



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 10 2004 038 808 A1 2005.12.22

(12)

Offenlegungsschrift

(21) Aktenzeichen: 10 2004 038 808.3

(22) Anmeldetag: 10.08.2004

(43) Offenlegungstag: 22.12.2005

(51) Int Cl.7: G05B 19/042

(30) Unionspriorität:

10/853,668 25.05.2004 US

(71) Anmelder:

Fisher-Rosemount Systems, Inc., Austin, Tex., US

(74) Vertreter:

Meissner, Bolte & Partner GbR, 80538 München

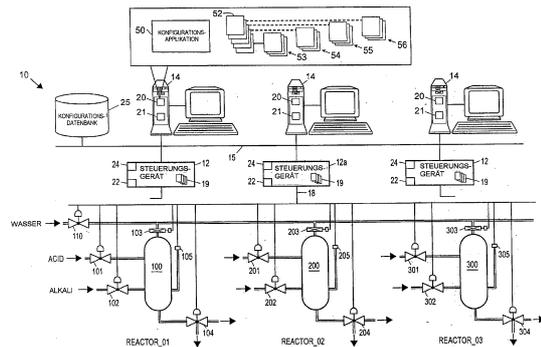
(72) Erfinder:

Lucas, Michael J., Leicestershire, GB; Webb, Arthur, Taguig, Manila, PH; Nixon, Mark J., Round Rock, Texas, US; Jundt, Larry O., Round Rock, Texas, US; Li, Jian, Austin, Texas, US; Stevenson, Dennis L., Round Rock, Texas, US; Ott, Michael G., Austin, Texas, US; Koska, Herschel O., San Marcos, Texas, US; Havekost, Robert B., Elgin, Texas, US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: Versionskontrolle für Objekte in einem Konfigurationssystem für Prozessanlagen

(57) Zusammenfassung: Ein Versionskontrollsystem unterstützt die Verwaltung von Versionen von Prozessanlagenelementen, die Entitäten in einer Prozessanlage repräsentieren oder repräsentieren können. Die Prozessanlagenelemente können beispielsweise Modulobjekte umfassen, die spezifisch Prozessentitäten der Prozessanlage repräsentieren können. Diese Modulobjekte können aus Modulklassenobjekten erzeugt werden, die generisch Prozessentitäten der Prozessanlage repräsentieren können. Versionsdaten werden gespeichert und einem Modulobjekt zugeordnet. Zu den Versionsdaten können Daten gehören, die eine Version eines Modulklassenobjekts bezeichnen, das genutzt wurde, um das Modulobjekt zu erzeugen. Die Versionsdaten können ferner Daten umfassen, die eine Version des Modulobjekts bezeichnen. Konfigurationssysteme, Versionskontrollsysteme, Anzeigesysteme, Fehlerbereinigungssysteme, Run-Time-Überwachungssysteme, Asset-Managementsysteme etc. können die einem Element zugeordneten Versionskontrolldaten prüfen oder ihre Anzeige zulassen.



Beschreibung

Querverweise auf zugehörige Anmeldungen

[0001] Die vorliegende Anmeldung ist eine Teilfortführungsanmeldung der am 18. Februar 2003 eingereichten US-Patentanmeldung Nr. 10/368,151 mit dem Titel "MODULE CLASS OBJECTS IN A PROCESS PLANT CONFIGURATION SYSTEM", die hiermit ausdrücklich durch Bezugnahme für alle Zwecke vollständig in diese Anmeldung einbezogen wird.

[0002] Die vorliegende Anmeldung steht im Zusammenhang mit der gleichzeitig anhängigen US-Patentanmeldung Nr. (Anwaltsregister Nr. 06005/40122) mit dem Titel "SECURITY FOR OBJECTS IN A PROCESS PLANT CONFIGURATION SYSTEM", die am gleichen Tag wie die vorliegende Anmeldung eingereicht wurde und hiermit ausdrücklich durch Bezugnahme für alle Zwecke vollständig in diese Anmeldung einbezogen wird.

Stand der Technik

Gebiet der Erfindung

[0003] Die vorliegende Erfindung bezieht sich allgemein auf Prozessanlagen und im Besonderen auf die Versionskontrolle im Zusammenhang mit der Konfigurierung einer Prozessanlage.

Hintergrund

[0004] Verteilte Prozesssteuerungssysteme, wie sie in der chemischen Industrie, der Petrochemie und anderen Prozessen eingesetzt werden, umfassen typischerweise ein oder mehrere, über analoge, digitale oder kombinierte analog/digitale Busse kommunikativ mit einer oder mehreren Feldvorrichtungen verbundene Prozesssteuerungsgeräte. Die Feldvorrichtungen, bei denen es sich beispielsweise um Ventile, Ventilsteller, Schalter und Messwertgeber (z.B. Temperatur-, Druck-, Pegel- und Durchflusssensoren) handeln kann, sind in der Prozessumgebung angeordnet und führen Prozessfunktionen aus, wie Öffnen und Schließen von Ventilen, Messen von Prozessparametern etc. Intelligente Feldvorrichtungen, wie die Feldvorrichtungen, die dem bekannten Feldbusprotokoll entsprechen, können auch Steuerungsberechnungen, Alarmfunktionen und sonstige Steuerungsfunktionen ausführen, die gewöhnlich im Steuerungsgerät implementiert werden. Die Prozesssteuerungsgeräte, die ebenfalls typischerweise in der Anlagenumgebung angeordnet sind, empfangen Signale entsprechend den Prozessmessungen, die von den Feldvorrichtungen vorgenommen wurden, und/oder andere, den Feldvorrichtungen zugehörige Informationen und führen eine Steuerungsapplikation aus, die beispielsweise verschiedene Steuerungsmodule betreibt, die Entscheidungen für die Prozesssteuerung

treffen, Steuersignale auf Basis der empfangenen Information erzeugen und mit den Steuerungsmodulen oder -blöcken koordinieren, die in den Feldvorrichtungen, z.B. den HART- und Feldbusfeldvorrichtungen, ausgeführt werden. Die Steuerungsmodule im Steuerungsgerät senden die Steuersignale über die Kommunikationsleitungen an die Feldvorrichtungen, um so den Betrieb der Prozessanlage zu steuern.

[0005] Information von den Feldvorrichtungen und dem Steuerungsgerät wird gewöhnlich über einen Datenhighway einer oder mehreren anderen Hardware-Vorrichtungen zur Verfügung gestellt, wie Bediener-Workstations, Personal Computern, der Datenhistorie, Reportgeneratoren, zentralisierten Datenbanken etc., die typischerweise in Steuerungs-zentralen oder anderen Standorten außerhalb der rauen Anlagenumgebung untergebracht sind. Diese Hardware-Vorrichtungen führen Applikationen aus, die beispielsweise einem Bediener die Ausführung prozessbezogener Funktionen ermöglichen, wie die Änderung von Einstellungen der Prozesssteuerungsroutine, die Änderung des Betriebs der Steuerungsmodul innerhalb der Steuerungsgeräte oder Feldvorrichtungen, die Anzeige des gegenwärtigen Prozesszustands, die Anzeige von durch Feldvorrichtungen und Steuerungsgeräte erzeugten Alarmen, die Simulation des Prozessbetriebs zum Zwecke der Personalschulung oder das Testen der Prozesssteuerungssoftware, die Führung und Aktualisierung einer Konfigurationsdatenbank etc.

[0006] Das beispielsweise von Fisher Rosemount Systems, Inc. vertriebene DeltaV™-Steuerungssystem umfasst zahlreiche gespeicherte und von verschiedenen, an den diversen Standorten innerhalb einer Prozessanlage angeordneten Vorrichtungen ausgeführte Applikationen. Mit einer Konfigurierungsapplikation, die in einer oder mehreren Bediener-Workstations residiert, können Nutzer Prozesssteuerungsmodul erzeugen oder ändern und diese über einen Datenhighway in zweckgebundene verteilte Steuerungsgeräte herunterladen. Diese Steuerungsmodul bestehen typischerweise aus kommunikativ miteinander verbundenen Funktionsblöcken, die Objekte in einem objektorientierten Programmprotokoll sind und auf Basis von Eingaben Funktionen innerhalb des Steuerungsschemas ausführen und Ausgaben zu anderen Funktionsblöcken innerhalb des Steuerungsschemas bereitstellen. Die Konfigurierungsapplikation kann einem Konfigurierer auch die Möglichkeit geben, Bediener-schnittstellen zu erzeugen oder zu ändern, die von einer Anzeigeapplikation genutzt werden, um einem Bediener Daten anzuzeigen, so dass er innerhalb der Prozesssteuerungsroutinen Einstellungen wie z.B. Sollwerte ändern kann. Jedes zweckgebundene Steuerungsgerät sowie in manchen Fällen Feldvorrichtungen speichern und führen eine Steuerungsapplikation aus, die die diesen zugeordneten und in diese herun-

tergeladenen Steuerungsmodul betreibt, um die aktuellen Prozesssteuerungsfunktionalität zu implementieren. Die Anzeigeapplikationen, die in einer oder mehreren Bediener-Workstations ausgeführt werden können, empfangen Daten von der Steuerungsapplikation über den Datenhighway und zeigen diese Daten den Konfigurierern, Bedienern oder Nutzern des Prozesssteuerungssystems an, die über die Nutzerschnittstellen zugreifen, und können eine Vielzahl von verschiedenen Anzeigen bereitstellen, wie z.B. Anzeigen für Bediener, Ingenieure, Techniker etc. Eine Datenhistorieapplikation wird typischerweise in einer Datenhistorie gespeichert und von dieser ausgeführt, die einige oder alle der über den Datenhighway bereitgestellten Daten erfasst und speichert, während eine Applikation der Konfigurationsdatenbank in einem weiteren Computer ausgeführt werden kann, der mit dem Datenhighway verbunden ist, um die jeweils aktuelle Konfiguration der Prozesssteuerungsroutine und die dazugehörigen Daten zu speichern. Alternativ kann die Konfigurationsdatenbank in der gleichen Workstation gespeichert sein wie die Konfigurierungsapplikation.

[0007] Gegenwärtig können Konfigurierungsapplikationen eine Bibliothek von Objekttemplates bzw. Templateobjekten umfassen, wie beispielsweise Funktionsblockobjekttemplates bzw. Funktionsblocktemplateobjekte und in einigen Fällen Steuerungsmodulobjekttemplates bzw. Steuerungsmodultemplateobjekte. Diese Konfigurierungsapplikationen werden genutzt, um eine Steuerungsstrategie für eine Prozessanlage zu konfigurieren. Alle Objekttemplates haben vorgegebene Eigenschaften, Einstellungen und Methoden, und der Ingenieur, der die Konfigurierungsapplikation anwendet, kann diese Objekttemplates auswählen und wie erforderlich Kopien der ausgewählten Objekttemplates in einem Konfigurationsbildschirm anordnen, um ein Steuerungsmodul zu entwickeln. Während der Auswahl und Anordnung der Objekttemplates im Konfigurationsbildschirm verbindet der Ingenieur die Eingänge und Ausgänge dieser Objekte miteinander und ändert ihre Parameter, Namen, Tags und anderen Eigenschaften, um ein spezifisches Steuerungsmodul für eine bestimmte Verwendung in der Prozessanlage zu erzeugen. Nach Erzeugung eines oder mehrerer derartiger Steuerungsmodul kann der Ingenieur das Steuerungsmodul dann instantiiieren und es in das oder die dafür vorgesehene(n) Steuerungsgeräte) bzw. die Feldvorrichtungen zur Ausführung während des Betriebs der Prozessanlage herunterladen.

[0008] Anschließend erstellt der Ingenieur im Allgemeinen ein oder mehrere Displays für Bediener, Wartungspersonal etc. innerhalb der Prozessanlage, indem er in einer Applikation zur Erzeugung von Bildschirmausgaben Displayobjekte auswählt und erstellt. Diese Bildschirmausgaben werden typischerweise auf systemweiter Basis in einer oder mehreren

der Workstations implementiert und stellen vorkonfigurierte Bildschirmausgaben für den Bediener oder das Wartungspersonal bezüglich des Betriebszustands des Steuerungssystems oder der Vorrichtungen innerhalb der Anlage bereit. Diese Bildschirmausgaben haben typischerweise die Form von Alarmdisplays, die von Steuerungsgeräten oder Vorrichtungen innerhalb der Prozessanlage erzeugte Alarme empfangen und anzeigen, von Steuerungsdisplays, die den Betriebszustand der Steuerungsgeräte und anderer Vorrichtungen innerhalb der Prozessanlage anzeigen, von Wartungsdisplays, die den Funktionszustand der Vorrichtungen innerhalb der Prozessanlage anzeigen, etc. Diese Bildschirmausgaben sind im Allgemeinen vorkonfiguriert, um auf bekannte Weise von den Prozesssteuerungsmodulen oder den Vorrichtungen innerhalb der Prozessanlage empfangene Information oder Daten anzuzeigen. In einigen bekannten Systemen werden Bildschirmausgaben mit Hilfe von Objekten erzeugt, die eine mit einem physikalischen oder logischen Element verbundene Grafik besitzen und kommunikativ mit dem physikalischen oder logischen Element verbunden sind, um Daten bezüglich des physikalischen oder logischen Elements zu empfangen. Das Objekt kann die Grafik auf dem Anzeigebildschirm auf Basis der empfangenen Daten ändern, beispielsweise um den halbvollen Zustand eines Behälters oder den von einem Durchflusssensor gemessenen Durchsatz etc. anzuzeigen.

[0009] Ähnlich wie die Applikation zur Steuerungskonfigurierung hat eine Applikation zur Erzeugung von Bildschirmausgaben graphische Anzeigeelementtemplates, wie z.B. Behälter, Ventile, Sensoren, Bedienelemente wie Schieberegler, Ein-/Ausschalter etc., die in jeder gewünschten Konfiguration auf einem Bildschirm angeordnet werden können, um eine Bildschirmausgabe für Bediener, Wartungspersonal o.ä. zu erzeugen. Bei ihrer Anordnung auf dem Bildschirm können einzelne Grafikelemente auf dem Bildschirm so miteinander verbunden werden, dass sie für die Nutzer Informationen oder Bildschirmausgaben der internen Abläufe der Prozessanlage bereitstellen. Um die graphische Bildschirmausgabe zu animieren, muss der Erzeuger der Bildschirmausgabe jedoch jedes Grafikelement von Hand mit den innerhalb der Prozessanlage erzeugten Daten verknüpfen, wie z.B. mit von den Sensoren gemessenen oder mit den Ventilstellungen anzeigenden Daten etc., und zwar durch die Vorgabe eines Kommunikationslinks zwischen dem Grafikelement und der relevanten Datenquelle innerhalb der Prozessanlage. Dieser Prozess ist langwierig, zeitraubend und ggf. fehlerträchtig.

[0010] Obwohl die Steuerungsobjekttemplates der Applikation zur Steuerungskonfigurierung und die Anzeigeelemente der Applikation zur Erzeugung von Bildschirmausgaben zweckmäßig sind, da sie kopiert und genutzt werden können, um viele verschiedene

Steuerungsmodul und graphische Displays zu erzeugen, besteht häufig die Notwendigkeit, zahlreiche gleiche Steuerungsmodul und graphische Displays für verschiedene Ausrüstungskomponenten innerhalb der Prozessanlage zu erzeugen. So gibt es beispielsweise in vielen mittleren bis großen Prozessanlagen zahlreiche Instanzen gleicher oder ähnlicher Ausrüstungskomponenten, die unter Verwendung des gleichen allgemeinen Grundsteuerungsmoduls und -displays gesteuert und angezeigt werden können. Zur Erzeugung dieser zahlreichen Steuerungsmodul und Displays wird jedoch ein allgemeines Steuerungsmodul oder Displaymodul erzeugt, das dann für jede dieser verschiedenen Ausrüstungskomponenten kopiert wird, auf die es anwendbar ist. Nach Erstellung der Kopie muss selbstverständlich jedes der neuen Steuerungs- oder Displaymodule von Hand in der Konfigurierungsapplikation geändert werden, um die besondere Ausrüstungskomponente vorzugeben, der es zugeordnet ist, und alle diese Steuerungs- und Displaymodule müssen dann instanziiert und in das Prozesssteuerungssystem heruntergeladen werden.

Aufgabenstellung

[0011] Leider sind die vorbeschriebenen Steuerungsmodul und Displayelemente in keiner Weise modular strukturiert, so dass nach Erstellung der Kopien jedes Steuerungsmodul und jede Bildschirmausgabe unter Verwendung der entsprechenden Konfigurierungsapplikation einzeln und von Hand geändert werden muss, um die Ausrüstungskomponente innerhalb der Anlage vorzugeben, der sie zugeordnet werden müssen. In einer Anlage mit zahlreichen Exemplaren des gleichen Ausrüstungstyps (d.h. replizierte Ausrüstungskomponenten) ist dieses Verfahren langwierig, zeitraubend und anfällig für vom Bediener verursachte Fehler. Außerdem haben diese verschiedenen Steuerungsmodul und Bildschirmausgaben nach ihrer Programmierung keine Kenntnis vom Vorhandensein der jeweils anderen Modul und Elemente. Um also die Steuerungsmodul nach ihrer Erzeugung zu ändern, muss der Ingenieur oder Bediener von Hand die gleiche Änderung an jedem der verschiedenen Steuerungsmodul für die unterschiedlichen, mehrfach vorhandenen Ausrüstungskomponenten vornehmen, was ebenfalls zeitintensiv und langwierig ist. Das gleiche Problem tritt bei den graphischen Darstellungen auf, die für die verschiedenen Sätze mehrfach vorhandener Ausrüstungskomponenten innerhalb der Anlage erstellt wurden. Anders gesagt, wurde ein spezifisches Steuerungsmodul oder eine spezifische graphische Darstellung einmal erzeugt (individuell oder durch das Kopieren eines Objekttemplates) und dann mit einem bestimmten Satz von Ausrüstungskomponenten innerhalb der Anlage verknüpft, so existiert dieses Steuerungsmodul bzw. diese graphische Darstellung als eine eigene Entität oder ein eigenes Objekt innerhalb

des Systems, ohne automatisch Kenntnis von der Existenz anderer gleicher oder ähnlicher Steuerungsmodul oder graphischer Darstellungen zu haben. Änderungen, die für jedes einzelne Steuerungsmodul und jede graphische Darstellung eines bestimmten Typs gelten, müssen daher einzeln an diesen Modul und graphischen Darstellungen vorgenommen werden.

[0012] Da außerdem jedes Steuerungsmodul und jedes Display ein individuelles Objekt ist, muss es in dem Sinn offen sein, dass alle internen Parameter, Darstellungen, Funktionsblöcke und andere Elemente für jeden Nutzer für Änderungen, Einsicht etc. verfügbar sind. Derzeit existiert kein Verfahren, um zu steuern, was bestimmte Bediener oder andere Nutzer in den Steuerungsmodul und Displays einsehen können bzw. worauf sie Zugriff haben, so dass keine Möglichkeit besteht, bestimmte Elemente der Steuerungsmodul und Displays, wie z.B. proprietäre Software und Verfahren, Alarmaktivitäten etc. für den Nutzer der Steuerungsmodul und Displays unsichtbar zu machen.

Ausführungsbeispiel

Zusammenfassung

[0013] Ein Versionskontrollsystem unterstützt die Verwaltung von Versionen von Prozessanlagenelementen, die Entitäten in einer Prozessanlage repräsentieren oder repräsentieren können. Die Prozessanlagenelemente können beispielsweise Modulobjekte umfassen, die spezifisch Prozessentitäten der Prozessanlage repräsentieren können. Diese Modulobjekte können aus Modulklassenobjekten erzeugt werden, die generisch Prozessentitäten der Prozessanlage repräsentieren können. Versionsdaten werden gespeichert und einem Modulobjekt zugeordnet. Zu den Versionsdaten können Daten gehören, die eine Version eines Modulklassenobjekts bezeichnen, das genutzt wurde, um das Modulobjekt zu erzeugen. Die Versionsdaten können ferner Daten umfassen, die eine Version des Modulobjekts bezeichnen. Konfigurierungssysteme, Versionskontrollsysteme, Anzeigesysteme, Fehlerbereinigungssysteme, Run-Time-Überwachungssysteme, Asset-Managementssysteme etc. können die einem Element zugeordneten Versionskontrolldaten prüfen oder ihre Anzeige zulassen.

Kurzbeschreibungen der Zeichnungen

[0014] [Fig. 1](#) ist ein Blockschaltbild eines verteilten Prozesssteuerungsnetzwerks, das innerhalb einer Prozessanlage angeordnet ist und das eine Bediener-Workstation aufweist, die unter Nutzung von Modulklassenobjekten Steuerungs- und Displayaktivitäten für die Prozessanlage konfiguriert;

[0015] [Fig. 2](#) ist ein Schaltbild einer Reaktoreinheit gemäß [Fig. 1](#);

[0016] [Fig. 3](#) ist ein Schaltbild einer Zumesausrüstungsentität bzw. einer Totalisatorausrüstungsentität, die in der Reaktoreinheit gemäß [Fig. 2](#) verwendet wird;

[0017] [Fig. 4](#) ist ein Schaltbild eines Auslassventilsystems der Reaktoreinheit gemäß [Fig. 2](#);

[0018] [Fig. 5](#) ist ein Blockschaltbild, welches die Beziehungen der Modulklassenobjekte und der zugeordneten Modulobjekte für Einheiten-, Ausrüstungs-, Steuerungs- und Display-Modulklassenobjekte verdeutlicht;

[0019] [Fig. 6](#) ist ein Blockschaltbild eines Einheiten-Modulklassenobjekts für Reaktoren, das verwendet werden kann, um Konfigurierungsmaßnahmen an Reaktoren der Anlage gemäß [Fig. 1](#) durchzuführen;

[0020] [Fig. 7](#) ist ein Blockschaltbild eines Zumesausrüstungs-Modulklassenobjekts, das verwendet werden kann, um Konfigurierungsmaßnahmen an Zumesungen der Anlage gemäß [Fig. 1](#) durchzuführen;

[0021] [Fig. 8](#) ist eine Darstellung eines ersten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0022] [Fig. 9](#) ist eine Darstellung eines zweiten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0023] [Fig. 10](#) ist eine Darstellung eines dritten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0024] [Fig. 11](#) ist eine Darstellung eines vierten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0025] [Fig. 12](#) ist eine Darstellung eines fünften Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0026] [Fig. 13](#) ist eine Darstellung eines sechsten

Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0027] [Fig. 14](#) ist eine Darstellung eines siebten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0028] [Fig. 15](#) ist eine Darstellung eines achten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren; und

[0029] [Fig. 16](#) ist eine Darstellung eines neunten Konfigurationsbildschirms, der von einem Konfigurationsbediener verwendet werden kann, um eine Prozessanlage unter Verwendung von Modulklassenobjekten zu konfigurieren;

[0030] [Fig. 17](#) ist ein Flussdiagramm einer beispielhaften Routine für die Konfigurierung eines Prozessanlagenelements zur Begrenzung des Zugriffs auf das Element;

[0031] [Fig. 18A](#) ist eine beispielhafte Bildschirmabgabe von Unterelementen einschließlich Funktionsblöcken eines Composite-Templates;

[0032] [Fig. 18B](#) ist eine beispielhafte Bildschirmabgabe von Unterelementen des Composite-Templates gemäß [Fig. 18A](#), in der die Unterelemente der Funktionsblöcke jedoch unsichtbar sind;

[0033] [Fig. 19](#) ist ein Flussdiagramm einer beispielhaften Routine für die Konfigurierung einer Gruppe von Prozessanlagenelementen für die Begrenzung des Zugriffs auf die Gruppe;

[0034] [Fig. 20](#) ist ein Flussdiagramm einer beispielhaften Routine, mit der bestimmt wird, ob die Anzeige von mit einer Prozessanlage verbundenen Daten zulässig ist;

[0035] [Fig. 21](#) ist ein Flussdiagramm einer beispielhaften Routine, mit der bestimmt wird, ob die Änderung eines Prozessanlagenelements oder einer Gruppe von Prozessanlagenelementen zulässig ist;

[0036] [Fig. 22](#) ist ein Flussdiagramm einer beispielhaften Routine für das Exportieren eines Prozessanlagenelements;

[0037] [Fig. 23](#) ist ein Blockdiagramm eines beispielhaften Konfigurationssystems, das verwendet werden kann, um Versionskontroll- und -prüfpfadinformation in Verbindung mit einer Prozessanlage bereit-

zustellen;

[0038] [Fig. 24](#) ist ein Flussdiagramm einer beispielhaften Routine, um die Verfolgung von Versionen eines Prozessanlagenelements zu vereinfachen;

[0039] [Fig. 25](#) ist ein Flussdiagramm einer beispielhaften Routine, um die Verfolgung von Versionen von Prozessanlagenelementen zu vereinfachen;

[0040] [Fig. 26](#) ist ein Flussdiagramm einer beispielhaften Routine zur einfacheren Konfigurierung eines Prozessanlagenelements; und

[0041] [Fig. 27](#) ist ein Flussdiagramm einer beispielhaften Routine zur einfacheren Konfigurierung von Prozessanlagenelementen;

[0042] [Fig. 28](#) ist eine beispielhafte Bildschirmausgabe einer Nutzerschnittstelle für die Konfigurierung eines Modulklassenobjekts;

[0043] [Fig. 29](#) ist eine weitere beispielhafte Bildschirmausgabe einer Nutzerschnittstelle für die Konfigurierung eines Modulklassenobjekts;

[0044] [Fig. 30](#) ist eine beispielhafte Bildschirmausgabe einer Nutzerschnittstelle für die Fehlerbereinigung eines Modulobjekts, in dem mit dem Modulobjekt verbundene Daten nicht unsichtbar sind; und

[0045] [Fig. 31](#) ist die beispielhafte Bildschirmausgabe gemäß [Fig. 30](#), in der jedoch mit dem Modulobjekt verbundene Daten unsichtbar sind.

DETAILLIERTE BESCHREIBUNG

[0046] [Fig. 1](#) zeigt eine Prozessanlage **10**, zu der ein oder mehrere Prozesssteuerungsgeräte **12** gehören, die beispielsweise über eine Ethernet-Verbindung oder einen Bus **15** mit einer Vielzahl von Workstations **14** verbunden sind. Die Steuerungsgeräte **12** sind außerdem über diverse Kommunikationsleitungen oder Busse **18** mit Vorrichtungen oder Ausrüstungskomponenten in der Prozessanlage **10** verbunden, wobei nur das an das Steuerungsgerät **12a** angeschlossene Kommunikationsleitungsbündel **18** in [Fig. 1](#) gezeigt ist. Die Steuerungsgeräte **12**, die beispielsweise unter ausschließlicher Verwendung des von Fisher-Rosemount Systems, Inc. vertriebenen Steuerungsgeräts DeltaV™ implementiert werden können, können mit Steuerungselementen wie beispielsweise Feldvorrichtungen und Funktionsblöcken in den Feldvorrichtungen kommunizieren, die über die gesamte Prozessanlage **10** verteilt sind, um eine oder mehrere Prozesssteuerungsroutinen **19** auszuführen und so die gewünschte Steuerung der Prozessanlage **10** zu implementieren. Die Workstations **14** (bei denen es sich z.B. um Personal Computer handeln kann), können von einem oder mehreren

Konfigurierungstechnikern benutzt werden, um die Prozesssteuerungsroutinen **19** zu konfigurieren, die von den Steuerungsgeräten **12** ausgeführt werden sollen, sowie die Displayroutinen, die von den Workstations **14** oder anderen Computern auszuführen sind, und um mit den Steuerungsgeräten **12** zu kommunizieren, um diese Prozesssteuerungsroutinen **19** in die Steuerungsgeräte **12** herunterzuladen. Ferner können die Workstations **14** Displayroutinen ausführen, die während des Betriebs der Prozessanlage **10** Information im Zusammenhang mit der Prozessanlage **10** oder Elementen dieser Anlage empfangen und anzeigen.

[0047] Jede der Workstations **14** beinhaltet einen Speicher **20** für die Speicherung von Applikationen, wie Konfigurationsapplikationen und Display- oder Anzeigeapplikationen, und für die Speicherung von Daten, wie z.B. zur Prozessanlage **10** gehörige Konfigurationsdaten. Jede der Workstations **14** umfasst auch einen Prozessor **21**, der die Applikationen ausführt, um einem Konfigurierungstechniker die Entwicklung von Prozesssteuerungsroutinen und anderen Routinen und das Herunterladen dieser Prozesssteuerungsroutinen in die Steuerungsgeräte **12** oder andere Computer zu ermöglichen, oder um die Erfassung und Anzeige von Information für einen Nutzer während des Betriebs der Prozessanlage **10** zu realisieren.

[0048] Außerdem gehört zu jedem Steuerungsgerät **12** ein Speicher **22**, der Steuerungs- und Kommunikationsapplikationen speichert, und ein Prozessor **24**, der die Steuerungs- und Kommunikationsapplikationen auf bekannte Weise ausführt. In einem Fall speichern und führen alle Steuerungsgeräte **12** eine Steuerungsapplikation aus, die eine Steuerungsstrategie unter Verwendung einer Anzahl unterschiedlicher, unabhängig ausgeführter Steuerungsmodule oder Blöcke **19** implementiert. Die Steuerungsmodule **19** können alle aus im Allgemeinen als Funktionsblöcke bezeichneten Elementen bestehen, wobei jeder Funktionsblock ein Teil oder eine Subroutine einer Gesamtsteuerungsroutine ist und (über als Links bezeichnete Kommunikationen) gemeinsam mit anderen Funktionsblöcken arbeitet, um Prozessregelungsschleifen innerhalb der Prozessanlage **10** zu implementieren. Wie allgemein bekannt, führen Funktionsblöcke, die Objekte in einem objektorientierten Programmprotokoll sein können, typischerweise entweder eine Eingabefunktion aus, die z.B. einem Messwertgeber, einem Sensor oder einer anderen Prozessparametermessvorrichtung zugeordnet ist, eine Steuerungsfunktion, die beispielsweise mit einer Steuerungsroutine, die PID-, Fuzzy-Logik und sonstige Steuerungen etc. ausführt, verbunden ist, oder eine Ausgabefunktion, die den Betrieb einer Vorrichtung steuert, beispielsweise eines Ventils, um eine physikalische Funktion innerhalb der Prozessanlage **10** auszuführen. Selbstverständlich existieren

Hybrid- oder andere Typen komplexer Funktionsblöcke, z.B. Model Predictive Controller (MPCs), Optimierer etc. Während das Fieldbus-Protokoll und das DeltaV™-Systemprotokoll in einem objektorientierten Programmprotokoll entwickelte und implementierte Steuerungsmodulare und Funktionsblöcke nutzen, könnten die Steuerungsmodulare unter Verwendung jedes gewünschten Steuerungsprogrammschemas entwickelt werden, einschließlich beispielsweise sequentiellen Funktionsblöcken, Kontaktplänen etc., ohne Beschränkung auf die Verwendung von Funktionsblöcken oder einer sonstigen speziellen Programmierertechnik.

[0049] Die Workstations **14** können über einen Bildschirm mit Darstellung der Steuerungselemente in den Prozesssteuerungsroutinen **19** eine graphische Darstellung der Prozesssteuerungsroutinen **19** in den Steuerungsgeräten **12** für einen Nutzer bereitstellen und die Konfiguration dieser Steuerungselemente für die Steuerung der Prozessanlage **10** zeigen. In dem System gemäß **Fig. 1** ist eine Konfigurationsdatenbank **25** mit dem Ethernet-Bus **15** verbunden, um von den Steuerungsgeräten **12** und den Workstations **14** benutzte Konfigurationsdaten zu speichern und durch das Erfassen und Speichern von in der Prozessanlage **10** erzeugten Daten für eine spätere Nutzung als Datenhistorie zu dienen.

[0050] In der Prozessanlage **10** der **Fig. 1** ist das Steuerungsgerät **12a** über Bus **18** kommunikativ mit drei Gruppen gleich konfigurierter Reaktoren verbunden (bei denen es sich um mehrfach vorhandene Ausrüstungskomponenten innerhalb der Anlage **10** handelt), die hierin als Reactor_01, Reactor_02 und Reactor_03 bezeichnet werden. Reactor_01 umfasst einen Reaktionsbehälter oder Tank **100**, drei Einlassventilsysteme (Ausrüstungsentitäten) **101**, **102** und **103**, die so angeschlossen sind, dass sie die Fluid-einlassleitungen für Säure, Alkali bzw. Wasser zum Reaktionsbehälter **100** kontrollieren, und ein Auslassventilsystem **104**, das den Fluidabfluss aus dem Reaktionsbehälter **100** kontrolliert. Ein Sensor **105**, bei dem es sich um jeden gewünschten Sensortyp, wie z.B. einen Pegelmessers, einen Temperaturfühler, einen Drucksensor etc. handeln kann, ist im oder in der Nähe des Reaktionsbehälters **100** angeordnet. Für den Zweck dieser Ausführungen wird angenommen, dass es sich beim Sensor **105** um einen Pegelmessers handelt. Ferner ist ein gemeinsam genutztes Wasserheadersystem **110** an die Wasserleitung stromaufwärts von jedem der Reaktoren Reactor_01, Reactor_02 und Reactor_03 angeschlossen, um eine Hauptkontrolle für den Wasserfluss zu jedem dieser Reaktoren bereitzustellen.

[0051] Dementsprechend gehören zu Reactor_02 ein Reaktionsbehälter **200**, drei Einlassventilsysteme **201**, **202** und **203**, ein Auslassventilsystem **204** und ein Pegelmessers **205**, während Reactor_03 einen

Reaktionsbehälter **300**, drei Einlassventilsysteme **301**, **302** und **303**, ein Auslassventilsystem **304** und einen Pegelmessers **305** umfasst. Im Beispiel der **Fig. 1** können die Reaktoren Reactor_01, Reactor_02 und Reactor_03 Salz produzieren, mit Zugabe von Säure durch die Einlassventilsysteme **101**, **201** und **301**, mit Zugabe von Alkali durch die Einlassventilsysteme **102**, **202** und **302**, während die Einlassventilsysteme **103**, **203** und **303** in Verbindung mit dem gemeinsam genutzten Wasserheader **110** Wasser in den Reaktionsbehälter **100** einleiten. Die Auslassventilsysteme **104**, **204** und **304** können so betätigt werden, dass das Produkt über eine Leitung, die in **Fig. 1** nach rechts gerichtet ist, abgezogen und Abfall oder anderes unerwünschte Material durch eine in **Fig. 1** nach unten gerichtete Leitung abgeleitet wird.

[0052] Das Steuerungsgerät **12a** ist über den Bus **18** kommunikativ mit den Ventilsystemen **101-104**, **110**, **201-204** und **301-304** und den Sensoren **105**, **205** und **305** verbunden, um den Betrieb dieser Elemente zu steuern und eine oder mehrere Operationen in Zusammenhang mit den Reaktoreinheiten Reactor_01, Reactor_02 und Reactor_03 auszuführen. Zu diesen allgemein als Phasen bezeichneten Operationen können beispielsweise das Befüllen der Reaktionsbehälter **100**, **200**, **300**, das Erhitzen des Materials in den Reaktionsbehältern **100**, **200**, **300**, das Entleeren der Reaktionsbehälter **100**, **200**, **300**, das Reinigen der Reaktionsbehälter **100**, **200**, **300** etc. gehören.

[0053] Bei den in **Fig. 1** dargestellten Ventilen, Sensoren und anderen Ausrüstungskomponenten kann es sich um beliebige Ausrüstungstypen handeln, z.B. auch um Fieldbus-Vorrichtungen, Standard 4-20mA-Vorrichtungen, HART-Vorrichtungen etc., die mit dem Steuerungsgerät **12** unter Verwendung aller bekannten oder gewünschten Kommunikationsprotokolle kommunizieren können, z.B. mit dem Fieldbus-Protokoll, dem HART-Protokoll, dem 4-20 mA-Analogprotokoll etc. Weiter können gemäß den hierin erläuterten Prinzipien andere Vorrichtungstypen mit den Steuerungsgeräten **12** verbunden und von ihnen gesteuert werden. Ferner können verschiedene Anzahlen und andere Typen von Steuerungsgeräten innerhalb der Anlage **10** angeschlossen werden, um andere mit der Prozessanlage **10** verbundene Vorrichtungen oder Bereiche zu steuern, und der Betrieb dieser zusätzlichen Steuerungsgeräte kann auf jede gewünschte Weise mit dem Betrieb des in **Fig. 1** gezeigten Steuerungsgeräts **12a** koordiniert werden.

[0054] Allgemein gesagt, die Prozessanlage **10** der **Fig. 1** kann für die Implementierung von Batchprozessen eingesetzt werden, bei denen beispielsweise eine der Workstations **14** oder das Steuerungsgerät **12a** eine Batchroutine ausführt, bei der es sich um

eine übergeordnete Steuerungsroutine handelt, die den Betrieb einer oder mehrerer Reaktoreinheiten (sowie anderer Ausrüstungskomponenten) leitet, um eine Reihe verschiedener Schritte (gewöhnlich als Phasen bezeichnet) auszuführen, die für die Herstellung eines Produkts, z.B. einer bestimmten Salzart, erforderlich sind. Zur Implementierung verschiedener Phasen nutzt die Batchroutine im Allgemeinen eine Rezeptur, die die auszuführenden Schritte, die den Schritten zugeordneten Mengen und Zeiten sowie die Reihenfolge dieser Schritte vorgibt. Zu den Schritten einer Rezeptur kann beispielsweise das Befüllen eines Reaktionsbehälters mit den geeigneten Materialien bzw. den Ingredienzien, das Mischen der Materialien im Reaktionsbehälter, deren Erhitzung im Reaktionsbehälter auf eine bestimmte Temperatur während einer gewissen Zeit, das Entleeren des Reaktionsbehälters und dessen Reinigung zur Vorbereitung auf den nächsten Batchlauf gehören. Jeder dieser Schritte definiert eine Phase des Batchlaufs und die Batchroutine im Steuerungsgerät **12a** führt einen unterschiedlichen Steuerungsalgorithmus für jede dieser Phasen aus. Selbstverständlich können sich die jeweiligen Materialien, ihre Mengen, die Temperaturen und Zeiten für die Erhitzung etc. für die verschiedenen Rezepturen voneinander unterscheiden, und diese Parameter können sich daher je nach hergestelltem Produkt und angewandter Rezeptur von Batchlauf zu Batchlauf ändern. Dem Fachmann ist klar, dass hierin zwar Steuerungsroutinen und Konfigurationen für Batchläufe in den in [Fig. 1](#) gezeigten Reaktoren beschrieben werden, dass solche Steuerungsroutinen aber auch genutzt werden können, um andere gewünschte Vorrichtungen zu steuern, um jeden anderen gewünschten Batchprozesslauf oder auf Wunsch auch kontinuierliche Prozessläufe auszuführen.

