图 14 示出第七配置屏幕，所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；

[0104] 图 15 示出第八配置屏幕，所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；

[0105] 图 16 示出第九配置屏幕，所述屏幕可被配置操作员用于通过使用模块类对象来配置加工厂；

[0106] 图 17 是用于配置加工厂项目以限制访问该项目的示例性程序的流程图；

[0107] 图 18A 是示出复合模板的、包括功能块的子元件的示例性屏幕显示；

[0108] 图 18B 是示出图 18A 的复合模板的子元件的示例性屏幕显示，但在其中隐藏了功能块子元件；

[0109] 图 19 是用于配置一组加工厂项目以限制访问该组的示例性程序的流程图；

[0110] 图 20 是用于确定是否允许查看与加工厂相关联的数据的示例性程序的流程图；

[0111] 图 21 是用于确定是否允许修改加工厂项目或一组加工厂项目的示例性程序的流程图；

[0112] 图 22 是用于输出加工厂项目的示例性程序的流程图；

[0113] 图 23 是示例性配置系统的方框图，所述系统可被用于提供版本控制和与加工厂相关联的跟踪记录信息 (audit trail information)；

[0114] 图 24 是用于方便跟踪加工厂项目版本的示例性程序的流程图；

[0115] 图 25 是用于方便跟踪加工厂项目版本的示例性程序的流程图；

[0116] 图 26 是用于方便配置加工厂项目的示例性程序的流程图；

[0117] 图 27 是用于方便配置加工厂项目的示例性程序的流程图；

[0118] 图 28 是用于配置模块类对象的用户界面的示例性屏幕显示；

[0119] 图 29 是用于配置模块类对象的用户界面的另一示例性屏幕显示；

[0120] 图 30 是用于调试模块对象的用户界面的示例性屏幕显示，其中未隐藏与模块对象相关联的数据；和

[0121] 图 31 是图 30 的示例性屏幕显示，但是其中隐藏了与模对象相关联的数据。

具体实施方式

[0122] 现在参照图 1，加工厂 10 包括一个或多个经由例如以太网连接或总线 15 与大量工作站 14 耦接的过程控制器 12。控制器 12 还经由多组通信线路或总线 18 与加工厂 10 中的器件或设备耦接，在图 1 中，仅仅图解了与控制器 12a 连接的一组通信线路 18 与。控制器 12，可以仅使用 Fisher Rosemount Systems 公司销售的 DeltaV™ 控制器来实现，上述控制

器能够与控制元件,如在整个加工厂 10 中分布的现场设备中的现场设备和功能块进行通信,以便执行一个或多个过程控制程序 19,借此来实现期望的对加工厂 10 的控制。工作站 14(例如,可以是个人计算机)可由一个或多个配置工程师用来设计由控制器 12 所执行的过程控制程序 19,以及将由工作站 14 或其他计算机所执行的显示程序,并与控制器 12 通信,以便将这些过程控制程序 19 下载到所述控制器 12 中。此外,所述工作站 14 可以执行在加工厂 10 运行期间接收和显示与加工厂 10 或其元件相关的信息的显示程序。

[0123] 每个工作站 14 包括存储器 20,用于存储应用程序如配置设计应用程序、显示或查看应用程序,以及数据,如与配置加工厂 10 配置相关的配置数据。每个工作站 14 还包括处理器 21,用于执行这些应用程序以便使得配置工程师能够设计过程控制程序和其他程序,以及将这些过程控制程序下载至所述控制器 12 或其他计算机,或者在加工厂 10 运行期间向用户收集并显示信息。

[0124] 更进一步,每个控制器 12 包括存储控制和通信应用程序的存储器 22,和以任何已知的方式执行所述控制和通信应用程序的处理器 24。在一个实例中,每个控制器 12 存储和执行实现控制策略的控制器应用程序,该控制策略使用许多不同、独立执行的控制模块或块 19。每个控制模块 19 可以由通常称作功能块的块构成,其中每个功能块是整个控制程序的一部分或子程序,并与其他功能块(通过被称为链路的通信)协同操作,以便实现加工厂 10 内的过程控制环路。众所周知,功能块可以是面向对象的编程协议中的对象,它们一般执行诸如与变送器、传感器、或其他过程参数测量装置相关联的输入功能,诸如与执行 PID、模糊逻辑等控制的控制程序相关联的控制功能,或控制一些设备,诸如阀门的操作,以执行加工厂 10 内某些物理功能的输出功能。当然,还存在混合的和其他类型的复合功能模块,诸如,模型预测控制器(MFC)、优化器等。当所述现场总线协议和 DeltaV™ 系统协议使用面向对象的编程协议中设计和实现的控制模块和功能块时,能够通过使用任何期望的控制编程方案,例如包括时序功能块、阶梯逻辑(ladder logic)等予以设计,但是并不限于使用功能块或任何其他的特定编程技术来设计。

[0125] 所述工作站 14 可借助显示屏为用户提供控制器 12 内过程控制程序 19 的图形表示,所述显示屏示例了过程控制程序 19 内的控制元件,以及以一种配置这些控制元件的方式提供加工厂 10 的控制。在图 1 的系统中,配置数据库 25 与以太网总线 15 相连接以用来存储控制器 12 和工作站 14 所使用的配置数据,并通过收集和存储加工厂 10 中产生的数据用作数据历史器以便将来之用。

[0126] 在图 1 所示的加工厂 10 中,所述控制器 12a 通过总线 18 与三组相似配置的反应器(为工厂 1010 内的复制设备)相连接,所述反应器在此称为反应器_01、反应器_02 和反应器_03。反应器_01 包括:反应器容器或罐 100;三个输入阀门系统(设备实体)101、102 和 103,被连接以便用于控制分别向反应器容器 100 提供酸、碱和水的流体入口线;以及输出阀门系统 104,被连接以便控制流出所述反应器容器 100 的流体。传感器 105,可以是任何期望类型的传感器,如水位传感器(level sensor)、温度传感器、压力传感器等等,该传感器位于反应堆容器 100 中或其附近。为了讨论,这里假定所述检测器 105 为水位传感器。此外,共用集管阀系统(shared header valve system)110 被连接在每个反应器:反应器_01、反应器_02 和反应器_03 的水管管道上游处,以提供用于控制到每个反应器的水流的主控制。

[0127] 类似地,反应器_02 包括反应器容器 200,三个输入阀门系统 201、202 和 203,输出阀门系统 204 和水位传感器 205,而反应器_03 包括反应器容器 300、三个输入阀门系统 301、302 和 303、输出阀门系统 304 和水位传感器 305。在图 1 的实例中,所述反应器 : 反应器_01、反应器_02 和反应器_03 可使用提供酸的输入阀门系统 101、201 和 301、提供碱的输入阀门系统 102、202 和 302 以及输入阀门系统 103、203 和 303,并联合向反应器容器 100 提供水的共用水集管 110 来产生盐。所述输出阀门系统 104、204 和 304 可被操作用于将产物送出指向图 1 右边的流体管道,并将废物或其他不需要的物质排出指向图 1 底部的流体管道。

[0128] 所述控制器 12a 通过总线 18 与阀门系统 101—104、110、201—204 和 301—304 以及传感器 105、205 和 305 通信耦接,以控制这些元件的操作从而执行一个或多个关于反应器单元 : 反应器_01、反应器_02 和反应器_03 的操作。上述的操作,通常称作阶段 (phase),例如可以包括填充反应器容器 100、200、300,加热反应器容器 100、200、300 中的物质,倾倒反应器容器 100、200、300,清洗所述反应器容器 100、200、300 等。

[0129] 图 1 中所示的阀门、传感器及其他设备可以是任何期望种类或类型的设备,例如现场总线设备、标准 4 — 20ma 设备、HART 设备等,并可使用任何公知或期望的通信协议,诸如现场总线协议、HART 协议、4 — 20ma 模拟协议等等的协议与控制器 12 通信。更进一步,根据这里讨论的原理其他类型的设备可以连接到控制器 12,并受控制器 12 的控制。而且,在工厂 10 内可连接其他数量和类型的控制器来控制与加工厂 10 相关联的其他设备或区域,并且上述其他控制器的操作能够以任何期望的方式与图 1 中所示例的控制器 12a 的操作相协调。

[0130] 一般而言,图 1 的加工厂 10 可用于实现批处理,其中例如,一个工作站 14 或控制器 12a 执行批执行程序,该批执行程序为引导一个或多个反应器单元 (及其他设备) 操作的高级控制程序,以执行一系列不同的步骤 (通常称作阶段),这些步骤是用于生产产品,如特定类型的盐所需的。为了实现不同的阶段,所述批执行程序使用通常所称的制法 (recipe),该制法规定了将要执行的步骤、与所述步骤相关联的数量和时间以及所述步骤的顺序。用于一个制法的步骤包括,例如,使用适当的物质或配料填充反应器容器,混合反应器容器内的所述物质,将反应器容器内部的所述物质加热一段时间到某一温度,排空反应器容器并随后清洗所述反应器容器以准备下一批运行。每个步骤限定批运行的阶段,并且所述控制器 12a 内的批执行程序将为每个阶段执行不同的控制算法。当然,对于不同的制法,具体材料、材料量、加热温度、时间等也是不同的,因此,根据被制造或被生产的产品和所使用的制法,这些参数可能在批运行之间发生变化。本领域技术人员将会理解,尽管这里描述了用于图 1 中所示例的反应器中的批运行的控制程序和配置,但是控制程序可用于控制其他期望的设备来执行任何其他期望的批处理运行,或如果需要,用于执行连续的处理运行。

[0131] 同样可以理解,可以在相同或不同的时间,在图 1 的每个不同反应器单元上执行批处理的相同阶段或步骤。此外,由于图 1 的反应器单元通常包括相同数量和类型的设备,除必须修改这种类属阶段控制程序以控制与不同反应器单元相关联的不同硬件或设备以外,还可以使用用于特定阶段的相同类属阶段控制程序控制每个不同的反应器单元。例如,为了执行反应器_01 的填充阶段 (其中所述反应器单元被填充),填充控制程序将开启与输

入阀门系统 101、102 和 103 相关联的一个或多个阀门一定量的时间,例如直到水位计 105 检测到所述容器 100 被填满时为止。可是,程序可以仅仅通过改变与阀门系统 201、202 和 203 相关联但不与阀门系统 101、102 和 103 相关联的(多个)输入阀门的指示,以及通过改变水位计 205 而不是水位计 105 的水位计指示,从而用于实现反应器 _02 的填充阶段。

[0132] 图 2 更详细地示出了图 1 中的一个反应器,特别是反应器 _01。正如图 1 所类似地示例,图 2 的反应器 _01 包括:反应器罐 100;用于将酸、碱和水输入到罐 100 中的输入阀门系统 101、102、103 和 110;用于将材料从所述罐 100 中排除的输出阀门系统 104;以及水位传感器 105。如图 2 进一步所图解,每个输入阀门系统 101、102 和 110 使用相似的设备实体,这些实体被称之为累加器,且该累加器包括彼此平行设置的两个阀门以及在两个阀门下游设置的流量测定设备。在图 3 中详细图解的输入阀门系统 101 的累加器包括:称作粗阀门的开 / 闭类型的阀门 101a;和称作精确阀门的开 / 闭类型的阀门 101b;以及设置在阀门 101a 和 101b 下游的流量计 101c。累加器 101 具有一个或多个控制模块或与其相关联的程序,所述控制模块或程序用于使用流量计 101c 的测量值来控制酸的输入。第一个这种控制程序可通过使用粗阀门 101a 和精确阀门 101b 的累加器 101 来执行快速流量控制,而第二个这种控制程序可通过使用粗阀门 101a 和精确阀门 101b 的累加器 101 来执行精确的流量控制。

[0133] 将从图 2 中知道,所述碱输入阀门系统 102 包括具有粗阀门 102a、精确阀门 102b 的累加器和流量计 102c,以及共用进水阀门系统 110 包括粗阀门 110a、精确阀门 110b 和流量计 110c。尽管累加器用在相同单元即反应器 _01 单元的不同位置处,但每个累加器 101、102 和 110 具有相同类型的复制设备。同样,反应器 _02 和反应器 _03 还包括位于输入阀门系统 201、202、301 和 302 中的累加器。

[0134] 类似地,所述输出阀门系统 104 为包括三个阀门的另一部分复制设备。如图 4 中最佳示例的,输出阀门系统 104 包括:主输出阀门 104a,必须为将从罐 100 中排出的任何物质开启;产品阀门 104b,它必须与主输出阀门 104a 共同开启以便从罐 100 送出产品;以及排水阀 104c,该阀必须与主输出阀门 104a 共同开启以便将物质,诸如废弃物、清洗液等从罐 100 中排入到排水区域或保护系统中。当然,一个或多个控制程序与输出阀门系统 104 相关联以控制阀门 104a、104b 和 104c 的状态,从而关闭罐 100、为罐 100 排水或从罐 100 排空产品。

[0135] 过去,为了产生用于控制与图 1 的反应器:反应器 _01、反应器 _02 和反应器 _03 相关联的不同设备的控制程序,配置工程师可能首先已创建出多个模板控制模块,这些模块本质上属于一类并被存储在,例如一个工作站 14 的库中。为了创建模板控制模块,工程师将不同的控制功能块图形化连接在一起,为反应器上执行的不同组件或回路提供控制程序。一旦通常在阀门或控制回路的基础上创建了所述同类的模板控制模块,就能够复制这些模板控制模块并且能够手动地将模板控制模块的副本联结(bind)到工厂 10 内的特定设备上,如反应器:反应器 _01、反应器 _02 和反应器 _03 内的特定设备。一旦直接或使用别名联结了这些模板控制模块,如美国专利第 6,385,496 号中更具体描述的,下载所述控制模块的联结副本到一个或多个控制器 12 中,并用于在它们所联结的反应器上执行过程控制动作。但是,从模板控制模块创建的联结控制模块与创建所根据的模板模块没有任何引用或关系,并且当它们用于过程控制系统中时,这些被联结的控制模块实际上是独立的控

制模块或对象。

[0136] 而且,在这些系统中,必须在控制模块级执行所述配置,这意味着必须为加工厂内的设备或回路的每一不同部分创建单独的控制模块。在控制模块级上,一般存在许多不同类型的控制模块,必须为加工厂内的每个过程实体创建并且联结这些控制模块。因此,配置工程师花费大量的时间来简单地复制并且联结各个控制模块到工厂内的设备的各个部分。例如,配置工程师可能已不得不为工厂中的反应器单元创建和复制许多控制模块,然后将这些控制模块中的每一个与反应器单元中的特定设备相联结。当在工厂中复制所述反应器单元时,配置工程师的任务将变得更加单调乏味,这是因为配置工程师必须为每个复制设备的许多控制模块执行这种复制和联结过程,而这种过程既耗时又充满人为引入的差错。

[0137] 尽管,在过去,配置工程师能够开发出一种单元模块,但是这些单元模块仅仅是用于能够在单元上运行并且不包括与该单元或控制模式 (control scheme) 相关联的设备指示的阶段 (phase) 的容器,所述控制模式用于控制所述单元中的设备的基本操作。而且,尽管为用于控制不同设备的控制单元创建了这些模板,但是不存在可以被用于指示工厂内的更高层的复制实体的控制模块的组件 (package),诸如设备和单元实体。换句话说,为给加工厂 10 内的不同复制实体创建控制程序,配置工程师必须为在最低控制层的设备的每个复制部分复制控制模块,并为设备的具体或特定部分或加工厂 10 内的其他实体调整每个控制模块。在具有大量复制设备的大型工厂中,该任务既耗时又充满配置差错。而且,对于与复制设备相关联的控制模块的改变,必须针对不同设备的每一个不同的控制模块手动地进行,这也单调乏味、耗时以及充满由操作员引入的差错。同样,必须单独并且与控制模块分开地为操作员创建显示,类似于所述控制模块,必须为加工内的设备单独地创建、修改并连接所述显示。

[0138] 为了使过程配置的创建和改变更容易以及消耗较少的时间,存储在图 1 的一个工作站 14 中的配置应用程序 50 包括一组模块类对象 52,这些对象用于配置所述过程控制工厂 10。所述模块类对象在配置具有许多组复制设备的工厂时特别有用。一般而言,能够为在加工厂 10 中被复制或使用的每个不同类型的物理单元或设备,和在加工厂 10 中被复制和使用的每种类型的控制动作,以及为加工厂 10 中被复制和使用的每个不同类型的显示应用程序等创建不同的模块类对象 52。一旦被创建,所述模块类对象 52 就可以被用于配置对应于该模块类对象的加工厂 10 的部件。

[0139] 模块类对象 52,实质上为过程实体的同属版本 (generic version),并且没有与任意特定的过程实体相连接,它们可以具有低级别的对象或与其相关联的实例 53、54、55 和 56 (在此称作模块对象或模块块)。每个模块对象都根据模块类对象来创建,并继承与创建模块对象所根据的所述模块类对象的相同结构和属性。但是,每个模块对象与加工厂 10 内的特定实体相联系。因此,尽管可能存在不同的模块对象 53 或为工厂 10 中实际存在的该类型的每个不同的反应器单元创建所述模块对象 53,但是可以创建单独的模块类对象 52 来表示特定类型的反应器单元 (无论在工厂 10 中设置有多少反应器单元)。

[0140] 根据模块类对象所创建的模块对象与模块类对象相关联,并且模块类对象拥有该模块对象。结果,对模块类对象所作的改变能够被自动地反映在与该模块类对象相关联的每个模块对象中或被传播至所述每个模块对象。因此,当利用与不同过程实体相联系的每

个不同模块对象，已经根据特定的模块类对象创建了许多模块对象时，通过简单地改变模块类对象以及使所述改变自动地向下传播至相关联的模块对象，从而可以改变每个不同的模块对象。

[0141] 同理，一旦已经由所述模块类对象创建了模块对象，则使用块编辑方法可以将这些模块对象与加工厂 10 内的特定设备相联结。特别是，由于特定模块类对象的模块对象全部都与相同的模块类对象相耦接或被其拥有，所以，例如使用电子表格类型的应用程序来将模块对象一起进行配置，所述应用程序使得规定模块对象和工厂 10 中与那些模块对象相关联的特定设备之间的特定关系变得更容易，并且耗时更少。

[0142] 图 1 的模块类对象 52，可以是通常称作面向对象的编程环境或语言中的对象。因此，这些对象能够拥有或引用其他对象。一般而言，所述模块类对象 52 为高级对象，高级对象能够包括各个元件，诸如控制程序、设备或其他与过程实体相关联的元件的指示或定义，以及方式的定义或指示，在所述方式中这些各个元件之间相互作用，诸如其中物理元件相互连接的方法，或逻辑元件与所述物理元件联合操作的方法。换句话说，模块类对象可以是，例如面向对象的编程语言中的对象，它提供用于控制和查看加工厂 10 内设备的特定部件或设备组、控制元件、显示等的基础，并对于产生用于配置过程控制工厂 10 内不同复制设备的该元件的多个实例是有用的。

[0143] 基本上，每个模块类对象是一种配置容器，它包括全部的不同控制和 / 或显示应用程序或程序形式的过程实体的类属定义，所述应用程序或程序可适用于通过控制器 12 使用该实体来控制实体或由工作站 14 来使用该实体以执行关于该实体的显示动作。所述模块类对象可以表示任何属性的过程实体，如单元、设备的一个部件、控制实体、显示应用程序等。在配置加工厂 10 的过程中，所述模块类对象可被用于利用与不同的实际过程实体相关联或与其相连接的每个配置实例（由模块类对象创建的模块对象），为任意数量的不同过程实体创建过程实体的配置实例，所述不同的过程实体遵循模块类对象提供的定义。这些不同的模块对象除了别的以外还包括，与设置在加工厂 10 内的特定过程实体相连接的控制程序和 / 或显示程序，并利用能够被下载并在图 1 的控制器 12 内使用的这些控制程序，对过程实体执行实际的控制动作，以及利用能够被下载到工作站 14 的显示程序，以在加工厂运行过程中执行与实体相关的实际显示动作。

[0144] 不同类型的模块类对象可反映不同范围的过程实体，因此它们包含被配置用于对不同范围的过程实体操作或关于该过程实体的控制和 / 或显示程序。过程实体，诸如单元的范围越大，则一般与模块类对象相关联的控制和 / 或显示程序就越多，并且使用这些模块类对象配置工厂的部分就越容易。但是，与模块类对象相关联的过程实体的范围越大，所述过程包括该范围内的复制设备的可能性就越小，因此，模块类对象被大规模使用的可能性就越小。反之，与模块类对象相关联的过程实体的范围越小，所述模块就越有可能用于所述工厂的各个不同位置处，但是当在任何特定实例中使用模块类对象时，所执行的配置的量就越少。无论如何，当使用所述模块类对象，特别是大范围的模块类对象，如在单元级时，该模块类对象能够为位于高于控制模块层的抽象层上的不同复制设备执行配置，这使得利用复制单元及其他设备配置一个加工厂变得更容易且耗时更少。

[0145] 在一个实例中，当配置过程控制系统时，配置工程师可为加工厂内复制的不同元件，诸如图 1 中所示的不同反应器创建一个单独的模块类对象。之后，所述配置工程师可

为图 1 中的每个实际的反应器创建模块类对象的实例（模块对象）。每个这样创建的模块对象将包括控制器 12a 所使用的用于操作图 1 中一个反应器的控制程序，特别是所述模块对象与图 1 中一个反应器内的设备相连接或与其联结。然后将这些控制程序下载到控制器 12a，并在所述加工厂 10 的运行过程中使用上述控制程序。但是，一旦创建了模块对象，则每个模块对象仍与模块类对象相连接，并可由模块类对象控制而发生改变，以提供或拒绝对模块对象的访问等。同样，能够一起创建和定义来自相同模块类对象的模块对象，例如，使用电子制表程序将所述模块对象与工厂内的设备相连接。

[0146] 尽管存在许多不同可能类型的模块类对象，并且能够在加工厂内创建或使用所述模块类对象以在加工厂内执行配置动作，但是这里讨论的作为实例的四种具体类型包括单元模块类对象、设备模块类对象、控制模块类对象和显示模块类对象。一般而言，为不同的控制范围设计每个不同类型的模块类对象或它们旨在用于加工厂 10 内的不同用途。单元模块类对象旨在用于表示（和配置）加工厂内宽泛范围设备的控制动作。特别地，单元模块类对象旨在模型化或用于配置内部相关联的一组设备（一般为复制设备）诸如，例如，图 1 的反应器，其中设备具有以某种公知的方式彼此协同工作的各个元件。

[0147] 设备模块类对象旨在用于表示（和配置）加工厂内较宽范围的物理设备的控制动作。与设备模块类对象相关联的设备通常是一个或多个物理实体，诸如阀门、流量计等，所述物理实体构成了一个单元的子系统，设备模块类对象可能包括一个或多个命令或算法，所述算法可以是将在设备部件上执行的命令驱动算法（CDA）、状态驱动算法（SDA）、顺序流程图（SFC）算法、功能方框图（FED）算法、阶段算法等。因此，设备模块类对象旨在用于配置单元内的许多低层元件或实体的控制，以便提供关于如在单元内部使用的设备的基本功能集。正如所知，通常当低层元件必须通过多个步骤被协调来完成一种功能时，使用命令驱动算法（命令驱动控制逻辑）。例如，可能需要将阀门开启特定的时间段，然后将它关闭，同时将另一阀门开启然后关闭该阀门。图 3 的累加器 101 使用这种类型的命令驱动算法来基于流量计的读数，首先启动然后操作粗阀门和精确阀门，以便通过累加器提供所需的总流量。状态驱动算法（状态驱动控制逻辑）可规定不同的低层元件的状态，所述元件能以单独的步骤予以操作。上述的状态驱动算法用于图 4 的输出阀门系统 104 中，在该系统中基于所述输出阀门系统 104 的状态分别（只在单独的步骤中）控制不同阀门的状态以关闭罐 100，排空所述罐 100 或从罐 100 传送产品。

[0148] 控制模块类对象旨在用于表示（和配置）加工厂内的各个控制元件或控制模块。控制模块类对象提供或规定将在工厂实体诸如阀门、仪表等，设备的部件或甚至在一个单元上执行的特定类型的控制。一般而言，控制模块类对象提供特定类型的控制程序，诸如一组通信互连功能块，所述功能块定义了在控制器中将执行的某个控制模块，并有利的用于执行加工厂内的复制控制动作。一般地，控制模块类对象可提供类属的控制策略来操作单个设备或一组相关的设备。

[0149] 显示模块类对象旨在用于表示（和配置）在加工厂 10 运行过程中由用户，诸如控制操作员查看的显示动作。因此，显示模块类对象可规定所需用于产生图 1 的操作员工工作站 14 内某种类型的显示的程序，以及所需在一个或多个工作站 14（以及加工厂 10 中任何其他的设备）中运行的程序，以使能显示从而在工厂 10 运行过程中从工厂获得适当的信息。显示类模块的类型包括，例如，警报显示、配置查看显示、操作查看显示、诊断显示等。当

然,显示模块类对象可提供表示加工厂内任何所需物理元件或实体的期望范围的显示或与其相连接的显示。例如,显示模块类对象可显示关于全部区域、一个单元、一个设备部件、一个控制元件或加工厂 10 内这些元件的任何组合的信息。

[0150] 参照图 5,分层体系结构图图解了在图 1 的配置应用程序 50 中使用的不同类型的模块类对象之间的互联,以及模块类对象和从这些模块类对象开发的模块对象之间的相互关系。从图 5 的图表顶部开始,通过模块类型将模块类对象分成单元模块类类型 400、设备模块类类型 402、控制模块类类型 404 和显示模块类类型 406 的其中之一。当然,也可以利用这里示例的、仅作为示意性模块类类型的四种类型,提供或使用其他类型的模块类对象。各个模块类对象(其可以是例如面向对象的编程语言中的高层对象,并为了说明起见在图 5 中以双轮廓线表示)落入每个不同类型的模块类 400、402、404 和 406 中。特别地,对于加工厂 10 内的不同单元或不同类型单元,可能有大量的不同的单元模块类对象。例如,反应器单元类模块对象 410 可以表示特定类型或配置的加工厂 10 内的反应器。同样,包装单元模块类对象 412 可表示加工厂 10 内的包装单元的特定类型或配置,以及干燥单元类模块对象 414 可表示加工厂 10 内干燥单元的特定类型或配置。当然,可能有多于一个表示彼此物理构造不同的反应器的反应器单元模块类对象。进一步,没有尝试列出工厂内所有类型的单元,这些单元能够由单元模块类对象来表示或模型化,本领域普通技术人员将会意识到在不同类型的工厂中存在许多不同类型的单元,所述单元可由元件模块类对象来表示或模型化。

