



(12) **Patentschrift**

(21) Aktenzeichen: **100 51 645.9**
(22) Anmeldetag: **18.10.2000**
(43) Offenlegungstag: **09.08.2001**
(45) Veröffentlichungstag
der Patenterteilung: **17.02.2011**

(51) Int Cl.⁸: **G05B 19/042** (2006.01)

Innerhalb von drei Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
09/420,182 18.10.1999 US

(73) Patentinhaber:
Fisher-Rosemount Systems, Inc., Austin, Tex., US

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

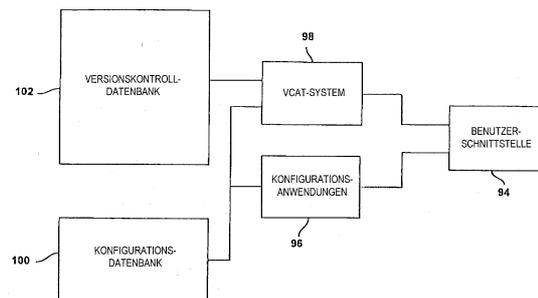
(72) Erfinder:
Hammack, Stephen Gerard, Austin, Tex., US;
Jundt, Larry Oscar, Round Rock, Tex., US; **Lucas,**
Michael J., Broughton Astley, Leicester, GB; **Dove,**
Andrew P., Austin, Tex., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:

DE	199 17 102	A1
US	59 40 294	A
US	59 03 897	A
US	57 84 577	A
US	56 19 716	A
US	55 49 200	A
EP	07 91 373	A1

(54) Bezeichnung: **Prozesssteuersystem und Verfahren zum Kontrollieren eines Prozesses**

(57) Hauptanspruch: Prozesssteuersystem, das enthält:
ein computerlesbares Medium (20);
einen Prozessor (18), der in Kommunikation mit dem computerlesbaren Medium (20) steht;
eine erste Datenbank (100), die erste Daten speichert, die eine erste Konfiguration des Prozesses mit einer Vielzahl von Prozesselementen (40, 42, 44, 46, 48, 50, 51, 52, 55) darstellen, wobei jedes Prozesselement (40, 42, 44, 46, 48, 50, 51, 52, 55) eine entsprechende erste Elementkonfiguration hat;
eine zweite Datenbank (102), die zweite Daten speichert, die eine zweite Konfiguration des Prozesses mit der Vielzahl von Prozesselementen (40, 42, 44, 46, 48, 50, 51, 52, 55) darstellen, wobei jedes Prozesselement (40, 42, 44, 46, 48, 50, 51, 52, 55) eine entsprechende zweite Elementkonfiguration hat;
eine Konfigurationsroutine (96), die in dem computerlesbaren Medium (20) gespeichert ist und so konfiguriert ist, dass sie von dem Prozessor (18) ausführbar ist, um eine Modifikation von mindestens einer der Vielzahl von ersten...



Beschreibung

[0001] Die vorliegende Erfindung betrifft allgemein Prozeßsteuersysteme und insbesondere ein Verfahren und eine Vorrichtung, die Modifikationen an dem Prozeß überwachen und aufzeichnen, um eine Versionskontrolle für den Prozeß zu gewähren.

[0002] Prozeßsteuersysteme, wie etwa die in chemischen und erdölverarbeitenden oder anderen Prozessen verwendeten, enthalten typischerweise mindestens eine zentrale Prozeßsteuereinrichtung, die mit mindestens einer Host- oder Bedienungsworkstation mit einer oder mehreren Anlageneinrichtungen über analoge und/oder digitale Busleitungen oder andere Kommunikationsleitungen oder -kanäle in Kommunikationsverbindung steht. Die Anlageneinrichtungen, bei denen es sich beispielsweise um Ventile, Ventilpositioniereinrichtungen, Schalter, Übertragungseinrichtungen (beispielsweise Temperatur-, Druck- und Durchflußmengensensoren) etc. handeln kann, führen Funktionen innerhalb des Prozesses aus, wie etwa das Öffnen oder Schließen von Ventilen und das Messen von Prozeßparametern. Die Prozeßsteuereinrichtung empfängt Signale, welche die von den Anlageneinrichtungen durchgeführten Prozeßmessungen und/oder andere Informationen, die zu den Anlageneinrichtungen gehören, wiedergeben, über eine Eingabe-/Ausgabe-Einrichtung (I/O), verwendet diese Information, um eine Steuerroutine umzusetzen und erzeugt anschließend Steuersignale, die über Busleitungen oder andere Kommunikationskanäle über die Eingabe-/Ausgabeeinrichtung an die Anlageneinrichtungen gesendet werden, um den Betriebsablauf des Prozesses zu regeln. Die Information von den Anlageneinrichtungen und der Steuereinrichtung wird typischerweise einer oder mehreren Anwendungen zur Verfügung gestellt, die von der Bedienungsworkstation ausgeführt werden, um eine Bedienungsperson in die Lage zu versetzen, jede gewünschte Funktion hinsichtlich des Prozesses auszuführen, wie etwa den gegenwärtigen Status des Prozesses zu betrachten, den Betriebsablauf des Prozesses zu modifizieren und den Prozeß zu konfigurieren.

[0003] Verschiedene Softwaretools wurden zur Unterstützung der Bedienungsperson bei der Konfiguration und anderen Modifikationen des Prozesses entwickelt. Derartige Tools bieten eine grafische Darstellung des Prozesses, die jede Funktion, die von den Anlageneinrichtungen und der Steuereinrichtung ausgeführt wird, jeweils als Element oder Objekt darstellt. Die Elemente können in einer hierarchischen Umgebung organisiert und dargestellt sein, so daß beispielsweise alle Funktionen, die von einer bestimmten Anlageneinrichtung ausgeführt werden, zusammen gruppiert oder gelistet sind. Die Elemente können auch innerhalb einer Steuervorlage dargestellt werden, so daß die Elemente in ihren Funktionsbeziehungen in dem Prozeß dargestellt sind. Beispielsweise kann eine Steuervorlage ein sequentielles Flußdiagramm bilden, das eine Reihe von miteinander verbundene Blöcken hat, welche mehrere Elemente darstellen, die durch die Verbindungslinien definierte Eingabe- und Ausgabebeziehungen haben (vgl. das Handbuch „Get started with your DeltaV™ Software“, Fisher-Rosemount, Part No. D800002X032, Juli 1998).

[0004] Jedes Element (sowie jegliche Eingabe- oder Ausgabebeziehung zwischen den Elementen) ist durch Daten definiert, die in einer Konfigurationsdatenbank gespeichert sind, die zu den Softwarekonfigurationstools gehört. Die Datenbank sowie die Softwaretools werden über ein Netzwerk zugreifbar gemacht, das typischerweise mehrere Workstations für zahlreiche Prozeßbedienungspersonen und andere Benutzer unterstützt, die jeweils Zugriff auf Daten für die Verwaltung, Prozeßkonfiguration und andere Zwecke haben. Die Modifikation der Daten in der Konfigurationsdatenbank kann jedoch zu Versionskontrollproblemen führen, wenn beispielsweise eine Bedienungsperson sich der Arbeit einer anderen Bedienungsperson nicht bewusst ist, oder wenn zu viele Modifikationen aufgetreten sind, um festzustellen, wie zu einer früheren Konfiguration oder Version des Prozesses zurückzukehren ist. Beispielsweise speichert ein früheres Prozeßsteuersystem, das die von Fisher-Rosemount Systems, Inc. verfügbare DeltaV™ Software verwendet, nur Daten, welche das Datum angeben, an dem die Modifikation durchgeführt wurde, sowie den für die Modifikation verantwortlichen Benutzer. Derartige Daten versetzen einen Benutzer nicht typischerweise in die Lage, die Konfiguration der früheren Versionen des Prozesses zu rekonstruieren (vgl. das Handbuch „Get started with your DeltaV™ Software“, Fisher-Rosemount, Part No. D800002X032, Juli 1998).

[0005] Ein weiteres Prozesssteuersystem mit Versionskontrolle und einer Speichereinrichtung zum Speichern unterschiedlicher Versionen ist aus der WO 97/29409 A1 bekannt.

[0006] Diese Versionskontrollwünsche wurden bei der Softwareentwicklung mit Konfigurationsverwaltungstools berücksichtigt, wie etwa ClearCase von Rational und Microsoft Visual SourceSafe®. Insbesondere verfolgen, kontrollieren und verwalten diese Produkte die Entwicklung einer Softwareroutine, um die nachfolgenden Fehlerbeseitigungs- und Entwicklungsarbeiten zu unterstützen. Zu diesem Zweck werden Daten, die sowohl die gegenwärtige als auch die vergangenen Versionen des Codes angeben, gespeichert. Derartige Konfigura-

tionsverwaltungstools sind gewöhnlich jedoch auf Textdarstellungen der Daten beschränkt und somit zum Speichern und Anzeigen von Informationen, die typischerweise in grafischer Art dargeboten werden, nicht gut geeignet.

[0007] Weiterführende Systeme zur Versionskontrolle von Source-Code sind aus der US 5,903,897 bekannt. Hier wird beispielsweise auch eine Check-in und Check-out Funktionalität beschrieben.

[0008] Von National Instruments wurde eine grafische Schnittstelle entwickelt, um die Programmierung im Zusammenhang mit Instrumentierungssystemen zu erleichtern. Gemäß auf der National Instruments Internet Website verfügbaren Informationen (www.ni.com) verwendet ein Produkt, das unter dem Namen „LabVIEW“ vertrieben wird, eine grafische Programmiersprache, die G-Sprache, um diese grafische Programmierschnittstelle zu unterstützen. Die Website beschreibt ferner das LabVIEW-Produkt, das ein Entwicklungstool enthält, welches es dem Programmierer erlaubt, die Unterschiede zwischen zwei Dateien mit in der G-Sprache abgefaßtem Code grafisch zu vergleichen.

[0009] Es ist Aufgabe der Erfindung, ein Prozeßsteuersystem bereitzustellen, das eine effiziente und konsistente Konfiguration eines Herstellungsprozesses ermöglicht. Des Weiteren soll ein entsprechendes Verfahren zum Kontrollieren des Prozesses angegeben werden.

[0010] Die Lösung der Aufgabe ergibt sich aus den Patentansprüchen. Unteransprüche beziehen sich auf bevorzugte Ausführungsformen der Erfindung.

[0011] Gemäß einem Aspekt der vorliegenden Erfindung enthält ein zum Kontrollieren eines Prozesses verwendbares System ein computerlesbares Medium, einen Prozessor, der mit dem computerlesbaren Medium in Kommunikation steht, und eine erste und eine zweite Datenbank. Die erste Datenbank speichert erste Daten, die eine erste Konfiguration des Prozesses darstellen, und die zweite Datenbank speichert zweite Daten, die eine zweite Konfiguration des Prozesses darstellen. Das erfindungsgemäße System enthält ferner eine Konfigurationsroutine und eine Versionskontrollroutine, die beide in dem computerlesbaren Medium gespeichert sind und so konfiguriert sind, daß sie von dem Prozessor ausführbar sind. Die Konfigurationsroutine erleichtert eine Modifikation der ersten Konfiguration des Prozesses und die Versionskontrollroutine speichert in der zweiten Datenbank dritte Daten, welche die Modifikation der ersten Konfiguration des Prozesses angeben.

[0012] Gemäß einer weiteren bevorzugten Ausführungsform enthalten die erste und die zweite Konfiguration des Prozesses jeweils eine Vielzahl von Prozeßelementen. Jedes Prozeßelement der ersten und der zweiten Vielzahl der Prozeßelemente hat eine jeweilige Elementkonfiguration, so daß die erste Konfiguration des Prozesses die Elementkonfiguration jedes Prozeßelements der ersten Vielzahl von Prozeßelementen enthält und die zweite Konfiguration des Prozesses die Elementkonfiguration jedes Prozeßelements der zweiten Vielzahl von Prozeßelementen enthält.

[0013] Gemäß einer weiteren bevorzugten Ausführungsform überwacht die Versionskontrollroutine die Modifikation der ersten Konfiguration des Prozesses, um die dritten Daten zu sammeln. Die Versionskontrollroutine kann die Modifikation überwachen, indem über die Konfigurationsroutine eine Einbuchungs-/Ausbuchungsprozedur gelegt wird. Alternativ ist die Einbuchungs-/Ausbuchungsprozedur automatisch.

[0014] Gemäß einem weiteren Aspekt der vorliegenden Erfindung enthält ein Prozeßsteuersystem, das ein Prozeßelement hat, einen Computer, der einen Prozessor hat, eine Prozeßkonfigurationsanwendung, die so ausgelegt ist, daß sie von dem Prozessor umgesetzt werden kann, um eine Version des Elements zu schaffen, und ein Versionskontrollsystem, das mit der Prozeßkonfigurationsanwendung in Kommunikation ist, um Modifikationen an der Version des Prozeßelements aufzuzeichnen und zu kontrollieren.

[0015] Gemäß einem weiteren Aspekt der vorliegenden Erfindung enthält ein Verfahren zur Steuerung eines Prozesses, der ein Prozeßelement hat, die Schritte des Schaffens einer Konfiguration des Prozesses, die eine erste Version des Prozeßelements hat, des Kontrollierens einer Modifikation der Konfiguration des Prozesses, um eine zweite Version des Prozeßelements zu schaffen, und des Aufzeichnens von Informationen, die zu der Modifikation der Konfiguration des Prozesses gehören.

[0016] Nachfolgend werden Ausführungsformen der Erfindung unter Bezug auf die Zeichnung näher beschrieben.

[0017] [Fig. 1](#) ist ein Blockdiagramm eines Prozeßsteuersystems, das eine Vielzahl von Workstations und eine

Steuereinrichtung zum Leiten und Konfigurieren eines Prozesses unter Verwendung eines Prozeßkonfigurationssystem, das in der Vielzahl von Workstations implementiert ist, aufweist;

[0018] [Fig. 2](#) ist ein Beispiel einer Bildschirmdarstellung, die von dem Prozeßkonfigurationssystem erzeugt wird, um eine Benutzerschnittstelle für dieses über eine der Workstations von [Fig. 1](#) zu schaffen;

[0019] [Fig. 3](#) ist ein weiteres Beispiel einer Bildschirmdarstellung, die von dem Prozeßkonfigurationssystem erzeugt wird, um eine weitere Benutzerschnittstelle für dieses über eine der Workstations in [Fig. 1](#) zu schaffen;

[0020] [Fig. 4](#) ist ein Blockdiagramm des Prozeßkonfigurationssystem, in das ein Versionskontroll- und Protokollierungssystem gemäß einer Ausführungsform der vorliegenden Erfindung integriert sind; und

[0021] [Fig. 5](#) bis [Fig. 19](#) sind Bildschirmdarstellungen, die von dem Versionskontroll- und Protokollierungssystem von [Fig. 4](#) erzeugt werden, um eine Benutzerschnittstelle für die Eingabe und Ausgabe von Informationen zu schaffen, die sich auf die Konfiguration von mehreren Versionen des Prozesses in Übereinstimmung mit zahlreichen Ausführungsformen der vorliegenden Erfindung beziehen.

[0022] Prozesse erfordern typischerweise eine kontinuierliche Gestaltung und Entwicklung in hohem Ausmaß. Dies hat zur Folge, daß es oftmals erwünscht ist, Daten aufzuzeichnen, die alle Modifikationen der Konfiguration des Prozesses angeben. Beispielsweise wird eine derartige Konfigurationsinformation in der pharmazeutischen Industrie bei der Überwachung durch Regierungsbehörden verwendet. Mit zunehmender Komplexität des Prozesses (sowie des Prozeßsteuersystems) wird es immer mühsamer, die Konfiguration einer früheren Version des Prozesses exakt zu bestimmen.

[0023] Die vorliegende Erfindung schafft ein System und ein Verfahren zum Aufzeichnen von Daten, die Modifikationen der Konfiguration eines Prozesses zusammen mit Informationen, die sich auf die Modifikation beziehen, wie etwa die Identität eines Benutzers, der für die Modifikation verantwortlich ist, Zeit und Datum der Modifikation und die hinter der Modifikation stehenden Überlegungen angeben. Die Daten werden gemäß vorliegender Erfindung als Versionen der Prozeßkonfiguration in einer Weise gespeichert, daß das erfindungsgemäße System und Verfahren einen Benutzer in die Lage versetzen, die Unterschiede zwischen jeweils zwei Versionen zu vergleichen und den Prozeß auf eine gewünschte frühere Konfigurationsversion zurückzusetzen.

