(19) Deutsches Patent- und Markenamt

(10) **DE 10 2010 036 914 A1** 2011.02.24

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2010 036 914.4**
(22) Anmeldetag: **09.08.2010**
(43) Offenlegungstag: **24.02.2011**

(51) Int Cl.[8]: ***G05B 19/04*** (2006.01)

(30) Unionspriorität:
12/539,399 11.08.2009 US

(71) Anmelder:
Fisher-Rosemount Systems, Inc., Austin, Tex., US

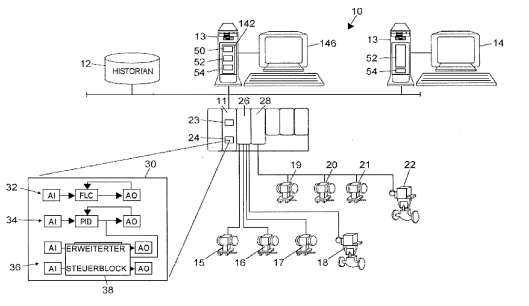
(74) Vertreter:
Meissner, Bolte & Partner GbR, 80538 München

(72) Erfinder:
Blevins, Terrence Lynn, Round Rock, Tex., US; Lucas, John Michael, Leire Lutterworth, Leicestershire, GB; Nixon, Mark J., Round Rock, Tex., US; Gilbert, Stephen Copps, Austin, Tex., US; Enver, Alper T., Austin, Tex., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Systemkonfiguration unter Verwendung von Vorlagen**

(57) Zusammenfassung: Ein Verfahren in einem Computersystem zur Entwicklung einer Prozesssteuerungsstrategie umfasst ein Bereitstellen einer Modulvorlage, die eine ersten Mehrzahl von Komponenten aufweist und einem Steuerungsvorgang zugeordnet wird, Empfangen einer Auswahl einer oder mehrerer der ersten Mehrzahl von Komponenten der Modulvorlage, Erzeugen einer Instanz eines Moduls auf der Basis der Modulvorlage, umfassend ein Instanziieren nur der einen oder mehreren ausgewählten der Mehrzahl von Komponenten und Zuordnen der erzeugten Instanz des Moduls zur Prozesssteuerungsstrategie.



Beschreibung

GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft im Allgemeinen Prozesssteuersysteme und insbesondere die Bereitstellung einer effizienten Konfiguration solcher Systeme.

STAND DER TECHNIK

[0002] Prozesssteuersysteme wie beispielsweise verteilte oder skalierbare Prozesssteuersysteme, wie sie in Chemie-, Erdöl- und anderen Prozessen verwendet werden, umfassen normalerweise eine oder mehrere Prozesssteuerungen, die über analoge, digitale oder kombinierte Analog-Digital-Busse untereinander sowie mit wenigstens einer Host- oder Bediener-Workstation und mit einem oder mehreren Feldgeräten kommunikativ verbunden sind. Die Feldgeräte, bei denen es sich zum Beispiel um Ventile, Ventilpositionierer Schalter und Messwertgeber (z. B. Temperatur-, Druck- und Durchflusssensoren) handeln kann, führen innerhalb des Verfahrens Funktionen wie beispielsweise das Öffnen oder Schließen von Ventilen und Messen von Prozessparametern aus. Die Prozesssteuerung empfängt Signale, die von den Feldgeräten vorgenommene Prozessmessungen und/oder sonstige Informationen in Bezug auf die Feldgeräte anzeigen, verwendet diese Informationen zur Implementierung einer Steuerroutine und erzeugt anschließend Steuersignale, die über die Busse an die Feldgeräte gesendet werden, um den Prozessbetrieb zu steuern. Die Informationen von den Feldgeräten und der Steuerung werden normalerweise für eine oder mehrere Anwendungen verfügbar gemacht, die von der Bediener-Workstation ausgeführt werden, um es einem Bediener zu ermöglichen, beliebige gewünschte Funktionen in Bezug auf den Prozess auszuführen, wie beispielsweise Anzeigen des aktuellen Prozessstatus, Modifizieren des Prozessbetriebs usw.

[0003] Einige Prozesssteuersysteme wie beispielsweise das DeltaV®-System, das von der Fisher Rosemount Systems, Inc., mit Sitz in Austin, Texas, vertrieben wird, verwenden Funktionsblöcke oder als Module bezeichnete Gruppen von Funktionsblöcken, die in der Steuerung und/oder in Feldgeräten angeordnet sind, zur Durchführung von Steuerungsvorgängen. In diesen Fällen kann die Steuerung oder eine andere Vorrichtung ein(en) oder mehrere Funktionsblöcke oder Module umfassen und ausführen, die jeweils Eingaben von anderen Funktionsblöcken (innerhalb derselben Vorrichtung oder innerhalb verschiedener Vorrichtungen) empfangen und/oder Ausgaben dafür bereitstellen und bestimmte Prozessvorgänge durchführen, wie beispielsweise das Messen oder Erkennen eines Prozessparameters, Steuern einer Vorrichtung oder Durchführen eines Steuerungsvorgangs, wie beispielsweise die Implementierung einer Proportional-Integral-Derivative(PID)-Steuerroutine. Die verschiedenen Funktionsblöcke und Module innerhalb eines Prozesssteuersystems sind im Allgemeinen so konfiguriert, dass sie (z. B. über einen Bus) miteinander kommunizieren, um einen oder mehrere Prozessregelkreise zu bilden.

[0004] In einigen Fällen können Funktionsblöcke dem von Foundation™ Fieldbus verbreiteten Standard entsprechen oder diesem ähnlich sein. Der Begriff „Funktionsblock", wie hierin verwendet, ist jedoch nicht darauf beschränkt, was die DeltaV- oder die Fieldbus-Protokolle als Funktionsblock festlegen, sondern umfasst stattdessen jeglichen anderen, beliebigen Steuersystem- und/oder Kommunikationsprotokolltypen zugeordneten Block-, Programm-, Hardware-, Firmwaretyp usw., der zur Implementierung von Steuerfunktionen verwendet werden kann. Außerdem kann sich der Begriff „Funktionsblock, wie hierin verwendet, im Allgemeinen auf einen Funktionsblock, der eine oder mehrere Steuerfunktionen enthält, einen Betriebsmittelblock, der einen oder mehrere Prozessparameter enthält, einen Transducer-Block, der einer Schnittstelle zu einem Sensor (z. B. Temperatursensor, Drucksensor usw.), einem Durchflussmesser, einem Ventil-Stellantrieb usw. entspricht, oder jeden anderen Blocktyp beziehen. Ferner kann sich Funktionsblock auf Basisfunktionsblöcke, wie beispielsweise diskrete Eingänge (DI), diskrete Ausgänge (DO), Analogeingänge (AO), Analogausgänge (AO), PID-Steuerungen, PD-Steuerungen, PI-Steuerungen, P-Steuerungen, Steuerungswähler, Vorspannungs-/Verstärkungsstationen usw., sowie auf erweiterte Funktionsblöcke wie beispielsweise Sollwert-Rampengeneratoren, Timer, analoge Alarme, diskrete Alarme, Totzeit usw. beziehen. Darüber hinaus kann ein Funktionsblock, wie hierin verwendet, ein geschachtelter Block sein, der zum Beispiel mehrere Fieldbus-Funktionsblöcke oder sogar einen oder mehrere ineinander geschachtelte Blöcke enthält. Es ist außerdem zu erwähnen, dass Funktionsblöcke, obwohl sie normalerweise die Form von Objekten innerhalb einer objektorientierten Programmierumgebung annehmen, im Allgemeinen unter Verwendung einer beliebigen gewünschten Datenstruktur in jeder geeigneten Softwareumgebung definiert werden können.

[0005] Demnach werden Prozesssteuerungen normalerweise so programmiert, dass sie verschiedene Algorithmen, Subroutinen oder Regelkreise (die alle Steuerroutinen sind) für jeden einer Anzahl von verschiedenen Regelkreisen ausführen, die für einen Prozess definiert oder darin enthalten sind, wie beispielsweise Durchflussregelkreise, Temperaturregelkreise, Druckregelkreise usw. Wie bereits erwähnt, umfasst jeder derartige Regelkreis einen oder mehrere Eingangsblöcke, wie beispielsweise einen Ana-

logeingangs(AI)-Funktionsblock, einen Einzel-Ausgangssteuerblock, wie beispielsweise einen Proportional-Integral-Derivative-(PID)- oder einen Fuzzy-Logik-Steuerfunktionsblock, und einen Ausgangsblock, wie beispielsweise einen Analogausgangs(AO)-Funktionsblock.

[0006] Steuerroutinen und die Funktionsblöcke, die diese Routinen implementieren, werden gemäß einer Anzahl von Steuerungsverfahren konfiguriert, die PID-Steuerungs-, Fuzzy-Logik-Steuerungs- und modellbasierte Verfahren wie beispielsweise eine Smith-Predictor- oder eine modellprädiktive Steuerung (MPC) umfassen. Bei modellorientierten Steuerungsverfahren basieren die in den Routinen zur Bestimmung der Reaktion der Regelung mit Rückführung verwendeten Parameter auf Änderungen der beeinflussten oder gemessenen Störungen, die als Prozesseingaben dienen. Eine Darstellung dieser Prozessreaktion auf Änderungen von Prozesseingaben kann als Prozessmodell beschrieben werden. Zum Beispiel kann ein parametriertes Prozessmodell erster Ordnung Werte für die Verstärkung, Totzeit und Zeitkonstante des Prozesses spezifizieren.

[0007] Bei einer typischen Anlage kann ein Ingenieur die Prozesssteuerungsstrategie unter Verwendung eines Konfigurationssystems definieren und konfigurieren, das auf einer Bediener-Workstation läuft. Einige Konfigurationssysteme können eine Bibliothek zum Speichern von Steuerelementen, wie beispielsweise Funktionsblöcken oder Modulen (die normalerweise aus einer Anzahl von Funktionsblöcken bestehen) umfassen, derart dass der Ingenieur eine Instanz eines ausgewählten Steuerelements gemäß einer bestimmten Anwendung auswählen und erzeugen kann. Das Konfigurationssystem kann dem Ingenieur zum Beispiel durch Herunterladen des Steuerelements auf eine Steuerung oder ein programmierbares Feldgerät auch die Vornahme von Modifikationen erlauben, um die erzeugte Instanz des ausgewählten Steuerelements vor dem Anwenden der Instanz auf die Prozesssteuerungsumgebung zu ändern.

[0008] Zum Beispiel speichert eine Vorlagenbibliothek in einem DeltaV-System verschiedene Modulvorlagen, welche die Basismess- und -steuerfunktionalität betreffen. Vorlagen in DeltaV können autonom oder klassenbasiert (d. h. mit aus der Klassenvorlage instanziierten Instanzen verknüpft und zum Übertragen von Änderungen der Klassenvorlage auf die Instanzen fähig) sein. Ein Ingenieur verwendet häufig eine oder mehrere Modulvorlagen als Ausgangspunkt beim Definieren und Konfigurieren des entsprechenden Prozesssteuerungsschemas. Da typische Modifikationen an den Modulvorlagen jedoch einen erheblichen technischen Aufwand mit sich bringen und bestimmte Check-in- und Check-out- sowie Dokumentationsvorgänge erfordern, kann das Arbeiten mit der Vorlagenbibliothek zeitraubend sein.

[0009] Um die Aufgabe des Konfigurierens eines Prozesssteuersystems zu vereinfachen, entwickelte Emerson™ Process Management eine Sammlung umfassender wiederverwendbarer Modulvorlagen und Modulklassen, die als Projekt Builder Library (PBL) bekannt ist. Im Allgemeinen behandeln die Modulvorlagen in der PBL-Bibliothek die größte in Erwägung gezogene Auswahl an Konfigurationsoptionen und -szenarien, die auf ein bestimmtes Modul anwendbar sind. Die zur PBL-Bibliothek beitragenden Ingenieure bauen auf internationalen Standards wie beispielsweise ISA 588.0, IEC 61508, IEC 61131-3 usw. auf und lassen Erfahrung und bewährte Methoden aus vielen Stunden der Anwendungs- und Projektentwicklung einfließen. Mithilfe der PBL-Bibliothek kann ein Ingenieur eine Modulvorlage auswählen, Werte von Modulparametern modifizieren, um die gewünschten Merkmale zu aktivieren und zu konfigurieren, und die für die jeweilige Anwendung unnötigen Merkmale zu deaktivieren. Zum Beispiel kann eine bestimmte Vorlage acht mögliche Eingänge in einen bestimmten Funktionsblock ermöglichen und demnach acht Eingangsblöcke umfassen, die diesen acht Eingängen entsprechen. Ein Benutzer, der nur einen dieser Eingänge benötigt, könnte sieben der acht Eingänge deaktivieren, indem er den entsprechenden Parameter den Wert FALSE (Falsch) zuweist. Eine typische PBL-Vorlage enthält demnach mehr Merkmale als ein für einen ähnlichen Zweck definiertes Modul der DeltaV-Bibliothek. Zum Beispiel kann eine PBL-Vorlage für kontinuierliche Steuerung alle Merkmale der entsprechenden DeltaV-Vorlage sowie zusätzliche Merkmale in Bezug auf Gerätezuteilung, Unterstützung für Vier-Spur-Eingänge mit Aktivierungs-/Deaktivierungsfähigkeit und First-out-Erkennung, bedingte Alarmierung mit Aktivierungs-/Deaktivierungsfähigkeit und Bedienerzugriff, Steuerungen zum Einstellen des Status von RCAS_IN- und ROUT_IN-Kanälen, Betriebssperre, um auf Wunsch zu verhindern, dass Bediener auf das Modul zugreifen, einen Fehlerparameter usw. enthalten. Kurz gesagt, enthält eine PBL-Modulvorlage wahrscheinlich die gesamte Funktionalität eines Moduls, die ein Ingenieur möglicherweise für ein bestimmtes Projekt benötigt, und, um das Modul zu verwenden, muss der Ingenieur normalerweise nur einige oder alle Werte von Modulparametern ändern.

[0010] Obwohl die PBL-Bibliothek den Vorgang des Konfigurierens der Prozesssteuerung erheblich vereinfachen kann, benötigen PBL-Modulvorlagen unglücklicherweise eine verhältnismäßig große Steuerungsspeicherkapazität. Da Ingenieure Modulvorlagen durch Modifizieren von Modulparametern individuell anpassen, erbt jede Instanz alle Funktionsblöcke von der Vatermodulvorlage zusammen mit den zugeordneten Parameter ungeachtet dessen, ob ein bestimmter Funktionsblock in der Instanz wirksam ist.

Außerdem bieten PBL-Vorlagen dem Benutzer nicht immer das Erlebnis des „What you see is what you can have" (Was du siehst, ist das, was du haben kannst), da jede Modulinstanz die gesamte Funktionalität der entsprechenden PBL-Modulvorlage beibehält und die Ingenieure viele Parameter überprüfen müssen, um zu bestimmen, welche Funktionsblöcke und Parameter tatsächlich in Verwendung sind.

KURZDARSTELLUNG

[0011] Eine Modulvorlage zur Verwendung in einer Prozesssteuerungskonfiguration enthält eine oder mehrere auswählbare Komponenten wie beispielsweise Logik- oder Funktionsblöcke und Prozessparameter. Beim Entwerfen eines Prozesssteuerungsschemas kann ein Bediener eine Modulvorlage für eine bestimmte Anwendung auswählen, die gewünschten Komponenten innerhalb der Modulvorlage auswählen und eine Modulinstanz erstellen, die nur die ausgewählten Komponenten enthält. Der Bediener kann auf diese Weise sowohl die Struktur der Modulinstanz als auch die Parameter entsprechend den Komponenten der Modulinstanz individuell anpassen. In einigen Ausführungsformen enthält die Modulvorlage zusätzlich eine oder mehrere nicht optionale Komponenten, und jede Modulinstanz, die unter Verwendung der Modulvorlage erstellt wurde, enthält demgemäß ungeachtet der optionalen Konfiguration der Modulinstanz und der vom Benutzer ausgewählten optionalen Komponenten jede der nicht optionalen Komponenten. Optional enthält eine Modulinstanz auch eine Vatervorlagenkennung, um eine Modulvorlage, aus der die Modulinstanz erstellt wurde, d. h. die „Vater"-Vorlage, zu identifizieren. Als eine andere Option kann die Modulinstanz ein Datenfeld enthalten, um zu spezifizieren, welche der in der Vatermodulvorlage verfügbaren Optionen bei der Instanziierung der Modulinstanz ausgewählt wurden. In einigen Ausführungsformen kann ein Bediener die Modulinstanz nach dem Instanziieren der Modulinstanz modifizieren. Wenigstens in einigen dieser Ausführungsformen kann der Bediener einige der vorher ausgewählten optionalen Komponenten deselektieren und/oder die vorher deselektierten optionalen Komponenten auswählen.

[0012] In einem weiteren Aspekt umfasst eine Softwareumgebung zum Konfigurieren eines Prozesssteuersystems eine Benutzeroberfläche zum Definieren einer Modulvorlage, Identifizieren einer oder mehrerer Komponenten der Modulvorlage und Zuweisen eines auswählbaren Attributs zu der identifizierten einen oder den identifizierten mehreren Komponenten, um anzuzeigen, dass die identifizierten Komponenten beim Erzeugen einer Modulinstanz auf der Basis der Modulvorlage ausgewählt oder weggelassen werden können. In einigen Ausführungsformen ermöglicht die Benutzeroberfläche es einem Benutzer, eine Gruppe von Komponenten auszuwählen und der ausgewählten Gruppe ein gemeinsames auswählbares Attribut zuzuweisen. Die Benutzeroberfläche kann auch ein Blocktypauswahlmerkmal unterstützen, um ein derartiges Zuordnen einer Komponente wie beispielsweise eines Funktionsblocks zu einer Liste mehrerer spezifizierter Typen zu ermöglichen, dass derselbe Benutzer oder ein anderer Bediener zu einem späteren Zeitpunkt den gewünschten Typ aus der Liste auswählen und die Komponente dem ausgewählten Typ in einer bestimmten Instanz zuordnen könnten. In einigen Ausführungsformen ermöglicht die Benutzeroberfläche es einem Benutzer, eine Komponente oder eine Gruppe von Komponenten in einer Modulvorlage auszuwählen und der ausgewählten Komponente oder Gruppe von Komponenten ein Namenskürzel zuzuweisen. Es ist vorgesehen, dass der Benutzer einen Namen wählt, der die Funktion beschreibt, welche die eine oder die mehreren Komponenten in einer auf der Modulvorlage basierten Instanz ausführen. Sobald ein Namenskürzel zugewiesen ist, zeigt die Benutzeroberfläche in einigen Ausführungsformen das Namenskürzel neben oder über der einen oder den mehreren Komponenten an, denen das Namenskürzel zugewiesen wurde. Wenn ein Benutzer einer Gruppe von Komponenten ein Namenskürzel zuweist, kann die Benutzeroberfläche außerdem einen die Gruppe einschließenden Rahmen, ein Lasso oder eine andere visuelle Anzeige darstellen, die darauf hinweist, dass das Namenskürzel einer Gruppe von Komponenten entspricht.

[0013] Die Softwareumgebung kann die Modulvorlage in einer Vorlagenbibliothek wie beispielsweise einer Datenbank speichern und die eine oder die mehreren gespeicherten Modulvorlagen einem Benutzer beim Entwurf oder der Konfiguration einer Prozesssteuerung über dieselbe oder eine andere Benutzeroberfläche zur Verfügung stellen. In einigen Ausführungsformen stellt die Benutzeroberfläche eine Kennung für jede optionale Komponente bereit, d. h. eine Komponente, die ausgewählt oder deselektiert werden kann.

[0014] Beim Wählen einer für eine bestimmte Anwendung geeigneten Modulvorlage kann der Benutzer die gewünschten Optionen auswählen und eine Steuerung auf der Benutzeroberfläche aktivieren, um eine Instanz eines Moduls zu erzeugen, das der ausgewählten Modulvorlage und den ausgewählten Optionen entspricht. Ein Benutzer kann mehrere Instanzen unter Verwendung derselben Modulvorlage erzeugen, wobei einige der Instanzen verschiedene ausgewählte Optionen aufweisen.

[0015] In einigen Ausführungsformen schützt die Softwareumgebung zum Konfigurieren eines Prozesssteuersystems eine mit einer Vatermodulvorlage verknüpfte Modulinstanz gegen fehlerhafte Bearbeitung, indem sie Benutzern nur die Modifikation jener

Komponenten in der Instanz gestattet, die in der Vatermodulvorlage bei der Erstellung der Instanz ausgewählt wurden. Falls gewünscht, kann die Benutzeroberfläche der Softwareumgebung es dem Benutzer ermöglichen, den Schutz aufheben, die Verknüpfung zur Vatermodulvorlage zu trennen und die Instanz wie gewünscht zu modifizieren. Zusätzlich oder alternativ kann die Benutzeroberfläche Instanzen, die mit Modulvorlagen mit auswählbaren Optionen verknüpft sind, schützen, indem sie dem Benutzer wenigstens bis zur Aufhebung durch einen ausdrücklichen Benutzerbefehl nur die Modifikation von Parameterwerten, aber nicht der Struktur der Instanz gestattet.