[0151] 类似地,有许多不同的设备模块类对象被用于表示、模型化和配置加工厂 10 内不同类型的设备。图 5 中图解的实例包括累加器设备模块类对象 416 和输出阀门设备模块类对象 418,每个模块类对象与加工厂 10 内不同类型的设备(并优选为复制设备)相关联。以同样的方式,图 5 中示例了许多不同类型的控制模块类对象,如开 / 关阀门控制模块类对象 422、水位传感器控制模块类对象 424 和流量计控制模块类对象 426。此外,图 5 中示例了显示模块类对象,如告警显示模块类对象 432、查看显示模块类对象 434 和诊断显示模块类对象 436。当然,根据在此描述的原理,在加工厂 10 的配置应用程序 50 中也可以创建和以任何其他期望的单元、设备、控制和显示模块类对象。

[0152] 每个模块类对象也可以具有与其相关联或由其所拥有的子对象(sub-object)。这些子对象可以是在它们中的模块类对象,或如图 5 中所示,可以被创建作为它们所属的模块类对象的实例的模块对象。图 5 示出了反应器单元模块类对象 410 具有与其相关联的称作反应器_01、反应器_02 和反应器_03 的三个反应器模块对象,其中这些反应器模块对象对应于(被联结到)图 1 中的相应反应器。图 5 还示出了累加器设备模块类对象 416,该对象具有或拥有称作水 1、酸 1、酸 2、碱 1 和碱 2 的五个不同模块对象。同样地,开 / 关阀门控制模块类对象 422 被图解为包括称作粗_阀门 1、粗_阀门 2、粗_阀门 3、精确_阀门 1、精确_阀门 2 和精确_阀门 3。以类似的方式,图 5 中的每个其他单元、设备、控制和显示模块类对象具有一个或多个与其相关联的模块对象。但是,为了简单起见,这些模块对象在图 5 中未示出。

[0153] 在图 5 的图表中,每个反应器_01、反应器_02 和反应器_03 单元模块对象,酸 1、酸 2、碱 1、碱 2 和水 1 累加器(设备)模块对象,粗_阀门 1、粗_阀门 2、粗_阀门 3、精确_阀门 1、精确_阀门 2 和精确_阀门 3 控制模块对象以及其他单元、设备、控制和显示模

块对象为与所述加工厂 10 内的实际单元、设备、控制模块或显示应用程序相联系的各个对象。例如,由于在工厂 10 中使用多个物理的酸累加器,所以将在配置程序中创建多个酸累加器模块对象,其中对于工厂 10 内的每个单独酸累加器存在单独的酸累加器模块对象。但是,每个单独的累加器模块对象与相同的累加器模块类对象 416 相联系或由其所拥有。当然,图 5 的图表仅示例了有限数量的模块类对象和与其相关联的模块对象,应该理解的是,可以提供其他类型的模块类对象,和由每个不同模块类对象,可以创建的任意期望数量的模块对象。

[0154] 图 5 的每个模块类对象(并且因而图 5 的每个模块对象)可包括定义或构成模块的物理或逻辑过程部件的定义或指示,以及如果需要,包括这些过程部件彼此物理或逻辑相互作用以执行加工厂 10 内的某种动作的方式来作为对象的一部分。例如,单元模块类对象会一般包括位于被定义为单元的过程实体内或过程这些过程实体的所有物理和控制部件的指示。所述单元模块类对象也可定义各个部件的特定构造以及这些部件是如何物理连接在一起以便作为一个单元而操作。同样,设备模块类对象将一般包括控制程序或控制模块以及命令,其中所述控制程序或控制模块用于控制被定义为设备的部件的实体,所述命令使用该控制程序或控制模块以定义其中当部件被布置在工厂 10 内时,所述部件之间物理或逻辑地相互作用以作为设备的一个部件而运行的方式。类似地,每个控制模块类对象一般以某种顺序的控制算法的形式来定义将在所述工厂内执行的控制动作。而且,除此之外,每个显示模块类对象,如果有的话,可以为特定类型的单元、设备、工厂区域、或工厂 10 内的任何其他物理或逻辑实体,定义显示屏幕配置和将要显示的信息以及将收集的数据和将要对收集数据执行的数据操作。

[0155] 作为模块类定义的一部分,模块类对象可以指示或定义其他在此包含或使用的模块类对象。在这种情况下,从该模块类对象创建的模块对象将根据在模块类级(module class level)定义的关系,包含、引用或包括从其他模块类对象所创建的其他模块对象。虽然不是绝对必要,但是单元模块类对象可包含其他的单元模块类对象、设备模块类对象、控制模块类对象和显示模块类对象,而设备模块类对象可以包含其他的设备模块类对象、控制模块类对象和显示模块类对象。控制模块类对象可以包含或引用其他的控制模块类对象和显示模块类对象。但是,如果需要,也可以使用其他模块类对象的相互关系。由在图 5 的图表的底部的大箭头图解这些包含关系,所述箭头指示,任意的显示模块类对象可包括在任意的控制、设备和单元模块类对象中或被其引用(refer to),任意的控制模块类对象可包括在任意的设备和单元模块类对象中或被其引用,以及任意的设备模块类对象可包括在任意的单元模块类对象中或被其引用。应当理解,模块类对象可以包含其他相同类型的模块类对象。例如,单元模块类对象可以包含如作为其定义一部分的另一单元模块类对象。以类似的方式,设备模块类对象可以包括另一设备模块类对象,控制模块类对象可以包括另一控制模块类对象以及显示模块类对象可以包括另一显示模块类对象。当然,如果需要,模块类对象可以多次使用或包含另一模块类对象。例如,由于被反应器单元类对象模型化的反应器包括多个累加器的实例,所以反应器单元模块类对象可以多次包含或使用所述累加器设备模块类对象。

[0156] 也应当理解,当第一模块类对象包含或使用第二模块类对象时,由第一模块类对象实例或作为该对象实例创建的任意模块对象将包含或使用由第二模块类对象实例或作

为该对象实例创建的模块对象。因此,当反应器单元模块类对象 410 使用作为其部件或部件的一部分的累加器模块类对象 416 时,所述反应器 _01 模块对象将使用或包括一个作为部件或部件的一部分的累加器模块对象,诸如酸 1 模块对象。类似地,如果累加器设备模块类对象包含或包括输出阀门设备模块类对象,由所述累加器设备模块类对象创建的模块对象,将被唯一地命名为如累加器 _1,以及所述模块对象包括从输出阀门设备模块类对象创建的模块对象,并被唯一地命名为输出 _ 阀门 _2。通过这种方式,如在模块类对象级定义的模块类对象(以及因此模块对象)之间的这种互连或引用能够在配置动作期间实现对象的高度可变性 (variability) 和高转移性 (transferability),以便在创建一组原始模块类对象,诸如控制和设备模块类对象之后,可通过引用该原始模块类对象来很容易地创建更复杂的模块类对象,如单元模块类对象。当然,当模块类对象可以引用或使用其他模块类对象时,它们也可以或替代地定义或使用简单对象或过程实体,如阀门、传感器等,这些实体不具有相联的模块类对象。因此,根据所使用的控制程序,在模块类对象本身内来完全地定义这些简单对象。

[0157] 在图 6 图解反应器单元模块类对象 410 的一个实例,以便示出描述或定义与单元模块类对象相关联或在其内部的实体的一种方式。如图 6 所示,所述反应器单元模块类对象 410 包括罐 500 的指示,这是加工厂 10 内的一个简单对象或元件,对于该对象,不存在模块类对象。由于不存在被控制或需要控制的低层动作,或执行关于所述罐的输入 / 输出动作,故以虚线示出所述罐 500。因此,包含罐 500 仅仅是为了示例与反应器单元模块类对象 410 相关联的其他对象之间的相互连接关系。所述反应器单元模块类对象 410 也包括三个分别称作酸、碱和水的累加器 501、502 和 510,它们为图 5 的累加器设备模块类对象 416 的三个不同的引用。以单元模块类对象 410 的剖面的方式图解了所述水累加器模块类对象 510,所述剖面由虚线分开以便指示该模块类对象是共享模块类对象,因此指出所述单元模块类对象 410 对具有与其他单元模块类对象的共享的对该对象的控制。图 6 的输出对象 504 是对图 5 的输出阀门设备模块类对象 418 的引用,所述水位传感器 505 是对图 5 的水位传感器控制模块类对象 424 的引用,以及进水阀门 503 是对阀门对象的引用,所述阀门对象可以是简单的阀门元件(以及在单元模块类对象 410 中被完全地定义),或可以是对配置策略 (configuration strategy) 中其他地方定义的阀门控制模块类对象的引用。为了定义不同元件之间的相互连接关系,还图解了反应器单元模块类对象 410 的不同实体或部件之间的物理互连。如上所述,所述单元模块类对象 410 或其他任意类型的模块类对象可以包括简单元件,该元件在所述模块类对象(包括与此相关联的任意类属控制程序)内被完全地定义,和 / 或可以包括对在模块类对象之外所定义的模块类对象的引用。

[0158] 所述单元模块类对象 410 也包括两个称作反应器查看显示 520 和反应器警报显示 522 的示例的显示模块类对象,它们是对图 5 的查看显示模块类对象 434 和警报显示模块类对象 432 的引用。这些对象定义了用于显示状态(如,所述罐的填充水位)的类属显示动作,以及与任意设备或反应器单元模块类对象 410 中所定义的反应器单元的部件相关联的警报。类似地,所述单元模块类对象 410 可以包括其他的元件,诸如在框 524 中所图解的阶段类对象,如剂量、混合、排水和冲洗阶段类对象,每个阶段对象定义了在单元模块类对象 410 所定义的单元上运行的类属控制程序。所述单元模块类对象可以具有零个或多个阶段

类对象的关联。可在别处定义所述阶段类对象 524 并以任何期望方式将其输入到单元模块类对象 410 中。在某种意义上,所述阶段类 524 是命令或程序,所述命令或程序可以在单元模块类对象 410 所定义的单元上运行以便执行不同的功能,诸如填充、加热、排空、和清洗所述元件等。

[0159] 此外,所述单元模块类对象 410 可以包括存储对由配置应用程序 50(图 1)从该单元模块类对象所创建的模块类对象的引用的存储器或区域 526。所述区域 526 实质上是模块对象的列表,所述模块对象可由元件模块类对象 410 创建并被其所拥有。(当然所拥有的模块对象的列表或其他指示可被存储在工作站中或以任何期望方式由配置应用程序 50 将其存入工作站中,并且不必物理地包含在所述单元模块类对象 410 中)。无论如何,在图 6 的实例中,所述单元模块类对象 410 拥有单元模块对象反应器_01、反应器_1、反应器_02 等,已经从反应器单元模块类对象 410 创建每一反应器。

[0160] 所述单元模块类对象 410 也包括一组方法 530,其中在配置动作期间或之后由单元模块类对象 410 执行这些方法。所述方法 530 可以包括改变管理方法或应用程序,其自动地传播对单元模块类对象 410 的改变到单元模块类对象 410 所拥有的每个模块对象 526。其他方法可以包括安全控制方法,所述方法执行关于单元模块类对象 410 和 / 或关于拥有的任意单元模块对象 526 的安全或访问控制,或者使得用户或配置工程师能够为模块类对象或任意由此创建的模块对象指定改变参数和 / 或安全参数的方法。当然,不同的方法 530 可以对单元模块类对象 410 执行其他过程或执行与其相关的其他过程。

[0161] 如果需要,所述单元模块类对象 410 可以控制将模块类对象 410 所作的改变传播到单元模块对象 526 中的方式,以及在单元模块对象 526 中建立安全访问的方式。提供这种功能的一种方式为在单元模块类对象 410 内设置一个或多个标记或参数以便指定,将改变传播到单元模块对象 526 以及在单元模块对象 526 中处理安全性的方式。特别地,可以设置一个或多个改变传播参数来指定对单元模块类对象 410 作出的改变是否自动传播至一个或多个单元模块对象 526。这些改变传播参数可保存在单元模块对象 526 中,或在基于逐个子元件,为整个单元模块对象指定是否对单元模块类对象作出的改变将反映在单元模块对象中。例如,所述单元模块类对象 410 可以包括全局改变参数 534(标记为“C”),其可被设置在从单元模块类对象 410 所创建的每个单元模块对象中,以便使能或禁止对单元模块类对象 410 作出的改变自动地反映在所述单元模块对象中。同样,每个子元件或块,诸如块 501-505、510、520、和 522 可以包括改变参数 536,该参数仅为该块指定对在单元模块类对象 410 中的该块所作出的改变是否将反映在该单元模块对象中。当然,可以不同地设置单元模块对象的不同块,以便例如对单元模块类对象 410 的酸块 501 作出的改变将传播至特定的一个模块对象 526 的相应酸块,但是因此对单元模块类对象 410 的碱块 502 所作出的改变将不会传播至该特定的一个模块对象的碱块。此外,从单元模块类对象所创建的不同单元模块对象可以具有彼此不同的改变参数集合,以使对单元模块类对象 410 内的碱块 502 作出的改变传播至第一单元模块对象 526 的相应碱块,而不传播到第二元件模块对象 526 的相应碱块。当然,单元模块类对象 410 的改变管理方法可以访问和使用单元模块对象 526 的改变参数,以当在单元模块类对象 410 中进行所述改变时,在这些对象内部产生或不产生改变。

[0162] 以类似的方式,单元模块类对象 410 可以包括一个或多个安全参数,这些参数规

定了在每个单元模块对象 526 中控制安全性或访问的方式。所述单元模块类对象 410 可以包括全局安全参数 538 (标记为“S”), 其可以为从反应器单元模块类对象 410 所创建的整个反应器单元模块对象提供任何所期望级别的安全性, 和 / 或可以包括用于单元模块类对象 410 的每个子元件, 诸如每个块 501 – 505、510、520、522 等的不同安全参数 540, 该参数逐块为那些程序块中的每个块规定了安全级别。所述全局安全参数 538 可以是锁定参数 (locking parameter), 所述锁定参数对除了具有预授予的安全访问级别的用户外的所有用户锁定单元模块类对象。当然, 所述安全参数 538 和 540 可以规定多个不同级别的安全中的任意一个, 诸如禁止访问、限制访问、访问特定类型用户或标识符的用户等, 并且所述安全级别可以在不同的块中, 以及由相同单元模块类对象所创建的不同单元模块对象中而设置不同。如果需要, 安全措施的部分可以包括提供对与单元模块类对象相关联的一个或多个方法或算法的加密。

[0163] 应当理解, 所述单元模块类对象 410 的改变和安全参数可被设置为缺省值, 以及从单元模块类对象 410 所创建的每个单元模块对象 526 的相应改变和安全参数, 在其被创建时可以具有该缺省值。但是, 该缺省改变和安全参数在该单元模块对象被创建之后, 也可在所述单元模块对象 526 中 (由用户利用适当的安全访问) 来分别改变。尽管在此讨论与反应器单元模块类对象相关的改变和安全参数, 但是可以以其他类型的单元模块类对象以及任何期望类型的设备模块类对象、控制模块类对象、显示模块类对象等等来提供类似的改变和安全参数。

[0164] 如果需要, 所述单元模块类对象 410 可以包括对为单元类模块对象存储的文件或与单元模块对象相关联的文件的引用 (references), 如 URL 或其他引用, 所述文件包括与所述单元或与单元模块类对象 410 相关联的单元的任何子元件相关的文件。这些引用在图 6 中被图解为附图标记 549。

[0165] 为了模块类对象的另一实例, 在图 7 中图解了累加器设备模块类对象 416。所述累加器设备模块类对象 416 包括称作粗阀门 550 和精确阀门 552 (二者都是开 / 关型控制模块类对象) 的控制模块类对象, 和称作流量计 (其为流量计控制模块类对象) 554 的控制模块类对象, 以及这些元件之间互连的指示。更进一步, 所述累加器设备模块类对象 416 包括显示模块类对象的引用, 包括对累加器警报显示模块类对象 560 的引用和对可在设备模块类对象 416 上执行的一个或多个算法 564 的引用。尽管所列的算法 564 包括快速累加 (Totalize_Fast) 命令和精确累加 (Totalize_Accurate) 命令, 但是也可以包括或使用其他命令或算法。更进一步, 与设备模块类对象相关联或包含在其中的命令算法 564 可以采用任何期望的形式, 并可以例如为, 命令驱动算法 (CDAs)、状态驱动算法 (SDAs)、顺序流程图 (SFC) 算法、功能框图 (FED) 算法、阶段算法等。可是, 一般而言, 所有算法 564 将是特定的类型, 如 CDA 或 SDA。当然, 所述算法 564 可以以任何期望的语言或编程环境, 诸如 C、C++ 编程环境、任何顺序功能图表编程环境、功能块编程环境等来编写。

[0166] 所述累加器设备模块类对象 416 也包括列表或存储器, 其中存储从设备模块类对象 416 所创建的一组称作酸 1、酸 2、碱 1、碱 2、水_Hdr1 (Water_Hdr1) 的指示 (并且如果需要, 为通信路径)。所述累加器设备模块类对象 416 还包括一组方法 570, 所述方法包括与全局改变参数 572 和 / 或基于对象的改变参数 574 结合使用以便控制改变到设备模块对象 566 的传播的改变管理方法。所述累加器设备模块类对象 416 还包括全局安全参数 580 以

及基于对象的安全参数 582。所述改变和安全参数 572、574、580 和 582 通常如对关于图 6 的单元模块类对象 410 的改变和安全参数的描述一样操作，并可应用于包括命令 564 的控制模块类对象 416 的任意元件。此外，正如此处特别描述的，所述改变和安全参数中的任意一个或两者都可设置在模块类对象中（以及从其创建的模块对象内）的任意期望级别，而不是全局地或在所述对象或元件级上。如果需要，模块类对象或从其所创建的任何模块对象的改变和安全参数，都可使用批处理配置应用程序 (bulk configuration application) 或程序，诸如表格程序来设置或配置，以便使得配置工程师能够为整个模块类对象或模块对象以简单直接的方式来设定或规定不同的改变和安全参数。

[0167] 如果需要，所述设备模块模块类对象 416 可以包括为设备模块类模块对象所存储或与其相关联的文件的引用，如 URL 或其他引用，所述文件包括与所述设备或与设备模块模块类对象 416 相关联的任意子设备相关联的文件。上述的引用在图 7 中被图解为附图标记 599。同样地，任意设备模块类对象 416 的算法，如任意的算法 564，可被加密或具有与其相关的安全参数，并利用这些参数来加密或解密这些算法 564。如果需要，可使用方法 570 执行这样的加密或解密。

[0168] 为了配置所述加工厂 10，配置工程师根据与配置应用程序 50（图 1）相关联的库内部的需要，创建单元、设备、控制和显示模块类对象。如果需要，所述配置工程师可以从较低的范围实体开始，诸如控制和显示模块类，然后为更高的范围实体，诸如可以使用或引用所述较低范围实体的设备和单元模块类对象开发模块类对象。此后，如果有必要则所述配置工程师可以选择或规定所述模块类对象，以创建对应于工厂内的每个过程实体的所选择模块类对象的实际模块对象。当配置复制设备时，配置工程师将由相同模块类对象为复制设备的每个实例创建一个模块对象。因此，配置工程师可以为图 1 的反应器创建一个反应器单元模块类对象（其包括如果这样的其他模块类对象在所述库中不存在，则创建由反应器单元模块类对象引用的其他模块类对象）。此后，配置工程师可以通过由反应器单元模块类对象为图 1 的每个反应器：反应器_01、反应器_02 和反应器_03 创建一个反应器单元模块类对象来配置所述过程。

[0169] 在由模块类对象创建一个或多个模块对象之后，配置工程师可以将模块对象（包括子对象或被引用的对象）与工厂内的特定设备相联结。因为该单元模块对象与单个单元模块类对象相关联，所以不同单元模块对象的别名、参数及其他变量可以使用例如，批处理应用程序如电子表格应用程序来一起指定。当然，通过将所述模块对象与特定设备相联结，配置工程师实际上正在指定由控制器 12 内的控制程序或控制模块使用的控制变量和通信路径名称，以便在加工厂 10 的运行期间执行管理动作或在加工厂 10 的运行期间，在如工作站 14 中显示由显示程序所使用的变量。在所述联结动作完成之后，配置工程师然后将该联结的控制程序下载至控制器 12 中以及将联结的显示程序下载到工作站 14 中。

[0170] 图 8-16 描述在配置工程师创建和使用模块类对象来配置加工厂 10 期间可以由图 1 的配置应用程序 50 所创建的屏幕显示。正如所理解的，图 8-16 中的大部分的屏幕显示包括在屏幕左侧的资源管理器视图，它提供了描述加工厂 10 的配置的组织树状结构。同样地，图 8-16 包括在其右侧的一个或多个信息视图。这些信息视图提供关于在资源管理器视图中所选定的一些元件的进一步信息。能够显示给用户或由用户在信息视图中改变的信息，通过图 6 和 7 的为每个不同模块类对象或其子元件所设置的控制和安全参数 534、

536、538、540、572、574、580 和 582 来控制和确定。因此，资源管理器视图内的特定元件是可以显示或陈列给用户，以便基于模块类对象中所设置并传播到资源管理器视图中所描述的模块对象的安全和控制参数来进行查看和 / 或改变。当然，如前所述，信息可以被一直隐藏，仅当用户输入密码或其他安全码时才可以显示或可改变，也可以一直是可显示且不可改变，以及始终是可显示和可改变的或这些或其他安全和改变参数的组合。更进一步，如果需要，可以使用高亮、变灰、颜色或其他技术在资源管理器视图中指示元件的可显示性或可改变性，以告知用户这些元件能够被更详细地显示或改变。

[0171] 在图 8 中，屏幕显示 600 包括在显示的左侧上示出的资源管理器结构视图 (explorer configuration view) 602 的部分。资源管理器视图部分 602 图解了存储大量模块类对象的库，这些模块类对象包括单元模块类对象 (Unit Classes) 604、设备模块类对象 (Equipment Module Classes) 606 和控制模块类对象 (Control Module Classes) 608。反应器单元模块类对象 (REACTOR) 610 (可对应于图 6 的反应器单元模块类对象 410) 被存储在单元模块类库 604 中，并包括大量子元件的指示，所述子元件包括剂量 (Dose)、配比 (Mix)、排水 (Drain) 和冲洗 (Flush) 阶段类对象，以及酸 (Acid)、碱 (Alkali)、水 (Water) 和输出设备 (Outlet) 模块类对象，进水 (Water_In) 和水位计量表 (Level_Meter) 控制模块类对象及其他所需的对象。因此，如在单元模块类库 604 中所定义的，反应器单元模块类对象 610 包括阶段类的指示以及设备模块类对象和控制模块类对象的指示。由于在屏幕 600 中选择了反应器单元模块类对象 610，所以在屏幕 600 的右侧 612 更详细地描述了所述对象的元件。

[0172] 更进一步，所述设备模块类库 606 包括累加器设备模块类对象 (TOTALIZER) 614 (其可以对应图 7 的累加器设备模块类对象 416) 和反应器输出设备模块类对象 (REACTOR_OUTLET) 616。所述累加器设备模块类对象 614 包括三个称作命令 _00001 (Command_00001)、命令 _00002 (Command_00002) 和命令 _00003 (Command_00003) 算法的不同部分 (如图 7 的算法 564)。所述模块类对象 614 也包括控制模块对象的引用，所述对象称作粗 _ 阀门 (COARSE_VALVE) 和精确 _ 阀门 (FINE_VALVE，为开 / 关型控制模块类对象) 和流量计 (Flow_Meter，其是流量计型控制模块类对象)。更进一步，所述反应器输出设备模块类对象 616 包括状态驱动装置控制算法，该算法具有称作状态 _00001 (State_00001)、状态 _00002 (State_00002) 和状态 _00003 (State_00003) 的不同状态，目标 (TARGET)、驱动 (DRIVE)、监视 (MONITOR) 和读回模块 (READBACK)，和输出 (OUTLET)、排水 (DRAIN) 和产品阀门 (PRODUCT) 控制模块对象 (其可以是对开 / 关控制模块类对象型的模块方块的指示或引用，并且其可称为输出、排水和产品或是可以是简单对象)。与所述累加器和反应器输出模块类对象 614 和 616 相关联的、命令和状态驱动算法可以是任何期望程序，并可引用通过利用这些命令而被使用的设备模块类对象内的控制模块对象。特别地，通过包含对应那些模块的模块块名以指示在执行所述算法时将被操作的设备的部件，该设备模块类对象的 CDA 或 SDA 命令算法可以包括引用控制模块 (或其他设备模块) 的表达式或动作。使用对应这些算法内的控制模块 (或另一设备模块) 的模块块名，规定由设备模块对象引用或与其相关联的控制模块对象，在所述设备模块对象中，设有算法并且在由设备模块类对象创建设备模块对象时，联结或示例化 (instantiate) 所指定的名称。

[0173] 图 9 图解了配置屏幕 620，其中所述累加器设备模块类对象 (TOTALIZER) 614 被放大以便图解与此相关联的运行逻辑，其包括具有命令 _00002 (Command_00002)、命令 _00003 (Command_00003) 和命令 _00099 (Command_00099) 的命令驱动算法。如在屏幕 620 的右侧 622 所图解的，更详细地提供了有关运行逻辑控制程序的信息。特别地，所述 Command_00002 是一种用于通过累加器控制准确的或精确的流量的总计精确程序。所述 Command_00003 是一种用于通过所述累加器被控制或提供快速流量的总计快速程序，而所述 Command_00099 是复位累加器的复位程序。屏幕 620 中的监视框是与提供监视动作的运行逻辑相关联的功能块。如根据图 8 和 9 所了解的，库 604、606 和 608 内的模块类对象中的每个元件具有子对象和子元件，并且在部分或所有的情况下，这些子元件涉及或包括将由计算机，诸如控制器或工作站在加工厂 10 运作期间执行的控制程序或显示程序。但是，所述控制和显示程序以及模块类对象内的其它设备引用本质上是同属的，以便它们不与工厂 10 内的实际的单个设备相连接或结合。