[0055] Ferner ist klar, dass die gleichen Phasen oder Schritte eines Batchprozesses in jeder der verschiedenen Reaktoreinheiten der [Fig. 1](#) zur gleichen oder zu verschiedenen Zeiten implementiert werden können. Ferner kann, da die Reaktoreinheiten der [Fig. 1](#) im Allgemeinen die gleiche Anzahl und die gleichen Typen von Ausrüstungskomponenten umfassen, die gleiche generische Phasensteuerungsroutine für eine bestimmte Phase angewendet werden, um jede der verschiedenen Reaktoreinheiten zu steuern, wobei jedoch diese generische Phasensteuerungsroutine geändert werden muss, um die diversen, den verschiedenen Reaktoreinheiten zugeordneten Hardware- oder Ausrüstungskomponenten zu steuern. Um eine Befüllungsphase für Reactor_01 (in der die Reaktoreinheit befüllt wird) zu implementieren, öffnet beispielsweise eine Befüllsteuerungsroutine ein oder mehrere der Einlassventilsystemen **101**, **102** und **103** zugeordnete Ventile für eine bestimmte Zeit, z.B. bis der Pegelmesser **105** feststellt, dass der Behälter **100** voll ist. Diese gleiche Steuerungsroutine kann jedoch auch angewandt werden, um eine

Befüllungsphase für Reactor_02 zu implementieren, und zwar lediglich durch die Änderung der Bezeichnung des bzw. der Einlassventile in die mit den Ventilsystemen **201**, **202** und **203** verbundene Bezeichnung anstelle der Bezeichnung für die Ventilsysteme **101**, **102** und **103** und durch die Änderung der Bezeichnung des Pegelmessers, nämlich Pegelmesser **205** anstelle von Pegelmesser **105**.

[0056] [Fig. 2](#) zeigt genauere Einzelheiten eines der Reaktoren der [Fig. 1](#), im Besonderen Reactor_01. Wie ähnlich in [Fig. 1](#) dargestellt, umfasst der Reactor_01 der [Fig. 2](#) den Reaktionsbehälter **100**, die Einlassventilsysteme **101**, **102**, **103** und **110** für die Zugabe von Säure, Alkali und Wasser in den Behälter **100**, das Auslassventilsystem **104** für das Ablassen von Material aus dem Behälter **100** und den Pegelmesser **105**. Wie in [Fig. 2](#) weiter gezeigt, benutzt jedes der Einlassventilsysteme **101**, **102** und **110** eine ähnliche, als Zumessung bzw. Totalisator bezeichnete Ausrüstungsentität, zu der zwei parallel angeordnete Ventile und eine stromabwärts nach den beiden Ventilen angeordnete Durchflussmessvorrichtung gehören. Die Zumessung für das Einlassventilsystem **101**, die in [Fig. 3](#) genauer gezeigt wird, umfasst ein als Grobventil bezeichnetes Absperrventil **101a**, ein als Feinventil bezeichnetes Absperrventil **101b** und einen stromabwärts von den Ventilen **101a** und **101b** angeordneten Durchflussmesser **101c**. Die Zumessung **101** besitzt ein oder mehrere ihr zugeordnete Steuerungsmodule oder Routinen, die genutzt werden, um unter Verwendung der vom Durchflussmesser **101c** durchgeführten Messungen die Säurezugabe zu steuern. Mit einer ersten derartigen Steuerungsroutine kann durch die Zumessung **101** unter Nutzung von Grobventil **101a** und Feinventil **101b** ein schneller Strom gesteuert werden, während eine zweite derartige Steuerungsroutine mit der Zumessung **101** unter Nutzung von Grobventil **101a** und Feinventil **101b** den exakten Strom steuert.

[0057] Wie aus [Fig. 2](#) zu ersehen, umfasst das Ventilsystem **102** für die Alkalizugabe eine Zumessung mit einem Grobventil **102a**, einem Feinventil **102b** und einem Durchflussmesser **102c**, und das gemeinsam genutzte Wassereinlassventilsystem **110** umfasst ein Grobventil **110a**, ein Feinventil **110b** und einen Durchflussmesser **110c**. Jede der Zumessungen **101**, **102** und **110** hat den gleichen Typ mehrfach vorhandener Ausrüstungskomponenten, obwohl sie an verschiedenen Stellen in der gleichen Einheit, nämlich dem Reactor_01, eingesetzt werden. Ebenso umfassen Reactor_02 und Reactor_03 Zumessungen an den Einlassventilsystemen **201**, **202**, **301** und **302**.

[0058] In gleicher Weise bildet das Auslassventilsystem **104** eine weitere der mehrfach vorhandenen Ausrüstungskomponenten, zu der drei Ventile gehören. Wie am besten aus [Fig. 4](#) erkennbar, umfasst

das Auslassventilsystem **104** ein Hauptauslassventil **104a**, das geöffnet werden muss, um Material aus dem Tank **100** abzulassen, ein Produktventil **104b**, das gemeinsam mit dem Hauptauslassventil **104a** geöffnet werden muss, um das Produkt aus dem Tank **100** abzuziehen, und ein Entleerungsventil **104c**, das gemeinsam mit dem Hauptauslassventil **104a** geöffnet werden muss, um Material, beispielsweise Abfallprodukte, Reinigungsflüssigkeit etc. vom Tank **100** in ein Ablass- oder Entsorgungssystem abzuleiten. Selbstverständlich werden dem Auslassventilsystem **104** eine oder mehrere Steuerungsroutinen zugeordnet, um den Zustand der Ventile **104a**, **104b** und **104c** zu steuern und den Behälter **100** zu schließen, zu entleeren oder um das Produkt aus dem Behälter **100** abzuziehen.

[0059] Um eine Steuerungsroutine für die Steuerung der verschiedenen, den Reaktoren Reactor_01, Reactor_02 und Reactor_03 der [Fig. 1](#) zugeordneten Ausrüstungskomponenten zu erstellen, erzeugte ein Konfigurierungstechniker bisher vielleicht zunächst eine Anzahl von generischen Steuerungsmodultemplates bzw. Templatesteuerungsmodulen, die beispielsweise in einer Bibliothek in einer der Workstations **14** gespeichert wurden. Um eine Steuerungsmodultemplate zu erzeugen, verband der Ingenieur graphisch verschiedene Steuerungsfunktionsblöcke miteinander, um eine Steuerungsroutine für verschiedene, im Reaktor ausgeführte Elemente oder Regelschleifen bereitzustellen. Nach der typischerweise auf Basis eines Ventils oder einer Regelschleife erfolgten Erzeugung der generischen Steuerungsmodultemplates konnten diese Steuerungsmodultemplates kopiert und ihre Kopien von Hand mit der entsprechenden Ausrüstungskomponente innerhalb der Anlage **10**, z.B. mit einer bestimmten Ausrüstungskomponente innerhalb der Reaktoren Reactor_01, Reactor_02 und Reactor_03 verknüpft werden. Nach ihrer Verknüpfung, die wie in der US-Patentschrift Nr. 6,385,496 genauer beschrieben entweder direkt oder unter Verwendung von Aliasnamen hergestellt wurde, wurden die verknüpften Kopien der Steuerungsmodulmodule in eines oder mehrere Steuerungsgeräte **12** heruntergeladen und benutzt, um Prozesssteuerungsmaßnahmen in den Reaktoren durchzuführen, mit denen sie verbunden waren. Die aus den Steuerungsmodultemplates erzeugten verknüpften Steuerungsmodulmodule hatten jedoch keinen Bezug bzw. keine Beziehung zu den Modultemplates, aus denen sie erzeugt wurden, und waren in Wirklichkeit bei ihrer Verwendung im Prozesssteuerungssystem eigenständige Steuerungsmodulmodule oder Objekte.

[0060] In diesen Systemen musste die Konfigurierung außerdem auf einer Steuerungsmodulebene durchgeführt werden, was bedeutet, dass für jede der verschiedenen Ausrüstungskomponenten oder Regelschleifen innerhalb der Prozessanlage ein eigenes Steuerungsmodul erzeugt werden musste. Auf

Steuerungsmodulebene gibt es typischerweise zahlreiche verschiedene Steuerungsmodultypen, die für jede der Prozessentitäten innerhalb der Prozessanlage erzeugt und mit ihr verbunden werden müssen. Im Ergebnis verbrachte der Konfigurierungstechniker viel Zeit lediglich mit dem Kopieren und Verbinden der einzelnen Steuerungsmodulmodule mit den individuellen Ausrüstungskomponenten innerhalb der Anlage. Beispielsweise musste ein Konfigurierungstechniker Dutzende von Steuerungsmodulmodulen für eine Reaktoreinheit in der Anlage erzeugen und kopieren und dann jedes einzelne dieser Steuerungsmodulmodule mit einer bestimmten Ausrüstungskomponente in dieser Reaktoreinheit verbinden. War die Reaktoreinheit in der Anlage mehrfach vorhanden, so wurde die Aufgabe des Konfigurierungstechnikers noch viel langwieriger, da der Konfigurierungstechniker diesen Kopier- und Verbindungsprozess für Dutzende von Steuerungsmodulmodulen für jede der mehrfach vorhandenen Ausrüstungskomponenten durchführen musste, was zeitintensiv und äußerst anfällig für menschliche Fehler war.

[0061] Obwohl früher ein Konfigurierungstechniker ein Einheitenmodul entwickeln konnte, waren diese Einheitenmodule lediglich Behälter für die Phasen, die in einer Einheit ablaufen konnten, und umfassten keine Ausrüstungsmerkmale in Verbindung mit der Einheit oder mit Steuerungsschemata, die für die Steuerung der Basisoperationen der Ausrüstungskomponenten innerhalb der Einheiten verwendet wurden. Obwohl auch Templates für Steuerungselemente erzeugt werden konnten, die für die Steuerung verschiedener Ausrüstungskomponenten genutzt wurden, gab es keine Steuerungsmodulpackages, die verwendet werden konnten, um auf mehrfach vorhandene übergeordnete Entitäten innerhalb der Anlage zu verweisen, z.B. auf Ausrüstungs- und Einheitenentitäten. Anders gesagt, um Steuerungsroutinen für verschiedene mehrfach vorhandene Entitäten innerhalb der Prozessanlage **10** zu erzeugen, musste der Konfigurierungstechniker Steuerungsmodulmodule für jede der mehrfach vorhandenen Ausrüstungskomponenten auf der niedrigsten Steuerungsebene kopieren und jedes dieser Steuerungsmodulmodule einer bestimmten Ausrüstungskomponente oder sonstigen Entität innerhalb der Prozessanlage **10** anpassen. In großen Anlagen mit einer Vielzahl mehrfach vorhandener Ausrüstungskomponenten konnte diese Aufgabe zeitraubend und anfällig für Konfigurierungsfehler sein. Außerdem mussten Änderungen an einem einer mehrfach vorhandenen Ausrüstungskomponente zugeordneten Steuerungsmodul von Hand an jedem dieser verschiedenen Steuerungsmodulmodule für die verschiedenen Ausrüstungskomponenten vorgenommen werden, was ebenfalls langwierig, zeitintensiv und anfällig für vom Bediener verursachte Fehler war. Ebenso mussten Displays für den Bediener von den Steuerungsmodulmodulen getrennt erzeugt werden und wie die Steuerungsmodulmodule mussten auch die

Displays individuell erzeugt, geändert und mit den Ausrüstungskomponenten innerhalb der Prozessanlage verbunden werden.

[0062] Für eine einfachere und weniger zeitraubende Erzeugung und Änderung einer Prozesskonfiguration umfasst eine in einer der Workstations **14** der [Fig. 1](#) gespeicherte Konfigurierungsapplikation **50** einen Satz von Modulklassenobjekten **52** zur Anwendung bei der Konfigurierung der Prozesssteuerungsanlage **10**. Die Modulklassenobjekte sind besonders nützlich bei der Konfigurierung einer Anlage mit zahlreichen Gruppen mehrfach vorhandener Ausrüstungskomponenten. Allgemein gesagt kann ein unterschiedliches Modulklassenobjekt **52** für jeden Typ der verschiedenen physikalischen Einheiten oder Ausrüstungskomponenten erzeugt werden, die innerhalb der Prozessanlage **10** mehrfach vorhanden sind oder genutzt werden, für jeden Typ einer Steuerungsmaßnahme, der in der Prozessanlage **10** mehrfach vorhanden ist oder genutzt wird, für jeden verschiedenen Typ von Displayapplikationen, der in der Prozessanlage **10** mehrfach vorhanden ist oder genutzt wird, etc. Nach ihrer Erzeugung können die Modulklassenobjekte **52** genutzt werden, um Elemente der Prozessanlage **10** zu konfigurieren, die Modulklassenobjekten entsprechen.

[0063] Die Modulklassenobjekte **52**, bei denen es sich im Wesentlichen um generische Versionen von Prozessentitäten handelt und die nicht mit einer bestimmten Prozessentität verbunden sind, können ihnen zugeordnete Objekte oder Instanzen **53**, **54**, **55** und **56** einer niedrigeren Ebene haben (die hierin als Modulobjekte oder Modulblöcke bezeichnet werden). Jedes Modulobjekt wird aus einem Modulklassenobjekt erzeugt und erbt die gleiche Struktur und die gleichen Eigenschaften wie das Modulklassenobjekt, aus dem es erzeugt wurde. Jedes Modulobjekt ist jedoch mit einer bestimmten Entität innerhalb der Prozessanlage **10** verknüpft. So kann ein einzelnes Modulklassenobjekt **52** erzeugt werden, um einen bestimmten Typ einer Reaktoreinheit zu repräsentieren (unabhängig davon, wie viele dieser Reaktoreinheiten in der Anlage **10** vorhanden sind), während ein anderes Modulobjekt **53** vorhanden sein oder für jede der verschiedenen Reaktoreinheiten dieses Typs, die tatsächlich in der Anlage **10** vorhanden sind, erzeugt werden kann.

[0064] Die aus einem Modulklassenobjekt erzeugten Modulobjekte werden dem Modulklassenobjekt zugeordnet und sind diesem zugehörig. Im Ergebnis können Änderungen am Modulklassenobjekt automatisch an jedes diesem Modulklassenobjekt zugeordnete Modulobjekt übertragen werden oder in diesem zur Verfügung stehen. Wenn eine Anzahl von Modulobjekten aus einem bestimmten Modulklassenobjekt erzeugt und jedes dieser verschiedenen Modulobjekte mit verschiedenen Prozessentitäten

verknüpft wurde, kann daher jedes der verschiedenen Modulobjekte einfach durch die Änderung des Modulklassenobjekts und die automatische Übertragung dieser Änderung auf die zugeordneten Modulobjekte geändert werden.

[0065] Auf die gleiche Weise können nach der Erzeugung der Modulobjekte aus einem Modulklassenobjekt diese Modulobjekte unter Verwendung eines Gruppenereditierungsverfahrens mit einer bestimmten Ausrüstungskomponente innerhalb der Prozessanlage **10** verknüpft werden. Insbesondere können, da die Modulobjekte eines bestimmten Modulklassenobjekts alle mit dem gleichen Modulklassenobjekt verbunden sind oder ihm zugehören, alle diese Modulobjekte beispielsweise unter Verwendung einer Applikation vom Typ Tabellenkalkulation gemeinsam konfiguriert werden, wodurch die Spezifizierung der besonderen Beziehungen zwischen den Modulobjekten und der entsprechenden Ausrüstungskomponente in der Anlage **10**, der diese Modulobjekte zugeordnet sind, einfacher wird und weniger Zeit in Anspruch nimmt.

[0066] Die Modulklassenobjekte **52** der [Fig. 1](#) können das sein, was in einer objektorientierten Programmumgebung oder -sprache gewöhnlich als Objekte bezeichnet wird. Im Ergebnis können diesen Objekten andere Objekte zugehörig sein bzw. sie können sich auf andere Objekte beziehen. Allgemein gesagt sind die Modulklassenobjekte **52** Objekte einer höheren Ebene, die Merkmale oder Definitionen individueller Elemente, wie z.B. von Steuerungsroutinen, Ausrüstungskomponenten oder anderen Elementen beinhalten können, die einer Prozessentität zugeordnet sind, zusammen mit einer Definition oder einer Kennzeichnung für die Art der Wechselbeziehungen zwischen diesen individuellen Elementen, z.B. die Art der Verbindung zwischen physikalischen Elementen oder die Art des Zusammenwirkens zwischen logischen Elementen und physikalischen Elementen. Anders gesagt, ein Modulklassenobjekt kann beispielsweise ein Objekt innerhalb einer objektorientierten Programmiersprache sein, das die Basis für die Steuerung und Ansicht einer bestimmten Ausrüstungskomponente oder einer Gruppe von Ausrüstungskomponenten bereitstellt, ein Steuerungselement, ein Display etc. innerhalb der Prozessanlage **10**, das für die Erzeugung zahlreicher Instanzen dieses Elements nützlich sein kann, die für die Konfigurierung verschiedener, mehrfach vorhandener Ausrüstungskomponenten innerhalb der Prozesssteuerungsanlage **10** verwendet werden können.

[0067] Grundsätzlich ist jedes Modulklassenobjekt ein Konfigurationsbehälter, der eine generische Definition einer Prozessentität in Form aller der verschiedenen Steuerungs- und/oder Displayapplikationen oder -routinen umfasst, die auf diese Entität anwendbar sind, um von den Steuerungsgeräten **12** genutzt

zu werden, um diese Entität zu steuern, oder um von den Workstations **14** für Displayaktivitäten in Bezug auf diese Entität genutzt zu werden. Das Modulklassenobjekt kann eine Prozessentität beliebiger Art repräsentieren, z.B. eine Einheit, eine Ausrüstungskomponente, eine Steuerungsentität, eine Displayapplikation etc. Während der Konfigurierung der Prozessanlage **10** kann das Modulklassenobjekt genutzt werden, um Konfigurierungsinstanzen der Prozessentität für eine beliebige Anzahl verschiedener Prozessentitäten zu erzeugen, die der vom Modulklassenobjekt bereitgestellten Definition entsprechen, wobei jede Konfigurierungsinstanz (das aus dem Modulklassenobjekt erzeugte Modulobjekt) einer anderen tatsächlichen Prozessentität zugeordnet oder mit ihr verknüpft wird. Diese verschiedenen Modulobjekte umfassen u.a. Steuerungsroutinen und/oder Displayroutinen, die mit besonderen, in der Prozessanlage **10** angeordneten Prozessentitäten verbunden sind, wobei diese Steuerungsroutinen in die Steuerungsgeräte **12** der [Fig. 1](#) heruntergeladen und dort genutzt werden können, um aktuelle Steuerungsmaßnahmen in den Prozessentitäten durchzuführen, und wobei die Displayroutinen in die Workstations **14** heruntergeladen werden können, um aktuelle Displayaktivitäten in Bezug auf die Entitäten während des Betriebs der Prozessanlage **10** auszuführen.

[0068] Unterschiedliche Modulklassenobjekttypen können für Prozessentitäten unterschiedlicher Anwendungsbereiche stehen und daher Steuerungs- und/oder Displayroutinen enthalten, die für die Anwendung auf bzw. in Bezug auf Prozessentitäten unterschiedlicher Anwendungsbereiche konfiguriert sind. Je größer der Anwendungsbereich der Prozessentität, z.B. einer Einheit, umso mehr Steuerungs- und/oder Displayroutinen werden typischerweise dem Modulklassenobjekt zugeordnet und umso einfacher ist die Konfigurierung von Abschnitten der Anlage unter Nutzung dieser Modulklassenobjekte. Je größer jedoch der Anwendungsbereich der mit einem Modulklassenobjekt verbundenen Prozessentität ist, umso unwahrscheinlicher ist es, dass der Prozess mehrfach vorhandene Ausrüstungskomponenten entsprechend diesem Anwendungsbereich umfasst, und daher umso geringer die Wahrscheinlichkeit, dass das Modulklassenobjekt in großem Maßstab von Nutzen ist. Je kleiner jedoch im Gegensatz dazu der Anwendungsbereich der mit einem Modulklassenobjekt verbundenen Prozessentität ist, desto wahrscheinlicher kann das Modulklassenobjekt in mehreren verschiedenen Standorten der Anlage genutzt werden, während gleichzeitig die geringste Konfigurierung bei Verwendung dieses Modulklassenobjekts in einer bestimmten Instanz erforderlich wird. Die Modulklassenobjekte ermöglichen auf jeden Fall die Konfigurierung verschiedener mehrfach vorhandener Ausrüstungskomponenten auf einer höheren Abstraktionsebene als der Steuerungsmodule-

bene, was die Konfigurierung einer Prozessanlage mit mehrfach vorhandenen Einheiten und anderen Ausrüstungskomponenten bei Verwendung von Modulklassenobjekten, speziell Modulklassenobjekten mit größerem Anwendungsbereich wie z.B. auf Einheitenebene, einfacher und weniger zeitraubend gestaltet.

[0069] In einem Ausführungsbeispiel kann ein Konfigurierungstechniker bei der Konfigurierung eines Prozesssteuerungssystems ein einziges Modulklassenobjekt für die verschiedenen in der Prozessanlage mehrfach vorhandenen Elemente, wie z.B. die verschiedenen Reaktoren der [Fig. 1](#) erzeugen. Anschließend kann der Konfigurierungstechniker Instanzen des Modulklassenobjekts (Modulobjekte) für jeden der tatsächlichen Reaktoren der [Fig. 1](#) erzeugen. Jedes so erzeugte Modulobjekt umfasst vom Steuerungsgerät **12a** genutzte Steuerungsroutinen, um einen der Reaktoren der [Fig. 1](#) zu betreiben, und ist spezifisch mit den Ausrüstungskomponenten in diesem einen der Reaktoren der [Fig. 1](#) verknüpft oder verbunden. Diese Steuerungsroutinen können dann in das Steuerungsgerät **12a** heruntergeladen und während des Betriebs der Prozessanlage **10** angewandt werden. Auch nachdem es erzeugt wurde, ist jedes der Modulobjekte immer noch mit dem Modulklassenobjekt verknüpft und kann von dem Modulklassenobjekt beeinflusst werden, um es zu ändern, Zugriff auf das Modulobjekt zu gestatten oder zu sperren etc. Ebenso können Modulobjekte aus dem gleichen Modulklassenobjekt gemeinsam erzeugt und definiert werden, beispielsweise können sie unter Verwendung eines Tabellenprogramms mit einer Ausrüstungskomponente innerhalb der Anlage verbunden werden.

[0070] Obwohl viele verschiedene Typen von Modulklassenobjekten möglich sind, die innerhalb einer Prozessanlage erzeugt oder verwendet werden können, um Konfigurierungsaktivitäten innerhalb der Prozessanlage auszuführen, werden hierin vier spezifische Typen als Beispiele erläutert, nämlich Einheiten-Modulklassenobjekte, Ausrüstungs-Modulklassenobjekte, Steuerungs-Modulklassenobjekte und Display-Modulklassenobjekte. Allgemein gesagt ist jeder Typ der verschiedenen Modulklassenobjekte für einen anderen Steuerungsumfang oder Einsatzbereich innerhalb der Prozessanlage **10** ausgelegt oder vorgesehen. Ein Einheiten-Modulklassenobjekt ist für die Darstellung (und Konfigurierung) von Steuerungsmaßnahmen für ein breites Spektrum von Ausrüstungskomponenten innerhalb einer Prozessanlage gedacht. Insbesondere ist ein Einheiten-Modulklassenobjekt dazu bestimmt, eine miteinander verbundene Gruppe von Ausrüstungskomponenten (typischerweise mehrfach vorhandene Komponenten), wie z.B. die Reaktoren der [Fig. 1](#), die individuelle, auf bekannte Art zusammenwirkende Elemente besitzen, zu modellieren bzw. für ihre Konfigurierung

genutzt zu werden.

[0071] Ein Ausrüstungs-Modulklassenobjekt ist zur Verwendung bei der Darstellung (und Konfigurierung) von Steuerungsmaßnahmen für ein weniger breites Spektrum physikalischer Ausrüstungskomponenten innerhalb der Prozessanlage bestimmt. Bei den einem Ausrüstungs-Modulklassenobjekt zugeordneten Ausrüstungskomponenten handelt es sich im Allgemeinen um eine oder mehrere physikalische Entitäten, wie Ventile, Durchflussmesser etc., die ein Untersystem einer Einheit bilden, und das Ausrüstungs-Modulklassenobjekt kann eine oder mehrere Anweisungen oder Algorithmen beinhalten, bei denen es sich um anweisungsgesteuerte Algorithmen (CDAs), zustandsgesteuerte Algorithmen (SDAs), sequentielle Folgesteuerungsalgorithmen (SFCs), Funktionsblockalgorithmen (FBDs), Phasenalgorithmen etc. handeln kann, die in Verbindung mit der betreffenden Ausrüstungskomponente auszuführen sind. Somit ist ein Ausrüstungs-Modulklassenobjekt dafür vorgesehen, die Konfigurierung der Steuerung mehrerer untergeordneter Komponenten oder Entitäten innerhalb einer Einheit zu bewirken, um einen Basisfunktionssatz für die betreffende Ausrüstungskomponente für die Verwendung in der jeweiligen Einheit bereitzustellen. Wie bekannt ist, wird ein anweisungsgesteuerter Algorithmus (anweisungsgesteuerte Steuerlogik) verwendet, wenn die untergeordneten Komponenten über mehrere Schritte aufeinander abgestimmt werden müssen, um eine Funktion auszuführen. Beispielsweise kann es sein, dass ein Ventil für einen bestimmten Zeitraum geöffnet und anschließend wieder geschlossen werden muss, während ein anderes Ventil geöffnet und geschlossen wird. Die Zumessung **101** der [Fig. 3](#) nutzt diese Art eines anweisungsgesteuerten Algorithmus, um zunächst das Grobventil anzusteuern und einzustellen, und dann das Feinventil auf Basis der Anzeige des Durchflussmessers zu regeln, um den gewünschten Gesamtdurchsatz durch die Zumessung bereitzustellen. Ein zustandsgesteuerter Algorithmus (zustandsgesteuerte Steuerlogik) kann die Zustände verschiedener untergeordneter Komponenten vorgeben, die in einem einzigen Schritt gesteuert werden können. Ein solcher zustandsgesteuerter Algorithmus könnte z.B. für das Auslassventilsystem **104** der [Fig. 4](#) verwendet werden, bei dem die Zustände der verschiedenen zugehörigen Ventile jeweils getrennt (aber in einem einzigen Schritt) auf Basis des gewünschten Zustands des Auslassventilsystem **104** zum Schließen des Behälters **100**, zum Entleeren des Behälters **100** oder zum Abziehen von Produkt aus dem Behälter **100** gesteuert werden.

[0072] Ein Steuerungs-Modulklassenobjekt ist gedacht, um zur Darstellung (und zur Konfigurierung) von individuellen Steuerungselementen oder Steuerungsmodulen innerhalb der Prozessanlage zu dienen. Mit einem Steuerungs-Modulklassenobjekt wird

eine bestimmte Form einer Steuerung bereitgestellt oder vorgegeben, die für eine Anlagenentität wie z.B. ein Ventil, eine Messeinrichtung etc., eine Ausrüstungskomponente oder eine ganze Einheit auszuführen ist. Allgemein gesagt, ein Steuerungs-Modulklassenobjekt stellt einen bestimmten Typ einer Steuerungsprogrammierung bereit, wie z.B. einen Satz kommunikativ miteinander verbundener Funktionsblöcke, die einige Steuerungsmodule definieren, die in einem Steuerungsgerät auszuführen sind, und die von Nutzen sind, um wiederholte Steuerungsaktivitäten innerhalb einer Prozessanlage auszuführen. In den meisten Fällen kann ein Steuerungs-Modulklassenobjekt eine generische Steuerungsstrategie bereitstellen, um eine einzelne Vorrichtung oder einen zusammenhängenden Satz von Vorrichtungen zu steuern.

[0073] Ein Display-Modulklassenobjekt ist zur Darstellung (und zur Konfigurierung) von Displayaktivitäten bestimmt, die während des Betriebs der Prozessanlage **10** für einen Nutzer wie z.B. einen Steuerungsbediener zur Darstellung gebracht werden. Ein Display-Modulklassenobjekt kann daher die Programmierung zur Erzeugung eines Displays eines bestimmten Typs innerhalb einer Bediener-Workstation **14** der [Fig. 1](#) spezifizieren, und außerdem die Programmierung, die auf einer oder mehreren der Workstations **14** (oder in anderen Vorrichtungen innerhalb der Prozessanlage **10**) ablaufen muss, damit das Display während des Anlagenbetriebs die korrekten Informationen betreffend die Anlage **10** erhält. Zu den Typen der Displayklassenmodule gehören beispielsweise Alarmdisplays, Konfigurationsdarstellungen, Funktionsdarstellungen, Diagnoseanzeigen etc. Selbstverständlich kann ein Display-Modulklassenobjekt ein Display bereitstellen, das jeden gewünschten Bereich physikalischer Elemente oder Entitäten innerhalb einer Prozessanlage repräsentieren kann bzw. diesen zugeordnet ist. Beispielsweise kann ein Display-Modulklassenobjekt Information über einen gesamten Bereich, eine Einheit, eine Ausrüstungskomponente, ein Steuerungselement oder eine beliebige Kombination dieser Elemente innerhalb der Prozessanlage **10** zur Darstellung bringen.

[0074] In [Fig. 5](#) sind anhand eines hierarchischen Graphen die Verbindungen zwischen verschiedenen Typen von Modulklassenobjekten wiedergegeben, wie sie in der Konfigurierungsapplikation **50** der [Fig. 1](#) eingesetzt sind, und außerdem die Beziehung zwischen den Modulklassenobjekten und den aus diesen Modulklassenobjekten entwickelten Modulobjekten. Im Graphen der [Fig. 5](#) sind die Modulklassenobjekte in senkrechter Richtung nach Modulklassentypen in einen Einheiten-Modulklassentyp **400**, einen Ausrüstungs-Modulklassentyp **402**, einen Steuerungs-Modulklassentyp **404** und einen Display-Modulklassentyp **406** unterteilt. Selbstverständlich können ebenso andere Typen von Modulklassenobjekt-

ten bereitgestellt oder verwendet werden, so dass die vier hierin dargestellten Typen lediglich als beispielhafte Modulklassentypen anzusehen sind. Die individuellen Modulklassenobjekte (wobei es sich um höhere Objekte handeln kann, die beispielsweise in einer objektorientierten Programmiersprache abgefasst sein können, und die in [Fig. 5](#) der Klarheit halber doppelt umrandet wiedergegeben sind) fallen unter die verschiedenen Modulklassentypen **400**, **402**, **404** und **406**. Insbesondere können viele verschiedene Einheiten-Modulklassenobjekte für verschiedene Einheiten oder Typen von Einheiten innerhalb der Prozessanlage **10** vorhanden sein. Beispielsweise kann ein Einheiten-Modulklassenobjekt **410** für einen Reaktor einen bestimmten Typ oder eine bestimmte Konfiguration eines Reaktors innerhalb der Prozessanlage **10** darstellen. Ebenso kann ein Modulklassenobjekt **412** für eine Verpackungseinheit einen bestimmten Typ oder eine bestimmte Konfiguration einer Verpackungseinheit innerhalb der Prozessanlage **10** repräsentieren, und ein Modulklassenobjekt **414** für eine Trocknereinheit kann einen bestimmten Typ oder eine bestimmte Konfiguration einer Trocknereinheit innerhalb der Prozessanlage **10** repräsentieren. Selbstverständlich können mehrere Modulklassenobjekte für Reaktoreinheiten vorhanden sein, um Reaktoren darzustellen, die sich untereinander in ihren physikalischen Eigenschaften unterscheiden. An dieser Stelle wird nicht versucht, alle unterschiedlichen Typen von Einheiten innerhalb einer Anlage aufzulisten, die durch ein Einheiten-Modulklassenobjekt repräsentiert oder modelliert werden können. Dem Fachmann ist jedoch klar, dass viele verschiedene Einheitentypen in verschiedenen Typen von Anlagen durch Einheiten-Modulklassenobjekte modelliert oder repräsentiert werden können.

[0075] Entsprechend können viele verschiedene Ausrüstungs-Modulklassenobjekte verwendet werden, um unterschiedliche Typen von Ausrüstungskomponenten innerhalb der Prozessanlage **10** darzustellen, zu modellieren und zu konfigurieren. Die in [Fig. 5](#) wiedergegebenen Beispiele enthalten ein Ausrüstungs-Modulklassenobjekt **416** für die Zumessung und ein Ausrüstungs-Modulklassenobjekt **418** für die Auslassventile, von denen jedes verschiedenen Typen von Ausrüstungskomponenten (und vorzugsweise mehrfach vorhandenen Ausrüstungskomponenten) innerhalb der Prozessanlage **10** zugeordnet ist. In der gleichen Weise können viele verschiedene Typen von Steuerungs-Modulklassenobjekten vorhanden sein, die in [Fig. 5](#) als ein Steuerungs-Modulklassenobjekt **422** für ein Absperrventil, ein Steuerungs-Modulklassenobjekt **424** für einen Pegelmesser und ein Steuerungs-Modulklassenobjekt **426** für einen Durchflussmesser dargestellt sind. Außerdem sind in [Fig. 5](#) Display-Modulklassenobjekte dargestellt, wie z.B. ein Display-Modulklassenobjekt **432** für Alarmmeldungen, ein Display-Modulklassenobjekt **434** für Ausgaben und ein Display-Modulklassen-

objekt **436** für Diagnosen. Selbstverständlich können beliebige andere gewünschte Einheiten-, Ausrüstungs-, Steuerungs- und Display-Modulklassenobjekte erzeugt und gemäß den hierin beschriebenen Prinzipien innerhalb der Konfigurierungsapplikation **50** der Prozessanlage **10** verwendet werden.

[0076] Jedes Modulklassenobjekt kann über Unterobjekte verfügen, die ihm zugeordnet oder zugehörig sind. Diese Unterobjekte können selbst wieder Modulklassenobjekte sein, oder sie können, wie in [Fig. 5](#) dargestellt, Modulobjekte sein, die als Instanzen aus den Modulklassenobjekten erzeugt wurden, zu denen sie gehören. Aus [Fig. 5](#) geht hervor, dass dem Einheiten-Modulklassenobjekt **410** für den Reaktor drei Reaktor-Modulobjekte mit den Bezeichnungen Reactor_01, Reactor_02 und Reactor_03 zugeordnet sind, wobei diese Reaktor-Modulobjekte den jeweiligen Reaktoren der [Fig. 1](#) entsprechen (d.h. mit diesen verknüpft sind). [Fig. 5](#) zeigt außerdem, dass das Ausrüstungs-Modulklassenobjekt **416** für die Zumessung fünf verschiedene Modulobjekte hat bzw. diese besitzt, die die Bezeichnungen Water1, Acid1, Acid2, Alkali1 und Alkali2 tragen. Ebenso ist beim Steuerungs-Modulklassenobjekt **422** für die Absperrventile gezeigt, dass es über Modulobjekte mit den Bezeichnungen Coarse_Valve1, Coarse_Valve2, Coarse_Valve3, Fine_Valve1, Fine_Valve2 und Fine_Valve3 verfügt. In entsprechender Weise können jedem der anderen Einheiten-, Ausrüstungs-, Steuerungs- und Display-Modulklassenobjekte der [Fig. 5](#) ein oder mehrere Modulobjekte zugeordnet sein. Der Klarheit halber sind diese Modulobjekte in [Fig. 5](#) jedoch nicht wiedergegeben.

[0077] Im Graphen der [Fig. 5](#) sind alle Einheiten-Modulobjekte Reactor_01, Reactor_02 und Reactor_03, die (Ausrüstungs-)Zumessmodulobjekte Acid1, Acid2, Alkali1, Alkali2 und Water1, die Steuerungs-Modulobjekte Coarse_Valve1, Coarse_Valve2, Coarse_Valve3, Fine_Valve1, Fine_Valve2 und Fine_Valve3 und die anderen Einheiten-, Ausrüstungs-, Steuerungs- und Display-Modulobjekte individuelle Objekte, die mit tatsächlichen Einheiten, Ausrüstungskomponenten, Steuerungsmodulen oder Displayapplikationen innerhalb der Prozessanlage **10** verknüpft sind. Zum Beispiel werden, da zahlreiche physikalisch vorhandene Säurezumessungen in der Anlage **10** eingesetzt sind, in der Konfigurierungsroutine zahlreiche Modulobjekte für Säurezumessungen erzeugt, so dass für jede der einzelnen in der Anlage **10** existierenden Säurezumessungen ein separates Modulobjekt für die Säurezumessung vorhanden ist. Dabei ist jedoch jedes der separaten Modulobjekte für die Zumessungen mit dem gleichen Modulklassenobjekt **416** für die Zumessungen verknüpft oder diesem zugehörig. Selbstverständlich zeigt der Graph der [Fig. 5](#) nur eine begrenzte Anzahl von Modulklassenobjekten und damit verbundenen Modulobjekten und es ver-

steht sich von selbst, dass andere Typen von Modulklassenobjekten bereitgestellt werden können, und dass jede gewünschte Anzahl von Modulobjekten aus jedem der verschiedenen Modulklassenobjekte erzeugt werden kann.