[0016] In einigen Ausführungsformen stellt eine Systementwicklungsschnittstelle der Softwareumgebung zum Konfigurieren eines Prozesssteuersystems eine Schnittstellenfunktion zum Auswählen einer Modulvorlage und zum Anfordern der Identifizierung aller unter Verwendung der Modulvorlage erstellten Modulinstanzen in einem ausgewählten Prozessbereich bereit. Optional kann die Systementwicklungsschnittstelle sowohl die identifizierten Modulinstanzen als auch die diesen Instanzen zugeordneten Konfigurationsoptionen anzeigen. Die Systementwicklungsschnittstelle kann eine weitere Funktion zur effizienten gemeinsamen Bearbeitung der identifizierten Instanzen bereitstellen (z. B. Auswählen oder Deselektieren einer Option der Vatermodulvorlage, Bearbeiten eines gemeinsamen Parameters, der durch die Vatermodulvorlage spezifiziert ist, usw.)

AUSFÜHRLICHE BESCHREIBUNG DER ZEICHNUNGEN

[0017] Fig. 1 ist eine schematische Darstellung eines Prozesssteuersystems, das eine Steuerung, mehrere Feldgeräte und eine Workstation umfasst, die verwendet werden kann, um die Steuerung und die Feldgeräte unter Verwendung von Modulvorlagen mit optionalen Komponenten zu konfigurieren.

[0018] Fig. 2 ist ein Blockdiagramm, das ein Beispiel einer bekannten hierarchischen Struktur von Steuerelementen veranschaulicht, die beim Konfigurieren von Prozesssteuersystemen wie dem in Fig. 1 dargestellten verwendet werden.

[0019] Fig. 3 stellt Komponenten eines bekannten Moduls, das zum Erstellen und Pflegen eines Steuerungsschemas einer verfahrenstechnischen Anlage verwendet wird, schematisch dar.

[0020] Fig. 4 ist ein Blockdiagramm, das den Prozess des Erzeugens mehrerer Modulinstanzen auf der Basis einer Modulvorlage mit einer oder mehreren optionalen Komponenten veranschaulicht.

[0021] Fig. 5 ist ein beispielhafter Bildschirm einer Benutzeroberfläche, der eine Modulvorlage mit optionalen Komponenten anzeigt.

[0022] Fig. 6A und Fig. 6B veranschaulichen beispielhafte Dialogbildschirme zum Konfigurieren von Eigenschaften von Komponenten eines Moduls.

[0023] Fig. 7 veranschaulicht eine beispielhafte Benutzeroberfläche zum Auswählen einer Modulvorlage mit optionalen Komponenten und Erzeugen einer Instanz eines Moduls unter Verwendung der Modulvorlage.

[0024] Fig. 8 ist ein beispielhafter Schnittstellenbildschirm in der Art eines Spreadsheets zum Anzeigen und Auswählen von Komponenten eines oder mehrerer Module, die unter Verwendung einer vorgegebenen Modulvorlage erzeugt wurden.

[0025] Fig. 9 ist ein beispielhafter Schnittstellenbildschirm in der Art eines Spreadsheets zum Anzeigen und Bearbeiten von Parameter eines oder mehrerer Module, die unter Verwendung einer vorgegebenen Modulvorlage erzeugt wurden.

[0026] Fig. 10 ist ein beispielhafter Schnittstellenbildschirm in der Art eines Spreadsheets zum Bearbeiten von bitmasken- oder strukturbasierten Parameter eines oder mehrerer Module, die unter Verwendung einer vorgegebenen Modulvorlage erzeugt wurden.

[0027] Fig. 11 und Fig. 12 sind beispielhafte Bildschirme Benutzeroberfläche, die Modulinstanzen anzeigen, die unter Verwendung der Modulvorlage von Fig. 5 erzeugt wurden.

[0028] Fig. 13 veranschaulicht einen Bildschirm der Benutzeroberfläche, der mehrere Komponenten eines Moduls zusammen mit Namenskürzeln von Komponenten und Gruppen von Komponenten anzeigt.

AUSFÜHRLICHE BESCHREIBUNG

[0029] Fig. 1 veranschaulicht ein beispielhaftes Prozesssteuersystem **10**, auf welches sich die Verfahren des Definierens, Bearbeitens und Verwendens von Modulvorlagen mit optionalen Komponenten anwenden lassen. Das Steuerungssystem **10** umfasst eine Prozesssteuerung **11**, die mit einem Data-Historian **12** und mit einer/einem oder mehreren Host-Workstations oder -Computern **13** (bei welchen es sich um einen beliebigen Typ von Personalcomputern, Workstations usw. handeln kann) verbunden ist, die jeweils einen Anzeigebildschirm **14** aufweisen. Die Steuerung **11** ist über Eingangs-/Ausgangs(I/O)-Karten **26** und **28** auch mit Feldgeräten **15** bis **22** verbunden. Der Data-Historian **12** kann ein beliebiger gewünschte Typ von Datenerfassungsein-

heit mit einem beliebigen gewünschten Speichertyp und einer beliebigen gewünschten oder bekannten Software, Hardware oder Firmware zum Speichern von Daten sein. Der Data-Historian **12** kann (wie in Fig. 1 dargestellt) von den Workstations **13** getrennt oder Teil einer der Workstations **13** sein. Die Steuerung **11**, bei der es sich beispielsweise um eine von der Fisher-Rosemount Systems, Inc. vertriebene DeltaV-Steuerung handeln kann, ist zum Beispiel über eine Ethernet-Verbindung oder ein beliebiges anderes gewünschtes Kommunikationsnetzwerk kommunikativ mit den Host-Computern **13** und dem Data-Historian **12** verbunden. Die Steuerung **11** ist außerdem unter Verwendung beliebiger gewünschter Hardware und Software, die zum Beispiel standardmäßigen 4–20-mA-Vorrichtungen und/oder einem beliebigen intelligenten Kommunikationsprotokoll, wie beispielsweise dem FOUNDATION Fieldbus-Protokoll, dem HART-Protokoll usw., zugeordnet ist, kommunikativ mit den Feldgeräten **15** bis **22** verbunden.

[0030] Die Feldgeräte **15** bis **22** können Vorrichtungen beliebigen Typs sein, wie beispielsweise Sensoren, Transmitter, Positionierer usw., während die I/O-Karten **26** und **28** I/O-Vorrichtungen beliebigen Typs sein können, die einem beliebigen gewünschten Kommunikations- oder Steuerungsprotokoll entsprechen. In der in Fig. 1 veranschaulichten Ausführungsform sind die Feldgeräte **15** bis **18** standardmäßige 4–20-mA-Vorrichtungen, die über analoge Leitungen mit der I/O-Karte kommunizieren, während die Feldgeräte **19** bis **22** intelligente Vorrichtungen wie beispielsweise Fieldbus-Feldgeräte sind, die über einen digitalen Bus unter Verwendung von Fieldbus-Protokoll-basierten Kommunikationsvorgängen mit der I/O-Karte **28** kommunizieren. Natürlich könnten die Feldgeräte **15** bis **22** beliebigen anderen gewünschten Draht- oder Drahtlosstandard(s) oder -protokollen entsprechen, einschließlich beliebiger Standards oder Protokolle, die in Zukunft entwickelt werden.

[0031] Die Steuerung **11** umfasst einen Prozessor **23**, der eine oder mehrere Prozesssteuerroutinen implementiert und überwacht, die Regelkreise umfassen können, und mit den Geräten **15** bis **22**, den Host-Computern **13** und dem Data-Historian **12** kommuniziert, um einen Prozess auf eine beliebige gewünschte Weise zu steuern. Die Steuerroutinen können in einem Steuerungsspeicher **24** gespeichert oder der Steuerung **11** anderweitig (z. B. unter den intelligenten Feldgeräten **19** bis **22** verteilt) zugeordnet sein. Es ist zu erwähnen, dass im Allgemeinen alle hierin beschriebenen Steuerroutinen oder -module Abschnitte enthalten können, die durch verschiedene Steuerungen oder andere Vorrichtungen implementiert oder ausgeführt werden können, falls gewünscht. Außerdem können die hierin beschriebenen Steuerroutinen oder -module, die im Prozesssteuersystem **10** implementiert werden sollen, jede beliebige Form annehmen, einschließlich Software, Firmware, Hardware usw. Zum Zwecke dieser Offenbarung kann ein Prozesssteuermodul ein beliebiger Teil oder Abschnitt eines Prozesssteuersystems sein, der zum Beispiel eine Routine, einen Block oder jegliches Element davon umfasst, das auf einem beliebigen computerlesbaren Medium gespeichert ist. Steuerroutinen, die Module oder beliebige Teile einer Steuerungsprozedur sein können, wie beispielsweise eine Subroutine, Teile einer Subroutine (wie beispielsweise Codezeilen) usw., können in jedem beliebigen gewünschten Softwareformat implementiert sein, wie beispielsweise unter Verwendung von objektorientierter Programmierung, unter Verwendung von Kontaktplan, Ablaufsprachen, Funktionsblockdiagrammen oder unter Verwendung von beliebigen anderen Software-Programmiersprachen oder Designparadigmen. Gleichermaßen können die Steuerroutinen zum Beispiel in einem oder mehreren EPROMs, EEPROMs, anwendungsspezifischen integrierten Schaltungen (ASIC) oder beliebigen anderen Hardware- oder Firmware-Elementen fest codiert sein. Demnach kann die Steuerung **11** so konfiguriert sein, dass sie eine Steuerungsstrategie oder Steuerroutine auf eine beliebige gewünschte Weise implementiert.

[0032] In einigen Ausführungsformen implementiert die Steuerung **11** eine Steuerungsstrategie unter Verwendung von Funktionsblöcken, obwohl sich die Verfahren zum Definieren, Bearbeiten und Verwenden von Modulvorlagen mit optionalen Parametern im Allgemeinen auf andere Steuerungsmethoden oder -regeln (z. B. Kontaktplan, Ablaufsprachen usw.) anwenden lassen können. Im Prozesssteuersystem **10** kann ein Funktionsblock jedem beliebigen Schema entsprechen, in welchem eine oder mehrere einer Steuerlogik, Betriebsmittellogik, Kommunikationslogik, Transducer-Logik usw. in einem Logikblock enthalten sind. Zur Vereinfachung der Erklärung werden die Begriffe „Logikblock" und „Funktionsbock" hierin austauschbar verwendet. Jeder Funktionsblock ist ein Objekt oder ein anderer Teil (z. B. eine Subroutine) der Gesamtsteuerungsstrategie ist und arbeitet mit anderen Funktionsblöcken über Kommunikationsverbindungen zusammen, um einen oder mehrere Prozessregelkreise innerhalb des Prozesssteuersystems **10** zu implementieren. Funktionsblöcke führen normalerweise eine von einer Eingangsfunktion wie beispielsweise jener, die einem Transmitter, einem Sensor oder einer anderen Vorrichtung zum Messen von Prozessparametern zugeordnet ist, einer Steuerfunktion wie beispielsweise jener, die einer Steuerroutine zugeordnet ist, die PID-, Fuzzy-Logik-Steuerung usw. durchführt, oder einer Ausgangsfunktion aus, die den Betrieb von bestimmten Vorrichtungen wie beispielsweise Ventilen steuert, um bestimmte physische Funktionen innerhalb des Prozesssteuersystems **10** auszuführen. Natürlich kön-

nen auch hybride und andere Typen von Funktionsblöcken vorhanden sein. Funktionsblöcke können in der Steuerung **11** gespeichert und von dieser ausgeführt werden, wie dies normalerweise der Fall ist, wenn diese Funktionsblöcke für standardmäßige 4–20-mA-Vorrichtungen und einige Typen von intelligenten Feldgeräten, wie beispielsweise HART-Vorrichtungen, verwendet werden oder diesen zugeordnet sind, oder sie können in den Feldgeräten selbst gespeichert und von diesen implementiert werden, wie dies bei Fieldbus-Vorrichtungen der Fall ist.

[0033] Wie durch die auseinandergezogene Ansicht von Block **30** in Fig. 1 veranschaulicht, kann die Steuerung **11** eine Anzahl von einschleifigen Steuerroutinen umfassen, die als Routine **32** und **34** dargestellt sind, und sie kann, falls gewünscht, eine oder mehrere erweiterte Regelkreise implementieren, die als Regelkreis **36** veranschaulicht sind. Jeder derartige Regelkreis wird normalerweise als Steuermodul oder einfach Modul bezeichnet. Die einschleifigen Steuerroutinen **32** und **34** sind so dargestellt, dass sie eine Einzelschleifenregelung unter Verwendung eines Fuzzy-Logik-Steuerblocks mit Einzeleingang und Einzelausgang und eines PID-Steuerblocks mit Einzeleingang und Einzelausgang ausführen, die jeweils an entsprechende Analogeingangs(AI)- bzw. Analogausgangs(AO)-Funktionsblöcke angeschlossen sind, die Prozesssteuerungsvorrichtungen wie beispielsweise Ventilen, Messvorrichtungen wie beispielsweise Temperatur- und Druckgebern oder beliebigen anderen Vorrichtungen im Prozesssteuersystem **10** zugeordnet sein können. Der erweiterte Regelkreis **36** ist so dargestellt, dass er einen erweiterten Steuerblock **38** mit Eingängen, die mit einem oder mehreren AI-Funktionsblöcken kommunikativ verbunden sind, und Ausgängen umfasst, die mit einem oder mehreren AO-Funktionsblöcken verbunden sind, obwohl die Eingänge und Ausgänge des erweiterten Steuerblocks **38** mit beliebigen anderen gewünschten Funktionsblöcken oder Steuerelementen verbunden sein können, um andere Typen von Eingaben zu empfangen und andere Typen von Steuerungsausgaben bereitzustellen. Der erweiterte Steuerblock **38** kann ein beliebiger Typ eines modellprädiktiven Steuer(MPC)-Blocks, eines Modellier- oder Steuerblocks für neuronale Netzwerke, eines Fuzzy-Logik-Steuerblocks mit mehreren Variablen, eines Echtzeit-Optimierungsblocks usw. sein. Es versteht sich von selbst, dass die in Fig. 1 dargestellten Funktionsblöcke, einschließlich des erweiterten Steuerblocks **38**, von der Steuerung **11** ausgeführt werden können oder alternativ in beliebigen anderen Verarbeitungsvorrichtungen, wie beispielsweise in einer der Workstations **13** oder auch einem der Feldgeräte **19** bis **22**, angeordnet sein und von diesen ausgeführt werden können.

[0034] Um die Steuerungsstrategie des Prozesssteuersystems **10** ohne übermäßigen Verbrauch von Speicherkapazität und Verarbeitungsressourcen schnell und effizient zu definieren, kann ein autorisierter Benutzer (z. B. ein Konfigurationsingenieur, ein Bediener usw.) Modulvorlagen mit einer oder mehreren optionalen Komponenten und null oder mehreren obligatorischen Komponenten definieren und verwenden. Insbesondere kann der Benutzer eine Modulvorlage entwickeln, indem er durch Funktionsblöcke auswählt oder definiert, die ausgewählten Funktionsblöcke miteinander verbindet, um eine Steuerroutine eines Moduls zu definieren, Prozessparameter des Moduls definiert, den Parametern Standardwerte zuweist usw. Wie im Folgenden unter Bezugnahme auf Fig. 4 bis 11C ausführlich erörtert wird, kann der Benutzer einigen oder allen Komponenten ein auswählbares Attribut zuweisen, um anzuzeigen, dass diese Komponenten in einer bestimmten Modulinstanz (oder nur „Instanz"), die auf der Modulvorlage basiert, optional ist. Der Benutzer kann auf ähnliche Weise optionale Parameter oder andere Modulattribute identifizieren, falls gewünscht, und die Modulvorlage zur Verwendung beim Konfigurieren des Prozesssteuersystems **10** in einem Repository, wie beispielsweise einer Datenbank, speichern. Die folgenden Beispiele veranschaulichen die anschließende Verwendung von Modulvorlagen mit optionalen Komponenten eingehender, die ein selektives Aktivierens von Komponenten, die in der Modulvorlage über auswählbare Attribute verfügbar sind, wenn eine Instanz für eine bestimmte Anwendung erzeugt wird, Herunterladen einer Instanz, die nur die obligatorischen und ausgewählten Komponenten aufweist, auf eine Steuerung und/oder ein intelligentes Feldgerät, Bearbeiten der Instanz gemäß der Modulvorlage und der ausgewählten optionalen Komponenten, Bearbeiten mehrerer Instanzen, die von einer gemeinsamen Modulvorlage abgeleitet sind, usw. umfasst. Es ist zu erwähnen, dass, obwohl die Verfahren in Bezug auf die Bearbeitung von optionalen Komponenten in erster Linie unter Bezugnahme auf Modulvorlagen erörtert werden, ähnliche Verfahren auch auf Modulklassen angewendet werden können. In der folgenden Erörterung werden die Begriffe „Modulvorlage" und „Modulklasse" daher austauschbar verwendet.

[0035] In dem in Fig. 1 veranschaulichten Beispiel umfassen die Workstations **13** (entweder individuell, auf verteilte Weise oder eine beliebige andere Art) ein Paket von Bedienerschnittstellenanwendungen **50**, die Operationen in Bezug auf Modulvorlagen mit optionalen Komponenten unterstützen. Ein autorisierter Benutzer kann das Paket von Bedienerschnittstellenanwendungen **50** auch verwenden, um auf verschiedene Funktionen von Vorrichtungen, Einheiten und anderen Elementen, die innerhalb des Prozesssteuersystems **10** angeschlossen sind, zuzugreifen und diese zu bearbeiten und zu überwachen. Das Paket von Bedienerschnittstellenanwendungen **50** kann sich in einem Speicher **52** der Workstation

13 befinden, und jede der Anwendungen oder Entitäten innerhalb des Pakets von Anwendungen **50** kann so ausgelegt sein, dass es auf jeweiligen Prozessoren **54**, die jeder Workstation **13** zugeordnet sind, ausgeführt wird. Obwohl das gesamte Pakt von Anwendungen **50** so dargestellt ist, dass es in der Workstation **13** gespeichert ist, können einige dieser Anwendungen oder andere Entitäten in anderen Workstations oder Rechengeräten gespeichert und ausgeführt werden, die innerhalb des Systems **10** oder diesem zugeordnet sind oder damit in Verbindung stehen. Außerdem kann das Paket von Anwendungen **50** Anzeigeausgaben für den der Workstation **13** zugeordneten Anzeigebildschirm **14** oder beliebige andere gewünschte Anzeigebildschirme oder Anzeigevorrichtungen, einschließlich tragbarer Vorrichtungen, Laptops, anderer Workstations, Drucker usw., bereitstellen. Gleichermaßen können die Anwendungen innerhalb des Pakets von Anwendungen **50** auf zwei oder mehr Computern oder Maschinen aufgeteilt und darauf ausgeführt werden, und sie können so konfiguriert sein, dass sie in Verbindung miteinander arbeiten.

[0036] Zur Unterstützung der hierin erörterten Verfahren kann das Paket von Anwendungen **50** eine Vorlagenentwicklungsanwendung oder -schnittstelle zum Erstellen und Modifizieren von Modulvorlagen mit optionalen Komponenten und eine Systemkonfigurationsanwendung oder -schnittstelle zum Anwenden (d. h. Instanziieren) von Modulvorlagen auf einen bestimmten Prozessbereich umfassen. Falls gewünscht, kann eine einzige Anwendung oder ein einziger Anwendungsrahmen sowohl die Modulvorlagenentwicklung als auch die Modulvorlagenanwendung unterstützen. Als ein Beispiel kann das Paket von Anwendungen **50** der von Emerson Process Management entwickelten digitalen PlantWeb®-Architektur entsprechen und konkret eine oder mehrere DeltaV-Anwendungen (z. B. Control Studio, Plant Explorer, Operate, usw.) umfassen. Die Unterstützung für die Entwicklung und Anwendung von Modulvorlagen mit optionalen Komponenten kann dann der Control Studio-Anwendung hinzugefügt werden, während Modulvorlagen mit optionalen Parametern zusammen mit den herkömmlicheren Vorlagen und Funktionen in der DeltaV-Bibliothek gespeichert werden können. Um mit diesem Beispiel fortzufahren, kann Plant Explorer weitere Unterstützung für Operationen in Bezug auf Modulvorlagen bereitstellen, indem Benutzer befähigt werden, die Modulvorlagenbibliothek zu durchsuchen und die gewünschten Modulvorlagen auszuwählen, unter Verwendung von Modulvorlagen erzeugte Modulinstanzen anzuzeigen, zu bestimmen, welche der enthaltenen Komponenten optional sind, und andere Operationen an der Instanz durchzuführen. Es ist zu erkennen, dass die Funktionalität in Bezug auf Modulvorlagen einer oder mehreren Softwareanwendungen auf beliebige Weise zugeordnet werden kann.

[0037] Zur besseren Darstellung der Struktur und der Funktionsweise von Modulvorlagen in einer Prozesssteuerungsumgebung veranschaulicht Fig. 2 eine allgemein anerkannte hierarchische Struktur von Steuerelementen, die beim Konfigurieren des Prozesssteuersystems **10** verwendet werden können, einschließlich einer Modulebene. Fig. 3 stellt dann üblicherweise verwendete Komponenten eines Moduls schematisch dar, das im Allgemeinen der Modulebene der in Fig. 2 dargestellten hierarchischen Struktur zugeordnet ist. Als Nächstes veranschaulichen Fig. 4 bis Fig. 10 die Verfahren des Entwickelns und Anwendens von Modulvorlagen mit optionalen Komponenten unter Bezugnahme auf verschiedene Beispiele von Benutzeroberflächen des Pakets von Bedienerschnittstellenanwendungen **50**. Konkret stellt Fig. 4 im Allgemeinen das Verfahren des Erzeugens einer Instanz eines Moduls auf der Basis einer Modulvorlage mit optionalen Komponenten dar; Fig. 5 bis Fig. 6B stellen einen beispielhaften Bildschirm der Benutzeroberfläche zur Entwicklung einer Modulvorlage mit optionalen Komponenten dar; und **Fig.** 7A bis 10C veranschaulichen mehrere Beispiele von unter Verwendung dieser Modulvorlage erzeugten Instanzen, wie auf einer entsprechenden Benutzeroberfläche dargestellt.