[0024] [Fig. 1](#) zeigt ein Prozeßsteuersystem **10**, das eine Prozeßsteuereinrichtung **12** enthält, die mit einer oder mehreren Hostworkstations oder Computern **14** (bei welchen es sich um jede Art von Personalcomputern, Workstations etc. handeln kann) über ein Kommunikationsnetzwerk **14**, wie zum Beispiel eine Ethernetverbindung oder dergleichen, verbunden ist. Jede der Workstations **14** enthält einen Prozessor **18**, einen Speicher **20** und einen Anzeigebildschirm **22**. Entsprechend enthält die Steuereinrichtung **12**, bei welcher es sich beispielsweise um die DeltaV™-Steuereinrichtung handeln kann, die von Fisher-Rosemont Systems, Inc. vertrieben wird, einen Prozessor **24** und einen Speicher **26** zum Speichern von Programmen, Steuerroutinen und Daten, die von dem Prozessor **24** verwendet werden, um die Steuerung eines Prozesses umzusetzen. Die Steuereinrichtung **12** ist über das Netzwerk **16** mit verschiedenen Anlageneinrichtungen in verschiedenen Einrichtungsnetzwerken verbunden, beispielsweise einschließlich eines Fieldbus-Einrichtungsnetzwerkes **30**, eines HART-Einrichtungsnetzwerkes **32** und eines Profibus-Einrichtungsnetzwerkes **34**. Selbstverständlich könnte die Steuereinrichtung **12** mit anderen Arten von Anlageneinrichtungsnetzwerken, wie zum Beispiel 4–20 mA-Einrichtungsnetzwerken oder anderen lokalen oder entfernten I/O-Einrichtungsnetzwerken zusätzlich zu den oder anstelle der in [Fig. 1](#) dargestellten Netzwerke verbunden sein. Die Steuereinrichtung **12** implementiert oder überwacht eine oder mehrere Prozeßsteuerroutinen, die darin gespeichert sind oder anderweitig mit ihr verbunden sind, und kommuniziert mit den Einrichtungen innerhalb der Einrichtungsnetzwerke **30**, **32** und **34** und mit den Hostworkstations **14**, um den Prozeß zu steuern und den Prozeß betreffende Informationen an einen oder mehrere Benutzer, Bedienungspersonen, Prozeßingenieure und dergleichen weiterzugeben.

[0025] In einem Beispiel enthält das Fieldbus-Einrichtungsnetzwerk **30** Fieldbus-Einrichtungen **40**, die über eine Fieldbus-Verbindung **42** mit einer Fieldbus-Master-I/O-Einrichtung **44** (allgemein als Verbindungs-Masteinrichtung bezeichnet) verbunden sind, welche wiederum über eine lokale Verbindung mit der Steuereinrichtung **12** verbunden ist. In ähnlicher Weise kann das HART-Einrichtungsnetzwerk **32** eine Anzahl von HART-Einrichtungen **46** enthalten, die über Kommunikationsleitungen mit einer HART-Master-I/O-Einrichtung **48** verbunden sind, welche mit der Steuereinrichtung **12** über einen lokalen Standardbus oder eine andere Kommunikationsleitung verbunden ist. Das Profibus-Einrichtungsnetzwerk **34** ist mit drei Profibus-Slave-Einrichtungen **50**, **51** und **52** dargestellt, die über eine Profibus-Verbindung oder einen Bus **53** mit einer Profibus-Master-I/O-Einrichtung **55** verbunden sind. Die Profibus-Master-I/O-Einrichtung **55** kann in Form einer

Profibus-PCMCIA-Karte vorliegen, die an einer Standard-I/O-Schnittstellenkarte angebracht ist.

[0026] Allgemein ausgedrückt kann das Prozeßsteuersystem **10** aus [Fig. 1](#) verwendet werden, um Stapelprozesse bzw. chargenweise arbeitende Prozesse zu implementieren, bei welchen beispielsweise eine der Workstations **14** oder die Steuereinrichtung **12** eine Stapelausführungsroutine ausführt, welche eine Steueroutine einer höheren Ebene ist, welche den Betrieb einer oder mehrerer Anlageneinrichtungen leitet (sowie eventuell anderer Geräte), um eine Reihe von verschiedenen Schritten (allgemein als Phasen bezeichnet) auszuführen, die zur Herstellung eines Produkts, wie etwa eines Lebensmittelprodukts oder eines Arzneimittels, erforderlich sind. Um verschiedene Phasen umzusetzen, verwendet die Stapelausführungsroutine ein allgemein so bezeichnetes Rezept, das die durchzuführenden Schritte, die mit den Schritten verbundenen Mengen und Zeiten, die Reihenfolge der Schritte und dergleichen festlegt. Die Schritte für ein Rezept könnten beispielsweise das Füllen eines Reaktorgefäßes (nicht dargestellt) mit den geeigneten Materialien, das Mischen der Materialien in dem Reaktorgefäß, das Erwärmen der Materialien in dem Reaktorgefäß auf eine bestimmte Temperatur über eine bestimmte Zeitdauer, das Leeren des Reaktorgefäßes und anschließend das Reinigen des Reaktorgefäßes zur Vorbereitung für die nächste Charge umfassen. Jeder dieser Schritte kann eine jeweilige Phase des Stapelablaufs bzw. eines Chargendurchlaufes bilden, und die Stapelausführungsroutine in der Steuereinrichtung **12** führt für jede einzelne dieser Phasen einen unterschiedlichen Steueralgorithmus durch. Selbstverständlich können die bestimmten Materialien, Materialmengen, Aufheiztemperaturen und -zeiten etc. für verschiedene Rezepte unterschiedlich sein, und folglich können diese Parameter von Chargendurchlauf zu Chargendurchlauf in Abhängigkeit von dem hergestellten Produkt und dem verwendeten Rezept verändert werden.

[0027] Das Prozeßsteuersystem **10** kann auch in der Lage sein, Prozeßabläufe kontinuierlicher Natur zusätzlich zu denjenigen, die als Teil eines Stapelablaufs ausgelöst werden, umzusetzen. In seinem Gebrauch hierin bezieht sich somit der Begriff "Prozeß" auf jede Anzahl von stapel- bzw. chargenweisen und/oder kontinuierlichen Prozeßabläufen, die von dem Prozeßsteuersystem **10** ausgeführt oder umgesetzt werden. Während der Ausführung des Prozesses befindet sich das Prozeßsteuersystem **10** in einem Laufzeitmodus. Die Parameter und Steueralgorithmen, die während der Laufzeit zu verwenden sind, werden definiert, während das Prozeßsteuersystem **10** in einem Konfigurationsbetriebsmodus ist. Während des Konfigurationsmodus (der sich in bestimmten Situationen mit Zeitperioden überlappen kann, in welchen Prozeßoperationen ausgeführt werden) werden eine oder mehrere Softwareanwendungen, die auf einer oder mehreren Workstations **14** ausgeführt werden, verwendet, um einen Benutzer in die Lage zu versetzen, die Parameter, Steueralgorithmen etc. für den Prozeß festzulegen und dabei allgemein die Konfiguration des Prozesses zu gestalten.

[0028] Wie [Fig. 2](#) zeigt, enthält des Prozeßsteuersystem **10** ein Prozeßkonfigurationssystem, das auf einer oder mehreren Prozeßkonfigurationsanwendungen basiert, wie zum Beispiel Control Studio™ und Recipe Studio™, die beide von Fisher-Rosemont Systems, Inc. verfügbar sind. Die Prozeßkonfigurationsanwendungen, die als Software implementiert sein können, werden verwendet, um entweder Stapel- oder kontinuierliche Prozeßabläufe (hier allgemein als "der Prozeß" bezeichnet) oder einen Teil davon zu gestalten. [Fig. 2](#) ist eine Bildschirmdarstellung eines Hauptkontrollfensters einer beispielhaften Prozeßkonfigurationsanwendung, die einen Benutzer in die Lage versetzt, den Prozeß **10** zu konfigurieren. Die Prozeßkonfigurationsanwendung von [Fig. 2](#) wird vorzugsweise auf einer der Workstations **14** ausgeführt, welche mit einem zentralen Server (nicht dargestellt) in dem Netzwerk **16** korrespondieren kann. Alternativ kann die Anwendung in verteilter Weise ausgeführt werden, so daß mehr als eine der Workstations **14** für die Implementierung verantwortlich ist.

[0029] Allgemein ausgedrückt zeigt die Prozeßkonfigurationsanwendung eine Benutzerschnittstelle, die das in [Fig. 2](#) gezeigte Hauptkontrollfenster hat. Das Hauptkontrollfenster enthält Pulldownmenüs **60** mit Textbefehlen, eine Befehlsleiste **62** mit Piktogrammen, einen Funktionsbibliotheksrahmen **64** und einen Flußdiagrammrahmen **66**. Jeder dieser Abschnitte der Benutzerschnittstelle kann gemäß den typischen Fenstertechniken in der Größe verändert oder positioniert werden. Eine Vielzahl von Symbolen **68** sind in dem Funktionsbibliotheksrahmen **64** dargestellt, die jeweils einen jeweiligen Funktionsblock darstellen können, der in ein sequentielles Flußdiagramm zu integrieren ist, das in dem Flußdiagrammrahmen **66** erstellt wird. Das sequentielle Flußdiagramm kann beispielsweise ein Steuerprogramm definieren, das während einer Phase des Prozesses, die gestaltet wird, auszuführen ist. Um das Steuerprogramm aufzubauen, betätigt ein Benutzer eines oder mehrere der Symbole **68**, in dem das bzw. die gewünschten Symbol(e) in dem Bibliotheksrahmen **64** beispielsweise mit einer Zeigeeinrichtung (z. B. einer Mouse) ausgewählt werden und das bzw. die betätigten Symbol(e) in den Flußdiagrammrahmen **66** gezogen werden. Die Prozeßkonfigurationsanwendung erstellt dann ein Objekt in dem Flußdiagrammrahmen **66**, wie etwa einen Additionsblock **70**, das einen Funktionsblock darstellt. Zu dem Block **70** in dem Flußdiagrammrahmen **66** gehörende Daten bestimmen die Funktion, die ausgeführt wird, sowie ob der Block **70** Eingangsanschlüsse, Ausgangsanschlüsse und dergleichen hat, die entsprechend

als Teil des Blockes wie gezeigt grafisch dargestellt werden. Funktionsblöcke und andere Objekte, die zu dem sequentiellen Flußdiagramm (wie etwa ein Eingabeparameter) hinzugefügt werden, werden durch Linien verbunden, die von dem Benutzer (mit einem Zeichenwerkzeug, das beispielsweise aus der Befehlsleiste **62** ausgewählt wird) in dem Flußdiagrammrahmen **66** gezogen werden, um die Datenbeziehung zwischen den Objekten festzulegen. Informationen, die die Zeitsequenz betreffen, in der jeder Funktionsblock auszuführen ist, können auch von dem Benutzer angegeben werden und in Verbindung mit dem dargestellten Objekt festgelegt werden.

[0030] Die Daten, die jeden Funktionsblock, jede Phase etc. darstellen, werden in einer Konfigurationsdatenbank gespeichert, auf die von der Prozeßkonfigurationsanwendung zugegriffen und die von dieser entwickelt wird. Die praktische Ausübung der vorliegenden Erfindung ist nicht auf einen bestimmten Typ von Datenbank beschränkt und so kann die Konfigurationsdatenbank jede Form bzw. jeden Aufbau annehmen. Beispielsweise müssen die in der Konfigurationsdatenbank gespeicherten Daten nicht in einer lokalen oder lokalisierten Weise gespeichert werden, so daß die in der Konfigurationsdatenbank gespeicherten Daten über das Netzwerk **16** verteilt sein können. Die Konfigurationsdatenbank umfaßt jedoch vorzugsweise eine objektorientierte Datenbank, die in dem Speicher **20** einer der Workstations **14** angeordnet ist, um die Daten zu speichern, welche die Prozeßkonfiguration darstellen, was beispielsweise die Beziehungen zwischen den Funktionsblöcken einschließt, die innerhalb des sequentiellen Flußdiagramms zugewiesen werden, das in dem Flußdiagrammrahmen **66** gestaltet wird.

[0031] Nachdem unter Verwendung der vorstehend beschriebenen Funktionen das sequentielle Flußdiagramm gestaltet wurde, kann das resultierende Steuerprogramm als ein Verbundelement der Prozeßkonfiguration gespeichert werden. Verbundelemente, die von dem Control-Studio™-System entwickelt wurden, können wiederum zum Aufbau von Modulen verwendet werden. Zu diesem Zweck können Daten, die ein Modul darstellen, auch in der Konfigurationsdatenbank gespeichert werden. Sowohl Verbundelemente als auch Module können unter Verwendung eines sequentiellen Flußdiagrammes dargestellt werden.

[0032] Weitere Beispiele für Elemente, die unter Verwendung eines sequentiellen Flußdiagrammes dargestellt werden können, schließen Rezepte, Vorgänge und Einheiten-Vorgänge ein. Diese Elemente können durch einen anderen Abschnitt des Prozeßkonfigurationssystems, der Recipe Studio™-Anwendung gestaltet werden, so daß sie einen Stapelablauf (oder einen Teil davon) darstellen, der als Teil des Prozesses auszuführen ist. Die Recipe Studio™-Anwendung erzeugt eine Benutzerschnittstelle ähnlich der in [Fig. 2](#) gezeigten und kann einen Bibliotheksrahmen zur Verfügung stellen, der Symbole hat, die Module sowie Funktionsblöcke darstellen. Daten, die diese Rezepte, Vorgänge und Einheiten-Vorgänge wiedergeben, können dann ebenso in der Konfigurationsdatenbank gespeichert werden.

[0033] Zusammen definieren die Daten, die diese Funktionsblöcke, Module, Rezepte etc. darstellen, den Prozeß in seiner gegenwärtigen Umsetzung durch das Prozeßsteuersystem **10**. Zu diesem Zweck wird auf die Konfigurationsdatenbank zugegriffen, wenn Befehle in die Steuereinrichtung **12** und die Anlageneinrichtungen heruntergeladen werden.

[0034] Wie [Fig. 3](#) zeigt, können die in der Konfigurationsdatenbank gespeicherten Daten einen Benutzer über eine Konfigurationsdatenbank-Verwaltungsschnittstelle, wie zum Beispiel DeltaV® Explorer, dargestellt werden, die nachfolgend als "Explorersystem" bezeichnet wird, wobei es sich versteht, daß jede Datenbankverwaltungsschnittstelle verwendet werden kann. Das Explorersystem stellt eine Konfigurationshierarchie in einer fensterartigen Umgebung dar, die eine Reihe von Konfigurationstools zum Modifizieren der Elemente der Hierarchie hat. Im einzelnen dargestellt enthält ein Hauptfenster **80**, das von dem Explorersystem entwickelt wird, einen Hierarchierahmen **82**, in dem die Konfigurationshierarchie dargestellt wird, und einen Inhaltsrahmen **84**, in dem zusätzliche Informationen für jede Komponente der Hierarchie dargestellt werden. Der Hierarchierahmen **82** enthält eine Vielzahl von Symbolen, die verschiedene Ebenen der Hierarchie kennzeichnen, wie etwa die oberste Ebene **86**, welche die gesamte Prozeßkonfiguration darstellt, sowie zwei sekundäre Ebenen, die einen Bibliotheksordner **88** und einen Systemkonfigurationsordner **90** darstellen. Der Bibliotheksordner **88** kann die Informationen enthalten, die während der Prozeßgestaltung verwendet werden, bei der Funktionsblöcke und/oder Module innerhalb einer Phase oder eines Prozesses als Teil eines Steuerprogramms angewendet werden. Der Systemkonfigurationsordner **90** enthält die Resultate der Gestaltungsvorgänge in einem oder mehreren darin enthaltenen Ordnern, wie etwa einem Rezeptordner **92**. Jeder Ordner unterhalb der sekundären Ebene kann innerhalb des Hierarchierahmens in Übereinstimmung mit bekannten Fenstertechniken erweitert oder verkleinert werden. Alternativ können die Komponenten eines Ordners in dem Inhaltsrahmen **84** durch Auswählen des Ordnersnamens unter Verwendung einer Zeigeeinrichtung oder dergleichen dargestellt werden. Beispielsweise verwendet der Inhalt eines Steuerprogramms LOOP, das zu AREA A gehört, eine An-

zahl von Elementen oder Objekten, deren Namen in dem Inhaltsrahmen **84** angegeben sind. Durch Auswählen eines dieser Elemente kann der Benutzer Informationen über die Eigenschaften des Elements (Objekttyp, zur Entwicklung des Elements verwendete Anwendung, Datenspeicherposition etc.) sowie eine grafische Darstellung eines sequentiellen Flußdiagrammes in einer Prozeßkonfigurationsanwendung (zum Beispiel Control Studio) oder eine Textdarstellung, wenn kein sequentielles Flußdiagramm vorhanden ist, erhalten.

[0035] Es sei angemerkt, daß die von den Prozeßkonfigurationsanwendungen (z. B. Control Studio™- und Recipe Studio™) und dem in Zusammenhang mit [Fig. 2](#) bzw. [Fig. 3](#) beschriebenen Explorersystem gebotenen Tools in einem gewünschten Ausmaß zu einer einzigen Anwendung integriert werden können. Ferner ist die exakte Art und Weise, in der die Konfiguration des Prozesses erzielt wird, in der praktischen Ausführungsform der vorliegenden Erfindung nicht wesentlich. So ist zum Zweck der Vereinfachung der Erläuterung im vorliegenden Kontext der Begriff "Konfigurationsanwendungen" so zu verstehen, daß er die in Verbindung mit [Fig. 2](#) beschriebenen Prozeßkonfigurationsanwendungen und auch das in Zusammenhang mit [Fig. 3](#) beschriebene Explorersystem einschließt.