[0078] Jedes der Modulklassenobjekte der [Fig. 5](#) (und somit auch jedes der Modulobjekte der [Fig. 5](#)) kann als Teil des Objekts eine Definition oder eine Angabe zu den physikalischen oder logischen Prozesselementen beinhalten, die das Modul definieren oder bilden, und wenn dies gewünscht ist, zur Art und Weise, in der diese Prozesselemente sowohl physikalisch als auch logisch zusammenwirken, um Aktivitäten innerhalb der Prozessanlage **10** zu realisieren. Beispielsweise werden Einheiten-Modulklassenobjekte typischerweise eine Angabe aller physikalischen Elemente und Steuerungselemente enthalten, die zu der als die Einheit definierten Prozessentität gehören oder diese bilden. Das Einheiten-Modulklassenobjekt kann außerdem den jeweiligen Aufbau der einzelnen Teile festlegen und vorgeben, wie diese Teile physikalisch verknüpft sind, um als eine Einheit zu fungieren. Entsprechend wird ein Ausrüstungs-Modulklassenobjekt typischerweise die Steuerungsroutinen oder Steuerungsmodule enthalten, die zur Steuerung der als Ausrüstungskomponente definierten Entität verwendet werden, und Anweisungen, die die Steuerungsroutinen oder Steuerungsmodule verwenden, um die Art und Weise festzulegen, in der die Teile physikalisch oder logisch zusammenwirken, um beim Einsatz in der Anlage **10** als Ausrüstungskomponente zu fungieren. Ebenso wird jedes Steuerungs-Modulklassenobjekt, typischerweise in Form eines geeigneten Steuerungsalgorithmus, eine Steuerungsaktivität festlegen, die in der Anlage auszuführen ist. Dazu kann jedes Display-Modulklassenobjekt unter anderem den Aufbau eines Anzeigebildschirms und die darzustellende Information definieren, ebenso die zu erfassenden Daten und die Art und Weise der Datenaufbereitung, die ggf. für bestimmte Typen von Einheiten, Ausrüstungskomponenten, Anlagenbereichen oder für andere physikalische oder logische Entitäten innerhalb der Anlage **10** auf die gesammelten Daten anzuwenden ist.

[0079] Als Teil einer Modulklassendefinition kann ein Modulklassenobjekt andere Modulklassenobjekte angeben oder definieren, die einzubeziehen oder einzusetzen sind. Wenn dies der Fall ist, wird das aus dem Modulklassenobjekt erzeugte Modulobjekt andere Modulobjekte, die entsprechend der auf Modulklassenebene definierten Beziehungen aus anderen Modulklassenobjekten erzeugt wurden, für sich übernehmen, auf diese verweisen oder sie einbeziehen. Obwohl nicht zwingend erforderlich, können Einheiten-Modulklassenobjekte andere Einheiten-Modulklassenobjekte, Ausrüstungs-Modulklassenobjekte, Steuerungs-Modulklassenobjekte und Display-Modulklassenobjekte für sich übernehmen, während

Ausrüstungs-Modulklassenobjekte andere Ausrüstungs-Modulklassenobjekte, Steuerungs-Modulklassenobjekte und Display-Modulklassenobjekte für sich übernehmen können. Steuerungs-Modulklassenobjekte können andere Steuerungs-Modulklassenobjekte und Display-Modulklassenobjekte für sich übernehmen oder auf diese verweisen. Wenn dies gewünscht ist, können jedoch auch andere Beziehungen zwischen den Modulklassenobjekten verwendet werden. Diese Übernahmebeziehungen sind durch die großen Pfeile am unteren Rand von [Fig. 5](#) veranschaulicht, die deutlich machen, dass jedes der Display-Modulklassenobjekte in jedes der Steuerungs-, Ausrüstungs- und Einheiten-Modulklassenobjekte einbezogen werden kann bzw. dass diese darauf Bezug nehmen können, dass jedes der Steuerungs-Modulklassenobjekte in jedes der Ausrüstungs- und Einheiten-Modulklassenobjekte einbezogen werden kann bzw. dass diese darauf Bezug nehmen können, und dass jedes der Ausrüstungs-Modulklassenobjekte in jedes der Einheiten-Modulklassenobjekte einbezogen werden kann bzw. dass dieses darauf Bezug nehmen kann. Es ist klar, dass Modulklassenobjekte andere Modulklassenobjekte des gleichen Typs übernehmen können. Beispielsweise kann ein Einheiten-Modulklassenobjekt ein anderes Einheiten-Modulklassenobjekt als Teil seiner Definition übernehmen. In entsprechender Weise kann ein Ausrüstungs-Modulklassenobjekt ein anderes Ausrüstungs-Modulklassenobjekt einbeziehen, ein Steuerungs-Modulklassenobjekt kann ein anderes Steuerungs-Modulklassenobjekt einbeziehen, und ein Display-Modulklassenobjekt kann ein anderes Display-Modulklassenobjekt einbeziehen. Selbstverständlich kann ein Modulklassenobjekt, wenn dies gewünscht ist, ein anderes Modulklassenobjekt mehrfach verwenden oder übernehmen. Beispielsweise kann ein Einheiten-Modulklassenobjekt für Reaktoren das Ausrüstungs-Modulklassenobjekt für die Zumessung mehrere Male übernehmen oder verwenden, da die durch das Einheiten-Modulklassenobjekt für Reaktoren modellierten Reaktoren mehrfach ausgeführte Zumessungen aufweisen.

[0080] Es versteht sich auch von selbst, dass dann, wenn ein erstes Modulklassenobjekt ein zweites Modulklassenobjekt übernimmt oder verwendet, jedes aus dem ersten Modulklassenobjekt erzeugte oder abgeleitete Modulobjekt ein Modulobjekt übernimmt oder verwendet, das aus dem zweiten Modulklassenobjekt erzeugt oder abgeleitet wurde. Wenn somit das Einheiten-Modulklassenobjekt **410** für Reaktoren ein Modulklassenobjekt **416** für eine Zumessung als Element oder Teil für sich selbst verwendet, wird das Modulobjekt Reactor_01 eines der Modulobjekte für Zumessungen, z.B. das Modulobjekt Acid1, als Element oder Teil für sich selbst verwenden oder einbeziehen. Entsprechend wird dann, wenn ein Ausrüstungs-Modulklassenobjekt für Zumessungen ein Ausrüstungs-Modulklassenobjekt für Auslassventile

übernimmt oder einbezieht, ein aus dem Ausrüstungs-Modulklassenobjekt für Zumessungen erzeugtes Modulobjekt, das durch einen eindeutigen Namen, beispielsweise Totalizer_1, gekennzeichnet ist, ein Modulobjekt einbeziehen, das aus dem Ausrüstungs-Modulklassenobjekt für Auslassventile erzeugt wurde und durch einen eindeutigen Namen, beispielsweise Outlet_Valve_2, gekennzeichnet ist. Auf diese Weise spiegeln sich die Beziehungen zwischen den Modulklassenobjekten, die auf Modulklassenobjektebene definiert wurden, in den Modulobjekten wider, die aus diesen Modulklassenobjekten entwickelt oder erzeugt wurden. Diese Verbindung bzw. Bezugnahme zwischen Modulklassenobjekten (und somit den Modulobjekten) ermöglicht hohe Variabilität und Übertragbarkeit der Objekte bei den Konfigurierungsaktivitäten, so dass nach dem Erstellen eines grundlegenden Satzes von Modulklassenobjekten, beispielsweise Steuerungs- und Ausrüstungs-Modulklassenobjekten, komplexere Modulklassenobjekte, wie beispielsweise Einheiten-Modulklassenobjekte, in einfacher Weise durch Bezugnahme auf die grundlegenden Modulklassenobjekte erzeugt werden können. Obwohl Modulklassenobjekte auf andere Modulklassenobjekte verweisen oder diese verwenden können, ist es selbstverständlich möglich, dass sie zusätzlich hierzu oder stattdessen einfache Objekte oder Prozessentitäten definieren oder verwenden, beispielsweise Ventile, Sensoren etc., denen kein eigenes Modulklassenobjekt zugeordnet ist. Diese einfachen Objekte werden vollständig durch entsprechende Steuerungsroutinen innerhalb des Modulklassenobjekts selbst definiert.

[0081] Ein beispielhaftes Einheiten-Modulklassenobjekt **410** für Reaktoren ist in [Fig. 6](#) dargestellt, um eine Art und Weise der Beschreibung oder Definition der Entitäten zu verdeutlichen, die einem Einheiten-Modulklassenobjekt zugeordnet sind oder in diesem vorhanden sind. Wie [Fig. 6](#) zeigt, enthält das Einheiten-Modulklassenobjekt **410** für Reaktoren eine Anzeige für einen Behälter **500**, bei dem es sich um ein einfaches Objekt oder Element innerhalb der Prozessanlage **10** handelt, für das kein Modulklassenobjekt existiert. Der Behälter **500** ist durch gestrichelte Linien dargestellt, da keine Steuerungseingriffe oder untergeordnete Aktivitäten erforderlich sind, um Steuerungsaufgaben oder Maßnahmen zum Befüllen bzw. Entleeren des Behälters auszuführen. Der Behälter **500** ist somit ausschließlich deswegen einbezogen, um die Verbindungen mit den anderen, dem Einheiten-Modulklassenobjekt **410** für Reaktoren zugeordneten Objekten wiederzugeben. Das Einheiten-Modulklassenobjekt **410** für Reaktoren beinhaltet außerdem drei Zumessungen **501**, **502** und **510** mit den Bezeichnungen Acid, Alkali und Water, bei denen es sich um drei verschiedene Referenzen zu dem aus [Fig. 5](#) ersichtlichen Ausrüstungs-Modulklassenobjekt **416** für Zumessungen handelt. Das Modulklassenobjekt **510** für die Wasserzumessung

ist in einem Abschnitt des Einheiten-Modulklassenobjekts **410** wiedergegeben, der durch gestrichelte Linien abgetrennt ist, um anzuzeigen, dass es sich hierbei um ein gemeinsam genutztes Modulklassenobjekt handelt, und dass somit das Einheiten-Modulklassenobjekt **410** zusammen mit anderen Einheiten-Modulklassenobjekten die Kontrolle über dieses Objekt hat. Das Objekt Outlet **504** der [Fig. 6](#) ist eine Referenz zum Ausrüstungs-Modulklassenobjekt **418** für Auslassventile in der [Fig. 5](#), der Pegelmesser **505** ist eine Referenz zum Steuerungs-Modulklassenobjekt **424** für Pegelmesser in der [Fig. 5](#), und das Ventil water_in **503** ist eine Referenz zu einem Ventilobjekt, bei dem es sich um ein einfaches Ventilelement (das somit vollständig innerhalb des Einheiten-Modulklassenobjekts **410** definiert ist) handeln kann, oder um eine Referenz zu einem Steuerungs-Modulklassenobjekt für Ventile, das an beliebiger anderer Stelle der Konfigurierungsstrategie definiert sein kann. Die physikalischen Verbindungen zwischen den verschiedenen Entitäten oder Teilen des Einheiten-Modulklassenobjekts **410** für Reaktoren sind ebenfalls dargestellt, um die Verbindungen zwischen den verschiedenen Elementen zu definieren. Wie oben angemerkt, können das Einheiten-Modulklassenobjekt **410** oder andere Modulklassenobjekte von beliebigem Typ einfache Elemente enthalten, die vollständig innerhalb des Modulklassenobjekts definiert sind (einschließlich jedweder verbundenen generischen Steuerungsroutinen), und/oder Referenzen zu Modulklassenobjekten, die außerhalb des Modulklassenobjekts definiert sind.

[0082] Das Einheiten-Modulklassenobjekt **410** enthält außerdem zwei beispielhafte Display-Modulklassenobjekte mit den Bezeichnungen Reactor viewing display **520** und Reactor alarm display **522**, bei denen es sich um Referenzen zum Display-Modulklassenobjekt **434** für die Anzeigen und zum Display-Modulklassenobjekt **432** für die Alarmer der [Fig. 5](#) handelt. Diese Objekte definieren generische Displayaktivitäten zur Anzeige der Zustände (beispielsweise Füllstand im Behälter) und der Alarmer, die den Ausrüstungskomponenten oder Teilen der Reaktoreinheit zugeordnet sind, die im Einheiten-Modulklassenobjekt **410** für Reaktoren definiert ist. In entsprechender Weise kann das Einheiten-Modulklassenobjekt **410** andere Elemente enthalten, z.B. Phasenklassenobjekte, im Feld **524** als die Phasenklassenobjekte Dose, Mix, Drain und Flush bezeichnet (Dosieren, Mischen, Entleeren und Spülen), die jeweils eine generische Steuerungsroutine definieren, die auf die vom Einheiten-Modulklassenobjekt **410** definierte Einheit anzuwenden ist. Das Einheiten-Modulklassenobjekt kann keine oder mehrere Zuordnungen zu Phasenklassenobjekten aufweisen. Die Phasenklassenobjekte **524** können an beliebiger Stelle definiert und in jeder gewünschten Weise in das Einheiten-Modulklassenobjekt **410** importiert werden. In gewissem Sinn stellen die Phasenklassen **524** Anweisungen

oder Routinen dar, die auf eine vom Einheiten-Modulklassenobjekt **410** definierte Einheit angewendet werden können, um verschiedene Funktionen auszuführen, beispielsweise Befüllen der Einheit, Erhitzen der Einheit, Entleeren der Einheit, Reinigen der Einheit etc.

[0083] Darüber hinaus kann das Einheiten-Modulklassenobjekt **410** einen Speicher bzw. einen Bereich **526** enthalten, in dem Referenzen zu den Modulklassenobjekten abgelegt sind, die mittels der Konfigurierungsapplikation **50** (**Fig. 1**) aus diesem Einheiten-Modulklassenobjekt **410** erzeugt werden. Der Bereich **526** besteht im Wesentlichen aus einer Liste der Modulobjekte, die vom Einheiten-Modulklassenobjekt **410** erzeugt wurden und diesem zugehörig sind. (Selbstverständlich können diese Liste oder andere Angaben zu den zugehörigen Modulobjekten in jeder gewünschten Weise in der Workstation oder in der Konfigurierungsapplikation **50** gespeichert werden und müssen nicht physikalisch im Einheiten-Modulklassenobjekt **410** enthalten sein). In jedem Fall verfügt das Einheiten-Modulklassenobjekt **410** im Beispiel der **Fig. 6** über die Einheitenmodulobjekte Reactor_01, Reactor_1, Reactor_02 etc., die alle aus dem Einheiten-Modulklassenobjekt **410** für Reaktoren erzeugt wurden.

[0084] Das Einheiten-Modulklassenobjekt **410** enthält außerdem einen Satz von Methoden **530**, die vom Einheiten-Modulklassenobjekt **410** entweder während oder nach den Konfigurierungsmaßnahmen zu beachten sind. Die Methoden **530** können eine Methode zur Änderung der Steuerungsstruktur oder eine Applikation beinhalten, mit der Veränderungen am Einheiten-Modulklassenobjekt **410** automatisch auf alle dem Einheiten-Modulklassenobjekt **410** zugehörigen Modulobjekte **526** übertragen werden. Andere Methoden können die Sicherheit des Systems betreffen, indem sie die Sicherheit bzw. die Zugriffsberechtigungen bezüglich des Einheiten-Modulklassenobjekts **410** und/oder bezüglich eines jeden der diesem zugehörigen Einheitenmodulobjekte **526** kontrollieren, oder Methoden, die es einem Nutzer oder einem Konfigurierungstechniker ermöglichen, Änderungs- und/oder Sicherheitsparameter für das Modulklassenobjekt oder für jedes der aus diesem erzeugten Modulobjekte zu spezifizieren. Selbstverständlich können die verschiedenen Methoden **530** jede andere Prozedur, die auf das oder in Bezug auf das Einheiten-Modulklassenobjekt **410** wirkt, ausführen.

[0085] Wenn dies gewünscht wird, kann das Einheiten-Modulklassenobjekt **410** die Art und Weise steuern, in der Änderungen, die am Modulklassenobjekt **410** vorgenommen wurden, auf die Einheitenmodulobjekte **526** übertragen werden, und ebenso die Art und Weise, in der die Zugriffssicherheit bzw. der Sicherheitszugriff in den Einheitenmodulobjekten **526**

realisiert ist. Ein Verfahren zur Bereitstellung dieser Funktionalität besteht im Setzen eines oder mehrerer Flags oder Parameter innerhalb des Einheiten-Modulklassenobjekts **410**, um die Art und Weise festzulegen, in der Änderungen auf die Einheitenmodulobjekte **526** zu übertragen sind, und wie deren Zugriffssicherheit zu handhaben ist. Insbesondere können ein oder mehrere Parameter betreffend die Übertragung von Änderungen gesetzt werden, um zu spezifizieren, ob am Einheiten-Modulklassenobjekt **410** vorgenommene Änderungen automatisch an eines oder mehrere der Einheitenmodulobjekte **526** zu übertragen sind. Die Parameter zur Steuerung der Änderungsübertragung können in den Einheitenmodulobjekten **526** gespeichert werden und für das gesamte Einheitenmodulobjekt oder für ein Unterelement auf Unterelementbasis spezifizieren, ob am Einheiten-Modulklassenobjekt vorgenommene Änderungen an die Einheitenmodulobjekte weitergegeben werden sollen. Beispielsweise kann das Einheiten-Modulklassenobjekt **410** einen globalen Änderungsparameter **534** (mit "C" bezeichnet) enthalten, der in jedem Einheitenmodulobjekt gesetzt werden kann, das aus dem Einheiten-Modulklassenobjekt **410** erzeugt wurde, um die automatische Weitergabe von am Einheiten-Modulklassenobjekt **410** vorgenommenen Änderungen an das Einheitenmodulobjekt zu sperren oder freizugeben. Ebenso kann ein jedes Unterelement bzw. jeder Block, beispielsweise die Blöcke **501-505**, **510**, **520** und **522**, einen Änderungsparameter **536** enthalten, der ausschließlich für den jeweiligen Block angibt, ob Änderungen, die im Einheiten-Modulklassenobjekt **410** an diesem Block vorgenommen wurden, an das Einheitenmodulobjekt weitergegeben sind. Selbstverständlich können die verschiedenen Blöcke eines Einheitenmodulobjekts unterschiedlich gesetzt sein, so dass beispielsweise Änderungen, die am Acid block (Säureblock) **501** des Einheiten-Modulklassenobjekts **410** vorgenommen werden, an den korrespondierenden Säureblock eines bestimmten unter den Modulobjekten **526** weitergegeben werden, jedoch derart, dass Änderungen, die am Alkali block (Alkaliblock) **502** des Einheiten-Modulklassenobjekts **410** vorgenommen werden, nicht an den Alkaliblock dieses bestimmten Einheitenmodulobjekts weitergegeben werden. Darüber hinaus können die verschiedenen Einheitenmodulobjekte, die aus einem Einheiten-Modulklassenobjekt erzeugt wurden, jeweils unterschiedlich gesetzte Änderungsparameter aufweisen, so dass Änderungen am Alkaliblock **502** innerhalb des Einheiten-Modulklassenobjekts **410** an den korrespondierenden Alkaliblock eines ersten der Einheitenmodulobjekte **526** weitergegeben werden, nicht aber an den korrespondierenden Alkaliblock eines zweiten der Einheitenmodulobjekte **526**. Selbstverständlich kann die Verfahrensmethode des Einheiten-Modulklassenobjekts **410** zur Veränderung der Steuerungsstruktur auf die Änderungsparameter der Einheitenmodulobjekte **526** zugreifen und sie nutzen, um eine Änderung an die-

sen Objekten vorzunehmen bzw. die Änderung zu unterlassen, wenn die betreffende Änderung am Einheiten-Modulklassenobjekt **410** vorgenommen wird.

[0086] In entsprechender Weise kann das Einheiten-Modulklassenobjekt **410** einen oder mehrere Sicherheitsparameter aufweisen, die die Art und Weise bestimmen, in der die Sicherheit oder der Zugriffsschutz in jedem der Einheitenmodulobjekte **526** kontrolliert wird. Das Einheiten-Modulklassenobjekt **410** kann einen globalen Sicherheitsparameter **538** enthalten (mit "S" bezeichnet), der jede gewünschte Höhe der Zugriffssicherheit zum gesamten Einheitenmodulobjekt für Reaktoren, das aus dem Einheiten-Modulklassenobjekt **410** für Reaktoren erzeugt wurde, einstellen kann, und/oder es kann verschiedene Sicherheitsparameter **540** für jedes Unterelement des Einheiten-Modulklassenobjekts **410** enthalten, beispielsweise für jeden der Blöcke **501-505**, **510**, **520**, **522** etc., die jeweils auf blockweiser Basis eine Höhe der Zugriffssicherheit für jeden dieser Blöcke spezifizieren. Der globale Sicherheitsparameter **538** kann als Zugangssperre wirken, die das Einheiten-Modulklassenobjekt für den Zugriff durch alle Nutzer sperrt, mit Ausnahme solcher, die über eine im Voraus bestimmte Höhe der Zugriffsberechtigung verfügen. Selbstverständlich können die Sicherheitsparameter **538** und **540** jede beliebige Höhe der Zugriffssicherheit entsprechend mehreren Stufen spezifizieren, beispielsweise kein Zugriff, begrenzter Zugriff, Zugriff für bestimmte Nutzergruppen oder einzelne Nutzer etc., und die Sicherheitshöhen können für die verschiedenen Blöcke und die verschiedenen Einheitenmodulobjekte, die aus dem gleichen Einheiten-Modulklassenobjekt erzeugt wurden, unterschiedlich gesetzt werden. Wenn gewünscht, können Teile der Sicherheitsmaßnahmen Verschlüsselungen für eine oder mehrere Methoden oder Algorithmen beinhalten, die dem Einheiten-Modulklassenobjekt zugeordnet sind.

[0087] Es versteht sich von selbst, dass die Änderungs- und Sicherheitsparameter des Einheiten-Modulklassenobjekts **410** mit Standardwerten vorbelegt werden können, und dass die entsprechenden Änderungs- und Sicherheitsparameter jedes Einheitenmodulobjekts **526**, das aus dem Einheiten-Modulklassenobjekt **410** erzeugt wurde, bei der Erzeugung die betreffenden Standardwerte übernehmen können. Jedoch können die Standardwerte für die Änderungs- und Sicherheitsparameter in den Einheitenmodulobjekten **526** auch individuell verändert werden (durch einen Nutzer mit entsprechender Zugriffsberechtigung), nachdem diese Einheitenmodulobjekte erzeugt wurden. Auch wenn die Änderungs- und Sicherheitsparameter hierin in Bezug auf ein Einheiten-Modulklassenobjekt für Reaktoren erläutert werden, können entsprechende Änderungs- und Sicherheitsparameter auch in anderen Typen von Einheiten-Modulklassenobjekten sowie in allen gewünsch-

ten Typen von Ausrüstungs-Modulklassenobjekten, Steuerungs-Modulklassenobjekten, Display-Modulklassenobjekten etc. bereitgestellt werden.

[0088] Wenn gewünscht, kann das Einheiten-Modulklassenobjekt **410** Referenzen wie beispielsweise URLs oder andere Verweise auf Dokumentation enthalten, die für das Einheiten-Modulklassenobjekt gespeichert oder diesem zugeordnet ist, einschließlich Dokumentation, die der dem Einheiten-Modulklassenobjekt **410** zugehörigen Einheit oder beliebigen Unterelementen der Einheit zugeordnet ist. Solche Verweise sind in [Fig. 6](#) als Referenzen **549** dargestellt.

[0089] Als weiteres Beispiel eines Modulklassenobjekts ist in [Fig. 7](#) ein Ausrüstungs-Modulklassenobjekt **416** für Zumessungen dargestellt. Das Ausrüstungs-Modulklassenobjekt **416** für Zumessungen enthält Steuerungs-Modulklassenobjekte mit den Bezeichnungen Coarse_Valve **550** und Fine_Valve **552** (beides Steuerungs-Modulklassenobjekte für Öffnen/Schließen) und ein Steuerungs-Modulklassenobjekt mit der Bezeichnung Flow_Meter **554** (bei dem es sich um ein Steuerungs-Modulklassenobjekt für Durchflussmesser handelt) sowie die angedeuteten Verbindungen zwischen diesen Elementen. Außerdem enthält das Ausrüstungs-Modulklassenobjekt **416** für Zumessungen Referenzen zu Display-Modulklassenobjekten einschließlich eines Display-Modulklassenobjekts **560** für Zumessungsalarme, und zu einem oder mehreren Algorithmen **564**, die im Ausrüstungs-Modulklassenobjekt **416** implementiert werden können. Auch wenn die dargestellte Auflistung der Algorithmen **564** eine Anweisung Totalize Fast und eine Anweisung Totalize_Accurate enthält, können beliebige andere Anweisungen oder Algorithmen ebenso einbezogen oder verwendet werden. Des Weiteren kann der Anweisungsalgorithmus **564**, der mit einem Ausrüstungs-Modulklassenobjekt verbunden oder in dieses einbezogen ist, jede beliebige Form haben, wobei es sich beispielsweise um anweisungsgesteuerte Algorithmen (CDAs – command driven algorithms), zustandsgesteuerte Algorithmen (SDAs – state driven algorithms), sequentielle Folgesteuerungsalgorithmen (SFC – sequential flow chart algorithms), Funktionsblockalgorithmen (FBD – function block diagram algorithms), Phasenalgorithmen etc. handeln kann. Allgemein gesagt, sind jedoch alle der Algorithmen **564** von einem bestimmten Typ, wie beispielsweise CDA oder SDA. Selbstverständlich können die Algorithmen **564** in jeder gewünschten Sprache oder Programmierumgebung geschrieben werden, beispielsweise den C- bzw. C⁺⁺-Umgebungen, in jeder beliebigen sequentiellen Folgesteuerungsumgebung, einer Funktionsblockumgebung etc.

[0090] Das Ausrüstungs-Modulklassenobjekt **416** für Zumessungen beinhaltet ferner eine Liste oder ei-

nen Speicher mit Angaben (und falls erforderlich Kommunikationspfaden) zu einem Satz zugehöriger Ausrüstungsmodulobjekte **566** mit der Bezeichnung Acid1, Acid2, Alkali1, Alkali2, Water_Hdr1, etc., die aus dem Ausrüstungs-Modulklassenobjekt **416** erzeugt wurden. Das Ausrüstungs-Modulklassenobjekt **416** für Zumessungen umfasst ferner einen Satz Methoden **570** einschließlich einer Methode zur Veränderung der Steuerungsstruktur, die gemeinsam mit dem globalen Änderungsparameter **572** und/oder den Änderungsparametern **574** auf Objektbasis verwendet werden kann, um die Übertragung von Änderungen auf die Ausrüstungsmodulobjekte **566** zu steuern. Das Ausrüstungs-Modulklassenobjekt **416** für Zumessungen beinhaltet ferner einen globalen Sicherheitsparameter **580** sowie einen Sicherheitsparameter **582** auf Objektbasis. Die Änderungs- und Sicherheitsparameter **572**, **574**, **580** und **582** fungieren generell wie beschrieben bezüglich der Änderungs- und Sicherheitsparameter des Einheiten-Modulklassenobjekts **410** der [Fig. 6](#) und können auf jedes der Elemente des Steuerungs-Modulklassenobjekts einschließlich der Anweisungen **564** angewandt werden. Des Weiteren können entweder nur die Änderungs- oder die Sicherheitsparameter oder beide auf jeder gewünschten anderen als der globalen Ebene oder auf Objekt- oder Elementebene innerhalb der Modulklassenobjekte (und damit innerhalb der daraus erzeugten Modulobjekte) bereitgestellt werden, wie hierin ausführlicher beschrieben. Wenn gewünscht, können die Änderungs- und Sicherheitsparameter eines Modulklassenobjekts oder eines daraus erzeugten Modulobjekts unter Verwendung einer Gruppenkonfigurierungsapplikation oder eines Gruppenkonfigurierungsprogramms, wie z.B. einem Tabellenkalkulationsprogramm, gesetzt oder konfiguriert werden, so dass ein Konfigurierungstechniker die verschiedenen Änderungs- und Sicherheitsparameter für das gesamte Modulklassenobjekt oder ein Modulobjekt auf einfache und unkomplizierte Weise setzen oder vorgeben kann.

[0091] Wenn gewünscht, kann das Ausrüstungs-Modulklassenobjekt **416** Referenzen wie beispielsweise URLs oder andere Verweise auf Dokumentation enthalten, die für das Ausrüstungs-Modulklassenobjekt gespeichert oder diesem zugeordnet ist, einschließlich Dokumentation, die der dem Ausrüstungs-Modulklassenobjekt **416** zugehörigen Ausrüstungskomponente oder beliebigen Unterelementen der Ausrüstungskomponente zugeordnet ist. Solche Verweise sind in [Fig. 7](#) als Referenzen **599** dargestellt. Ebenso kann jeder der Algorithmen des Ausrüstungs-Modulklassenobjekts **416**, beispielsweise jeder der Algorithmen **564**, verschlüsselt sein oder über einen zugeordneten Sicherheitsparameter verfügen, der diese Algorithmen **564** verschlüsselt oder entschlüsselt. Eine solche Verschlüsselung oder Entschlüsselung kann, wenn gewünscht, durch eine Methode **570** erfolgen.

[0092] Um die Prozessanlage **10** zu konfigurieren, legt ein Konfigurierungstechniker nach Bedarf die erforderlichen Einheiten-, Ausrüstungs-, Steuerungs- und Display-Modulklassenobjekte in einer der Konfigurierungsapplikation **50** ([Fig. 1](#)) zugeordneten Bibliothek an. Der Konfigurierungstechniker kann, wenn gewünscht, mit den Entitäten mit jeweils geringerem Umfang beginnen, beispielsweise den Steuerungs- und Displaymodulklassen, und anschließend die Modulklassenobjekte für die Entitäten mit größerem Umfang, wie z.B. die Ausrüstungs- und Einheiten-Modulklassenobjekte entwickeln, die die Entitäten mit geringerem Umfang verwenden oder auf diese Bezug nehmen können. Danach kann der Konfigurierungstechniker die Modulklassenobjekte wie erforderlich auswählen oder spezifizieren, um tatsächliche Modulobjekte entsprechend den gewählten Modulklassenobjekten für jede der Prozessentitäten innerhalb der Anlage zu erzeugen. Bei der Konfigurierung mehrfach vorhandener Ausrüstungskomponenten erzeugt der Konfigurierungstechniker ein Modulobjekt für jede Instanz der mehrfach vorhandenen Ausrüstungskomponenten aus dem jeweils gleichen Modulklassenobjekt. Somit kann der Konfigurierungstechniker für die Reaktoren der [Fig. 1](#) ein Einheiten-Modulklassenobjekt für Reaktoren erzeugen (was die Erzeugung der anderen Modulklassenobjekte beinhaltet, auf die das Einheiten-Modulklassenobjekt für Reaktoren Bezug nimmt, wenn solche anderen Modulklassenobjekte nicht bereits in der Bibliothek vorhanden sind). Danach kann der Konfigurierungstechniker den Prozess konfigurieren, indem er für jeden der Reaktoren Reactor_01, Reactor_02 und Reactor_03 der [Fig. 1](#) ein Einheitenmodulobjekt für Reaktoren aus dem Einheiten-Modulklassenobjekt für Reaktoren erzeugt.

[0093] Nach der Erzeugung eines oder mehrerer Modulobjekte aus einem Modulklassenobjekt kann der Konfigurierungstechniker die Modulobjekte (einschließlich der Unterobjekte oder der durch Referenz einbezogenen Objekte) mit den spezifischen Ausrüstungskomponenten der Anlage verbinden. Da die Einheitenmodulobjekte einem einzigen Einheiten-Modulklassenobjekt zugeordnet sind, können die Aliasnamen, Parameter und andere Variablen für die verschiedenen Einheitenmodulobjekte gemeinsam mittels beispielsweise einer Gruppenverarbeitungsapplikation, wie z.B. einer Tabellenkalkulation spezifiziert werden. Selbstverständlich spezifiziert der Konfigurierungstechniker eigentlich bei der Verknüpfung der Modulobjekte mit bestimmten Ausrüstungskomponenten die Steuervariablen und Kommunikationspfadnamen, die von den Steuerungsroutinen oder Steuerungsmodulen in den Steuerungsgeräten **12** genutzt werden, um Steuerungsaktivitäten im Betrieb der Prozessanlage **10** auszuführen, oder die Displayvariablen, die von den Displayroutinen im Betrieb der Prozessanlage **10** beispielsweise in den Workstations **14** verwendet werden. Nachdem die

Maßnahmen zur Verknüpfung abgeschlossen sind, kann der Konfigurierungstechniker die gebundenen Steuerungsroutinen in die Steuerungsgeräte **12** und die gebundenen Displayroutinen in die Workstations **14** herunterladen.

[0094] **Fig. 8 – Fig. 16** zeigen Bildschirmdarstellungen, die von der Konfigurierungsapplikation **50** der **Fig. 1** erzeugt werden können, wenn der Konfigurierungstechniker bei der Konfigurierung Modulklassenobjekte erzeugt und verwendet, um die Prozessanlage **10** zu konfigurieren. Wie zu erkennen ist, enthalten die meisten der Bildschirmdarstellungen der **Fig. 8 – Fig. 16** ein Explorer-Fenster im linken Bildschirmteil, in dem anhand einer Baumstruktur die Konfiguration der Prozessanlage **10** wiedergegeben ist. Ebenso enthalten die meisten der Bildschirmdarstellungen der **Fig. 8 – Fig. 16** ein oder mehrere Informationsfenster im rechten Bildschirmteil. Diese Informationsfenster stellen weitere Information über im Explorer-Fenster ausgewählte Elemente bereit. Die Information, die einem Nutzer zur Anzeige gebracht oder die von einem Nutzer in den Informationsfenstern geändert werden kann, kann mittels der Steuerungs- und Sicherheitsparameter **534, 536, 538, 540, 572, 574, 580** und **582** der **Fig. 6** und **Fig. 7** festgelegt oder kontrolliert werden, die für jedes der verschiedenen Modulklassenobjekte oder deren Unterelemente gesetzt wurden. Folglich kann ein bestimmtes Element des Explorer-Fensters anhand der Sicherheits- und Steuerungsparameter, die im Modulklassenobjekt gesetzt sind, für einen Nutzer einsehbar und/oder veränderbar dargestellt und an die im Explorer-Fenster aufgeführten Modulobjekte übertragen werden. Wie weiter oben ausgeführt, kann Information selbstverständlich als grundsätzlich versteckt, als nur nach entsprechender Passwort- oder Sicherheitscodeeingabe eines Nutzers einsehbar oder veränderbar, als stets einsehbar aber unveränderlich, als stets einsehbar und veränderbar oder unter jeder beliebigen anderen Kombination dieser oder anderer Sicherheits- und Änderungsparameter gehalten werden. Wenn gewünscht, kann außerdem die Einsehbarkeit oder Veränderbarkeit eines Elements im Explorer-Fenster durch Hervorhebung, Abschattierung, Farbgebung oder beliebige andere Techniken angezeigt werden, um dem Nutzer Hinweise zu geben, welche Elemente in detaillierterer Form dargestellt oder verändert werden können.

[0095] In **Fig. 8** enthält eine Bildschirmdarstellung **600** einen Abschnitt eines Explorer-Konfigurationsfensters **602**, der auf der linken Seite des Display gezeigt ist. Der Abschnitt des Explorer-Fensters **602** zeigt eine Bibliothek, in der eine Vielzahl von Modulklassenobjekten gespeichert ist, einschließlich Einheiten-Modulklassenobjekten **604**, Ausrüstungs-Modulklassenobjekten **606** und Steuerungs-Modulklassenobjekten **608**. Ein Einheiten-Modulklassenobjekt **610** für Reaktoren (das dem Einheiten-Modulklas-

senobjekt **410** für Reaktoren der **Fig. 6** entsprechen kann) ist in der Einheiten-Modulklassenbibliothek **604** gespeichert und enthält Angaben zu einer Vielzahl von Unterelementen einschließlich die Phasenklassenobjekte Dose, Mix, Drain und Flush und die Ausrüstungs-Modulklassenobjekte Acid, Alkali, Water und Outlet, Steuerungs-Modulklassenobjekte Water_In und Level_Meter und weitere gewünschte Elemente. Somit enthält entsprechend der Definition in der Einheiten-Modulklassenbibliothek **604** das Einheiten-Modulklassenobjekt **610** für Reaktoren Angaben zu Phasenklassen sowie Angaben zu Ausrüstungs-Modulklassenobjekten und Steuerungs-Modulklassenobjekten. Da im Bildschirm **600** das Einheiten-Modulklassenobjekt **610** für Reaktoren ausgewählt ist, sind dessen Elemente im rechten Teil **612** des Bildschirms **600** in detaillierterer Form aufgelistet.

[0096] Des Weiteren enthält die Ausrüstungs-Modulklassenbibliothek **606** ein Ausrüstungs-Modulklassenobjekt **614** für Zumessungen (das dem Ausrüstungs-Modulklassenobjekt **416** für Zumessungen in **Fig. 7** entsprechen kann) und ein Ausrüstungs-Modulklassenobjekt Reactor_Outlet **616**. Das Ausrüstungs-Modulklassenobjekt **614** für Zumessungen enthält drei verschiedene Abschnitte eines Algorithmus (z.B. einen der Algorithmen **564** der **Fig. 7**) mit den Bezeichnungen Command_00001, Command_00002 und Command_00003. Das Modulklassenobjekt **614** enthält außerdem Referenzen zu Steuerungsmodulobjekten mit den Bezeichnungen Coarse_Valve und Fine_Valve (bei denen es sich um Steuerungs-Modulklassenobjekte für Absperrelemente handelt) und Flow_Meter (bei dem es sich um ein Steuerungs-Modulklassenobjekt für Durchflussmesser handelt). Weiter enthält das Ausrüstungs-Modulklassenobjekt Reactor_Outlet **616** einen zustandsgesteuerten Algorithmus, der verschiedene Zustände mit den Bezeichnungen State_00001, State_00002 und State_00003, Module mit den Bezeichnungen Target, Drive, Monitor und Readback, und Steuerungsmodulobjekte für Ventile mit den Bezeichnungen Outlet, Drain und Product umfasst (wobei es sich bei den letzteren um Angaben zu oder um Referenzen auf Modulblöcke des Typs Steuerungs-Modulklassenobjekt für Absperrelemente handeln kann, die die Bezeichnungen Outlet, Drain und Product tragen, oder wobei diese auch nur einfache Objekte sein können). Bei den anweisungs- und zustandsgesteuerten Algorithmen, die den Modulklassenobjekten Totalizer **614** und Reactor_Outlet **616** zugeordnet sind, kann es sich um beliebige, wie gewünscht aufgebaute Routinen handeln, die auf Steuerungsmodulobjekte innerhalb des Ausrüstungs-Modulklassenobjekts Bezug nehmen können, die in Verbindung mit den betreffenden Anweisungen verwendet werden sollen. Insbesondere können die CDA- und SDA-Anweisungsalgorithmen der Ausrüstungs-Modulklassenobjekte Ausdrücke oder Maß-

nahmen enthalten, die auf Steuerungsmodule (oder andere Ausrüstungsmodule) Bezug nehmen, indem sie die Bezeichnungen der Modulblöcke übernehmen, die diesen Modulen entsprechen, um anzugeben, welche Ausrüstungskomponente bei der Ausführung des Algorithmus zu manipulieren ist. Die Verwendung des Namens des Modulblocks, der dem Steuerungsmodul (oder einem anderen Ausrüstungsmodul) innerhalb dieser Algorithmen entspricht, spezifiziert das Steuerungsmodulobjekt, auf das vom Ausrüstungsmodulobjekt, in dem sich der Algorithmus befindet, Bezug genommen wird bzw. das diesem zugeordnet ist, und die spezifischen Bezeichnungen werden eingebunden oder instantiiert, wenn das Ausrüstungsmodulobjekt aus dem Ausrüstungs-Modulklassenobjekt erzeugt wird.