[0038] Bezug nehmend auf Fig. 2 stellt ein hierarchisches Modell **100** eine technische Vorgehensweise zur Entwicklung einer Prozesssteuerungsstrategie von oben nach unten dar. Ausgehend von einer Anlage **102** auf der höchsten Ebene einer baumähnlichen Struktur umfasst das Modell **100** mehrere Ebenen, auf welchen ein Benutzer Steuerelemente anzeigen oder konfigurieren kann. Die Anlage **102** kann zum Beispiel ein Chemiewerk, eine Ölraffinerie, eine automatische Fabrik oder eine beliebige andere Umgebung mit einem gesteuerten und (wenigstens teilweise) automatischen Prozess sein. Wie in Fig. 2 dargestellt, kann die Anlage **102** einen oder mehrere Anlagenbereiche **104** umfassen, welche ihrerseits Module **106** umfassen können, um eine grundlegende oder erweiterte Steuerungsaufgabe durchzuführen, wie unter Bezugnahme auf Fig. 2 ausführlicher erörtert wird. Ein Funktionsblockdiagramm **108** kann die Steuerlogik des entsprechenden Moduls **106** als einen (in einem trivialen Fall) oder mehrere (in einem nicht trivialen Fall) miteinander verbundene Funktionsblöcke **110** definieren. In diesem Sinne definiert das Funktionsblockdiagramm **108** die Struktur des Moduls **106**. Die Funktionsblöcke **110** wiederum entsprechen Parameter **112**. In diesem Beispiel ist ein Proportional-Integral-Derivative(PID)-Funktionsblock für eine Steuerungsfunktion verantwortlich, die von Abstimmparametern wie beispielsweise Verstärkung, einem bestimmten Sollwert (z. B. Zieldruck), einem Eingangssignal usw. abhängt. Im Paket von Anwendungen **50** kann jedes der Elemente **102** bis **112** durch eine Datenstruktur, ein Softwareobjekt oder eine beliebige andere Datenaggregation dargestellt

sein. Wenigstens einige der Elemente entsprechen auch physischen Elementen im Prozesssteuersystem **100**, Teilen von physischen Elementen oder umgekehrt Gruppierungen von physischen Elementen Zum Beispiel kann ein PID-Schleifenmodul **106** einem Ventil, einem Positionierer und einem Sensor entsprechen, während der Verstärkungsabstimmungsparameter **112** einem Wert, der in einem Steuerungsspeicher gespeichert ist, einem Signal, das über eine analoge Signalisierungsleitung weitergeleitet wird, usw. entsprechen kann.

[0039] Als Nächstes veranschaulicht Fig. 3 die Komponenten des Moduls **106**, die unter Bezugnahme auf Fig. 3 beschrieben werden. Um mit physischen Entitäten unter Verwendung von drahtgebundener oder drahtloser Signalisierung zu interagieren, umfasst das Modul **106** I/O-Komponenten **120**. Vom Standpunkt des Pakets von Anwendungen **50** können die I/O-Komponenten **120** protokollspezifische Treiber sein, die einem bestimmten Softwareobjekt zugeordnet sind. Andererseits zeigt das Modul **106** Benutzern grafische Informationen über Anzeigen **122** an, die zum Beispiel ein grafisches Symbol, einen Dynamo mit Animationsfähigkeiten usw. umfassen können. Das Modul **106** kann auch eine Verlaufserfassungskomponente **124**, um Ereignisdaten zu erfassen und zu verwalten, eine Alarmkomponente **126**, eine Bedingungskomponente **128** und eine Algorithmuskomponente **130** umfassen, um den Betrieb der Komponenten des Moduls **106** zum Beispiel als Funktionsblockdiagramm **108** (siehe Fig. 2) zu verwalten. Wie in Fig. 3 dargestellt, können Parameter **132** verschiedenen Komponenten **120** bis **130** zugeordnet sein.

[0040] Bezug nehmend auf Fig. 4 kann eine Konfigurationsumgebung oder ein Konfigurationssystem **150**, das zum Beispiel durch das Paket von Anwendungen **50** (siehe Fig. 1) unterstützt wird, wenigstens einige der unter Bezugnahme auf Fig. 2 und Fig. 3 erörterten allgemeinen Methoden verwenden. Insbesondere kann eine Modulvorlagenbibliothek **152** eine Anzahl von Modulvorlagen zur Verwendung in verschiedenen Anwendungen und Szenarien enthalten, die zum Beispiel eine Modulvorlage **154** für eine PID mit Verstärkungsplanungssteuerung, eine Modulvorlage **156** für PID mit Totzeitkompensations-Regelkreis und eine Modulvorlage **158** für einen standardmäßigen PID-Regelkreis umfassen, wie in einer auseinander gezogenen Ansicht dargestellt. Im Allgemeinen behandelt jede Modulvorlage **154**, **156** und **158** eine breite Auswahl von Funktionsanforderungen. Benutzer, die zur Modulvorlagenbibliothek **152** beitragen, können Konfigurationsingenieure sein, die zum Beispiel Daten in Bezug auf zahlreiche Projekte analysieren und möglicherweise versuchen, jedes mögliche Szenarium in eine einzige Modulvorlage einzubeziehen. Andere Beitragende können Benutzer sein, die Vorlagen in Vorbereitung für eine häufige anschließende Duplizierung einer bestimmten identischen oder ähnlichen Funktionalität in einem bestimmten Projekt entwickeln, und es können dieselben Benutzer sein, die später unter Verwendung dieser Modulvorlagen Instanzen erstellen. In jedem Fall enthalten die Modulvorlagen **154**, **156** und **158** normalerweise wenigstens einige Komponenten (z. B. Funktionsblöcke, Gruppe von Funktionsblöcken, Parameter usw.), die ein Benutzer im Kontext einer bestimmten Anwendung überflüssig findet.

[0041] Um diese und andere Überlegungen (z. B. Sparen von Steuerungsspeicher, Bewahren der Verarbeitungsleistung, Reduzieren der Komplexität der Darstellung eines Moduls für einen Benutzer usw.) zu behandeln, ermöglicht die Konfigurationsumgebung es Benutzern zum Beispiel, Instanzen mit verschiedenen Funktionsblöcken von derselben Modulvorlage **158** abzuleiten. Obwohl die Instanzen **160** und **162** auf der Modulvorlage **158** basieren, unterscheidet sich die Instanz **160** folglich strukturell von der Instanz **162**. Selbstverständlich können die Instanzen **160** und **162** auch verschiedene Parameter enthalten.

[0042] Beim Entwerfen und Bearbeiten der Modulvorlage **158** kann ein Benutzer, wie beispielsweise ein Konfigurationsingenieur, Funktionsblöcken **170**, **172** und **174** ein obligatorisches Attribut zuweisen, um anzuzeigen, dass wenigstens die Funktionsblöcke **170**, **172** und **174** in jeder auf der Modulvorlage **158** basierten Instanz anwesend sein müssen. Alternativ kann die Konfigurationsumgebung **150** das obligatorische Attribut jedem Funktionsblock in einer Modulvorlage standardmäßig zuordnen, und der Benutzer kann das Standardattribut gegebenenfalls außer Kraft setzen, falls gewünscht. Außerdem kann der Ingenieur Funktionsblöcken **176** bis **179** ein optionales Attribut zuweisen, um anzuzeigen, dass ein Benutzer später wählen kann, ob er diese Funktionsblöcke in eine bestimmte Instanz aufnimmt. Falls gewünscht, kann der Ingenieur die Funktionsblöcke **178** und **179** zu einer Gruppe zusammenschließen und den Funktionsblöcken **178** und **179** ein gemeinsames optionales Attribut zuweisen. Darüber hinaus kann der Ingenieur einem Funktionsblock **182** ein ersetzbares Attribut zuweisen, um die Ersetzung des Funktionsblocks **182** durch verschiedene Blöcke zu ermöglichen. Wenigstens in einigen Fällen spezifiziert der Ingenieur eine zulässige Gruppe von Funktionsblöcken, aus welchen der Funktionsblock **182** ausgewählt werden kann. In dem in Fig. 4 veranschaulichten Beispiel spezifiziert die Modulvorlage **158** drei Funktionsblöcke, die als der Funktionsblock **182** verwendet werden könnten. Bei Fertigstellung des Entwurfs der Modulvorlage **158** kann der Ingenieur die Modulvorlage **158** in der Bibliothek **152** speichern, die sich im Speicher der Workstation **13** (siehe Fig. 1), in einer separaten Datenbank, in einem Online-Repository usw. befindet.

[0043] Eine Vorlagenerzeugungsmaschine **184**, die in der Konfigurationsumgebung läuft, kann die Definition einer neuen Modulvorlage, Modifikationen an einer bestehenden Modulvorlage und andere mit der Konfiguration und Verwaltung von Vorlagen in Verbindungen stehende Funktionen unterstützen. Die Vorlagenerzeugungsmaschine **184** kann mit einer Benutzeroberfläche zusammenarbeiten, die es Benutzern ermöglicht, eine oder mehrere Komponenten (z. B. Funktionsböcke **176** bis **179**) auszuwählen, einzelnen Komponenten oder Gruppen von Komponenten obligatorische oder optionale Attribute zuzuweisen und andere Funktionen durchzuführen, die im Folgenden ausführlicher erörtert werden. Außerdem kann eine Modulinstanziierungsmaschine **186** als Reaktion auf Benutzerbefehle Modulinstanzen **160** und **162** erzeugen. Insbesondere kann die Modulinstanziierungsmaschine **186** Speicherplatz für die obligatorische Komponente und die ausgewählten optionalen Komponenten zuordnen, auf einer Steuerung und/oder einem Feldgerät ausführbare Anweisungen erzeugen usw. Zusätzlich kann die Modulinstanziierungsmaschine **186** die Erfassung von Altdaten (z. B. Ereignisdaten) für die ausgewählten optionalen Komponenten in den erzeugten Modulinstanzen **160** und **162** ermöglichen.

[0044] Vor dem Erzeugen oder zum Zeitpunkt des Erzeugens der Instanzen **160** und **162** (der Instanziierungsprozess ist durch die Pfeile **190** und **192** dargestellt) kann die Konfigurationsumgebung dem Benutzer während jeweiliger Optionsauswahlstufen **194** und **196** anzeigen, dass die „Vater"-Modulvorlage **158** optionale Komponenten enthält. Die Konfigurationsumgebung **150** kann ein grafisches Dialogfenster, ein textbasiertes Dialogfenster, ein Spreadsheet oder beliebige andere Formen von Benutzeroberflächen anzeigen, um zu verlangen, dass der Benutzer spezifiziert, welche der optionalen Komponenten **176** bis **179** in die Instanz **160** oder **162** aufgenommen werden sollte und welcher der drei möglichen Funktionsblöcke dem Funktionsblock **182** zugeordnet werden sollte. Optional kann die Konfigurationsumgebung **150** auch anzeigen, dass die Funktionsblöcke **170**, **172**, und **174** obligatorisch sind. In diesem Beispiel wählt der Benutzer beim Erstellen der Instanz **160** alle in der Modulvorlage **158** verfügbaren Funktionsblöcke aus und wählt, die Funktionsblöcke **177**, **178** und **179** aus der Instanz **162** wegzulassen. Ferner spezifiziert der Benutzer, dass der Funktionsblock **182** in der Instanz **160** vom Typ 1 und in der Instanz vom Typ 2 ist. Der Benutzer erzeugt demnach zwei strukturell verschiedene Instanzen **160** und **162** auf der Basis der gemeinsam benutzten Modulvorlage **158**.

[0045] In grafischen Darstellungen der Instanzen **160** und **162** kann die Konfigurationsumgebung **150** das ursprüngliche strukturelle Layout der Vatermodulvorlage **158** bewahren und Leerstellen anzeigen, an welchen die deselektierten Komponenten in der Modulvorlage **158** angeordnet sind. Auf diese Weise stellt die Konfigurationsumgebung **150** einen bestimmten Grad des Bedienungskomforts für den Benutzer bereit, da die Instanzen **160** und **162** leicht als Ableitungen der Modulvorlage **158** identifiziert werden können. Sollte außerdem der Benutzer zu einem späteren Zeitpunkt entscheiden, einige der ursprünglich abgelehnten optionalen Komponenten zu reaktivieren, zeigt die Konfigurationsumgebung **150** diese reaktivierten Komponenten in den vorbelegten Positionen an. Um ein konkretes Beispiel zu nehmen, kann der Benutzer beschließen, die Instanz **162** zu bearbeiten und den optionalen Funktionsblock **177** zu einem späteren Zeitpunkt auszuwählen. Die Konfigurationsumgebung **150** zeigt den Funktionsblock **177** in der durch die Modulvorlage **158** spezifizierten Position an.

[0046] Weiterhin Bezug nehmend auf Fig. 4 kann jede der Instanzen **160** und **162** eine Vatervorlagenkennung **200** enthalten, welche diese Instanzen logisch mit der Modulvorlage **158** verknüpft. Wie im Folgenden ausführlicher erörtert wird, kann die Vatervorlagenkennung **200** für verschiedene Zwecke verwendet werden, welche die Identifizierung aller auf der Modulvorlage **158** basierten Instanzen innerhalb eines bestimmten Prozessbereichs, die gleichzeitige Bearbeitung der Instanzen **160** und **162** unter Verwendung eines gemeinsamen Dialogs usw. umfassen. Ferner kann jede der Instanzen **160** und **162** auch ein entsprechendes Ausgewählte-Optionen-Feld **202** enthalten, um anzuzeigen, welche der in der Modulvorlage **158** verfügbaren Optionen in der jeweiligen Instanz aktiviert wurden. In einigen Ausführungsformen kann das Ausgewählte-Optionen-Feld **202** eine Bitmaske sein, wobei jedes Bit einer optionalen Komponente entspricht und ein im Bit gespeicherter Wert die Anwesenheit oder Abwesenheit der Komponente in der Instanz anzeigt. Jede der Vatervorlagenkennung **200** und des Ausgewählte-Optionen-Feldes **202** kann als Teil von Objektdaten der Instanzen **160** oder **162** gespeichert werden oder separat gespeichert und mit den Instanzen **160** und **162** logisch verknüpft werden.

[0047] In einigen Ausführungsformen kann die Konfigurationsumgebung **150** die Vatervorlagenkennung **200** als ein Form von Schutz verwenden, um Benutzer auf die Bearbeitung nur jener Komponenten und Parameter einzuschränken, die durch die Modulvorlage spezifiziert sind. Nötigenfalls kann der Benutzer den Schutz unter Verwendung eines vordefinierten Befehls oder Vorgangs aufheben. Die Konfigurationsumgebung **150** kann es Benutzern auch ermöglichen, Verknüpfungen zwischen Instanzen und Modulvorlagen zu trennen. Zum Beispiel kann der Benutzer wünschen, strukturelle Änderungen an der Instanz **162** außerhalb des Rahmens vorzunehmen, der durch die Vatermodulvorlage **158** vorgeschrieben

ist. Die Konfigurationsumgebung **150** kann automatisch einen Dialog erzeugen, um den Benutzer zu informieren, dass durch die Vornahme dieser Änderungen die Verbindung mit der Vatervorlage verloren geht. Bei Empfang einer Bestätigung vom Benutzer kann die Konfigurationsumgebung **150** die Felder **200** und **204** löschen oder modifizieren, um die Instanz **162** von der Modulvorlage **158** zu trennen. Bei nachfolgenden Operationen behandelt die Konfigurationsumgebung **150** die Instanz als „von Grund auf" gebildet, d. h. aus Basisfunktionsblöcken, die in einer herkömmlichen Bibliothek verfügbar sind.

[0048] Nunmehr unter Bezugnahme auf Fig. 5 kann die Konfigurationsumgebung **150**, die durch einige oder alle des Pakets von Anwendungen **50** (siehe Fig. 1) unterstützt wird, einen beispielhaften Schnittstellenbildschirm oder ein beispielhaftes Schnittstellenfenster **220** zum Konfigurieren von Modulvorlagen, wie beispielsweise die zuvor erörterten Modulvorlagen **154**, **156** und **158**, umfassen. Der Schnittstellenbildschirm **220** kann eine Symbolleiste **222** mit textlichen Pull-down-Menüs **224** und einem piktografischen Menü **226**, ein Schablonenfenster **228**, das Schablonenelemente **240** anzeigt, und ein Diagrammfenster **242** enthalten. Falls gewünscht, kann die Symbolleiste **222** eine Schnittstelle in der Art einer Funktionsleiste sein, die zum Beispiel der von der Microsoft Corporation entwickelten ähnelt. Das Schablonenfenster **228** kann eine beliebige Anzahl von vordefinierten Steuerelementen oder -blöcken wie beispielsweise Funktionsblöcken oder Betriebsmittelblöcken enthalten. In einigen Ausführungsformen kann das Schablonenfenster **228** auch eine thematische Anordnung von Blöcken (z. B. allen Blöcke in Bezug auf Rückkopplungsregelung) oder andere Gruppierungsformen bereitstellen. Ferner kann jedes Schablonenelement ein Symbol (z. B. ein allgemein verwendetes technisches Symbol, ein erläuterndes Bild usw.) zur schnellen visuellen Identifizierung der damit verbundenen Funktion umfassen. Beim Entwerfen einer Modulvorlage unter Verwendung des Schnittstellenbildschirms **220** kann ein Benutzer ein Schablonenelement aus dem Schablonenfenster **228** auswählen, das ausgewählte Schablonenelement zu einer gewünschten Stelle innerhalb des Diagrammfensters **242** ziehen und das ausgewählte Schablonenelement an der gewünschten Stelle ablegen. Das Paket von Anwendungen **50** kann demnach eine intuitive Konfigurationsumgebung bereitstellen, wobei Benutzer eine einfache Drag-and-Drop-Schnittstelle verwenden können, um schnell Modulvorlagen zu bilden. Es ist jedoch zu erwähnen, dass die Konfigurationsumgebung **150** auch über einen anderen Benutzerschnittstellentyp wie beispielsweise Textbefehle, um ein Beispiel zu nennen, mit Benutzern interagieren kann.

[0049] Im Allgemeinen kann die Konfigurationsumgebung **150** in Bezug auf die Benutzeroberfläche einige oder alle eines Haupt-/Bearbeitungsbereichs zur textlichen oder grafischen Konfiguration von Modulen, Steuerroutinen und anderen Teile der Steuerungsstrategie des Prozesssteuersystems **10**; eines Explorerbereichs zum Durchsuchen verschiedener Bibliotheken-Steuerelementen und Prozessbereiche; eines Palettenbereichs zum Darstellen von Symbolen zur raschen Auswahl und Anordnung im Hauptbereich durch Ziehen und Ablegen, zum Beispiel mit einer optionalen Unterteilung in Funktionsgruppen; eines Alarmbereichs zur Anzeige von Kennungen, Prioritäten und Beschreibungen verschiedener Alarme; eines Parameterbereichs zum Auflisten von Namen, Standardwerten, aktuellen Werten usw. verschiedener Betriebsparameter; eines Referenzbereichs zum Auflisten von Adressen physischer Module; eines Schiebereglers, damit ein Benutzer leicht die gewünschte Größe von Grafiken auswählen kann, die im Hauptbereich dargestellt werden; eines Beobachtungsbereichs; eines Formelbereichs; einer Statusleiste zur Anzeige des aktuellen Status der Konfigurationsumgebung **150** (z. B. „Bearbeiten") usw. umfassen. Kurz gesagt, können Informationen in Bezug auf das Prozesssteuersystem **10** im Allgemeinen und auf Module und Modulvorlagen im Besonderen auf beliebige gewünschte Weise dargestellt werden.

[0050] Im Diagrammfenster **242** veranschaulicht Fig. 5 ein konkretes Beispiel einer Modulklasse oder -vorlage, die unter Verwendung der hierin behandelten Verfahren entwickelt wurde. Insbesondere unterstützt die Modulklasse PID_STD einen standardmäßigen PID-Algorithmus zum Steuern einer Prozessvariablen auf der Basis eines Sollwerts und eines Fehlersignals. Ein Konfigurationsingenieur kann die Steuermodulvorlage PID_STD so auslegen, dass sie den Großteil der oder, falls möglich, die gesamte Leistungsfähigkeit enthält, die für die komplexeste in Erwägung gezogene PID-Steuerung erforderlich ist. In dieser Hinsicht ähnelt die Modulvorlage PID_STD der in der PBL-Bibliothek verfügbaren Vorlage PID_LT. Es ist jedoch zu erwähnen, dass Fig. 5 der Klarheit und der Kürze halber bestimmte Blöcke und Parameter einer PID_LT-Vorlage nicht darstellt.

[0051] Ein Diagramm **250** veranschaulicht ein mögliches Layout der Komponenten der Modulvorlage PID_STD. Beim Entwurf dieser und anderer Modulvorlagen kann der Benutzer auf ein Symbol **252** im piktografischen Menü zeigen und klicken, um ein Bereichswähler-Tool zu aktivieren, einen Rahmen (oder eine andere Form) **254** um einen Verriegelungsbedingungs-Funktionsblock **256** und einen Boole-Eingangslastfaktor(BFI)-Funktionsblock **258** herum ziehen, um einen Bereich **260** zu definieren, mit der rechten Maustaste klicken, um ein Menü „Eigenschaften" zu öffnen, und das optionale Attribut den Verriegelungsbedingungs-Funktionsblöcken **256** und **258** zuweisen. Falls gewünscht, kann das optionale Attribut für die Verriegelungsbedingungs-Funktions-

blöcke **256** und **258** im Ganzen gelten. Alternativ kann das Schnittstellenfenster **220** ein Symbol im piktografischen Menü **226** zum Aktivieren einer Funktion enthalten, welche diese Schritte des Auswählens eines oder mehrerer Funktionsblöcke und Zuweisens des optionalen Attributs zur Auswahl kombiniert. Selbstverständlich kann das Schnittstellenfenster **220** oder ein anderes Schnittstellenelement der Konfigurationsumgebung **150** beliebige gewünschte Typen von Auswahl- und Eigenschaftszuweisungs-Tools, wie beispielsweise Textmarker-Tools, textbasierte Befehlsdialoge, Programmierungsskriptunterstützung usw. enthalten.