[0036] [Fig. 4](#) zeigt die Erzeugung einer Benutzerschnittstelle **94** auf der Anzeige **22** einer der Workstations **14** ([Fig. 1](#)), um einen Benutzer in die Lage zu versetzen, eine oder mehrere Konfigurationsanwendungen **96** zu implementieren. Gemäß vorliegender Erfindung wird die Benutzerschnittstelle auch in Verbindung mit einem Versionskontroll- und Protokollierungssystem **98** (nachfolgend als VCAT-System bezeichnet) generiert, welches im allgemeinen mit den Konfigurationanwendungen **96** zusammenarbeitet, um historische Informationen, die die Konfiguration des Prozesses betreffen, aufzuzeichnen und zu verwalten. Sowohl die Konfigurationanwendungen **96** als auch das VCAT-System **98** greifen auf eine Konfigurationsdatenbank **100** zu und stehen auch anderweitig mit dieser in Kommunikation, welche wie vorstehend beschrieben Daten speichert, welche die gegenwärtige Konfiguration des Prozesses darstellen. Das VCAT-System **98** steht auch in Kommunikation mit einer Versionskontrolldatenbank **102**, die gemäß vorliegender Erfindung von diesem verwaltet wird.

[0037] Die Versionskontrolldatenbank **102** enthält Konfigurationshistoriendaten, die eine beliebige Anzahl von früheren Versionen jedes Elements angeben, das in der Prozeßkonfiguration verwendet wurde oder wird. Zusammengenommen können die Historiendaten für alle Elemente verwendet werden, um frühere Konfigurationen des Prozesses zu rekonstruieren. Genauer ausgedrückt sind für jedes Element in der Konfigurationsdatenbank **100** (ebenso wie für diejenigen, die nicht mehr in der Konfigurationsdatenbank **100** sind) Daten, welche die Konfiguration dieses Elements darstellen, für eine Vielzahl von Versionen gespeichert. Beispielsweise kann ein Element bei drei Gelegenheiten seit seiner Erstellung modifiziert worden sein. Die Versionskontrolldatenbank **102** würde dann Daten enthalten, welche die Konfiguration des Elements zum Zeitpunkt der Erstellung angeben, welche als "Version 1" bezeichnet werden können, sowie Daten, die die Konfiguration des Elements nach jeder der drei Modifikationen angeben, welche "Version 2", "Version 3" und "Version 4" entsprechen würden.

[0038] Die Konfigurationshistoriendaten stellen somit alle Modifikationen dar, die an den Funktionsblöcke, Modulen, Phasen, Rezepten und anderen Aspekten der Prozeßkonfiguration vorgenommen wurden. Die Modifikationen können unter Verwendung der Konfigurationsanwendungen **96** ausgeführt werden, was jedoch nicht der Fall sein muß. In diesem Fall wird die praktische Umsetzung der vorliegenden Erfindung jedoch vorzugsweise durch Integrieren der Funktionalität des VCAT-Systems **98** in die Benutzerschnittstelle **94**, die von den Konfigurationsanwendungen **96** erzeugt wird, erreicht, wie schematisch in [Fig. 4](#) dargestellt und nachfolgend im Detail erläutert. Zu diesem Zweck können die Konfigurationsanwendungen **96** und das VCAT-System **98** zu einem einzelnen integrierten System kombiniert werden, was jedoch nicht zwingend der Fall sein muß. Der Deutlichkeit halber werden im erforderlichen Ausmaß Aufgaben, die gemäß vorliegender Erfindung ausgeführt werden, den Konfigurationsanwendungen **96** und dem VCAT-System **98** separat zugewiesen.

[0039] Das VCAT-System **98** wird vorzugsweise unter Verwendung einer oder mehrerer der Workstations **14** in einer Weise implementiert, welche die Überwachung jeder Modifikationen an der Prozeßkonfiguration erlaubt. Diese Überwachung kann durch das vorstehend beschriebene integrierte System weiter ausgebaut werden. Alternativ kann das VCAT-System **98** auf einer Workstation oder einer anderen Einrichtung ausgeführt werden, die nicht der Workstation **14** entspricht, die von der Bedienungsperson oder dem Prozeßkonstrukteur verwendet wird, die aber mit dem Prozeßsteuersystem **10** in Kommunikation steht, um Daten aufzuzeichnen, welche die Konfigurationmodifikationen wiedergeben.

[0040] Die Daten in einer der beiden Datenbanken **100**, **102** oder in beiden Datenbanken können in einem computerlesbaren Medium gespeichert sein, das physisch an einem anderen Ort innerhalb des Prozeßsteuersystems **10** angeordnet ist, beispielsweise in dem Speicher **20**, oder einem magnetischen oder optischen Spei-

chermedium, das zu einer der Workstations **14** gehört. Alternativ können eine der beiden Datenbanken **100**, **102** oder beide Datenbanken an einem entfernten Ort gespeichert sein, so daß die Workstation **14** auf die darin gespeicherten Daten über ein Netzwerk, wie zum Beispiel ein Intranet, das Internet, oder ein anderes Kommunikationsmedium zugreift. Ferner müssen die in den Datenbanken **100**, **102** gespeicherten Daten nicht auf demselben computerlesbaren Medium sein, so daß jeder Teil einer der beiden Datenbanken **100**, **102** in einer Speichereinrichtung oder einem Speichermedium gespeichert sein kann, das sich von den Einrichtungen oder Medien, in welchen die anderen Teile gespeichert sind, unterscheidet.

[0041] In [Fig. 4](#) ist das VCAT-System **98** einzeln und getrennt von der Versionskontrolldatenbank **102** dargestellt. In einer alternativen Ausführungsform bildet die Versionskontrolldatenbank **102** einen Abschnitt des VCAT-Systems **98**. In ähnlicher Weise können die Konfigurationsdatenbank **100** und die Versionskontrolldatenbank **102** separate und einzeln vorliegende Datenstrukturen bilden, was jedoch nicht zwingend der Fall sein muß. Das heißt, daß die Datenbanken **100**, **102** in demselben Speichermedium angeordnet sein können und tatsächlich Abschnitte einer gemeinsamen Datenbank bilden können, die dem Prozeßsteuersystem **10** zugewiesen ist. Entsprechend sollte der Begriff "Datenbank" in seiner Verwendung hierin nicht auf eine bestimmte Datenstruktur beschränkt sein.

[0042] In einer Ausführungsform umfaßt die Versionskontrolldatenbank **102** eine relationale Datenbank. Alternativ kann die Versionskontrolldatenbank **102** unter Verwendung einer Referenzanwendung, welche Versionskontrolltools, wie zum Beispiel Microsoft Visual SourceSafe®, zur Verfügung stellt, als Verwahrstelle der Versionskontrolldaten erzeugt werden. In einer weiteren alternativen Ausführungsform kann die Versionskontrolldatenbank **102** auf Dateien basieren.

[0043] [Fig. 5](#) ist eine Bildschirmdarstellung für die Benutzerschnittstelle **94** ähnlich der in [Fig. 3](#) gezeigten, wobei jedoch die von dem VCAT-System **98** gebotenen Funktionen mit den Konfigurationsanwendungen **96** integriert sind. In einem derartigen integrierten System wurden ein oder mehrere Tools zu den Konfigurationsanwendungen **96** zum Implementieren und Steuern des VCAT-Systems **98** im Kontext der Verwaltung des Prozeßsteuersystems **10** hinzugefügt. Wenn beispielsweise das VCAT-System **98** aktiviert ist (wie weiter unten im Detail erläutert wird) und ein Benutzer versucht, ein von den Konfigurationsanwendungen **96** verwaltetes Element zu betrachten und/oder zu modifizieren, indem es durch einen Doppelklick oder anderweitig ausgewählt wird, wird in der Benutzerschnittstelle **94** ein Dialog **110** erzeugt, der den Benutzer darauf hinweist, daß das Element aus der Konfigurationsdatenbank **100** "ausgebucht" werden muß, bevor er fortfahren kann. Wenn ein Benutzer ein neues Element zu der Prozeßkonfiguration hinzufügt, fügt ein derartiges integriertes System automatisch das neue Element sowohl zu der Konfigurationsdatenbank **100** als auch zu der Versionskontrolldatenbank **102** hinzu und bucht das Element aus, um dessen Erstellung innerhalb der Konfigurationsanwendungen **96** zu erleichtern.

[0044] Kurz gesagt legt das VCAT-System **98** vorzugsweise eine Einbuchungs-/Ausbuchungsumgebung über die Gestaltung des Prozesses. In dieser Umgebung muß ein Element ausgebucht werden (entweder von Hand oder automatisch), bevor Modifikationen vorgenommen werden können, so daß das VCAT-System **98** Informationen, die das Element betreffen, aufzeichnen kann, und dabei allgemein eine bevorstehende "Einbuchungs"-Operation vorbereiten kann. Wenn beispielsweise das Element "MCOMMAND", das in dem Inhaltsrahmen **84** gezeigt ist, eine Modifikation erfordert, würde der Benutzer das Element in dem Inhaltsrahmen **84** ausgewählt haben und die Option "JA" in dem Dialog **110** gewählt haben, woraufhin auf die Konfigurationsdatenbank **100**, welche die gegenwärtige Version des Prozesses speichert, zugegriffen wird, um die zu diesem Element gehörenden Daten abzurufen. Die abgerufenen Daten stellen Informationen dar, die auf verschiedene Weisen, beispielsweise als Text oder grafisch, dargestellt werden können. Somit kann die von den Konfigurationsanwendungen **96** (beispielsweise Control Studio™) gebotene Funktionalität erforderlich sein, um die abgerufenen Informationen zu betrachten oder zu modifizieren. In jedem Fall wird die geeignete Anwendung verwendet, um die Informationen zu betrachten und gewünschte Modifikationen durchzuführen, woraufhin der Benutzer typischerweise ein Speicherungstask ausführen würde.

[0045] Es sei angemerkt, daß die von den Konfigurationsanwendungen **96** gebotene Funktionalität, die sich nur auf das Betrachten der Konfiguration eines Elements bezieht, nicht von der Integration mit dem VCAT-System **98** betroffen sein muß. Das heißt, daß das VCAT-System **98** vorzugsweise dem Benutzer eine "Nurlese"-Kopie der Konfigurationsinformation ermöglicht, ohne daß der Einbuchungs-/Ausbuchungsvorgang erforderlich wäre oder ausgelöst würde.

[0046] Das Auslösen eines Speicherungstasks zum Speichern einer solchen modifizierten Information, die zu dem Element gehört, geht vorzugsweise dem Einbuchen des Elements voraus. Im einzelnen veranlasst das

Ausführen des Speicherungstasks zunächst die Konfigurationsanwendungen **96**, Daten, welche die modifizierte Information darstellen, die zu dem Element gehört, in der Konfigurationsdatenbank **100** zu speichern. Anschließend beginnt das VCAT-System **98** die Ausführung des Einbuchungstasks, um die Daten, welche die modifizierten Informationen darstellen, in der Versionskontrolldatenbank **102** zu speichern. Die Ausführung des Einbuchungstasks kann das Erzeugen eines weiteren Dialoges (nicht dargestellt) enthalten, der es einem Benutzer erlaubt, einen Kommentar hinsichtlich des Einbuchungsvorganges einzugeben. Der Kommentar kann beispielsweise auf den Grund gerichtet sein, der der Modifikation der Konfiguration zugrunde liegt. Daten, die diesen Kommentar darstellen, werden anschließend in der Versionskontrolldatenbank **102** gespeichert und mit den Daten, welche die modifizierte Konfiguration darstellen, in Bezug gesetzt.

[0047] Die Ausbuchungs-/Einbuchungsumgebung ist vorzugsweise auf nur diejenigen Elemente in der Konfigurationsdatenbank **100** beschränkt, die "versionsfähig" sind. Allgemein wird ein Element als versionsfähig betrachtet, wenn für dieses historische Konfigurationsinformationen in Verbindung mit dem Element selbst geführt bzw. gespeichert werden. In einer bevorzugten Ausführungsform zählen zu den versionsfähigen Elementen diejenigen Elemente, deren Konfiguration gestaltet bzw. programmiert wurde, wie z. B. ein beliebiges Modul, ein Verbundfunktionsblock und dergleichen. Obgleich andere Elemente, wie etwa Modulparameter, als nicht versionsfähig betrachtet werden, können historische Informationen im Wege der Konfigurationsinformation gespeichert werden, die für das Element, welches die Modulparameter enthält, oder ein anderes nicht versionsfähiges Element gespeichert wurde. Ein weiteres Beispiel eines nicht versionsfähigen Elements ist ein Schritt oder ein Übergang.

[0048] In einer Ausführungsform ist der Ausbuchungsvorgang auf das Ausmaß beschränkt, daß nur ein einzelner Ausbuchungsvorgang erlaubt ist. Auf diese Weise ist es nicht möglich, daß zwei oder mehr Benutzer versuchen, dasselbe Element auszubuchen. In dem Fall, daß ein Benutzer versucht, ein bereits ausgebuchtes Element auszubuchen, erzeugt das VCAT-System **98** ein Dialogfenster (nicht dargestellt), um den Benutzer über den bereits vorliegenden Ausbuchungsvorgang zu informieren, zusammen mit der Identität des Benutzers, der für den früheren Ausbuchungsvorgang verantwortlich ist. Alternativ kann das VCAT-System **98** die Funktionalität enthalten, die erforderlich ist, um gleichzeitige Modifikationen jedes Elements zu verfolgen.

[0049] In einer anderen Ausführungsform veranlasst das Auslösen des Ausbuchungsvorganges das VCAT-System, das ausgewählte Element zu analysieren, um zu bestimmen, welche Elemente in der Versionskontrolldatenbank **102** von Modifikationen an dem ausgewählten Element betroffen sein können. Andere Elemente können beispielsweise betroffen sein, wenn sie das ausgewählte Element verwenden. Die Elemente, die von Modifikationen an dem ausgewählten Element betroffen sein können, können in einem Dialogfenster (nicht dargestellt) identifiziert werden, das von dem VCAT-System **98** erzeugt wird, um dem Benutzer die Auswahl zu erlauben, welches der Elemente erforderlichenfalls auch ausgebucht werden sollte. Ein ähnliches Dialogfenster kann während des Einbuchungsvorganges erzeugt werden, um den Benutzer über ausgebuchte Elemente in Kenntnis zu setzen, auf welche das ausgewählte Element Bezug nimmt, die davon abhängig sind bzw. anderweitig mit diesem verbunden sind.

[0050] Die Option zum Auslösen von Ausbuchungs- und Einbuchungsoperationen oder jeder anderen Aufgabe, die in dem VCAT-System **98** ausgeführt wird, kann in der Benutzerschnittstelle **94** auf verschiedene Weise vorgesehen sein. In einer fensterartigen Umgebung wie der vorstehend im Zusammenhang mit den Konfigurationsanwendungen **96** beschriebenen und gezeigten, kann ein Benutzer aus einer Menüleiste **112** der höchsten Ebene eine Option "Tools" auswählen, was zur Darstellung einer Vielzahl von verfügbaren Tasks in Übereinstimmung mit der bekannten Fenstertechnik führt. Das gleiche oder ein ähnliches Menü können innerhalb der Benutzerschnittstelle durch eine Betätigung der rechten Maustaste auf einem Element in dem Hauptfenster **80** der Konfigurationsanwendung **96** erzeugt werden. In diesem Fall kann das Menü als ein "Kontextmenü" bezeichnet werden.

[0051] [Fig. 6](#) und [Fig. 7](#) zeigen Beispiele eines Drop-Down-Menüs **114** und eines Kontextmenüs **116**, die jeweils innerhalb der Benutzerschnittstelle **94** von dem VCAT-System **98** erzeugt werden können. Das Drop-Down-Menü **114** enthält eine Vielzahl von Taskelementen, die von dem Benutzer durch eine Zeigeeinrichtung oder dergleichen ausgewählt werden können. Ein Versionskontrollelement **118** ist hervorgehoben dargestellt (nach der entsprechenden Auswahl durch einen Benutzer), um ein Versionskontrolluntermenü **120** in Übereinstimmung mit bekannten Fenstertechniken zu erzeugen. Das Versionskontrolluntermenü **120** enthält wiederum eine Vielzahl von Taskelementen, die Tasks entsprechen, die von dem VCAT-System **98** ausgelöst und/oder ausgeführt werden können. Das Kontextmenü **116** enthält in ähnlicher Weise eine Vielzahl von Taskelementen, die den Tasks des VCAT-Systems **98** entsprechen, welche auf das ausgewählte Element in dem Hauptfenster **80** einwirken können. In beiden Menüs können Befehle, die aus einem beliebigen Grund

(beispielsweise fehlende Anwendbarkeit oder fehlende Autorisierung) nicht zur Ausführung zur Verfügung stehen, in einer unterschiedlichen Schriftart oder einem unterschiedlichen Schriftstil oder dergleichen als deaktiviert dargestellt werden. Beispielsweise ist in dem Untermenü **120** aus [Fig. 6](#) das ausgewählte Element vermutlich noch auszubuchen und demgemäß kann der Einbuchungstask nicht ausgelöst werden. Jeder der Tasks, die über die Menüs **116** und **120** zur Verfügung gestellt werden, werden nachfolgend im Zusammenhang mit dem Betriebsablauf des VCAT-Systems **98** beschrieben.