[0097] [Fig. 9](#) zeigt einen Konfigurationsbildschirm **620**, in dem das Ausrüstungs-Modulklassenobjekt **614** für Zumessungen expandiert dargestellt ist, um die damit verbundene Ablauflogik aufzuzeigen, die die anweisungsgesteuerten Algorithmen mit den Bezeichnungen `Command_00002`, `Command_00003` und `Command_00099` enthält. Im rechten Teil **622** des Bildschirms **620** sind detailliertere Informationen zu den Steuerungsmodulobjekten der Ablauflogik enthalten. Insbesondere handelt es sich bei `Command_00002` um eine Routine für die exakte Zumessung, die für die Steuerung des exakten oder feinen Stroms der Zumessung genutzt wird. `Command_00003` ist eine Routine für die Grobzumessung, die genutzt wird, um den schnellen Strom der Zumessung zu steuern bzw. durch diesen zu befüllen, und `Command_00099` ist eine Rücksetzroutine, die die Zumessung zurücksetzt. Der Block mit der Bezeichnung Monitor im Bildschirm **620** ist ein der Ablauflogik zugeordneter Funktionsblock, der Überwachungsaktivitäten bereitstellt. Wie aus [Fig. 8](#) und [Fig. 9](#) hervorgeht, hat jedes der Elemente innerhalb der Modulklassenobjekte der Bibliotheken **604**, **606** und **608** Unterobjekte und Unterelemente, und in einigen oder allen Fällen nehmen die Unterelemente Bezug auf oder übernehmen Steuerungsmodulobjekte oder Displayroutinen, die im Betrieb der Prozessanlage **10** von einem Computer wie beispielsweise einem Steuerungsgerät oder einer Workstation auszuführen sind. Die Steuerungs- und Displayroutinen und die sonstigen Bezugnahmen auf Ausrüstungskomponenten innerhalb der Modulklassenobjekte sind jedoch generischer Natur, so dass sie mit keinen bestimmten tatsächlichen Ausrüstungskomponenten der Prozessanlage **10** verknüpft oder verbunden sind.

[0098] [Fig. 10](#) zeigt eine weitere Bildschirmdarstellung **630**, die die Details der Routine `Command_00002` in der Ablauflogik des Ausrüstungs-Modulklassenobjekts Totalizer **614** aufzeigt. Wie in einem Pop-up-Fenster **632**, das von einem Editor **634** zur Programmierung sequentieller Funktionspläne erzeugt wird, zu erkennen ist, enthält die

Routine `Command_00002` eine Folge von Schritten, wobei jeder Schritt eine Folge von Maßnahmen umfasst. Nur die erste Maßnahme (A1) des ersten Schritts (S1) der Routine `Command_00002` ist im Fenster **632** dargestellt. Dabei ist anzumerken, dass die erste Maßnahme des ersten Schritts der Routine `Command_00002` den Parameter `Reset` des Steuerungsmodulobjekts `Flow_Meter` auf "True" setzt, und anschließend auf Basis der Zustände der Variablen `SP` den Parameter `Target_State` des Steuerungsmodulobjekts (oder Modulblocks) `Coarse_Valve` auf "Valve:Open" und den Parameter `Target_State` des Steuerungsmodulobjekts (oder Modulblocks) `Fine_Valve` auf "Valve:Open" setzt. Es versteht sich von selbst, dass dieser Abschnitt der Routine `Command_00002` auf die Steuerungsmodulobjekte (oder Modulblöcke) `Flow_Meter`, `Coarse_Valve` und `Fine_Valve` Bezug nimmt, die an diesem Punkt in generischer Form dem Ausrüstungs-Modulklassenobjekt Totalizer zugeordnet sind, dass aber die Routine `Command_00002` jedes tatsächlichen Ausrüstungsmodulobjekts für Zumessungen, das aus dem Ausrüstungs-Modulklassenobjekt für Zumessungen erzeugt wurde, auf den tatsächliche Namen des Steuerungsmodulobjekts Bezug nehmen wird, das mit dem oder als Teil des Ausrüstungsmodulobjekts für Zumessungen erzeugt wurden.

[0099] [Fig. 11](#) ist eine Bildschirmdarstellung **640**, die Details eines Steuerungsalgorithmus zeigt, der der Phasenklasse Dose **642** zugeordnet ist, die vom Einheiten-Modulklassenobjekt Reactor **610** der [Fig. 8](#) verwendet werden oder auf die dieses Bezug nehmen kann. Insbesondere enthält die Phasenklasse Dose **642** einen Satz verschiedener allgemeiner Steuerungsmaßnahmen, zu denen Abort, Fail, Hold, Restart und Run gehören. Die mit der Maßnahme Run verbundene Logik (als `Run_Logic` bezeichnet) umfasst eine Folge von Schritten, die im Bildschirmabschnitt **644** als Step 1 (S1), Step 2 (S2) und Step 3 (S3) bezeichnet sind, außerdem zwei Übergänge (Transitions) T1 und T2 und einen Abschluss (Termination). Ein Editor für sequentielle Funktionspläne **646** liefert eine grafische Darstellung der Schritte und der Übergänge in Form eines Blockdiagramms. Ein Pop-up-Bildschirm **648** bringt die tatsächliche, mit Action 1 (A1) von Step 1 (S1) verbundene Steuerlogik zur Ausgabe. Insbesondere enthält die Ablauflogik im Fenster **648**, bei dem es sich um einen Logikeditor handelt, die Maßnahmen für die Zuteilung des gemeinsam genutzten Wasserheaders für den Reaktor, das Öffnen des Ventils `water_in` und das anschließende Rücksetzen aller dem Reaktor zugehörigen Zumessungen.

[0100] Da die Phasenklasse Dose unabhängig vom Reaktor geschrieben ist, verwendet sie Aliasnamen für den Bezug auf Ausrüstungskomponenten, die zur Laufzeit oder zu einem früheren Zeitpunkt spezifiziert werden, die aber selbstverständlich nicht im Einhei-

ten-Modulklassenobjekt für Reaktoren spezifiziert werden können. Jedoch beziehen sich die Referenzen auf tatsächliche Vorrichtungen oder andere Prozesselemente, die im Steuerungsalgorithmus **648** verwendet werden, auf solche Vorrichtungen, die innerhalb des gleichen Modulklassenobjekts vorliegen, in diesem Fall im Einheiten-Modulklassenobjekt für Reaktoren. In der Steuerungsroutine **648** werden Aliasnamen mit dem Symbol # versehen, so dass #WATER_IN# ein Aliasname für das Steuerungsmodul water_in im gleichen Modul ist, d.h. im Einheiten-Modulklassenobjekt **610**. In gleicher Weise beziehen sich die Bezeichnungen #WATER#, #ACID# und #ALKALI# auf die Zumessungs-Modulblöcke Water, Acid und Alkali im Einheiten-Modulklassenobjekt Reactor **610**. Selbstverständlich ist klar, dass die Ausrüstungs-Modulklassenobjekte für Zumessungen und Reaktorauslass und das Steuerungs-Modulklassenobjekt für Absperrventile dem Einheiten-Modulklassenobjekt für Reaktoren durch entsprechende Ausrüstungsmodulobjekte für Zumessungen und Reaktoren (mit beispielsweise den Bezeichnungen Alkali1, Acid1 etc.) und Steuerungsmodulobjekte für Absperrventile (mit beispielsweise den Bezeichnungen Coarse_Valve1, Fine_Valve1, Outlet1 etc.) zugeordnet sind und auf diese Weise logische Teile des betreffenden Einheitenklassenobjekts werden.

[0101] [Fig. 12](#) zeigt eine Bildschirmdarstellung **650**, in der das Ausrüstungs-Modulklassenobjekt Reactor_Outlet **616** detaillierter dargestellt ist. Die dem Ausrüstungs-Modulklassenobjekt Reactor_Outlet **616** zugeordnete Run_Logic ist im Bildschirmabschnitt **652** gezeigt und umfasst die Elemente Drive, Monitor und Readback, und insbesondere einen zustands-gesteuerten Algorithmus mit vier separaten Zuständen, die zur Steuerung der Ventile dienen, die dem Ausrüstungs-Modulklassenobjekt Reactor_Outlet **616** zugeordnet sind. Der rechte Bildschirmabschnitt **654** enthält Einzelheiten zu den Elementen Drive mit den Bezeichnungen Drain SP, Outlet_SP und Product_SP in den Steuerblöcken oder Steuerungsmodulen, die zum Öffnen oder Schließen der tatsächlichen Ventile verwendet werden, und außerdem die Kommunikationspfadspezifikationen für jedes Steuerungsmodul Target_State in den Modulblöcken mit den Bezeichnungen Drain, Outlet und Product. Eine Rücklesepfadspezifikation stellt den Kommunikationspfad für den Readback-Parameter von Current_State in jedem der Modulblöcke Drain, Outlet und Product bereit, außerdem werden die Zustände der Ventile für die verschiedenen Zustände der Ausrüstungskomponente Reactor_Outlet angegeben. Demgemäß zeigen die Einzelheiten zu Drive im Bildschirmabschnitt **654**, dass bei Reactor_Outlet **616** im geschlossenen Zustand die Entleerungs-, Auslass- und Produktventile alle geschlossen sind, dass während des Entleerungsvorgangs (Draining) die Entleerungs- und Auslassventile geöffnet sind, während das Produktventil

geschlossen ist, und dass während des Abziehens des Produkts das Entleerungsventil geschlossen und das Auslass- und das Produktventil geöffnet sind. Es ist klar, dass die Bildschirmdarstellung **650** der [Fig. 12](#) nur eine Art und Weise verdeutlicht, die zustandsgesteuerte Steuerlogik für ein Ausrüstungs-Modulklassenobjekt zu spezifizieren, und dass ebenso gut jede gewünschte andere Art und Weise gewählt werden kann.

[0102] Wenn gewünscht, können selbstverständlich diese oder ähnliche Bildschirme wie die Bildschirme 9 - 12 von einem Konfigurierungstechniker verwendet werden, um Steuerungsalgorithmen innerhalb der Phasenklassen Dose oder anderen zu erzeugen und zu spezifizieren, oder auch für beliebige der anderen Module, wie z.B. Einheiten-Modulklassenobjekte, Ausrüstungs-Modulklassenobjekte und Display-Modulklassenobjekte, um auf diese Weise jedes gewünschte Modulklassenobjekt zu erzeugen.

[0103] Nach dem Erzeugen eines oder mehrerer Modulklassenobjekte gemäß vorstehender Beschreibung kann der Konfigurierungstechniker anschließend diese Modulklassenobjekte verwenden, um Elemente innerhalb der Prozessanlage **10** zu konfigurieren. [Fig. 13](#) zeigt einen Bildschirm **660**, der eine hierarchische Darstellung **661** aufweist, in der eine Systemkonfiguration **662** der Prozessanlage **10** zur Anzeige gebracht wird. Die Systemkonfiguration **662** enthält einen Abschnitt mit der Bezeichnung Control Strategies **664**, der die Steuerungsstrategien für die Prozessanlage **10** und insbesondere für die Nutzung bei der Steuerung der Prozessentitäten der Prozessanlage **10** spezifiziert. Die Steuerungsstrategien **664** sind in logische Gruppierungen unterteilt, beispielsweise entsprechend den physikalischen Bereichen der Anlage **10**. Ein Bereich Area_A **665** ist im Fenster **661** wiedergegeben. Area_A **665** umfasst eine Sektion zur Salzherstellung **666** mit der Bezeichnung Salts, die eine Vielzahl von Teilen oder Beispielen mehrfach vorhandener Ausrüstungskomponenten enthalten kann, die für die Salzherstellung genutzt werden.

[0104] Um die Sektion der Anlage **10** zur Salzherstellung zu konfigurieren, kann der Konfigurierungstechniker das Einheiten-Modulklassenobjekt **610** für Reaktoren aus der Einheiten-Modulklassenbibliothek **604** der [Fig. 8](#) wählen und dieses durch Ziehen oder auf andere Weise unter die Überschrift Salts **666** kopieren, um eine Instanz eines Reaktors entsprechend der durch das Einheiten-Modulklassenobjekt **610** für Reaktoren spezifizierten Form zu erzeugen. Dieser Reaktor ist in [Fig. 13](#) als das Einheitenmodulobjekt Reactor_1 **668** wiedergegeben. Die Konfigurierungsapplikation **50** kann das Einheitenmodulobjekt Reactor_1 **668** automatisch auf Basis des Namens des Einheiten-Modulklassenobjekts, aus dem es erzeugt wurde, benennen, und zwar in einer Art

und Weise, die die Eindeutigkeit innerhalb des Konfigurierungssystems sicherstellt. Wie in [Fig. 13](#) gezeigt, enthält das Einheitenmodulobjekt Reactor_1 **668** ein Ausrüstungsmodulobjekt für Zumessungen Acid_1, ein Ausrüstungsmodulobjekt für Zumessungen Alkali_1, und ein Ausrüstungsmodulobjekt Outlet 2, die den Modulklassenobjekten für Säure, Alkali und Auslassventile entsprechen, die im Einheiten-Modulklassenobjekt **610** für Reaktoren spezifiziert sind. Außerdem umfasst das Einheitenmodulobjekt Reactor_1 **668** die Einheitenphasen Dose, Drain, Flush und Mix (aus den Phasenklassen abgeleitet) gemäß der Spezifikation durch das Modulklassenobjekt **610**. Ein Ordner mit der Bezeichnung Aliases stellt eine Aliasauflösungstabelle bereit, die die Zuordnung der Aliasnamen enthält, die von den Steuerungsroutinen an beliebigen Stellen des Einheitenmodulobjekts Reactor_1 **668** verwendet werden. Durch die Aliasauflösungstabelle im Einheitenmodulobjekt Reactor_1 **668** ist es möglich, die Steuerungsroutinen vor der oder zur Laufzeit zu analysieren, um sie einer bestimmten Ausrüstungskomponente innerhalb der Prozessanlage **10** zuzuordnen. Die Verwendung von Einheitenphasen und Aliasauflösungstabellen ist detailliert in der US-Patentschrift Nr. 6,385,496 beschrieben, die für die Anmelderin dieses Patents angemeldet wurde, und die hierin ausdrücklich durch Referenz einbezogen ist. Außerdem enthält das Einheitenmodulobjekt Reactor_1 **668** die Modulobjekte Water_HDR1 und Water_In, die dem Ausrüstungs-Modulklassenobjekt für die Wasserzumessung entsprechen, auf die im Einheitenklassenobjekt Reactor **610** und im Steuerungs-Modulklassenobjekt für Absperrventile Water_In des Einheitenklassenobjekts Reactor **610** Bezug genommen wird. Selbstverständlich kann die Konfigurierungsapplikation **50** wiederum automatisch das Bezeichnungsschema, das für die Elemente des Einheitenmodulobjekts Reactox_1 **688** verwendet wurde, auf Basis der Namen der entsprechenden Elemente im Einheiten-Modulklassenobjekt Reactor **610** bereitstellen, aus dem die einzelnen Elemente des Einheitenmodulobjekts Reactor_1 **668** erzeugt wurden, und zwar in einer Art und Weise, die die Eindeutigkeit innerhalb der Konfigurierungsstrategie sicherstellt.

[0105] Aus der Erläuterung unter Bezug auf das Einheiten-Modulklassenobjekt Reactor **610** geht hervor, dass Water_HDR1 ein gemeinsam genutzter Wasserheader ist (entsprechend dem Wassereinlassventilsystem **110** der [Fig. 1](#)). Deshalb wird ein separates Ausrüstungsmodulobjekt unter dem Eintrag Salts **666** mit der Bezeichnung Water_HDR1 **670** bereitgestellt. Selbstverständlich kann der Konfigurierungstechniker das Modulobjekt Water_HDR1 **670** aus dem Ausrüstungs-Modulklassenobjekt **614** für Zumessungen der [Fig. 8](#) erzeugen, indem er das Element durch Kopieren oder Ziehen in das Bildschirmfenster **661** kopiert. Wie hierbei zu erwarten, umfasst der gemeinsam genutzte Wasserheader

Water_HDR1 **670** in der Darstellung spezielle Steuerungsmodulobjekte für Absperrventile (mit den Bezeichnungen Coarse_Valve4 und Fine_Valve4) und ein spezielles Steuerungsmodulobjekt für einen Durchflussmesser (mit der Bezeichnung Flow_Meter4), die den Steuerungs-Modulklassenobjekten Coarse_Valve, Fine_Valve und Flow_Meter des Ausrüstungs-Modulklassenobjekt **614** für Zumessungen der [Fig. 8](#) entsprechen. Außerdem bezieht sich die Referenz Water_HDR1 im Modulobjekt Reactor_1 **668** auf das Ausrüstungsmodulobjekt Water_HDR1 **670**.

[0106] Selbstverständlich könnten beliebige der Ausrüstungs- oder Steuerungsmodule innerhalb eines Einheiten-Modulklassenobjekts als gemeinsam genutzte oder als nicht gemeinsam genutzte Modulobjekte gekennzeichnet werden. Ein nicht gemeinsam genutztes Modulobjekt ist vollständig dem übergeordneten Modulobjekt zugehörig, in dem das nicht gemeinsam genutzte Modulobjekt erzeugt wurde. Ein gemeinsam genutztes Modulobjekt ist mehreren übergeordneten Modulobjekten zugehörig oder diesen zugeordnet. Die Eigenschaft gemeinsam genutztes bzw. nicht gemeinsam genutztes Modulobjekt wirkt sich auf die Anzeige eines Modulobjekts im Explorerfenster aus. Insbesondere bewirkt die Kennzeichnung als nicht gemeinsam genutztes Modulobjekt, dass das Modulobjekt nur unter dem übergeordneten Modulobjekt innerhalb der Steuerungsstrategie aufgelistet wird, während eine Kennzeichnung als gemeinsam genutztes Modulobjekt dazu führt, dass der gemeinsam genutzte Modulblock bzw. das Modulobjekt unter jedem der übergeordneten Modulobjekte aufgelistet wird, die das Element gemeinsam nutzen, und außerdem als eigenständiges Modulobjekt in der Explorerhierarchie eingetragen wird.

[0107] Es ist klar, dass der Konfigurierungstechniker innerhalb der Sektion Control Strategies **664** des Konfigurierungssystems eine Steuerungsstrategie für einen vollständigen Reaktor erzeugen kann, indem er einfach ein Einheiten-Modulklassenobjekt **610** für Reaktoren kopiert und damit eine Instanz desselben erzeugt (das Einheitenmodulobjekt Reactor **668**), die über alle Elemente verfügt, die im Einheiten-Modulklassenobjekt **610** für Reaktoren spezifiziert sind. Die einzelnen Einheiten und insbesondere die Steuerungselemente des Einheitenmodulobjekts Reactor **668** können jedoch mit bestimmten Entitäten innerhalb der Prozessanlage **10** verknüpft oder verbunden werden, indem Bezeichnungen für Eingänge/Ausgänge verwendet werden, die von Modulobjekt zu Modulobjekt variieren, wobei ein Dialogfenster zur Konfigurierung verwendet wird, das generiert wird, damit ein Konfigurierungstechniker die Art und Weise spezifizieren kann, in der die Modulobjekte in die Prozessanlage eingebunden werden. Wie weiter oben verdeutlicht, ist das erzeugte Modulobjekt Reactor **668** immer noch mit dem Einheiten-Mo-

dulklassenobjekt **610** für Reaktoren verbunden und diesem zugehörig, so dass Änderungen, die am Einheiten-Modulklassenobjekt **610** für Reaktoren vorgenommen werden, wenn gewünscht automatisch auf das Modulobjekt Reactor_1 **668** übertragen werden können und somit auch in diesem zur Verfügung stehen. In jedem Fall kann der Konfigurierungstechniker die erzeugten Modulobjekte verändern oder individuell anpassen, z.B. das Modulobjekt Reactor_1 **668**, so dass Änderungen am Stamm-Modulklassenobjekt akzeptiert oder zurückgewiesen werden, und zwar auf einer globalen (objektweiten) Basis oder auf einer Basis von Element zu Element.

[0108] Der rechte Abschnitt **672** des Bildschirms **660** zeigt bzw. listet die Parameter, Alarmer etc. auf, die mit jedem der Elemente im Modulobjekt Reactor_1 **668** (da dieses das in der Sektion **661** angewählte Element ist) verbunden sind, und außerdem die diesen zugeordneten Ventile, Filter und weitere Informationen. Selbstverständlich kann der Konfigurierungstechniker diese Daten für jedes der Elemente des Modulobjekts Reactor_1 **668** wie gewünscht ändern oder spezifizieren, um so einzelne Modulobjekte, die aus dem Modulklassenobjekt erzeugt wurden, in jeder gewünschten Weise anzupassen.

[0109] [Fig. 14](#) zeigt eine Bildschirmausgabe **680**, die die Konfiguration des Ausrüstungsmodulobjekts Acid1 **682** des Einheitenmodulobjekts Reactor_1 **668** verdeutlicht. Da das Ausrüstungsmodulobjekt Acid1 **682** in [Fig. 14](#) angewählt ist, sind die Elemente desselben im rechten Abschnitt **684** des Bildschirms **680** aufgezeigt. Diese Elemente umfassen Steuerungsmodulobjekte Coarse_Valve1, Fine_Valve1 und Flow_Meter1 und sichtbare Parameter.

[0110] Ein Pop-up-Fenster **686** stellt Information bezüglich der Modulunterobjekte bereit, die dem Ausrüstungsmodulobjekt Acid1 zugeordnet sind, und ebenso hinsichtlich der Modulklassen, aus der diese Modulunterobjekte herkommen bzw. erzeugt wurden. Insbesondere gehört das Ausrüstungsmodulobjekt Acid1 zum Ausrüstungs-Modulklassenobjekt für Zumessungen, die Modulobjekte Coarse_Valve1 und Fine_Valve1 gehören zum Steuerungs-Modulklassenobjekt für Absperrventile aus der Bibliothekssektion **608** der [Fig. 8](#), und das Modulobjekt Flow_Meter1 gehört zum Steuerungs-Modulklassenobjekt Flow_Meter in der Bibliothekssektion **608** der [Fig. 8](#).

[0111] Das Pop-up-Fenster Acid1 Configuration **686** weist Reiter mit den Bezeichnungen Parameters, IO (Eingabe/Ausgabe), Alarms und Fieldbus auf. Der Reiter Parameters kann dazu dienen, die Parameter eines jeden der Elemente der Zumessung Acid1 des Modulobjekts Reactor_1 **668** zu spezifizieren oder einzusetzen. Der Reiter IO kann dazu dienen, die

Ein-/Ausgabeverbindungen zwischen jedem der Elemente des Modulobjekts der Zumessung Acid1 **682** des Modulobjekts Reactor_1 **668** zu spezifizieren, um damit die einzelnen Elemente mit den tatsächlichen Ausrüstungskomponenten innerhalb der Prozessanlage **10** zu verbinden. Wenn gewünscht, können diese Verbindungen mit Hilfe eines Tabellenkalkulationsalgorithmus bzw. -programms entsprechend [Fig. 15](#) spezifiziert werden, um die Verbindungen zwischen allen Elementen eines Modulobjekts (auf jeder Ebene) und den tatsächlichen Elementen innerhalb der Prozessanlage **10** auf einer globalen oder Gruppenbasis zu spezifizieren. Das Tabellenschema **688** der [Fig. 15](#) liefert eine tabellarische Ansicht, die es einem Konfigurierungstechniker ermöglicht, die Grobventile verschiedener Ausrüstungsmodulobjekte für Zumessungen insgesamt mit den Ausrüstungskomponenten der Prozessanlage **10** zu verbinden. In diesem Fall sind die Pfadnamen IO_Out und IO_Readback für die Ventile mit den Bezeichnungen VLV-101A, VLV201A, VLV-301A und VLV-401A spezifiziert. Wenn dies gewünscht ist, können jedoch auch verschiedene Elemente des gleichen Moduls gemeinsam spezifiziert werden.

[0112] Bezug nehmend auf [Fig. 14](#) kann der Reiter Alarms des Pop-up-Fensters **686** dazu dienen, Alarmer in Verbindung mit dem Modulobjekt Acid_1 zu spezifizieren oder zu konfigurieren, mit dem Reiter Fieldbus können Verbindungen zu einzelnen Fieldbus-Vorrichtungen wie erforderlich spezifiziert werden. Selbstverständlich können ähnliche Konfigurierungsbildschirme für andere Modulobjekte innerhalb des Modulobjekts Reactor_1 **668** oder für das gesamte Modulobjekt Reactor_1 **668** eingesetzt werden. Ebenso können die gleichen oder ähnliche Konfigurierungsbildschirme für andere Modulobjekte verwendet werden. Es ist klar, dass die Konfigurierungsbildschirme auf jeder Modulobjektebene einschließlich der Einheitenmodulebene, der Ausrüstungsmodulebene, der Steuerungsmodulebene etc. ausgegeben und genutzt werden können.

[0113] [Fig. 16](#) zeigt eine Bildschirmdarstellung **690**, die ein Explores-Fenster **661** des Konfigurierungssystems für die Prozessanlage **10** enthält, in der der Konfigurierungstechniker ein weiteres Modulobjekt **692** für Reaktoren mit der Bezeichnung Reactor_1_1 hinzugefügt hat, das aus dem Einheiten-Modulklassenobjekt **610** für Reaktoren der [Fig. 8](#) kopiert oder erzeugt wurde. Selbstverständlich kann der Konfigurierungstechniker bei der Erzeugung des Modulobjekts Reactor_1_1 **692** abweichende Parameter, abweichende Verknüpfungen zu anderen Vorrichtungen und sogar abweichende Steuerungsroutinen spezifizieren, die in den einzelnen Unterelementen zur Anwendung kommen sollen. Wie [Fig. 16](#) zeigt, enthält das Modulobjekt Reactor_1_1 Unterobjekte für jedes der Ausrüstungs-Modulklassenobjekte innerhalb des Einheiten-Modulklassenobjekts **610** für Reaktoren

der [Fig. 8](#), einschließlich der Ausrüstungsmodulobjekte Acid2, Alkali2 und Outlet4. Das Modulobjekt Reactor_1_1 enthält außerdem Steuerungsmodulobjekte Water_In2 und Level_Meter2, eine Referenz zu den gemeinsam genutzten Ausrüstungsmodulobjekten mit den Bezeichnungen Water_HDR1 und den Einheitenphasen Dose, Drain, Flush und Mix. Die ausschließlich dem Modulobjekt Reactor_1_1 **692** zugeordneten Unterobjekte tragen eindeutige Bezeichnungen, während die gemeinsam genutzten Einheiten, beispielsweise das Modulobjekt Water_HDR1, mit dem Namen des gemeinsam genutzten Wasserheaders bezeichnet sind.

[0114] Es ist klar, dass der Konfigurierungstechniker in entsprechender Weise auch andere Einheiten-Modulklassenobjekte, Ausrüstungs-Modulklassenobjekte, Steuerungs-Modulklassenobjekte und Display-Modulklassenobjekte verwenden kann, um Konfigurierungselemente für Einheiten-, Ausrüstungs-, Steuerungs- und Displayelemente innerhalb der Prozesssteuerungsumgebung entsprechend den jeweils beschriebenen Prinzipien zu erzeugen. Außerdem ist klar, dass die Nutzung dieser Modulklassenobjekte, die von sehr differenzierter Art sein können, dem Konfigurierungstechniker deutliche Vorteile bei der Erzeugung einer Konfiguration bieten, da der Techniker nicht jedes individuelle Steuerungselement einzeln erzeugen oder jedes individuelle Steuerungselement einzeln aus einer Steuerungstemplate kopieren muss, sondern stattdessen Modulklassenobjekte verwenden kann, die auf immer höherer Ebene oder mit immer größerem Umfang bereitgestellt werden, um immer umfassendere Gruppenkonfigurierungselemente für die Konfigurierung der Prozessanlage **10** zu erzeugen.

[0115] Außerdem kann der Konfigurierungstechniker Änderungen an den Konfigurierungselementen verschiedener Prozesseentitäten auf einer globalen Basis vornehmen, indem er ein oder mehrere der Einheiten-Modulklassenobjekte ändert und dabei die Übertragung dieser Änderungen auf jedes der Modulobjekte veranlasst, die aus den jeweiligen Einheiten-Modulklassenobjekten erzeugt wurden und diesen zugeordnet sind. Dieses Merkmal macht Änderungen innerhalb der Konfiguration, die nach der eigentlichen Erzeugung der Konfiguration notwendig werden, einfacher und weniger zeitraubend. Darüber hinaus kann der Konfigurierungstechniker eine Zugriffsebene für verschiedene Elemente oder Komponenten von Modulobjekten innerhalb des Konfigurierungssystems spezifizieren, indem er Sicherheitsparameter innerhalb der Modulklassenobjekte setzt. Wie oben erwähnt, kann der Konfigurierungstechniker einen Zugriffsschutz auf modulweiser Basis auf jeder Ebene einrichten, beispielsweise auf der Einheitenmodulebene, der Ausrüstungsmodulebene, der Steuerungsmodulebene und der Displaymodulebene. Auf diese Weise können einige Elemente eines

Einheitenmodulobjekts einsehbar sein, während andere unsichtbar sind.

[0116] Selbstverständlich können nach der kompletten Erstellung des Konfigurierungssystems und nachdem die Modulobjekte mit den einzelnen Prozessentitäten innerhalb der Prozessanlage **10** verbunden sind, die Steuerungs- und Displaymodule oder diesen Modulen zugeordneten Elemente in die jeweiligen Steuerungsgeräte **12** und Workstations **14** der [Fig. 1](#) heruntergeladen werden, von denen sie im Betrieb der Prozessanlage **10** ausgeführt werden.

[0117] [Fig. 17](#) ist ein Flussdiagramm einer beispielhaften Routine **700** für die Konfigurierung eines Prozessanlagenelements zur Begrenzung des Zugriffs auf das Element. Die Routine **700** kann durch ein oder mehrere Rechnersysteme wie z.B. eine oder mehrere Workstations **14** und/oder die Steuerungsgeräte **12** der Prozessanlage **10** der [Fig. 1](#) implementiert werden. Die Routine **700** kann auch durch ein von der Prozessanlage **10** getrenntes Rechnersystem implementiert werden. So könnte beispielsweise ein Drittanbieter die Routine **700** nutzen, um den Zugriff z.B. auf ein vom Drittanbieter entwickeltes Modulobjekt, Modulklassenobjekt, eine Modultemplate, eine Composite-Template etc. zu begrenzen. Das entwickelte Modulobjekt, Modulklassenobjekt, die Composite-Template etc. könnten dann von einem Kunden des Drittanbieters genutzt werden, aber es wäre dem Kunden beispielsweise nicht möglich, Einsicht in einige oder alle der internen Elemente des entwickelten Modulobjekts, Modulklassenobjekts, der Composite-Template etc. zu nehmen. Generell kann es sich bei dem Prozessanlagenelement, das konfiguriert werden soll, beispielsweise um ein Modulobjekt, ein Modulklassenobjekt, eine Composite-Template etc. handeln. Die Routine **700** kann als Teil einer Konfigurierungsapplikation, wie z.B. der Konfigurierungsapplikation **50**, implementiert werden. So kann die Routine **700** z.B. während oder nach der Konfigurierung eines Modulobjekts, eines Modulklassenobjekts, einer Composite-Template etc. unter Verwendung der Konfigurierungsapplikation **50** implementiert werden. Die Routine **700** kann aber auch durch eine von der Konfigurierungsapplikation **50** getrennte Applikation implementiert werden.

[0118] In einem Block **702** wird eine Nutzereingabe empfangen, die anzeigt, dass der Zugriff auf das Prozessanlagenelement gesteuert werden muss (d.h. dem Prozessanlagenelement zugeordnete Daten müssen geschützt werden). Ein Konfigurierer des Elements kann z.B. ein Kästchen markieren, ein Bedienelement oder ein Menüelement etc. anwählen, das anzeigt, dass dem Element zugeordnete Daten geschützt werden müssen. Die Nutzereingabe kann wahlweise auch anzeigen, welche dem Element zugeordneten Daten geschützt werden müssen. Der Nutzer kann beispielsweise entscheiden, bestimmte

Daten zu schützen, während andere Daten ungeschützt bleiben.

[0119] Alternativ kann dem Nutzer keine Wahlmöglichkeit gegeben werden, welche Daten detektiert werden. Beispielsweise kann es möglich sein, dass dem Nutzer nur die Wahl offensteht, ob ein Element geschützt werden soll oder nicht. Wenn ein Nutzer entscheidet, einem Modulobjekt oder einem Modulklassenobjekt zugeordnete Daten zu schützen, kann ein dem Modulobjekt oder dem Modulklassenobjekt zugeordneter Datensatz geschützt werden, während nicht im Satz enthaltene Daten nicht geschützt sind. Beispielsweise können Funktionsblöcke, die ein Modulobjekt oder ein Modulklassenobjekt bilden, sowie ihre Verbindungen geschützt werden, aber die Parameter des Modulobjekts oder des Modulklassenobjekts sind nicht geschützt.

[0120] [Fig. 18A](#) und [Fig. 18B](#) sind beispielhafte Bildschirmdarstellungen, die von einer Konfigurationsapplikation wie der Konfigurationsapplikation **50** der [Fig. 1](#) erzeugt werden können. Ähnliche Bildschirmdarstellungen könnten durch andere Applikationstypen erzeugt werden, wie z.B. durch eine Applikation, die genutzt wird, um einen Prozess während des Ablaufs zu überwachen, durch eine Applikation zur Fehlerbereinigung für die Fehlerbereinigung einer Konfiguration, durch eine Asset-Managementapplikation etc. Die Bildschirmdarstellung **720** umfasst einen Abschnitt **722** eines Explorer-Fensters mit einer Bibliothek **724** von Composite-Templates. Eine der Composite-Templates ist die Composite-Template DYN_FULL **726**. Die Composite-Template DYN_FULL **726** beinhaltet verschiedene Unterelemente, die, da die Composite-Template DYN_FULL **726** im Explorer-Fenster **722** angewählt wurde, auf der rechten Seite **728** der Bildschirmdarstellung **720** zur Ansicht gebracht werden. Die Unterelemente, aus denen die Composite-Template DYN_FULL **726** besteht, umfassen Parameter und Funktionsblöcke **730**.

[0121] In der Bildschirmdarstellung **720** wurde die Composite-Template DYN_FULL **726** nicht als geschützt konfiguriert und somit werden die Funktionsblöcke **730** auf der rechten Seite **728** der Bildschirmdarstellung **720** zur Ansicht gebracht. Dagegen zeigt [Fig. 18B](#) eine Bildschirmdarstellung **732**, die der als geschützt konfigurierten Composite-Template DYN_FULL **726** entspricht. In der Bildschirmdarstellung **732** sind die Funktionsblöcke **730** daher in der rechten Seite **728** nicht gezeigt (d.h. sie sind unsichtbar). Die Parameter der Composite-Template DYN_FULL **726** erscheinen jedoch weiterhin, obwohl diese als geschützt konfiguriert wurde.

[0122] In dem ebenfalls in [Fig. 17](#) abgebildeten Block **704** kann der Nutzer aufgefordert werden, ein Passwort einzugeben. Beispielsweise kann ein Fens-

ter, ein Bildschirm, ein akustisches Signal etc. den Nutzer auffordern, ein Passwort einzugeben. Wie nachstehend beschrieben, kann das Passwort zu einem späteren Zeitpunkt genutzt werden, um auf die geschützten Daten zuzugreifen und/oder das Element neu zu konfigurieren, um den Schutz aufzuheben. In einem Block **706** kann das Passwort empfangen werden. Das Passwort kann beispielsweise über eine Tastatur, einen Tastenblock, einen Sensor-Bildschirm, einen Lichtgriffel, eine Maus etc. eines Rechnersystems eingegeben werden.

[0123] In einem Block **708** kann eine Angabe zu den dem Prozessanlagenelement zugeordneten und zu schützenden Daten gespeichert und dem Element zugeordnet werden. Ist das Element beispielsweise ein Objekt, so kann die Angabe als Teil dieses Objekts gespeichert werden. In einem weiteren Beispiel könnte die Angabe getrennt vom Objekt gespeichert und ein Link als Verweis könnte als Teil des Objekts gespeichert werden. In noch einem anderen Beispiel könnte die Angabe als Teil eines Modulklassenobjekts gespeichert werden, wenn das Prozessanlagenelement ein aus dem betreffenden Modulklassenobjekt erzeugtes Element ist. Hat der Nutzer keine Wahlmöglichkeit, welche Daten geschützt werden sollen, kann der Block **708** entfallen.

[0124] In einem Block **710** kann das im Block **706** empfangene Passwort gespeichert und dem Element zugeordnet werden. Wenn das Element z.B. ein Objekt ist, kann das Passwort als Teil dieses Objekts gespeichert werden. In einem weiteren Beispiel könnte das Passwort getrennt vom Objekt gespeichert und ein Link zum Passwort als Teil des Objekts gespeichert werden. In noch einem anderen Beispiel könnte das Passwort als Teil eines Modulklassenobjekts gespeichert werden, wenn das Prozessanlagenelement ein aus dem betreffenden Modulklassenobjekt erzeugtes Element ist.

[0125] Wahlweise können die zu schützenden Daten und/oder das Passwort verschlüsselt werden. Bei diesen Implementierungen können die Blöcke **708** und **710** das Speichern und das Zuordnen der verschlüsselten Daten und/oder des verschlüsselten Passworts umfassen. Außerdem könnten die geschützten oder die verschlüsselten Daten als Teil eines Modulklassenobjekts gespeichert werden, wenn das Prozessanlagenelement ein aus dem betreffenden Modulklassenobjekt erzeugtes Element ist.