[0052] Weiterhin Bezug nehmend auf Fig. 5 kann der Benutzer in ähnlicher Weise einen Rahmen **270** um einen Verriegelungsbedingungs-Funktionsblock **272** ziehen, um einen Bereich **274** zu definieren. In einigen Ausführungsformen kann die Konfigurationsumgebung **150** es einem Benutzer ermöglichen, das optionale Attribut direkt einem ausgewählten Funktionsblock zuzuordnen, wenn zum Beispiel der Benutzer nicht wünscht, mit Aggregaten von Funktionsblöcken zu arbeiten. Demgemäß könnte der Benutzer auf den Verriegelungsbedingungs-Funktionsblock **272** klicken, ein Optionsauswahlmenü öffnen und das optionale Attribut diesem zuweisen. Wie in Fig. 5 weiter veranschaulicht, kann der Benutzer auch getrennte Rahmen um die Verriegelungsbedingungs-Funktionsblöcke **276** und **278** sowie um mehrere Funktionsblöcke und Parameter in einem Bereich **280** zeichnen, die gemeinsam für die Gerätezuteilungslogik verantwortlich sind. Es ist zu erkennen, dass der Bereich **280** zusätzlich zu einem PID-Zuteilungsblock **282** auch mehrere Parameter wie beispielsweise ACQUIRE_ID (Eingangsparameter **284**), OVR_ENAB (Eingangsparameter **286**) und OWNER_ID (Ausgangsparameter **288**) enthält. Demnach unterstützt die Konfigurationsumgebung **150** eine Auswahl von Funktionsblöcken, Parametern oder Kombinationen von Funktionsblöcken und Parametern.

[0053] Neben dem Zuweisen der optionalen Eigenschaft zu Funktionsblöcken oder Gruppen von Funktionsblöcken kann der Benutzer optional jedem der Bereiche **260**, **274**, **280** usw. auch einen Namen zuweisen. Zum Beispiel kann der Benutzer dem Bereich **260** den Namen TRACK 1 (Spur 1), dem Bereich **274** TRACK 2 (Spur 2) und dem Bereich **280** ARBITRATION (Zuteilung) zuweisen. Vorzugsweise spiegelt jeder Name die aggregierte Funktion wider, die dem ausgewählten Bereich im spezifischen Kontext der Modulvorlage zugeordnet ist. Genauer gesagt zeigt der Name TRACK 1 einem künftigen Benutzer der PID_STD-Vorlage an, dass der entsprechende Verriegelungsbedingungs-Funktionsblock **272** einen zweiten Eingang oder eine zweite „Spur" (track) in einer mehrspurigen PID-Schleife versorgen kann.

[0054] Um mit dem unter Bezugnahme auf Fig. 5 erörterten Beispiel fortzufahren, kann der Benutzer die Modulvorlage PID_STD in einer Vorlagenbibliothek des Prozessteuersystems **10** (zum Beispiel im Speicher **52** angeordnet), in einem Online-Repository oder in einem separaten Speichergerät speichern. Falls gewünscht, kann der Benutzer die Modulvorlage PID_STD einer Modulklasse zuordnen. In jedem Fall kann die Modulvorlage PID_STD in einem vordefinierten Format gespeichert werden, das vom Konfigurationssystem **150** erkannt wird, so dass derselbe oder ein anderer Benutzer Modul-Instanzen auf der Basis der Modulvorlage PID_STD erstellen kann.

[0055] Fig. 6A und Fig. 6B veranschaulichen beispielhafte Menüs und Dialoge, welche die Konfigurationsumgebung **150** anzeigen kann, um einige der zuvor erörterten Attributzuweisungsverfahren zu unterstützen. Konkret veranschaulicht Fig. 6A einen Funktionsblock **300** und ein Pull-down-Menü **302**, das zum Beispiel als Reaktion auf ein Doppelklicken des Benutzers auf dem Funktionsblock **300** in einem Popup-Fenster geöffnet werden kann. Das Pull-down-Menü **302** kann einen Menüeintrag Eigenschaften enthalten, der bei Auswahl einen Benutzerdialog **304** zum Aktivieren oder Deaktivieren der optionalen Eigenschaft und Zuweisen eines Namenskürzels des Funktionsblocks **300** öffnen kann. In diesem Beispiel umfasst der Benutzerdialog **304** ein Optionsfeld mit den Optionen ENABLE (Aktivieren) und DISABLE (Deaktivieren) und ein Texteingabefeld **308**, das der Benutzer optional ausfüllen kann. Fig. 6B stellt ein ähnliches Pull-down-Menü **312** und einen ähnlichen Benutzerdialog **314** dar. Die Konfigurationsumgebung **150** wendet das Pull-down-Menü **312** jedoch auf einen Bereich innerhalb eines Rahmens **316** an, der die Funktionsblöcke **31** und **320** umgibt. Anders ausgedrückt, gilt die über den Dialog **314** spezifizierte Option für eine Gruppierung von mehreren Funktionsblöcken.

[0056] In einem Beispiel, das in Fig. 7 veranschaulicht ist, stellt ein Schnittstellenfenster **350** des Konfigurationssystems **150** eine praktische und effiziente Drag-and-Drop-Schnittstelle zum Erstellen und individuellen Anpassen von mehreren Instanzen auf der Basis einer bestimmten klassenbasierten Modulvorlage PID_LOOP durch Auswählen der gewünschten Optionen bereit. Das Schnittstellenfenster **350** in dieser Ausführungsform enthält einen Modul-Explorerbereich **352**, einen Detailanzeigebereich **354** und eine Symbolleiste **356**. Der Modul-Explorerbereich **352** kann verschiedene Elemente des Prozessteuerungssystems **10** in Form eines hierarchischen Baumes **360** auflisten, der einen Blibliothekenzweig **362** umfasst, der wiederum einen Modulvorlagenzweig **364** enthält. Im Allgemeinen können die Elemente im Modulvorlagenbereich **352** in beliebiger gewünschter Weise aufgelistet und dargestellt werden. Weiter Bezug nehmend auf den hierarchischen Baum **360** kann

ein Systemkonfigurationszweig **366** Elemente in Bezug auf eine bestimmte Steuerungsstrategie auflisten, die eine Gruppe von Elementen für einen bestimmten Prozessbereich „A" unter dem Zweig **368** enthält.

[0057] Um zum Beispiel eine oder mehrere Modulinstanzen aus einer Modulvorlage PID_LOOP für kontinuierliche Steuerung zu erstellen, kann der Benutzer die Modulklasse PID_LOOP zum gewünschten Prozessbereich oder zur gewünschten Prozessregion ziehen, wie beispielsweise dem Bereich, der dem Zweig **368** entspricht. Die Prozessregion kann ein Bereich, eine Einheit, eine Zelle oder eine beliebige andere Entität sein, für welche die Konfigurationsumgebung **150** eine Modulzuweisung ermöglicht. Die Konfigurationsumgebung **150** kann zu diesem Zeitpunkt automatisch eine Modulentwicklungsschnittstelle **380** zum Definieren und Benennen eines oder mehrerer Module, die aus der Modulvorlage PID_LOOP erstellt werden sollen, und zum Auswählen optionaler Komponenten für das eine oder die mehreren Module anzeigen. Alternativ kann der Benutzer die Modulentwicklungsschnittstelle **380** unter Verwendung eines dedizierten Menüeintrags oder eines bestimmten Befehls starten. Die Modulentwicklungsschnittstelle **38** kann eine Definitionsregisterkarte **382**, eine Konfigurationsregisterkarte **384**, ein Prozessbereichsauswahlmenü **386** (z. B. eine Pull-down-Liste) und ein Modulvorlagenauswahlmenü **388** umfassen. Der Benutzer kann den Definitions- und Konfigurationsprozess durch Klicken auf die OK-Taste **390** abschließen. In einigen Ausführungsformen ermöglicht eine ADD-Taste **392** es dem Benutzer, Modulinstanzen hinzuzufügen, die auf der ausgewählten Modulvorlage, d. h. der Vorlage PID_LOOP, basieren.

[0058] In einigen Ausführungsformen kann die Konfigurationsumgebung **150** die Modulentwicklungsschnittstelle **380** nur dann anzeigen, wenn die entsprechende Modulvorlage wenigstens eine optionale Komponente enthält. Anders ausgedrückt, ist beim Erzeugen einer Instanz das Anzeigen der Systementwicklungsschnittstelle unnötig, wenn eine bestimmte Modulvorlage nur obligatorische Komponenten enthält.

[0059] Fig. 7 stellt die Modulentwicklungsschnittstelle **380** dar, wobei die Definitionsregisterkarte **382** aktiviert und demgemäß ein Definitionsbereich **400** im Vordergrund ist. Im Allgemeinen kann der Definitionsbereich **400** ein beliebiges Layout haben und beliebige Typen von Steuerungen, Selektoren und/oder Dialogen umfassen. Das Beispiel in Fig. 7 veranschaulicht jedoch ein besonders praktisches Tabellen- oder Spreadsheet-Layout des Definitionsbereichs **400**, in dem jedes einzelne Modul durch ein eindeutiges Identifizierungskennzeichen in einer Spalte **402** identifiziert ist, jede optionale Komponente der Modulvorlage PID_LOOP in einer separaten Spalte wie beispielsweise Spalte **404**, **406** oder **408** identifiziert ist, und jede Zeile **422** bis **430** die für das entsprechende Modul ausgewählten Optionen spezifiziert. Mit anderen Worten identifiziert jede Zelle im Definitionsbereich **400** eine Option für ein bestimmtes Tupel Moduloptionale Komponente, wobei eine Leerstelle den Wegfall des Elements anzeigt. Demnach zeigt die Zeile **420** an, dass für eine Modulinstanz, die durch das Identifizierungskennzeichen 100FIC605 identifizierbar ist, ein standardmäßiger (STD) Typ oder eine standardmäßige (STD) „Variante" der optionalen Komponente, die in der Spalte **404** identifiziert ist (in diesem Beispiel ein Funktionsblock, welcher dem Vorwärtsregelungs-Eingangswertparameter zugeordnet ist) ausgewählt wurde, während die Zeile **422** anzeigt, dass eine Modulinstanz mit dem Identifizierungskennzeichen 100LIC310 diese Komponente gar nicht enthält. Um ein weiteres Beispiel zu betrachten, zeigt die Information im Definitionsbereich **400** an, dass jede der sechs aufgelisteten Instanzen die optionale Komponente LP_PID enthält, während aber die Instanzen in Zeile **422**, **424** und **426** eine LP_PID-Komponente vom Typ STD enthalten, enthält die Instanz in Reihe **426** eine LP_PID-Komponente vom Typ FF, enthält die Instanz in Reihe **428** eine LP_PID-Komponente vom Typ STD mit erweitertem Alarm und enthält die Instanz in Reihe **430** eine LP_PID-Komponente vom Typ FF mit erweitertem Alarm.

[0060] Im Allgemeinen kann ein Benutzer auf jede Zelle im Definitionsbereich **400** klicken, um eine bestimmte Option auszuwählen oder zu deselektieren, einen Wert (z. B. Komponententyp) in der Zelle zu ändern usw. Der Benutzer kann auch einen Schieberegler **432** betätigen, um die gewünschte Spalte zu fokussieren. In diesem und ähnlichen Schnittstellenfenstern der Konfigurationsumgebung **150** kann die Modulidentifizierungskennzeichenspalte **52** in derselben Position bleiben, wenn der Schieberegler **432** betätigt wird. In einigen Ausführungsformen stellt jede Zelle automatisch eine Pull-down-Liste von Typen bereit, die für die jeweilige optionale Komponente verfügbar sind. Wie bereits erwähnt, können bestimmte optionale Komponenten in einer Instanz entweder anwesend oder abwesend sein, und die entsprechenden Zellen können dann eine Kontrollmarkierung oder einen anderen visuellen Indikator anzeigen, wenn der Benutzer wählt, die optionale Komponente in die Instanz aufzunehmen. Falls gewünscht, kann der Definitionsbereich **400** auch obligatorische Komponenten auflisten, um Benutzern zu helfen, alle Funktionsblöcke zu identifizieren, die der ausgewählten Vorlage zugeordnet sind. In diesem Fall werden die obligatorischen Komponenten vorzugsweise in einer grau unterlegten Form oder unter einer eigenen Registerkarte dargestellt, um Verwechslungen zu vermeiden.

[0061] Es ist außerdem zu erkennen, dass der Definitionsbereich **400** neben Funktionsblöcken auch Parameter und andere Komponenten der Modulvorlage auflisten kann (siehe Fig. 3). Außerdem kann der Definitionsbereich **400** optional Namenskürzel anzeigen, die Komponenten oder Gruppen von Komponenten zugewiesen sind, wie zuvor unter Bezugnahme auf Fig. 6A und Fig. 6B erörtert.

[0062] In Anbetracht des Vorhergesagten ist zu erkennen, dass die Konfigurationsumgebung **150** eine Auflistung (oder andere Form von Identifizierung) von optionalen Komponenten unter Verwendung von beliebigen gewünschten Layouts wie beispielsweise einer tabellarischen Form, die mehrere Instanzen und mehrere Komponenten enthält, einem hierarchischen Baum mit optionalen Komponenten, die unter instanzspezifischen Zweigen aufgelistet sind, einer eigenen Registerkarte für jede Instanz usw. bereitstellen kann. Außerdem kann die Konfigurationsumgebung **150** beliebige Typen von Selektoren für jede optionale Komponente bereitstellen, um es dem Benutzer zu ermöglichen, zu bestimmen, ob die optionale Komponente ausgeschlossen oder in die einzige durch die Modulvorlage spezifizierte Variante oder in eine oder mehrere durch die Modulvorlage spezifizierte Typen aufgenommen werden sollte. In einigen Ausführungsformen kann der Selektor einen mehrstufigen Dialog umfassen, um zuerst zu bestimmen, ob die optionale Komponente in die jeweilige Instanz aufgenommen werden. sollte, und anschließend, falls die Modulvorlage mehrere mögliche Typen von Varianten der optionalen Komponenten spezifiziert, zu bestimmen, welcher der möglichen Typen von Varianten verwendet werden sollte. Anders ausgedrückt, kann die Konfigurationsumgebung **150** einen Typselektor getrennt vom oder integriert in den Anwesenheitsselektor für eine optionale Komponente bereitstellen.

[0063] Fig. 8 veranschaulicht einen weiteren Betriebszustand der Modulentwicklungsschnittstelle **380**. Konkret ordnet die Modulentwicklungsschnittstelle **380** den Konfigurationsbereich **450** im Vordergrund an, wenn ein Benutzer die Konfigurationsregisterkarte **384** aktiviert. Das Layout des Konfigurationsbereichs **450** kann ähnlich dem Layout des Definitionsbereichs **400** sein. Im Beispiel von Fig. 8 listet eine Spalte **452** Identifizierungskennzeichen von einzelnen Modulinstanzen auf, und die Spalten **454** bis **458** listen optionale Komponenten der Modulvorlage PID_LOOP auf. Der Benutzer kann auf eine beliebige gewünschte Zelle klicken und die Auswahl einer optionalen Komponente modifizieren. Außerdem kann der Benutzer auf eine Spalte **460** klicken, um eine Parameterkonfigurationsanzeige der Modulentwicklungsschnittstelle **380** zu aktivieren. Es ist zu erkennen, dass der Konfigurationsbereich **450** es einem Benutzer ermöglicht, mehrere Module unter Benutzung einer gemeinsamen Quellenvorlage rasch zu konfigurieren. Um außerdem die Geschwindigkeit und Effizienz der Konfiguration zu verbessern, stellt die Modulentwicklungsschnittstelle **380** durch Unterstützen eines gemeinsamen Konfigurationsbereichs **450** auch eine klare Anzeige von funktionellen Unterschieden zwischen Modulen bereit. Insbesondere kann der Benutzer durch Nachschlagen in der tabellarischen Auflistung von optionalen Komponenten, wie in Fig. 8 dargestellt, funktionelle Unterschiede zwischen den Instanzen mit den Identifizierungskennzeichen 100FIC101, 100DIC222 und 100TIC450 sofort identifizieren, ohne einen separaten Schnittstellenbildschirm für jedes Modul zu aktivieren. Ferner reduziert die Modulentwicklungsschnittstelle **380** die Wahrscheinlichkeit von Bedienerfehlern, wenn zum Beispiel der Benutzer denselben Typ von optionaler Komponente in der Spalte **458** in jeder Instanz auszuwählen wünscht. Darüber hinaus kann die Modulentwicklungsschnittstelle **380** in einigen Ausführungsformen eine Funktion zum Anwenden einer gemeinsamen Auswahloption (z. B. Typ FF) auf jede Zeile des Konfigurationsbereichs **450** für eine bestimmte optionale Komponente (z. B. AI1/LP_IN) bereitstellen.

[0064] Unter Bezugnahme auf Fig. 9 kann die Modulentwicklungsschnittstelle **380** auch den Parameterkonfigurationsbereich **470** in den Vordergrund stellen, der im Allgemeinen dem Parameterkonfigurationsbereich **450** ähnelt. Der Parameterkonfigurationsbereich **470** zeigt jedoch auch den Namen des entsprechenden Funktionsblocks auf einer Leiste **472** an, um die optionale Komponente zu identifizieren, für welche die Parameter gelten, und enthält ein Aufwärtsordnersymbol **474**, das einen Schnellrücksprung auf eine vorherige Ebene der Konfigurationsanzeige ermöglicht. Wenigstens in einigen der Ausführungsformen kann der Benutzer numerische oder alphanumerische Parameter direkt in Zellen eintippen, die verschiedenen Modul/Parameter-Tupeln entsprechen. Der Parameterkonfigurationsbereich **470** kann durch Klicken auf die Modulparameterspalte **460** (siehe Fig. 8) aktiviert werden, wie zuvor angegeben. Alternativ kann der Benutzer direkt auf die grafische oder textliche Kennung des Moduls zum Beispiel im Modul-Explorerbereich **352** oder dem Detailanzeigebereich **354** (siehe Fig. 7) klicken.

[0065] In einigen Fällen kann ein Parameter wie beispielsweise der Parameter LP_IN in Spalte **478** als eine Struktur oder Bitkette definiert werden. Fig. 10 veranschaulicht noch einen weiteren Betriebszustand der Modulentwicklungsschnittstelle **380**, wobei ein Benutzer LP_IN oder einen ähnlichen zusammengesetzten Parameter modifiziert. In dieser Ansicht zeigt die Leiste **472** den Namen des Parameters an, und der Konfigurationsbereich **470** ist dementsprechend aktualisiert, um Informationen anzuzeigen, die für den bitmasken- oder strukturbasierten Parameter relevant sind. In diesem Sirene stellt die

Modulentwicklungsschnittstelle **380** eine wirksame Lösung bereit, um in Komponenten und Parameter „hineinzuzoomen".

[0066] Um auf Fig. 7 zurückzukommen, kann der Benutzer nach der Eingabe der Optionen und Modulidentifizierungskennzeichen auf die OK-Taste **390** klicken. Als Reaktion erzeugt die Konfigurationsumgebung **150** sechs Modulinstanzen, die jeweils gemäß den Optionen konfiguriert sind, die in den jeweiligen Zeilen **420** bis **430** spezifiziert sind, und ordnet die erzeugten Instanzen im Zweig **368** an der Stelle an, an die der Benutzer die Modulvorlage PID_LOOP gezogen hat. In einigen Ausführungsformen kann die Modulentwicklungsschnittstelle **380** einen zusätzlichen Parameter für jede Instanz enthalten, um den Speicherstelle der Instanz zu spezifizieren. Falls gewünscht, kann die Konfigurationsumgebung **150** den Prozess des Anwendens der konfigurierten Instanzen auf die entsprechenden physischen Elemente, wie beispielsweise Steuerungen, automatisch einleiten, oder der Benutzer kann ein separate Steuerung aktivieren, um mit dem Downloadprozess zu beginnen. Es ist auch vorgesehen, dass Instanzen, die auf einer Modulvorlage mit optionalen Komponenten basieren, bestimmte Attribute (z. B. die Vatervorlagenkennung **200**, das Ausgewählte-Optionen-Feld **202** bis **204** usw.) nicht zu speichern brauchen. Um Steuerungsspeicherkapazität zu bewahren, können diese Attribute im Speicher der Workstation **13**, in einer lokalen oder in einer Online-Datenbank usw. gespeichert werden.

[0067] Alternativ kann die Konfigurationsumgebung **150** die Konfigurationsinformationen in Bezug auf die sechs zu instanziierenden Modulinstanzen an eine andere Komponente des Pakets von Anwendungen **50** übertragen. Zum Beispiel kann das Paket von Anwendungen **50** eine Modulinstanziierungsmaschine umfassen, die Workstation-Speicherplatz für Modulinstanzen zuordnet, mit einer Datenbank zum Aktualisieren von Instanzreferenzen und anderen Informationen kommuniziert und Modulinstanzen auf die Steuerung **11** und/oder die intelligenten Feldgeräte **19** bis **22** (siehe Fig. 1) direkt oder über eine Kommunikationskomponente des Pakets von Anwendungen **50** herunterlädt. Es ist zu erkennen, dass im Allgemeinen die Funktionalität in Bezug auf die Wiedergabe von Benutzerschnittstellen-Bildschirmen, die Belegung von Verarbeitungs- und Speicherressourcen und die Anwendung von Modulinstanzen auf Steuerungen und Feldgeräte auf beliebige gewünschte Weise unter den Workstations **13** und in einigen Fällen unter anderen Rechengeräten verteilt werden kann.