[0052] Wie [Fig. 8](#) zeigt, wird dann, wenn der Benutzer das Element "Optionen" in dem Kontextmenü **116** ([Fig. 7](#)) oder alternativ ein Element "Einstellungen", das in einem anderen Menü (nicht dargestellt) angeboten wird, auswählt, ein Versionskontrolleinstellungsdialog **122** von dem VCAT-System **98** zur Anzeige auf der Benutzerschnittstelle **94** erzeugt. Der Versionskontrolleinstellungsdialog **122** ermöglicht es dem Benutzer, Einstellungen des Benutzers für die von dem VCAT-System **98** geschaffene Umgebung einzurichten. Wie nachfolgend im Detail in Tabelle 1 angegeben, bietet der Versionkontrolleinstellungsdialog **122** Kontrollkästchen zum Umschalten zwischen jeweils drei Optionen. Die erste Option bestimmt, ob der Benutzer es bevorzugt, Elemente von Hand auszubuchen. Die zweite Option in dem Versionkontrolloptionendialog **122** gibt an, ob ein Element automatisch ausgebucht wird, wenn der Benutzer die Modifikation eines Elements versucht oder anderweitig auslöst. In dem Fall, daß Elemente automatisch ausgebucht werden, wird das VCAT-System **98** für den Prozesskonstrukteur, der die Konfigurationsanwendungen **96** verwendet, transparenter. Die dritte und letzte Option handelt davon, ob Elemente automatisch ausgebucht und eingebucht werden, wenn der Benutzer ein Element nach einer Sitzung, die eine Möglichkeit zur Modifikation bietet, versucht zu speichern.

[0053] Es sei angemerkt, daß der Versionskontrolleinstellungsdialog **122** andere Optionen enthalten kann, die von dem jeweiligen Benutzer ausgewählt oder aktiviert werden können. Beispielsweise kann jeder Benutzer die Option haben, keine Kommentare während eines Einbuchungsvorganges abzugeben. Weitere Details über das Vorsehen derartiger Kommentare sind jedoch nachfolgend in Verbindung mit Tabelle 2 angegeben.

Tabelle 1 – Versionskontrolleinstellungsdialog

Name	Typ	Min	Max	Standard	Inhalt
Manuelles Ausbuchen	Optionsfeld	n/a	n/a	ausgewählt	Bestimmt, ob ein Dialog angezeigt wird, um den Benutzer zum Ausbuchen eines Elements zur Modifikation aufzufordern
Automatisches Ausbuchen	Optionsfeld	n/a	n/a	nicht ausgewählt	Gibt an, ob das Element automatisch ausgebucht wird
Automatisch Ausbuchen und Einbuchen	Optionsfeld	n/a	n/a	nicht ausgewählt	Gibt an, ob das Element automatisch ausgebucht und eingebucht wird, wenn eine Elementkonfiguration modifiziert wird.

[0054] Der Benutzer kann die Auswahl des automatischen Ausbuchens wünschen, da beispielsweise Veränderungen an einem versionsfähigen Element Veränderungen an einem oder mehreren anderen versionsfähigen Elementen betreffen und verursachen können. Beispielsweise kann die Modifikation eines Elements, wie etwa eines Verbundfunktionsblocks, sich auf ein oder mehrere Module auswirken, welche diesen Verbundfunktionsblock verwenden. Das VCAT-System **98** bestimmt vorzugsweise während jedes Ausbuchungsvorganges, welches andere versionsfähige Element ausgebucht werden muss, um die Konfiguration eines Elements zu ändern. Die Modifikation dieser anderen versionsfähigen Elemente kann als "Folgeveränderungen" bezeichnet werden. Wenn der Benutzer das manuelle Ausbuchen gewählt hat, fordert das VCAT-System **98** den Benutzer auf, diese Elemente auszubuchen. Wenn das automatische Ausbuchen aktiviert ist, fordert das VCAT-System **98** den Benutzer nicht auf und bucht automatisch jedes der anderen Elemente aus, für welche Folgeveränderungen auftreten können.

[0055] Der Benutzer kann Elemente manuell ausbuchen und einbuchen, indem der geeignete Befehl verwendet wird, der über das Versionskontrolluntermenü **120** oder das Kontextmenü **116** angeboten wird. Vorzugsweise bestimmt das VCAT-System **98** dann, ob das ausgewählte Element ein versionsfähiges Element ist. Zu

dieser Zeit bestimmt das VCAT-System **98** auch, ob das ausgewählte Element bereits ausgebucht wurde. In dem Fall, in dem das ausgewählte Element bereits ausgebucht ist, alarmiert ein Meldungsdialog (nicht dargestellt) den Benutzer über diese Tatsache und gibt dem Benutzer die Gelegenheit, beispielsweise über eine Schaltfläche den Meldungsdialog zu bestätigen und in den vorherigen Zustand der Benutzerschnittstelle **94** zurückzukehren.

[0056] Da die Konfiguration des Prozesses **10** in hierarchischer Weise festgelegt ist, muss das VCAT-System **98** das Ausbuchen von Elementen ermöglichen, die untergeordnete Elemente haben, die auch versionsfähig sind. Entsprechend wird dem Benutzer während eines Ausbuchungsvorganges die Option gegeben, derartige Elemente rekursiv auszubuchen. Eine ähnliche Option wird in Verbindung mit der Einbuchungsoperation vorgesehen. In einer Ausführungsform erzeugt das VCAT-System **98** dann, wenn eine rekursive Ausbuchung oder Einbuchung von dem Benutzer ausgewählt wurde, ein Dialogfenster (nicht dargestellt), das dem Benutzer eine Liste von versionsfähigen untergeordneten Elementen zeigt, die ausgebucht (oder eingebucht) werden können. Der Benutzer kann anschließend jedes beliebige der aufgeführten Elemente auswählen (oder die Auswahl zurücknehmen), um eine selektive rekursive Operation auszulösen.

[0057] Wenn das VCAT-System **98** in die Konfigurationsanwendungen **96** integriert ist, kann das Erscheinungsbild der in dem Hauptfenster **80** dargestellten Elemente modifiziert sein, um anzuzeigen, daß das Element ausgebucht wurde. In einer Ausführungsform überlagert eine Buchungsmarke das zu dem Element gehörende Symbol. Es versteht sich, daß eine Vielzahl von anderen Erscheinungsformen verwendet werden kann. Buchungsmarken mit verschiedenen Farben können auch verwendet werden, um bestimmte Benutzer zu bezeichnen, oder wenn das Element von dem gegenwärtigen Benutzer oder einem anderen Benutzer ausgebucht wurde.

[0058] Wie [Fig. 9](#) zeigt, können ausgebuchte Elemente von Hand unter Verwendung der geeigneten Befehle, die in dem Versionskontrolluntermenü **120** und dem Kontextmenü **116** verfügbar gemacht werden, eingebucht werden. Ein Einbuchungsdialogfenster **124** wird vorzugsweise durch Auslösen der Einbuchungsprozedur durch den Benutzer erzeugt, wenn der Benutzer die Auswahl getroffen hat, mit einem manuellen Einbuchungsvorgang weiter zu arbeiten. Das Einbuchungsdialogfenster **214** enthält ein Kommentarfeld **126** zur Aufnahme von Kommentaren des Benutzers über die Modifikation des Elements. Der Kommentar kann beispielsweise eine Erklärung erhalten, warum die Modifikationen durchgeführt wurden. Zusätzlich zu dem Kommentarfeld **126** bietet das Einbuchungsdialogfenster **124** mehrere andere Optionen für den Benutzer, die jeweils im Detail nachfolgend in Tabelle 2 beschrieben sind. Insbesondere wird dem Benutzer die Option geboten, zu bestimmen, daß alle untergeordneten Elemente, die zu dem einzubuchenden Element gehören (die ebenfalls ausgebucht sind) eingebucht werden sollten, sowie die Auswahl, daß das Element für weitere Modifikationen ausgebucht bleiben sollte. Wie vorstehend angeführt, speichert die Einbuchungsoperation auch Daten, welche die Konfiguration darstellen, in der Versionskontrolldatenbank **102** als die letzte Konfigurationsversion. In diesem Fall besteht jedoch die Möglichkeit, obgleich das Element eingebucht wurde, weitere Modifikationen an der Konfiguration des Elements durchzuführen, weiterhin, da das Element ausgebucht bleibt. Das Einbuchungsdialogfenster **124** hat "OK"- und "Abbrechen"-Schaltflächen **128** und **130** zum Ausführen oder Abbrechen des Einbuchungsvorganges sowie eine "Unterschiede"-Schaltfläche **132**, welche einen Vergleich der gegenwärtigen Version des Elements (wie durch die in der Konfigurationsdatenbank **100** gespeicherten Daten dargestellt) und der modifizierten Version, die einzubuchen ist, auslöst. Die Vergleichsinformation wird von dem VCAT-System **98** durch Zugriff auf die Konfigurationsdatenbank **100** und die letzte Version in der Versionskontrolldatenbank **102** erzeugt und über die Benutzerschnittstelle **94** dargestellt. Das Erzeugen und Darstellen der "Unterschiede"-Information wird nachfolgend im Detail erläutert.

Tabelle 2 – Versionkontrollleinbuchungsdialog

Name	Typ	Min	Max	Standard	Inhalt
Rekursiv	Kontrollkästchen	n/a	n/a	nicht aktiviert	Bucht untergeordnete Elemente ein, die von diesem Benutzer ausgebucht wurden; vorzugsweise nur in bestimmten Ebenen der Konfigurationshierarchie anwendbar
Ausgebucht halten	Kontrollkästchen	n/a	n/a	nicht aktiviert	Zeigt an, ob das VCAT-System 98 das Element ausgebucht halten soll
Unterschiede	Schaltfläche	n/a	na/	n/a	Unterschiede zwischen der letzten Version (die einzubuchen ist) und der gegenwärtigen Version in der Konfigurationsdatenbank
Kommentar	Bearbeitungsfeld	n/a	n/a	leer	Vom Benutzer eingetragene Beschreibung der Veränderung

[0059] Beim Einbuchen eines Elements kann das VCAT-System **98** das Erscheinungsbild des Elements verändern, um anzuzeigen, daß das Element eingebucht wurde. Dieses Verändern des Erscheinungsbildes kann beispielsweise in dem Entfernen einer Buchungsmarke über dem Symbol, das zu diesem gehört, bestehen. Das VCAT-System **98** kann ferner die Funktionalität enthalten, welche es dem Benutzer erlaubt, eine Statusaktualisierung für alle Elemente in der Konfigurationsdatenbank **100** anzufordern, um sicherzustellen, daß diejenigen Elemente, die ausgebucht sind, mittels einer Buchungsmarke oder dergleichen auch so gekennzeichnet werden. Diese Funktionalität kann beispielsweise erforderlich sein, wenn zwei oder mehr Benutzer an der Konfiguration des Prozesses zur gleichen Zeit arbeiten und einer der Benutzer kürzlich ein Element ausgebucht hat, das auf der Benutzerschnittstelle **94** eines anderen Benutzers dargestellt ist.

[0060] Die Auswahl der "Abbrechen"-Schaltfläche in dem Einbuchungsdialogfenster **124** schließt das Einbuchungsdialogfenster ohne Auslösen eines Einbuchungsvorganges.

[0061] Das VCAT-System **98** kann jedoch auch einen Benutzer in die Lage versetzen, einen "Ausbuchen rückgängig"-Task auszulösen, der die Buchungsmarke über dem Element entfernt und, wenn das Element modifiziert wurde, die Daten, die die letzte Version angeben, aus der Versionskontrolldatenbank **102** abrufen und die Daten in die Konfigurationsdatenbank **100** importiert. Dieser Vorgang stellt die Konfiguration des versionsfähigen Elements in dem Zustand wieder her, der zum Zeitpunkt des Ausbuchungsvorganges der aktuelle Zustand war. Dieser Task kann über ein Drop-Down- oder Kontextmenü in Verbindung mit dem VCAT-System **98** verfügbar gemacht werden.

[0062] Die Konfigurationsanwendungen **96** können zusätzliche Wege bieten, um die Ausbuchungs-/Einbuchungsoperationen auszulösen. Beispielsweise kann der Benutzer ein Element in dem Explorersystem auswählen und die "Eigenschaften", die zu diesem gehören, über eine Betätigung der rechten Maustaste oder eine ähnliche Zeigeeinrichtungsbetätigung auswählen. Ein Eigenschaften-Fenster wird anschließend gemäß den Standardfenstertechniken erzeugt, das das Editieren des Inhalts von verschiedenen "Eigenschaften"-Feldern ermöglicht, die innerhalb des Fensters angezeigt werden. Ein Modul kann beispielsweise eine dazugehörige Textbeschreibung haben oder bestimmte Parameter während der Ausführung innerhalb des Prozesses verwenden. Die Textbeschreibung und die Parameterwerte können als "Eigenschaften" des Moduls angegeben werden. Wenn der Benutzer eine Eigenschaft modifiziert und eine "OK" oder "Ausführen"-Schaltfläche betätigt, fordert das VCAT-System **98** den Benutzer auf, das Element auszubuchen (wenn das automatische Ausbuchen nicht aktiviert ist), woraufhin ein Einbuchungsvorgang wie vorstehend beschrieben ausgelöst wird. Ähnliche Aktionen können durch Nutzen eines "Öffnen"-Befehls erzeugt werden, der in einem Befehlsmenü der Prozeßkonfigurationsanwendungen zur Verfügung gestellt wird.

[0063] Ein Ausbuchungsvorgang kann auch von dem VCAT-System **98** ausgelöst werden, wenn ein Benutzer

versucht, ein Element in dem Explorersystem neu zu benennen. In Übereinstimmung mit Standardfenstertechniken kann der Benutzer den Namen auswählen, der zu einem Element gehört, um die Bearbeitung desselben zu ermöglichen. Sobald der Benutzer mit dem Bearbeiten des Namens fertig ist und versucht, die Modifikation einzugeben, wird der neue Name nur dann akzeptiert, wenn das Element ausgebucht wird. Wenn das Element bereits ausgebucht ist, wird das Element in der Konfigurationsdatenbank **100** neu benannt und Daten, die eine neue Version des Elements anzeigen, werden in der Versionskontrolldatenbank **102** gespeichert, wenn dieses eingebucht wird. Zu dieser Zeit kann das VCAT-System **98** weitere Daten erzeugen, die den Kommentar darstellen, der die Namensänderungsoperation beschreibt, und anschließend diese Daten mit den Daten verbinden, welche die neue Version des Elements bezeichnen. Wenn das neu zu benennende Element noch nicht ausgebucht ist, wird der Benutzer über ein Dialogfenster (nicht dargestellt) aufgefordert, das Element auszubuchen (wenn das automatische Ausbuchen nicht aktiviert ist).

[0064] Wie vorstehend angeführt werden Daten, die jede frühere Konfiguration eines Elements darstellen, in der Versionskontrolldatenbank zusammen mit den Daten gespeichert, die eine diesen zugewiesene Version wiedergeben. Die Version wird vorzugsweise durch eine Nummer identifiziert, kann jedoch auch in jeder anderen Weise bezeichnet werden. Das Versehen jeder früheren Konfiguration mit einer Versionsnummer kann den Benutzer während der Analyse des Inhalts der Versionskontrolldatenbank **102** unterstützen, ebenso wenn der Prozeß in der Laufzeitumgebung ist. Zu diesem Zweck wird die Versionsnummer oder ein Identifizierungskennzeichen eines Elements vorzugsweise als ein Elementparameter eingeschlossen, wenn das Element in die Steuereinrichtung **12** und/oder eine Stapelausführungsroutine heruntergeladen wird (die Stapelausführungsroutine sitzt in der Workstation **14**, um die Ausführung eines Stapelprozesses zu verwalten und hat in Verbindung damit die Aufsicht über die Steuereinrichtung **12**). Mit anderen Worten enthalten die Daten, die von den Workstations **14** heruntergeladen werden, wenn das Prozeßsteuersystem auf die Implementierung eines Prozesses vorbereitet wird, Versionsidentifizierungsdaten für bestimmte Elemente in der Konfigurationsdatenbank **100** (das heißt jedes Modul, jede Phase, jedes Vorgangselement etc.), welche an dem Herunterladevorgang beteiligt sind. Die Versionsinformation ist anschließend in der Laufzeitumgebung für eine Prozeßbedienungsperson über die Steuereinrichtung **12** und/oder die Stapelausführung verfügbar. Die Kenntnis der Versionsinformation kann die Kommunikation zwischen den Prozeßbedienungspersonen und den Prozeßgestaltern, die in den Konfigurationsanwendungen **96** arbeiten, erleichtern.