[0174] 图 10 图解了另一屏幕显示 630，该显示示出了所述累加器设备模块类对象 (TOTALIZER) 614 的运行逻辑中的命令 _00002 (Command_00002) 程序的细节。如由顺序功能图表程序编辑器 634 开发的弹出窗口 632 中所图解的，所述 Command_00002 程序包括一系列步骤，并且每个步骤包括一系列动作。在窗口 632 中仅图解了 Command_00002 程序的第一步骤 (S1) 的第一动作 (A1)。正如所标注的，Command_00002 程序的第一步骤的第一个动作将 Flow_Meter 控制模块对象参数 Reset 设置为“True”，并随后根据变量 SP 的情况，将粗 _ 阀门 (COARSE_VALVE) 控制模块对象（或模块方块）的目标 _ 状态 (TARGET_STATE) 参数设置为“阀门：开启 (Valve:Open)”，并将精确 _ 阀门 (FINE_VALVE) 控制模块对象（或模块方块）的 Target_State 参数设置为“Valve:Open”。应当理解，该段 Command_00002 程序引用与累加器设备模块类对象在该点类属地相关联的 Flow_Meter、Coarse_Valve 和 Fine_Valve 控制模块对象（或模块），但从累加器设备模块类对象创建的任意实际的累加器设备模块对象的 Command_00002，将引用从累加器设备模块创建的或作为其一部分的控制模块对象的实际名称。

[0175] 图 11 图解了屏幕视图 640，该视图示出与剂量阶段类 (DOSE) 642 相关联的控制算法的细节，所述剂量阶段类可用在图 8 的反应器单元模块类对象 610 中使用或被其引用。特别地，所述剂量阶段类 642 包括一组不同的通用控制动作，这些动作包括中断 (ABORT)、失败 (FAIL)、保持 (HOLD)、重启 (RESTART) 和运行 (RUN)。与运行动作相关联的逻辑（称为运行逻辑 (Run_logic)）包括屏幕区 644 中所图解的一系列步骤，如步骤 1 (S1)、步骤 2 (S2) 和步骤 3 (S3)，两个转换 T1 和 T2 以及结束 (TERMINATION) 步骤。顺序功能图表编辑器 646 以方块图的形式图形图解上述步骤和转换。弹出屏幕 648 图解了与步骤 1 (S1) 的动作 1 (A1) 相关联的实际控制逻辑。特别地，在作为逻辑编辑器的屏幕 648 中的运算逻辑，包括获得用于反应器的公用汇水管 (water header) 的动作，打开进水阀门 (water_in valve) 并随后复位与所述反应器相关联的所有累加器。

[0176] 因为所述剂量阶段类独立于所述反应器编写，所以它使用别名以便引用在运行时间或其之前所指定的设备，当然，不能在反应器单元模块类对象中指定所述设备。但是，在反应器单元模块类对象的情况下，对控制算法 648 内使用的实际设备或其他过程元件的引用，是对处于相同单元模块类对象中的那些设备的引用。在控制程序 648 中，借助“# 符号”

描述别名,使得“#_WATE_IN#”为相同模块,也就是单元模块类对象 610 中的进水控制模块的别名。同样地,所述“#WATER#”、“#ACID#”和“#ALKALI#”分配指反应器单元模块类对象 610 内的水、酸和碱累加器模块。当然,应当理解,所述累加器和反应器输出设备模块类对象以及开 / 关阀门控制模块类对象,通过相应的累加器和反应器设备模块对象(例如称作碱 1、酸 1 等)和开 / 关控制模块对象(例如称作粗 _ 阀门 (Coarse_Valve1)、精确 _ 阀门 (Fine_Valve)1、输出 (Outlet)1 等)与反应器单元模块对象相关联,从而变为单元类对象的逻辑部分。

[0177] 图 12 图解了屏幕显示 650,其中更详细地示例了所述反应器输出 (REACTOR_OUTLET) 设备模块类对象 616。与所述反应器输出设备模块类对象 616 相关联的运行逻辑 (Run_Logic) 在屏幕部分 652 中被示为包括驱动元件 (DRIVE)、监视元件 (MONITOR)、读回元件 (READBACK) 以及更重要地,包括状态驱动算法,该算法具有用于控制与反应器输出设备模块类对象 616 相关的阀门的四个单独的状态。右侧屏幕部分 654 包括关于驱动项的细节,所述驱动项包括用于开启和关闭实际阀门的控制程序或控制模块中的驱动项排水 _ SP (DRAIN_SP)、输出 _ SP (OUTLET_SP) 和产品 _ SP (PRODUCT_SP) 的名称,以及包括称为排水 (DRAIN)、输出 (OUTLET) 和产品 (PRODUCT) 模块中的每个控制模块目标 _ 状态 (TARGET_STATE) 的通信路径规定。读回路径 (Readback path) 规定为每个排水、输出和产品模块中的当前 _ 状态 (CURRENT_STATE) 的读回参数 (readback parameter) 提供了通信路径,并且也为反应器输出设备的不同状态提供阀门状态。因此,屏幕部分 654 内的驱动细节指示,在反应器输出 616 的关闭 (Closed) 状态操作过程中,排水、输出、和产品阀门全部被关闭 (Closed),在排水 (Draining) 操作过程中,排水和输出阀门打开 (Open) 而产品阀门被关闭 (Closed),并且在释放产品 (Released Product) 操作过程中,所述排水阀门关闭 (Closed) 而输出和产品阀门开启 (Open)。正如将要理解的,当图 12 的屏幕显示 650 图解为设备模块类对象规定状态驱动控制逻辑的一种方式时,也可以使用任何其他期望的方法。

[0178] 当然如果需要,这些和屏幕 9 — 12 相类似的屏幕,可由配置工程师使用以创建和规定剂量或其他阶段类中的控制算法,或为任何其他模块,如单元模块类对象、设备模块类对象和显示模块类对象创建任何期望的模块类对象。

[0179] 在如上所述创建一个或多个模块类对象后,所述配置工程师随后可使用这些模块类对象来配置加工厂 10 内的元件。图 13 图解了具有分层体体系结构视图 661 的屏幕显示 660,该屏幕示出了加工厂 10 的系统配置 (System Configuration) 662。所述系统配置 662 包括称为控制策略 664 的区,该区为加工厂 10 规定了控制策略,并特别地,用于控制加工厂 10 的过程实体。所述控制策略 664 (Control Strategies) 被划分成逻辑组,诸如被分成工厂 10 的物理区域。在视图 661 中图解了区域 _ A (AREA_A) 665。AREA_A665 包括称作食盐 (SALTS) 的食盐制造区 666,该区域可以包括多个用于生产食盐的复制设备的部件或实例。

[0180] 为了配置工厂 10 的食盐制造区,配置工程师可以选择图 8 的单元模块类库 604 中的反应器单元模块类对象 610,将其拖拉至或否则拷贝至 SALTS666 下,以创建由反应器单元模块类对象 610 所指定的形式的反应器实例。这些反应器在图 13 中图解为反应器 _ 1 (REACTOR_1) 单元模块对象 668。所述配置应用程序 50 可以基于创建单元模块对象的单元模块类对象的名称自动地命名反应器 _ 1 单元模块对象 668,但在某种意义上,名称在配置系统是唯一的。如图 13 所示,反应器 _ 1 单元模块对象 668 包括酸 _ 1 (ACID1) 累加器

设备模块对象、碱_1(ALKALI1) 累加器设备模块对象和输出_2(OUTLET2) 设备模块对象,它们对应于在反应器单元模块类对象 610 中所指定的酸、碱和输出模块类对象。更进一步,如所述模块类对象 610 所规定的,所述反应器_1 单元模块对象 668 包括剂量(DOSE)、排水(DRAIN)、冲洗(FLUSH) 和混合(MIX) 单元阶段(由阶段类开发出的)。在称作别名的文件夹中提供别名解析表,它包括所述反应器_01 单元模块对象 668 中任意处的控制程序中所使用的别名的解析列表。所述反应器_1 单元模块对象 668 内的别名解析表使得控制程序能在运行时间之前或期间被解析,以便与加工厂 10 中的特定设备相联结。在美国专利第 6,385,496 号中描述了单元阶段和别名解析表的使用,所述专利被转让给本专利的受让人,并在此特别地包含引作参考。更进一步,反应器_1 单元模块对象 668 包括水_WATER_HDR1(WATER_HDR1) 和进水(WATER_IN) 模块对象,其分别对应反应器单元类对象 610 中所指示的水累加器设备模块类对象和反应器单元类对象 610 的进水开 / 关阀门控制模块类对象。当然,所述配置应用程序 50 也可基于反应器单元模块类 610 中的相应元件名称,自动地提供反应器_01 单元模块对象 668 的元件所使用的命名方案,反应器_1 单元模块对象 668 的每个单独元件由反应器单元模块类对象 610 所创建,并在某种方式上使得这些名字在配置策略中是唯一的。

[0181] 正如从关于反应器单元模块类对象 610 的讨论中所理解的,所述 WATER_HDR1 是共用汇水管(对应于图 1 的进水阀门系统 110)。因此,在食盐分配 666 下的单独设备模块对象被提供为所谓的 WATER_HDR1670。当然,配置工程师可以通过将所述元件复制或拖到屏幕区 661,由图 8 的累加器设备模块类对象 614 创建 WATER_HDR1670 模块对象。正如在这种情况下所期望的,共用汇水管 WATER_HDR1670 被示为包括特定开 / 关阀门控制模块对象(称作 Coarse_Valve4 和 Fine_Valve4) 以及特定流量计(Flow_Meter) 控制模块对象(称作 Flow_Meter4),它们对应图 8 的累加器设备模块类对象 614 的粗 _ 阀门、精确 _ 阀门和流量 _ 计量表控制模块类对象。此外,所述反应器_1 模块对象 668 内的 WATER_HDR1 引用 WATER_HDR1 设备模块对象 670。

[0182] 当然,任何所述设备或控制模块在单元模块类对象中可被指定为共用或非共用的模块对象。非共用模块对象完全地被其中创建该非共用模块对象的高层模块对象所拥有。共用模块对象被一个以上的高层模块对象所拥有或与其相关联。模块对象的共用或非共用属性影响在资源管理器视图中对模块对象的描述。特别地,非共用模块对象分配使得模块对象仅在控制策略内高层对象下予以描述,而共用模块对象分配使得共用模块方块或模块对象在每个共用该元件的高层模块对象下或在资源管理器分层结构中的独立模块对象下予以描述。

[0183] 正如将要理解的,配置工程师能在配置系统的控制策略区 664 中,通过简单地复制反应器单元模块类对象 610 以及创建其实例(反应器单元模块对象 668),为整个反应器创建控制策略,所述实例具有反应器单元模块类对象 610 内指定的全部元件。然而,单个单元以及特别地反应器单元模块对象 668 的控制元件,通过使用从模块对象变化到模块对象的输入 / 输出指示,以及使用被建立以使配置工程师能够规定模块对象在加工厂中联结方式的配置对话框,来将所述单元与加工厂 10 的特定实体相连接或联结。同样,如上面所指出的,所创建的反应器模块对象 668 仍然被连接到反应器单元模块类对象 610 并由其拥有,使得反应器单元模块类对象 610 内进行的变化能够自动地传播至反应器_1 模块对象 668,

以及如果需要,可以自动地反映在反应器_1 模块对象 668 中。无论如何,配置工程师可以改变或个别地处理已创建的模块对象,诸如反应器_1 单元模块对象 668,以便在全局(对象范围)基础上或者逐元件,来接收或拒绝来自母模块类对象的变化。

[0184] 屏幕 660 的右手部分 672 示出或列出了与反应器_1 模块对象 668 中的全部元件(如在区 661 中所选定的元件)相关联的参数、警报等以及值,过滤和其他与此相关联的信息。当然,配置工程师可为反应器_1 模块对象 668 的每个元件按照所需改变或规定该数据,从而以任何期望方式处理由模块类对象所创建的单个模块对象。

[0185] 图 14 示出了屏幕显示 680,所述屏幕示出了反应器_1 (REACTOR_1) 单元模块对象 668 的酸 1 (ACID1) 设备模块对象 682 的配置。因为在图 14 中选择了所述酸 1 设备模块对象 682,所以在屏幕 680 的右手部分 684 中示出了其元件。这些元件包括粗_阀门 1 (COARSE_VALVE1)、精确_阀门 1 (FINE_VALVE1) 和流量_计量表 1 (FLOW_METER1) 控制模块对象以及显示的参数。

[0186] 弹出窗口 686 提供了关于与所述酸 1 设备模块对象相关的子模块对象以及模块类的信息,其中由所述模块类产生或创建所述子模块对象。特别是,所述酸 1 设备模块对象属于累加器设备模块类对象,粗_阀门 1 (COARSE_VALVE1) 和精确_阀门 1 (FINE_VALVE1) 模块对象属于图 8 的库区域 608 的开/关阀门控制模块类对象,以及 Flow_Meter1 模块对象属于图 8 的库区域 608 中的流量_计量表控制模块类对象。

[0187] 所述酸 1 配置弹出屏幕 686 的图表(tabbed)显示,还包括参数、IO(输入/输出)、警报和现场总线分配。所述参数表可用于规定或填充反应器_1 模块对象 668 的酸 1 累加器的每个元件的参数。所述 IO 表可用于规定反应器_1 模块对象 668 的酸 1 累加器模块对象 682 的每个元件之间的输入/输出连接,以便将各个元件与加工厂 10 中的实际设备相联结。如果需要,如图 5 中所示的,可使用电子数据表算法或程序规定这些联结,以便在全局或批处理基础上规定模块对象(在任意级)的全部元件和加工厂 10 内的实际元件之间的相互连接。图 15 的电子表格显示 688 示出了电子表格视图,使配置工程师能够同时将来自不同累加器设备模块的粗阀门与加工厂 10 内的设备进行联结。在这种情况下,为称作 VLV-101A、VLV-201A、VLV-301A 和 VLV-401A 的阀门规定 IO_输出 (IO_OUT) 路径名和 IO_读回 (IO_READBACK) 路径名称。但是,如果需要,能够以批方式规定相同模块的不同元件。

[0188] 再次参照图 14,弹出屏幕 686 的警报表可用于规定或设置与酸_1 模块对象相关联的警报,并且现场总线表可根据需要规定现场总线设备的连接。当然,类似的配置屏幕能用于反应器_1 模块对象 668 中的其他模块对象和整个反应器_1 模块对象 668。同理,相同或类似的配置屏幕可用于其他模块对象。应当理解,所述配置屏幕可在模块对象的任何层次上显示和使用,所述层包括单元模块层、设备模块层、控制模块层等。

[0189] 图 16 图解了屏幕显示 690,其中包括加工厂 10 的配置系统的资源管理器窗口 661,配置工程师已增加了另外的反应器模块对象 692,称为反应器_1_1 (REACTOR_1_1),所述反应器由图 8 的反应器单元模块类对象 610 复制或创建。当然,当创建反应器_1_1 模块对象 692 时,配置工程师可以规定不同的参数、到不同设备的不同连接以及甚至将用于其各个子元件的不同控制程序。如图 16 中所示,反应器_1_1 模块对象包括图 8 的反应器单元模块类 610 中的每个设备模块类对象的子对象,其包括酸 2 (ACID2)、碱 2 (ALKALI2) 和输出 4 (OUTLET4) 设备模块对象。反应器_1_1 模块对象也包括进水 2 (WATER_IN2),水位_计

量表 2 (LEVEL_METER2) 控制模块对象, 和称作水_HDR1 (WATER_HDR1) 和剂量 (DOSE)、排水 (DRAIN)、冲洗 (FLUSH) 和混合 (MIX) 单元阶段的共用设备模块对象的引用。尽管使用公用汇水管的名称示出了共用单元, 如 WATER_HDR1 模块对象, 但是将为仅与反应器_1_1 模块对象 692 相关联的子对象给出唯一的名称。

[0190] 应当理解, 配置工程师以类似方式, 根据这里所述的原理, 可以使用任何其他的单元模块类对象、设备模块类对象和控制模块类对象以及显示模块类对象, 以为过程控制环境中的单元、设备、控制元件和显示元件建立配置元件。也应当理解, 使用这些模块类对象, 本质上这些模块类对象是非常详细的, 这将非常有利于配置工程师创建配置, 这是因为工程师不必分别地创建每个单独的控制元件或分别从控制模板复制每个单独的控制元件, 而是, 工程师可使用在更高层或范围提供的模块类对象来建立更大或更大批的、用于配置加工厂 10 的配置项。

[0191] 更进一步, 配置工程师可以在全局的基础上, 通过改变一个或多个单元模块类对象以及使这些改变传播至由这些单元模块类对象创建和以及与这些单元模块类对象相关联的每个模块对象, 对不同过程实体的配置的元件作出改变。在已经创建配置之后, 该特征使得配置内的改变更容易以及耗时更少。此外, 配置工程师可以通过设定模块类对象内的安全参数, 来规定配置系统内部模块对象的不同元件或部件的访问等级。如上所述, 配置工程师通过任何层次上的模块基础来规定关于模块的安全性, 所述层次如单元模块层、设备模块层、控制模块层和显示模块层。以这种方式, 单元模块对象的一些元件可以是可查看的, 同时其他的元件可以是不可查看的。

[0192] 当然, 一旦完成所述配置系统并且模块对象与加工厂 10 内的单独过程实体相联结, 则可以将所述控制和显示模块或与这些模块相关联的元件下载到图 1 的适当的控制器 12 和工作站 14, 以便在加工厂 10 的运行期间执行。

[0193] 图 17 是实例程序 700 的流程图, 该程序用于设置加工厂项目以限制对项目的访问。所述程序 700 可由一个或多个计算系统, 如图 1 的加工厂 10 的一个或多个工作站 14 和 / 或控制器 12 执行。所述程序 700 也可以由与所述加工厂 10 独立的计算系统执行。例如, 第三方提供者能使用程序 700 来限制对例如由第三方提供者所开发的模块对象、模块类对象、模块模板、复合模板等的访问。然后, 所开发的模块对象、模块类对象、复合模板等可以由第三方提供者的顾客使用, 但所述顾客不能例如查看已开发的模块对象、模块类对象、复合模板等的部分或全部内部项目。通常, 将被配置的加工厂项目例如可以是模块对象、模块类对象、复合模板。所述程序 700 可作为配置应用程序, 如配置应用程序 50 的一部分而被执行。例如, 可在模块对象、模块类对象、复合模板等的配置期间或之后使用配置应用程序 50 执行所述程序 700。同样, 程序 700 可由与配置应用程序 50 相独立的应用程序执行。

[0194] 在方块 702, 接收用户输入信号, 该信号指示将要控制对加工厂项目的访问 (也就是, 与加工厂项目相关联的数据将受到保护)。例如, 配置所述项目的人员可以选择框、选择按钮、选定菜单项目等, 其指示与所述项目相关联的将受到保护的数据。用户输入信号也可选地指示与所述项目相关联的将受到保护的数据。例如, 所述用户可以选择保护部分数据而使得其他数据不受到保护。

[0195] 可替换地, 可使得用户不能选择将被检测的数据。例如, 用户仅能在保护或不保护项目之间进行选择。在一个实例中, 如果用户选择保护与模块对象或模块类对象相关联的

数据，则与模块对象或模块类对象的一组数据可受到保护，反之，不在该组的数据未受到保护。例如，构成模块对象或模块类对象的功能块及其它们的互连将受到保护，但模块对象或模块类对象的参数将不会受到保护。

[0196] 图 18A 和 18B 是示例性的屏幕显示，它可由配置应用程序，如图 1 的配置应用程序 50 所创建。可通过其他类型的应用程序，如用于在过程运行期间监视过程的应用程序、用于调试配置的调试应用程序、资产管理应用程序等应用程序创建类似的屏幕显示。屏幕显示 720 包括具有复合模板的库 724 的资源管理器视图的一部分 722。一个复合模板是 DYN_FULL 复合模板 726。所述 DYN_FULL 复合模板 726 包括各种子元件，因为在资源管理器视图 722 中选择了 DYN_FULL 复合模板 726，所以在屏幕显示 720 的右手侧 728 中描述所述各种子元件。构成 DYN_FULL 复合模板 726 的子元件包括参数和功能块 730。

[0197] 在屏幕显示 720 中，所述 DYN_FULL 复合模板 726 没有设置成受保护，并且在屏幕显示 720 的右手侧 728 中描述了功能块 730。相反，图 18B 示出了对应设置成受保护的 DYN_FULL 复合模板 726 的屏幕显示 732。因此，在所述屏幕显示 732 中，在右手侧 728 中没有描述功能块 730（即，它们被隐藏了）。尽管 DYN_FULL 复合模板 726 已经被设置为受保护，仍描述了其参数。

[0198] 再次参照图 17，在方块 704，提示用户输入密码。例如，窗口、屏幕、音频提示等，都可提示用户输入密码。如下文所述，可在随后的位置及时地使用所述密码以访问所保护的数据和 / 或重新配置所述项目以取消保护。在方块 706，接收所述密码。例如，可通过计算系统的键盘、袖珍键盘、触摸屏、光笔、鼠标等输入所述密码。

[0199] 在方块 708，与将受到保护的加工厂项目相关联的数据的指示可被存储，并与所述项目相关联。例如，如果项目是一个对象，则所述指示可被存储作为该对象的一部分。作为另一实例，所述指示可以与所述对象分开存储，并且与所述指示的链接可被存储作为对象一部分。仍为另一实例，如果加工厂项目是由模块类对象建立的项目，则所述指示可以作为模块类对象的一部分被存储。如果不允许用户选择那些数据将受到保护，则方块 708 可被忽略。

[0200] 在方块 710 处，在方块 706 接收的密码可被存储并与所述项目相关联。例如，如果项目是一个对象，则所述密码可作为该对象的一部分被存储。作为另一实例，所述密码能与所述对象分开存储，并且与所述密码的链接可作为对象一部分被存储。作为另一实例，如果加工厂项目是由模块类对象创建的项目，则所述密码可作为模块类对象的一部分被存储。

[0201] 可选地，可以加密将要被保护的数据和 / 或密码。在该执行过程中，方块 708 和 710 可以包括存储和将加密的数据和 / 或加密的密码相关联。进一步，如果加工厂项目是由模块类对象创建的项目，则所述被保护的数据或加密的数据可以作为模块类对象的一部分被存储。

[0202] 如果所述项目是模块类对象，那么可以将被保护的数据和密码的指示传播到由模块类对象创建的项目。

[0203] 类似于程序 700 的程序可用于“无保护”的加工厂项目或与所述项目相关联的特定数据。也就是，可以接收指示与将被保护的加工厂项目相关联的数据的用户输入。然而，可以提示所述用户输入密码。如果所接收的密码与在程序 700 的块 710 的保护加工厂项目相关联的、所存储的密码相匹配，则可以提示用户关于受保护数据的那些（是否全部的）将

不受保护。进而,与在程序 700 的方块 708 的项目相关联的、所存储的指示,可以被修改以指示不将保护部分或全部的数据。

[0204] 图 19 是实例程序 750 的流程图,该程序用于设置加工厂项目的组以限制对该组的访问。所程序 750 可由一个或多个计算系统,如图 1 中加工厂 10 的一个或多个工作站 14 和 / 或控制器 12 所执行。该组中的加工厂项目可以是,例如模块对象、模块类对象和 / 或复合模板。所述程序 750 可作为配置应用程序,如配置应用程序 50 的一部分而被执行。例如,可在该组中的项目的配置之后,使用配置应用程序 50 执行所述程序 750。同样,程序 750 可由与配置应用程序 50 相独立的应用程序执行。

[0205] 在方块 752,接收用户输入,该输入指示对项目的组的访问将受到限制(也就是,加工厂项目的组将受到保护)。例如,可保护对与该组中的每个项目相关联数据的访问。另外,可以不允许向该组增加项目和从该组删除项目。该组项目可以包括,例如项目的库、项目的文件夹、项目的类、项目的种类等。

[0206] 配置该组项目的人员可通过,如高亮显示该组项目、选择库名、文件夹名、类名、种类名等首先选定该组。参照图 18A,一个人可以例如,选择子库 734、选择种类 736、高亮显示在种类 736 中的多个模块等。然后,该人可以如选择框、选择按钮、选定菜单项目等,其指示将要被保护的组。

[0207] 再次参照图 19,在方块 754,提示用户输入密码。例如,窗口、屏幕、音频提示等可提示用户输入密码。如下文所述,可在随后的位置及时地使用所述密码以便访问被保护的组和 / 或重新配置该组以取消保护。在方块 756,接收所述密码。例如,可通过计算系统的键盘、袖珍键盘、触摸屏、光笔、鼠标等输入所述密码。

[0208] 在方块 758,该组项目将要被保护的指示可被存储并与该项目组相关联。例如,如果所述组是项目的种类,则将要被保护的指示可被存储并且该指示与种类相关联,从而如果随后将尝试修改该种类,则可确定该种类是受到保护的。另外,指示可与该组中的每个项目相关联。例如,如果项目是对象,则位于被保护组中的项目的指示可被存储作为每个对象的一部分。同样,所述指示可与每个对象分开存储,并且对所述指示的链接可作为每个对象的一部分被存储。

[0209] 在方块 760,在方块 756 所接收的密码可被存储并与所述项目组相关联。例如,如果组是项目的种类,则可存储所述密码并且该密码与种类相关联。另外,所述密码可与该组中的每个项目相关联。例如,如果项目是对象,密码可作为每个对象的一部分被存储。同样,所述密码可与每个对象分开存储,并且对所述密码的链接可作为每个对象的一部分被存储。