[0068] Es ist zu erkennen, dass die Modulentwicklungsschnittstelle **380** zu einem späteren Zeitpunkt aufgerufen werden kann, wie beispielsweise dann, wenn sich die Anforderungen des Prozesssteuersystems **10**, um ein früher instanziiertes Modul zu bearbeiten. Außerdem kann ein Benutzer eine bestimmte Funktion aufrufen, um eine Modulvorlage auszuwählen und automatisch alle Modulinstanzen zu identifizieren, die aus der spezifizierten Modulvorlage instanziiert wurden. Auf diese Weise kann der Benutzer mehrere Modulinstanzen rasch lokalisieren und aktualisieren, selbst wenn diese Modulinstanzen infolge einer früheren Auswahl von verschiedenen Optionen in der entsprechenden Vatermodulvorlage nicht identisch sind. Zum Beispiel kann ein Bediener wählen, mehrere alte 4–20-mA-Feldgeräte, die eine bestimmte Funktion ausführen, durch Fieldbus-Vorrichtungen zu ersetzen, welche dieselbe oder eine ähnliche Funktion in einem bestimmten Prozessbereich ausführen. Um diese Änderungen in der Konfigurationsumgebung **150** entsprechend zu konfigurieren, kann der Bediener eine Funktion aufrufen, die alle Module in einem bestimmten Prozessbereich identifiziert, die von einer Modulvorlage abgeleitet wurden, die für die jeweilige Funktion entwickelt wurde (vorausgesetzt natürlich, dass die Instanzen ursprünglich unter Verwendung einer Modulvorlage mit optionalen Instanzen erstellt wurde). Unter Verwendung von Schnittstellenbildschirmen ähnlich den zuvor unter Bezugnahme auf Fig. 7 bis Fig. 10 erörterten kann der Benutzer eine oder mehrere geeignete optionale Komponenten in den mehreren Instanzen zum Beispiel durch Spezifizieren eines neuen Typs oder einer neuen Variante eines Funktionsblocks so ändern, dass sie der neuen Fieldbus-Konfiguration entsprechen. In diesem Sinne kann der Benutzer auf Module anwendbare Änderungen effizient verbreiten, ohne jedes Modul einzeln zu lokalisieren, einen geeigneten Dialog aufzurufen usw. In einigen Ausführungsformen kann das Konfigurationssystem **150** auch eine andere Benutzerschnittstelle als ein Spreadsheet zur gemeinsamen Bearbeitung von Instanzen bereitstellen.

[0069] In einem anderen Szenarium kann der Benutzer problemlos eine Komponente, die in einer Modulvorlage verfügbar ist, zu einer Instanz hinzufügen, die auf der Modulvorlage basiert, in welcher die Komponente ursprünglich nicht ausgewählt war. Die Konfigurationsumgebung **150** kann Schnittstellenbildschirme ähnlich der Modulentwicklungsschnittstelle **380** darstellen. Außerdem kann der Benutzer Hinzufügungen zu und Löschungen von mehreren Instanzen verbreiten, die eine gemeinsame Vatermodulvorlage teilen.

[0070] In einem anderen Aspekt können Modulvorlagen standardmäßige Anzeigeelemente oder Gruppen von Anzeigeelementen enthalten, die so ausgelegt sind, dass sie Auswahlen von optionalen Komponenten widerspiegeln. Demgemäß können Modulinstanzen nur jene Anzeigeelemente speichern oder nur auf jene Anzeigeelemente verweisen, die in der Instanz tatsächlich ausgewählt wurden. Wenn die

Konfigurationsumgebung **150** eine grafische Darstellung einer Modulinstanz erzeugt, kann die grafische Darstellung die ausgewählten Optionen widerspiegeln, um die Klarheit und Einheitlichkeit der Darstellung zu verbessern. Im Gegensatz dazu erzeugen wenigstens einige der heute verfügbaren Konfigurationsumgebungen manchmal überhäufte und verwirrende Anzeigen von Modulen, die von Vorlagen abgeleitet sind, durch Abbilden von Komponenten, die durch auf FALSE (Falsch) oder null gesetzte Parameter deaktiviert wurden und in einer physischen Entsprechung des Moduls tatsächlich abwesend sind.

[0071] Um die Vorteile von flexiblen Grafiken, die Modulvorlagen der Konfigurationsumgebung **150** zugeordnet sind, besser zu veranschaulichen, stellen Fig. 11 und Fig. 12 zwei Beispiele von Modulinstanzen dar, die unter Verwendung der zuvor erörterten Vorlage PID_STD erzeugt wurden. Unter gleichzeitiger Bezugnahme auf Fig. 5 und Fig. 11 entspricht ein Moduldiagramm **500** auf einem Schnittstellenbildschirm **502** einer Instanz, die auf der Modulvorlage PID_STD basiert, wobei die Verriegelungsbedingungs-Funktionsblöcke **256**, **258** und **272** ausgewählt wurden, während die Funktionsblöcke **276** und **278** zusammen mit dem Funktionsblock und Parameter im Bereich **280** weggelassen wurden. Andererseits entspricht ein Moduldiagramm **520** auf einem Schnittstellenbildschirm **522** (Fig. 12) einer anderen Instanz, die auf der Modulvorlage PID_STD basiert. Die Instanz von Fig. 12 umfasst den optionalen Verriegelungsbedingungs-Funktionsblock **256** und die Zuteilungsfunktionalität, die dem Bereich **280** zugeordnet ist. Selbstverständlich können beide Instanzen auch nichtoptionale Komponenten enthalten. Die in den Instanzen von Fig. 11 enthaltenen Blöcke (und die zugehörige Verdrahtung) unterscheiden sich, auch wenn beide aus der derselben PID_STD-Vorlage instanziiert wurden. Die Schnittstellenbildschirme **502** und **522** ordnen jedoch optionale Blöcke **256**, **272** usw. in den Positionen an, die in der Modulvorlage PID_STD spezifiziert sind. Auf diese Weise kann ein Benutzer, der mit mehreren Instanzen arbeitet, die auf der Vorlage PID_STD basieren, funktionelle Unterschiede zwischen den Instanzen unverzüglich identifizieren.

[0072] Um nur jene Anzeigeelemente zu aktivieren, die in einer bestimmten Instanz verwendet werden, kann die Konfigurationsumgebung **150** das Ausgewählte-Optionen-Feld **202** bis **204** und das Vatervorlagenfeld **200** (siehe Fig. 4) verwenden. Insbesondere kann die Konfigurationsumgebung **150** durch Folgen der Verknüpfung zur Vatermodulvorlage Anzeigeelemente abrufen oder auf Anzeigeelemente verweisen und unter Verwendung des Ausgewählte-Optionen-Feldes **202** bestimmen, welche Anzeigeelemente bei der Wiedergabe des Diagramms **500** oder **520** verwendet werden sollten. Außerdem kann die Konfigurationsumgebung **150** neben statischen Anzeigeelementen wie beispielsweise Funktionsblocksymbolen und Standardformen in ähnlicher Weise Dynamos, Animationen und andere grafische Elemente unterstützen. Zum Beispiel kann die Konfigurationsumgebung ein Dynamoskript, das den ausgewählten Optionen und den nicht optionalen Komponenten entspricht, automatisch erzeugen (oder aus einem Repository auswählen).

[0073] Außerdem können in Bezug auf die grafische Darstellung von Modulinstanzen Namenskürzel, die so spezifiziert werden, wie zuvor unter Bezugnahme auf Fig. 6A und Fig. 6B beschrieben, in gedruckten Dokumenten, im Offline-Modus und in anderen Kontexten sowohl innerhalb als auch außerhalb der Konfigurationsumgebung **150** angezeigt werden. Wie in Fig. 13 veranschaulicht, kann ein Schnittstellenbildschirm **600** zum Beispiel ein Diagramm **602** anzeigen, in dem Funktionsblöcke mit Namenskürzeln dargestellt sind, die den entsprechenden optionalen Komponenten zugewiesen sind. Es ist zu erkennen, dass, da die Funktionsblöcke **612** und **614** in diesem Beispiel einer einzigen aggregierten optionalen Komponente zugeordnet sind, umgeben von einem Rahmen **616**, die Funktionsblöcke **612** und **614** gemeinsam ein einziges Namenskürzel benutzen, das in einer Ecke des Feldes **616** angezeigt wird.

[0074] Unter allgemeiner Bezugnahme auf Fig. 3 bis Fig. 13 ist auch vorgesehen, dass eine Modulvorlage optional eine Softwaresperre enthalten kann, um Änderungen an der Vorlage zu verhindern, nachdem eine oder mehrere Instanzen unter Verwendung der Vorlage erstellt wurden.

[0075] Es ist auch zu erkennen, dass, obwohl die Verfahren zur Entwicklung und Anwendung von Modulvorlagen mit optionalen Komponenten zuvor in erster Linie in Bezug auf PID-Schleifen erörtert wurden, eine Modulvorlagenbibliothek eine große Vielfalt von Steuerroutinen umfassen und zum Beispiel Modulvorlagen für analoge Steuerung (z. B. PID-Schleifen), Überwachung (z. B. Analogeingang-Überwachung), Motor mit zwei Betriebszuständen (Steuerungen für Pumpen und Rührwerke), Zweiwegventilregelung, Simulation usw. enthalten kann.

[0076] Obwohl das vorliegende System und die vorliegenden Verfahren unter Bezugnahme auf spezifische Beispiele beschrieben wurden, die lediglich Veranschaulichungszwecken dienen und die Erfindung nicht einschränken, ist für Durchschnittsfachleute zu erkennen, dass Änderungen, Hinzufügungen und/oder Weglassungen an den offenbarten Ausführungsformen vorgenommen werden können, ohne vom Geist und Rahmen der Erfindung abzuweichen.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Nicht-Patentliteratur

- ISA 588.0 **[0009]**
- IEC 61508 **[0009]**
- IEC 61131-3 **[0009]**

Patentansprüche

1. Verfahren zur Prozesssteuerungskonfiguration in einem Computersystem zum Entwickeln einer Prozesssteuerungsstrategie einer verfahrenstechnischen Anlage, umfassend:
Bereitstellen einer Modulvorlage mit einer Mehrzahl von Komponenten auf einer Benutzeroberfläche des Computersystems, wobei die Mehrzahl von Komponenten eine Mehrzahl von wenigstens teilweise miteinander verbundenen Logikblöcken und eine Vielzahl von diesen zugeordneten Parameter zum Ausführen eines Prozesssteuerungsvorgangs in der verfahrenstechnischen Anlage umfasst;
Empfangen einer Auswahl einer oder mehrerer der Mehrzahl von Komponenten der Modulvorlage über die Benutzeroberfläche, um einen Satz einer oder mehrerer ausgewählter Komponenten und einen Satz von null oder mehreren deselektierten Komponenten zu definieren;
Erzeugen einer Modulinstanz auf der Basis der Modulvorlage auf dem Computersystem, umfassend ein Instanziieren der Modulinstanz, um den Satz einer oder mehrerer ausgewählter Komponenten aufzunehmen und den Satz von null oder mehreren deselektierten Komponenten nicht aufzunehmen; und
Zuordnen der erzeugten Modulinstanz zur Prozesssteuerungsstrategie, umfassend ein Erzeugen eines Satzes von auf einem Prozessor ausführbaren Anweisungen zur Verwendung in einer physischen Vorrichtung, die in der verfahrenstechnischen Anlage gemäß der erzeugten Modulinstanz arbeitet.

2. Verfahren nach Anspruch 1, ferner umfassend ein Anzeigen auf einer Benutzeroberfläche einer grafischen Darstellung der erzeugten Modulinstanz, umfassend ein Anzeigen von Anzeigeelementen, die dem Satz einer oder mehrerer ausgewählter Komponenten entsprechen, und ein Nichtanzeigen von Anzeigeelementen, die dem Satz von null oder mehreren deselektierten Komponenten entsprechen.

3. Verfahren nach Anspruch 1, wobei es sich bei der Mehrzahl von Komponenten um eine erste Mehrzahl von Komponenten handelt; wobei die Modulvorlage ferner eine zweite Mehrzahl von Komponenten enthält und wobei das Erzeugen der Modulinstanz ferner umfasst:
bedingungsloses Aufnehmen jeder der zweiten Mehrzahl von Komponenten in die erzeugte Modulinstanz.

4. Verfahren nach Anspruch 1, ferner umfassend:
Bereitstellen einer Vorlagenbibliothek mit einer Mehrzahl von Modulvorlagen; und wobei das Bereitstellen der Modulvorlage ein Bereitstellen der Modulvorlage als Reaktion auf eine Auswahl einer der Mehrzahl von Modulvorlagen umfasst.

5. Verfahren nach Anspruch 1, wobei das Bereitstellen der Modulvorlage ein Anzeigen eines Dialogbildschirms umfasst, der die Mehrzahl von Komponenten auflistet und eine jeweilige Auswahlsteuerung für jede der Mehrzahl von Komponenten aufweist, wobei der Dialogbildschirm vorzugsweise ein Spreadsheet mit einer Mehrzahl von Zeilen und einer Vielzahl von Spalten aufweist, die einen Satz von Zellen des Spreadsheets definieren, wobei eine der Mehrzahl von Zeilen oder der Vielzahl von Spalten eine Mehrzahl von Instanzen definiert, die auf der Basis der Modulvorlage erzeugt werden sollen, und die andere der Mehrzahl von Zeilen oder der Vielzahl von Spalten die Mehrzahl von Komponenten so spezifiziert, dass jede des Satzes von Zellen einer jeweiligen der Mehrzahl von Instanzen und einer jeweiligen der Mehrzahl von Komponenten entspricht.

6. Verfahren nach Anspruch 1, wobei die Modulvorlage eine Mehrzahl von Typen spezifiziert, welchen eine bestimmte Komponente aus dem Satz einer oder mehrerer ausgewählter Komponenten zugeordnet werden kann, und das Verfahren ferner umfasst:
Empfangen einer Typauswahl für die bestimmte Komponente, wobei die Typauswahl einen der Mehrzahl von Typen identifiziert; und wobei
das Erzeugen der Modulinstanz ferner ein Instanziieren der bestimmten Komponente gemäß der Typauswahl umfasst, wobei die Mehrzahl von Typen vorzugsweise wenigstens einem einer Mehrzahl von Protokollen, einer Mehrzahl von Alarmfähigkeiten oder einer Mehrzahl von Signaloptionen, einschließlich einer Rückkopplungsoption und einer Vorwärtskopplungsoption, entspricht.

7. Verfahren nach Anspruch 6, ferner umfassend:
Empfangen einer Modifikationsanforderung für die erzeugte Modulinstanz, wobei die Modifikationsanforderung die bestimmte Komponente und einen neuen Typ aus der Mehrzahl von Typen spezifiziert; und
Aktualisieren der erzeugten Modulinstanz gemäß der Modifikationsanforderung, umfassend ein Zuordnen der bestimmten Komponente zum neuen Typ.

8. Verfahren nach Anspruch 1, ferner umfassend:
Bereitstellen einer Benutzeroberfläche zum Definieren der Modulvorlage;
Bereitstellen einer Steuerung, um jede der Mehrzahl von Komponenten einem optionalen Attribut zuzuordnen, um eine optionale Anwesenheit jeder der Mehrzahl von Komponenten in der erzeugten Modulinstanz zu spezifizieren.

9. Verfahren nach Anspruch 1, wobei das Erzeugen der Modulinstanz auf der Basis der Modulvorlage ferner ein Zuordnen eines Ausgewählte-Optionen-Attributs zur Modulinstanz umfasst, wobei das Ausgewählte-Optionen-Attribut jede der Mehrzahl von Komponenten identifiziert und spezifiziert, ob die

identifizierte Komponente in der erzeugten Modulinstanz enthalten ist, wobei das Ausgewählte-Optionen-Attribut vorzugsweise eine Bitmaske ist.

10. Verfahren nach Anspruch 1, ferner umfassend ein Bereitstellen einer Schnittstelle zum Modifizieren der Modulinstanz, umfassend ein Beschränken von Modifikationen der Modulinstanz gemäß dem Satz einer oder mehrerer ausgewählter Komponenten.

11. Verfahren nach Anspruch 1, wobei die Auswahl eine erste Auswahl ist, der Satz einer oder mehrerer ausgewählter Komponenten ein erster Satz einer oder mehrerer ausgewählter Komponenten ist, der Satz von null oder mehreren deselektierten Komponenten ein erster Satz von null oder mehreren deselektierten Komponenten ist, die Modulinstanz eine erste Modulinstanz ist, und das Verfahren ferner umfasst:
Empfangen einer zweiten Auswahl einer oder mehrerer der Mehrzahl von Komponenten der Modulvorlage, um einen zweiten Satz einer oder mehrerer ausgewählter Komponenten und einen zweiten Satz von null oder mehreren deselektierten Komponenten zu definieren;
Erzeugen einer zweiten Modulinstanz auf der Basis der Modulvorlage; umfassend ein Instanziieren der zweiten Modulinstanz, um den zweiten Satz einer oder mehrerer ausgewählter Komponenten aufzunehmen und den zweiten Satz von null oder mehreren deselektierten Komponenten nicht aufzunehmen; und
wobei die erste Modulinstanz wenigstens eine Komponente enthält, die in der zweiten Modulinstanz nicht enthalten ist.

12. Verfahren nach Anspruch 1, ferner umfassend:
Empfangen einer mit der Modulvorlage assoziierten Anforderung zur Modulinstanzidentifizierung;
Identifizieren jeder Modulinstanz, die der Prozesssteuerungsstrategie zugeordnet ist und auf der Modulvorlage basiert; und
Anzeigen von Informationen auf einer interaktiven Benutzeroberfläche, die wenigstens eine Identität jeder identifizierten Modulinstanz beschreiben, das Verfahren vorzugsweise ferner umfassend:
Empfangen einer Modifikationsanforderung, die mit einer des Satzes einer oder mehrerer ausgewählter optionaler Komponenten assoziiert ist;
automatisches Anwenden der Modifikationsanforderung auf die eine des Satzes einer oder mehrerer optionaler Komponenten für jede identifizierte Modulinstanz.

13. Verfahren nach Anspruch 1, wobei das Erzeugen der Modulinstanz ein Ermöglichen einer Erfassung von Altdaten umfasst, die mit dem Satz einer oder mehrerer ausgewählter Komponenten assoziiert sind.

14. Verfahren nach Anspruch 1, wobei das Erzeugen der Modulinstanz ferner ein Zuordnen des Moduls zu einem Dynamoskript umfasst, wobei das Dynamoskript wenigstens dem Satz einer oder mehrerer ausgewählter Komponenten entspricht.

15. Verfahren zur Prozesssteuerungskonfiguration in einem Computersystem zum Entwickeln einer Modulvorlage zur Verwendung beim Konfigurieren einer Prozessteuerungsstrategie einer verfahrenstechnische Anlage, wobei die Modulvorlage als Daten auf einem computerlesbaren Medium gespeichert wird und das Verfahren umfasst:
Bereitstellen einer Darstellung der Modulvorlage auf einer interaktiven Benutzeroberfläche, die dem Computersystem zugeordnet ist, wobei die Modulvorlage eine Mehrzahl von Komponenten zum Ausführen eines Prozesssteuerungsvorgangs enthält;
Empfangen einer Anforderung über die interaktive Benutzerschnittstelle, eine der Mehrzahl von Komponenten einem optionalen Attribut zuzuordnen, um eine optionale Komponente zu definieren, und
Zuordnen der optionalen Komponente zur Modulvorlage;
wobei die Modulvorlage ausführbar ist, um wenigstens eine Modulinstanz als einen Satz von Anweisungen zu erzeugen, die durch eine oder mehrere physische Vorrichtungen in der verfahrenstechnischen Anlage ausführbar sind, derart dass die wenigstens eine Modulinstanz die optionale Komponente basierend auf einer Benutzereingabe bedingt aufnimmt oder ausschließt.

16. Verfahren nach Anspruch 15, wobei das Bereitstellen der Darstellung der Modulvorlage umfasst:
Bereitstellen einer Mehrzahl von strukturellen Komponenten auf einem Benutzeroberflächen-Bildschirm, wobei jede der Mehrzahl von strukturellen Komponenten einer vordefinierten Funktion zugeordnet wird;
Empfangen einer Auswahl von zweien oder mehreren der Mehrzahl von strukturellen Komponenten;
Empfangen von Komponenteninteraktionsinformationen, die eine Interaktion zwischen den zwei oder mehreren ausgewählten der Mehrzahl von strukturellen Komponenten beschreiben;
Empfangen eines Satzes von Parametern, die den zwei oder mehreren ausgewählten der Mehrzahl von strukturellen Komponenten zugeordnet sind;
Zuordnen der zwei oder mehreren ausgewählten der Mehrzahl von strukturellen Komponenten, der Komponenteninteraktionsinformationen und des Satzes von Parameter zur Modulvorlage.