[0126] Wenn das Element ein Modulklassenobjekt ist, kann die Angabe der zu schützenden Daten und des Passworts an Elemente übertragen werden, die aus dem Modulklassenobjekt erzeugt wurden.

[0127] Eine der Routine **700** ähnliche Routine könnte genutzt werden, um den Schutz des Prozessanlagenelements oder spezieller, dem Element zugeord-

ner Daten aufzuheben. Hierfür könnte eine Nutzereingabe mit der Angabe empfangen werden, dass der Schutz von einem geschützten Prozessanlagenelement zugeordneten Daten aufgehoben werden soll. Der Nutzer könnte dann aufgefordert werden, ein Passwort einzugeben. Entspricht das empfangene Passwort dem in Block **710** der Routine **700** gespeicherten und dem geschützten Prozessanlagenelement zugeordneten Passwort, könnte der Nutzer zur Angabe aufgefordert werden, bei welchen (oder ob bei allen) der geschützten Daten der Schutz aufzuheben ist. Die gespeicherte und dem Element in Block **708** der Routine **700** zugeordnete Angabe könnte dann geändert werden, um anzuzeigen, dass der Schutz einiger oder aller Daten aufgehoben wurde.

[0128] [Fig. 19](#) ist ein Flussdiagramm einer beispielhaften Routine **750** für die Konfigurierung einer Gruppe von Prozessanlagenelementen, um den Zugriff auf die Gruppe zu begrenzen. Die Routine **750** kann von einem oder mehreren Rechnersystemen implementiert werden, z.B. durch eine oder mehrere der Workstations **14** und/oder die Steuerungsgeräte **12** der Prozessanlage **10** in [Fig. 1](#). Bei den Prozessanlagenelementen der Gruppe kann es sich beispielsweise um Modulobjekte, Modulklassenobjekte und/oder Composite-Templates handeln. Die Routine **750** kann als Teil einer Konfigurierungsapplikation, wie z.B. der Konfigurierungsapplikation **50**, implementiert werden. Die Routine **750** kann beispielsweise nach der Konfigurierung von Elementen der Gruppe unter Verwendung der Konfigurierungsapplikation **50** implementiert werden. Die Routine **750** kann aber auch durch eine von der Konfigurierungsapplikation **50** getrennte Applikation implementiert werden.

[0129] In einem Block **752** wird eine Nutzereingabe empfangen, die anzeigt, dass der Zugriff auf die Gruppe von Elementen gesteuert werden muss (d.h. die Gruppe von Prozessanlagenelementen muss geschützt werden). Beispielsweise kann der Zugriff auf jedem Element der Gruppe zugeordnete Daten geschützt werden. Außerdem kann die Hinzufügung von Elementen zu der Gruppe oder das Löschen von Elementen aus der Gruppe nicht zulässig sein. Die Gruppe von Elementen kann beispielsweise eine Bibliothek von Elementen, einen Ordner mit Elementen, eine Klasse von Elementen, eine Kategorie von Elementen etc. umfassen.

[0130] Ein Konfigurierer der Gruppe von Elementen kann die Gruppe zunächst beispielsweise durch Markieren der Gruppe von Elementen, die Anwahl des Namens einer Bibliothek, eines Ordners, einer Klasse, einer Kategorie etc. auswählen. Unter Bezugnahme auf [Fig. 18A](#) könnte die betreffende Person z.B. die Unterbibliothek **734** oder die Kategorie **736** auswählen, eine Vielzahl von Modulen der Kategorie **736** markieren etc. Dann kann die Person z.B. ein Käst-

chen markieren oder ein Bedienelement, ein Menüelement etc. anwählen, die anzeigen, dass die Gruppe geschützt werden soll.

[0131] In [Fig. 19](#) kann der Nutzer in einem Block **754** aufgefordert werden, ein Passwort einzugeben. Beispielsweise kann ein Fenster, ein Bildschirm, ein akustisches Signal etc. den Nutzer auffordern, ein Passwort einzugeben. Wie nachstehend beschrieben, kann das Passwort zu einem späteren Zeitpunkt für den Zugriff auf die geschützte Gruppe und/oder die Neukonfigurierung der Gruppe zur Aufhebung des Schutzes genutzt werden. In einem Block **756** kann das Passwort empfangen werden, das beispielsweise über eine Tastatur, einen Tastenblock, einen Sensor-Bildschirm, einen Lichtgriffel, eine Maus etc. eines Rechnersystems eingegeben werden kann.

[0132] In einem Block **758** kann die Angabe, dass die Gruppe von Elementen geschützt werden soll, gespeichert und der Gruppe von Elementen zugeordnet werden. Wenn es sich beispielsweise bei der Gruppe um eine Kategorie von Elementen handelt, kann die Angabe, dass die Kategorie geschützt werden muss, gespeichert werden, und diese Angabe kann der betreffenden Kategorie zugeordnet werden, so dass bei einem nachfolgenden Versuch, die Kategorie zu ändern, festgestellt werden kann, dass die Kategorie geschützt ist. Zusätzlich kann eine Angabe mit jedem Element der Gruppe verbunden werden. Handelt es sich bei den Elementen beispielsweise um Objekte, so kann die Angabe, dass das Element Teil einer geschützten Gruppe ist, als Teil jedes Objekts gespeichert werden. Die Angabe kann auch getrennt von jedem Objekt gespeichert werden, wobei ein Link als Verweis als Teil jedes Objekts gespeichert werden kann.

[0133] In einem Block **760** kann das im Block **756** empfangene Passwort gespeichert und der Gruppe von Elementen zugeordnet werden. Handelt es sich z.B. bei der Gruppe um eine Kategorie von Elementen, kann das Passwort gespeichert werden und dieses Passwort kann der Kategorie zugeordnet werden. Zusätzlich kann das Passwort jedem Element der Gruppe zugeordnet werden. Wenn es sich bei den Elementen beispielsweise um Objekte handelt, so kann das Passwort als Teil jedes Objekts gespeichert werden. Das Passwort kann auch getrennt von jedem Objekt gespeichert werden, wobei ein Link zum Passwort als Teil jedes Objekts gespeichert werden kann.

[0134] Entsprechend der Routine **700** in [Fig. 17](#) kann eine der Routine **750** ähnliche Routine genutzt werden, um den Schutz der Gruppe von Prozessanlagenelementen aufzuheben.

[0135] [Fig. 20](#) ist ein Flussdiagramm einer beispiel-

haften Routine **770** für die Entscheidung, ob die Einsicht in Daten, die einem Prozessanlagenelement zugeordnet sind, das geschützt oder ungeschützt sein kann, zulässig sein soll. Die Routine **770** kann durch ein oder mehrere Rechnersysteme, wie eine oder mehrere der Workstations **14** und/oder die Steuerungsgeräte **12** der Prozessanlage **10** in [Fig. 1](#) implementiert werden. Das Prozessanlagenelement kann beispielsweise ein Modulobjekt, ein Modulklassenobjekt, eine Composite-Template etc. umfassen. Die Routine **770** kann als Teil einer Konfigurierungsapplikation, wie z.B. der Konfigurierungsapplikation **50** implementiert werden. Die Routine **770** kann auch als Teil einer anderen Applikation, beispielsweise einer Überwachungs-Applikation, einer Applikation zur Fehlerbereinigung, einer Asset-Managementapplikation etc. implementiert werden.

[0136] In einem Block **772** kann eine Anforderung zur Anzeige von einem Prozessanlagenelement zugeordneten Daten empfangen werden. Ein Nutzer kann beispielsweise ein Element in einem Explorer-Fenster anwählen und versuchen, direkt in ein Modul einzuklicken, um Komponenten des Moduls etc. zur Ansicht zu bringen. Unter Bezugnahme auf [Fig. 18B](#) als Beispiel hat ein Nutzer die Composite-Template DYN_FULLL **726** im Abschnitt **722** des Explorer-Fensters in der Bildschirmdarstellung **732** angewählt. Damit sollen Unterelemente der Composite-Template DYN_FULLL **726** im rechten Teil **728** der Bildschirmdarstellung **732** zur Ansicht gebracht werden.

[0137] In einem Block **774** wird festgestellt, ob dem Element zugeordnete Daten geschützt sind. Es kann beispielsweise festgestellt werden, dass eine dem Element zugeordnete Angabe existiert, wie z.B. die in Block **708** der [Fig. 17](#) gespeicherte Angabe, die anzeigt, dass mit dem Element verbundene Daten geschützt sind. Weiter kann beispielsweise festgestellt werden, ob eine dem Element zugeordnete Angabe vorhanden ist, wie die in Block **758** der [Fig. 19](#) gespeicherte Angabe, die anzeigt, dass das Element Teil einer geschützten Gruppe von Elementen ist.

[0138] Sind die dem Element zugeordneten Daten nicht geschützt, so kann der Ablauf zu einem Block **776** wechseln, in dem dem Element zugeordnete Daten angezeigt werden. Wenn der Nutzer das Element beispielsweise in einem Explorer-Fenster angewählt hat, können Unterelemente dieses Elements in einem Abschnitt eines Bildschirms angezeigt werden. Unter Bezugnahme auf [Fig. 18A](#) als Beispiel hat ein Nutzer die Composite-Template DYN_FULLL **726** in Abschnitt **722** des Explorer-Fensters in der Bildschirmdarstellung **720** angewählt. Damit werden Unterelemente der Composite-Template DYN_FULLL **726** einschließlich der Funktionsblöcke **730** im rechten Abschnitt **728** der Bildschirmdarstellung **720** angezeigt. Fordert der Nutzer andererseits beispiels-

weise Einsicht in ein Modul an, um Komponenten des Moduls einzusehen, so könnten die Komponenten dieses Moduls auf einem Bildschirm oder in einem Fenster sichtbar gemacht werden.

[0139] Wird in Block **774** festgestellt, dass die dem Element zugeordneten Daten geschützt sind, so kann der Ablauf zu einem Block **786** wechseln, in dem mit dem Element verbundene geschützte Daten nicht angezeigt werden, wogegen ungeschützte Daten angezeigt werden können.

[0140] Wenn in Block **774** festgestellt wird, dass die dem Element zugeordneten Daten geschützt sind, kann der Ablauf wahlweise zu einem Block **778** wechseln. In Block **778** kann der Nutzer beispielsweise über ein Fenster, einen Bildschirm oder mittels eines akustischen Signals aufgefordert werden, ein Passwort einzugeben. In einem Block **780** kann das vom Nutzer eingegebene Passwort empfangen werden. Das Passwort kann z.B. über eine Tastatur, einen Tastenblock, einen Sensor-Bildschirm, einen Lichtgriffel, eine Maus etc. eines Rechnersystems eingegeben werden. In einem Block **782** kann das im Block **780** empfangene Passwort mit einem gespeicherten Passwort verglichen werden, das dem Element zugeordnet ist. Das empfangene Passwort kann beispielsweise mit dem in Block **710** der [Fig. 17](#) gespeicherten Passwort oder mit dem in Block **760** der [Fig. 19](#) gespeicherten Passwort verglichen werden.

[0141] In einem Block **784** kann bestimmt werden, ob das in Block **780** empfangene Passwort dem gespeicherten und dem Element zugeordneten Passwort entspricht. Ist das der Fall, so können die dem Element zugeordneten Daten in Block **776** angezeigt werden. Stimmen die Passwörter nicht überein, so werden dem Element zugeordnete geschützte Daten nicht angezeigt, während ungeschützte Daten in Block **786** zur Ansicht gebracht werden können. Beispielsweise werden in [Fig. 18B](#) Parameter des Moduls DYN_FULLL **726** im rechten Abschnitt **728** der Bildschirmdarstellung **732** angezeigt, während die Funktionsblöcke, aus denen das Modul DYN_FULLL **726** besteht, nicht angezeigt werden.

[0142] Es ist klar, dass die Blöcke **778**, **780**, **782** und **784** optional sind und entfallen können. In diesem Fall kann, wie vorstehend erläutert, der Ablauf in Block **786** fortgesetzt werden, wenn in Block **774** festgestellt wird, dass die Daten geschützt sind.

[0143] [Fig. 21](#) ist ein Flussdiagramm einer beispielhaften Routine **800** für die Entscheidung, ob die Änderung eines Prozessanlagenelements oder einer Gruppe von Prozessanlagenelementen zulässig sein soll, das bzw. die geschützt oder ungeschützt sein kann. Die Routine **800** kann durch ein oder mehrere Rechnersysteme wie eine oder mehrere der Work-

stations **14** und/oder der Steuerungsgeräte **12** der Prozessanlage **10** in [Fig. 1](#) implementiert werden. Zu den Prozessanlagenelementen können z.B. Modulobjekte, Modulklassenobjekte und/oder Composite-Templates gehören. Die Routine **800** kann als Teil einer Konfigurierungsapplikation wie der Konfigurierungsapplikation **50** implementiert werden. Die Routine **800** kann auch als Teil einer anderen Applikation wie beispielsweise einer Asset-Managementapplikation implementiert werden.

[0144] In einem Block **802** kann eine Anforderung auf Änderung eines Prozessanlagenelements oder einer Gruppe von Prozessanlagenelementen empfangen werden. Ein Nutzer kann beispielsweise ein Element in einem Explorer-Fenster anwählen, um es zu ändern, es kann eine Komponente eines Elements zur Änderung anwählen etc. Weiter kann der Nutzer z.B. versuchen, ein Element aus einer Gruppe von Elementen zu löschen, ein neues Element zu einer Gruppe von Elementen hinzuzufügen etc.

[0145] In einem Block **804** wird festgestellt, ob das Element oder die Gruppe von Elementen geschützt ist. Es kann beispielsweise festgestellt werden, ob eine dem Element zugeordnete Angabe vorhanden ist, wie die in Block **708** der [Fig. 17](#) gespeicherte Angabe, die anzeigt, dass das Element geschützt ist. Weiter kann beispielsweise festgestellt werden, ob eine dem Element zugeordnete Angabe existiert, wie die in Block **758** in [Fig. 19](#) gespeicherte Angabe, die anzeigt, dass das Element Teil einer geschützten Gruppe von Elementen ist. Als noch ein weiteres Beispiel kann festgestellt werden, ob es eine der Gruppe von Elementen zugeordnete Angabe gibt, wie die in Block **758** der [Fig. 19](#) gespeicherte Angabe, die anzeigt, dass die Gruppe von Elementen geschützt ist.

[0146] Sind die dem Element oder der Gruppe von Elementen zugeordneten Daten nicht geschützt, kann der Ablauf zu Block **806** wechseln, in dem Änderungen zulässig sind. Wird in Block **804** jedoch festgestellt, dass das Element oder die Gruppe von Elementen geschützt ist, kann der Ablauf zu einem Block **808** wechseln. In Block **808** kann eine Meldung (z.B. ein Fenster, ein Bildschirm, ein akustisches Signal etc.) erzeugt werden, die anzeigt, dass das Element oder die Gruppe von Elementen, die der Nutzer zu ändern versucht, geschützt ist. Wahlweise kann Block **808** entfallen. In diesem Fall kann das Element oder die Gruppe von Elementen so zur Ansicht gebracht werden, dass dem Nutzer der geschützte Zustand des Elements oder der Gruppe von Elementen angezeigt wird. Eine Darstellung des Elements oder der Gruppe von Elementen auf einem Bildschirm kann beispielsweise abgeschattiert sein oder ein Symbol etc. enthalten, um anzuzeigen, dass das Element oder die Gruppe geschützt ist. In einem anderen Beispiel kann keine Angabe vorhanden sein, die dem Nutzer anzeigt, dass das Element oder die Gruppe

geschützt ist. In diesem Fall ist es dem Nutzer einfach nur unmöglich, das Element oder die Gruppe zu ändern. Nach Block **808**, oder wenn Block **808** entfällt, kann die Routine beendet werden.

[0147] Wahlweise kann der Ablauf in einem Block **810** fortgesetzt werden. In Block **810** kann der Nutzer beispielsweise über ein Fenster, einen Bildschirm, ein akustisches Signal etc. aufgefordert werden, ein Passwort einzugeben. In einem Block **812** kann das vom Nutzer eingegebene Passwort empfangen werden, das beispielsweise mittels einer Tastatur, eines Tastenblocks, eines Sensor-Bildschirms, eines Lichtgriffels, einer Maus etc. eines Rechnersystems eingegeben werden kann. In einem Block **814** kann das in Block **812** empfangene Passwort mit einem gespeicherten und dem Element oder der Gruppe von Elementen zugeordneten Passwort verglichen werden. Das empfangene Passwort kann z.B. mit dem in Block **710** der [Fig. 17](#) oder mit dem in Block **760** der [Fig. 19](#) gespeicherten Passwort verglichen werden.

[0148] In einem Block **816** kann bestimmt werden, ob das in Block **812** empfangene Passwort dem gespeicherten und dem Element oder der Gruppe von Elementen zugeordneten Passwort entspricht. Stimmen die Passwörter überein, kann die Änderung in Block **806** zugelassen werden. Besteht keine Übereinstimmung der Passwörter, kann in Block **818** eine Nachricht erzeugt werden, dass die Änderung nicht zugelassen wird. Alternativ oder zusätzlich kann die Nachricht angeben, dass das in Block **812** empfangene Passwort nicht richtig war.

[0149] Wie vorstehend erläutert, sind die Blöcke **810**, **812**, **814**, **816** und **818** optional und können entfallen. In diesem Fall endet die Routine nach Block **808**. Auch Block **808** ist optional und kann entfallen. In diesem Fall kann die Routine enden oder ggf. zum optionalen Block **810** wechseln, wenn in Block **804** festgestellt wurde, dass das Element oder die Gruppe geschützt ist.

[0150] Obwohl in [Fig. 17](#) und [Fig. 19-Fig. 21](#) Passwörter für die Begrenzung des Zugriffs auf Prozessanlagenelemente behandelt wurden, können alternativ oder zusätzlich zu Passwörtern andere Techniken für die Zugriffsbegrenzung angewandt werden. Beispielsweise könnten eine oder mehrere im Voraus genehmigte Sicherheitsebenen für den Zugriff mit Angabe des zum Zugriff auf Elemente befugten Personals, eine Auflistung von Merkmalen bestimmter Personen (z.B. Log-In-Namen, ID-Nummern etc.) mit Zugriffsberechtigung, biometrische Daten etc. verwendet werden. Ferner können einem Element oder einer Gruppe von Elementen zusätzliche Daten im Zusammenhang mit einer Sicherheitsebene zugeordnet werden, zum Beispiel ob jeder Zugriff verboten ist, ob Read-only-Zugriff gestattet ist, unterschiedliche Sicherheitsebenen für verschiedene Per-

sonen und/oder Freigabestufen etc. Außerdem kann bei manchen Implementierungen dem Nutzer die Wahl verschiedener Sicherheitsebenen für Unterelemente eines Elements gestattet werden. Generell können Zugriffsschutzdaten (z.B. Passwort, biometrische Daten, im Voraus genehmigte Sicherheitsebenen für den Zugriff, eine Auflistung befugter Personen und/oder IDs etc.) gespeichert und einem Prozessanlagenelement zugeordnet werden, und diese Zugriffsschutzdaten können mit Nutzersicherheitsdaten (z.B. ein von einem Nutzer eingegebenes Passwort, die ID des Nutzers, die Log-In-ID des Nutzers, vom Nutzer erhaltene biometrische Daten etc.) verglichen werden, um festzustellen, ob dem Nutzer der Zugriff auf das Element und/oder bestimmte Unterelemente des Elements gestattet ist.

[0151] In [Fig. 17](#) können, wenn gewünscht, die Blöcke **704**, **706** und **710** entfallen. Ein Nutzer könnte beispielsweise angeben, dass das Prozessanlagenelement oder dem Prozessanlagenelement zugeordnete Daten geschützt werden müssen. Dann könnte der Nutzer eine der Routine **750** in [Fig. 19](#) ähnliche Routine anwenden, um eine Gruppe von Elementen zu schützen, die das Prozessanlagenelement beinhaltet, das der Nutzer zu schützen wünscht. Auf diese Weise könnte der Nutzer andere daran hindern, die Konfiguration des Elements zu ändern, um die geschützten Daten einzusehen.

[0152] [Fig. 22](#) ist ein Flussdiagramm einer beispielhaften Routine **830** für das Exportieren eines Prozessanlagenelements. Die Routine **830** kann durch ein oder mehrere Rechnersysteme wie eine oder mehrere der Workstations **14** und/oder die Steuerungsgeräte **12** der Prozessanlage **10** in [Fig. 1](#) implementiert werden. Die Routine **830** kann auch durch ein von der Prozessanlage **10** getrenntes Rechnersystem implementiert werden. So könnte die Routine **830** beispielsweise durch ein Rechnersystem eines Drittanbieters oder ein Rechnersystem einer anderen Prozessanlage implementiert werden, um ein Prozessanlagenelement zu einem Kunden oder zur Prozessanlage **10** zu exportieren. Zum Prozessanlagenelement kann beispielsweise ein Modulobjekt, ein Modulklassenobjekt oder eine Composite-Template gehören. Die Routine **830** kann als Teil einer Konfigurationsapplikation wie z.B. der Konfigurationsapplikation **50** implementiert werden. Die Routine **830** kann aber auch als Teil einer anderen Applikation, wie beispielsweise einer Asset-Managementapplikation oder einer z.B. vom Drittanbieter angewandten Applikation implementiert werden.

[0153] In einem Block **832** kann eine Anforderung empfangen werden, ein Prozessanlagenelement zu exportieren. Ein Nutzer kann beispielsweise ein Element in einem Explorer-Fenster auswählen, um es zu exportieren. In einem anderen Beispiel kann der Nutzer eine Gruppe von Prozessanlagenelementen an-

wählen, um sie zu exportieren, und die Routine **830** kann für jedes Element der Gruppe implementiert werden.

[0154] In einem Block **834** wird festgestellt, ob mit dem Element verbundene Daten geschützt sind. Es kann beispielsweise festgestellt werden, ob eine dem Element zugeordnete Angabe vorliegt, wie die in Block **708** der [Fig. 17](#) gespeicherte Angabe, die anzeigt, dass dem Element zugeordnete Daten geschützt sind. Beispielsweise kann auch festgestellt werden, ob eine dem Element zugeordnete Angabe vorliegt, wie die in Block **758** der [Fig. 19](#) gespeicherte Angabe, die anzeigt, dass das Element Teil einer geschützten Gruppe von Elementen ist. Sind dem Element zugeordnete Daten nicht geschützt, so kann der Ablauf in einem Block **836** fortgesetzt werden, in dem das Element exportiert werden kann.

[0155] Wird jedoch in Block **834** festgestellt, dass die mit dem Element verbundenen Daten geschützt sind, kann der Ablauf zu einem Block **838** wechseln. In Block **838** können die geschützten Daten höherwertig geschützt werden. In Block **840** kann dann ein dem Element zugeordnetes Passwort verschlüsselt werden. So wird beispielsweise das in Block **706** der [Fig. 17](#) oder das in Block **756** der [Fig. 19](#) empfangene Passwort verschlüsselt.

[0156] In einem Block **842** kann das Element exportiert werden, wobei jedoch die geschützten Daten durch die verschlüsselten geschützten Daten und das Passwort durch das verschlüsselte Passwort ersetzt werden. Ist das Prozessanlagenelement ein aus einem Modulklassenobjekt erzeugtes Element, so könnten die verschlüsselten Daten und/oder das verschlüsselte Passwort beispielsweise als Teil des Modulklassenobjekts oder des Elements exportiert werden.

[0157] [Fig. 23](#) ist ein Blockdiagramm eines beispielhaften Konfigurationssystems **900**, das verwendet werden kann, um eine Versionskontrolle und Prüfpfadinformation in Verbindung mit einer Prozessanlage bereitzustellen. Weitere Aspekte von Versionskontroll- und Prüfpfadtechniken, die verwendet werden können, werden in der US-Patentschrift Nr. 6,449,624 mit dem Titel "Version Control and Audit Trail in a Process Control System" beschrieben, die für die Anmelderin dieses Patents angemeldet wurde. Es ist klar, dass die in der US-Patentschrift Nr. 6,449,624 beschriebenen Techniken nicht angewandt werden müssen und auch davon abweichende Techniken verwendet werden können. Die US-Patentschrift Nr. 6,449,624 wird hiermit durch Bezugnahme ausdrücklich zu allen Zwecken in ihrer Gesamtheit in diese Anmeldung einbezogen.

[0158] Das Konfigurationssystem **900** umfasst eine Nutzerschnittstelle **904**, die beispielsweise in ei-

ner Display-Vorrichtung der Workstation **14** ([Fig. 1](#)) oder in einer anderen Rechnervorrichtung erzeugt werden kann. Die Nutzerschnittstelle **904** kann es einem Nutzer ermöglichen, eine Prozessanlage oder einen Abschnitt einer Prozessanlage über eine oder mehrere Konfigurierungsapplikationen **906** zu konfigurieren. Zu den Konfigurierungsapplikationen **906** kann eine Konfigurierungsapplikation wie z.B. die Konfigurierungsapplikation **50** ([Fig. 1](#)) gehören. Die Nutzerschnittstelle **904** kommuniziert außerdem mit einem Versionskontroll- und Prüfpfadsystem **908** (im Folgenden "das VCAT-System"), das generell mit den Konfigurierungsapplikationen **906** zusammenwirkt, um zurückliegende Informationen bezüglich der Konfigurierung der Prozessanlage oder des Abschnitts der Prozessanlage zu protokollieren und zu verwalten. Sowohl die Konfigurierungsapplikationen **906** als auch das VCAT-System **908** haben Zugriff auf eine Konfigurationsdatenbank **912**, in der für eine aktuelle Konfiguration der Prozessanlage oder des Abschnitts der Prozessanlage (im Folgenden die "Prozesskonfiguration") repräsentative Daten gespeichert sein können und kommunizieren mit dieser. Das VCAT-System **908** kommuniziert ferner mit einer Versionskontrolldatenbank **910**.

[0159] Die Versionskontrolldatenbank **910** kann zurückliegende Konfigurationsdaten beinhalten, die eine Anzahl früherer Versionen jedes Elements anzeigen, die in der Prozesskonfigurierung verwendet wurden. Insgesamt können die historischen Daten aller Elemente genutzt werden, um frühere Konfigurationen des Prozesses zu rekonstruieren. Insbesondere sind für jedes Element der Konfigurationsdatenbank **912** (auch für solche, die nicht mehr in der Konfigurationsdatenbank **912** vorliegen) für die Konfiguration des betreffenden Elements repräsentative Daten für eine Vielzahl von Versionen gespeichert. Ein Element kann beispielsweise seit seiner Erzeugung dreimal verändert worden sein. Die Versionskontrolldatenbank **910** würde daher Daten enthalten, die die Konfiguration des Elements zum Zeitpunkt der Erzeugung anzeigen, die als "Version 1" bezeichnet werden kann, sowie Daten, die die Konfiguration des Elements nach jeder der drei Änderungen anzeigen und die der "Version 2", "Version 3" und "Version 4" entsprechen würden.

[0160] Die historischen Konfigurationsdaten können somit Daten enthalten, die für alle an Modulobjekten und Modulklassenobjekten vorgenommenen Änderungen repräsentativ sind. Die Änderungen können, müssen aber nicht unter Verwendung der Konfigurierungsapplikationen **906** vorgenommen werden. Die Nutzerschnittstelle **904**, die Konfigurierungsapplikationen **906** und das VCAT-System **908** können insgesamt oder in Teilen zu einem einzigen integrierten System kombiniert werden. Der Klarheit halber werden die Aufgaben hierin jedoch beschrieben, als ob sie der Nutzerschnittstelle **904**, den Konfigurierungs-

applikationen **906** und dem VCAT-System **908** getrennt zugeordnet wären.

[0161] Das VCAT-System **908** kann unter Verwendung eines oder mehrerer Rechnersysteme, wie z.B. der Workstation **14** ([Fig. 1](#)) so implementiert werden, dass eine Überwachung der Änderungen der Prozesskonfiguration möglich ist. Die Nutzerschnittstelle **904** und die Konfigurierungsapplikationen **906** können in dem/den gleichen Rechnersystemen) implementiert werden wie das VCAT-System **908**, oder in anderen Rechnersystemen, die mit dem Rechnersystem kommunizieren, die das VCAT-System **908** implementieren.

[0162] Die Daten in der einen oder in beiden Datenbanken **910**, **912** können in einem computerlesbaren Medium gespeichert werden, das physikalisch an einer beliebigen Stelle innerhalb der Prozessanlage **10** angeordnet sein kann, wie z.B. in einem mit den Workstations **14** verbundenen flüchtigen Speicher oder in einem magnetischen oder optischen Speichermedium. Alternativ können die eine oder beide Datenbanken **910**, **912** an einem entfernten Standort so gespeichert werden, dass ein Rechnersystem wie die Workstation **14** über ein Netzwerk, wie beispielsweise ein Intranet, das Internet oder ein beliebiges anderes Kommunikationsmedium Zugriff auf die darin gespeicherten Daten hat. Ferner brauchen die in jeder der Datenbanken **910**, **912** gespeicherten Daten nicht im gleichen computerlesbaren Medium gespeichert zu werden, so dass ein beliebiger Abschnitt jeder der Datenbanken **910**, **912** in einer jeweils anderen Speichervorrichtung oder auf einem anderen Medium gespeichert werden kann als in der Vorrichtung bzw. dem Medium, in der/dem andere Abschnitte gespeichert sind.

[0163] In [Fig. 23](#) wird das VCAT-System **908** als unterschiedlich und getrennt von der Versionskontrolldatenbank **910** gezeigt. Alternativ könnte die Versionskontrolldatenbank **910** einen Abschnitt des VCAT-Systems **908** bilden. In gleicher Weise können die Konfigurationsdatenbank **912** und die Versionskontrolldatenbank **910** getrennte und verschiedene Datenstrukturen bilden, was jedoch nicht zwingend ist. Das heißt, die Datenbanken **910**, **912** können im gleichen Speichermedium abgelegt sein und sogar Abschnitte einer gemeinsamen Datenbank speziell für die Prozessanlage **10** bilden. Der hierin benutzte Begriff "Datenbank" sollte daher so verstanden werden, dass er sich nicht auf eine bestimmte Datenstruktur beschränkt.

[0164] [Fig. 24](#) ist ein Flussdiagramm einer beispielhaften Routine **930** für eine einfachere Verwaltung von Versionen eines Prozessanlagenelements. Das Prozessanlagenelement kann beispielsweise Modulobjekte und/oder Modulklassenobjekte umfassen. Die Routine **930** kann z.B. durch das VCAT-System

908 der [Fig. 23](#) implementiert werden und wird unter Bezug auf [Fig. 23](#) erläutert.

[0165] In einem Block **932** kann eine Änderung eines Prozessanlagenelements detektiert werden. Beispielsweise kann ein Nutzer versuchen, der Prozesskonfiguration unter Verwendung der Konfigurierungsapplikationen **906** ein neues Element hinzuzufügen. Als weiteres Beispiel kann der Nutzer die Sicherung eines Elements in der Konfigurationsdatenbank **912** anfordern, nachdem er dieses verändert hat. Als noch ein weiteres Beispiel kann der Nutzer das Eintragen eines Elements in die Konfigurationsdatenbank **912** anfordern, nachdem es von ihm verändert wurde. Die US-Patentschrift Nr. 6,449,624 beschreibt Techniken im Zusammenhang mit den Verfahren zum Eintragen und Entfernen, die angewandt werden können.

[0166] In einem Block **934** kann festgestellt werden, welche Modulklassenobjekte, wenn überhaupt, genutzt wurden, um das geänderte Element zu erzeugen. So könnte z.B. ein erstes Modulklassenobjekt genutzt worden sein, um ein zweites Modulklassenobjekt zu erzeugen, das seinerseits genutzt wurde, um das Element zu erzeugen. Wurden ein oder mehrere Modulklassenobjekte genutzt, um das veränderte Element zu erzeugen, so kann in Block **936** die Version jedes genutzten Modulklassenobjekts festgestellt werden.

[0167] In einem Block **938** wird ein Versionsidentifizierer für das geänderte Element erzeugt. Der Versionsidentifizierer kann Ziffern, Buchstaben, Symbole etc. beinhalten. Der Versionsidentifizierer zeigt die Version an, die der in Block **932** detektierten Änderung entspricht. War die in Block **932** detektierte Änderung z.B. die Erzeugung des Elements, so kann der Versionsidentifizierer eine Ursprungsversion bezeichnen. Wenn es sich als weiteres Beispiel bei der in Block **932** detektierten Änderung um eine Änderung einer ersten Version des Elements handelt, kann der Versionsidentifizierer eine zweite Version bezeichnen.

[0168] In einem Block **940** kann der in Block **938** erzeugte Versionsidentifizierer gespeichert und dem geänderten Element zugeordnet werden. Handelt es sich bei dem Element beispielsweise um ein Objekt, so kann der Versionsidentifizierer als Teil dieses Objekts gespeichert werden. Als weiteres Beispiel könnte der Versionsidentifizierer getrennt vom Objekt gespeichert werden und ein Link zum Versionsidentifizierer könnte als Teil des Objekts gespeichert werden.

[0169] In einem Block **942** kann der Versionsidentifizierer des in Block **936** bestimmten Modulklassenobjekts bzw. mehrerer Modulklassenobjekte, aus dem bzw. denen, wenn überhaupt, das Element er-

zeugt wurde, gespeichert und dem veränderten Element zugeordnet werden. Handelt es sich bei dem Element beispielsweise um ein Objekt, so kann der Versionsidentifizierer als Teil dieses Objekts gespeichert werden. Als weiteres Beispiel könnten der/die Versionsidentifizierer getrennt vom Objekt gespeichert und ein Link oder Links zum/zu den Versionsidentifizierer(n) als Teil des Objekts gespeichert werden. Als ein weiteres Beispiel könnte der in Block **938** erzeugte Versionsidentifizierer unter Verwendung der in Block **936** bestimmten Versionsidentifizierer erzeugt werden. Die in Block **936** bestimmten Versionsidentifizierer könnten beispielsweise in die in Block **938** erzeugten Versionsidentifizierers verschlüsselt werden. In diesem Fall könnte Block **940** Block **942** beinhalten.

[0170] [Fig. 25](#) ist ein Flussdiagramm einer beispielhaften Routine **948** für eine einfachere Verwaltung der Versionen von Prozessanlagenelementen. Die Prozessanlagenelemente können beispielsweise Modulobjekte und/oder Modulklassenobjekte umfassen. Die Routine **948** kann z.B. durch das VCAT-System **908** der [Fig. 23](#) implementiert werden.

[0171] In einem Block **950** kann bestimmt werden, ob eine neue Version eines Modulklassenobjekts erzeugt wurde. Es könnte beispielsweise detektiert werden, wann die neue Version eines Modulklassenobjekts eingetragen wurde. In einem Block **952** können Prozessanlagenelemente bestimmt werden, die unter Nutzung des Modulklassenobjekts erzeugt wurden. Unter Bezugnahme auf [Fig. 6](#) könnte beispielsweise der Speicher **526** geprüft werden, um festzustellen, welche Modulobjekte aus dem Modulklassenobjekt erzeugt wurden. Außerdem könnte der Speicher **526** oder ein anderer Speicher oder Speicherabschnitt geprüft werden, um zu bestimmen, welche anderen Modulklassenobjekte aus dem Modulklassenobjekt erzeugt wurden.

[0172] In einem Block **954** können die neue Version des Modulklassenobjekts oder die am Modulklassenobjekt vorgenommenen Änderungen auf die in Block **952** bestimmten Elemente übertragen werden. Dann kann in einem Block **958** der Versionsidentifizierer der neuen Version des Modulklassenobjekts gespeichert und den in Block **952** bestimmten Elementen zugeordnet werden. Als ein Beispiel werden die Versionsidentifizierer der in Block **952** bestimmten Elemente nicht zur Verfügbarmachung der neuen Versionen der Elemente aktualisiert. Optional werden in einem anderen Beispiel die Versionsidentifizierer der in Block **952** bestimmten Elemente in einem Block **959** aktualisiert. Danach können die neuen Versionsidentifizierer der Elemente gespeichert und den Elementen zugeordnet werden.

[0173] Die in den Blöcken **940** und **942** der [Fig. 24](#) und in den Blöcken **958** und **959** der [Fig. 25](#) spei-

cherten Versionsidentifizierer können einen Nutzer dabei unterstützen, zurückzuverfolgen welche Versionen der Elemente und welche Versionen der Modulklassenobjekte, aus denen die Elemente erzeugt wurden, in einer aktuellen Konfiguration verwendet werden. Beispielsweise könnte eine Konfigurierungsapplikation, eine Asset-Managementapplikation etc. die Konfigurationsdatenbank **912** ([Fig. 23](#)) abfragen, um die Versionen der Elemente und die Versionen der Modulklassenobjekte zu bestimmen, aus denen die Elemente erzeugt wurden. Als weiteres Beispiel könnte eine Applikation zur Fehlerbereinigung, eine Überwachungsapplikation etc. einen Speicher eines Steuerungsgeräts **12** prüfen, um beispielsweise die Versionen der Elemente und die Versionen der Modulklassenobjekte zu bestimmen, aus denen die Elemente erzeugt wurden.

[0174] [Fig. 26](#) ist ein Flussdiagramm einer beispielhaften Routine **960** für eine einfachere Konfigurierung eines Prozessanlagenelements. Die Routine **960** kann z.B. implementiert werden, wenn ein Prozessanlagenelement durch eine Konfigurierungsapplikation in einer Bildschirmausgabe zur Anzeige gebracht wird. Das Prozessanlagenelement kann beispielsweise Modulobjekte und/oder Modulklassenobjekte umfassen. Die Routine **960** kann z.B. durch die Konfigurierungsapplikationen **906** und/oder das VCAT-System **908** der [Fig. 23](#) implementiert werden und wird unter Bezug auf [Fig. 23](#) erläutert.