[0210] 如图 17 的程序 700,类似于程序 750 的程序可被用于使加工厂项目组“不受保护”。

[0211] 图 20 是用于确定是否允许查看与加工厂项目相关联的数据的实例程序 770 的流程图,所述加工厂项目可以或不可以受保护。所述程序 770 可由一个或多个计算系统,如图 1 中加工厂 10 的一个或多个工作站 14 和 / 或控制器 12 执行。所述加工厂项目例如可以包括模块对象、模块类对象、复合模板。所述程序 770 可作为配置应用程序,如配置应用程序 50 的一部分被执行。同样,所述程序 770 可作为另一应用程序,如监视应用程序、调试应用程序、资产管理应用程序等的一部分而被执行。

[0212] 在方块 772,可以接收查看与加工厂项目相关联数据的请求。例如,用户可以在资

源管理器窗口中选择一个项目,试图“钻入 (drill down)”到模块中以查看模块的部件等。参照图 18B,作为一个实例,用户在屏幕显示 732 中的资源管理器窗口的部分 722 中,已经选定 DYN_FUL 复合模板 726。因此,DYN_FULL 复合模板 726 的子元件将显示在屏幕显示 732 的右手部分 728 中。

[0213] 在方块 774,确定是否保护任何与该项目相关联的数据。例如,可以确定是否存在与所述项目相关联的指示,如在图 17 中的方块 708 存储的指示,它指示保护与所述项目相关联的数据。作为另一实例,可以确定是否存在与所述项目相关联的指示,如在图 19 中的方块 758 时所存储的指示,它指示保护作为项目组的一部分的项目。

[0214] 如果不保护与所述项目相关联的数据,则所述流程可进行到方块 776,在此显示与项目相关联的数据。例如,如果用户在资源管理器窗口中已经选定了该项目,则该项目的子元件可在显示屏的一部分中显示。参照图 18A,作为一个实例,用户在屏幕显示 720 中的资源管理器窗口的部分 722 中已经选定 DYN_FUL 复合模板 726。因此,包括功能块 730 的 DYN_FULL 复合模板 726 的子元件被显示在屏幕显示 720 的右手部分 728 中。作为另一实例,如果用户请求“钻入”到模块中以查看该模块的部件,那么可在显示屏或窗口中显示该模块的部件。

[0215] 如果在方块 774 确定保护与所述项目相关联的数据,则流程可进行到方块 786,在此不显示与所述项目相关联的被保护数据,但可显示未受保护的数据。

[0216] 可选地,如果在方块 774 确定保护与项目相关联的数据,则流程可以进行到方块 778。在所述方块 778,使用如窗口、显示屏、音频提示等提示用户输入密码。在方块 780 处,接收由用户输入的密码。例如,可通过计算系统的键盘、袖珍键盘、触摸屏、光笔、鼠标等输入所述密码。在方块 782 处,比较在方块 780 处所接收的密码与所存储的、并与所述项目相关联的密码。例如,所接收到的密码可与在图 17 的方块 710 时所存储的密码,或与在图 19 的方块 760 时所存储的密码相比较。

[0217] 在方块 784 处,可以确定是否在方块 780 所接收的密码与所存储的、并与所述项目相关联的密码相匹配。如果所述密码相配,可在方块 776 显示与所述项目相关联的数据。如果所述密码不匹配,不显示与所述项目相关联的被保护数据,而在方块 786 显示未被保护的数据。参照图 18B,例如,在显示屏 732 的右手部分显示 DYN_FULL 模块 726 的参数,但没有显示 DYN_FULL 模块 726 包含的功能块。

[0218] 应当理解,方块 778、780、782、和 784 是可选的并可被省略。在这种情况下,如上所述,如果在方块 774 确定保护与项目相关联的数据,则流程可以进行到方块 786。

[0219] 图 21 是用于确定是否允许修改加工厂项目或加工厂项目组的实例程序 800 的流程图,所述加工厂项目组可以或不可以被保护。所述程序 800 可由一个或多个计算系统,如图 1 中加工厂 10 的一个或多个工作站 14 和 / 或控制器 12 执行。所述加工厂项目例如可以包括模块对象、模块类对象和 / 或复合模板。所述程序 800 可作为配置应用程序,如配置应用程序 50 的一部分被执行。同样,所述程序 800 可作为另一应用程序,如资产管理应用程序的一部分被执行。

[0220] 在方块 802,可以接收修改加工厂项目或加工厂项目的组的请求。例如,用户可以在资源管理器窗口中选择修改的项目,选择修改的项目的部件等。作为另一实例,用户可以尝试从项目组中删除一个项目,将一个新的项目增加到项目组中等。

[0221] 在方块 804，确定是否保护项目或项目组。例如，可以确定是否存在与所述项目相关联的指示，如在图 17 的方块 708 处存储的指示，它指示保护所述项目。作为另一实例，可以确定是否存在与所述项目相关联的指示，如在图 19 的方块 758 存储的指示，它指示保护作为项目组一部分的项目。作为另一实例，可以确定是否存在与所述项目相关联的指示，如在图 19 的方块 758 处存储的指示，它指示保护项目组。

[0222] 如果与所述项目或项目组相关联的数据不被保护，则所述流程可进行到方块 806，在此允许进行修改。可是，如果在方块 804 确定保护项目或项目组，则所述流程可进行到方块 808。在方块 808，可产生指示用户正试图修改的所述项目或项目组受到保护的通知（如，窗口、显示屏、音频提示等）。可选地，所述方块 808 可被省略。在这种情况下，可能以这种方式显示项目或项目组，以便向用户指示该项目或项目组受到保护。例如，可以将显示屏上的项目或项目组描述为“灰色”，包括指示项目或组受到保护的符号等。在另一实例中，可不存在向用户指示项目或组受保护，在这种情况下，用户可以仅仅不能修改所述项目或组。在方块 808 之后，或如果方块 808 被省略，所述程序结束。

[0223] 可选地，所述流程可以进行到方块 810。在所述方块 810，使用如窗口、显示屏、音频提示等提示用户输入密码。在方块 812，接收由用户输入的密码。例如，可通过计算系统的键盘、袖珍键盘、触摸屏、光笔、鼠标等输入所述密码。在方块 814，可以比较在方块 812 所接收的密码和所存储的、与所述项目或项目组相关联的密码。例如，可以比较所接收的密码与在图 17 的方块 710 所存储的密码，或在图 19 的方块 760 所存储的密码。

[0224] 在方块 816，可以确定是否在方块 812 所接收的密码与所存储的、与所述项目或项目组相关联的密码相匹配。如果所述密码相匹配，可在方块 806 允许修改。如果密码不相匹配，可在方块 818 处产生不允许修改的通知。可替换地，或另外，所述通知可指示出在方块 812 所接收的密码不是正确的密码。

[0225] 如上所述，方块 810、812、814、816 和 818 是可选的并可省略，在这种情况下，在方块 808 之后所述程序可以结束。同样，方块 808 是可选和可省略的，在这种情况下，如果在方块 804 确定所述项目或组受到保护，则所述程序可以结束或进行到可选方块 810。

[0226] 虽然在图 17 和 19-21 中，已经论述了用于限制对加工厂项目进行访问的密码，但是可替换地或另外地使用其他除了密码之外的用于限制访问的技术。例如，可以使用指示被授权访问项目人员的预先授权 (preauthorized) 的安全访问等级，被授权访问项目的特定人的指示的列表（如登录名、ID 号等）、生物学数据 (biometric) 等。进一步，与保护等级有关的附加数据可以与项目或项目组，诸如，是否禁止所有访问，是否允许只读访问，不同人的不同等级保护和 / 或消除等级等相关联。此外，在某些实现过程中，可允许用户选择用于项目的子元件的不同级别的安全性。通常，可存储访问安全数据（如，密码、生物学数据、预授权的安全访问等级、被授权的人和 / 或 ID 的列表等）并将其与加工厂项目相关联，并且这些访问安全数据可与用户安全数据（如，由用户输入的密码、用户的 ID、用户的登录 ID、从用户获得的生物学数据等）相比较，以确定是否允许用户访问项目和 / 或项目的特定子元件。

[0227] 参照图 17，如果需要，可以省略方块 704、706 和 710。例如，用户可以指出加工厂项目或与所述加工厂项目相关联的数据将受保护。然后，用户可以使用类似于图 19 的程序 750 的程序来保护一组项目，其中包括用户希望保护的加工厂项目。这样，用户可以阻止其

他人改变项目的配置,以便查看所保护的数据。

[0228] 图 22 是用于输出加工厂项目的实例程序 830 的流程图。所述程序 830 可由一个或多个计算系统,如图 1 的加工厂 10 的一个或多个工作站 14 和 / 或控制器 12 执行。同样,所述程序 830 也可以由与所述加工厂 10 独立的计算系统执行。例如,所述程序 830 可由第三方提供者的计算系统或其他加工厂的计算系统执行,以便向顾客或加工厂 10 输出加工厂项目。所述加工厂项目例如可以包括模块对象、模块类对象、复合模板。所述程序 830 可作为配置应用程序,如配置应用程序 50 的一部分被执行。同样,所述程序 830 可作为另一应用程序,如资产管理应用程序或第三方提供者所使用的应用程序的一部分而被执行。

[0229] 在方块 832,可以接收输出加工厂项目的请求。例如,用户可在资源管理器窗口中选择用于输出的项目。作为另一实例,用户可以选择用于输出的一组加工厂项目,并且为该组中的每个项目执行所述程序 830。

[0230] 在方块 834,确定是否保护任何与项目相关联的数据。例如,可以确定是否存在与所述项目相关联的指示,如在图 17 的方块 708 所存储的指示,它指示保护与所述项目相关联的数据。作为另一实例,可以确定是否存在与所述项目相关联的指示,如在图 19 的方块 758 所存储的指示,它指示保护作为项目组一部分的项目。如果与所述项目相关联的数据不受保护,则所述流程可能进行到方块 836,在此可输出所述项目。

[0231] 可是,如果在方块 834 确定保护与项目相关联的数据,那么所述流程可进行到方块 838。在方块 838,可以保护所述受保护的数据。然后,在方块 840,可以加密与所述项目相关联的密码。例如,可以加密在图 17 发方块 706 处所接收的密码或在图 19 的方块 756 处所接收的密码。

[0232] 在方块 842,输出所述项目,但其中使用加密的保护数据替换保护的数据以及用加密的密码替换密码。如果加工厂项目是由模块类对象建立的项目,则加密的数据和 / 或加密的密码可以作为例如模块类对象或项目的一部分被输出。

[0233] 图 23 是实例配置系统 900 的方框图,它可用于提供与加工厂相关联的版本控制和检查跟踪信息。可使用版本控制和检查跟踪技术的其他方面,在转让给本申请的受让人的、专利号为第 6,449,624 号、题为“过程控制系统中的版本控制和检查跟踪”以及的美国专利中予以描述。应当理解,不需要使用美国专利第 6,449,624 号中所描述的技术,而还可以使用不同的技术。为了所有目的,将美国专利 No. 6,449,624 的全部内容在此包含引作参考。

[0234] 配置系统 900 包括例如可以在工作站 14 或其他计算装置(图 1)的显示设备上产生的用户接口 904。所述用户接口 904 可以使得用户能够通过一个或多个配置应用程序 906 来配置加工厂或加工厂的一部分。所述配置应用程序 906 可以包括配置应用程序,如配置应用程序 50(图 1)。用户接口 904 也与版本控制和检查跟踪系统 908(在下文为“VCAT 系统”)相接口,其通常与配置应用程序 906 合作以记录和管理关于加工厂或加工厂一部分配置的历史信息。配置应用程序 906 和 VCAT 系统 908 都访问并与配置数据库 912 分别通信,所述配置数据库可存储表示加工厂或加工厂一部分(在下文为“过程配置”)的当前配置的数据。所述 VCAT 系统 908 也与版本控制数据库 910 相通信。

[0235] 所述版本控制数据库 910 可以包括指示在过程配置中使用的每个项目的任何数量的在先版本的配置历史数据。总之,所有项目的历史数据可用于重构过程的历史配置。更具体而言,对于配置数据库 912 的每个项目(以及那些已经不在配置数据库 912 中的项

目),代表项目的配置的数据可被存储为多个版本。例如,因为自从建立所述项目以来,所以项目已经经历了三次修改。因此,所述版本控制数据库 910 在将要建立的时候包括指示项目配置的数据,其可被称作“版本 1”,以及在三次修改中每次之后指示项目配置的数据,其对应于“版本 2”、“版本 3”和“版本 4”。

[0236] 因此,所述配置历史数据可以包括表示对模块对象和模块类对象作出的全部修改的数据。所述修改可以但不必使用配置应用程序 906 进行。用户界面 904、配置应用程序 906 和 VCAT 系统 908 中的任意一个或全部可以被组合为单个的集成系统。可是,为了说明简便起见,任务在这里将被描述为分别归属于用户接口 904、配置应用程序 906 和 VCAT 系统 908。

[0237] 可使用一个或多个计算装置,如工作站 14(图 1),以通过允许对过程配置的修改进行监视的方式来实现 VCAT 系统 908。可在与 VCAT 系统 908 相同的(多个)计算装置上,或在与执行 VCAT 系统 908 的(多个)计算装置相通信的不同计算装置上执行用户接口 904 和配置应用程序 906。

[0238] 在数据库 910、912 其中之一或两者中的数据可存储在计算机可读介质中,所述介质如易失性存储器或与工作站 14 相关联的磁性或光存储介质,并物理地位于加工厂 10 内的任何位置。可替换地,数据库 910、912 其中之一或两者可被存储在远端位置中,使得计算装置如工作站 14 通过网络如企业内部网、因特网或任何其他通信介质访问存储在其中的数据。此外,存储在每个数据库 910、912 中的数据不必存储在同一计算机可读介质中,从而使得数据库 910、912 之一的任意部分可存储在不同于存储其他部分的设备或者介质的存储器装置或介质中。

[0239] 在图 23 中,所述 VCAT 系统 908 被示为不同于所述版本控制数据库 910 并与其相独立。可替换地,所述版本控制数据库 91C 可来自 VCAT 系统 908 的一部分。类似地,所述配置数据库 912 和所述版本控制数据库 910 可以但不是必需组成单独而不同的数据结构。也就是说,所述数据库 910、912 可以位于相同的存储介质中,实际上,可构成加工厂 10 专用的公共数据库的部分。相应地,如在此使用的“数据库”,应当理解为不限制于任何特定的数据结构。

[0240] 图 24 是实例程序 930 的流程图,其用于方便加工厂项目版本的跟踪。所述加工厂项目例如可以包括模块对象和 / 或模块类对象。所述程序 930 例如可由图 23 的 VCAT 系统 908 实现,并将参照图 23 对此予以说明。

[0241] 在方块 932,可检测对加工厂项目的修改。例如,用户可以使用配置应用程序 906 试图为处理配置添加一个新的项目。作为另一实例,在修改完所述项目后,用户可以请求将项目保存到配置数据库 912 中。作为另一实例,在修改完所述项目后,用户可以请求将项目“登入 (check-in)”到配置数据库 912 中。美国专利第 6,449,624 号描述了与可使用的“登入”和“登出 (check-out)”方法有关的技术。

[0242] 在方块 934,可以确定,如果有的话,用于创建所述被修改项目的模块类对象。例如,第一模块类对象可以已经用于创建第二模块类对象,它依次用于创建所述项目。如果一个或多个模块类对象被用于创建所修改的项目,则在方块 936,可以确定所使用的每个模块类对象的版本。

[0243] 在方块 938,生成关于修改项目的版本标识符。所述版本标识符可以包括数字、字

母、符号等。所述版本标识符指示反映在方块 932 所检测到的修改版本。例如,如果在方块 932 检测到的修改是项目的创建,则版本标识符指示最初的版本。作为另一实例,如果在方块 932 处检测的修改是项目的第一版本的修改,则版本标识符可指示第二版本。

[0244] 在方块 940 处,在方块 938 所产生的版本标识符可以被存储并与所修改的项目相关联。例如,如果项目是一个对象,则所述版本标识符可作为该对象的一部分被存储。作为另一实例,所述版本标识符可以与所述对象分开存储,并且对所述版本标识符的链接可以作为该对象的一部分而被存储。

[0245] 在方块 942,如果有的话,可以存储在方块 936 所确定的一个或多个模块类对象的(多个)版本标识符,并将其与所述修改的项目相关联,其中由所述模块类对象创建所述项目。例如,如果项目是一个对象,则(多个)版本标识符可作为该对象的一部分被存储。作为另一实例,(多个)版本标识符可以与所述对象分开存储并且所述(多个)版本标识符的一个或多个链接可以作为所述对象的一部分而被存储,在另一实例中,可以使用在方块 936 处所确定的版本标识符生成在方块 938 所生成的版本标识符,例如,在方块 936 处所确定的版本标识符可在方块 938 处所生成的版本标识符中被编码。在这种情况下,所述方块 940 可以包括方块 942。

[0246] 图 25 是实例程序 948 的流程图,其用于方便跟踪加工厂项目版本。所述加工厂项目例如可以包括模块对象和 / 或模块类对象。所述程序 948 例如可以由图 23 的 VCAT 系统 908 执行。

[0247] 在方块 950 处,可以确定是否已经创建模块类对象的新版本。例如,可以在新版本模块类对象已经“登入”时进行检测。在方块 952 处,可以确定通过使用模块类对象所创建的加工厂项目。参照图 6,例如,可以检查存储器 526 以便确定由模块类对象所创建的模块对象。同样,可以检查存储器 526 或其他存储器或区域以便确定由模块类对象所创建的其他模块对象。

[0248] 在方块 954 处,模块类对象的新版本或对模块类对象所作的改变,可被传播至方块 952 处所确定的项目中。然后,在方块 958 处,可以存储模块类对象的新版本的版本标识符并将其与方块 952 所确定的项目相关联。在一个实施例中,在方块 952 处所确定的项目的版本标识符未被更新以反映该项目的新版本。可选地,在一个实例中,在方块 952 处所确定的项目的版本标识符在方块 959 处被更新。然后,可存储项目的新版本标识符并将其与所述项目相关联。

[0249] 在图 24 的方块 940 和 942 处以及图 25 的方块 958 和 959 处所存储的版本标识符,可帮助用户跟踪在当前配置正在使用的项目的版本、创建所述项目的模块类对象的版本。例如,配置应用程序、资产管理应用程序等可以检查配置数据库 912(图 23),以便确定项目版本和创建项目的模块类对象的版本。作为另一实例,调试应用程序、监视应用程序等可以检查例如控制器 12 的存储器,以便确定项目版本和创建项目的模块类对象的版本。

[0250] 图 26 是实例程序 960 的流程图,它用于方便配置加工厂项目。所述程序 960,例如可在配置应用程序将加工厂项目显示在显示屏上时予以执行。所述加工厂项目例如可以包括模块对象和 / 或模块类对象。所述程序 960 例如可由图 23 的配置应用程序 906 和 / 或 VCAT 系统 908 执行,并参照图 23 予以说明。

[0251] 在方块 962 处,如果有的话,可以确定用于建立所述修改的项目的模块类对象。例

如,第一模块类对象可以已经用于建立第二模块类对象,其依次用于建立所述项目。如果一个或多个模块类对象被用于建立所述项目,则在方块 964 处,可以确定这些模块类对象中的每个模块类对象的当前版本。在方块 966 处,可以确定用于建立所述项目的每个模块类对象的版本。例如,可以检索在图 24 中的方块 942 处所存储的并与所述项目相关联的版本标识符。

[0252] 在方块 968 中,可为用于建立所述项目的每个模块类对象确定,是否模块类对象的当前版本不同于用于建立所述项目的模块类对象的版本。如果没有差别,则程序结束。可是,如果检测到一个或多个差别,可在方块 970 处产生通知。所述通知一般可以指示用于建立所述项目的模块类对象的新版本是可用的。所述通知可包括文本、符号、颜色、音频提示等。例如,文本框或窗口可相邻显示或与配置显示屏上项目的图形表示重叠显示。作为另一实例,符号可在项目图示上或其附近予以显示。作为另一实例,可改变项目表示的颜色、亮度等。通知本身可指示(多个)模块类对象具有一个可用的新版本。可替换地,所述通知可不告知用户(多个)模块类对象具有可用的新版本,但用户可以通过执行附加的步骤,例如选定菜单项目、“双击”符号、“钻入”到所述项目中等来确定该信息。

[0253] 可选地,用户因此可以被提供用户界面机制,用于启动处理以通过使用模块类对象的新版本更新项目来修改所述加工厂项目。例如,用户可以配有例如,用于启动所述处理的按钮。作为另一实例,所述用户可以配有用户界面机制以选择将被用于更新加工厂项目的具有新版本的可能的几个模块类对象的其中之一。例如,用户可以选择框、高亮显示对应于包含新版本的模块类对象的图形表示等,其中这些模块类对象用于更新加工厂项目。

[0254] 图 27 是实例程序 980 的流程图,它用于方便配置加工厂项目。例如,可在加工厂项目正被修改和/或“登入”时,执行所述程序 980。所述加工厂项目例如可以包括模块类对象。所述程序 980 例如可由图 23 的配置应用程序 906 和/或 VCAT 系统 908 执行,并参照图 23 予以说明。

[0255] 在方块 982 处,可以确定是否已经建立模块类对象的新版本。例如,可以在新版本模块类对象已经“登入”时进行检测。在方块 984 处,可以确定通过使用类对象建立的加工厂项目。参照图 6,例如,可检查存储器 526 以便确定由模块类对象建立的模块对象。同样,可检验查存储器 526 或其他存储器或区域以便确定由模块类对象建立的其他模块对象。

[0256] 在方块 986,可以提示用户有关是否应将对模块类对象的改变传播至在方块 984 所识别的一个或多个加工厂项目。例如,文本框、屏幕、音频提示等可用于提示用户。

[0257] 可选地,用户可以配有用户界面机制以便选择在方块 984 所识别的特定加工厂项目,其中所述改变被被传播。例如,用户可以选择框、高亮显示对应于加工厂项目的图形表示等,其中所述改变被传播至该加工厂项目。

[0258] 在方块 988,接收对由方块 986 生成的用户对提示的响应。在方块 990,可确定是否用户响应指示将传播所述改变。如果将不传播所述改变,则程序结束。如果将传播方块 992 的改变,则所述改变将被传播至方块 984 所确定的加工厂项目。可选地,所述改变可被传播至方块 984 处所确定的加工厂项目中的所选定的项目。

[0259] 尽管图 17、19—22 和 24—27 描述了有关加工厂项目,但类似的程序序可用于限制对将由加工厂项目执行的控制步骤的顺序有关的访问、跟踪对其产生的改变以及传播其相关的传送。例如,类似于图 17、19—22 和 24—27 中的程序,可用于限制对阶段类、单元阶

段和 / 或阶段类组或单元阶段组的访问,跟踪对其产生的改变和传播其相关的改变。

[0260] 图 28 是用于配置模块类对象的用户界面的实例屏幕显示 1000。在所述屏幕显示 1000 中,示出了称作“计算 (CALCULATION)”的模块类对象的细节。所述屏幕显示 1000 包括用于编辑所述对象的编辑部分 1004。如图 28 所示,所述计算模块类对象包括乘法器功能块 1008 和加法器功能块 1012。所述屏幕显示 1000 也包括部分 1020,该部分指示如果限制对计算模块类对象的访问,则将不会隐藏计算模块类对象的参数。将不被隐藏的参数包括,指示由计算模块类对象建立的模块对象版本的版本指示符 1026,以及指示计算模块类对象的版本的指示符 1028。

[0261] 图 29 是用于确定将不会被隐藏的计算模块类对象参数的用户界面的另一实例屏幕显示 1030。所述屏幕显示 1030 包括按钮 1032 和 1034,其用于分别增加和删除来自将不被隐藏的参数组的参数。所述屏幕显示 1030 示出了乘法器 (MULTIPLIER) 参数 1038 将被添加到未被隐藏的参数组中。同样参照图 28,所述乘法器参数 1038 对应于乘法器块 1008 的输入 “IN2”。

[0262] 图 30 是用于调试应用程序的用户界面的一个实例屏幕显示 1040。在所述屏幕显示 1040 中,示出了计算 _1 (CALCULATION_1) 模块对象的细节。所述计算 _1 模块对象已经由计算模块类对象所建立。在本实施例中,没有限制对计算模块类对象的访问。因此,屏幕显示 1040 通常示出了计算 _1 模块对象包括的子元件的细节。例如,部分 1044 示出了计算 _1 模块对象包括乘法器功能块 1008 和加法器功能块 1012。所述屏幕显示 1040 也包括指示计算 _1 模块对象的参数的部分 1048。例如,所述计算 _1 模块对象包括版本指示符 1026 和 1028 以及乘法器参数 1038。另外,所述屏幕显示 1040 也包括指示计算 _1 模块对象所包含的功能块的部分 1050。也就是,所述部分 1050 指示包括加法器功能块和乘法器功能块的计算 _1 模块对象。

[0263] 图 31 为在已经限制访问计算模块类对象的情形下的实例屏幕显示 1040。因此,屏幕显示 1040 通常没有示出计算 _1 模块对象包括的子元件的全部细节。例如,部分 1044 和 1050 没有示出计算 _1 模块对象包括乘法器功能块 1008 和加法器功能块 1012。可是,所述部分 1048 仍指示计算 _1 模块对象的参数。

[0264] 本领域技术人员应当理解,可以使用在此所描述技术的各种修改。例如,可以改变方块的顺序,省略一些方块,增加新的方块等。进一步,能够组合在此描述的各种技术。例如,类似于图 17 的程序 700 的程序,可用于保护与模块类对象相关联的一些数据。然后,可由模块类对象建立模块模板。接下来,类似于图 17 的程序 700 的程序,可用于保护与模块模板相关联的一些数据,所述模块模板相对于模块类对象未受到保护。然后,可以建立模块模板的特定实例,以及其包括有关于模块类对象和模块模板所规定的保护。