17. Verfahren nach Anspruch 15, wobei das Empfangen der Anforderung ein Empfangen einer ersten Anforderung umfasst, die optionale Komponente eine erste optionale Komponente ist, und das

Verfahren ferner umfasst:
Empfangen einer zweite Anforderung, eine andere der Mehrzahl von Komponenten dem optionalen Attribut zuzuordnen, um eine zweite optionale Komponente zu definieren; und
Zuordnen der ersten optionalen Komponente und der zweiten optionalen Komponente zu einer Gruppe optionaler Komponenten, wobei die gesamte Gruppe optionaler Komponenten selektiv in die wenigstens eine Modulinstanz aufgenommen oder davon ausgeschlossen wird, das Verfahren vorzugsweise ferner umfassend:
Empfangen eines Namenskürzels;
Zuordnen des Namenskürzels zur Gruppe optionaler Komponente.

18. Verfahren nach Anspruch 15, wobei das Empfangen der Anforderung ferner ein Empfangen einer Anzeige einer Mehrzahl von der optionalen Komponente zuzuordnenden Typen umfasst, wobei genau einer der Mehrzahl von Typen der optionalen Komponente selektiv in die wenigstens eine Modulinstanz aufgenommen oder davon ausgeschlossen wird.

19. Prozesssteuerungskonfigurationssystem,
das als ein Satz von Anweisungen auf einem computerlesbaren Medium gespeichert ist und auf einem Prozessor ausgeführt werden kann, zum Konfigurieren eines Prozesssteuersystems, umfassend:
eine Benutzeroberfläche zum Entwickeln einer Modulinstanz, die sich wenigstens teilweise auf einer physischen Vorrichtung ausführen lässt, wobei die Benutzeroberfläche umfasst:
einen Schnittstellenbildschirm zum Darstellen einer Modulvorlage mit einer Mehrzahl von Komponenten, die eine oder mehrere strukturelle Komponenten umfassen, die der Steuerlogik und einem oder mehreren Parameter zugeordnet sind; wobei die Mehrzahl von Komponenten einen Satz von optionalen Komponenten umfasst; und
eine benutzerbedienbare Steuerung zum Empfangen einer Auswahl einer oder mehrerer optionaler Komponenten aus dem Satz von optionalen Komponenten, um einen Satz einer oder mehrerer ausgewählter Komponenten und einen Satz von null oder mehreren deselektierten Komponenten zu definieren;
eine Modulinstanziierungsmaschine zum Erzeugen einer Modulinstanz auf der Basis der Modulvorlage; wobei die Modulinstanziierungsmaschine den Satz einer oder mehrerer ausgewählter Komponenten instanziiert und den Satz von null oder mehreren deselektierten Komponenten nicht instanziiert.

20. System nach Anspruch 19, ferner umfassend eine Datenbank zum Speichern einer Mehrzahl von Modulvorlagen.

21. System nach Anspruch 19, wobei die eine oder die mehreren strukturellen Komponenten Funktionsblöcke sind und jeder Funktionsblock umfasst:
einen oder mehrere Eingänge, die jeweils so ausgelegt sind, dass sie einen jeweiligen Prozesssteuerungsparameter empfangen;
einen oder mehrere Ausgänge; und
eine Steuerlogik, die auf die eine oder die mehreren Eingänge reagiert, um ein Steuersignal an dem einen oder den mehreren Ausgängen zu erzeugen.

22. System nach Anspruch 19, wobei die Benutzeroberfläche eine erste Benutzeroberfläche ist und das System ferner umfasst:
eine zweite Benutzeroberfläche zum Entwickeln der Modulvorlage, umfassend:
eine erste benutzerbedienbare Steuerung zum Auswählen der Mehrzahl von Komponenten und Spezifizieren einer Interaktion dazwischen, um einen Steueralgorithmus zu definieren; und
eine zweite benutzerbedienbare Steuerung zum Spezifizieren des Satzes von optionalen Komponenten und Zuordnen eines jeweiligen optionalen Attributs zu jeder Komponente in dem Satz von optionalen Komponenten.

23. Prozesssteuerungskonfigurationssystem,
das als ein Satz von Anweisungen auf einem computerlesbaren Medium gespeichert ist und auf einem Prozessor ausgeführt werden kann, zum Entwickeln einer Modulvorlage, wobei die Modulvorlage beim Definieren einer Steuerungsstrategie in einer verfahrenstechnischen Anlage verwendet wird und das System umfasst:
einen interaktiven Bildschirm zum Anzeigen einer Darstellung der Modulvorlage; wobei die Modulvorlage als Daten auf einem computerlesbaren Medium gespeichert ist und eine Mehrzahl von strukturellen Komponenten, die jeweils einer jeweiligen logischen Operation zugeordnet sind, und eine Vielzahl von Parametern enthält;
eine erste benutzerbedienbare Steuerung zum Empfangen einer Auswahl von wenigstens einer der Mehrzahl von strukturellen Komponenten;
eine zweite benutzerbedienbare Steuerung zum Zuordnen der wenigstens einen ausgewählten der Mehrzahl von strukturellen Komponenten zu einem optionalen Attribut; und
eine Vorlagenerzeugungsmaschine zum Erzeugen der Modulvorlage gemäß der Auswahl der wenigstens einen der Mehrzahl von strukturellen Komponenten und dem optionalen Attribut; wobei
wenigstens eine Modulinstanz auf der Basis der Modulvorlage als ein Satz von Anweisungen erzeugt wird, die durch eine oder mehrere physische Vorrichtungen in der verfahrenstechnischen Anlage ausführbar sind; wobei die wenigstens eine Modulinstanz die optionale Komponente gemäß einer Benutzerauswahl aufnimmt oder ausschließt.

24. Softwaresystem nach Anspruch 23, wobei die erste benutzerbedienbare Steuerung ferner so aus-

gelegt ist, dass sie eine optionale Gruppe mit mehreren der Mehrzahl von strukturellen Komponenten auswählt; und wobei die zweite benutzerbedienbare Steuerung die optionale Gruppe dem optionalen Attribut zuordnet, so dass die gesamte optionale Gruppe gemäß der Benutzerauswahl in die wenigstens eine Modulinstanz aufgenommen oder davon ausgeschlossen wird, das Softwaresystem vorzugsweise ferner umfassend eine dritte benutzerbedienbare Steuerung zum Empfangen eines Namenskürzels und Zuordnen des Namenskürzels zur optionalen Gruppe.

25. Softwaresystem nach Anspruch 23, ferner umfassend eine dritte benutzerbedienbare Steuerung zum Zuordnen der wenigstens einen ausgewählten der Mehrzahl von strukturellen Komponenten zu einer Mehrzahl von Komponententypen, wobei die wenigstens eine Modulinstanz die wenigstens eine ausgewählte der Mehrzahl von strukturellen Komponenten gemäß der Typauswahl aufnimmt oder ausschließt.

26. Modulvorlage, die als Daten auf einem computerlesbaren Medium zum Erzeugen einer Modulinstanz auf der Basis der Modulvorlage gespeichert ist, wobei sich die Modulinstanz wenigstens teilweise auf einer Vorrichtung ausführen lässt, die in einem Prozesssteuersystem arbeitet; wobei die Modulvorlage umfasst:
Blockdaten, die eine Mehrzahl von Blöcken beschreiben, wobei jeder Block einer Prozesssteuerungsfunktion zugeordnet ist;
Verbindungsdaten, die logische Verbindungen zwischen wenigstens einigen der Mehrzahl von Blöcken beschreiben;
Parameterdaten, die eine Vielzahl von Prozessparametern beschreiben, die der Mehrzahl von Blöcken zugeordnet sind; und
Optionsdaten, welche die Kennung einer optionalen Komponente umfassen, die einem der Mehrzahl von Blöcken entspricht und die basierend auf einer Benutzerauswahl optional in der Modulinstanz anwesend ist.

27. Modulvorlage nach Anspruch 26, wobei der eine der Mehrzahl von Blöcken ein erster der Mehrzahl von Blöcken ist, und wobei die optionale Komponente ferner einem zweiten der Mehrzahl von Blöcken entspricht, wobei der der Mehrzahl von Blöcken und der zweite der Mehrzahl von Blöcken eine Gruppe optionaler Komponenten definieren, die Modulvorlage vorzugsweise ferner umfassend einen Gruppennamen, welcher der Gruppe optionaler Komponenten zugeordnet ist.

28. Modulvorlage nach Anspruch 26, wobei die Kennung der optionalen Komponente eine erste Kennung der optionalen Komponente ist und der eine der Mehrzahl von Blöcken ein erster der Mehrzahl von Blöcken ist; und wobei die Optionsdaten ferner die Kennung einer zweiten optionalen Komponente umfassen, die einem zweiten der Mehrzahl von Blöcken entspricht; wobei jede der ersten optionalen Komponente und der zweiten optionalen Komponente gemäß einer Benutzerauswahl unabhängig von der anderen der ersten optionalen Komponente und der zweiten optionalen Komponente in der Modulinstanz instanziiert ist.

29. Modulvorlage nach Anspruch 26, wobei die Optionsdaten ferner ein Typauswahlfeld umfassen, das eine Mehrzahl von Typen spezifiziert, die dem einen der Mehrzahl von Blöcken zugeordnet sind.

30. Prozesssteuersystem zum Implementieren einer Steuerungsstrategie in einer verfahrenstechnischen Anlage, umfassend:
eine Mehrzahl von Vorrichtungen zum Implementieren der Steuerungsstrategie, die eine Steuerung und eine Mehrzahl von Feldgeräten zum Durchführen jeweiliger physischer Schritte in der verfahrenstechnischen Anlage umfassen;
ein computerlesbares Medium zum Speichern einer Modulvorlage als Daten, die mit einer Mehrzahl von Logikblöcken und einer Vielzahl von Parametern assoziiert sind; wobei die Modulvorlage zum Erzeugen von Modulinstanzen als Sätze von Anweisungen verwendet wird, die auf der Mehrzahl von Vorrichtungen ausgeführt werden können; und wobei wenigstens einer der Mehrzahl von Logikblöcken einem optionalen Attribut zugeordnet ist, um eine optionale Komponente zu definieren;
ein Konfigurationssystem, das umfasst:
eine Benutzeroberfläche, die auf einem Prozessor ausgeführt werden kann, um eine Benutzerinteraktion mit der Modulvorlage zu unterstützen;
eine Modulinstanziierungsmaschine zum Erzeugen einer Modulinstanz auf der Basis der Modulvorlage;
wobei die Modulinstanziierungsmaschine die optionale Komponente instanziiert, wenn die optionale Komponente ausgewählt ist, und die optionale Komponenten nicht instanziiert, wenn die optionale Komponente nicht ausgewählt ist.

31. System nach Anspruch 30, wobei die Benutzeroberfläche einen Schnittstellenbildschirm umfasst, um einen oder mehrere der Mehrzahl von Logikblöcken auszuwählen und den einen oder die mehreren ausgewählten der Mehrzahl von Benutzerblöcken dem optionalen Attribut zuzuordnen.

32. System nach Anspruch 30, wobei die Modulinstanziierungsmaschine eine Erfassung von Altdaten ermöglicht, die mit der optionalen Komponente assoziiert sind, wenn die optionale Komponente in der erzeugten Modulinstanz ausgewählt ist.

33. Verfahren nach Anspruch 30, wobei die Modulinstanziierungsmaschine die erzeugte Modulin-

stanz einem Dynamoskript zuordnet, wobei das Dynamoskript wenigstens der optionalen Komponente entspricht, wenn die optionale Komponente ausgewählt ist.

Es folgen 14 Blatt Zeichnungen

Anhängende Zeichnungen

Fig 1

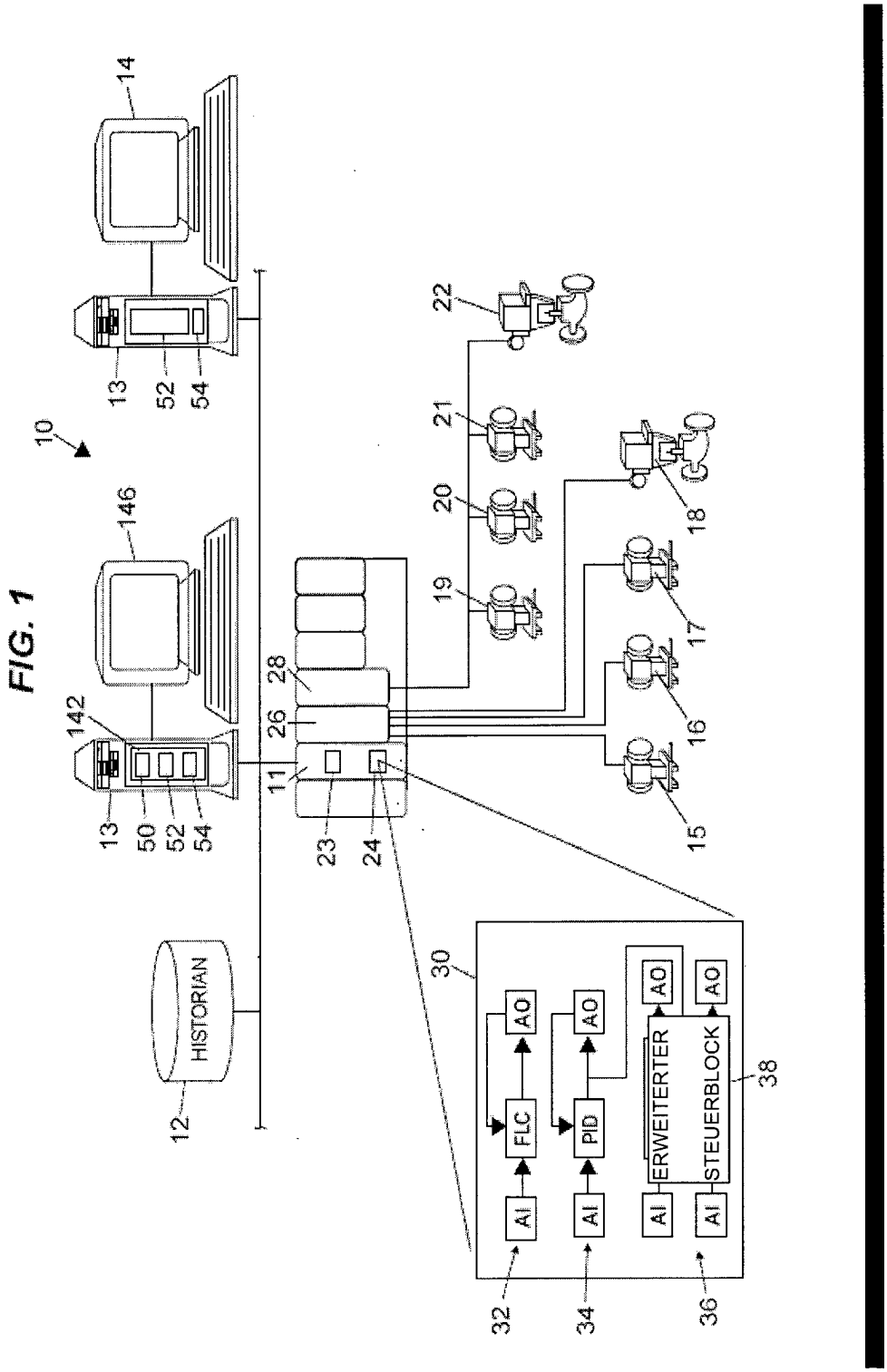
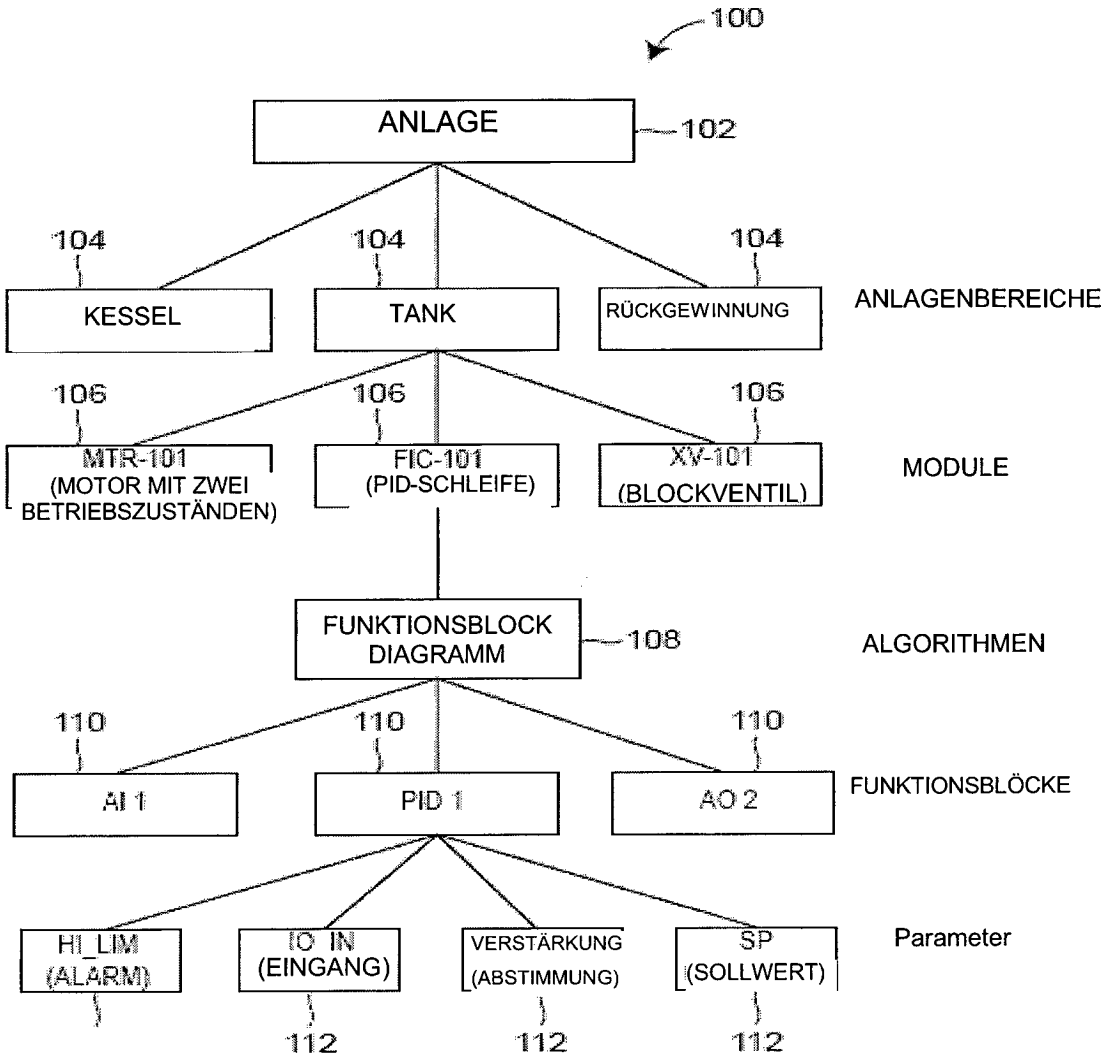
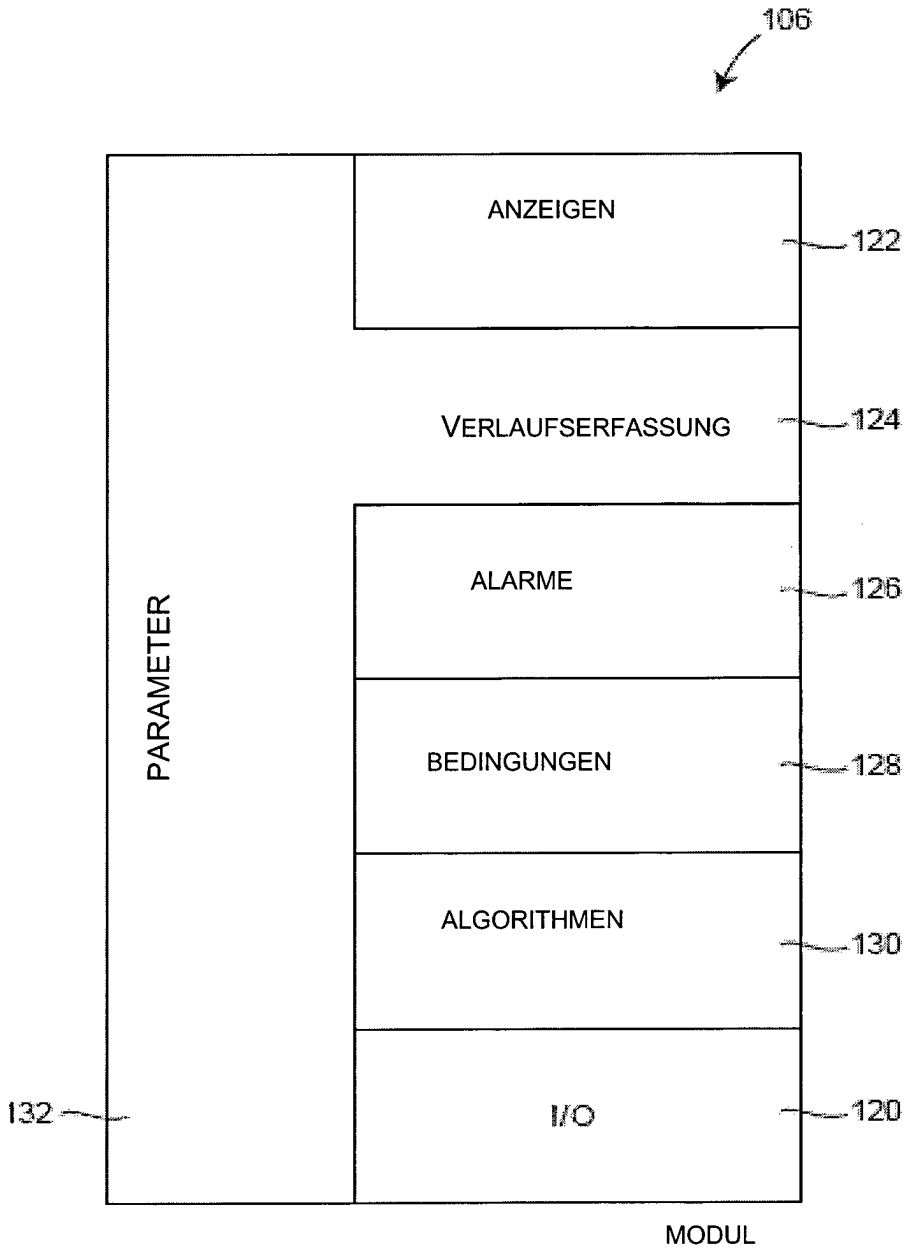