[0065] Es sei angemerkt, daß das Herunterladen eines Elements in die Steuereinrichtung **12** oder in die Laufzeitumgebung allgemein eines oder mehrere Dialogfenster (nicht dargestellt) einschließt, die von den Konfigurationsanwendungen **96** für die Benutzerschnittstelle **94** erzeugt werden, welche den Benutzer durch den Herunterladevorgang leiten. Diese Dialogfenster können den gleichen Fenstern entsprechen, die von den Konfigurationsanwendungen **96** erzeugt werden, wenn das VCAT-System **98** nicht in diese integriert ist. Das VCAT-System **98** kann jedoch verifizieren, daß alle Elemente, die heruntergeladen werden sollen, nicht ausgebucht sind, bevor das Ablaufen des Herunterladevorganges zugelassen wird. Wenn Elemente ausgebucht sind, kann das VCAT-System **98** ein oder mehrere Fehlerdialogfenster (nicht dargestellt) erzeugen, die den Benutzer über die Situation informieren und die ausgebuchten Elemente kennzeichnen. Derartige Fehlerdialogfenster können den Einbuchungsvorgang erleichtern, indem in diesen eine Schaltfläche vorgesehen ist, die einen Einbuchungsvorgang von einem oder mehreren ausgebuchten Elementen auslösen, die in dem Dialogfenster identifiziert werden.

[0066] Wie [Fig. 10](#) zeigt, kann die Versionsidentifizierungsinformation den Prozeßgestaltern über einen Versionsüberwachungsprotokollbericht (nachfolgend "Überwachungsprotokollbericht" genannt) verfügbar gemacht werden, der von dem VCAT-System **98** gemäß einer Ausführungsform der vorliegenden Erfindung erzeugt wird. Allgemein wird jede Veränderung oder Modifikation an der Prozeßkonfiguration in der Versionskontrolldatenbank **102** als eine bestimmte Version, die zu dem Prozeß gehört, aufgezeichnet. Die Daten, welche die Version und den Gegenstand der Modifikation angeben, können mit Daten verknüpft sein, welche das Einbuchungsdatum und die Einbuchungszeit, den Benutzer, der das Element eingebucht hat, und die Gründe für die Modifikation darstellen. Es sei angemerkt, daß das VCAT-System **98** auch Daten speichern kann, welche das Ausbuchungsdatum und die Ausbuchungszeit sowie die Identität des Benutzers, der das Element ausbucht, darstellen.

[0067] Das Überwachungsprotokoll kann dem Benutzer über ein Überwachungsprotokolldialogfenster **140** zur Verfügung gestellt werden, das von dem VCAT-System **98** für die Benutzerschnittstelle **94** erzeugt wird. Genauer ausgedrückt kann ein "Zeige Historie"-Task von dem Benutzer durch Verwenden einer der vorstehend beschriebenen Techniken ausgelöst werden, zu welchen beispielsweise das Kontextmenü **116** gehört, während der Benutzer innerhalb einer der Konfigurationsanwendungen **96** ist. Das Überwachungsprotokoll der Historie kann auch dem Benutzer über jede andere Anzeigeeinrichtung, wie zum Beispiel einen Drucker, zur

Verfügung gestellt werden oder kann in einer elektronischen Version des Protokolls eingebettet sein, so daß beispielsweise Daten, die das Überwachungsprotokoll darstellen, elektronisch über ein Netzwerk weitergegeben werden.

[0068] Das Überwachungsprotokolldialogfenster **140** enthält einen Tabellenabschnitt **142**, der einen Kopfabschnitt **144** hat, welcher einer Vielzahl von Feldnamen zum Darstellen von Versionsinformationen bezeichnet, die unter Bezug auf eine Versionsnummer, einen Benutzernamen, der mit dem Ausbuchen verbunden ist, auf Datum und Zeit des Einbuchens und eine Beschreibung der Modifizierung oder der Aktion bezogen sind. Die Versionsinformationen werden in Zeilen (oder Einträgen) unter den Feldnamen wiedergegeben, wie [Fig. 10](#) zeigt. Jeder Eintrag ist durch eine Zeigeeinrichtung oder anderweitig auswählbar, um eine Vielzahl von Operationen auszulösen. Die von dem VCAT-System **98** in Verbindung mit dem Überwachungsprotokolldialogfenster gebotene Funktionalität ist nachfolgend in Tabelle 3 zusammengefaßt.

Tabelle 3 – Überwachungsprotokolldialogfenster

Name	Typ	Min	Max	Standard	Inhalt
Version, Benutzer, Datum, Aktion,	Kontrollliste	n/a	n/a	n/a	Jede Zeile entspricht einem Historieneintrag für das Element
Schließen	Schaltfläche	n/a	n/a	aktiviert	Schließt den Dialog
Rückzug	Schaltfläche	n/a	n/a	aktiviert	Rückzug auf die ausgewählte Version
Unterschiede	Schaltfläche	n/a	n/a	aktiviert	Vergleicht zwei ausgewählte Versionen oder die ausgewählte Version mit der gegenwärtigen Version.
Details	Schaltfläche	n/a	n/a	aktiviert	Zeigt Kommentare der ausgewählten Version
Betrachten	Schaltfläche	n/a	n/a	aktiviert	Zeigt das exportierbare Element im Textformat oder grafischen Format

[0069] Wie Tabelle 3 zeigt, kann der Benutzer die Details einer Versions des Elements betrachten, indem die Zeile des Tabellenabschnitts **142** ausgewählt wird, die dieser Version entspricht, und anschließend die Betrachten-Schaltfläche **143** betätigt wird. Auf die in der Versionskontrolldatenbank gespeicherten Daten wird anschließend zugegriffen, um die Konfigurationsinformation für die ausgewählte Version des Elements entweder im Textformat oder im grafischen Format oder in beiden Formaten basierend auf der Art des Elements darzustellen. Bestimmte Elemente können beispielsweise nur Text bilden und können somit nicht im grafischen Format betrachtet werden. Beispielsweise können versionsfähige Elemente, wie etwa eine Workstation oder I/O-Einrichtung, nur Parameter enthalten, die modifiziert werden können. Die Parameter liegen in Textform vor und somit ist für derartige Elemente keine grafische Ansicht verfügbar. Einige Elemente können jedoch sowohl im grafischen Format als auch im Textformat betrachtet werden und schließen allgemein Elemente, wie zum Beispiel Module, Verbundfunktionsblöcke oder andere Elemente, die auf einem sequentiellen Flußdiagrammalgorithmus basieren, ein. In diesen Fällen kann der Benutzer aufgefordert werden, eines der Formate zu wählen oder alternativ können beide Formate dargestellt werden.

[0070] Ein Beispiel eines im Textformat gezeigten Elements ist in [Fig. 11](#) dargestellt, bei der es sich um eine Bildschirmdarstellung handelt, die von dem VCAT-System **98** für die Benutzerschnittstelle **94** erzeugt wird. Die Bildschirmdarstellung enthält ein Textansichtsfenster **146**, das einen Rahmen **148** für die zu dem Element gehörende Textinformation hat. Das Textansichtsfenster **146** bietet dem Benutzer verschiedene Betrachtungsoptionen oder Operationen, die nachfolgend beschrieben werden.

[0071] [Fig. 12](#) ist ein Beispiel einer Bildschirmdarstellung eines Elements, das im grafischen Format dargestellt wird. Wie die Bildschirmdarstellung für das Textformat wird auch das grafische Format über ein grafisches Ansichtsfenster **150** mit einem Rahmen **152** für die zu dem Element gehörende grafische Information dargestellt. Wie [Fig. 11](#) und Tabelle 3 zeigen, kann der Benutzer auch jegliche Kommentare, die zu der Zeit des Einbuchens für eine bestimmte Version aufgezeichnet wurden, analysieren, indem eine Details-Schaltfläche **154**

nach dem Auswählen der gewünschten Version betätigt wird. Die Kommentare sind vorzugsweise über das Einbuchungsdialogfenster **124** gespeichert worden, es können jedoch auch andere Schemata verwendet werden, um Kommentare darstellende Daten mit einer bestimmten Version zu verbinden. Das Überwachungsprotokolldialogfenster **140** erlaubt es ferner dem Benutzer, zwei Versionen des Elements durch Verwendung einer Zeigeeinrichtung (oder einer anderen Auswahleinrichtung) auszuwählen, um die beiden entsprechenden Einträge in dem Tabellenabschnitt **142** auszuwählen. Das Auswählen von zwei Einträgen ermöglicht es einem Benutzer, eine Unterschiede-Operation durch Auswählen einer Unterschiede-Schaltfläche auszulösen, was wiederum das VCAT-System **98** veranlasst, einen Unterschiede-Fensterdialog zu erzeugen, welcher die Konfigurationsinformation für jede ausgewählte Version eines Elements in einem Format darstellt, das dem Benutzer das Vergleichen der beiden Versionen erlaubt. Die Details der Unterschiede-Operation werden nachfolgend im Detail erörtert. Wenn nur ein einzelner Eintrag ausgewählt wurde und die Unterschiede-Schaltfläche ausgewählt wird, erzeugt das VCAT-System **98** das Unterschiede-Fenster in der Weise, daß die ausgewählte Version mit der gegenwärtigen Version des in der Konfigurationsdatenbank **100** gespeicherten Elements verglichen wird.

[0072] Das Überwachungsprotokolldialogfenster **140** erlaubt es allgemein auch einem Benutzer, eine der früheren Versionen der Konfiguration des Elements zu implementieren. Genauer ausgedrückt kann das VCAT-System **98** von dem Benutzer angewiesen werden, eine Routine auszulösen, welche auf die Versionskontrolldatenbank **102** zugreift, um die Daten abzurufen, die eine frühere Konfigurationsversion darstellen, um die in der Konfigurationsdatenbank **100** in Verbindung mit dem ausgewählten Element gespeicherten Daten zu modifizieren. Die Modifikation der Konfigurationsdatenbank **100** kann dann durchgeführt werden, indem die Daten von der Versionskontrolldatenbank **102** in die Konfigurationsdatenbank **100** importiert werden. Um diesen "Rückzug" auf eine frühere Konfigurationsversion eines Elements zu implementieren enthält gemäß einer Ausführungsform das Überwachungsprotokolldialogfenster **140** eine Rückzug-Schaltfläche **158**, die von dem Benutzer ausgewählt werden kann, sobald ein Eintrag in dem Tabellenabschnitt **142** ausgewählt wurde.

[0073] In einer Ausführungsform bietet das VCAT-System **98** dem Benutzer die Option, sich auf eine Konfiguration zurückzuziehen, die in die Steuereinrichtung **12** zur Nutzung innerhalb der Laufzeitumgebung heruntergeladen wurde. Diese Rückzugsoperation kann durch Auswählen eines Taskelements wie etwa "heruntergeladene Datei wiederherstellen" aus einem Drop-Down- oder Kontextmenü ausgelöst werden und das Erzeugen eines spezialisierten Überwachungsprotokolldialogfensters (nicht dargestellt) durch das VCAT-System **98** einschließen, das nur diejenigen Versionen aufführt, die heruntergeladen wurden. Wie weiter unten im Detail beschrieben wird, enthält die Versionskontrolldatenbank **102** vorzugsweise Daten, die angeben, ob eine Version zur Verwendung in der Laufzeitumgebung heruntergeladen wurde oder nicht. Eine Version kann dann von dem Benutzer ausgewählt werden, woraufhin die zugehörige Konfigurationsinformation aus der Versionskontrolldatenbank **102** von dem VCAT-System **98** abgerufen wird und als die letzte Konfigurationsversion gespeichert wird.

[0074] Vorzugsweise bestimmt das VCAT-System **98**, ob die Konfigurationsdatenimportoperation erfolgreich ist, indem andere Elemente in der Konfigurationshierarchie und in dem Prozeß geprüft werden, die von dem Element abhängig, sind. Wenn beispielsweise eine Konfigurationsversion eines Moduls wiederhergestellt wird, kann die Konfigurationsversion ein bestimmtes untergeordnetes Element (zum Beispiel einen Verbundfunktionsblock) benötigen, das vorhanden sein muss. Das Modul wird dann als "abhängig" in dem Sinn bezeichnet, daß der Erfolg der Rückzugsoperation von der andauernden Existenz des untergeordneten Elements abhängig ist. Ein Modul kann zahlreiche verschiedene Arten von Abhängigkeiten haben, wie zum Beispiel eingebettete Funktionsblöcke, verknüpfte Funktionsblöcke, Enumerationssätze, Alarmtypen, andere Module, Anlagenbereiche und Steuereinrichtungen. Demgemäß prüft in einer Ausführungsform das VCAT-System **98** jedes untergeordnete Element des Elements, das einer Rückzugsoperation unterzogen wurde, um zu bestimmen, ob versionsfähige Elemente gelöscht wurden.

[0075] In einer bevorzugten Ausführungsform werden die Daten, die in die Konfigurationsdatenbank importiert wurden, in die Versionskontrolldatenbank **102** sowie die letzte Konfigurationsversion des Elements eingebucht. Dieses Einbuchen tritt gleichzeitig mit der Modifikation der Konfigurationsdatenbank **100** auf und schließt vorzugsweise das Speichern von Daten ein, die einen Kommentar darstellen, der angibt, daß der Benutzer auf eine frühere Konfigurationsversion des Elements zurückgegriffen hat. Alternativ kann das VCAT-System **98** ein Dialogfenster erzeugen, um dem Benutzer die Option zu bieten, einen selbstverfaßten Kommentar einzugeben, einen Standardkommentar einzugeben oder keinen Kommentar einzugeben. Wenn das Element ausgebucht wurde, bevor der Rückzugstask ausgelöst wurde, kann das VCAT-System **98** das Element ausgebucht halten, nachdem der Rückzugstask vollendet wurde, was jedoch nicht zwangsweise der Fall sein muss.

[0076] Schließlich enthält das Überwachungsprotokolldialogfenster **140** eine Schaltfläche zum Schließen des Dialogfensters **140** und Zurückkehren zu der Benutzerschnittstelle **94** in ihrem vorherigen Zustand.

[0077] Wie [Fig. 10](#) zeigt, ist eine der Aktionen (abgesehen von einer Einbuchungsoperation) die in dem Überwachungsprotokoll aufgezeichnet ist, eine Kennzeichnungsoperation für die gesamte Prozesskonfiguration. Wie [Fig. 13](#) zeigt, kann der Benutzer eine Kennzeichnungsoperation durch Auswählen eines versionsfähigen Elements innerhalb beispielsweise eines der Module, das unter Verwendung der Konfigurationsanwendungen gestaltet wurde, und Auswählen des Kennzeichnungs-Tasks über entweder das Drop-Down-Menü oder das Kontextmenü, die von dem VCAT-System **98** erzeugt werden, auslösen. Ansprechend darauf erzeugt das VCAT-System **98** ein Kennzeichnungs-Dialogfenster **162**, das ein Kennzeichenfeld **164** und ein Kommentarfeld **166** enthält. Der Benutzer wird anschließend aufgefordert, ein Kennzeichen einzugeben, das an dem ausgewählten Element (und untergeordneten Elementen) in der Versionskontrolldatenbank **102** angewandt wird. Die letzte Konfigurationsversion jedes Elements, die in der Versionskontrolldatenbank **102** gespeichert ist, wird anschließend dem vom Benutzer eingegebenen Kennzeichen zugewiesen. Die in der Versionskontrolldatenbank **102** gespeicherten Überwachungsprotokoll Daten werden auch aktualisiert, so daß sie das Zuweisen des Kennzeichens wiedergeben, welches vorzugsweise an der jüngsten Konfigurationsversion angewandt wird. Genauer ausgedrückt, gibt die Versionsaktion in den Überwachungsprotokoll-Konfigurationshistoriendaten wieder, daß die Version einem neu zugewiesenen Kennzeichen entspricht, das den eingegebenen Namen trägt, und die damit verbundenen Details entsprechen den Informationen, die von dem Benutzer in dem Kommentarfeld **166** des Kennzeichnungsdialogfensters **162** eingegeben wurden.

[0078] Das Vergeben eines Kennzeichens kann für einen Prozeßgestalter in dem Sinn vorteilhaft sein, daß der Prozeßgestalter mittels eines Kennzeichens, wie etwa "AUF STEUEREINRICHTUNG 1 GELADEN" notieren kann, welche Konfigurationsversionen von verschiedenen Elementen in die Steuereinrichtung **12** zur Implementierung in dem Prozeß heruntergeladen wurden. Vorzugsweise weist jedoch das VCAT-System **98** jedem heruntergeladenen Element automatisch ein Kennzeichen zu, um die Konfigurationsversion als in die Laufzeiteinrichtung heruntergeladen zu kennzeichnen. Daten, die weitere Informationen darstellen, wie etwa das Datum und die Zeit des Herunterladens und die Elemente, die gleichzeitig heruntergeladen wurden, können auch in Verbindung mit dem Kennzeichen gespeichert werden.