[0265] 当被执行时,在此所描述的任何软件可被存储在任何计算机可读的存储器中,诸如磁盘、光盘或其他存储介质,以及存储在计算机或处理器的 RAM 或 ROM 中等等。同样地,使用任何已知或期望的传送方法将该软件传送给用户、加工厂或操作员工作站,所述方法例如包括,以计算机可读磁盘或其他可传输的计算机存储装置或通过通信信道,所述通信信道诸如电话线、因特网、万维网、任何其他局域网或广域网等(其中,传送被视为与通过可传输的存储介质提供上述软件相同或与其可以互换)。此外,可以不调制或加密而直接提供该软件,或在通信信道上发送之前使用任何适合的调制载波和 / 或加密技术对软件进行

调制和 / 或加密。

[0266] 虽然已经参照具体实施例描述了本发明，但是所述实施例仅是为了示例性的目的而不是限制本发明，因此对于本领域普通技术人员而言将显而易见的是，在不脱离本发明的精神和范围的情况下，可以对所公开的实施例进行各种改变、添加或删除。

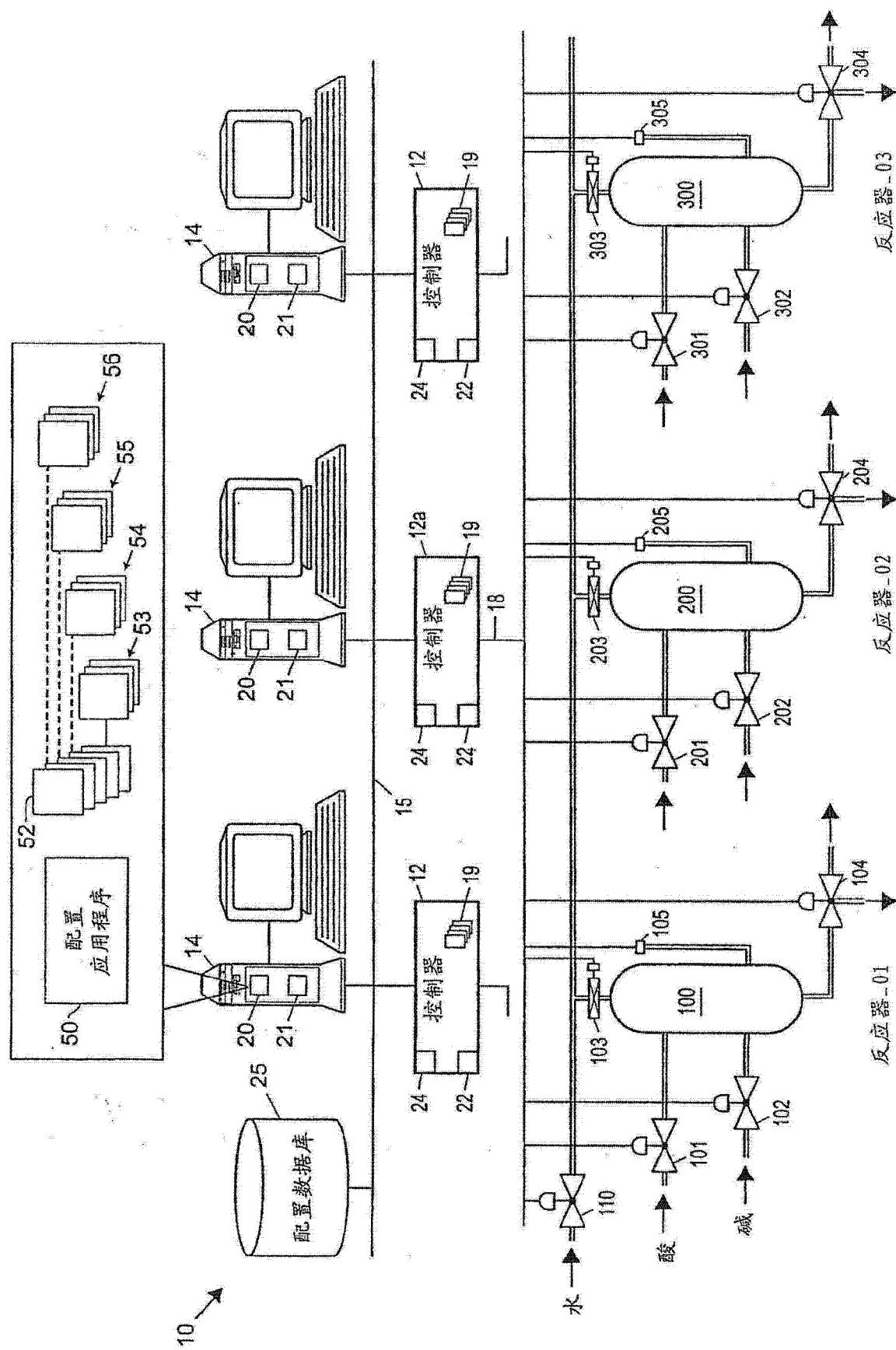


图 1

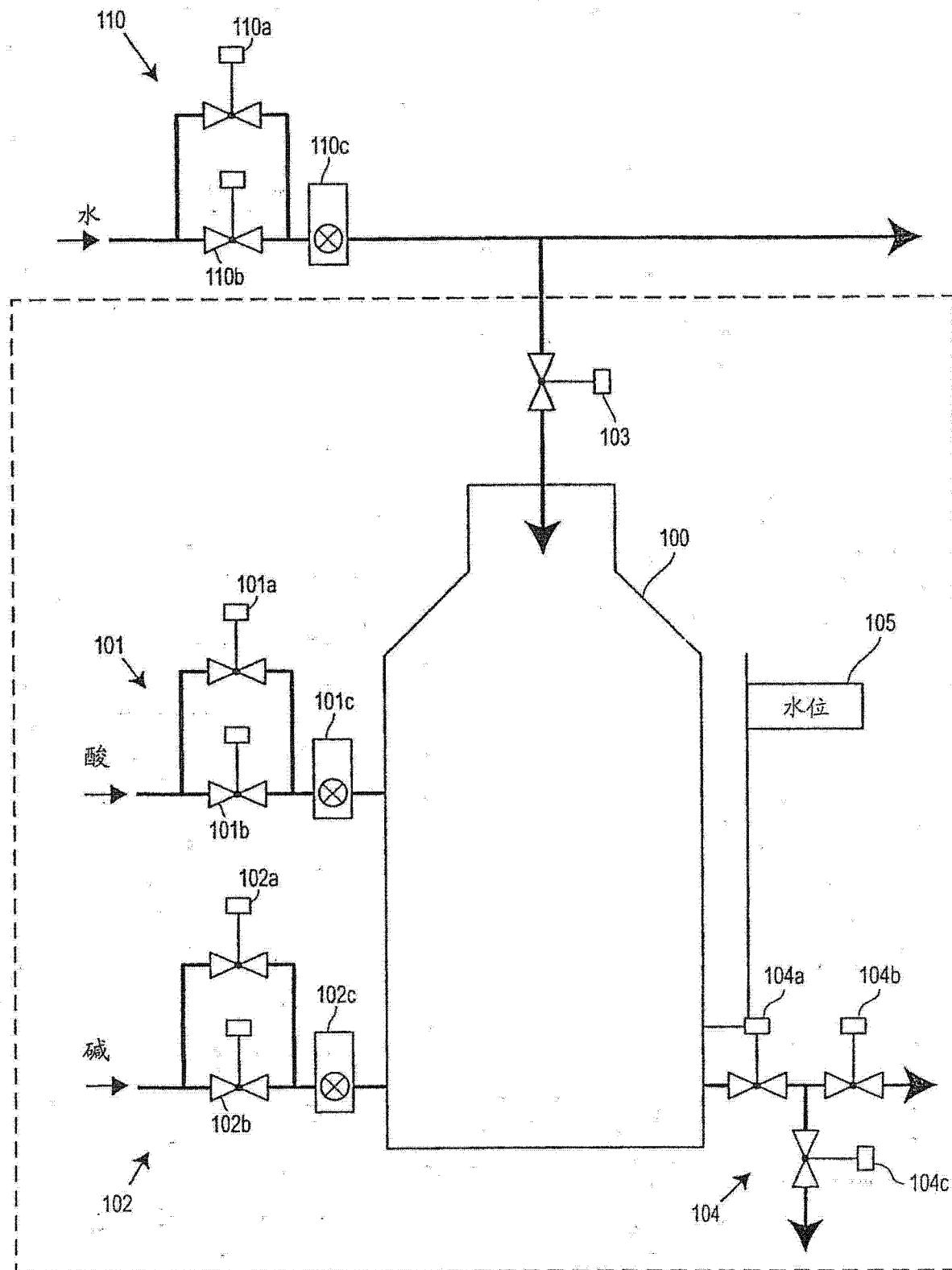


图 2

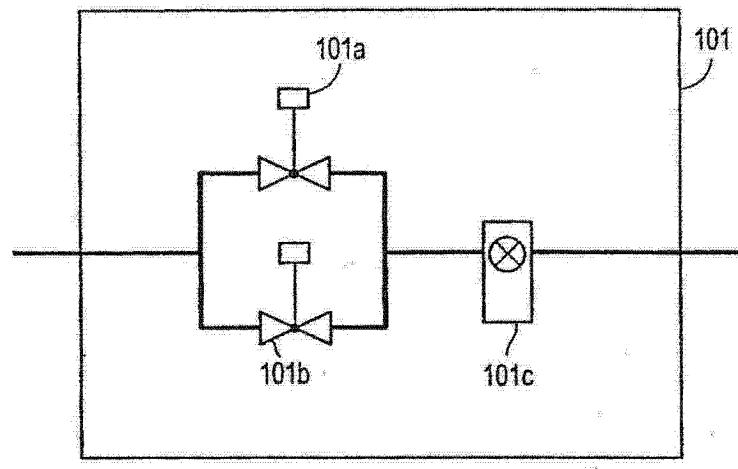


图 3

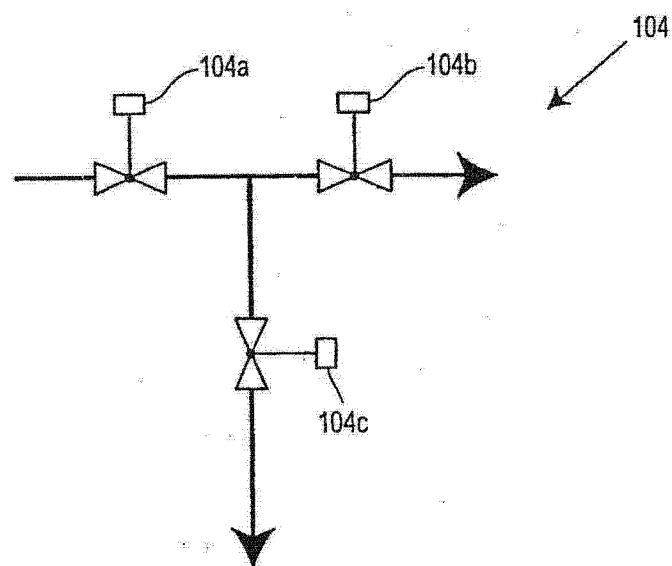


图 4

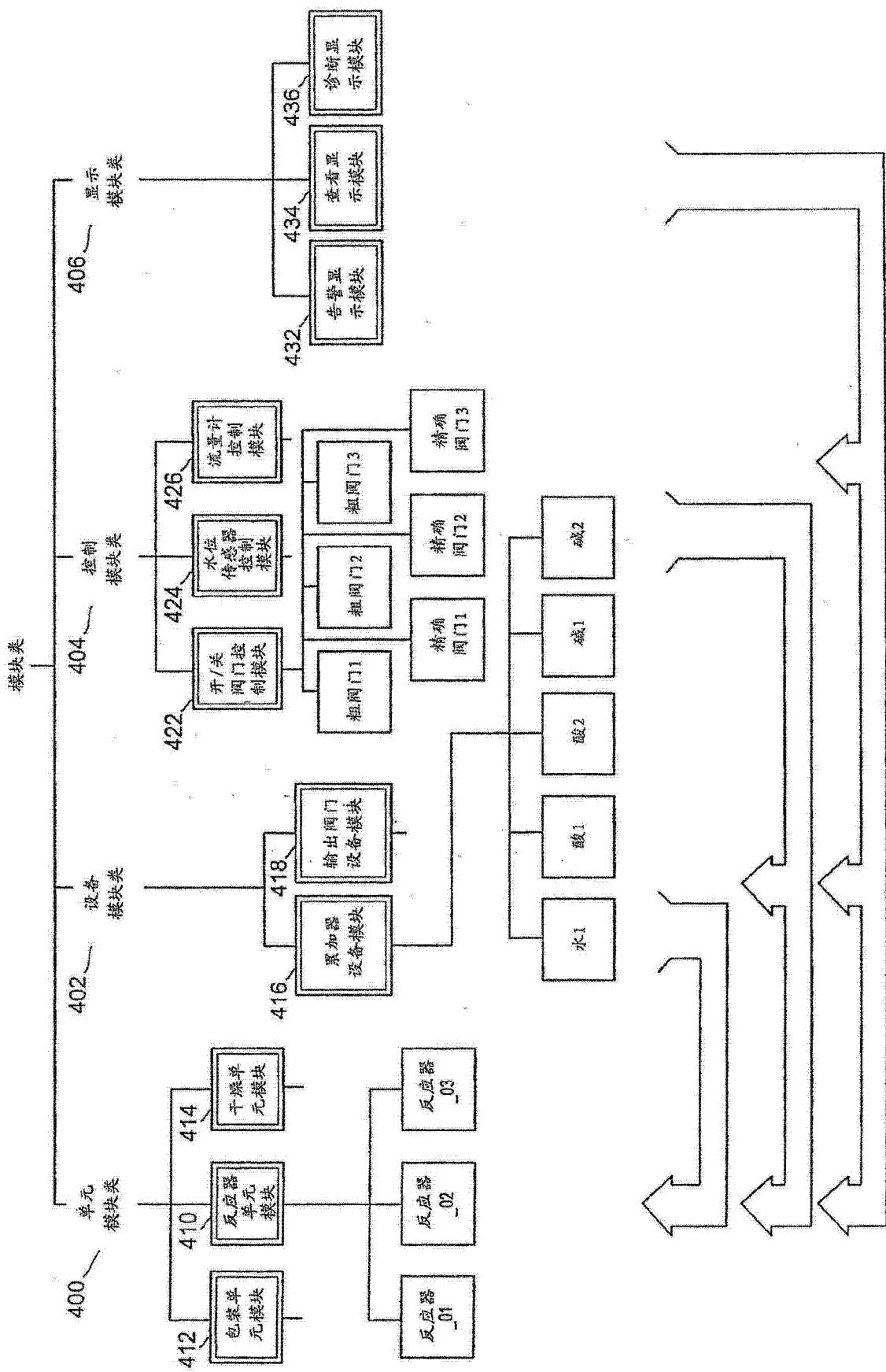


图 5

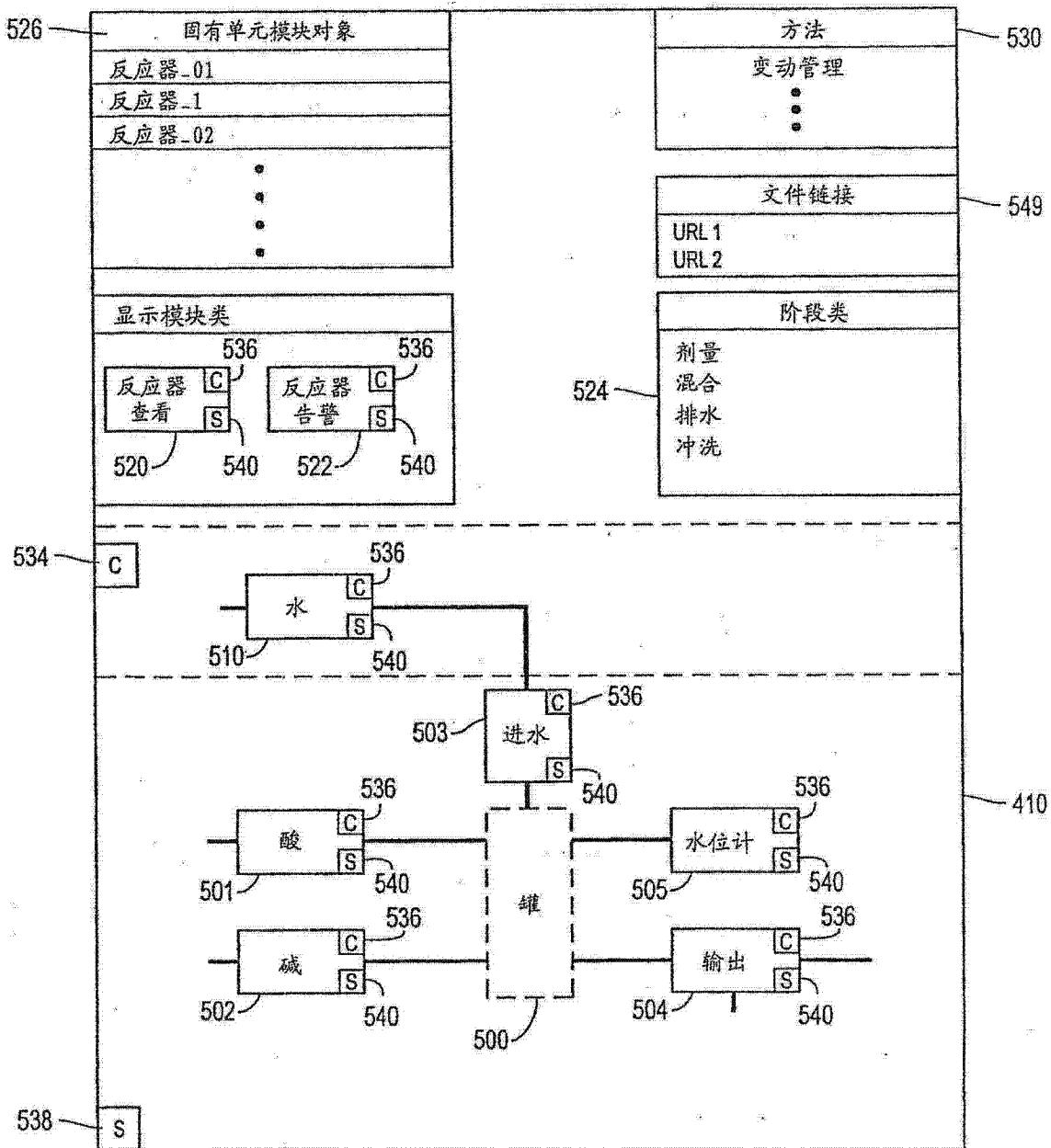


图 6

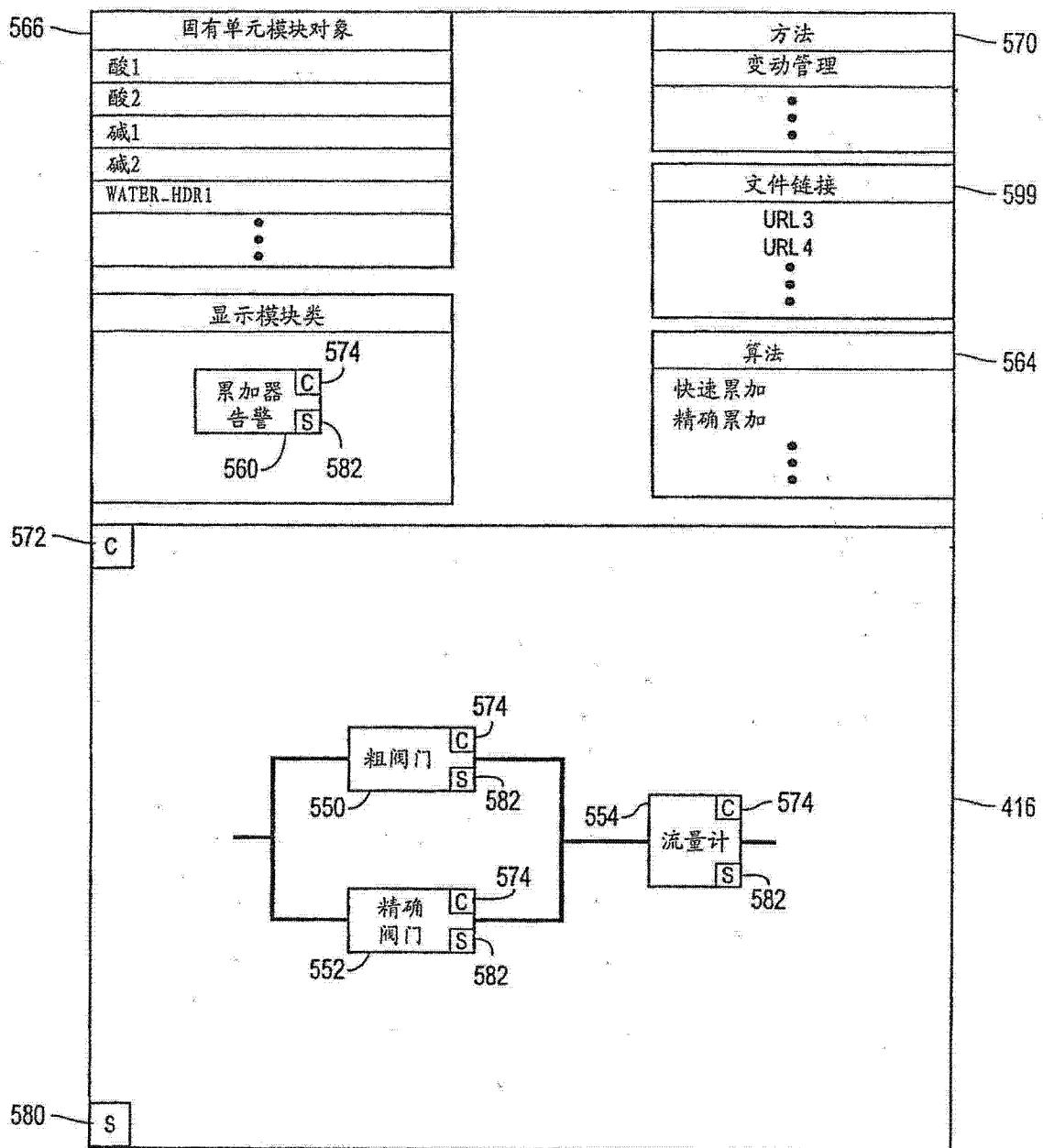
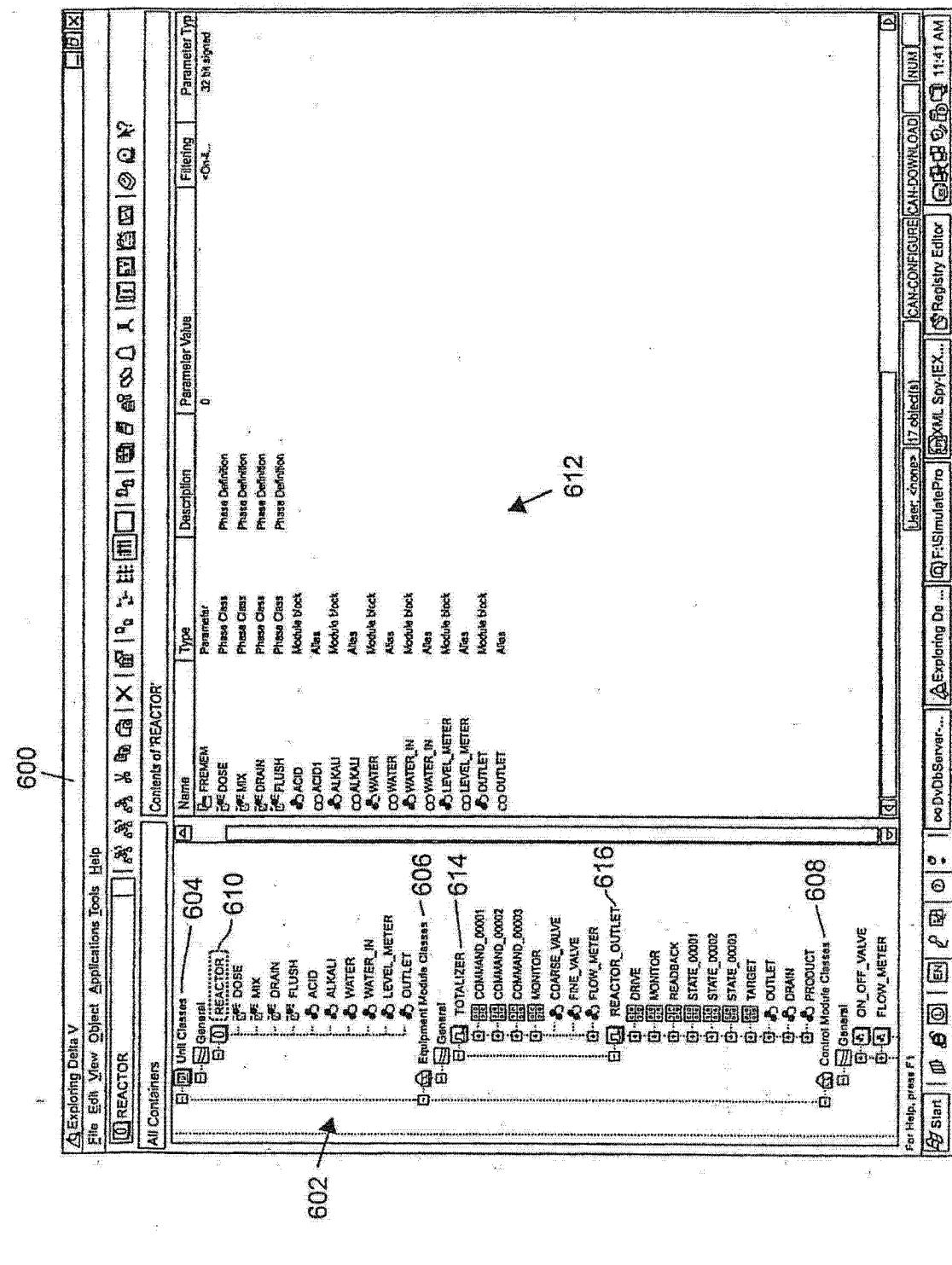