Fig 2



(STAND DER TECHNIK)

Fig 3



(STAND DER TECHNIK)

Fig 4

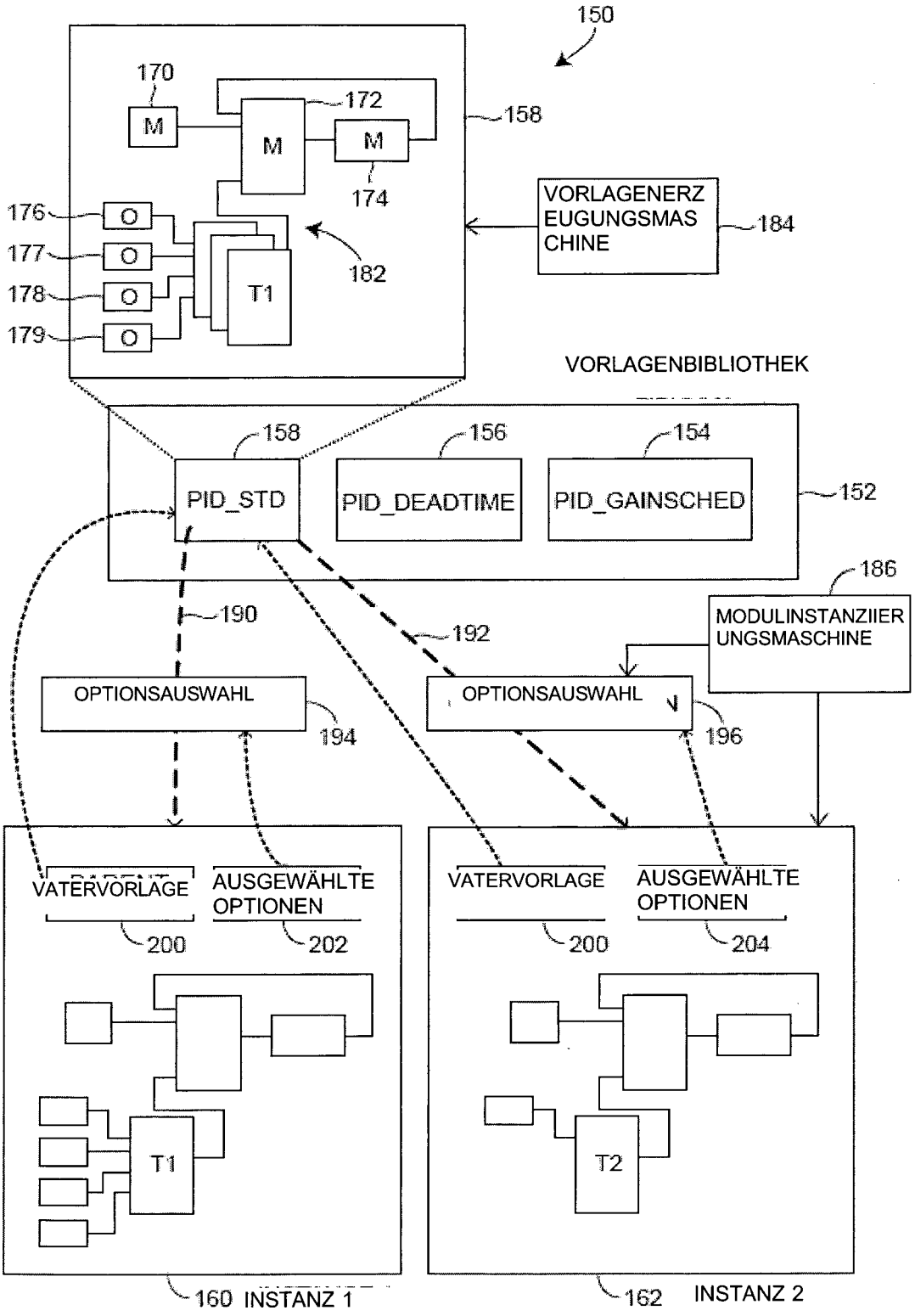

150
170
M
172
M
M
158
174
VORLAGENERZEUGUNGSMASCHINE
184
176
177
178
179
O
O
O
O
T1
182
VORLAGENBIBLIOTHEK
158
156
154
PID_STD
PID_DEADTIME
PID_GAINSCHED
152
186
MODULINSTANZIIERUNGSMASCHINE
190
192
OPTIONSAUSWAHL
194
OPTIONSAUSWAHL
196
VATERVORLAGE
AUSGEWÄHLTE OPTIONEN
200
202
VATERVORLAGE
AUSGEWÄHLTE OPTIONEN
200
204
T1
T2
160 INSTANZ 1
162 INSTANZ 2

Fig 5

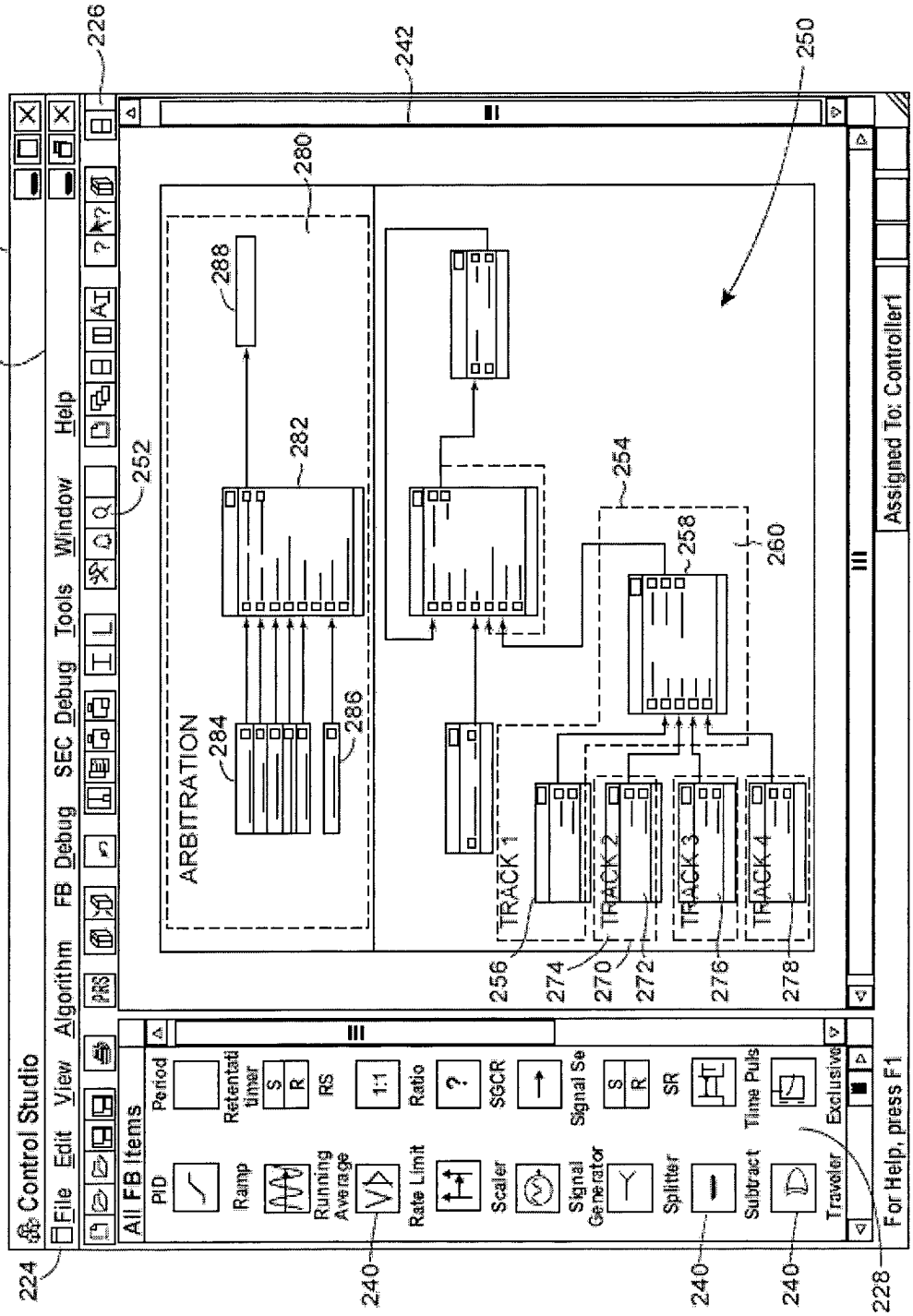

Control Studio
File Edit View Algorithm FB Debug SEC Debug Tools Window Help
All FB Items
PID
Ramp
Running Average
Rate Limit
Scaler
Signal Generator
Splitter
Subtract
Traveler
Period
Retentati timer
RS
1:1
Ratio
SGCR
Signal Se
SR
Time Puls
Exclusive
For Help, press F1
Assigned To: Controller1
ARBITRATION
TRACK 1
TRACK 2
TRACK 3
TRACK 4
220
222
224
226
228
240
242
250
252
254
256
258
260
270
272
274
276
278
280
282
284
286
288

Fig 6A

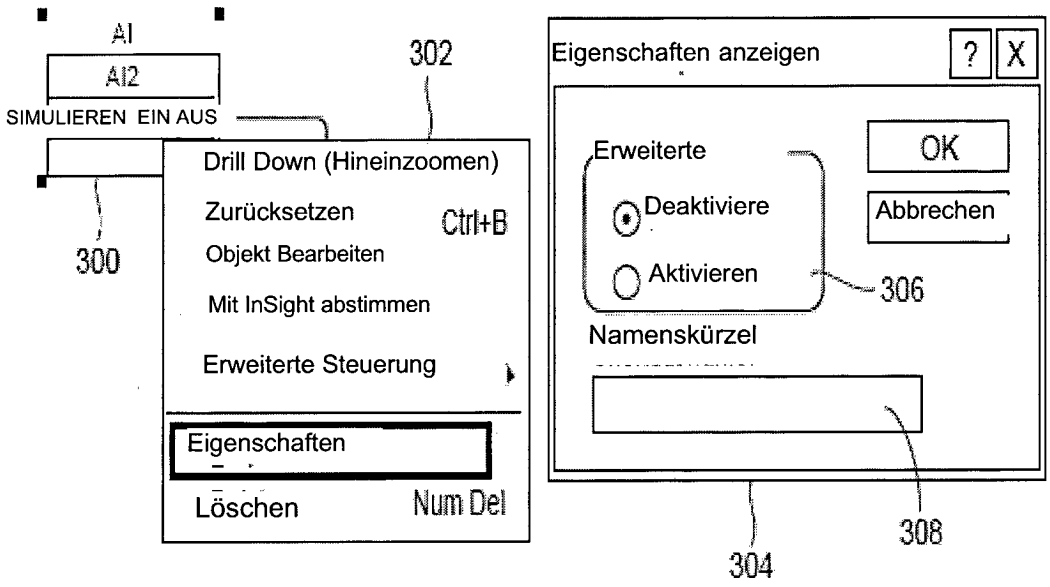
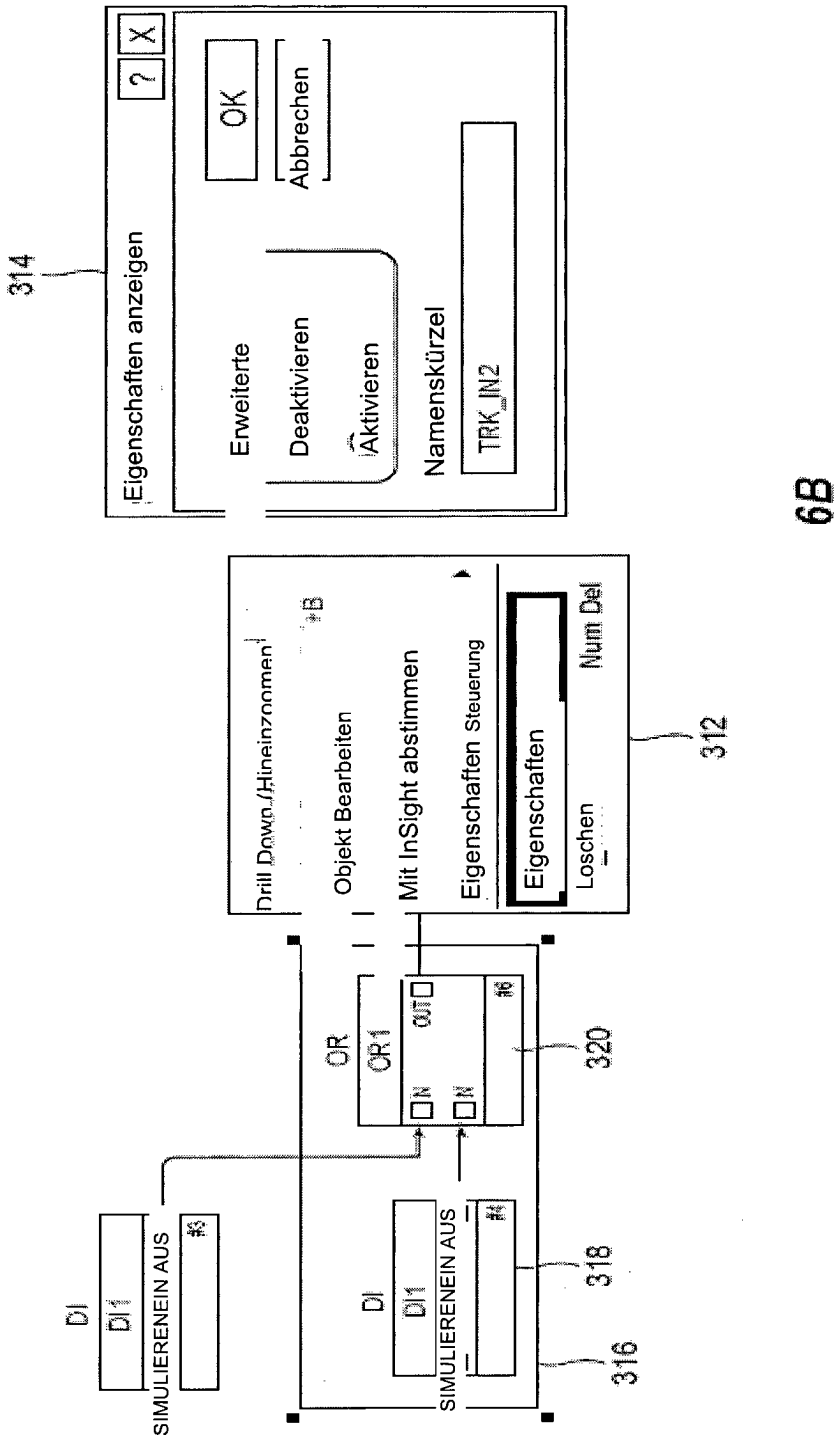

AI
AI2
SIMULIEREN EIN AUS
302
300
Drill Down (Hineinzoomen)
Zurücksetzen
Ctrl+B
Objekt Bearbeiten
Mit InSight abstimmen
Erweiterte Steuerung
Eigenschaften
Löschen
Num Del
Eigenschaften anzeigen
?
X
Erweiterte
Deaktiviere
Aktivieren
306
OK
Abbrechen
Namenskürzel
308
304

Fig 6B


314
Eigenschaften anzeigen
?
X
OK
Abbrechen
Erweiterte
Deaktivieren
Aktivieren
Namenskürzel
TRK_IN2
312
Drill Down/Hineinzoomen
Objekt Bearbeiten
Mit InSight abstimmen
Eigenschaften Steuerung
Eigenschaften
Loschen
Num Del
320
OR
OR1
IN
IN
OUT
#6
318
DI
DI1
SIMULIERENEIN AUS
#4
316
DI
DI1
SIMULIERENEIN AUS
#5

6B

Fig 7

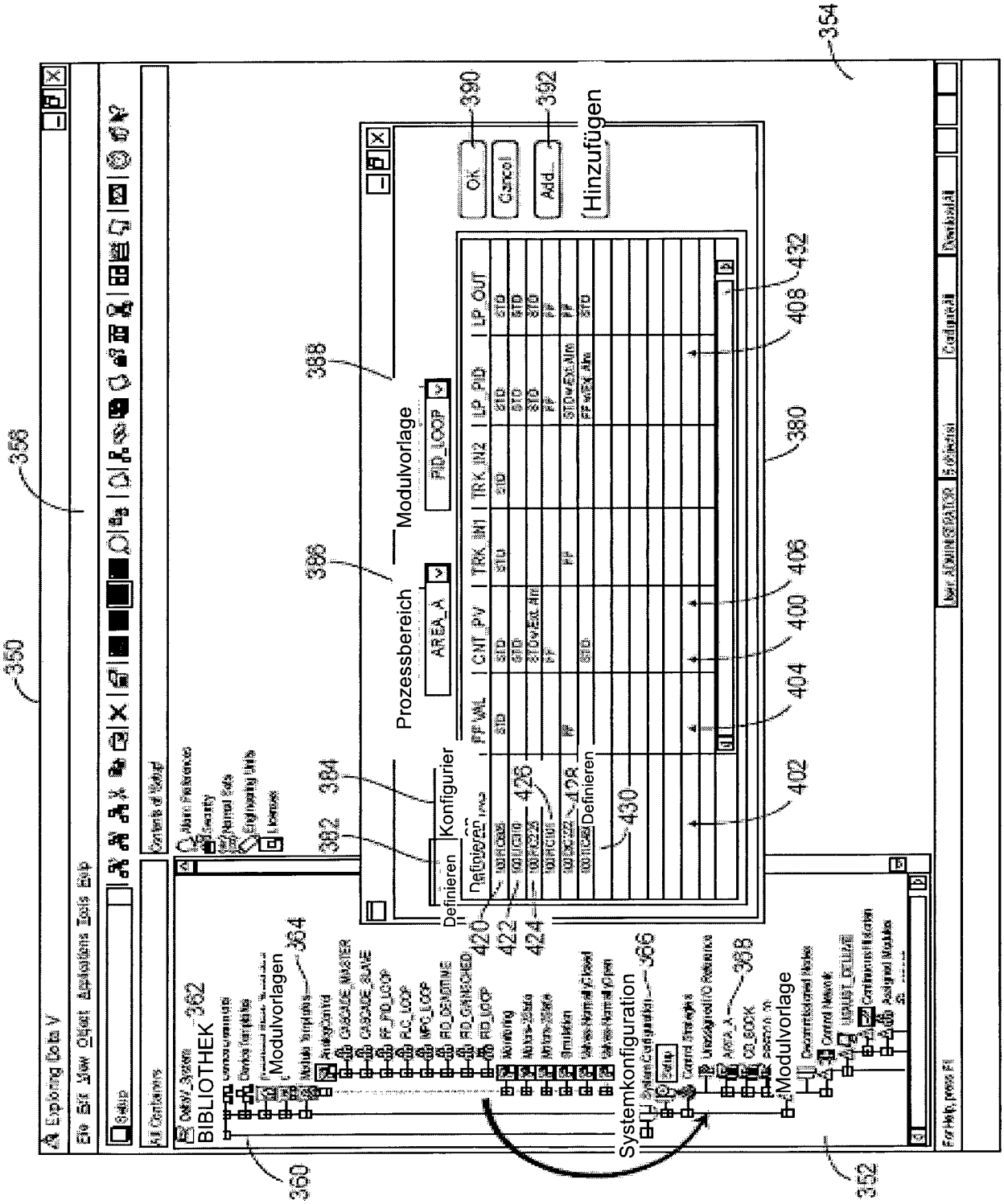

350
352
354
356
358
360
362
364
366
368
380
382
384
386
388
390
392
400
402
404
406
408
420
422
424
426
428
430
432
BIBLIOTHEK
Modulvorlagen
Systemkonfiguration
Modulvorlage
Prozessbereich
Modulvorlage
Konfigurier
Definieren
Definieren
AREA_A
PID_LOOP
Ok
Cancel
Add
Hinzufügen