[0079] Das VCAT-System **98** kann dann entsprechend die Fähigkeit zur Verfügung stellen, die Konfiguration des gesamten Prozesses oder eines Abschnittes davon auf die Version zurückzusetzen, die mit einem Kennzeichen verbunden ist, das mittels des vorstehend beschriebenen Vorganges für jedes der betroffenen Elemente in der Versionskontrolldatenbank **102** gesetzt wurde. In einer Ausführungsform ist diese Fähigkeit nur auf das Stammelement bzw. das Element in der höchsten Ebene in der Konfigurationshierarchie anwendbar, die unter Verwendung der Konfigurationsanwendungen **96** entwickelt wurde. In diesem Fall kann ein Task "System auf Kennzeichen rücksetzen" ausgelöst werden, wenn der Benutzer das Stammelement ausgewählt hat und weiter das entsprechende Taskelement aus einem Drop-Down-Menü oder einem Kontextmenü ähnlich dem in [Fig. 7](#) gezeigten Menü auswählt. Ein Warndialog (nicht dargestellt) kann von dem VCAT-System **98** erzeugt werden, der den Benutzer zur Bestätigung der Absicht auffordert, das System auf ein Kennzeichen rückzusetzen. Wenn der Benutzer angibt, daß der Task "System auf Kennzeichen rücksetzen" gewünscht ist, kann der Benutzer dann aufgefordert werden, den Namen des Kennzeichens einzugeben, auf welches das System rückgesetzt wird. Alternativ kann das VCAT-System **98** das System automatisch auf das letzte Kennzeichen rücksetzen, das in der Versionskontrolldatenbank **102** gesetzt wurde. In jedem Fall greift dann, wenn der Task "System auf Kennzeichen rücksetzen" ausgeführt wird, das VCAT-System **98** auf die Versionskontrolldatenbank **102** zu, um die Daten abzurufen, welche die Konfigurationsversion darstellen, die zu dem Kennzeichen für jedes Element gehört, und importiert die abgerufenen Daten in die Konfigurationsdatenbank **100**. Das VCAT-System **98** modifiziert anschließend die Versionskontrolldatenbank **102**, so daß sie die Rücksetzung wiedergibt, indem Daten gespeichert werden, die eine neue Konfigurationsversion für jedes Element angeben, sowie Daten, die zu diesen gehören, welche die Rücksetzung auf das Kennzeichen wiedergeben.

[0080] In einer Ausführungsform können diese Daten mit der Konfigurationsversion verbunden sein, indem Daten gespeichert werden, die einen Kommentar darstellen, der die Rücksetzung auf das Kennzeichen angibt.

[0081] Gemäß einer Ausführungsform speichert das VCAT-System **98** vorzugsweise weiterhin Daten in der Versionskontrolldatenbank **102**, die ein Element darstellen, das gelöscht oder in anderer Form aus der Konfigurationsdatenbank **100** entfernt wurde. In diesem Fall kann auf die Versionskontrolldatenbank **102** zurückgegriffen werden, wenn der Benutzer die Entscheidung trifft, ein Rücksetzen auf eine Prozeßkonfiguration durchzuführen, die das gelöschte Element verwendet. Wie [Fig. 14](#) zeigt, kann das VCAT-System **98** einen Benutzer auch in die Lage versetzen, Elemente zu entfernen, die aus der Konfigurationsdatenbank **100** gelöscht wur-

den. Zu diesem Zweck erzeugt das VCAT-System **98** ein Wiederherstellen/Entfernen-Dialogfenster **168**, das eine Liste dieser gelöschten Elemente enthält. Das Wiederherstellen/Entfernen-Dialogfenster **168** kann nach Auswählen eines Wiederherstellen/Entfernen-Taskelements entweder in einem Drop-Down-Menü oder einem Kontextmenü erzeugt werden. Ein derartiges Taskelement kann aktiviert werden oder anderweitig dem Benutzer zur Verfügung gestellt werden, nachdem ein Element innerhalb der Konfigurationsanwendungen **96** ausgewählt wurde. Wenn das ausgewählte Element ein oder mehrere untergeordnete Elemente hat die aus der Konfigurationsdatenbank **100** gelöscht wurden, kann der Benutzer die Wiederherstellungsoperation über das Dialogfenster **168** auslösen, um Daten, die eines oder mehrere der gelöschten Elemente darstellen, zu der Konfigurationsdatenbank **100** hinzuzufügen.

[0082] Das Wiederherstellen/Entfernen-Dialogfenster **168** enthält eine Wiederherstellen-Schaltfläche **170** zum Wiederherstellen eines oder mehrerer gelöschter Elemente in der Liste, die von dem Benutzer ausgewählt wurden. Beim Wiederherstellen eines gelöschten Elements kann das VCAT-System **98** auf die Versionskontrolldatenbank **102** zugreifen, um die letzte Konfigurationsversion des gelöschten Elements abzurufen, und anschließend die dazugehörigen Daten in die Konfigurationsdatenbank **100** importieren. Alternativ fordert das VCAT-System **98** den Benutzer zum Auswählen einer früheren Konfigurationsversion auf, soweit mehrere Versionen in der Versionskontrolldatenbank **102** gespeichert sind.

[0083] Das Auswählen einer Entfernen-Schaltfläche **172**, die in dem Dialogfenster **168** vorgesehen ist, führt andererseits zu dem Entfernen aller Daten, die zu dem Element gehören, aus der Versionskontrolldatenbank **102**. Weitere Details hinsichtlich des Wiederherstellen/Entfernen-Dialogfensters **168** sind nachfolgend in Tabelle 4 angegeben.

Tabelle 4 – Wiederherstellen/Entfernen-Dialogfenster

Name	Typ	Min	Max	Standard	Inhalt
Elemente	Kontrollkästchenliste	n/a	n/a	Liste von untergeordneten Elementen, die basierend auf einem ausgewählten übergeordneten Element gelöscht wurden	Liste aller untergeordneten Elemente, die basierend auf einem ausgewählten übergeordneten Element gelöscht wurden
Wiederherstellen	Schaltfläche	n/a	n/a	n/a	Element wird in der Konfigurationsdatenbank wiederhergestellt, indem die Daten aus der Versionskontrolldatenbank abgerufen werden und das Element in die Konfigurationsdatenbank importiert wird
Entfernen	Schaltfläche	n/a	n/a	n/a	Element und alle Überwachungsprotokollhistoriendaten werden aus der Versionskontrolldatenbank dauerhaft entfernt

[0084] Gemäß einer Ausführungsform der Erfindung unterstützt das VCAT-System **98** eine Versionskontroll- und Überwachungsprotokollfunktionalität für Elemente, die nicht von den Konfigurationsanwendungen **96** erstellt, entwickelt oder verwaltet werden. Beispielsweise kann ein Benutzer ein Dokument in einem Textverarbeitungsprogramm erstellen, welches die Funktionalität eines bestimmten Elements beschreibt. Es wäre wünschenswert, jede Bearbeitung des Dokuments zu speichern und zu verfolgen, da die Konfiguration und somit die Beschreibung modifiziert werden. Zu diesem Zweck wird das Hauptkontrollfenster **80**, das von dem Explo-

rsystem für die Benutzerschnittstelle **94** erzeugt wird, verwendet, um ein neues Versionskontrollelement zu bezeichnen. Im einzelnen erweitert zunächst der Benutzer ein "Benutzer-Arbeitsraum"-Objekt (das heißt ein Objekt in dem Explorersystem). Ein Taskelement, das auf das Erstellen eines neuen Ordners in dem Hierarchierahmen **82** gerichtet ist, wird dann vom Benutzer ausgewählt, und ein neuer Ordner wird gemäß bekannten Fensterbearbeitungstechniken erstellt und benannt. In ähnlicher Weise wird ein Taskelement ausgewählt, das auf das Erstellen eines neuen Elements gerichtet ist, und dem Benutzer wird anschließend Gelegenheit gegeben, ein Dokument oder eine Datei zu suchen, um sie als neues Element in den soeben erstellten Ordner einzusetzen. Sobald das Dokument oder die Datei in den Ordner eingesetzt sind, wird diese in der Versionskontrolldatenbank **102** als weiteres Element hinzugefügt, für welches Versionen und versionsbezogene Informationen gespeichert werden können. Nachfolgend können jede Version der Daten oder des Dokuments aus der Versionskontrolldatenbank **102** unter Verwendung eines Abrufen-Taskelements abgeholt werden, das dem Benutzer über ein Drop-Down-Menü oder ein Kontextmenü zur Verfügung gestellt wird. Der Abrufen-Task bietet dem Benutzer ein Dialogfenster (nicht dargestellt), das die Eingabe einer gewünschten Speicherposition für die abgerufene Datei anfordert, wie zum Beispiel eine Diskette. Die ursprüngliche Anwendung, die zum Erstellen dieser Datei oder des Dokuments genutzt wurde (beispielsweise eine Textverarbeitungsanwendung) kann anschließend verwendet werden, um auf den Inhalt der Datei oder des Dokuments von der bezeichneten Speicherposition zuzugreifen.

[0085] Wie [Fig. 15](#) zeigt, erzeugt das VCAT-System **98** vorzugsweise ein Mitteilungsdialogfenster **174** zur Anzeige von Versionskontrollbestätigungsinformationen für den Benutzer über die Benutzerschnittstelle **94**. Das Mitteilungsdialogfenster **174** enthält ein Textfeld **176**, in dem Textbeschreibungen jeder Versionskontrolloperation entsprechend ihrem Auftreten dokumentiert sind. Allgemein erlaubt es das Mitteilungsdialogfenster **174** einem Benutzer, durch eine Historie der Operationen zu scrollen, die vorgenommen wurden, seit eine Löschen-Schaltfläche **178** zuletzt von dem Benutzer betätigt wurde. Eine Abbrechen-Schaltfläche **180** ist ebenso vorgesehen, um den Benutzer in die Lage zu versetzen, eine Versionskontrolloperation bei der nächsten Gelegenheit abzubrechen. Wenn beispielsweise der Benutzer eine Rückzugsoperation ausgelöst hat, kann die Ausführung der Operation durch das VCAT-System **98** durch das Auswählen der Abbrechen-Schaltfläche **180** beendet werden, sofern die Durchführung der Operation beispielsweise nicht bereits vollendet wurde. Das Mitteilungsdialogfenster **174** kann automatisch gemäß bekannten Fenstertechniken verkleinert werden, wenn die Versionskontrolloperationen für eine bestimmte Zeitdauer nicht durchgeführt wurden. Das Mitteilungsdialogfenster **174** wird dann automatisch auf der Benutzerschnittstelle **94** erneut dargestellt, wenn die nächste Versionskontrolloperation ausgelöst wird.

[0086] Die Art und Weise, in der das VCAT-System **98** dem Benutzer eine Textdarstellung der Konfiguration eines Elements über das Textansichtsfenster **146** bietet, wird nachfolgend in Verbindung mit [Fig. 11](#) beschrieben. Das VCAT-System **98** kann in der Lage sein, die Konfigurationsinformation jedes Elements in der Versionskontrolldatenbank **102** in einem Textformat darzustellen, wobei diese Möglichkeit nicht unbedingt vorhanden sein muss. Wie vorstehend erläutert, können einige Elemente gemäß ihrer Natur Konfigurationsinformationen haben, die auch grafisch dargestellt werden können.

[0087] Das Format, in dem die Konfigurationsinformation für ein Element dargestellt wird, ist für die praktische Ausführung der vorliegenden Erfindung nicht wesentlich. Das VCAT-System **98** nutzt jedoch vorzugsweise ein Textformat, bei dem das Ausmaß des horizontalen Scrollens, das für den Benutzer zum Betrachten sowohl einer einzelnen Konfigurationsversion eines Elements als auch beim Vergleichen von zwei Versionen erforderlich ist, minimiert wird. Ferner ist bevorzugt, daß Schlüsselwörter und Kennzeichen verwendet werden, um Attribute, wie zum Beispiel Objekttyp und Eigenschaften, zu kennzeichnen. Zu den Beispielen von verwendbaren Kennzeichen zur Bezeichnung von Elementtypen zählen: "PARAMETER", "FUNKTIONSBLOCK" und "STEUEREINRICHTUNG". Beispiele von Eigenschaftskennzeichen sind: "NAME" und "BESCHREIBUNG". Die Schlüsselwörter und Kennzeichen können dann in die entsprechende Sprache übersetzt werden. Beispielsweise würde eine japanische Version der Benutzerschnittstelle **94** die Schlüsselwörter auf japanisch anzeigen. Bedingungsansagen und andere logische Aussagen in der Sprache können ebenso in ein leicht zu lesendes Format zur Darstellung in der Textansicht übersetzt werden.

[0088] In einer Ausführungsform enthält das Textformat für jedes Element begrenzende Sprachelemente, wie etwa BEGINN und ENDE. Beispielsweise hat die Konfigurationsversion Nummer 13 eines Moduls einen Parameter, der von den folgenden BEGINN und ENDE Aussagen begrenzt ist:

```
BEGINN Parameter "ABS_PRESS_CF"
ENDE Parameter "ABS_PRESS_CF"
```

[0089] In einer anderen Ausführungsform kann der Name eines Elements nicht in Verbindung mit einem Kennzeichen identifiziert sein, wie etwa in dem folgenden Beispiel:

```
BEGINN FUNKTIONS_BLOCK
NAME = "AI1"
DEFINITION = "AI"
BESCHREIBUNG = "Analog Eingabe"
LINKS = 200
OBEN = 200
HÖHE = 125
BREITE = 400
ENDE FUNKTIONS_BLOCK
```

[0090] Das Textansichtsdialogfenster **146** enthält eine Suchen-Schaltfläche **200**, eine Abwärtssuche-Schaltfläche **202** und eine Aufwärtssuche-Schaltfläche **204**. Die Schaltflächen **200**, **202** und **204** erlauben es allgemein einem Benutzer, die in dem Rahmen **148** angebotene Textinformation in Übereinstimmung mit einer darin ausgewählten Zeichenkette mit einer Zeigeeinrichtung oder dergleichen gemäß bekannten Fensterarbeitstechniken zu durchsuchen. Zu diesem Zweck kann die Betätigung der Suchen-Operation durch Auswählen der Suchen-Schaltfläche **200** das VCAT-System **98** veranlassen, ein Suchen-Dialogfenster (nicht dargestellt) zu erzeugen, um das Durchsuchen des Rahmens **148** durch einen Benutzer zu erleichtern. Eine Zeichenkette kann dann in ein Feld eingegeben werden und das erste Auftreten der Zeichenkette kann beispielsweise durch Betätigung einer OK-Schaltfläche (nicht dargestellt) in dem Suchen-Dialogfenster gefunden werden. Der Benutzer kann anschließend die Schaltflächen **202** und **204** verwenden, um Instanzen der Zeichenkette entweder in Aufwärtsrichtung oder in Abwärtsrichtung jeweils zu finden. Weitere Informationen über diese Operationen und andere, die über das Textansichtsdialogfenster **146** vorgesehen sind, sind nachfolgend in Tabelle 5 angegeben.

Tabelle 5 – Textansichtsdialogfenster

Name	Typ	Min	Max	Standard	Inhalt
Suchen	Schaltfläche	n/a	n/a	aktiviert	Erlaubt dem Benutzer die Eingabe einer Suchzeichenkette
Abwärtssuche	Schaltfläche	n/a	n/a	deaktiviert	Findet die nächste Instanz der Suchzeichenkette
Aufwärtssuche	Schaltfläche	n/a	n/a	deaktiviert	Findet eine vorherige Instanz der Suchzeichenkette

[0091] An diesem Punkt ist es sinnvoll, im Detail die Art und Weise zu beschreiben, in der die in dem Dialogfenster **146** dargelegte Textinformation durch das VCAT-System **98** aus den Daten, welche diese darstellen, die in der Versionskontrolldatenbank **102** gespeichert sind, erzeugt wird. Es versteht sich, daß die Art und Weise, in der die Versionskontrolldaten gespeichert werden, für die praktische Ausführung von bestimmten Aspekten der vorliegenden Erfindung nicht wesentlich ist. In einer Ausführungsform der vorliegenden Erfindung werden die Versionskontrolldaten jedoch vorzugsweise in der Versionskontrolldatenbank **102** in einem auf Dateien basierenden Format gespeichert. In einer alternativen Ausführungsform werden die Daten in einer objektorientierten Weise gespeichert, so daß die Versionskontrolldatenbank eine objektorientierte Datenbank umfaßt.

[0092] Die Versionskontrolldatenbank **102** ist vorzugsweise unter Verwendung der DeltaV™-Database Administrator-Anwendung verwaltet, die von Fisher-Rosemont Inc. zur Verfügung gestellt wird und in dem erforderlichen Ausmaß modifiziert wurde, um die beiden Datenbanken **100**, **102** handzuhaben. Alternativ kann der Microsoft SQL Server® Enterprise Manager für diesen Zweck verwendet werden.