图 7



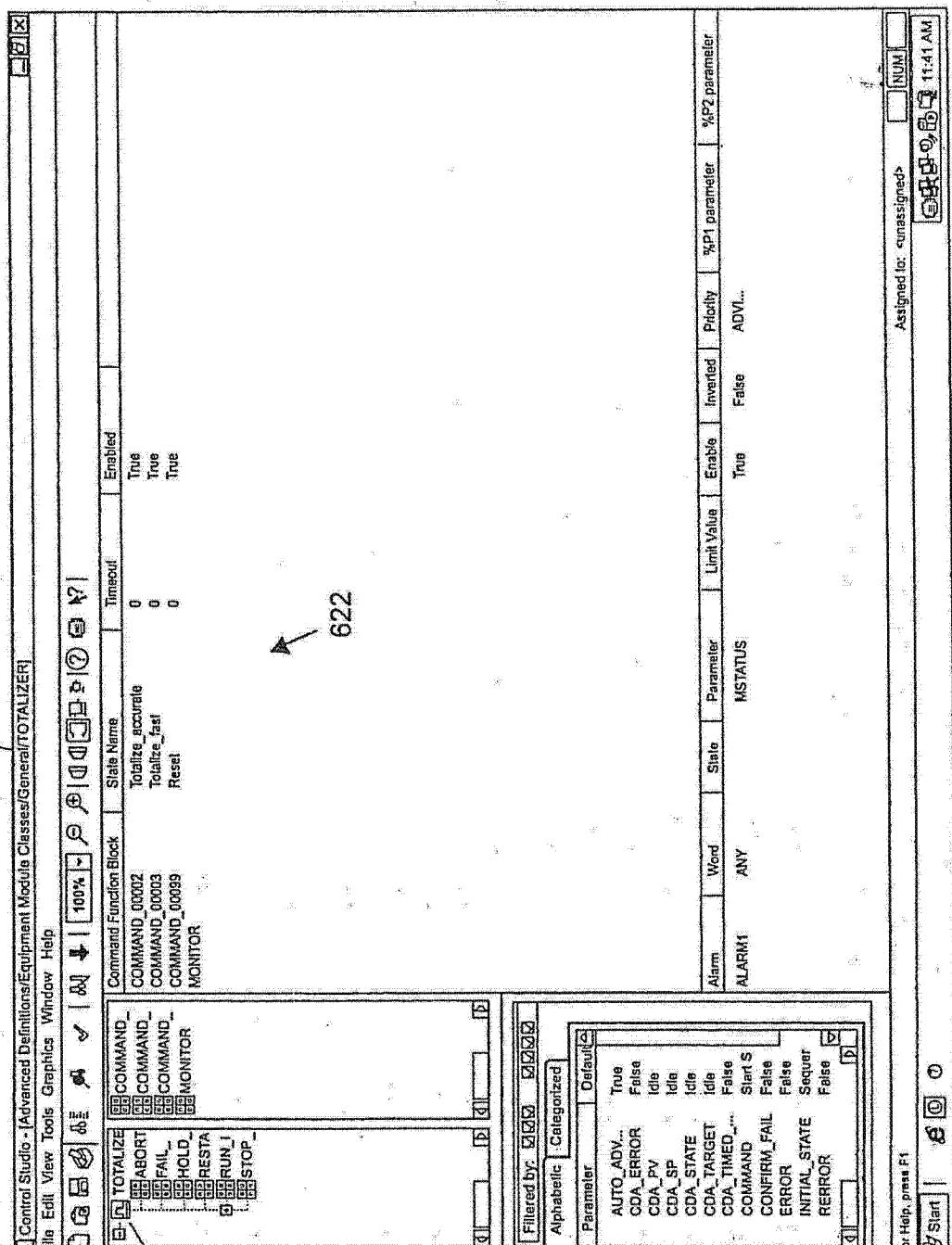


图 9

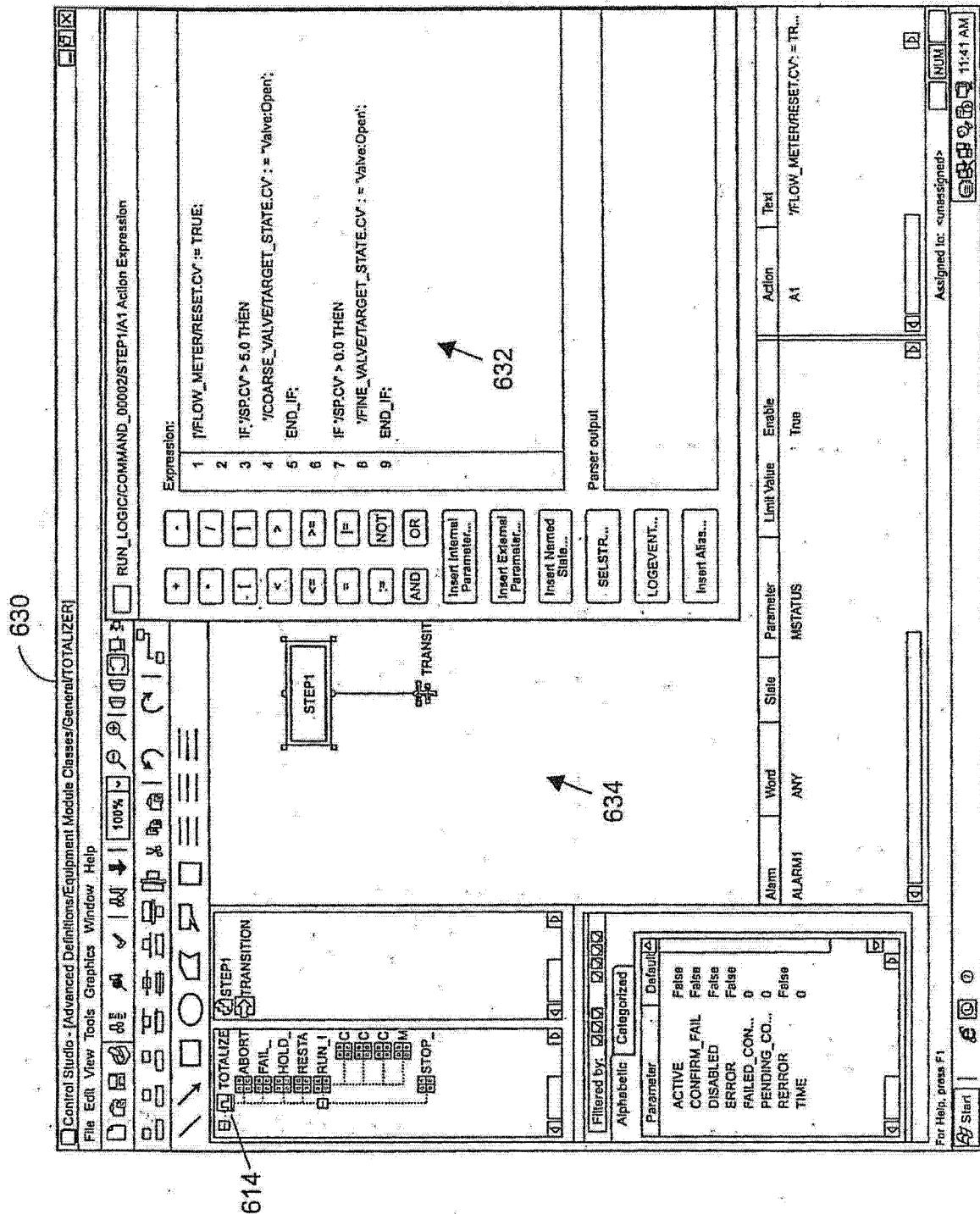


图 10

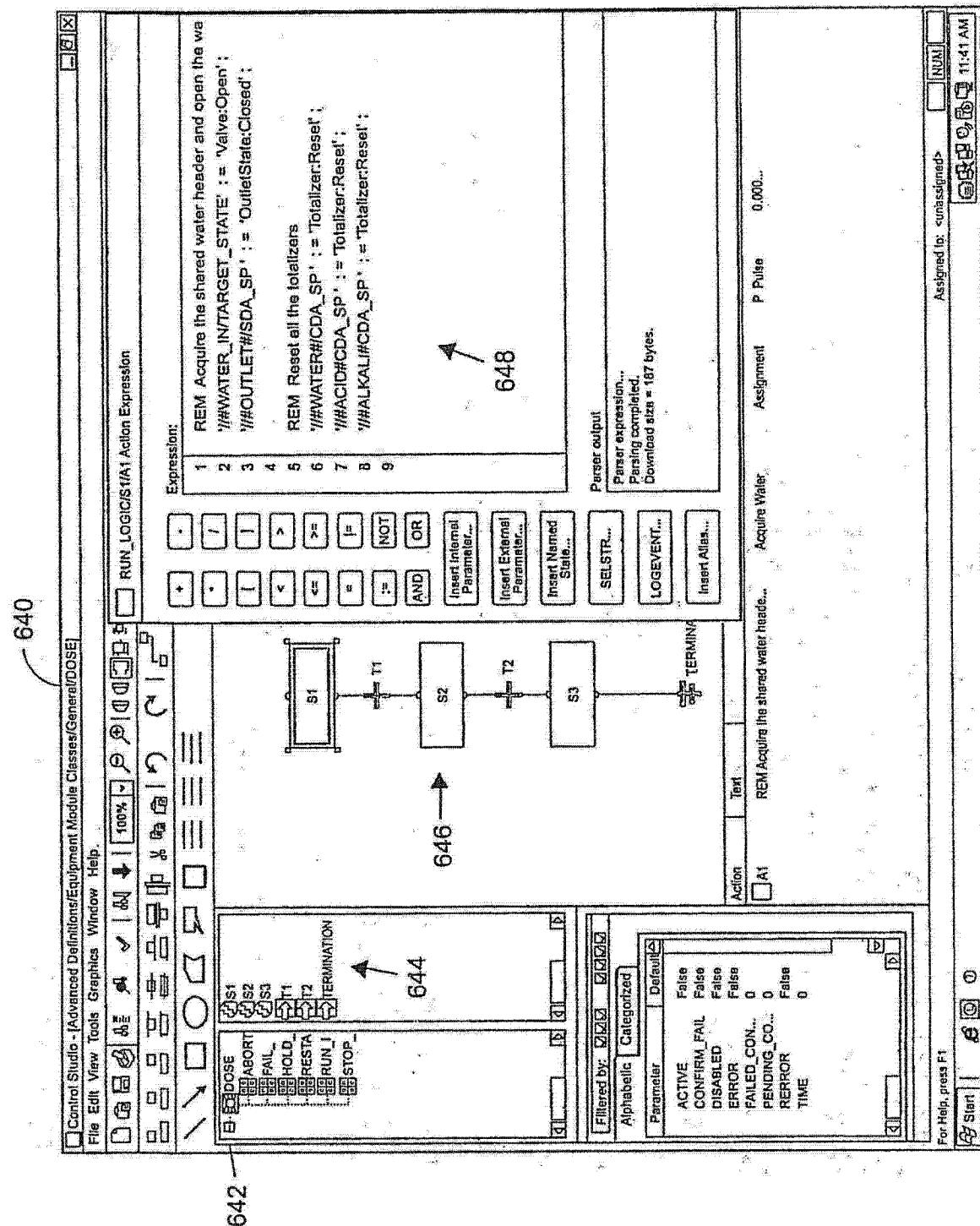


图 11

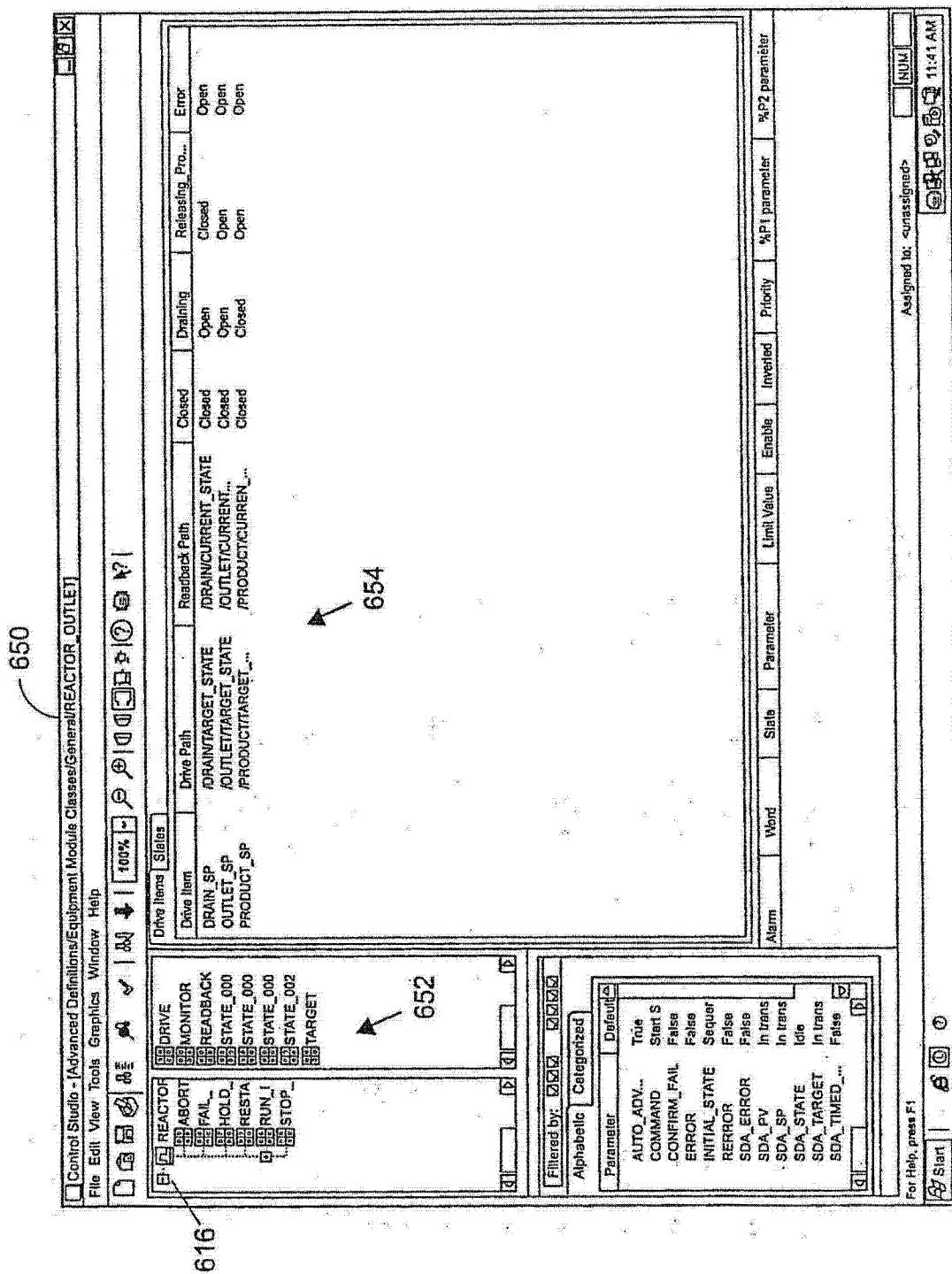


图 12

660

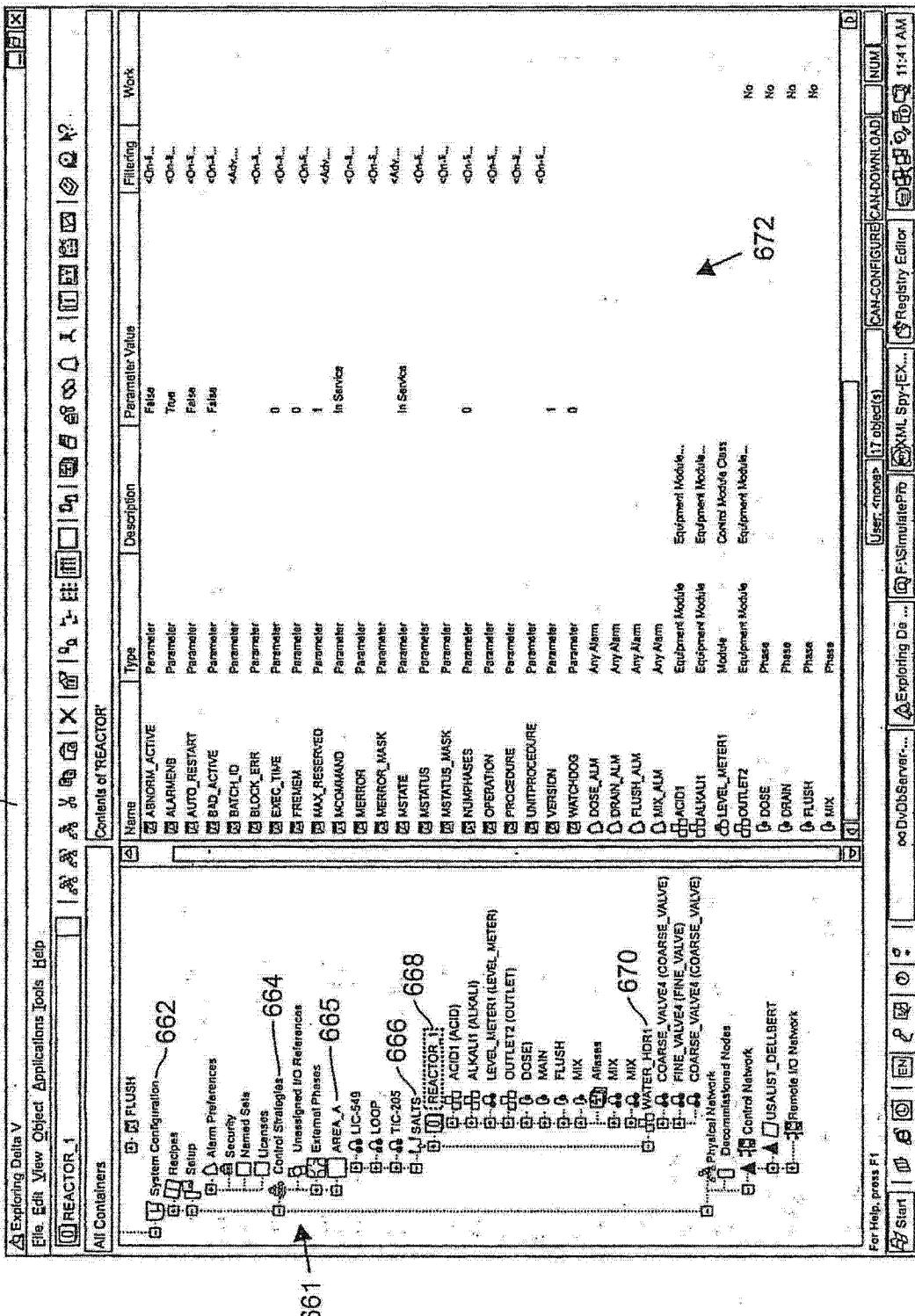
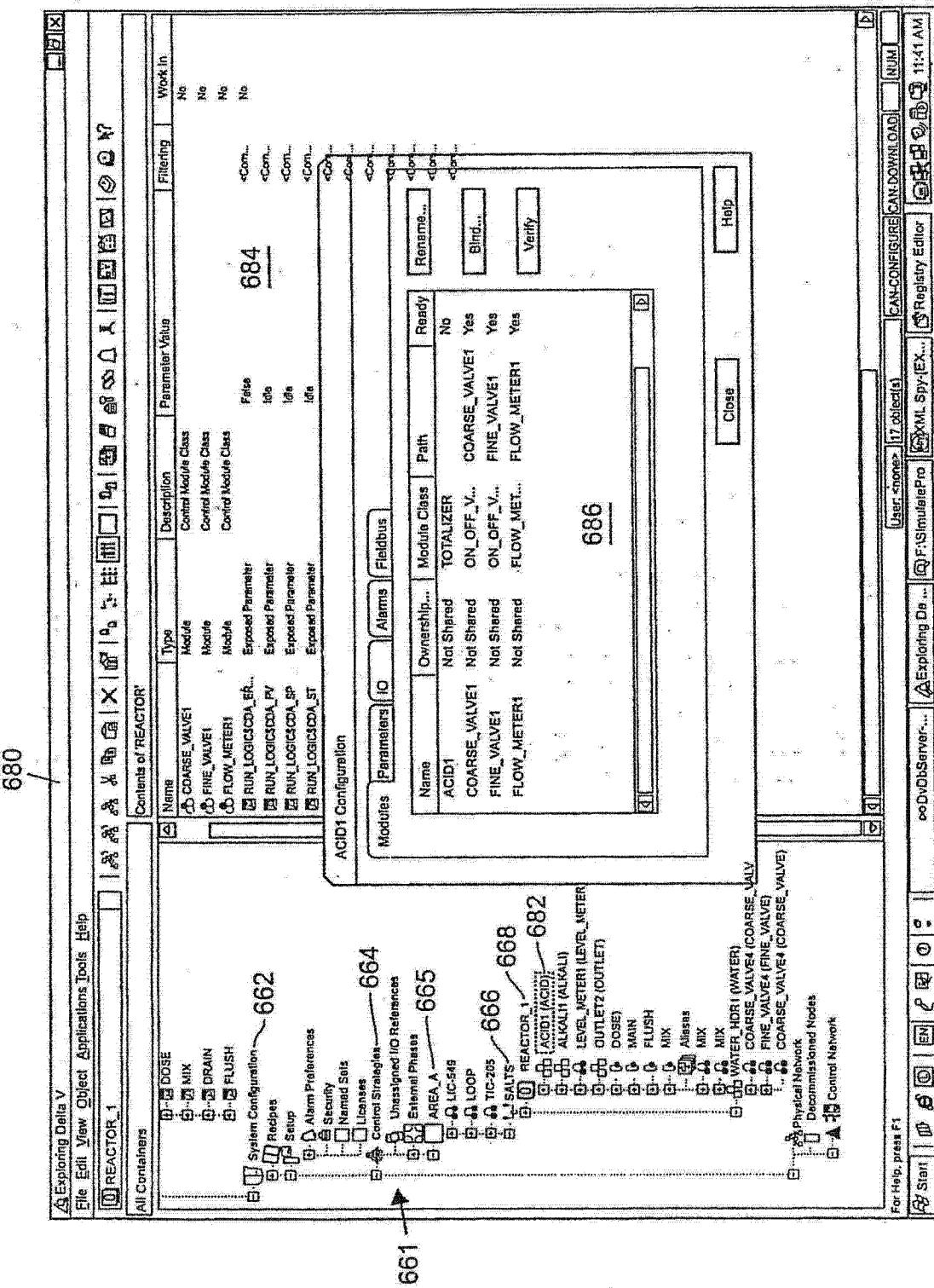


图 13



688

A	B	C	D	E
1		Modules from placeholder TOTALIZER/COARSE VALVE		
2				
3		IO_OUT	IO_READBACK	
4	VLV-101A	VLV-01/OUT D	VRB-01/FIELD VAL D	
5	VLV-201A	VLV-02/OUT D	VRB-02/FIELD VAL D	
6	VLV-301A	VLV-03/OUT D	VRB-03/FIELD VAL D	
7	VLV-401A	VLV-04/OUT D	VRB-04/FIELD VAL D	
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				

Navigation buttons: [Back] [Forward] [Home] [Search] [Help] [Exit]