[0175] In einem Block **962** kann bestimmt werden, welche Modulklassenobjekte, wenn überhaupt, genutzt wurden, um das Element zu erzeugen. Beispielsweise könnte ein erstes Modulklassenobjekt genutzt worden sein, um ein zweites Modulklassenobjekt zu erzeugen, das seinerseits verwendet wurde, um das Element zu erzeugen. Wenn ein oder mehrere Modulklassenobjekte genutzt wurden, um das Element zu erzeugen, kann in Block **964** eine aktuelle Version jedes dieser Modulklassenobjekte bestimmt werden. In einem Block **966** kann die Version jedes der Modulklassenobjekte bestimmt werden, die genutzt wurden, um das Element zu erzeugen. Beispielsweise könnten die in Block **942** der [Fig. 24](#) gespeicherten und dem Element zugeordneten Versionsidentifizierer abgerufen werden.

[0176] In einem Block **968** kann für jedes der für die Erzeugung des Elements genutzten Modulklassenobjekte bestimmt werden, ob sich die aktuelle Version des Modulklassenobjekts von der Version des Modulklassenobjekts unterscheidet, die genutzt wurde, um das Element zu erzeugen. Gibt es keine Unterschiede, so kann die Routine beendet werden. Werden jedoch ein oder mehrere Unterschiede detektiert, so kann in Block **970** eine Nachricht erzeugt werden. Im Allgemeinen kann die Nachricht anzeigen, dass eine neue Version eines für die Erzeugung des Elements genutzten Modulklassenobjekts verfügbar ist.

Die Nachricht kann Text, Symbole, Farben, ein akustisches Signal etc. beinhalten. Beispielsweise kann ein Textfeld oder -fenster neben oder überlappend mit einer graphischen Darstellung des Elements auf einem Konfigurationsbildschirm angezeigt werden. Als weiteres Beispiel kann ein Symbol in oder in der Nähe der Darstellung des Elements zur Ansicht gebracht werden. Als noch ein anderes Beispiel kann die Farbe, die Helligkeit etc. der Darstellung des Elements verändert werden. Die Nachricht selbst kann anzeigen, für welche Modulklassenobjekte eine neue Version verfügbar ist. Alternativ kann die Nachricht dem Nutzer zwar nicht anzeigen, für welche Modulklassenobjekte eine neue Version verfügbar ist, aber der Nutzer kann zusätzliche Schritte ausführen, um diese Information beispielsweise durch die Anwahl eines Menuelements, Doppelklicken auf ein Symbol, Hineinklicken in das Element selbst etc. zu erhalten.

[0177] Optional kann dem Nutzer dann eine Nutzerschnittstellenführung bereitgestellt werden, um einen Prozess für die Änderung des Prozessanlagenelements durch die Aktualisierung des Elements unter Nutzung der neuen Versionen der Modulklassenobjekte zu initiieren. Dem Nutzer kann beispielsweise eine Bedientaste zur Verfügung gestellt werden, um den Prozess zu starten. Als weiteres Beispiel kann für den Nutzer eine Nutzerschnittstellenführung bereitgestellt werden, um zu wählen, welche von möglicherweise mehreren Modulklassenobjekten, die neue Versionen haben, genutzt werden sollen, um das Prozessanlagenelement zu aktualisieren. Dem Nutzer kann es beispielsweise möglich sein, Kästchen zu markieren, graphische Darstellungen hervorzuheben etc., die den Modulklassenobjekten mit neuen Versionen entsprechen, die genutzt werden sollen, um das Prozessanlagenelement zu aktualisieren.

[0178] [Fig. 27](#) ist ein Flussdiagramm einer beispielhaften Routine **980** für eine einfachere Konfigurierung von Prozessanlagenelementen. Die Routine **980** kann beispielsweise implementiert werden, wenn Prozessanlagenelemente geändert und/oder eingetragen werden. Das Prozessanlagenelement kann z.B. ein Modulklassenobjekt umfassen. Die Routine **980** kann beispielsweise durch die Konfigurierungsapplikationen **906** und/oder das VCAT-System **908** der [Fig. 23](#) implementiert werden und wird unter Bezug auf [Fig. 23](#) erläutert.

[0179] In einem Block **982** kann festgestellt werden, ob eine neue Version eines Modulklassenobjekts erzeugt wurde. Es könnte beispielsweise detektiert werden, wann eine neue Version eines Modulklassenobjekts eingetragen wurde. In einem Block **984** können die Prozessanlagenelemente bestimmt werden, die unter Nutzung des Modulklassenobjekts erzeugt wurden. Unter Bezug auf [Fig. 6](#) könnte beispielsweise der Speicher **526** geprüft werden, um die

Modulobjekte festzustellen, die aus dem Modulklassenobjekt erzeugt wurden. Der Speicher **526** oder ein anderer Speicher oder Speicherabschnitt könnten auch geprüft werden, um festzustellen, welche anderen Modulklassenobjekte aus dem Modulklassenobjekt erzeugt wurden.

[0180] In einem Block **986** könnte der Nutzer aufgefordert werden anzugeben, ob die Änderungen an den Modulklassenobjekten auf eines oder mehrere der in Block **984** bezeichneten Prozessanlagenelemente übertragen werden sollen. Es kann z.B. ein Textfeld, ein Fenster, ein Bildschirm, ein akustisches Signal etc. genutzt werden, um den Nutzer zu dieser Angabe aufzufordern.

[0181] Optional kann für den Nutzer eine Nutzerschnittstellenführung bereitgestellt werden, um spezifische, in Block **984** bezeichnete Prozessanlagenelemente auszuwählen, auf die Änderungen übertragen werden sollen. Dazu kann es dem Nutzer ermöglicht werden, Kästchen zu markieren, graphische Darstellungen hervorzuheben etc., die den Prozessanlagenelementen entsprechen, auf die Änderungen übertragen werden sollen.

[0182] In einem Block **988** wird eine Nutzerreaktion auf die von Block **986** erzeugte Aufforderung erhalten. In einem Block **990** kann bestimmt werden, ob die Nutzerreaktion angibt, dass die Änderungen übertragen werden sollen. Sollen die Änderungen nicht übertragen werden, so kann die Routine beendet werden. Sollen die Änderungen übertragen werden, so werden in einem Block **992** die Änderungen auf die in Block **984** bezeichneten Prozessanlagenelemente übertragen. Optional können die Änderungen auf ausgewählte der in Block **984** bezeichneten Prozessanlagenelemente übertragen werden.

[0183] Obwohl die [Fig. 17](#), [Fig. 19-Fig. 22](#) und [Fig. 24-Fig. 27](#) in Bezug auf Prozessanlagenelemente beschrieben wurden, können ähnliche Routinen genutzt werden, um den Zugriff auf Prozessanlagenelemente zu begrenzen, an diesen Elementen vorgenommene Änderungen zurückzuverfolgen und Änderungen in Bezug auf Sequenzen von Steuerungsschritten, die von diesen Elementen ausgeführt werden sollen, zu übertragen. Den Routinen der [Fig. 17](#), [Fig. 19-Fig. 22](#) und [Fig. 24-Fig. 27](#) ähnliche Routinen können z.B. genutzt werden, um den Zugriff auf Phasenklassen, Einheitenphasen und/oder Gruppen von Phasenklassen oder Einheitenphasen zu begrenzen, daran vorgenommene Änderungen zurückzuverfolgen oder Änderungen zu übertragen.

[0184] [Fig. 28](#) ist eine beispielhafte Bildschirmdarstellung **1000** einer Nutzerschnittstelle für die Konfigurierung eines Modulklassenobjekts. Im Bildschirm **1000** werden Details eines Modulklassenobjekts mit der Bezeichnung "CALCULATION" gezeigt. Der Bild-

schirm **1000** umfasst einen Editierungsabschnitt **1004** für die Editierung des Objekts. Wie in [Fig. 28](#) gezeigt, umfasst das Modulklassenobjekt CALCULATION einen Multipliziererfunktionsblock **1008** und einen Addiererfunktionsblock **1012**. Der Bildschirm **1000** beinhaltet außerdem einen Abschnitt **1020**, der Parameter des Modulklassenobjekts CALCULATION anzeigt, die nicht versteckt sind, wenn der Zugriff auf das Modulklassenobjekt CALCULATION begrenzt ist. Die nicht versteckten Parameter umfassen einen Versionsidentifizierer **1026**, der eine Version eines Modulobjekts anzeigt, das aus dem Modulklassenobjekt CALCULATION erzeugt wurde und einen Versionsidentifizierer **1028**, der eine Version des Modulklassenobjekts CALCULATION anzeigt.

[0185] [Fig. 29](#) ist eine weitere beispielhafte Bildschirmdarstellung **1030** einer Nutzerschnittstelle, um Parameter des Modulklassenobjekts CALCULATION zu bestimmen, die nicht versteckt werden. Der Bildschirm **1030** umfasst die Tastensymbole **1032** und **1034** für das Hinzufügen bzw. das Löschen von Parametern einer Gruppe von Parametern, die nicht versteckt sind. Die Bildschirmdarstellung **1030** zeigt, dass ein Parameter MULTIPLIER **1038** zu der Gruppe von Parametern hinzugefügt wird, die nicht versteckt sind. Mit Bezug auf [Fig. 28](#) entspricht der Parameter MULTIPLIER **1038** auch dem Eingang "IN2" des Multipliziererfunktionsblocks **1008**.

[0186] [Fig. 30](#) ist eine beispielhafte Bildschirmdarstellung **1040** einer Nutzerschnittstelle für eine Applikation zur Fehlerbereinigung. Im Bildschirm **1040** werden Details eines Modulobjekts CALCULATION_1 zur Ansicht gebracht. Das Modulobjekt CALCULATION_1 wurde aus dem Modulklassenobjekt CALCULATION erzeugt. In diesem Beispiel wurde der Zugriff auf das Modulklassenobjekt CALCULATION nicht begrenzt. Der Bildschirm **1040** zeigt daher generell Details der Unterelemente, aus denen das Modulobjekt CALCULATION_1 aufgebaut ist. Beispielsweise zeigt ein Abschnitt **1044**, dass das Modulobjekt CALCULATION_1 den Multipliziererfunktionsblock **1008** und den Addiererfunktionsblock **1012** beinhaltet. Der Bildschirm **1040** umfasst ferner einen Abschnitt **1048**, der Parameter des Modulobjekts CALCULATION_1 anzeigt. Das Modulobjekt CALCULATION_1 beinhaltet beispielsweise die Versionsidentifizierer **1026** und **1028** und den Parameter MULTIPLIER **1038**. Ferner umfasst der Bildschirm **1040** auch einen Abschnitt **1050**, der die Funktionsblöcke anzeigt, aus denen sich das Modulobjekt CALCULATION_1 zusammensetzt. Insbesondere zeigt der Abschnitt **1050**, dass das Modulobjekt CALCULATION_1 einen Addiererfunktionsblock und einen Multipliziererfunktionsblock beinhaltet.

[0187] [Fig. 31](#) ist die beispielhafte Bildschirmdarstellung **1040** für den Fall, dass der Zugriff auf das Modulklassenobjekt CALCULATION begrenzt wurde.

Der Bildschirm **1040** zeigt daher generell nicht alle Details der Unterelemente, aus denen sich das Modulobjekt CALCULATION_1 zusammensetzt. Die Abschnitte **1044** und **1050** zeigen beispielsweise nicht, dass das Modulobjekt CALCULATION_1 den Multipliziererfunktionsblock **1008** und den Addiererfunktionsblock **1012** beinhaltet. Abschnitt **1048** zeigt jedoch nach wie vor die Parameter des Modulobjekts CALCULATION_1 an.

[0188] Es ist für den Fachmann klar, dass verschiedene Variationen der hierin beschriebenen Techniken angewandt werden können. So kann beispielsweise die Reihenfolge der Blöcke verändert werden, Blöcke können entfallen, neue Blöcke können hinzugefügt werden etc. Außerdem könnten verschiedene hierin beschriebene Techniken kombiniert werden. Eine der Routine **700** in [Fig. 17](#) ähnliche Routine könnte z.B. genutzt werden, um einige einem Modulklassenobjekt zugeordnete Daten zu schützen. Dann könnte eine Modultemplate aus dem Modulklassenobjekt erzeugt werden. Als Nächstes könnte eine der Routine **700** der [Fig. 17](#) ähnliche Routine genutzt werden, um einige einer Modultemplate zugeordnete Daten zu schützen, wenn diese nicht im Zusammenhang mit dem Modulklassenobjekt geschützt wurde. Anschließend könnte eine besondere Instanz der Modultemplate erzeugt werden, die den in Zusammenhang mit dem Modulklassenobjekt und der Modultemplate spezifizierten Schutz umfasst.

[0189] Nach der Implementierung kann jede hierin beschriebene Software in einem computerlesbaren Speicher abgelegt werden, wie z.B. einer Magnetplatte, einer Laserplatte oder einem anderen Speichermedium, in einem RAM oder ROM eines Rechners oder Prozessors etc. Ebenso kann diese Software für einen Nutzer, eine Prozessanlage oder eine Bediener-Workstation bereitgestellt werden, indem bekannte oder gewünschte Bereitstellungsverfahren genutzt werden, z.B. eine computerlesbare Platte oder eine andere transportable Computerspeichereinrichtung, oder die Übertragung über einen Kommunikationskanal wie z.B. eine Telefonleitung, das Internet, das World Wide Web, ein beliebiges anderes lokales Netzwerk (LAN) oder Fernnetz (WAN) etc. (wobei diese Bereitstellung als die gleiche oder als eine mit der Bereitstellung dieser Software auf einem transportablen Speichermedium austauschbare Methode gilt). Ferner kann diese Software direkt ohne Modulation oder Verschlüsselung bereitgestellt oder unter Nutzung einer geeigneten Modulationsträgerwelle und/oder Verschlüsselungstechnik vor der Übertragung über einen Kommunikationskanal moduliert und/oder verschlüsselt werden.

[0190] Während die vorliegende Erfindung unter Bezugnahme auf spezifische Ausführungsformen beschrieben worden ist, die die Erfindung nur verdeutlichen und nicht einschränken sollen, wird es für den

Fachmann offensichtlich sein, daß Änderungen, Hinzufügungen oder Weglassungen an den offenbarten Ausführungsformen vorgenommen werden können, ohne vom Geist und Geltungsbereich der Erfindung abzuweichen.

Patentansprüche

1. Objektentität zur Nutzung bei der Konfigurierung einer Prozessanlage, wobei die Objektentität aufweist:

einen computerlesbaren Speicher;
ein Klassenobjekt, das in dem computerlesbaren Speicher gespeichert ist, wobei das Klassenobjekt eine Prozessentität innerhalb der Prozessanlage generisch repräsentiert und dabei umfasst:
einen ersten Speicherbereich, der dafür ausgelegt ist, Merkmale eines oder mehrerer Prozesselemente, die die Prozessentität bilden, zu speichern;
einen zweiten Speicherbereich, der dafür ausgelegt ist, eine Referenz zu einem oder mehreren Modulobjekten zu speichern, wobei jedes Modulobjekt aus dem Klassenobjekt erzeugt wurde, um Merkmale eines oder mehrerer spezifischer Prozesselemente einzubeziehen, die dem einen oder den mehreren Prozesselementen entsprechen, aus denen die Prozessentität besteht, und die mit dem einen oder den mehreren spezifischen Prozesselementen verbundene Programmierung, die im Betrieb der Prozessanlage ausgeführt werden soll; und
einen dritten Speicherbereich, der dafür ausgelegt ist, Daten mit Angaben zu einer Version des Klassenobjekts zu speichern.

2. Objektentität nach Anspruch 1, die außerdem aufweist:

einen vierten Speicherbereich, der dafür ausgelegt ist, eine Referenz zu einem oder mehreren anderen Modulklassenobjekten zu speichern, aus denen das Klassenobjekt erzeugt wurde; und
einen fünften Speicherbereich, der dafür ausgelegt ist, Daten zu speichern, die Angaben zu den jeweiligen Versionen des einen oder mehrerer dieser anderen Modulklassenobjekte enthalten.

3. Objektentität nach Anspruch 1, wobei die Prozessentität eine Einheit umfasst und wobei das eine oder mehrere Prozesselemente ein oder mehrere Unterelemente der Einheit aufweisen.

4. Objektentität nach Anspruch 1, wobei die Prozessentität eine Ausrüstungsentität umfasst, und wobei das eine oder mehrere Prozesselemente ein oder mehrere Unterelemente der Ausrüstungsentität beinhalten.

5. Objektentität nach Anspruch 1, wobei die Prozessentität ein Steuerungselement beinhaltet und wobei das eine oder mehrere Prozesselemente eine generische Steuerungsroutine beinhalten, die ausge-

legt ist, um für die Bereitstellung von Steuerungsmaßnahmen innerhalb der Prozessanlage genutzt zu werden.

6. Objektentität nach Anspruch 1, wobei die Prozessentität ein Displayelement umfasst und wobei das eine oder mehrere Prozesselemente eine generische Displayroutine beinhalten, die ausgelegt ist, um für die Bereitstellung von Displayaktivitäten für einen Nutzer innerhalb der Prozessanlage genutzt zu werden.

7. Objektentität zur Nutzung für die Konfigurierung einer Prozessanlage, bei der die Objektentität aufweist:

einen computerlesbaren Speicher;

ein im computerlesbaren Speicher gespeichertes Klassenobjekt, wobei das Klassenobjekt spezifisch eine Prozessentität innerhalb der Prozessanlage repräsentiert und umfasst:

einen ersten Speicherbereich, der dafür ausgelegt ist, Merkmale einer oder mehrerer Funktionen zu speichern, die im Betrieb der Prozessanlage von der Prozessentität ausgeführt werden sollen;

einen zweiten Speicherbereich, der dafür ausgelegt ist, Daten mit Angaben zu einer Version des Klassenobjekts zu speichern;

einen dritten Speicherbereich, der dafür ausgelegt ist, eine Referenz zu einem oder mehreren Modulklassenobjekten zu speichern, aus denen das Klassenobjekt erzeugt wurde; und

einen vierten Speicherbereich, der dafür ausgelegt ist, Daten mit Angaben zu den jeweiligen Versionen des einen oder mehrerer Modulklassenobjekte zu speichern.

8. Objektentität nach Anspruch 7, wobei die Prozessentität eine Einheit umfasst und wobei das eine oder mehrere Prozesselemente ein oder mehrere Unterelemente der Einheit beinhalten.

9. Objektentität nach Anspruch 7, wobei die Prozessentität eine Ausrüstungsentität umfasst und wobei das eine oder mehrere Prozesselemente ein oder mehrere Unterelemente der Ausrüstungsentität umfassen.

10. Objektentität nach Anspruch 7, wobei die Prozessentität ein Steuerungselement beinhaltet und wobei das eine oder mehrere Prozesselemente eine generische Steuerungsroutine beinhalten, die ausgelegt ist, um für die Bereitstellung von Steuerungsmaßnahmen innerhalb der Prozessanlage genutzt zu werden.

11. Objektentität nach Anspruch 7, wobei die Prozessentität ein Displayelement beinhaltet und wobei das eine oder mehrere Prozesselemente eine generische Displayroutine umfassen, die ausgelegt ist, um für die Bereitstellung von Displayaktivitäten für einen

Nutzer innerhalb der Prozessanlage genutzt zu werden.

12. Verfahren zur Überwachung von Änderungen in einer Konfiguration mindestens eines Abschnitts einer Prozessanlage, wobei das Verfahren aufweist: Detektierung einer Änderung eines Modulklassenobjekts, das generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann; Bestimmung eines oder mehrerer Modulobjekte, die aus dem Modulklassenobjekt erzeugt wurden, wobei jedes Modulklassenobjekt generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann; Übertragung der Änderung des Modulklassenobjekts auf das eine oder mehrere Modulobjekte; Erzeugung eines Versionsidentifizierers für das geänderte Modulklassenobjekt; und Zuordnung des Versionsidentifizierers für das geänderte Modulklassenobjekt zu jedem von dem einen oder den mehreren Modulobjekten.

13. Verfahren nach Anspruch 12, das ferner die Erzeugung der jeweiligen Versionsidentifizierer für das eine oder mehrere Modulobjekte beinhaltet; und Zuordnung der Versionsidentifizierer für das eine oder mehrere Modulobjekte zu dem bzw. den entsprechenden Modulobjekten.

14. Verfahren nach Anspruch 12, wobei mit dem Detektieren der Änderung des Modulklassenobjekts detektiert wird, dass das geänderte Modulklassenobjekt in einer Konfigurationsdatenbank gesichert wurde.

15. Verfahren nach Anspruch 12, wobei mit dem Detektieren der Änderung des Modulklassenobjekts detektiert wird, dass das geänderte Modulklassenobjekt über ein Versionskontrollsystem eingecheckt wurde.

16. Verfahren nach Anspruch 12, wobei die Zuordnung des Versionsidentifizierers für das geänderte Modulklassenobjekt zu jedem von dem einen oder den mehreren Modulobjekten die Speicherung des Versionsidentifizierers für das geänderte Modulklassenobjekt als Teil von jedem von dem einen oder den mehreren Modulobjekten umfasst.

17. Verfahren nach Anspruch 12, wobei zur Zuordnung des Versionsidentifizierers für das geänderte Modulklassenobjekt zu jedem von dem einen oder den mehreren Modulobjekten die Verbindung von jedem von dem einen oder den mehreren Modulobjekten mit dem Versionsidentifizierer für das geänderte Modulklassenobjekt gehört.

18. Konfigurierungsmanagementsystem für die Überwachung von Änderungen in einer Konfigurati-

on, die einer Prozessanlage zugeordnet ist, das aufweist:

ein computerlesbares Medium;
einen mit dem computerlesbaren Medium kommunikativ verbundenen Prozessor;
eine Datenbank, die erste Daten und zweite Daten speichert, wobei die ersten Daten für ein Modulklassenobjekt repräsentativ sind, das generisch eine Prozessentität innerhalb der Prozessanlage repräsentieren kann, und wobei die zweiten Daten repräsentativ sind für ein oder mehrere Prozessanlagenelemente, die aus dem Modulklassenobjekt erzeugt wurden;
Konfigurierungs- und Versionskontrollroutinen, die im computerlesbaren Medium gespeichert und konfiguriert sind, um vom Prozessor ausgeführt zu werden, um die Änderung des Modulklassenobjekts zu vereinfachen;
eine Änderung des Modulklassenobjekts zu detektieren;
das eine oder mehrere Prozessanlagenelemente zu bestimmen, die aus dem Modulklassenobjekt erzeugt wurden;
das eine oder mehrere Prozessanlagenelemente auf Basis des geänderten Modulklassenobjekts zu ändern;
einen Versionsidentifizierer für das geänderte Modulklassenobjekt zu erzeugen; und
um den Versionsidentifizierer für das geänderte Modulklassenobjekt jedem von dem einen oder den mehreren Prozessanlagenelementen zuzuordnen.

19. Konfigurierungsmanagementsystem nach Anspruch 18, wobei die Konfigurations- und Versionskontrollroutinen konfiguriert sind, um vom Prozessor ausgeführt zu werden, um jeweils Versionsidentifizierer für das eine oder mehrere Prozessanlagenelemente zu erzeugen; und um die Versionsidentifizierer für das eine oder mehrere Prozessanlagenelemente dem entsprechenden einen oder mehreren Prozessanlagenelementen zuzuordnen.

20. Konfigurierungsmanagementsystem nach Anspruch 18, wobei die Konfigurations- und Versionskontrollroutinen konfiguriert sind, um vom Prozessor ausgeführt zu werden, um die Änderung des Modulklassenobjekts zu detektieren, indem detektiert wird, ob dass das geänderte Modulklassenobjekt in der Datenbank gesichert wurde.

21. Konfigurierungsmanagementsystem nach Anspruch 18, wobei die Konfigurations- und Versionskontrollroutinen konfiguriert sind, um vom Prozessor ausgeführt zu werden, um einem Nutzer das Auschecken des Modulklassenobjekts zu ermöglichen;
einem Nutzer das Einchecken des Modulklassenobjekts zu ermöglichen;
die Änderung des Modulklassenobjekts zu detektie-

ren, indem detektiert wird, ob das geänderte Modulklassenobjekt eingecheckt wurde.

22. Konfigurierungsmanagementsystem nach Anspruch 18, wobei die Konfigurations- und Versionskontrollroutinen konfiguriert sind, um vom Prozessor ausgeführt zu werden, um den Versionsidentifizierer für das geänderte Modulklassenobjekt als Teil von jedem von dem einen oder den mehreren Modulobjekten zu speichern.

23. Konfigurierungsmanagementsystem nach Anspruch 18, wobei die Konfigurations- und Versionskontrollroutinen konfiguriert sind, um vom Prozessor ausgeführt zu werden, um jedes von dem einen oder den mehreren Modulobjekten mit dem Versionsidentifizierer für das geänderte Modulklassenobjekt zu verbinden.

24. Verfahren für die Überwachung von Änderungen in einer Konfiguration mindestens eines Abschnitts einer Prozessanlage, wobei das Verfahren aufweist:

Detektierung einer Änderung eines Prozessanlagenelements, wobei das Prozessanlagenelement eine Prozessentität innerhalb der Prozessanlage repräsentieren kann;
Bestimmung eines oder mehrerer Modulklassenobjekte, aus denen das geänderte Prozessanlagenelement erzeugt wurde, wobei jedes Modulklassenobjekt generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann;
Bestimmung jeweiliger Versionsdaten, die jedem von dem einen oder den mehreren Modulklassenobjekten zugeordnet sind;
Erzeugung eines Versionsidentifizierers für das geänderte Prozessanlagenelement;
Zuordnung des Versionsidentifizierers zum geänderten Prozessanlagenelement;
und
Zuordnen der jeweiligen Versionsdaten, die jedem von dem einen oder den mehreren Modulklassenobjekten zugeordnet sind, zu dem geänderten Prozessanlagenelement.

25. Verfahren nach Anspruch 24, wobei das Prozessanlagenelement ein Modulklassenobjekt umfasst, das generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

26. Verfahren nach Anspruch 24, wobei das Prozessanlagenelement ein Modulobjekt beinhaltet, das spezifisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

27. Verfahren nach Anspruch 24, wobei zur Zuordnung der jeweiligen, jedem von dem einen oder den mehreren Modulklassenobjekten zugeordneten Versionsdaten zu dem geänderten Prozessanlagenelement die Speicherung der jeweiligen, jedem von

dem einen oder den mehreren Modulklassenobjekten zugeordneten Versionsdaten als Teil des geänderten Prozessanlagenelements gehört.

28. Verfahren nach Anspruch 24, wobei zur Zuordnung der jeweiligen, jedem von dem einen oder den mehreren Modulklassenobjekten zugeordneten Versionsdaten zu dem geänderten Prozessanlagenelement die Verbindung des geänderten Prozessanlagenelements mit den Versionsdaten gehört, die jedem von dem einen oder den mehreren Modulklassenobjekten zugeordnet sind.

29. Verfahren nach Anspruch 24, wobei zum Detektieren der Änderung des Prozessanlagenelements gehört, eine oder mehrere Änderungen des einen oder mehrerer Modulklassenobjekte zu detektieren, aus denen das geänderte Prozessanlagenelement erzeugt wurde.

30. Verfahren nach Anspruch 24, wobei zum Detektieren der Änderung des Prozessanlagenelements gehört, eine Änderung des Prozessanlagenelements zu detektieren, die das eine oder mehrere Modulklassenobjekte nicht beeinflussen würde, aus denen das geänderte Prozessanlagenelement erzeugt wurde.

31. Konfigurierungsmanagementsystem für die Verfolgung einer Konfiguration, die einer Prozessanlage zugeordnet ist, das aufweist:
 ein computerlesbares Medium;
 einen mit dem computerlesbaren Medium kommunikativ verbundenen Prozessor;
 eine Datenbank, die erste Daten und zweite Daten speichert, wobei die ersten Daten für ein Prozessanlagenelement repräsentativ sind, das eine Prozessentität innerhalb der Prozessanlage repräsentieren kann, und wobei die zweiten Daten für eine Vielzahl von Modulklassenobjekten repräsentativ sind, die generisch für Prozessentitäten innerhalb der Prozessanlage repräsentativ sein können;
 eine im computerlesbaren Medium gespeicherte Konfigurierungsroutine, die konfiguriert ist, um vom Prozessor für eine vereinfachte Änderung des Prozessanlagenelements ausgeführt zu werden;
 eine im computerlesbaren Medium gespeicherte Versionskontrollroutine, die konfiguriert ist, um vom Prozessor ausgeführt zu werden, um:
 eine Änderung des Prozessanlagenelements zu detektieren; die jeweiligen Versionsdaten zu bestimmen, die jedem von einem oder mehreren Modulklassenobjekten zugeordnet sind, aus denen das geänderte Prozessanlagenelement erzeugt wurde;
 einen Versionsidentifizierer für das geänderte Prozessanlagenelement zu erzeugen;
 den Versionsidentifizierer dem geänderten Prozessanlagenelement zuzuordnen; und
 um die jeweiligen, jedem von dem einen oder den mehreren Modulklassenobjekten, aus denen das ge-

änderte Prozessanlagenelement erzeugt wurde, zugeordneten Versionsdaten dem geänderten Prozessanlagenelement zuzuordnen.

32. Konfigurierungsmanagementsystem nach Anspruch 31, wobei die gespeicherte Versionskontrollroutine konfiguriert ist, um vom Prozessor ausgeführt zu werden, um eine Änderung des Prozessanlagenelements zu detektieren, die durch eine oder mehrere Änderungen des einen oder mehrerer Modulklassenobjekte verursacht wurde, aus denen das geänderte Prozessanlagenelement erzeugt wurde.

33. Konfigurierungsmanagementsystem nach Anspruch 31, wobei die gespeicherte Versionskontrollroutine konfiguriert ist, um vom Prozessor ausgeführt zu werden, um eine Änderung des Prozessanlagenelements zu detektieren, die das eine oder mehrere Modulklassenobjekte nicht beeinflussen würde, aus denen das geänderte Prozessanlagenelement erzeugt wurde.

34. Konfigurierungsmanagementsystem nach Anspruch 31, wobei die Prozessanlagenelemente ein Modulobjekt umfassen, das spezifisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

35. Konfigurierungsmanagementsystem nach Anspruch 31, wobei die Prozessanlagenelemente ein Modulobjekt umfassen, das generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

36. Verfahren zur Benachrichtigung eines Nutzers von einer geänderten Konfiguration, die mindestens einem Abschnitt einer Prozessanlage zugeordnet ist, wobei das Verfahren umfasst:
 Bestimmung eines oder mehrerer Modulklassenobjekte, aus denen ein Prozessanlagenelement erzeugt wurde, wobei das Prozessanlagenelement eine Prozessentität innerhalb der Prozessanlage repräsentieren kann, und wobei das eine oder mehrere Modulklassenobjekte generisch für Prozessentitäten innerhalb der Prozessanlage repräsentativ sein können;
 Bestimmung einer jeweils aktuellen Version von jedem von dem einen oder den mehreren Modulklassenobjekten;
 Bestimmung einer jeweiligen Version von jedem von dem einen oder den mehreren Modulklassenobjekten, aus denen das Prozessanlagenelement erzeugt wurde;
 Bestimmung, ob zwischen der jeweiligen Version, aus der das Prozessanlagenelement erzeugt wurde, und der jeweils aktuellen Version für jedes von dem einen oder den mehreren Modulklassenobjekten ein Unterschied besteht; und
 Erzeugung einer Nachricht, die anzeigt, dass eine neuere Version für mindestens eines von dem einen oder den mehreren Modulklassenobjekten, aus de-

nen das Prozessanlagenelement erzeugt wurde, existiert, wenn festgestellt wird, dass zwischen der jeweiligen Version, aus der das Prozessanlagenelement erzeugt wurde, und der jeweils aktuellen Version ein Unterschied für mindestens eines von dem einen oder den mehreren Modulklassenobjekten besteht.

37. Verfahren nach Anspruch 36, wobei das Prozessanlagenelement ein Modulklassenobjekt umfasst, das generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

38. Verfahren nach Anspruch 36, wobei das Prozessanlagenelement ein Modulobjekt umfasst, das spezifisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

39. Verfahren nach Anspruch 36, das ferner die Abfrage an einen Nutzer umfasst, ob das Prozessanlagenelement auf Basis der neueren Version von dem mindestens einen von dem einen oder den mehreren Modulklassenobjekten geändert werden soll.

40. Verfahren nach Anspruch 39, bei dem außerdem das Prozessanlagenelement auf Basis der neueren Version von dem mindestens einen von dem einen oder den mehreren Modulklassenobjekten verändert wird.

41. Konfigurierungssystem zur Konfigurierung mindestens eines Abschnitts einer Prozessanlage, das aufweist:

ein computerlesbares Medium;

einen mit dem computerlesbaren Medium kommunikativ verbundenen Prozessor;

eine Datenbank, die erste Daten und zweite Daten speichert, wobei die ersten Daten für ein Prozessanlagenelement repräsentativ sind, das eine Prozessentität innerhalb der Prozessanlage repräsentieren kann, und wobei die zweiten Daten für eine Vielzahl von Modulklassenobjekten repräsentativ sind, die generisch für Prozessentitäten innerhalb der Prozessanlage repräsentativ sein können;

eine im computerlesbaren Medium gespeicherte Konfigurierungsroutine, die konfiguriert ist, um vom Prozessor ausgeführt zu werden, um eine Änderung des Prozessanlagenelements zu vereinfachen;

eine im computerlesbaren Medium gespeicherte Benachrichtigungsroutine, die konfiguriert ist, um vom Prozessor ausgeführt zu werden in der mehrere Modulklassenobjekte zu bestimmen, aus denen das Prozessanlagenelement erzeugt wurde;

eine jeweilige aktuelle Version von jedem von dem einen oder den mehreren Modulklassenobjekten zu bestimmen;

eine jeweilige Version von jedem von dem einen oder den mehreren Modulklassenobjekten zu bestimmen, aus denen das Prozessanlagenelement erzeugt wurde;

zu bestimmen, ob zwischen der jeweiligen Version, aus der das Prozessanlagenelement erzeugt wurde, und der jeweils aktuellen Version ein Unterschied für jedes von dem einen oder den mehreren Modulklassenobjekten besteht; und

um eine Nachricht zu erzeugen, die anzeigt, dass eine neuere Version von mindestens einem von dem einen oder den mehreren Modulklassenobjekten existiert, aus denen das Prozessanlagenelement erzeugt wurde, wenn festgestellt wird, dass zwischen der jeweiligen Version, aus der das Prozessanlagenelement erzeugt wurde, und der jeweils aktuellen Version für mindestens eines von dem einen oder den mehreren Modulklassenobjekten ein Unterschied besteht.

42. Konfigurierungssystem nach Anspruch 41, das außerdem eine im computerlesbaren Medium gespeicherte Aktualisierungsroutine umfasst, die konfiguriert ist, um vom Prozessor ausgeführt zu werden, um das Prozessanlagenelement auf Basis der neueren Version von dem mindestens einen von dem einen oder den mehreren Modulklassenobjekten zu aktualisieren, aus denen das Prozessanlagenelement erzeugt wurde.

43. Verfahren zur Benachrichtigung eines Nutzers von einer geänderten Konfiguration, die mindestens einem Abschnitt einer Prozessanlage zugeordnet ist, wobei das Verfahren umfasst:

Detektieren einer neuen Version eines Modulklassenobjekts, das generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann; Bestimmen eines oder mehrerer Prozessanlagenelemente, die aus einer oder mehreren älteren Versionen des Modulklassenobjekts erzeugt wurden, wobei jedes von dem einen oder den mehreren Prozessanlagenelementen jeweils eine entsprechende Prozessentität innerhalb der Prozessanlage repräsentieren kann; und

Erzeugen einer Nachricht, die anzeigt, dass eines oder mehrere der Prozessanlagenelemente aus einer oder mehreren älteren Versionen des Modulklassenobjekts erzeugt wurden.

44. Verfahren nach Anspruch 43, wobei mindestens eines von dem einen oder den mehreren Prozessanlagenelementen ein Modulklassenobjekt umfasst, das generisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

45. Verfahren nach Anspruch 43, wobei mindestens eines von dem einen oder den mehreren Prozessanlagenelementen ein Modulobjekt umfasst, das spezifisch für eine Prozessentität innerhalb der Prozessanlage repräsentativ sein kann.

46. Verfahren nach Anspruch 43, das außerdem eine Abfrage an den Nutzer umfasst, ob das eine oder mehrere Prozessanlagenelemente auf Basis

der neuen Version des Modulklassenobjekts geändert werden sollen.

47. Konfigurierungsmanagementsystem für die Verfolgung einer Konfiguration, die einer Prozessanlage zugeordnet ist, das aufweist:

- ein computerlesbares Medium;
- einen mit dem computerlesbaren Medium kommunikativ verbundenen Prozessor;
- eine Datenbank, die erste Daten und zweite Daten speichert, wobei die ersten Daten repräsentativ für ein Prozessanlagenelement sind, das eine Prozessentität innerhalb der Prozessanlage repräsentieren kann, und wobei die zweiten Daten repräsentativ sind für eine Vielzahl von Modulklassenobjekten, die generisch repräsentativ für Prozessentitäten innerhalb der Prozessanlage sein
- eine im computerlesbaren Medium gespeicherte Konfigurierungsroutine, die konfiguriert ist, um vom Prozessor für eine einfachere Änderung des Prozessanlagenelements ausgeführt zu werden;
- eine im computerlesbaren Medium gespeicherte Versionskontrollroutine, die konfiguriert ist, um vom Prozessor ausgeführt zu werden, um:
 - eine Änderung des Prozessanlagenelements zu detektieren;
 - die jeweiligen Versionsdaten zu bestimmen, die jedem von einem oder mehreren Modulklassenobjekten zugeordnet sind, aus denen das geänderte Prozessanlagenelement erzeugt wurde;
 - einen Versionsidentifizierer für das geänderte Prozessanlagenelement zu erzeugen;
 - den Versionsidentifizierer dem geänderten Prozessanlagenelement zuzuordnen; und
 - um die jeweiligen Versionsdaten, die jedem von dem einen oder den mehreren Modulklassenobjekten zugeordnet sind, aus denen das geänderte Prozessanlagenelement erzeugt wurde, dem geänderten Prozessanlagenelement zuzuordnen.

48. Konfigurierungsmanagementsystem nach Anspruch 47, wobei die gespeicherte Versionskontrollroutine konfiguriert ist, um vom Prozessor ausgeführt zu werden, um eine Änderung des Prozessanlagenelements zu detektieren, die durch eine oder mehrere Änderungen des einen oder mehrerer Modulklassenobjekte bewirkt wurde, aus denen das geänderte Prozessanlagenelement erzeugt wurde.