[0093] Um die Textinformation zu erzeugen, welche die Konfigurationsversion darstellt, führt das VCAT-System **98** eine Routine aus, die allgemein auf die Versionskontrolldatenbank **102** zugreift, um die zugehörigen Daten in einer Weise zu exportieren, die in ein Format entweder auf Text- oder Grafikbasis übersetzt werden

können. Zu diesem Zweck speichert während eines Einbuchungsvorganges das VCAT-System **98** eine auf Text basierende Darstellung der Versionskontrolldaten in einer Datei in Übereinstimmung mit einer Auszeichnungssprache, wie zum Beispiel XML (Extensible Markup Language). Der in dem XML-Dokument enthaltene Text, der an diesem Punkt erzeugt wird, kann zu einem einzelnen Zeichenstring gereiht werden, der in der Versionskontrolldatenbank **102** gespeichert wird. Insbesondere kann in einer Ausführungsform jedes versionsfähige Element einen Datenbankeintrag haben, der jeder Konfigurationsversion entspricht. In diesem Fall hat jeder Konfigurationsversionseintrag ein Feld, das dafür ausgelegt ist, daß eine einzelne Zeichenkette bzw. ein einzelner Zeichenstring aus XML-Text darin gespeichert wird, der die Versionskontrolldaten, die zu dieser Konfigurationsversion gehören, darstellt. Vorzugsweise bilden diese Konfigurationsversionseinträge eine Tabelle einer Vielzahl von Tabellen in der Versionskontrolldatenbank **102**, bei der es sich in diesem Fall um eine relationale Datenbank handelt. Die relationale Datenbank kann andere Tabellen enthalten, die auf das Speichern der folgenden Tatsachen gerichtet sind: 1.) ob jedes versionsfähige Element gelöscht ist, das Identifizierungskennzeichen der aktuellen Version, ob das Element gegenwärtig ausgebucht ist, und, sofern zutreffend, an wen; 2.) und die Überwachungsprotokollinformation für jedes versionsfähige Element.

[0094] In einer bevorzugten Ausführungsform werden alle Konfigurationsdaten, die zum vollständigen Definieren einer Konfiguration eines Elements erforderlich sind, für jede Version separat gespeichert. Alternativ können die Konfigurationsdaten in einer Weise gespeichert werden, die nur die Unterschiede zwischen Versionen definiert, wobei in diesem Fall die Daten, die mit mehreren Datenbankfeldern verbunden sind, abgefragt werden müssen, um das XML-Dokument für eine bestimmte Version zu entwickeln.

[0095] Die Versionskontrolldaten werden dadurch in strukturierter Weise dargelegt, die ohne weiteres manipuliert und analysiert werden kann, wenn später auf die Versionskontrolldatenbank **102** zugegriffen wird. Um die in dem Beispiel von [Fig. 11](#) gezeigte Konfigurationsinformation darzustellen, interpretiert das VCAT-System **98** die XML-Darstellung der Versionskontrolldaten, um die Konfigurationsinformationen in einem leicht lesbaren Textformat wiederzugeben. Die Art und Weise, in der die XML-formatierten Daten verarbeitet werden, sind dem Durchschnittsfachmann bekannt, werden jedoch nachfolgend im Detail beschrieben.

[0096] Zur Zeit des Einreichens der vorliegenden Anmeldung ist XML eine dem Durchschnittsfachmann bekannte Auszeichnungssprache, und man geht davon aus, daß gegenwärtig die Standardisierung als XML 1.0 durch das World Wide Web-Konsortium (www.w3.org) im vollem Gange ist. Während andere Datenschemata verwendet werden können, ist die Verwendung einer beliebigen aus einer Anzahl von Auszeichnungssprachen bevorzugt, um die Erzeugung sowohl von Text- als auch Grafikdarstellungen der Konfiguration des Elements zu erleichtern.

[0097] Allgemein ausgedrückt hat ein in XML ausgeführtes Dokument eine logische Struktur, die aus Deklarationen, Kommentaren und Verarbeitungsanweisungen sowie aus Knoten, die mit einem einzelnen Rootelement (oder Dokumentenelement) beginnen, besteht. Beispielsweise kann ein versionsfähiges Element ein Anfangselement bzw. Rootelement "MODUL" haben, um einen Abschnitt einer Steuerstrategie darzustellen. Dieses MODUL-Element kann einen Attributknoten mit der Bezeichnung NAME enthalten, welcher einen Textknoten hat, welcher den Namen des Moduls enthält. Zusätzlich kann das Modul Elementknoten zur Beschreibung des Moduls enthalten. Diese Elemente können beispielsweise ALGORITHMUS_TYP, BESCHREIBUNG, AUSFÜHRUNGSGESCHWINDIGKEIT und MODUL_TYP einschließen. Jedes dieser Elemente kann einen Textknoten haben, der einen Wert aufrechterhält, der diesen bestimmten Abschnitt der Konfiguration darstellt. Der MODUL-Elementknoten kann ferner Elementknoten enthalten, die zusätzliche Elemente enthalten, so daß dadurch eine Hierarchie geschaffen wird. Beispielsweise kann ein MODUL einen SCHRITT enthalten und der SCHRITT kann ein oder mehrere AKTION-Elemente enthalten. Die AKTION-Elemente können wiederum Elemente wie zum Beispiel NAME, BESCHREIBUNG, AKTION_TYP und dergleichen enthalten.

[0098] Somit stellt das XML-Format die Versionskontrolldaten mit der Hilfe von Kennzeichen in ähnlicher Weise wie bei der in Hypertext Markup Language(HTML)-Dokumenten gefundenen Weise fest. In XML-Dokumenten können die Kennzeichen jedoch an die Art der zu speichernden Daten wie vorstehend beschrieben angepasst werden. Beispielsweise ist der Funktionsblock, für den die Textdarstellung vorstehend dargestellt wurde, nachfolgend im XML-Format angegeben:

```

<Funktions_Block>
  <Name->AI1</Name>
  <Definition->AI</Definition>
  <Beschreibung>Analoge Eingabe</Beschreibung>
  <Links>200</Links>
  <Oben>200</Oben>
  <Höhe>125</Höhe>
  <Breite>400</Breite>
</Funktions_Block>

```

[0099] Auf die in einem XML-Dokument gespeicherten Daten wird in Übereinstimmung mit einem Objektmodell zugegriffen, das das Analysieren des Dokuments vorsieht, um eine Datenbaumstruktur zu schaffen, die eine Vielzahl von Knoten hat, die zu den Versionskontrolldaten gehören. In der praktischen Ausführung der vorliegenden Erfindung kann ein beliebiges von verschiedenen Objektmodellen verwendet werden, wie zum Beispiel das Dokumentenobjektmodell in seiner Anwendung auf XML.

[0100] Die Textinformation, die in dem Textansichtsdialogfenster **146** über die Benutzerschnittstelle **94** dargestellt wird, wird erzeugt, indem der XML-Text in das Datenbankfeld geladen wird, um die vorstehend beschriebenen Objektknoten für die Konfigurationsversion zu erstellen. Das Dokumentenobjektmodell kann dann verwendet werden, um die von den Objektknoten geschaffene Datenstruktur zu durchlaufen, um die Konfigurationsinformation zu extrahieren und das in [Fig. 11](#) gezeigte Textformat für die ausgewählte Version und das ausgewählte Element zu erzeugen.

[0101] Das Grafikansichtsdialogfenster **150** aus [Fig. 12](#) wird ebenfalls aus der vorstehend beschriebenen Routine auf XML-Basis erzeugt. Wie dem Durchschnittsfachmann ohne weiteres verständlich ist, kann das gleiche XML-Dokument, das zum Erzeugen der Textinformation verwendet wird, in der Lage sein, eine grafische Darstellung zu unterstützen, bei Vorhandensein der erforderlichen Objekte in der Baumstruktur zusammen mit dem Wissen, wie bestimmte Objekte darzustellen sind. Bei einer begrenzten Anzahl von verfügbaren Objekttypen wird dieses Wissen durch das VCAT-System **98** zur Verfügung gestellt. Wenn beispielsweise das XML-Dokument angibt, daß ein Element ein Funktionsblock ist, kann das VCAT-System **98** eine Routine zur Darstellung einer allgemeinen Zeichnung eines Funktionsblockes auslösen und dabei erneut auf die in dem XML-Dokument vorgesehenen Daten Bezug nehmen, um den Funktionsblock in Übereinstimmung mit anderen Parametern zu zeichnen.

[0102] Es versteht sich, daß nicht jedes Element grafisch dargestellt werden kann. Elemente, die grafisch dargestellt werden können, schließen jedoch ohne Einschränkung diejenigen ein, die allgemein durch einen Funktionsblock oder einen sequentiellen Flußdiagrammalgorithmus definiert sind.

[0103] Das Grafikansichtsdialogfenster **150**, das aus der Auswahl des Benutzers einer Konfigurationsversion in dem Überwachungsprotokolldialogfenster **140** resultiert, bietet einem Benutzer verschiedene Optionen, um das Prüfen der Konfiguration zu erleichtern. Insbesondere enthält das Dialogfenster **150** eine Textansichtschaltfläche **206**, die es dem Benutzer erlaubt, die Konfigurationsinformation, die in dem Rahmen **152** dargestellt wird, in einem Textformat in Übereinstimmung mit vorstehend beschriebener Vorgehensweise darzustellen. Die Textansicht-Schaltfläche **206** kann auch auf ein bestimmtes untergeordnetes Element, das im Rahmen **152** dargestellt wird, gerichtet sein. Im einzelnen kann, nachdem ein Benutzer das untergeordnete Element ausgewählt hat, die Textansicht-Schaltfläche **206** betätigt werden, um die Konfigurationsinformation des untergeordneten Elements im Textformat zu betrachten. Alternativ kann das untergeordnete Element durch Betätigen der rechten Maustaste oder dergleichen ausgewählt werden, um ein Kontextmenü **207** zu erzeugen, welches den Textansicht-Task als auszuwählendes Element anbietet.

[0104] Das Dialogfenster **150** enthält ferner eine Schaltfläche **208** zum Anzeigen eines übergeordneten Elements und eine Drilldown-Schaltfläche **210**. Die Drilldown-Schaltfläche **210** bietet dem Benutzer die Option, die Konfigurationsinformation für ein untergeordnetes Element (im möglichen Umfang) des ursprünglich ausgewählten Elements in einem grafischen Format darzustellen. Sobald ein Element (wie zum Beispiel ein Verbundfunktionsblock **212**), das in dem Rahmen **152** dargestellt ist, von dem Benutzer ausgewählt wird, wird die Drilldown-Schaltfläche **210** aktiviert, so daß sie von einem Benutzer mit einer Zeigeeinrichtung oder derglei-

chen ausgewählt werden kann. Die Schaltfläche **210** kann dann gewählt werden, um tiefer in das untergeordnete Element einzudringen, um die grafische Konfigurationsinformation in Bezug darauf darzustellen. Der Drill-down-Task kann auch unter Verwendung des Kontextmenüs **207** ausgelöst werden.

[0105] Die Schaltfläche **208** zum Darstellen von übergeordneten Elementen gibt dem Benutzer die Option, zu einer grafischen Ansicht des übergeordneten Elements zurückzukehren, indem die Ansicht des untergeordneten Elements entfernt wird. Alternativ kann das untergeordnete Element in einem separaten Dialogfenster angezeigt werden, so daß die Anzeigeoperation für das übergeordnete Element das VCAT-System **98** veranlasst, das zu dem untergeordneten Element gehörende Dialogfenster zu schließen und dadurch zur grafischen Ansicht des übergeordneten Elements zurückzukehren.

[0106] In einer alternativen Ausführungsform wird dann, wenn keine grafische Ansicht für ein untergeordnetes Element zur Verfügung steht (da es beispielsweise keinen Verbundfunktionsblock oder kein Modul bildet), ein Textansicht-Dialogfenster erzeugt. Es versteht sich, daß schließlich der Vorgang des weiteren Vordringens in untergeordnete Elemente in einem Grafikanzeige-Dialogfenster zu einem Textansicht-Dialogfenster führt.

[0107] Es sei angemerkt, daß die Verwendung eines XML-Dokuments, das als Zwischenstufe zwischen der Versionskontrolldatenbank **102** und der Erzeugung der Benutzerschnittstelle **94** direkt während des kontinuierlichen Betriebes erstellt wird, eine effiziente, flexible Vorgehensweise für die Präsentation von Konfigurationsinformationen darstellt. Wichtiger ist jedoch, daß das Erzeugen eines XML-Dokuments für jede Konfigurationsversion auch den raschen und effizienten Vergleich von zwei Versionen eines ausgewählten Elements bietet.

[0108] Das Vergleichen von Versionen eines Elements ist im allgemeinen Umfeld der Versionskontrolle nützlich, um die allgemein so bezeichneten "Unterschiede" zwischen Versionen zu bestimmen. Hinsichtlich des vorstehend dargelegten Prozeßsteuersystems **10** wäre es vorteilhaft, die Unterschiede zwischen zwei Versionen eines Elements sowohl in Textform als auch grafisch darzustellen. In einer bevorzugten Ausführungsform erzeugt das VCAT-System **98** Unterschiede-Dialogfenster sowohl für die Textdarstellung als auch für die grafische Darstellung der Unterschiede zwischen zwei Versionen unter Verwendung der vorstehend beschriebenen Verarbeitung der in der Versionskontrolldatenbank **102** gespeicherten Daten auf XML-Basis. Da die Baumstruktur eines XML-Dokuments gemäß dem Objektmodell ohne weiteres analysiert werden kann, können zwei Versionen eines Elements ohne weiteres verglichen werden, indem die entsprechenden beiden XML-Dokumente analysiert werden und die analysierten Daten Objekt für Objekt verglichen werden.

[0109] [Fig. 16](#) zeigt ein Unterschiede-Dialogfenster **220** im Textformat, das einen ersten Rahmen **222** und einen zweiten Rahmen **224** zum Darstellen von Konfigurationsinformationen für eine erste Konfigurationsversion und eine zweite Konfigurationsversion hat. Sowohl das horizontale als auch das vertikale Scrollen der Textinformation, die in jedem Rahmen **222**, **224** angezeigt ist, ist unter Verwendung bekannter Fensterarbeitstechniken vorgesehen. Der erste Rahmen **222** kann die Konfigurationsinformation für eine ältere Version (beispielsweise Version Nummer 13) darstellen, während der zweite Rahmen die Konfigurationsinformation für eine jüngere Version (beispielsweise Version 14) darstellen kann. Es sei jedoch angemerkt, daß die Versionen nicht aufeinanderfolgend numeriert sein müssen, da ja zwei beliebige Versionen durch das VCAT-System **98** verglichen werden können. Als ältere Version kann der erste Rahmen **222** eine oder mehrere Textzeilen enthalten, die als Resultat einer Ausbuchungs-/Einbuchungsmodifikation der Konfiguration gelöscht wurden. Vorzugsweise ist der zu derartigen gelöschten Zeilen gehörige Text in einer anderen Farbe (beispielsweise blau), einem anderen Zeichensatz oder einem anderen Zeichenstil als der übrige Text dargestellt. Um anzugeben, daß der Text in blauer Farbe beispielsweise eine gelöschte Zeile darstellt, kann eine Schaltfläche **226** den Text "gelöschte Zeilen" in blauer Farbe enthalten. In ähnlicher Weise kann der Rahmen **224** eine oder mehrere Textzeilen enthalten, die als Resultat einer Modifikation an der Konfiguration des Elements eingefügt wurden. Derartige Textzeilen können beispielsweise in grüner Farbe dargestellt werden, ebenso wie eine Schaltfläche **228** mit dem grünfarbigen Text "eingefügter Text". Schließlich kann der Inhalt einer oder mehrere Textzeilen zwischen den Versionen geändert worden sein (jedoch nicht vollständig gelöscht). Derartige Zeilen können in roter Farbe in beiden Rahmen **222**, **224** zusammen mit einer Schaltfläche **230** dargestellt werden, welche den rotfarbigen Text "veränderter Text" hat. Es versteht sich, daß jede Farbe oder jedes Stilschema verwendet werden kann, um die vorstehend bezeichneten Arten von Unterschieden zu unterscheiden und den veränderten Text gegenüber dem Text, der beiden Versionen gemeinsam ist, abzuheben.

[0110] Das Unterschiede-Dialogfenster **220** in Textdarstellung enthält ferner eine Suchen-Schaltfläche **232**, eine Abwärtssuche-Schaltfläche **234**, eine Aufwärtssuche-Schaltfläche **236**, eine Abwärts-Schaltfläche **238**, eine Aufwärts-Schaltfläche **240**, welche dieselben Navigationsfunktionen bieten, die vorstehend unter Bezug auf das Textansicht-Dialogfenster **146** beschrieben wurden. Diese Navigationstools sind vorzugsweise auf bei-

de Rahmen **222**, **224** anwendbar, so daß jeder vertikale Scrollvorgang, der von dem Benutzer ausgelöst wird, zu dem Scrollen der beiden Rahmen führt. In einer alternativen Ausführungsform können diese Navigations-tools (sowie andere Standard-Scrolltechniken in einer Fensterumgebung) auf einen der Rahmen **222**, **224** angewendet werden, und zwar beispielsweise darauf basierend, welcher Rahmen von einem Benutzer ausgewählt wurde.

[0111] Die Textinformation für die Versionen, die verglichen werden, muss nicht nebeneinander dargestellt werden. Beispielsweise kann das textliche Unterschiede-Dialogfenster alternativ einen einzelnen Rahmen enthalten, in dem der gemeinsame Text, der veränderte Text, die eingefügten Zeilen und die gelöschten Zeilen dargestellt sind und durch ein ähnliches Farb- oder Stilschema unterschieden sind. Unterschiede zwischen den beiden Versionen können auch in diesem Kontext durch Verwendung von roter Unterstreichung, Unterstreichung, abgrenzender Zeichensetzung und dergleichen dargestellt werden.