图 15

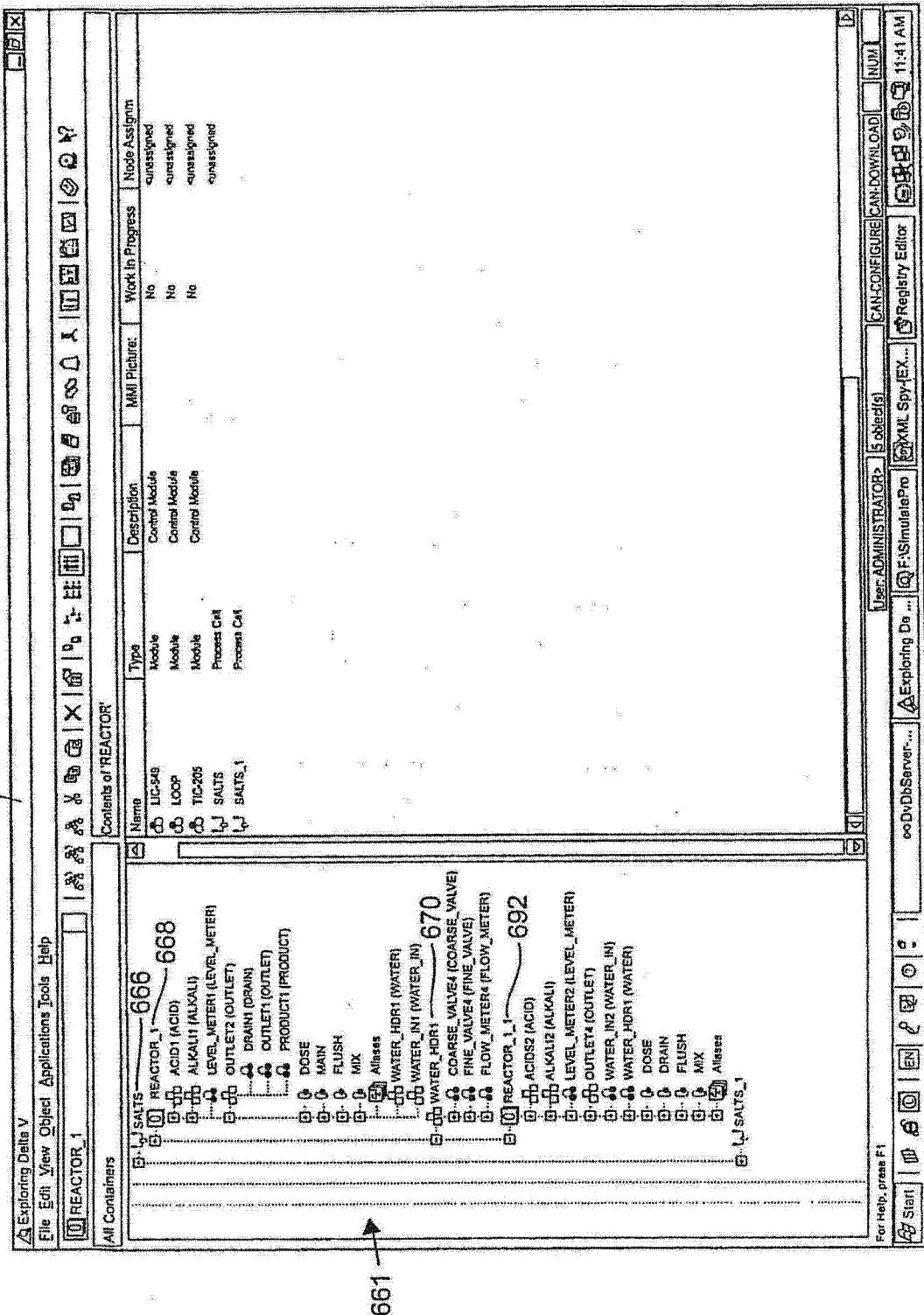


图 16

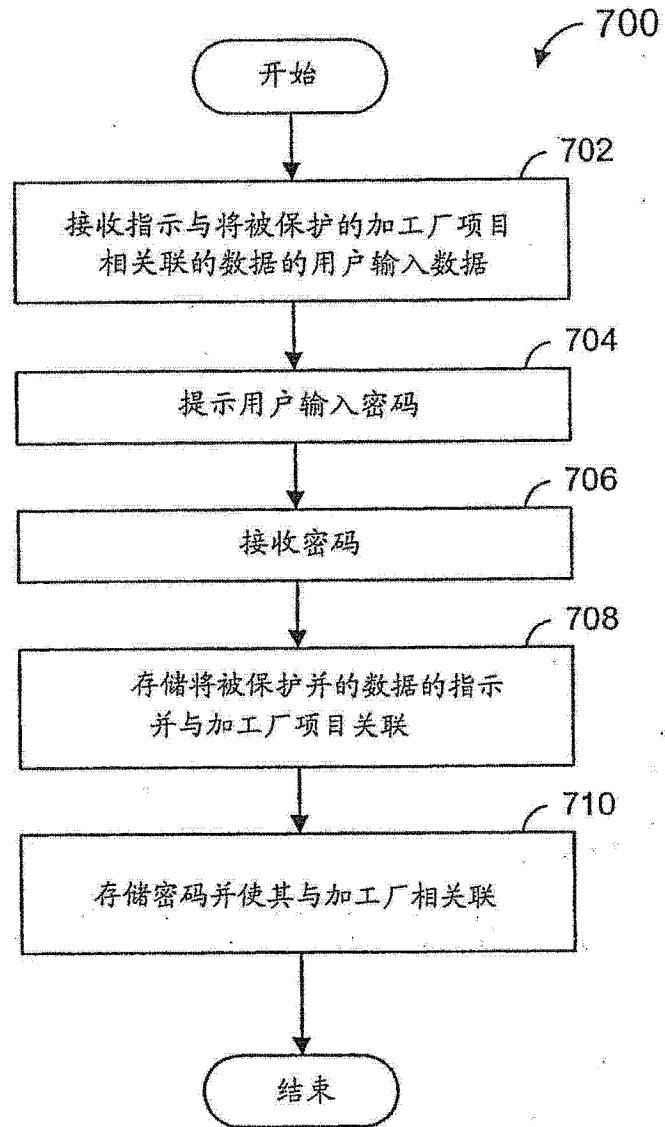


图 17

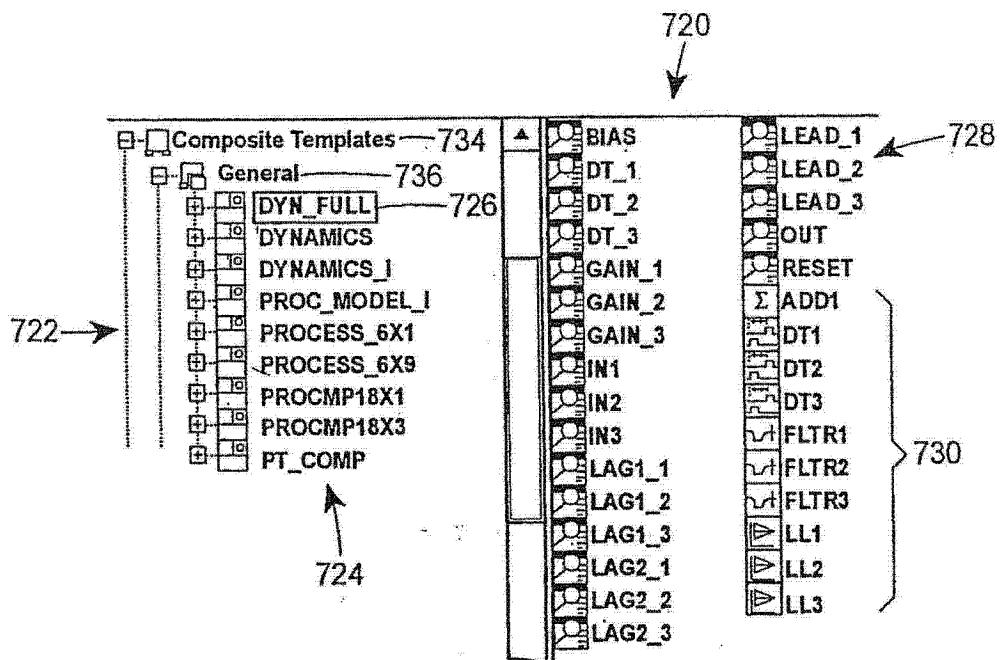


图 18A

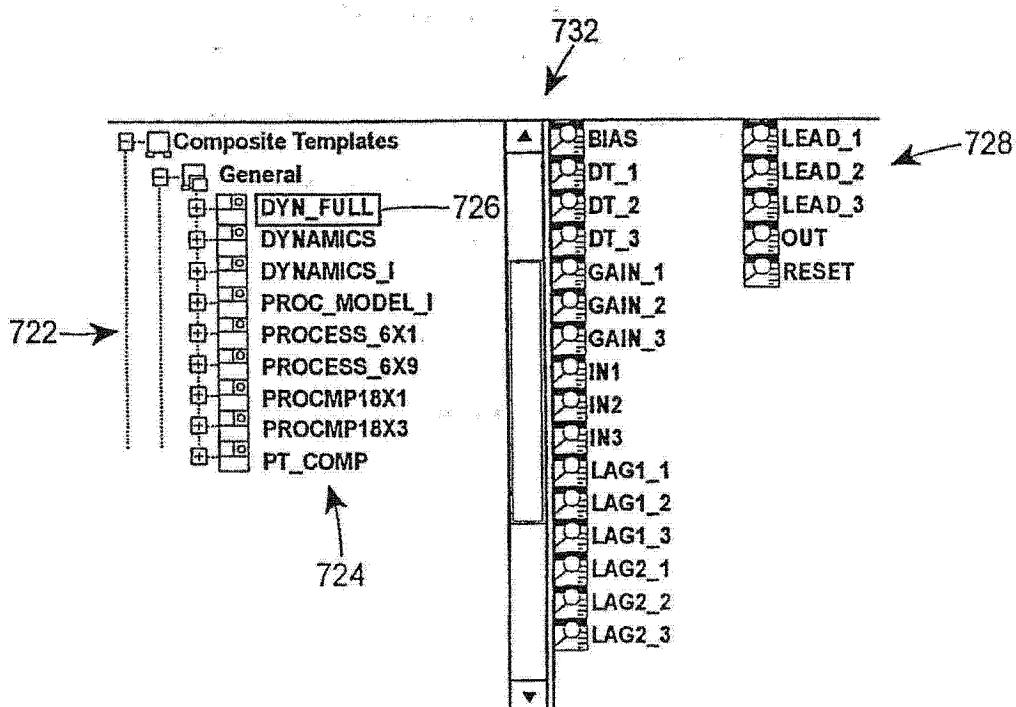


图 18B

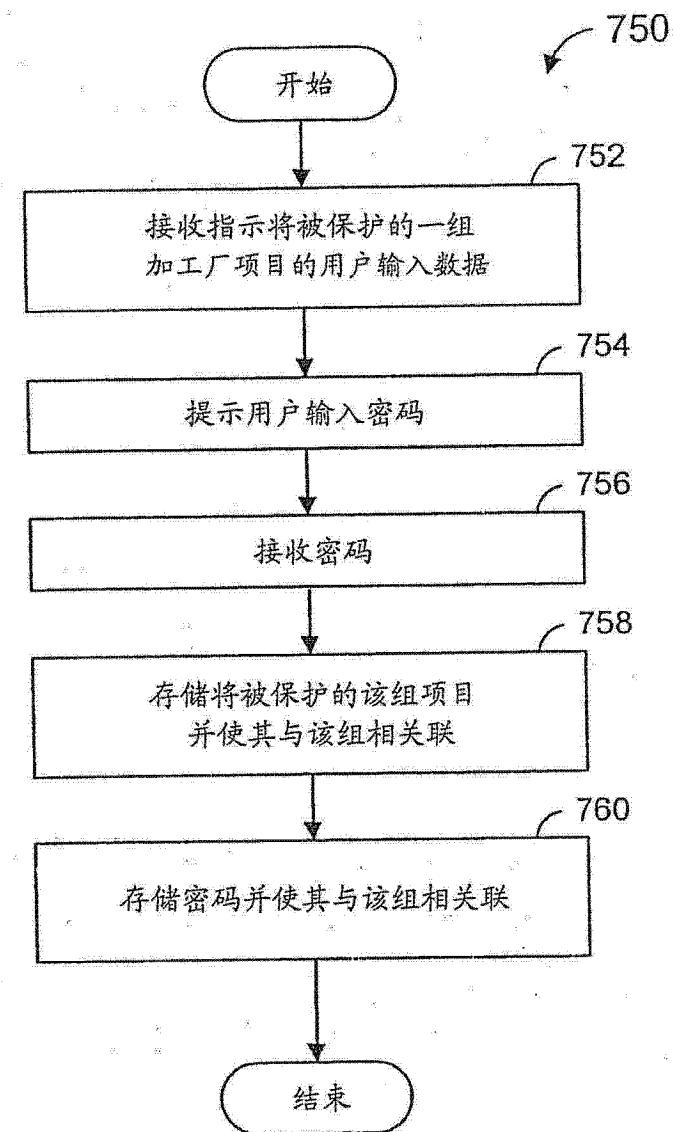


图 19

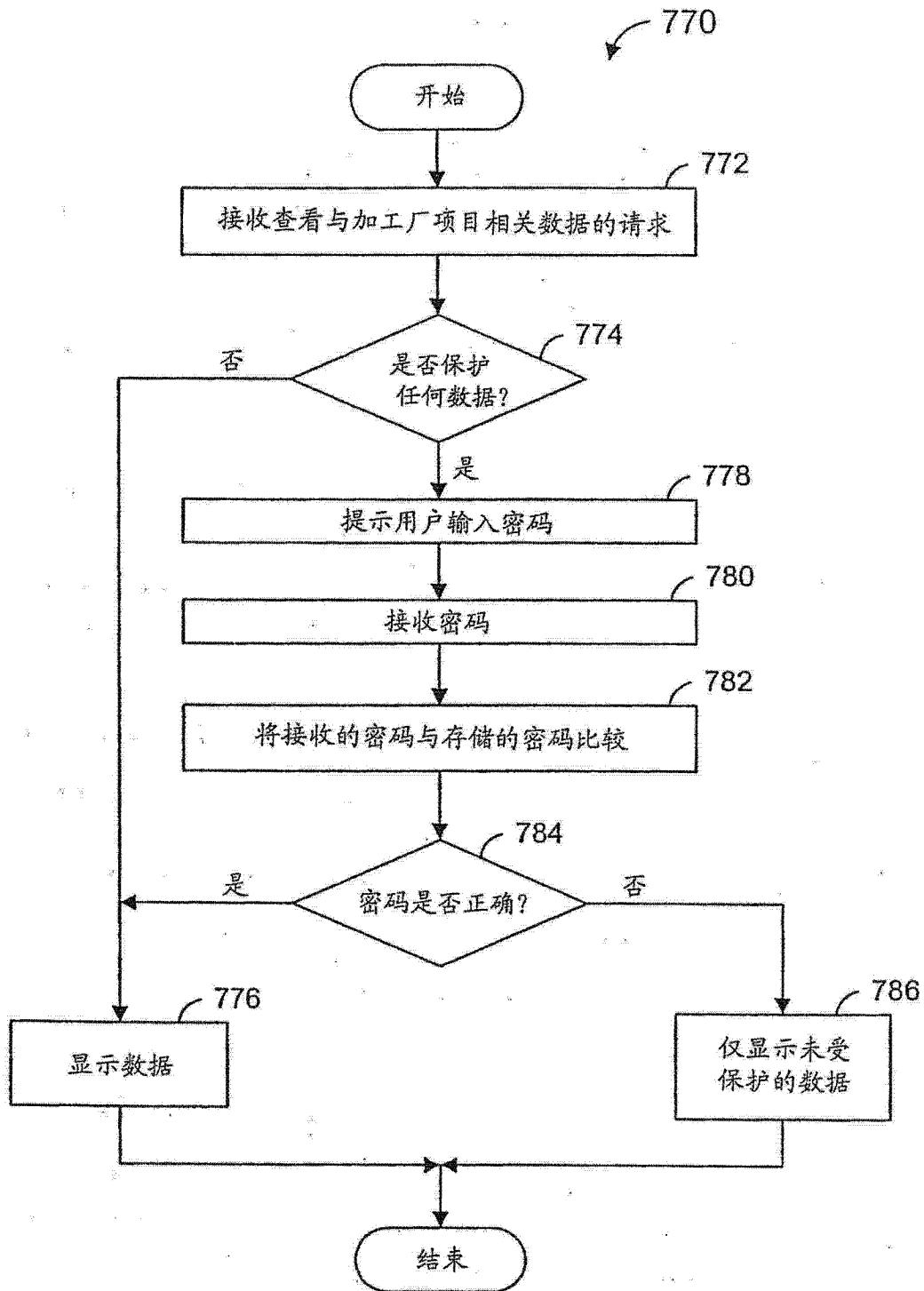


图 20

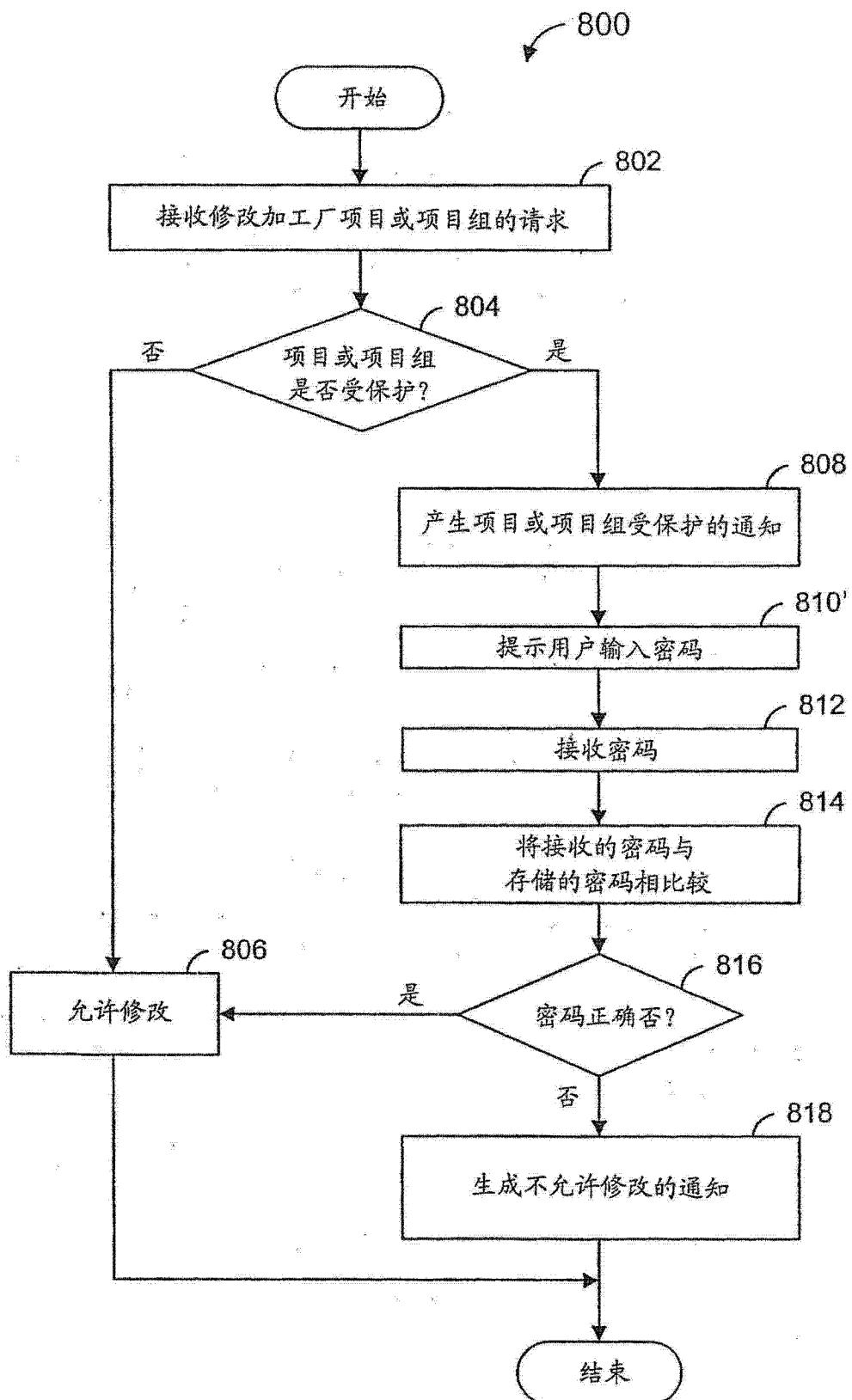


图 21

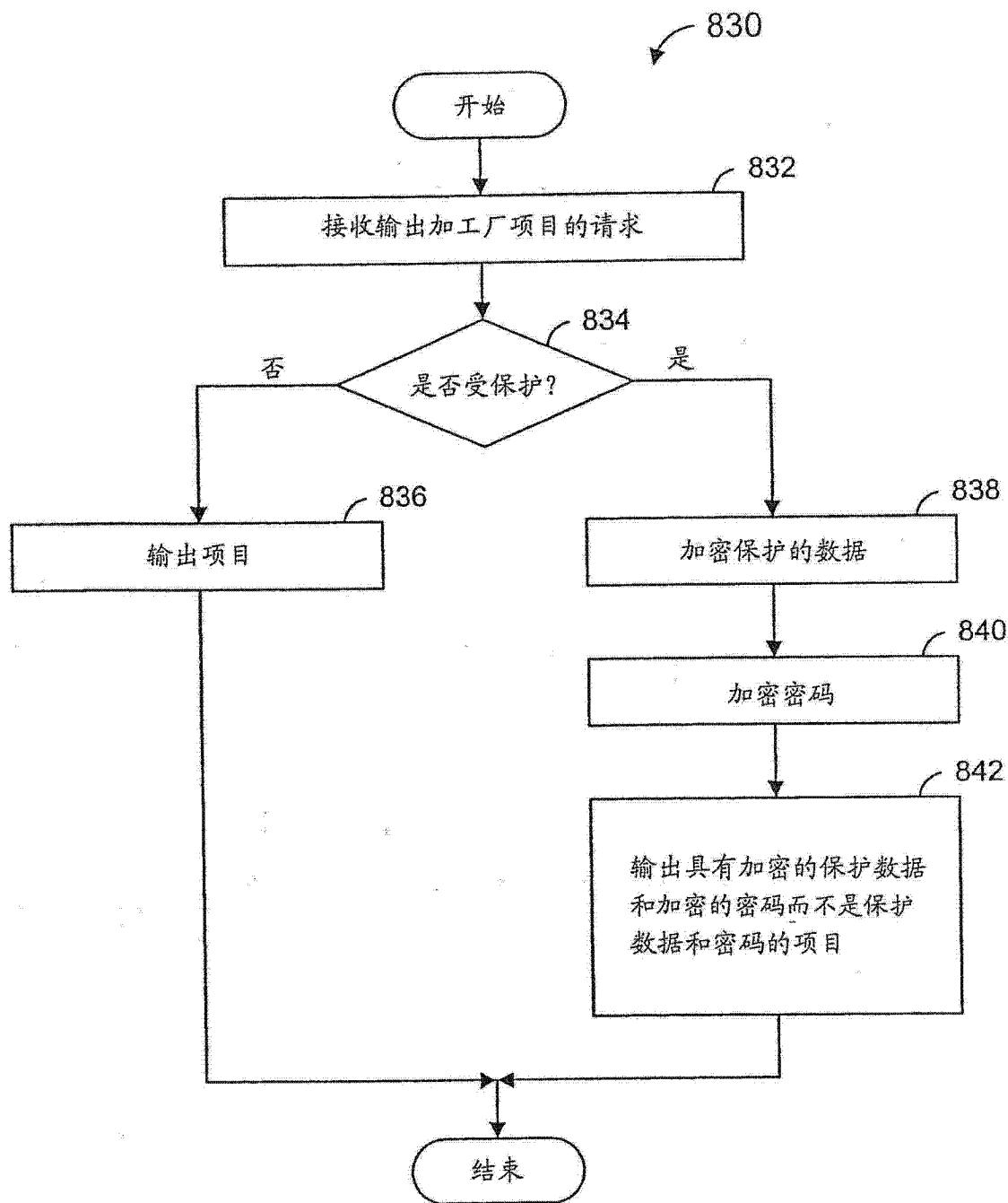


图 22