49. Konfigurierungsmanagementsystem nach Anspruch 47, wobei die gespeicherte Versionskontrollroutine konfiguriert ist, um vom Prozessor ausgeführt zu werden, um eine Änderung des Prozessanlagenelements zu detektieren, die das eine oder mehrere Modulklassenobjekte, aus denen das geänderte Prozessanlagenelement erzeugt wurde, nicht beeinflussen würde.

50. Konfigurierungsmanagementsystem nach Anspruch 47, wobei die Prozessanlagenelemente ein Modulobjekt umfassen, das spezifisch repräsentativ

für eine Prozessentität innerhalb der Prozessanlage sein kann.

51. Konfigurierungsmanagementsystem nach Anspruch 47, wobei die Prozessanlagenelemente ein Modulobjekt umfassen, das generisch repräsentativ für eine Prozessentität innerhalb der Prozessanlage sein kann.

Es folgen 30 Blatt Zeichnungen

Anhängende Zeichnungen

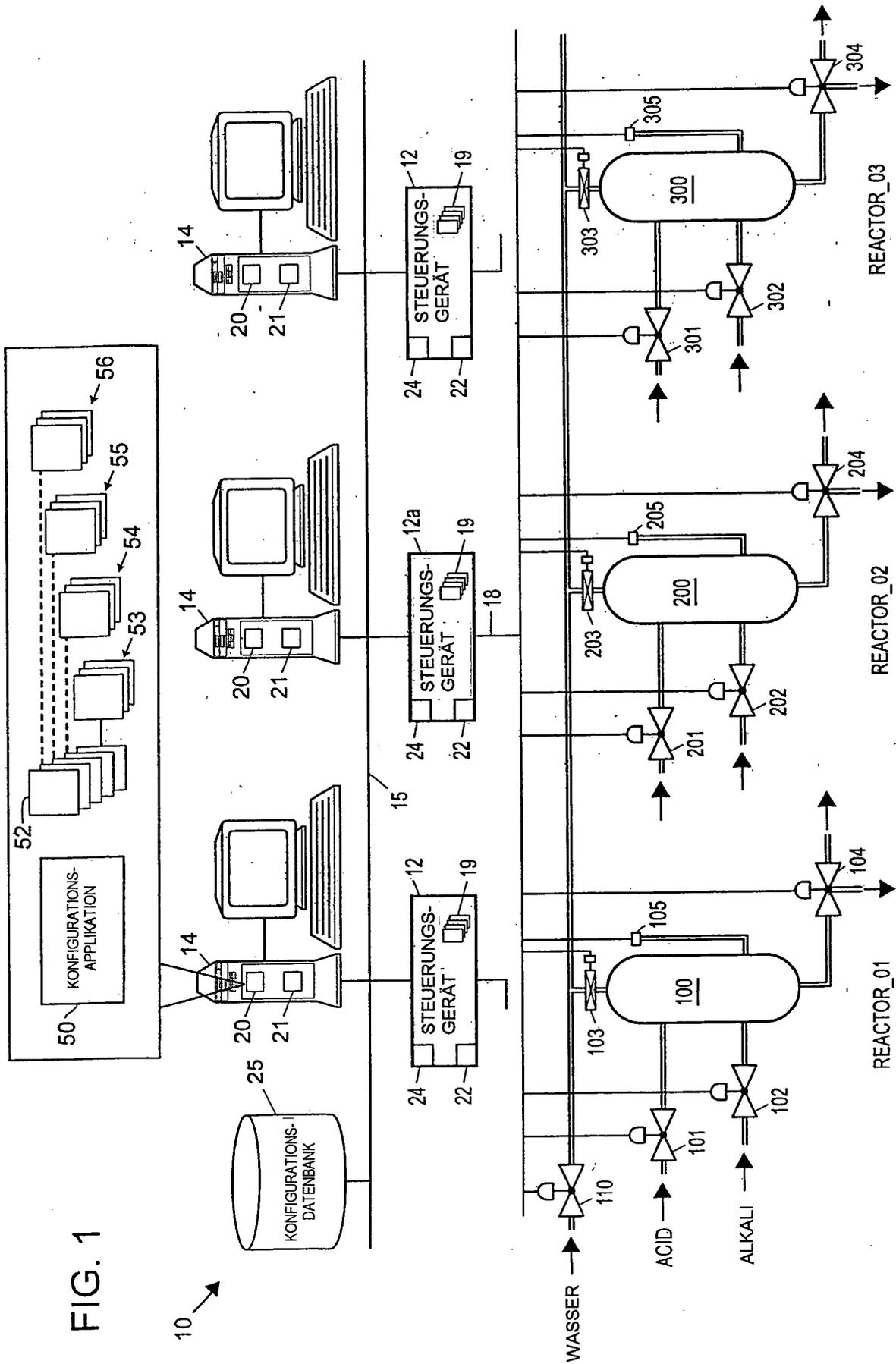


FIG. 1

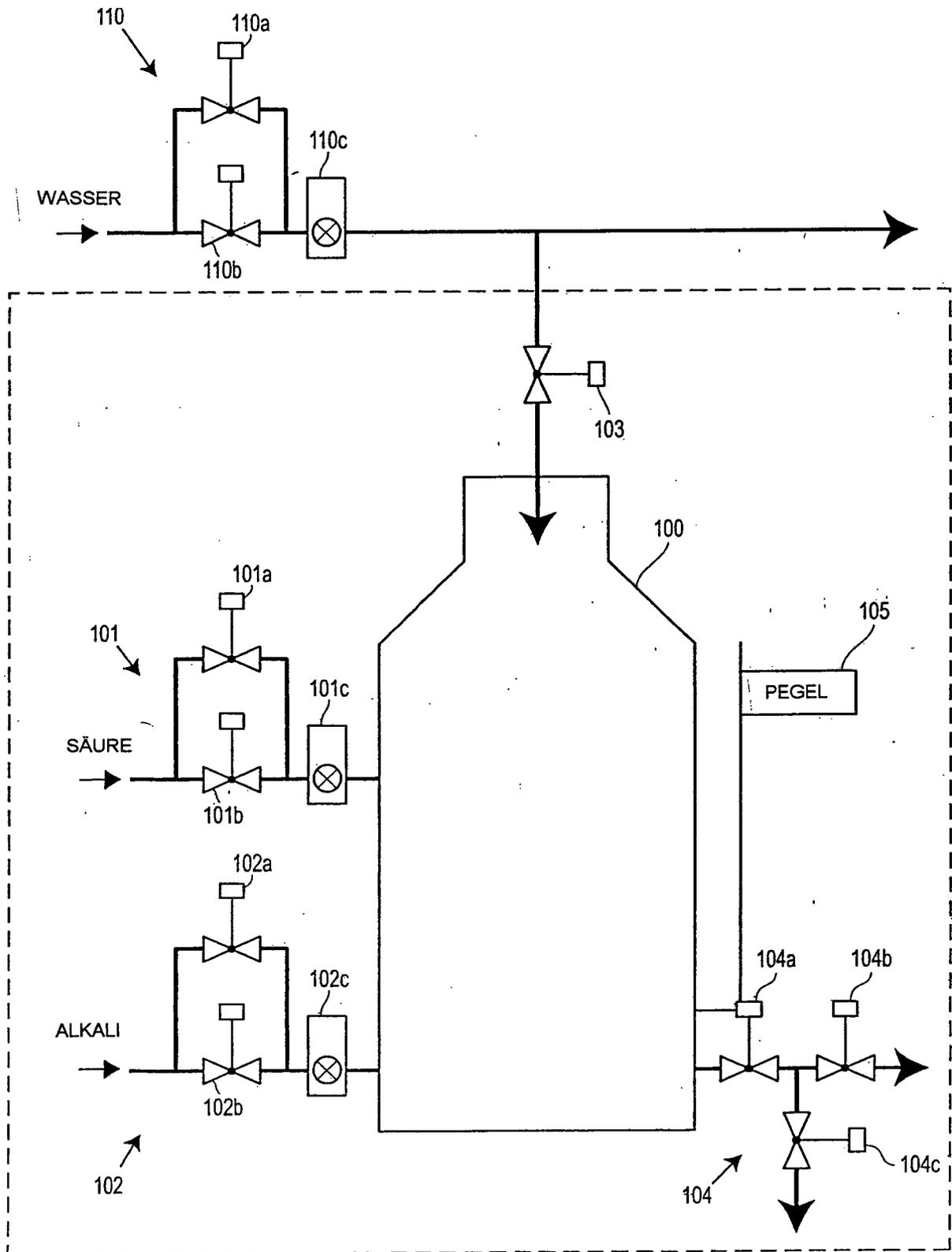


FIG. 2

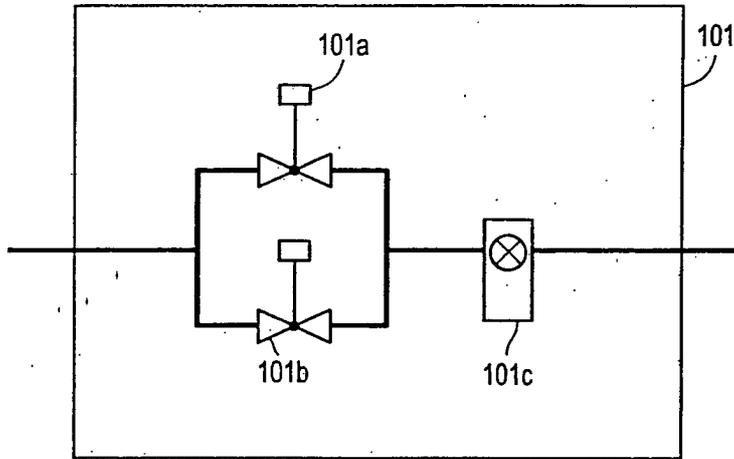


FIG. 3

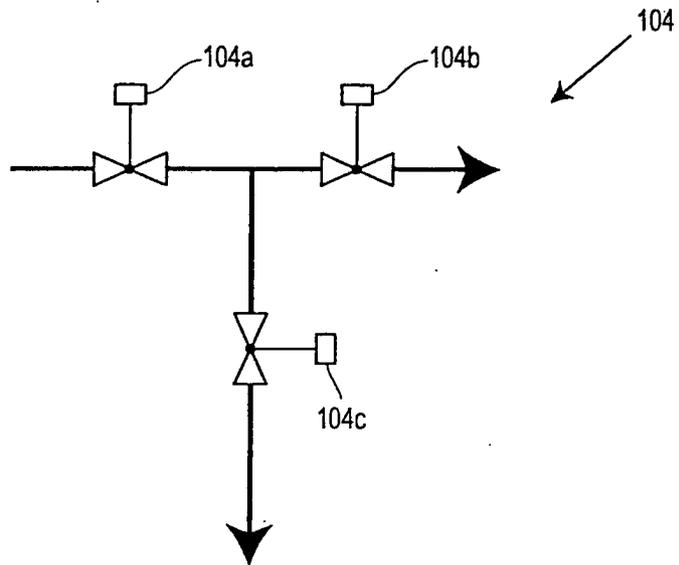


FIG. 4

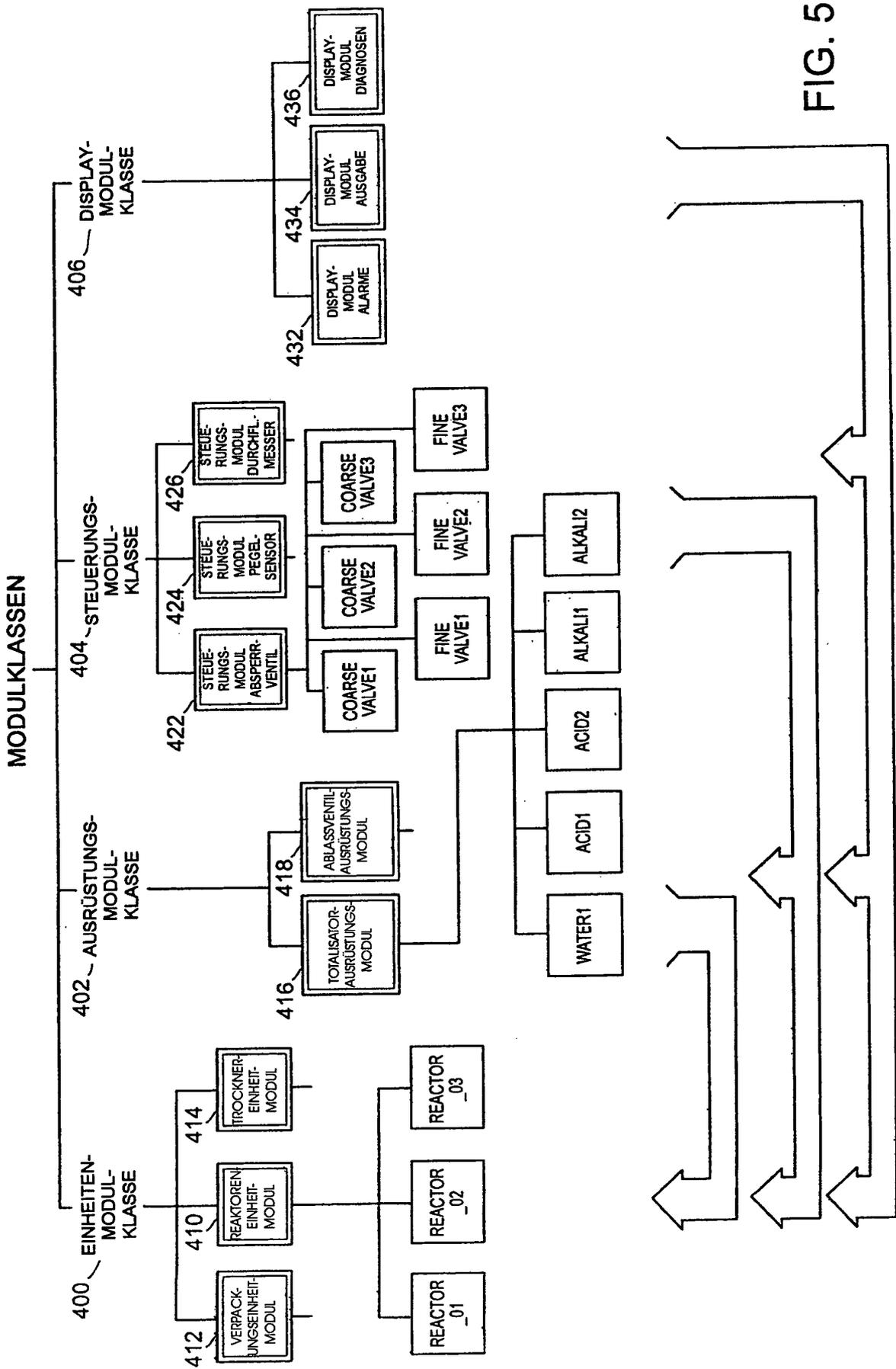


FIG. 5

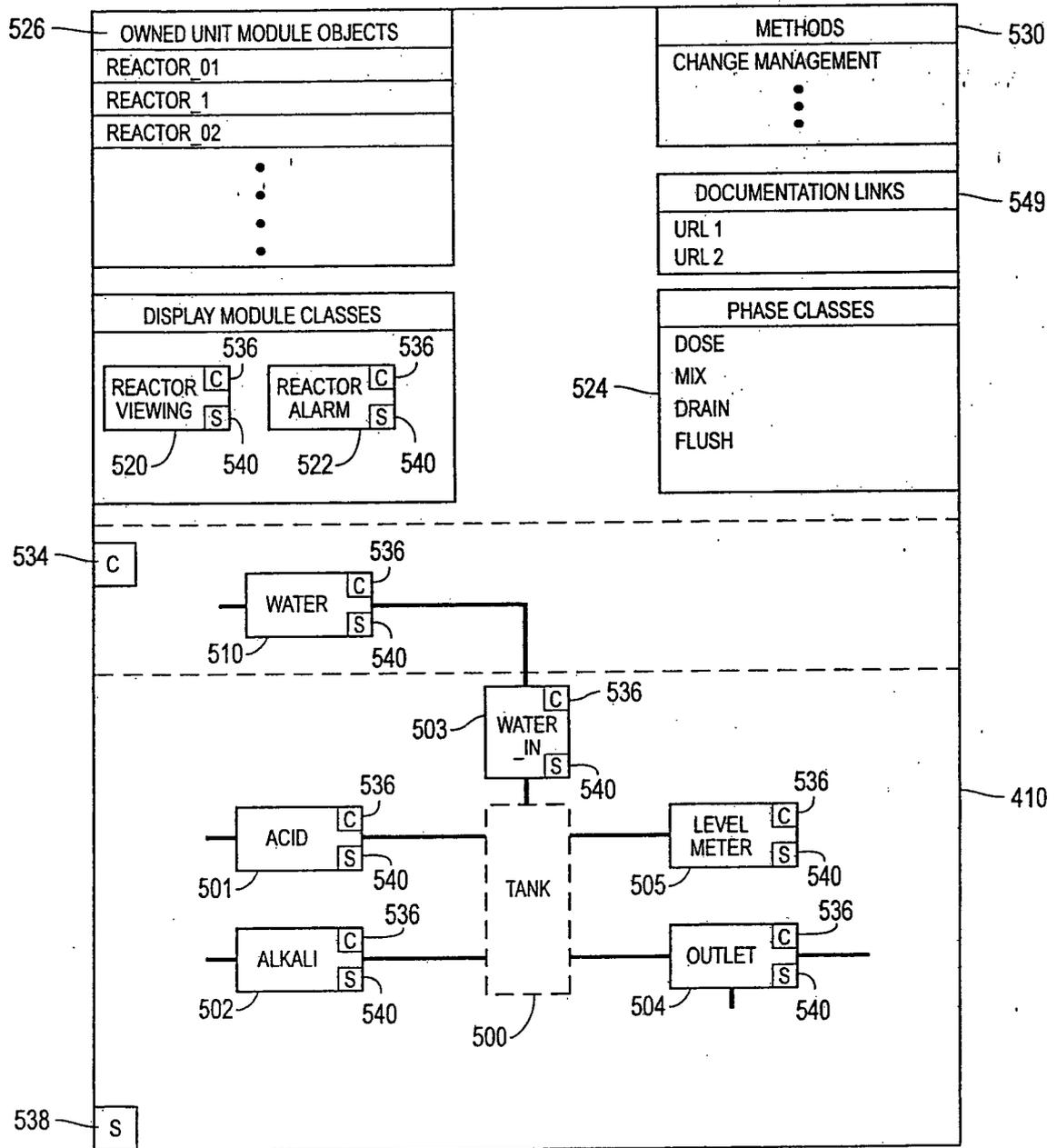


FIG. 6

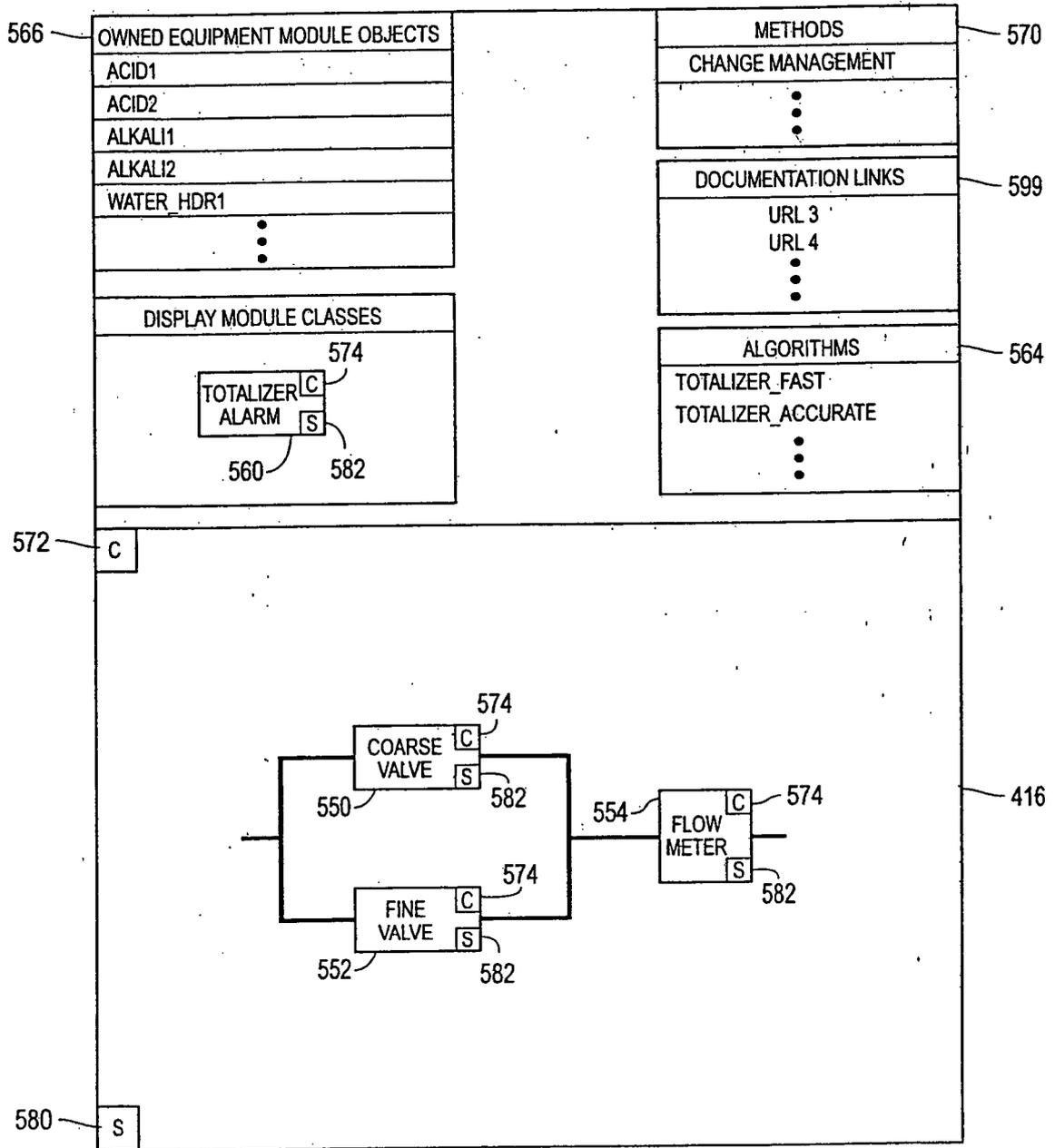


FIG. 7

The screenshot shows the 'Exploring Delta V' interface. On the left, a tree view displays the 'REACTOR' structure with components labeled 604, 610, 606, 614, 616, and 608. The main area shows a detailed diagram of these components. On the right, a table lists parameters for the reactor.

Name	Type	Description	Parameter Value	Filtering	Parameter Typ
FREMEM	Parameter		0	<On-I...	32 bit signed
DOSE	Phase Class	Phase Definition			
MIX	Phase Class	Phase Definition			
DRAIN	Phase Class	Phase Definition			
FLUSH	Phase Class	Phase Definition			
ACID	Module block				
ACID1	Alias				
ALKALI	Module block				
ALKALI	Alias				
WATER	Module block				
WATER	Alias				
WATER_IN	Module block				
WATER_IN	Alias				
LEVEL_METER	Module block				
LEVEL_METER	Alias				
OUTLET	Module block				
OUTLET	Alias				

600

602

612

FIG. 8

620

Control Studio - [Advanced Definitions/Equipment Module Classes/General/TOTALIZER]

File Edit View Tools Graphics Window Help

100%

Command Function Block

Command	State Name	Timeout	Enabled
COMMAND_00002	Totalize accurate	0	True
COMMAND_00003	Totalize fast	0	True
COMMAND_00099	Reset	0	True

MONITOR

614

622

Filtered by: Alphanumeric Categorized

Parameter	Default
AUTO_ADV..	True
CDA_ERROR	False
CDA_PV	Idle
CDA_SP	Idle
CDA_STATE	Idle
CDA_TARGET	Idle
CDA_TIMED...	False
COMMAND...	Start S
CONFIRM_FAIL	False
ERROR	Sequer
INITIAL_STATE	False
ERROR	False

Alarm	Word	State	Parameter	Limit Value	Enable	Inverted	Priority	%P1 parameter	%P2 parameter
ALARM1	ANY		MSTATUS		True	False	ADVI...		

Assigned to: <unassigned>

NUM 11:41 AM

FIG. 9

630

Control Studio - [Advanced Definitions/Equipment Module Classes/General/TOTALIZER]

File Edit View Tools Graphics Window Help

100%

614

TOTALIZE STEP1
 ABORT
 FAIL
 HOLD
 RESTA
 RUN J
 C
 C
 C
 M
 STOP

STEP1
 TRANSIT

634

Expression:

```

1 /FLOW_METER/RESET.CV := TRUE;
2
3 IF /SP.CV > 5.0 THEN
4   /COARSE_VALVE/TARGET.STATE.CV := "Valve:Open";
5 END_IF;
6
7 IF /SP.CV > 0.0 THEN
8   /FINE_VALVE/TARGET.STATE.CV := "Valve:Open";
9 END_IF;
    
```

632

Parser output

630

Alarm	Word	State	Parameter	Limit Value	Enable	Action	Text
ALARM1	ANY		MSTATUS		True	A1	/FLOW_METER/RESET.CV := TR...

Assigned to: <unassigned>

NUM 11:41 AM

Start

For Help, press F1

Parameter Default
 ACTIVE False
 CONFIRM_FAIL False
 DISABLED False
 ERROR False
 FAILED_CON... 0
 PENDING_CO... 0
 RERRR False
 TIME 0

FIG. 10

660

Exploiting Delta V
File Edit View Object Applications Tools Help

REACTOR_1

All Containers

Contents of REACTOR

Name	Type	Description	Parameter Value	Filtering	Work
<input checked="" type="checkbox"/> FLUSH	Parameter		False	<On-f...	
<input checked="" type="checkbox"/> ABNORM_ACTIVE	Parameter		True	<On-f...	
<input checked="" type="checkbox"/> ALARMENB	Parameter		False	<On-f...	
<input checked="" type="checkbox"/> AUTO_RESTART	Parameter		False	<Adv...	
<input checked="" type="checkbox"/> BAD_ACTIVE	Parameter		0	<On-f...	
<input checked="" type="checkbox"/> BATCH_ID	Parameter		0	<On-f...	
<input checked="" type="checkbox"/> BLOCK_ERR	Parameter		1	<Adv...	
<input checked="" type="checkbox"/> EXEC_TIME	Parameter		In Service	<On-f...	
<input checked="" type="checkbox"/> FREMEM	Parameter		In Service	<On-f...	
<input checked="" type="checkbox"/> MAX_RESERVED	Parameter		In Service	<Adv...	
<input checked="" type="checkbox"/> MCOMMAND	Parameter		In Service	<On-f...	
<input checked="" type="checkbox"/> MERROR	Parameter		In Service	<Adv...	
<input checked="" type="checkbox"/> MERROR_MASK	Parameter		In Service	<On-f...	
<input checked="" type="checkbox"/> MSTATE	Parameter		In Service	<Adv...	
<input checked="" type="checkbox"/> MSTATUS	Parameter		In Service	<On-f...	
<input checked="" type="checkbox"/> MSTATUS_MASK	Parameter		In Service	<Adv...	
<input checked="" type="checkbox"/> NUMPHASES	Parameter		0	<On-f...	
<input checked="" type="checkbox"/> OPERATION	Parameter		0	<On-f...	
<input checked="" type="checkbox"/> PROCEDURE	Parameter		1	<On-f...	
<input checked="" type="checkbox"/> UNITPROCEDURE	Parameter		0	<Adv...	
<input checked="" type="checkbox"/> VERSION	Parameter		1	<On-f...	
<input checked="" type="checkbox"/> WATCHDOG	Parameter		0	<On-f...	
<input type="checkbox"/> DOSE_ALM	Any Alarm				
<input type="checkbox"/> DRAIN_ALM	Any Alarm				
<input type="checkbox"/> FLUSH_ALM	Any Alarm				
<input type="checkbox"/> MIX_ALM	Any Alarm				
<input type="checkbox"/> ACID1	Equipment Module				
<input type="checkbox"/> ALKALI1	Equipment Module				
<input type="checkbox"/> LEVEL_METER1	Equipment Module				
<input type="checkbox"/> OUTLET2	Equipment Module				
<input type="checkbox"/> DOSE	Phase				No
<input type="checkbox"/> DRAIN	Phase				No
<input type="checkbox"/> MIX	Phase				No

Exploiting De ...
F:\SimulatePro
XML Spy\EX...
Registry Editor
CAN-CONFIGURE
CAN-DOWNLOAD
NUM
11:41 AM

661

FIG. 13

680

Exploring Delta V
File Edit View Object Applications Tools Help

REACTOR_1

All Containers

Contents of REACTOR

Name	Type	Description	Parameter Value	Filtering	Work In
COARSE_VALVE1	Module	Control Module Class			No
FINE_VALVE1	Module	Control Module Class			No
FLOW_METER1	Module	Control Module Class			No
RUN_LOGIC3CDA_ER...	Exposed Parameter		False	<Com...	No
RUN_LOGIC3CDA_FV	Exposed Parameter		Idle	<Com...	No
RUN_LOGIC3CDA_SP	Exposed Parameter		Idle	<Com...	No
RUN_LOGIC3CDA_ST	Exposed Parameter		Idle	<Com...	No

684

ACID1 Configuration

Name	Ownership...	Module Class	Path	Ready
ACID1	Not Shared	TOTALIZER		No
COARSE_VALVE1	Not Shared	ON_OFF_V...	COARSE_VALVE1	Yes
FINE_VALVE1	Not Shared	ON_OFF_V...	FINE_VALVE1	Yes
FLOW_METER1	Not Shared	FLOW_MET...	FLOW_METER1	Yes

686

661

662

664

665

666

668

682

DOSE
MIX
DRAIN
FLUSH
System Configuration
Recipes
Setup
Alarm Preferences
Security
Named Sets
Licenses
Control Strategies
Unassigned I/O References
External Phases
AREA_A
LIC-549
LOOP
TIC-205
SALTS
REACTOR 1
ACID1 (ACID)
ALKALI (ALKALI)
LEVEL_METER1 (LEVEL_METER)
OUTLET2 (OUTLET)
DOSE
MAIN
FLUSH
MIX
Aliases
MIX
MIX
WATER_HDR1 (WATER)
COARSE_VALVE4 (COARSE_VALV)
FINE_VALVE4 (FINE_VALVE)
COARSE_VALVE4 (COARSE_VALVE)
Physical Network
Decommissioned Nodes
Control Network

Start | Help, press F1

60DvDbServer... | Exploring De... | F:SimulatePro | XML Spy|EX... | Replary Editor | CAN-CONFIGURE | CAN-DOWNLOAD | NUM | 11:41 AM

FIG. 14

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					

IO Parameters Fieldbus Placeholders

FIG. 15

690

The screenshot shows a software interface with a menu bar at the top (File, Edit, View, Object, Applications, Tools, Help) and a toolbar. The main area is divided into two panes. The left pane, labeled 'All Containers', shows a tree structure of components under 'REACTOR_1'. The right pane, labeled 'Contents of REACTOR', shows a table of these components with columns for Name, Type, Description, MMI Picture, Work in Progress, and Node Assignment.

Tree Structure (Left Pane):

- REACTOR_1
 - SALTS
 - REACTOR_1
 - ACID1 (ACID) 666
 - ALKALY (ALKALI) 668
 - LEVEL_METER1 (LEVEL_METER)
 - OUTLET2 (OUTLET)
 - DRAIN1 (DRAIN)
 - OUTLET1 (OUTLET)
 - PRODUCT1 (PRODUCT)
 - DOSE
 - MAIN
 - FLUSH
 - MIX
 - Aliases
 - WATER_HDR1 (WATER)
 - WATER_IN1 (WATER_IN) 670
 - WATER_HDR1
 - COARSE_VALVE4 (COARSE_VALVE)
 - FINE_VALVE4 (FINE_VALVE)
 - FLOW_METER4 (FLOW_METER)
 - REACTOR_1_1 692
 - ACIDS2 (ACID)
 - ALKALIZ (ALKALI)
 - LEVEL_METER2 (LEVEL_METER)
 - OUTLET4 (OUTLET)
 - WATER_IN2 (WATER_IN)
 - WATER_HDR1 (WATER)
 - DOSE
 - DRAIN
 - FLUSH
 - MIX
 - Aliases
 - SALTS_1

Table (Right Pane):

Name	Type	Description	MMI Picture	Work in Progress	Node Assignment
LIC-549	Module	Control Module		No	←unassigned
LOOP	Module	Control Module		No	←unassigned
TIC-205	Module	Control Module		No	←unassigned
SALTS	Process Cell				←unassigned
SALTS_1	Process Cell				←unassigned

661

FIG. 16

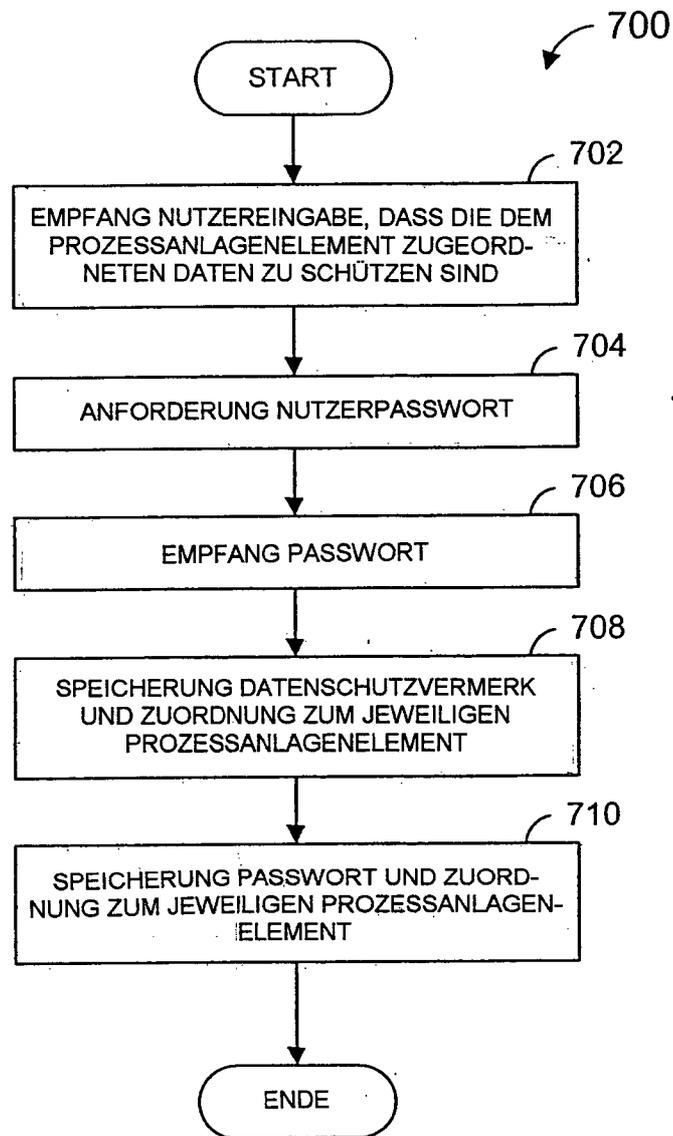


FIG. 17

FIG. 18A

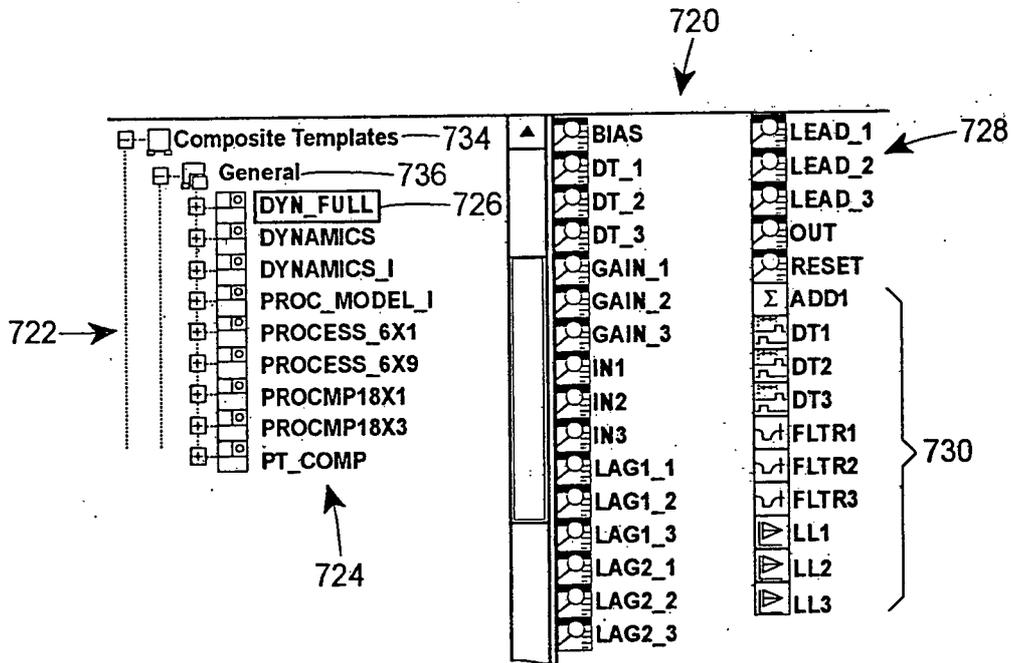
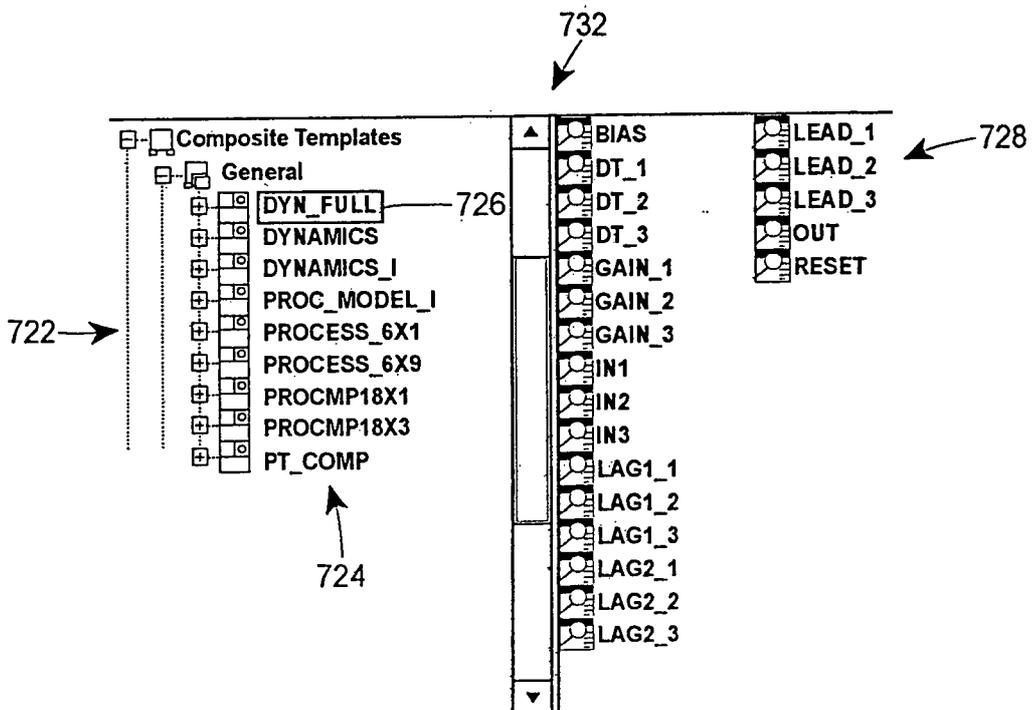