Fig 8

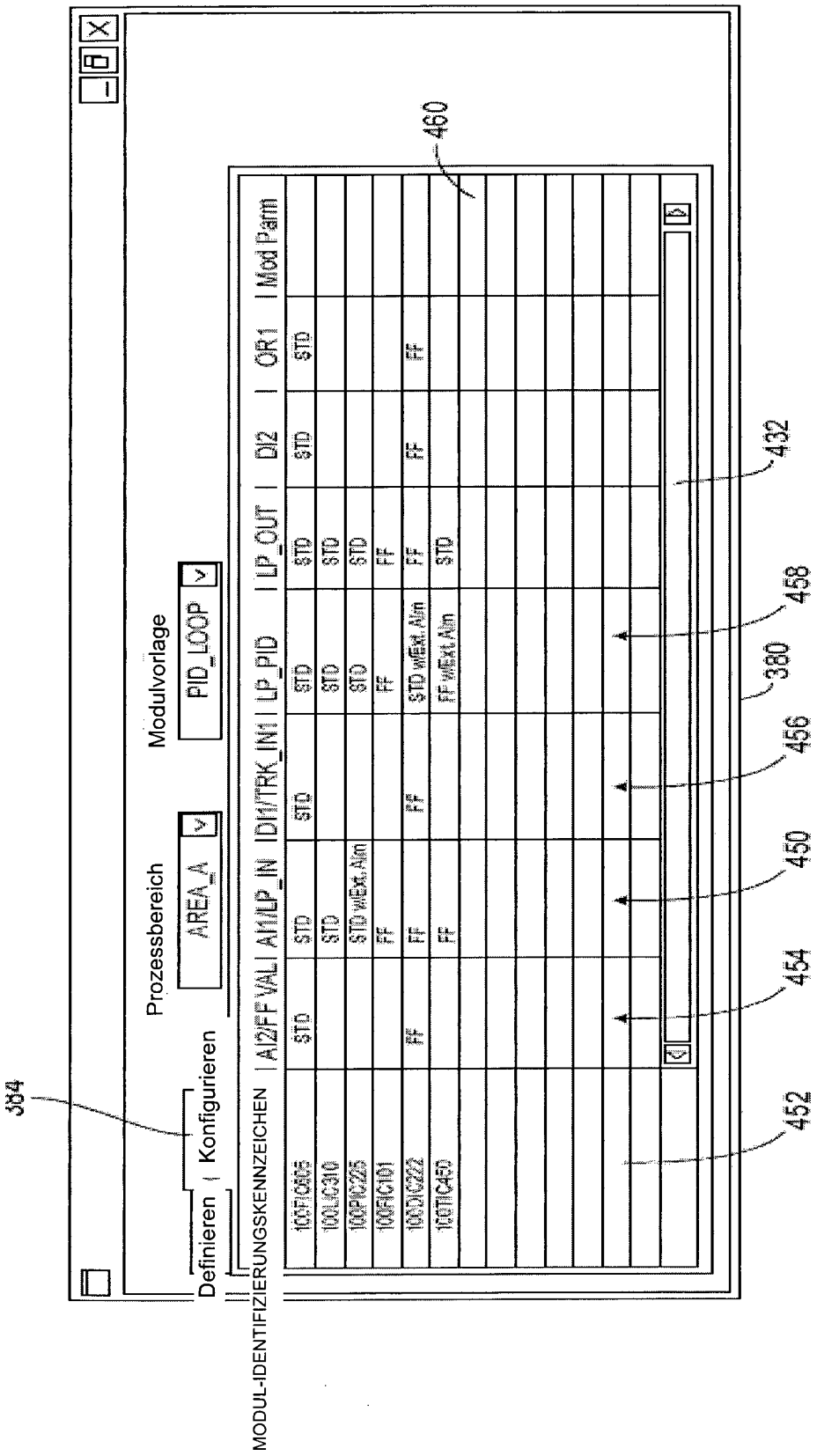

384
Definieren
Konfigurieren
MODUL-IDENTIFIZIERUNGSKENNZEICHEN
Prozessbereich
AREA_A
Modulvorlage
PID_LOOP
A1/2/FF VAL
A1/LP_IN
DI1/TRK_IN1
LP_PID
LP_OUT
DI2
OR1
Mod Parm
100FIC605
100LIC310
100PIC225
100FIC101
100DIC222
100TIC450
STD
FF
STD w/Ext. Alm
460
432
458
380
456
450
454
452

Fig 9

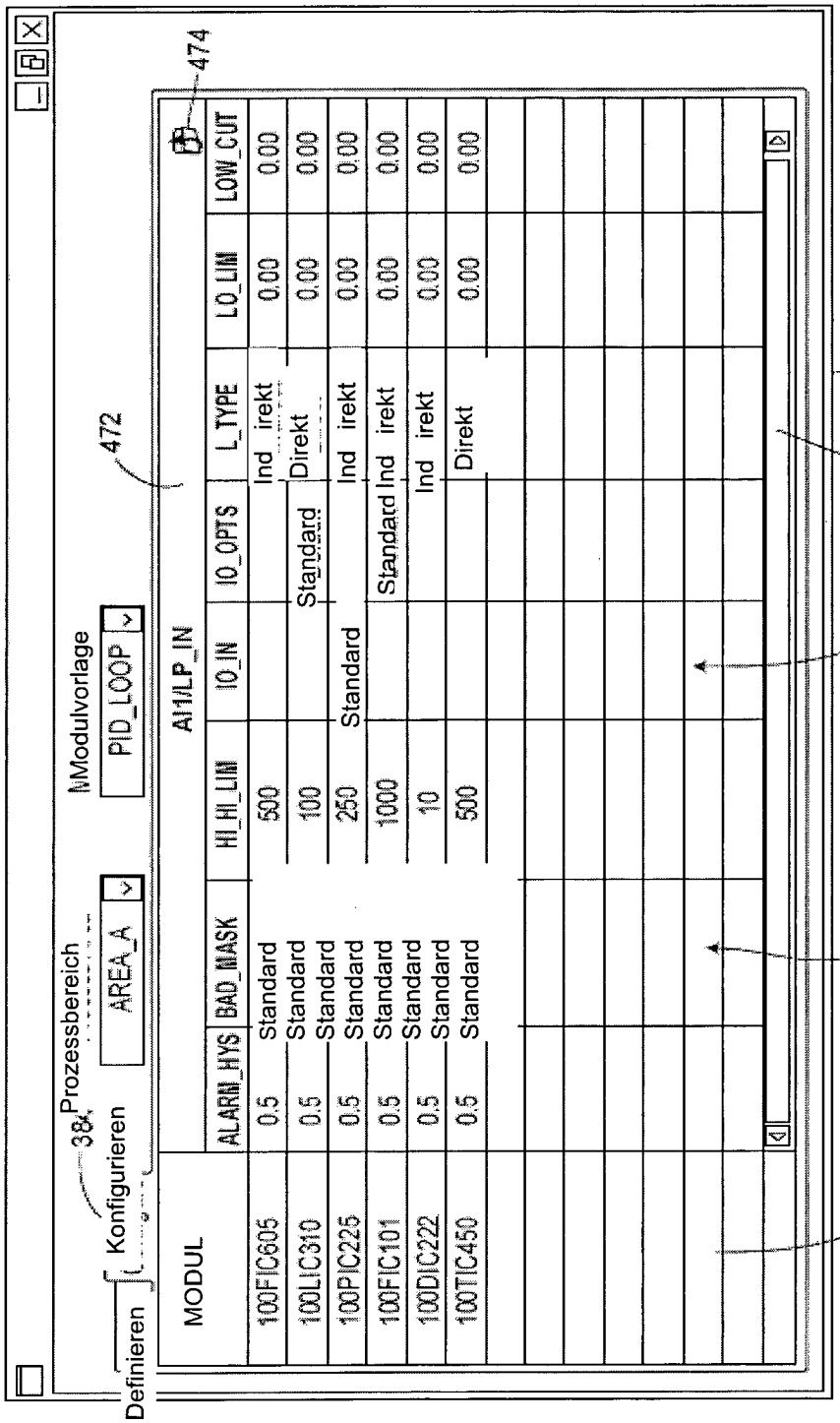

Definieren
Konfigurieren
Prozessbereich
AREA_A
Modulvorlage
PID_LOOP
AI1/LP_IN
MODUL
ALARM_HYS
BAD_MASK
HI_HI_LIM
IO_IN
IO_OPTS
L_TYPE
LO_LIM
LOW_CUT
100FIC605 0.5 Standard 500 Ind irekt 0.00 0.00
100LIC310 0.5 Standard 100 Standard Direkt 0.00 0.00
100PIC225 0.5 Standard 250 Standard Ind irekt 0.00 0.00
100FIC101 0.5 Standard 1000 Standard Ind irekt 0.00 0.00
100DIC222 0.5 Standard 10 Ind irekt 0.00 0.00
100TIC450 0.5 Standard 500 Direkt 0.00 0.00
380
472
474
432
478
470
452

Fig 10

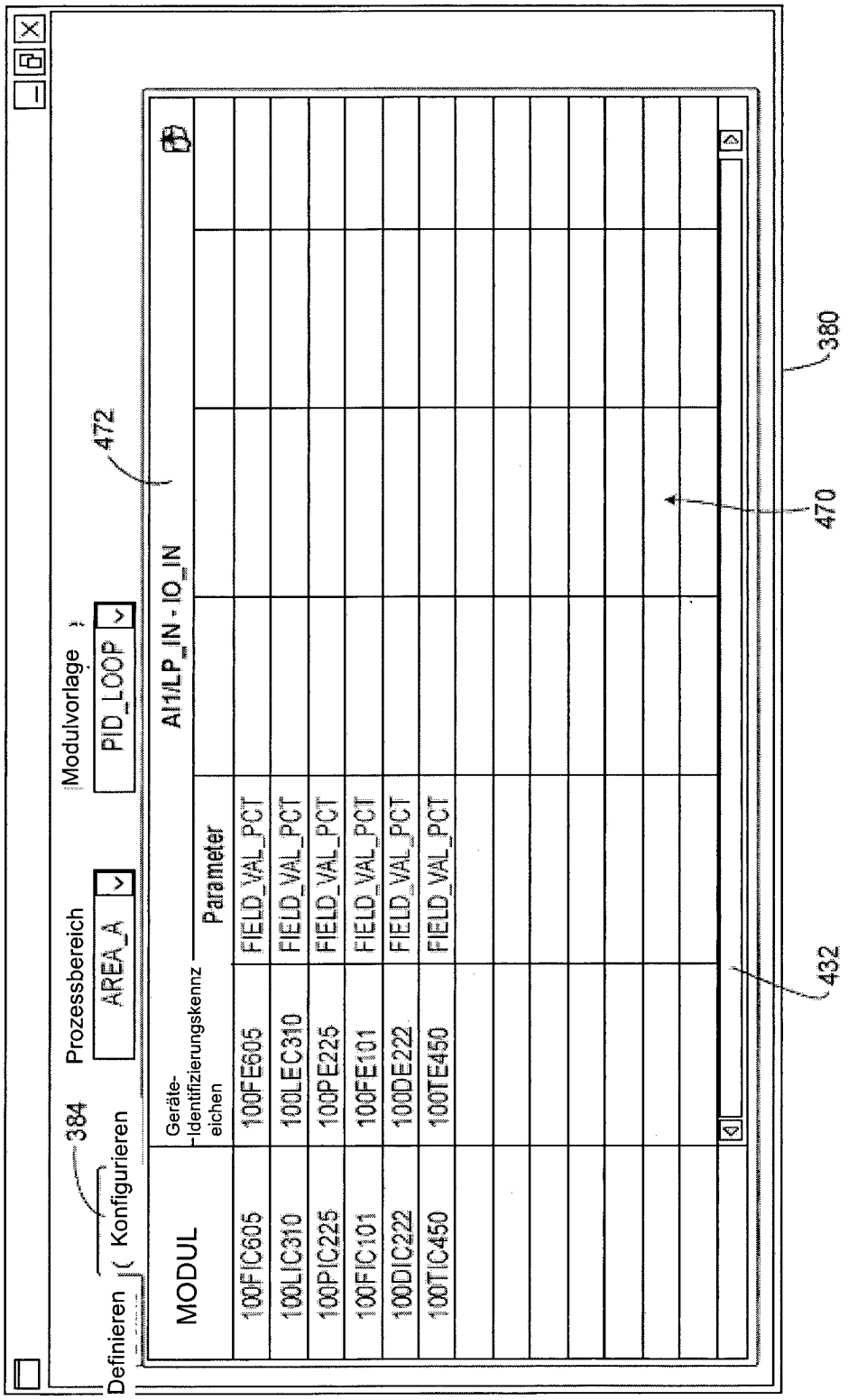

Definieren
Konfigurieren
384
Prozessbereich
AREA_A
Modulvorlage
PID_LOOP
472
AI1/LP_IN - IO_IN
MODUL
Geräte-Identifizierungskennzeichen
Parameter
100FIC605 100FE605 FIELD_VAL_PCT
100LIC310 100LEC310 FIELD_VAL_PCT
100PIC225 100PE225 FIELD_VAL_PCT
100FIC101 100FE101 FIELD_VAL_PCT
100DIC222 100DE222 FIELD_VAL_PCT
100TIC450 100TE450 FIELD_VAL_PCT
380
470
432

Fig 11

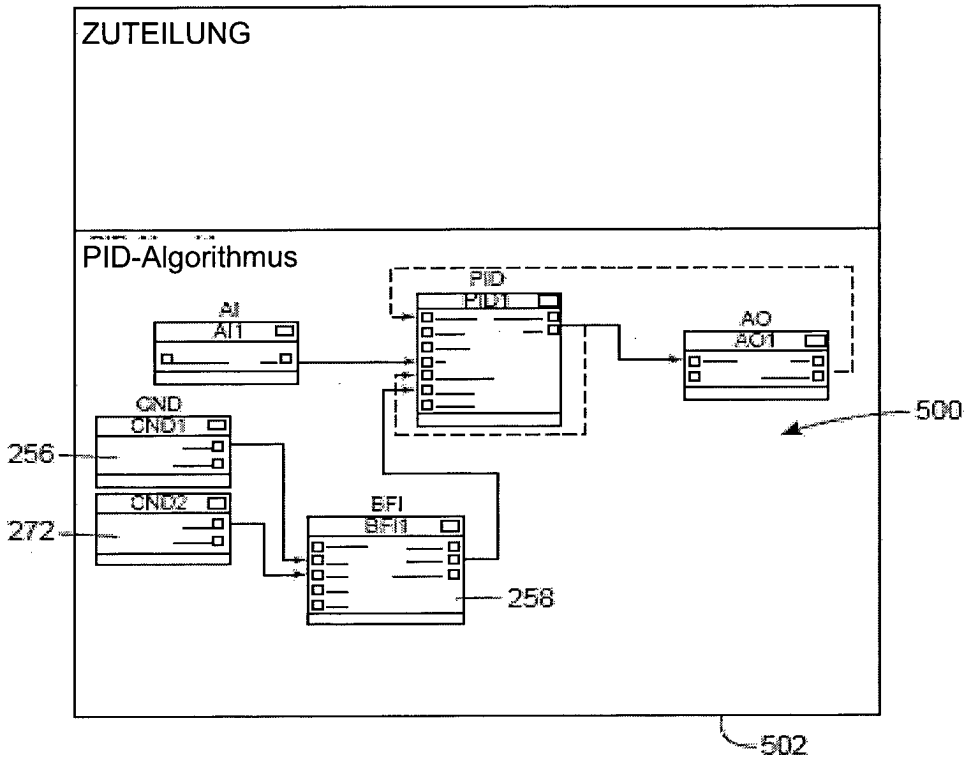

ZUTEILUNG
PID-Algorithmus
PID
PID1
AI
AI1
AO
AO1
CND
CND1
CND2
BFI
BFI1
256
272
258
500
502

Fig 12

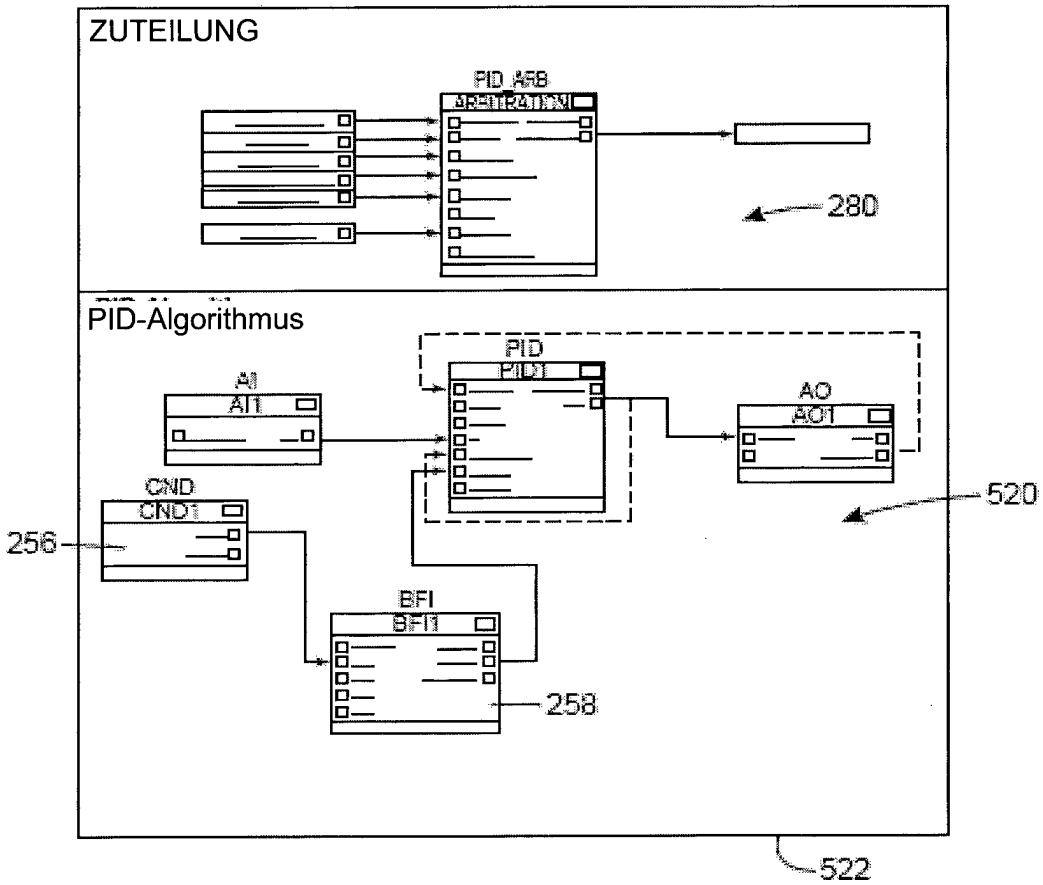

ZUTEILUNG
PID AFB
280
PID-Algorithmus
PID
PID1
AI
AI1
AO
AO1
CND
CND1
256
520
BFI
BFI1
258
522

Fig 13

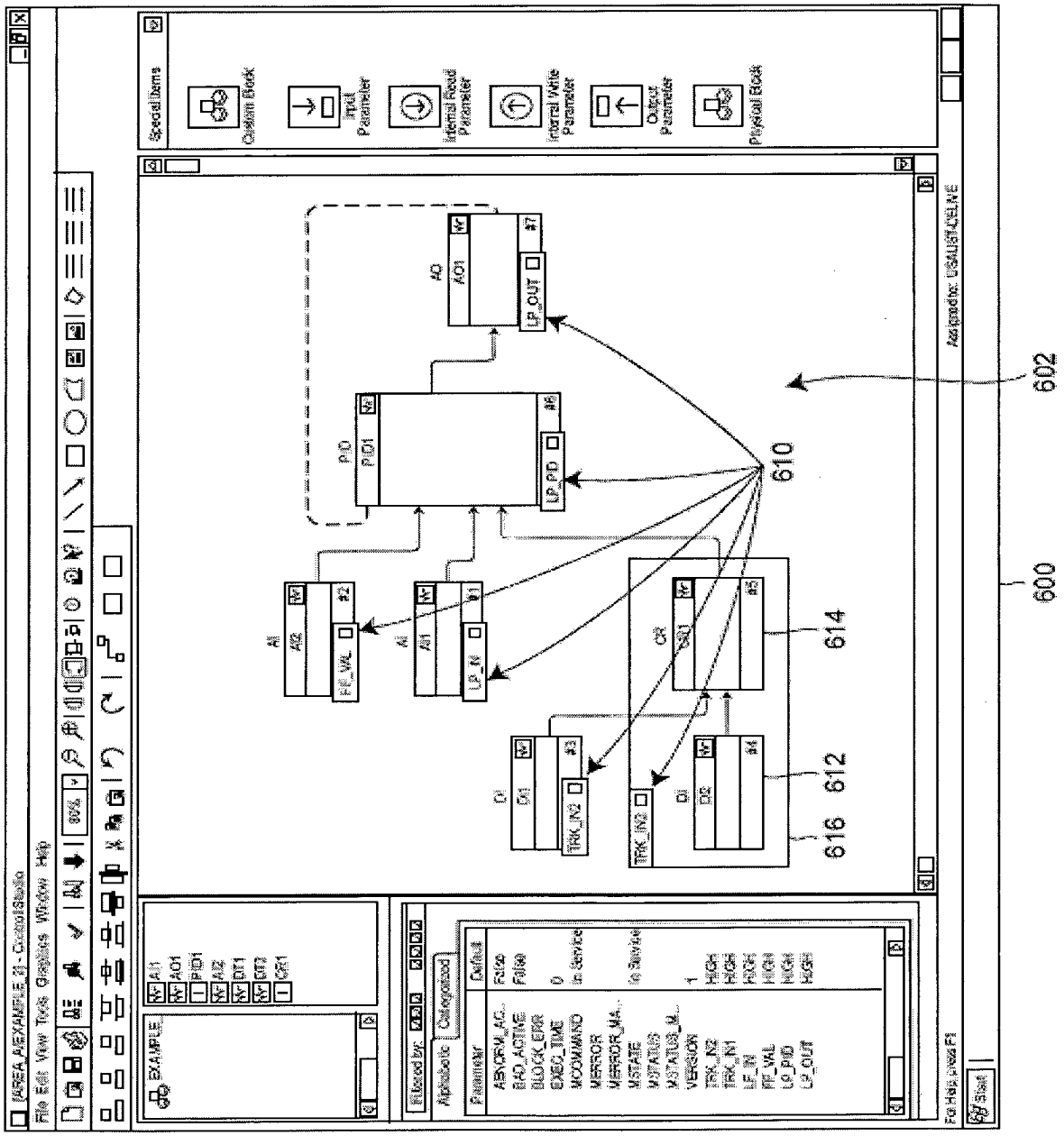


FIG. 13