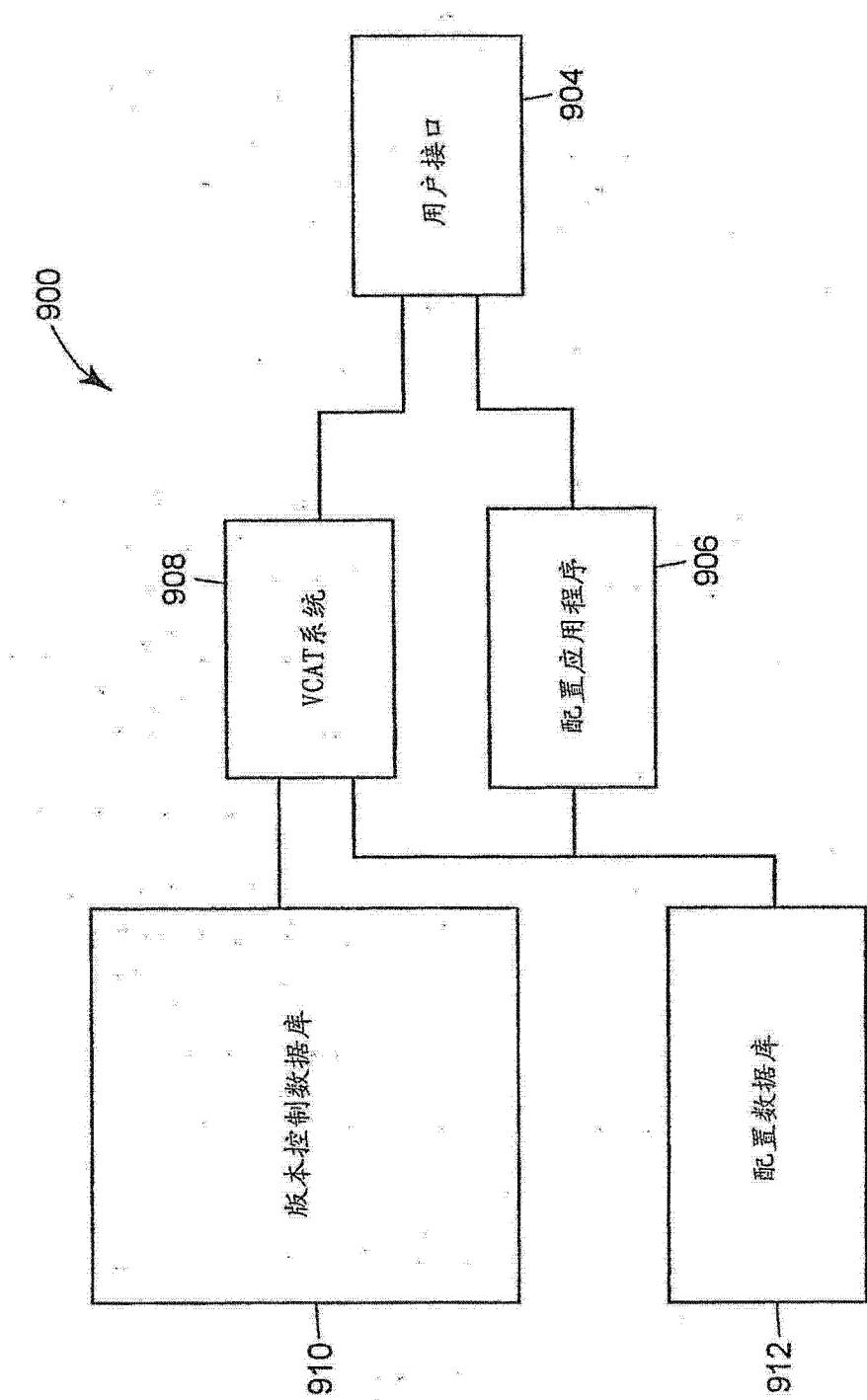


图 23

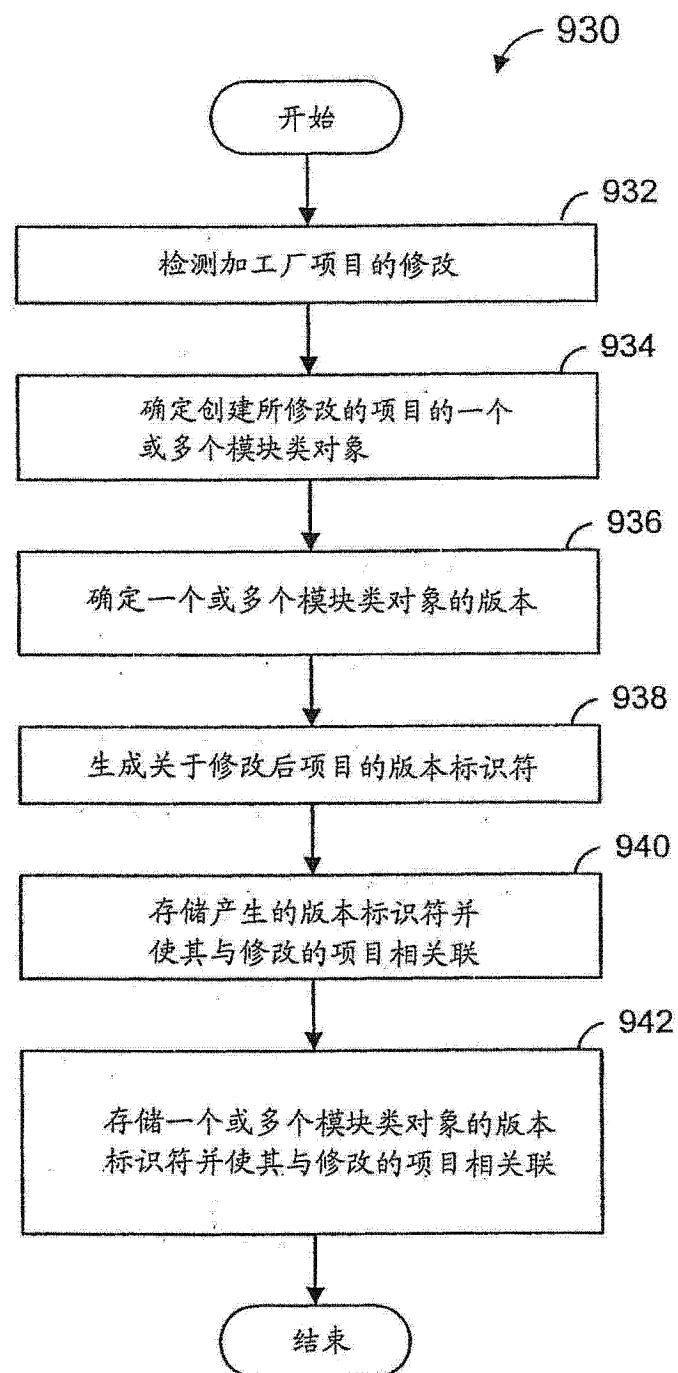


图 24

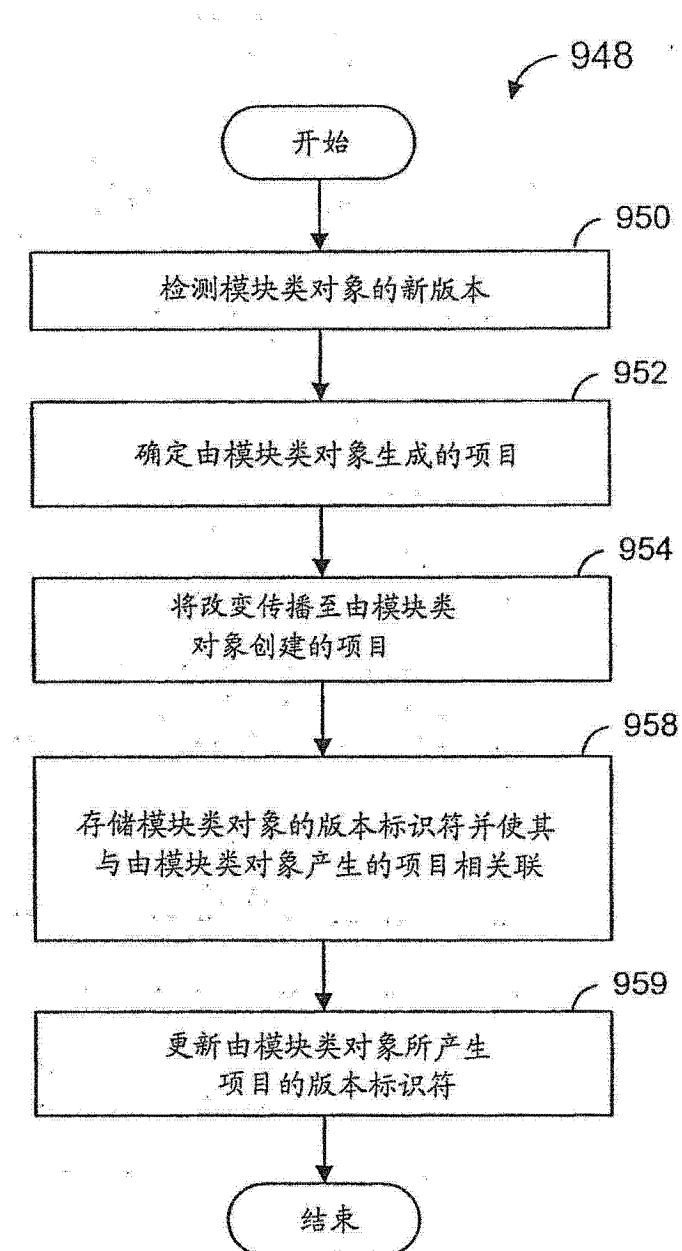


图 25

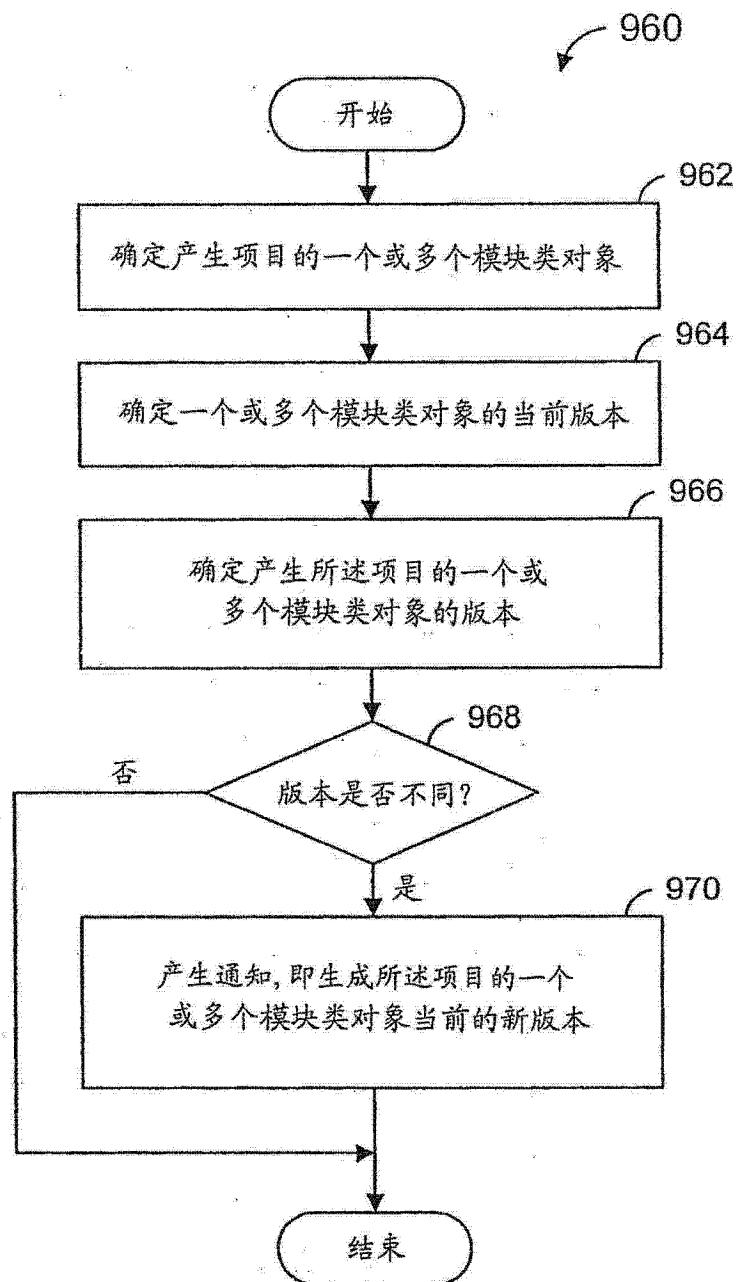


图 26

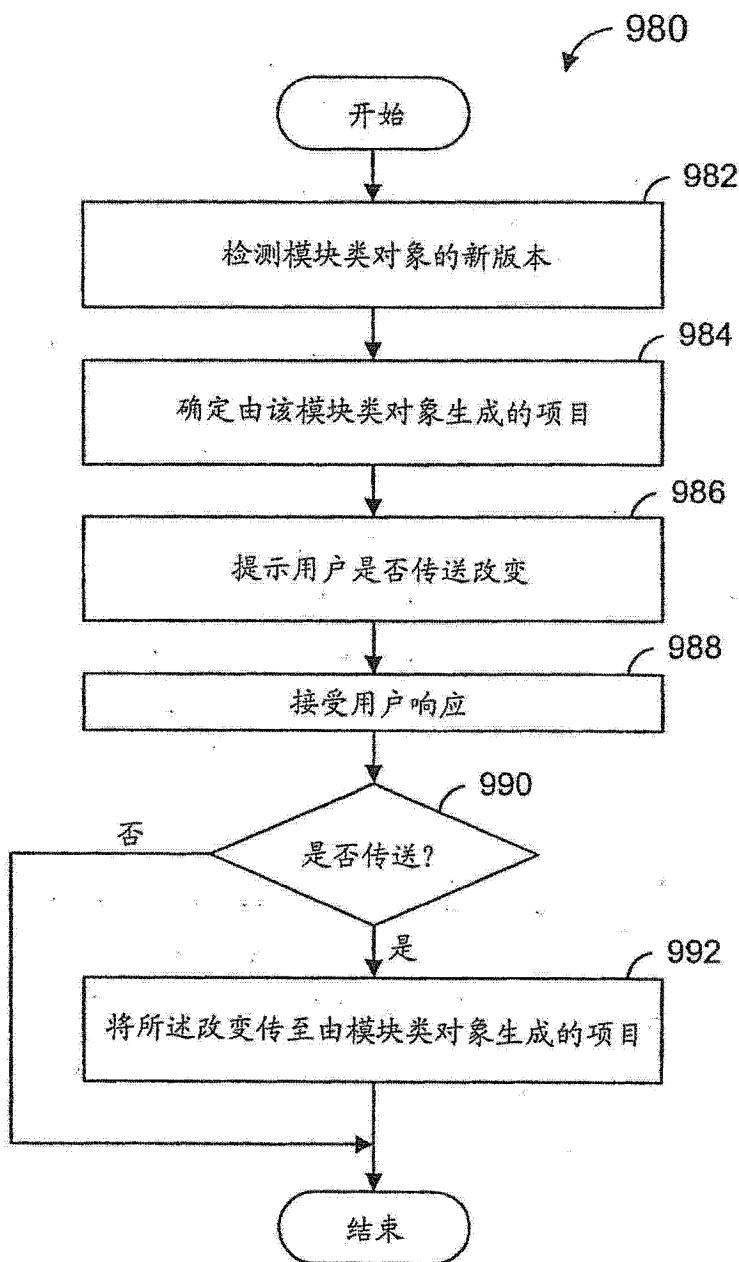


图 27

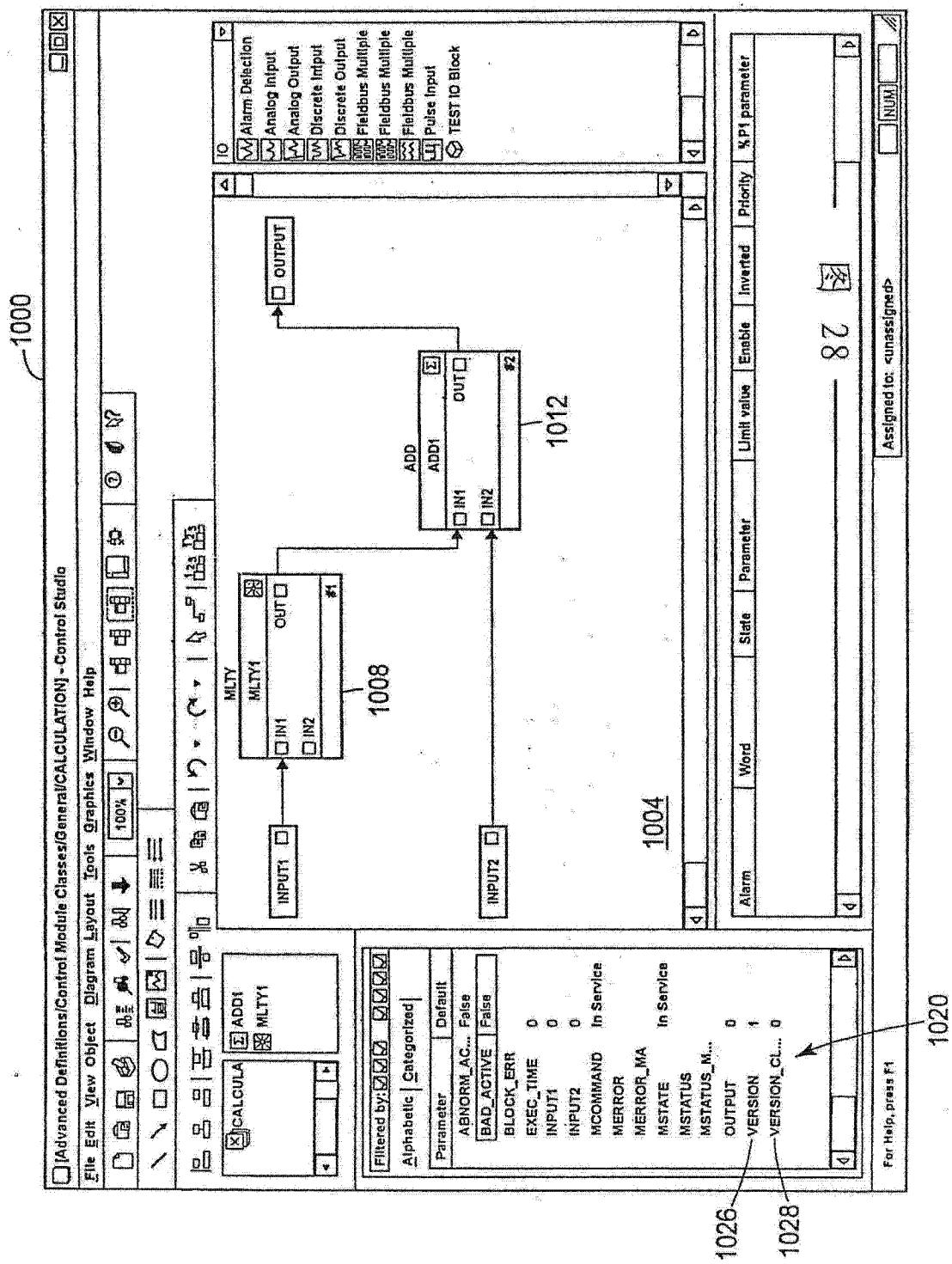


图 28

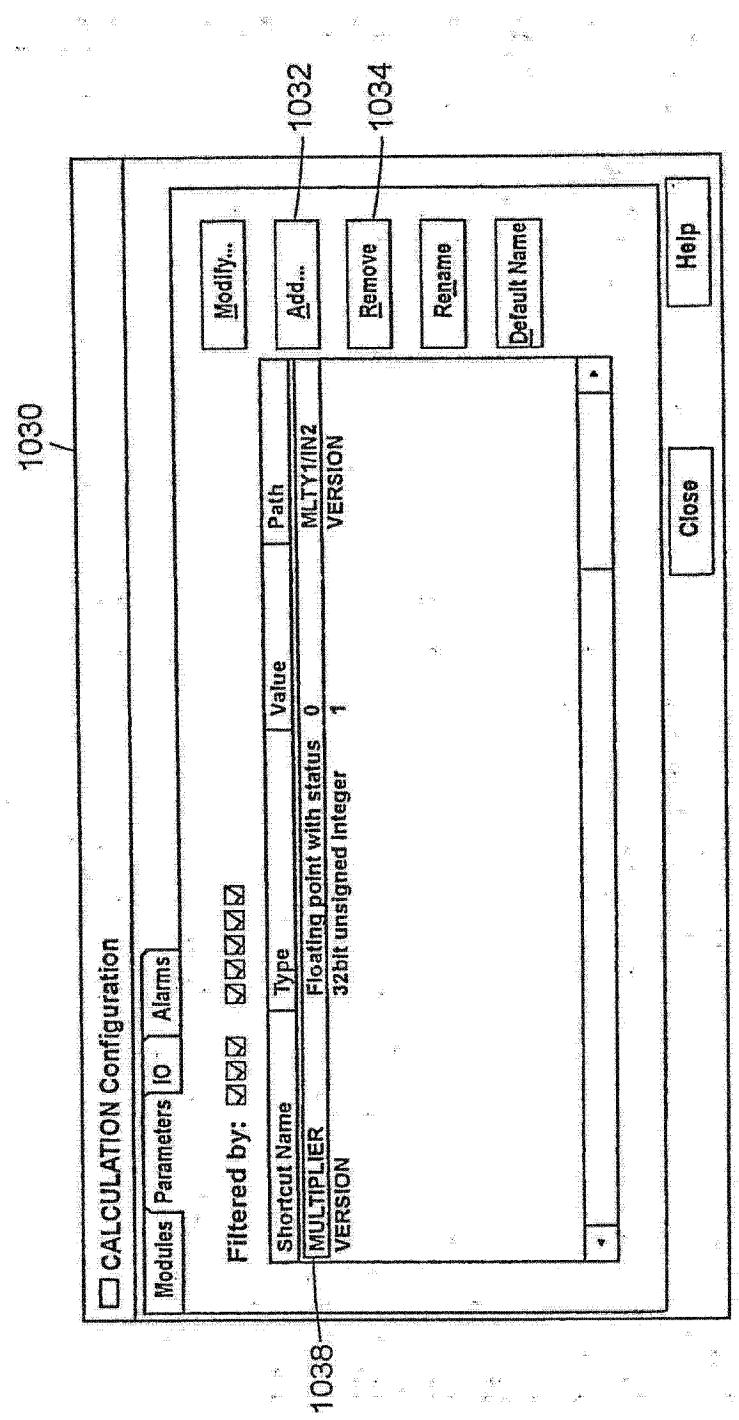


图 29

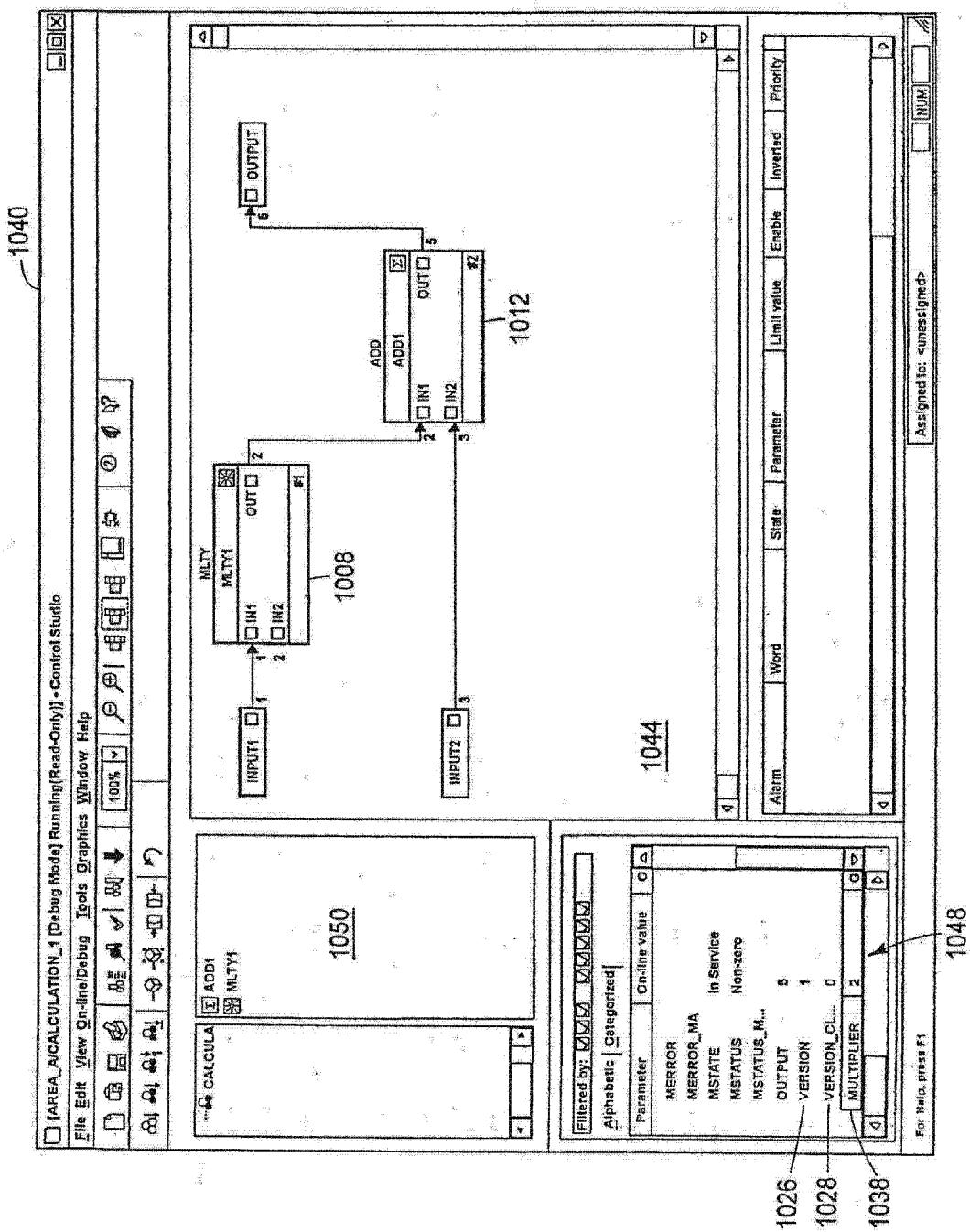


图 30

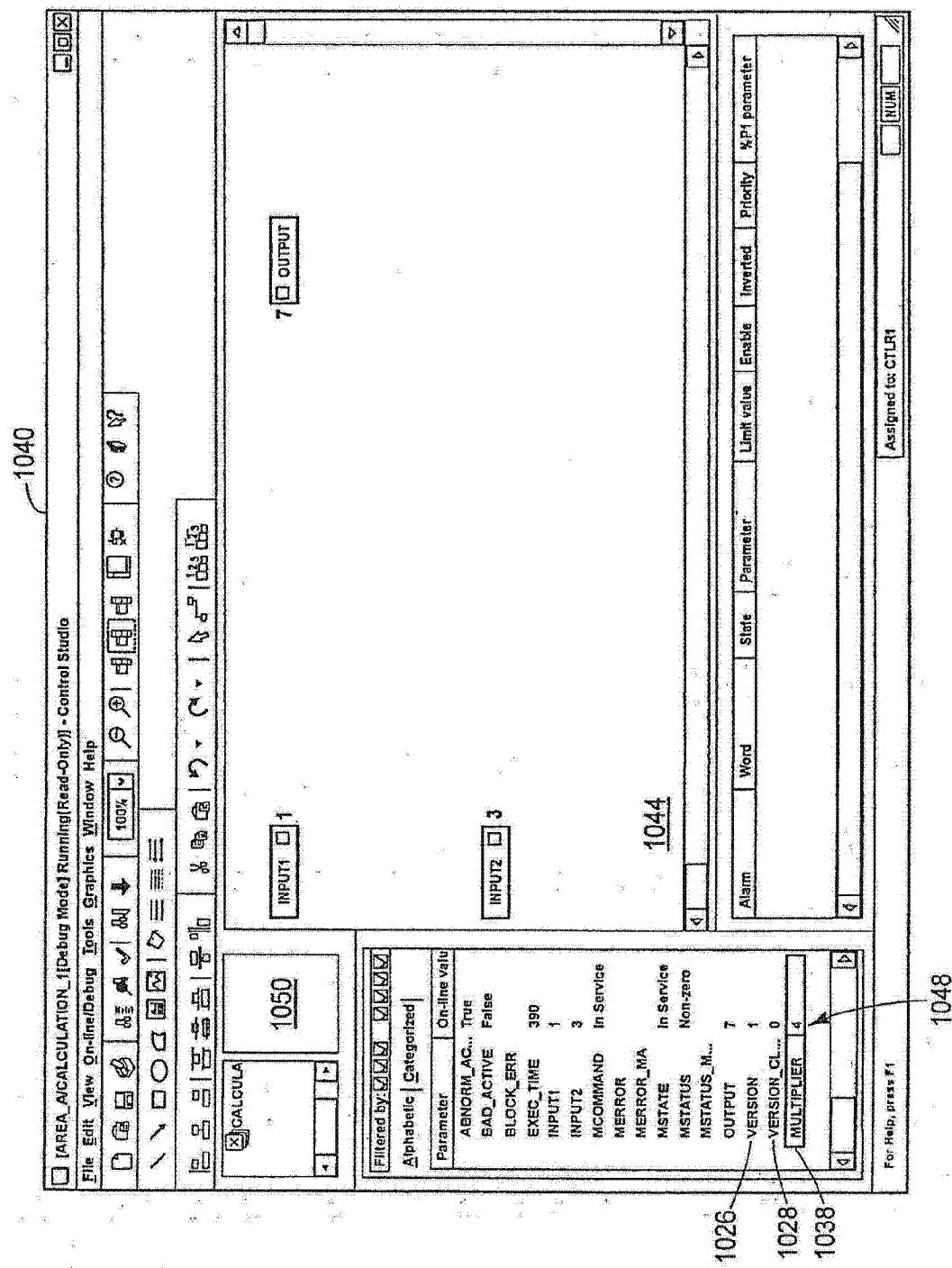


图 31