FIG. 18B



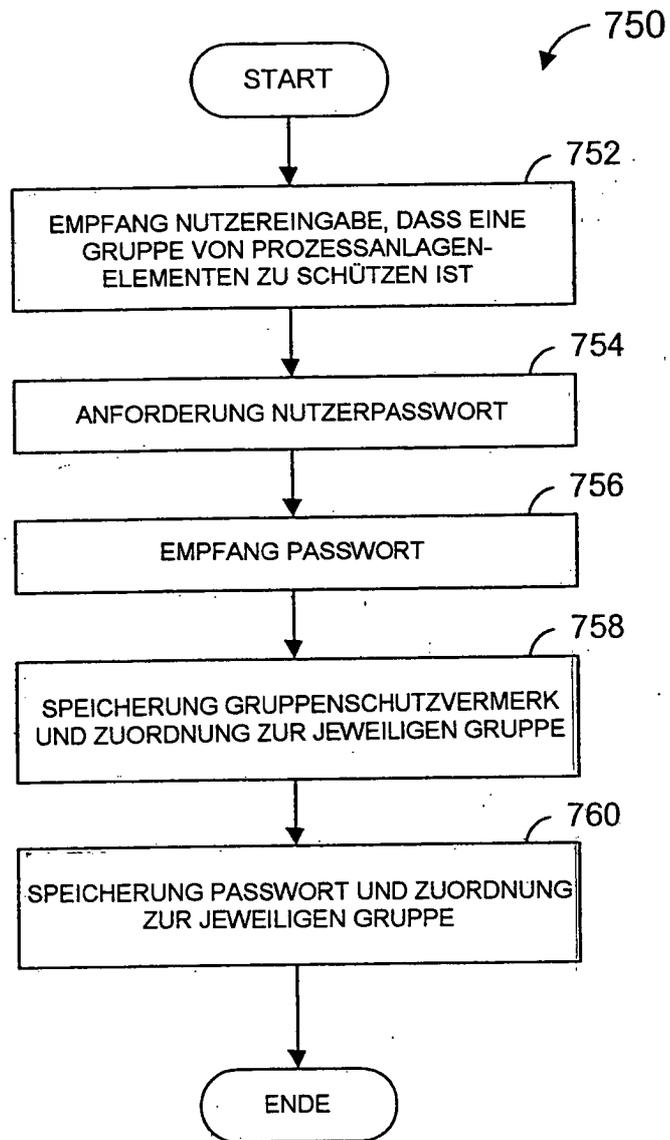


FIG. 19

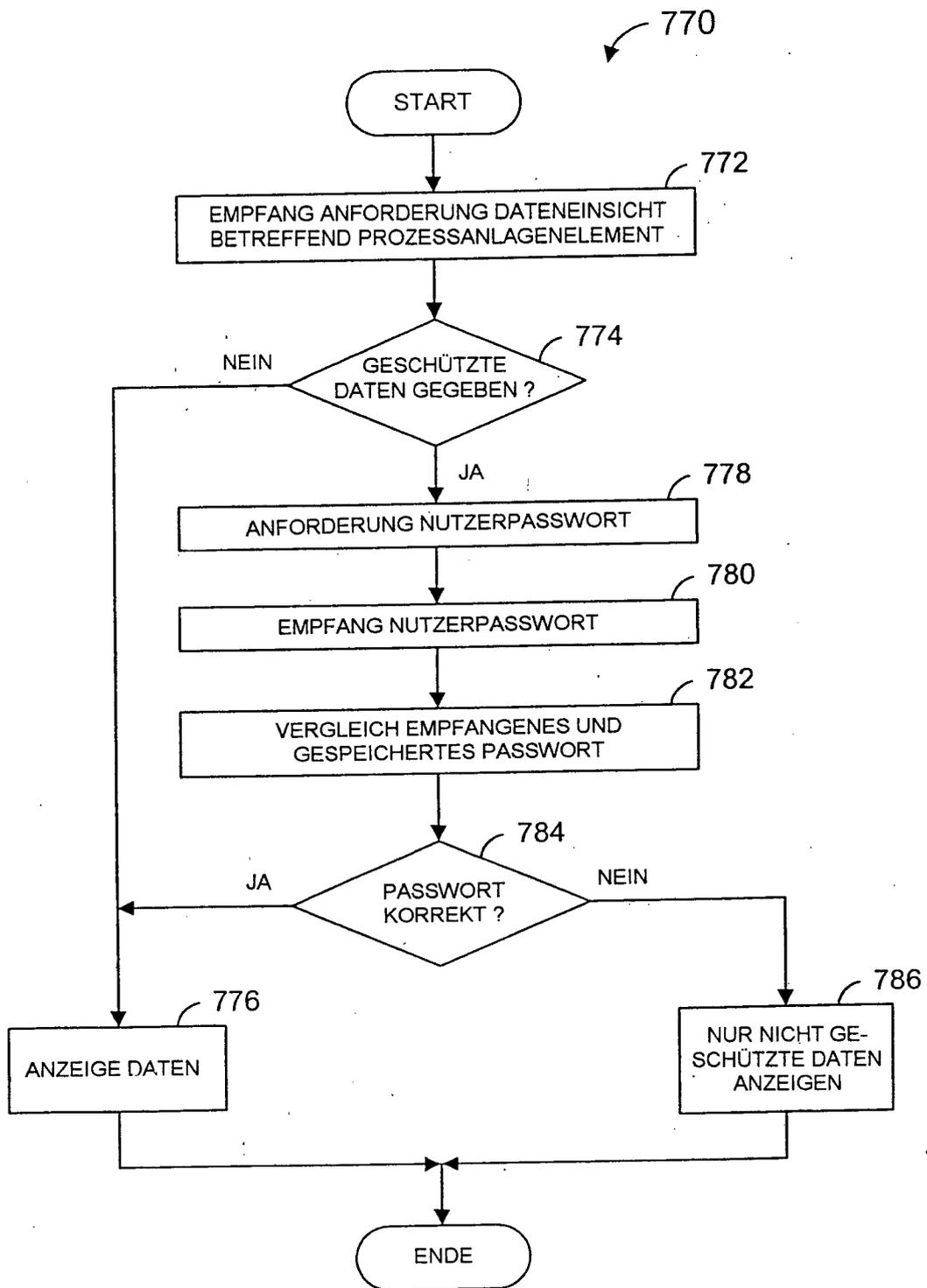


FIG. 20

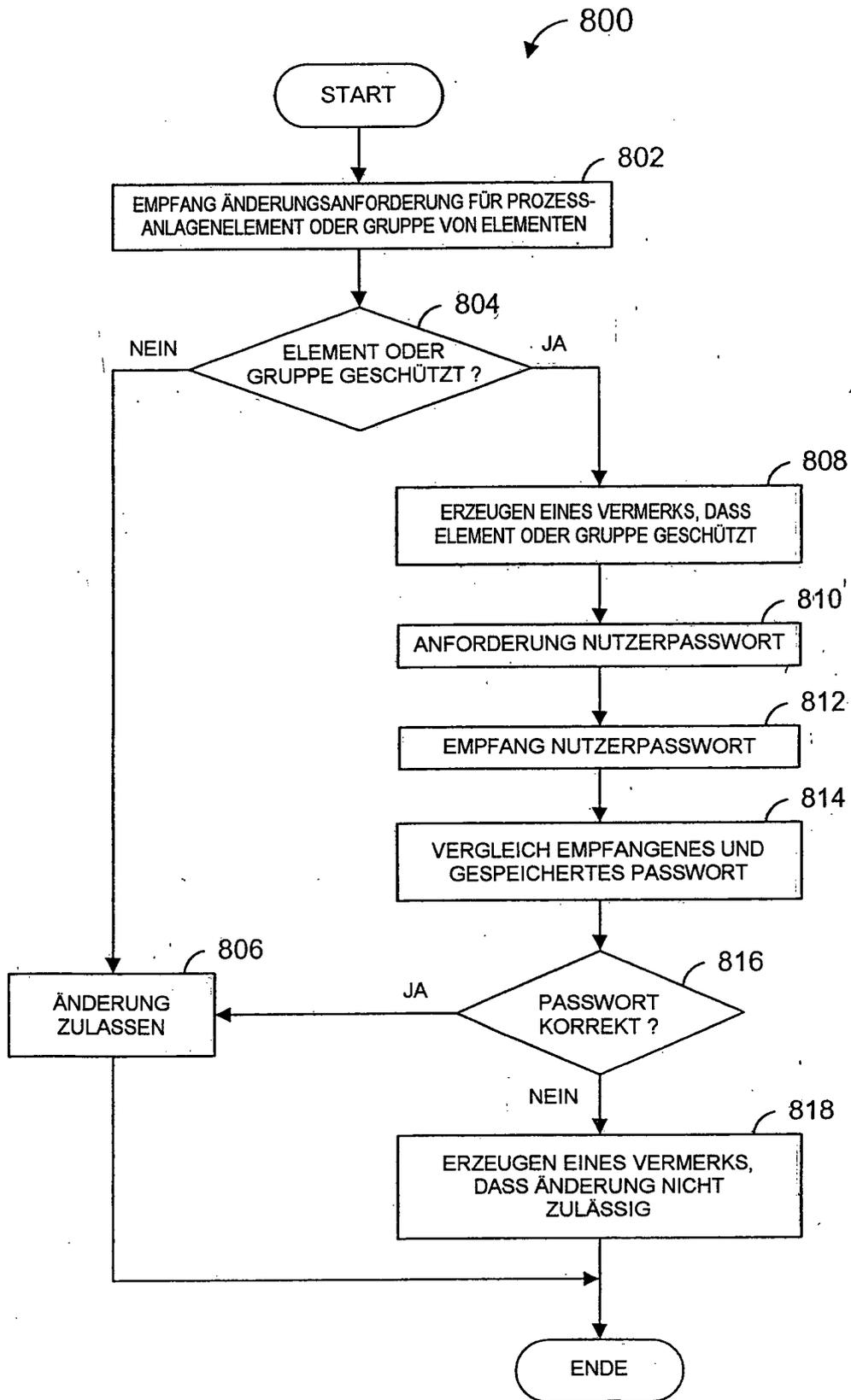


FIG. 21

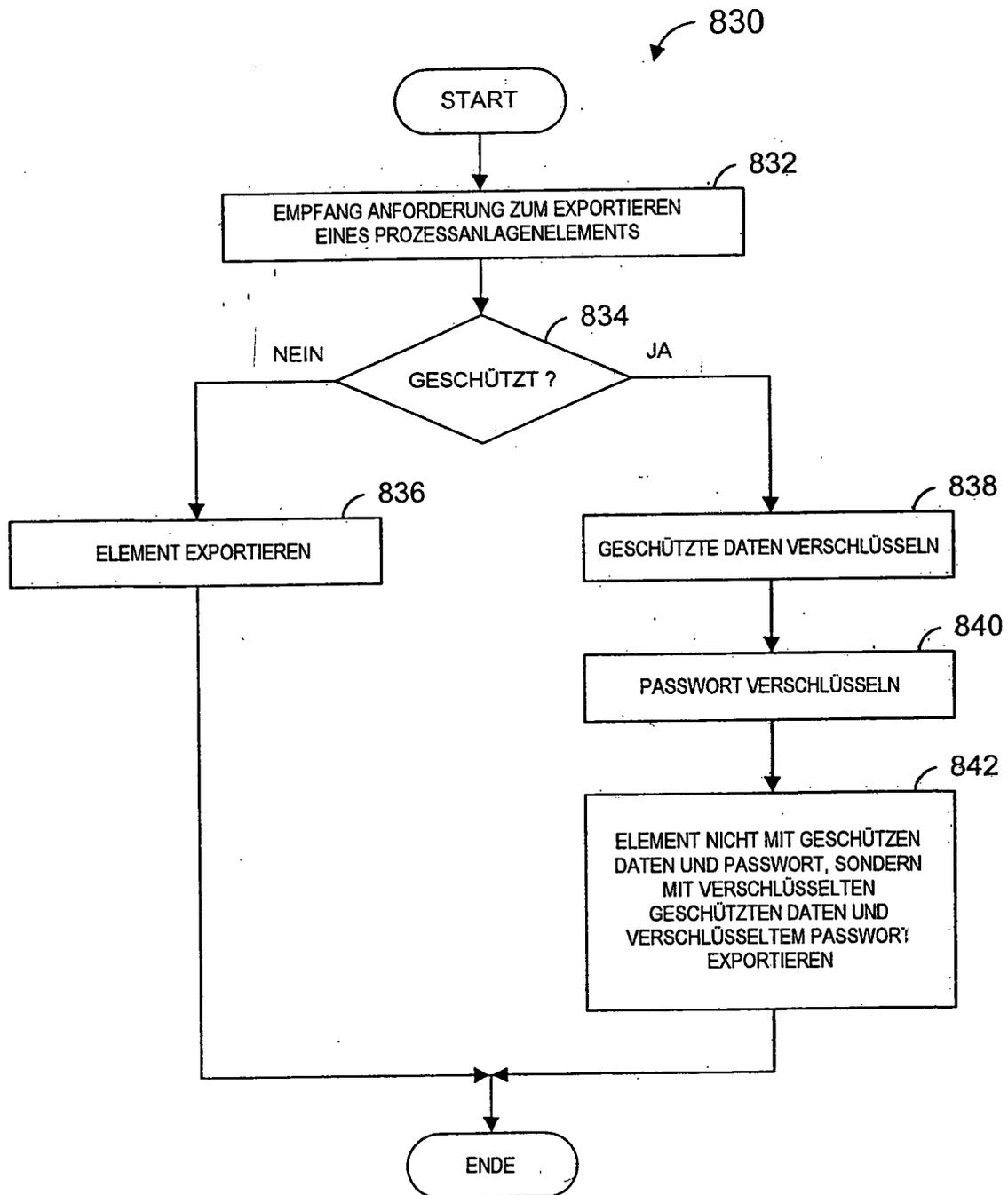
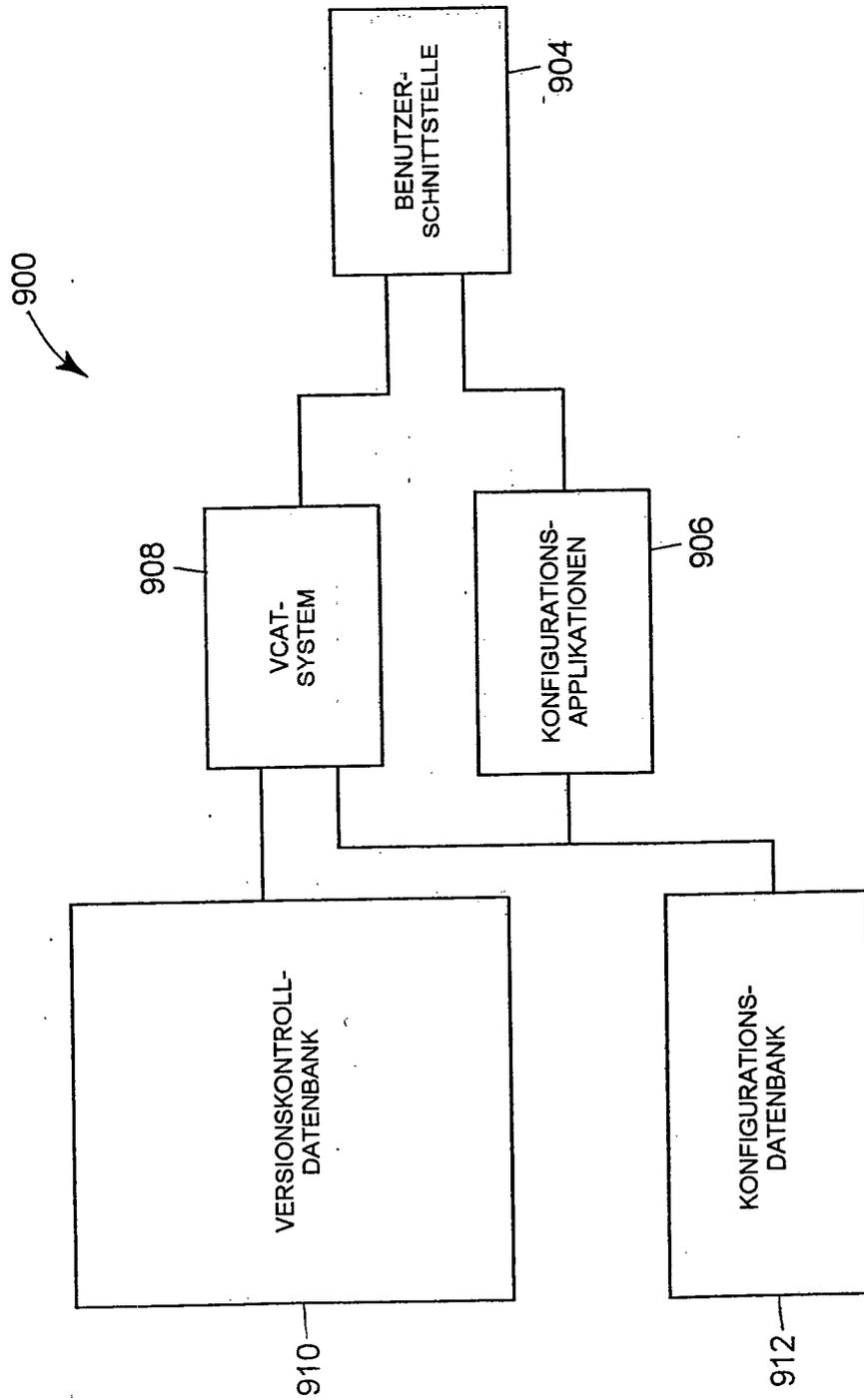


FIG. 22

FIG. 23



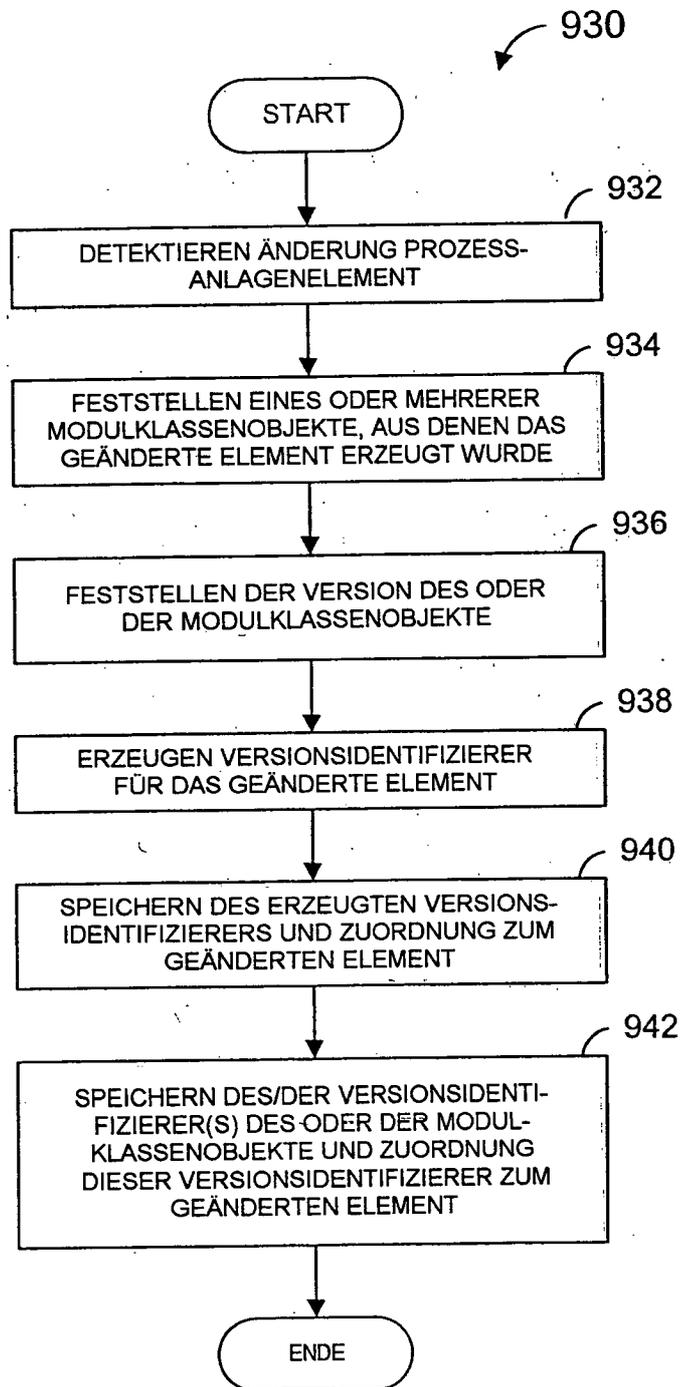


FIG. 24

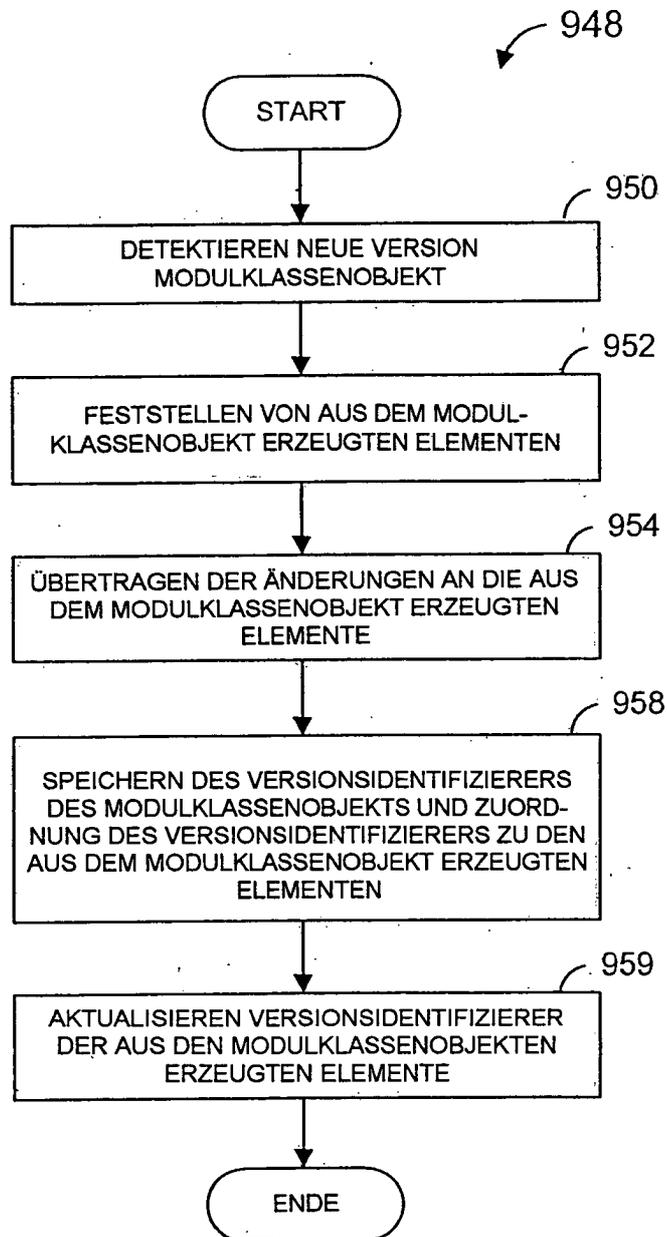


FIG. 25

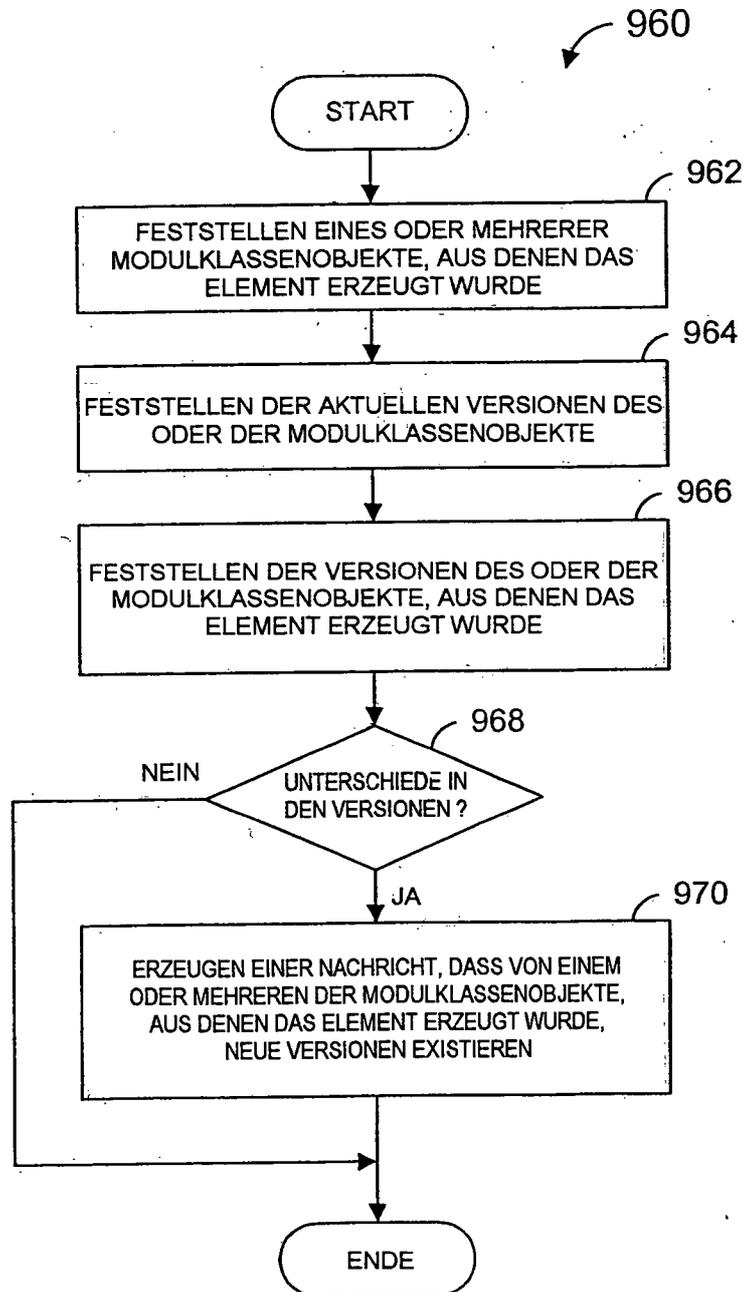


FIG. 26

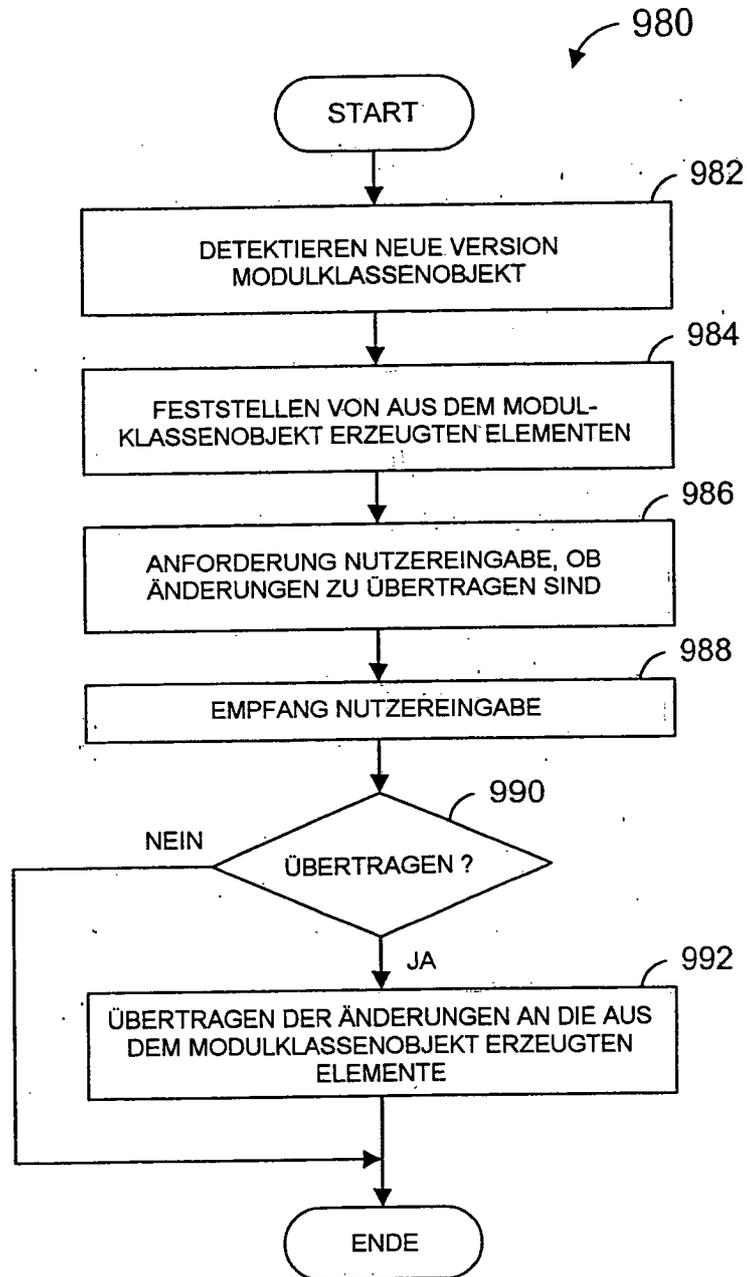


FIG. 27

FIG. 28

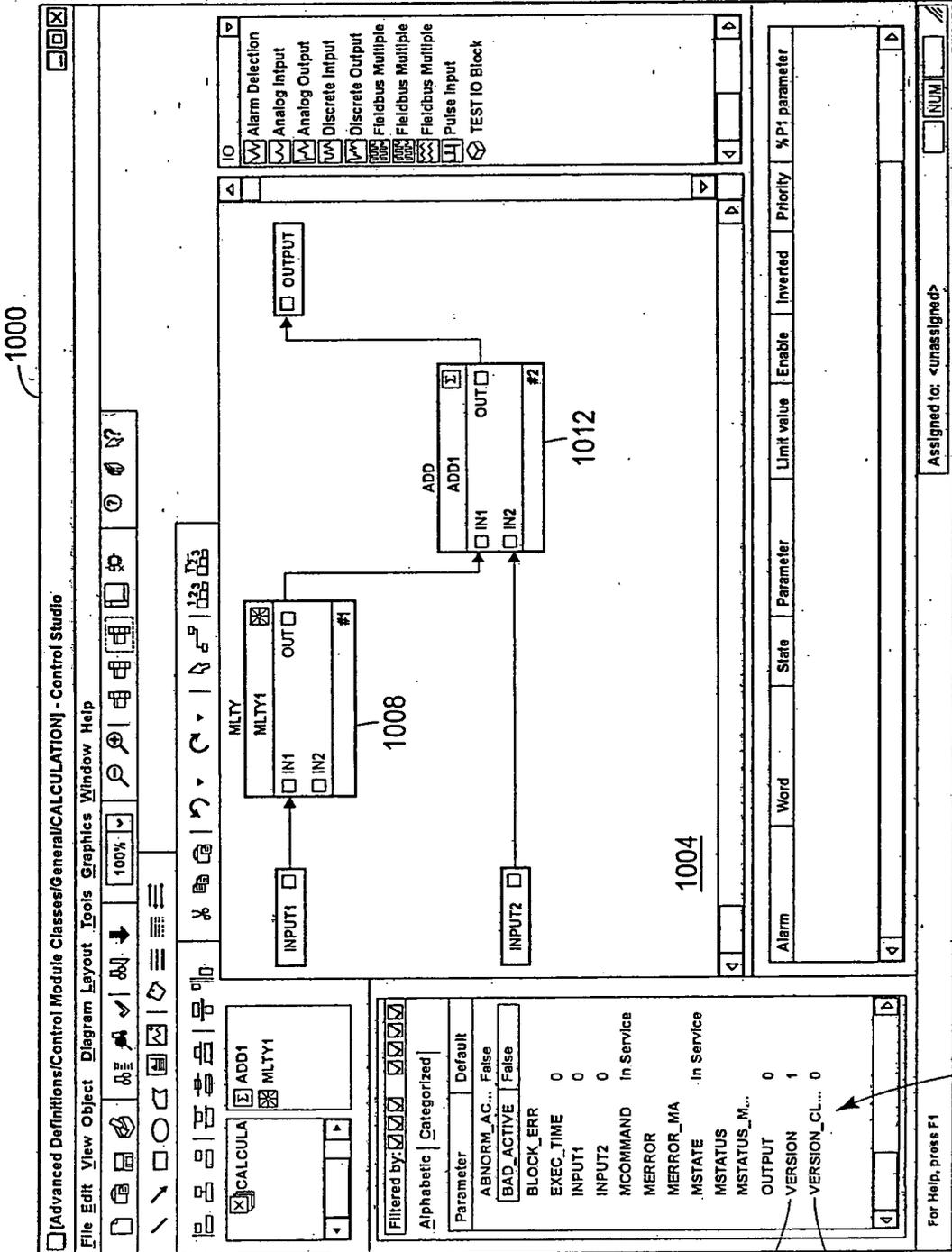
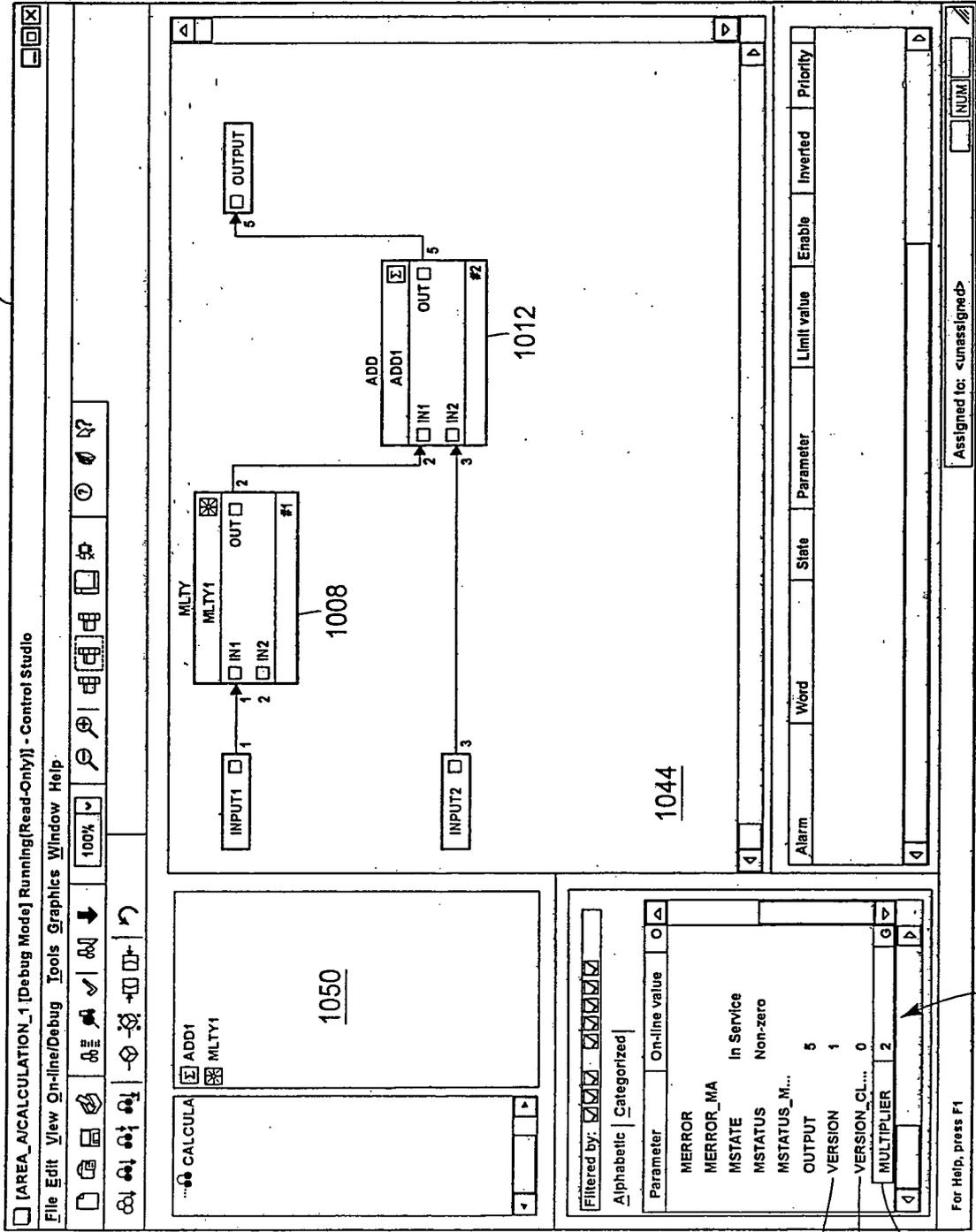


FIG. 30

1040



1026
1028
1038

FIG. 31