[0112] Ein grafisches Unterschiede-Dialogfenster **250** ist in [Fig. 17](#) dargestellt. Das grafische Unterschiede-Dialogfenster **250** basiert auf einem Fenster mit einem Rahmen ähnlich dem grafischen Ansichtsdialogfenster **150**, kann alternativ jedoch auch zwei Rahmen nebeneinander im Fenster verwenden, um die Versionen gegenüberzustellen. Um zwischen gemeinsamen Objekten, gelöschten Objekten, hinzugefügten Objekten und veränderten Objekten zu unterscheiden, kann auch hier ein Farbschema verwendet werden. Die Farben können als Hintergrundfarben der Objekte, als Umrandung, als Mattierung, als Umriß etc. angewandt werden. Schaltflächen **252**, **254** und **256** sind als Tasten für das Farbschema in derselben Weise wie vorstehend in Verbindung mit dem Unterschiede-Dialogfenster **220** in Textdarstellung dargelegt ausgeführt. Beispielsweise sind ein Schrittelelement S1 und ein Übergangselement T1 mit einer farbigen Begrenzung (zum Beispiel rot) dargestellt, um anzugeben, daß die Elemente modifiziert wurden. Ein weiteres Schrittelelement S2 und ein weiteres Übergangselement T2 sind mit einer Begrenzung in anderer Farbe (zum Beispiel blau) dargestellt, um anzuzeigen, daß diese Elemente gelöscht oder entfernt wurden.

[0113] Es sei angemerkt, daß das Farbschema auch auf die Linien angewandt werden kann, die die Elemente verbinden.

[0114] Beispiele für Veränderungen an Elementen, die durch eine farbige Umrißlinie bzw. einen farbigen Rahmen dargestellt werden, schließen das Hinzufügen einer Aktion zu einem Schritt und das Verändern des Ausdrucks eines Übergangs ein. Veränderungen an der Ausführungsreihenfolge der Funktionsblöcke können durch Einfassen in Umrißlinien bzw. Umrahmen etc. nur des Abschnitts des Objekts, der sich auf die Darstellung der Ausführungsreihenfolge bezieht, gekennzeichnet werden. Dieser Abschnitt kann beispielsweise unten an einem Objekt angeordnet sein. In dem Fall, daß ein Objekt zwischen Versionen neu benannt wurde, kann das Farbschema verwendet werden, um die beiden Versionen des Objekts als gelöscht und hinzugefügt darzustellen. Alternativ kann das VCAT-System **98** dem Benutzer die Möglichkeit bieten, nur wesentliche Unterschiede zwischen Versionen darzustellen, so daß kosmetische Veränderungen nicht dargestellt werden.

[0115] Das grafische Unterschiede-Dialogfenster **250** enthält ferner eine Textanzeigeschaltfläche **258**, die dieselbe Funktion wie die Schaltfläche **206** hinsichtlich sowohl des ausgewählten Elements als auch jedes untergeordneten Elements bietet.

[0116] Wenn ein untergeordnetes Element als modifiziert dargestellt wird, bietet das VCAT-System **98** dem Benutzer die Möglichkeit, durch eine entsprechende Auswahl des Elements in das Element einzudringen, um entweder ein Unterschiede-Dialogfenster in Textdarstellung oder ein weiteres grafisches Unterschiede Dialogfenster für das untergeordnete Element zu erzeugen. In einer Ausführungsform ist das tiefere Eindringen nur bei denjenigen Elementen möglich, für welche ein grafisches Dialogfenster erzeugt werden kann. Im allgemeinen ist jedoch die Art des erzeugten Fensters von der Art des Elements wie vorstehend erläutert abhängig. Das tiefere Eindringen in ein untergeordnetes Element kann auch durch Auswählen desselben und anschließendes Auswählen einer Drilldown-Schaltfläche **260** ausgelöst werden. Das Zurückkehren zu dem vorherigen grafischen Unterschiede-Dialogfenster **250** kann durch Auswählen einer Schaltfläche **262** zum Anzeigen des übergeordneten Elements geschehen, die in derselben Weise wie die Schaltfläche **208** arbeitet.

[0117] Wenn ein Element, das entfernt wurde, an derselben Position in dem grafischen Unterschiede-Dialogfenster ist wie ein Element, das hinzugefügt wurde, kann eines der Elemente verdeckt sein. Wenn daneben ein Kommentar verändert wird, ist der alte Kommentar in der grafischen Darstellung nicht sichtbar. Um es dem Benutzer zu ermöglichen, ein verstecktes Element oder einen versteckten Kommentar zu sehen, kann das VCAT-System **98** einen Mechanismus vorsehen (über einen Menüpunkt oder eine Auswahlsequenz unter Verwendung einer Zeigeeinrichtung oder dergleichen), um eine Umschaltmöglichkeit zu bieten, welches Element

oder welcher Kommentar dargestellt (das heißt in der obersten Ebene ist) und welches Element verdeckt wird.

[0118] In einer bevorzugten Ausführungsform enthalten die vorstehend beschriebenen Textansichts- und Grafikantritts-Dialogfenster einschließlich derjenigen, die auf das Betrachten von Unterschieden gerichtet sind, eine Schaltfläche oder eine Taskelementsequenz, die es einem Benutzer erlaubt, zwischen der Text- und der Grafikanzeige umzuschalten. Das VCAT-System **98** enthält vorzugsweise diese Funktionalitäten nur bei Dialogfenstern für Elemente, die in beiden Formaten dargestellt werden können.

[0119] Es sei angemerkt, daß das VCAT-System **98** eine Sicherheitsprüfung über die Versionskontrollumgebung legen kann. Beispielsweise kann das VCAT-System **98** bei einer Operation (beispielsweise Rückzug), die eine bestimmte Ebene oder Art der Autorisierung erfordert, vor der Durchführung der Operation bestimmen, ob der Benutzer zum Auslösen der Operation autorisiert ist.

[0120] Wenn ein neuer Benutzer zu dem VCAT-System **98** hinzugefügt wird, kann das VCAT-System **98** ein Dialogfenster für einen neuen Benutzer (nicht dargestellt) erzeugen, das eine Vielzahl von Kontrollkästchen hat, die einer Vielzahl von VCAT Operationen entsprechen, die ausgewählt oder nicht ausgewählt werden können, um dem Benutzer das gewünschte Ausmaß an Autorisierung zu verleihen.

[0121] Es versteht sich ferner, daß das VCAT-System **98** einem Benutzer, der die entsprechende Autorisierung hat, die Option des Aktivierens und des Deaktivierens des VCAT-Systems **98** bieten kann. Wenn es deaktiviert ist, würde das VCAT-System **98** das Arbeiten der Konfigurationsanwendungen **96** ermöglichen, ohne Einbuchung/Ausbuchung und andere Vorgänge aufzuerlegen. Wenn es erneut aktiviert ist, führt das VCAT-System **98** vorzugsweise eine Synchronisierungsroutine durch, welche auf die Konfigurationsdatenbank **100** zugreift, um die gegenwärtigen Konfigurationsdaten mit den letzten in der Versionskontrolldatenbank **102** gespeicherten Konfigurationsdaten zu vergleichen. Für jedes Element, bei dem eine Konfigurationsmodifikation während der Zeitdauer, während der das VCAT-System **98** deaktiviert war, auftrat, wird eine Konfigurationsversion hinzugefügt, und Daten, welche die gegenwärtige Version in der Konfigurationsdatenbank **100** darstellen, werden in Verbindung damit gespeichert. Neue Elemente können auch in der Versionskontrolldatenbank **102** im erforderlichen Ausmaß erzeugt werden. Ferner kann ein Kennzeichen allen Elementen in der Versionskontrolldatenbank **102** zugewiesen werden, um anzuzeigen, daß das VCAT-System **98** wieder aktiviert wurde.

[0122] Die Synchronisierungsroutine kann auch zu jeder anderen Zeit ausgeführt werden. In bestimmten Fällen sollte beachtet werden, daß das VCAT-System **98** möglicherweise einige Ausbuchungsoperationen und/oder das Rückgängigmachen von einigen Ausbuchungsoperationen in dem erforderlichen Ausmaß zur Modifikation der Versionskontrolldatenbank **102** ausführen muss.

[0123] Das VCAT-System **98** kann ferner eine Datenbanksicherungs/Wiederherstellungsroutine ähnlich der Routine enthalten, die zur Sicherung und Wiederherstellung der Konfigurationsdatenbank **100** verwendet wird, welche von dem Explorersystem verwaltet wird. Andere allgemeine Datenbankhilfsprogramme, wie etwa eine "Datenbank reinigen"-Routine, die auf das Wiederherstellen der Datenspeicherstruktur der Versionskontrolldatenbank **102** gerichtet ist, können auch als Teil des VCAT-Systems **98** enthalten sein.

[0124] Die Benutzerschnittstelle **94** ist vorzugsweise nicht die einzige Ausgabeeinrichtung, mit der die Versionskontrollinformation den Benutzern und Bedienungspersonen des VCAT-Systems **98** und der Konfigurationsanwendungen **96** dargeboten wird. Insbesondere können eine oder mehrere Routinen von dem VCAT-System **98** implementiert werden, um die Erzeugung von Berichten zur Übertragung an einen Drucker oder eine andere Anzeigeeinrichtung zu unterstützen. Beispielsweise kann das VCAT-System **98** dem Benutzer die Möglichkeit bieten, Taskelemente auszuwählen, die auf das Erzeugen von Berichten gerichtet sind, welche Versionskontrollinformationen darstellen, die sich darauf beziehen, welche Elemente ausgebucht sind, das Prüfprotokoll eines bestimmten Elements, die von einem bestimmten Benutzer ausgebuchten Elemente, jegliche gelöschte (jedoch nicht entfernte) Elemente und eine Liste von Ausbuchungen nach Datum oder einem anderen Parameter, für welche Daten in der Versionskontrolldatenbank **102** gespeichert sein können. Derartige Informationen können über jede Anzeigeeinrichtung abgegeben werden, sei es lokal oder entfernt angeordnet, und können in einem beliebigen Format bzw. in Übereinstimmung mit einem beliebigen Protokoll abgegeben werden.

[0125] Um das Erzeugen derartiger Berichte zu erleichtern, enthält das VCAT-System **98** vorzugsweise ein Abfragesystem. Das Abfragesystem erlaubt allgemein dem Benutzer, Suchkriterien anzugeben, die auf eine beliebige Anzahl von Gegenständen, wie zum Beispiel Modifikationen oder Aktionen durch einen bestimmten Benutzer, Modifikationen oder Versionskontrollereignisse, die während eines angegebenen Zeitrahmens auf-

traten, Modifikationen oder Versionskontrollereignisse, die in einer bestimmten Version oder einem bestimmten Kennzeichen auftraten, und Modifikationen oder andere Versionskontrollereignisse, die sich auf ein bestimmtes Element oder einen Elementbereich beziehen, gerichtet sind. Nachdem der Benutzer eines oder mehrere Kriterien für eine Abfrage eingegeben hat, löst das VCAT-System **98** eine Suchroutine aus, welche auf die Versionskontrolldatenbank **102** zugreift und den Inhalt entsprechend analysiert. In einer bevorzugten Ausführungsform führt das VCAT-System **98** eine Suchroutine im Hintergrund aus, so daß andere Versionskontrolloperationen gleichzeitig eingeleitet und ausgeführt werden können.

[0126] Das Abfragesystem kann eines oder mehrere Dialogfenster zur Anzeige über die Benutzerschnittstelle **94** erzeugen. Beispiele von zwei solchen Dialogfenstern sind in [Fig. 18](#) und [Fig. 19](#) gezeigt. Ein Dialogfenster **280** für die Optionen eines Historienberichts bzw. eines Archivberichts kann durch Auswählen eines Elements und anschließendes Auswählen eines Taskelements, das darauf gerichtet ist, entweder über ein Drop-Down-Menü oder ein Kontextmenü erzeugt werden. Das Dialogfenster **280** hat Kontrollkästchen, die darauf gerichtet sind, ob der zu erstellende Historienbericht Versionskontrollinformationen enthält, welche Merkmale und untergeordnete Elemente betreffen. Ein Benutzerfeld **282** ist ebenfalls vorgesehen, um es zu ermöglichen, daß der Historienbericht auf Modifikationen oder Versionskontrollereignisse gerichtet ist, die von einem bestimmten Benutzer ausgelöst wurden. Schließlich kann ein Feld **284** "von Datum" und ein Feld **286** "bis Datum" verwendet werden, um eine bestimmte Zeitperiode für den Historienbericht anzugeben. Um das Eingeben des Anfangs- und Enddatums für die Zeitperiode zu erleichtern, können ein Popup-Kalender oder Beispieldaten in einem Fenster, das für jedes der Felder **284**, **286** erzeugt wird, gemäß bekannten Fensterprogrammtechniken vorgesehen sein.

[0127] Wie [Fig. 19](#) zeigt, kann das Abfragesystem ein allgemeines Such-Dialogfenster **290** erzeugen, das einen Statussuchabschnitt **292** und einen Suchbereichsabschnitt **294** zur Angabe bestimmter Suchkriterien enthält. Der Statussuchabschnitt **292** enthält Kontrollkästchen, um das VCAT-System **98** anzuweisen, alle ausgebuchten Elemente oder von einem bestimmten Benutzer ausgebuchten Elemente zu suchen. Der Benutzer kann über ein Benutzerfeld **296** eingegeben werden, das in der Lage ist, Benutzernamen über ein Drop-Down-Fenster vorzuschlagen. Der Suchbereichsabschnitt **294** enthält mehrere Kontrollkästchen, die es dem Benutzer erlauben, die Versionskontrollinformation zu suchen, die sich nur auf das ausgewählte Element, das ausgewählte Element und alle untergeordneten Elemente, oder alle Elemente in der Versionskontrolldatenbank **102** beziehen.

[0128] Die Bildschirmanzeigen von [Fig. 5](#) bis [Fig. 19](#) sowie weitere Bildschirmanzeigen können unter Verwendung eines Fensterformats mit Standardbefehlen für grafische Bildschirmoberflächen unter Verwendung von Fenstern erzeugt und modifiziert werden, obgleich auch jedes andere Format verwendet werden kann. Das Format dieser Bildschirmanzeigen kann drastisch verändert werden, wenn beispielsweise das VCAT-System **98** in Verbindung mit anderen Konfigurationsanwendungen als dem Explorersystem oder den anderen vorstehend genannten Anwendungen verwendet wird.

[0129] Obgleich die Konfigurationsanwendungen **96** und das VCAT-System **98**, die hier beschrieben wurden, vorzugsweise in einer oder mehreren Softwareroutinen implementiert werden, können sie in einer Routine implementiert sein, die in Hardware, Firmware etc. ausgeführt ist, und können durch jeden anderen Prozessor, der mit dem Prozeßsteuersystem **10** verbunden ist, implementiert werden. Somit können alle vorstehend beschriebenen Operationen und Vorgänge in einer Standard-Mehrzweck-CPU implementiert werden oder auf speziell gestalteter Hardware oder Firmware nach Wunsch vorliegen. Bei der Implementierung in Software kann die Softwareroutine in jedem computerlesbaren Speicher, wie beispielsweise auf einer Magnetplatte, einer Laserplatte oder einem anderen Speichermedium, in einem RAM oder ROM eines Computers oder Prozessors und dergleichen gespeichert werden. Entsprechend kann die Software zu einem Benutzer oder einem Prozeßsteuersystem über jedes bekannte oder gewünschte Lieferverfahren geliefert werden, darunter beispielsweise auf einer computerlesbaren Platte oder einem anderen transportablen Computerspeichermechanismus oder über einen Kommunikationskanal, wie etwa eine Telefonleitung, das Internet etc. (was als gleich oder austauschbar mit dem Liefern dieser Software über ein transportables Speichermedium betrachtet wird).

Patentansprüche

1. Prozeßsteuersystem, das enthält:
 ein computerlesbares Medium (**20**);
 einen Prozessor (**18**), der in Kommunikation mit dem computerlesbaren Medium (**20**) steht;
 eine erste Datenbank (**100**), die erste Daten speichert, die eine erste Konfiguration des Prozesses mit einer Vielzahl von Prozesselementen (**40**, **42**, **44**, **46**, **48**, **50**, **51**, **52**, **55**) darstellen, wobei jedes Prozesselement

(**40, 42, 44, 46, 48, 50, 51, 52, 55**) eine entsprechende erste Elementkonfiguration hat; eine zweite Datenbank (**102**), die zweite Daten speichert, die eine zweite Konfiguration des Prozesses mit der Vielzahl von Prozesselementen (**40, 42, 44, 46, 48, 50, 51, 52, 55**) darstellen, wobei jedes Prozesselement (**40, 42, 44, 46, 48, 50, 51, 52, 55**) eine entsprechende zweite Elementkonfiguration hat; eine Konfigurationsroutine (**96**), die in dem computerlesbaren Medium (**20**) gespeichert ist und so konfiguriert ist, dass sie von dem Prozessor (**18**) ausführbar ist, um eine Modifikation von mindestens einer der Vielzahl von ersten Elementkonfigurationen der ersten Konfiguration des Prozesses zu erleichtern, um die zweite Konfiguration des Prozesses mit der Vielzahl von zweiten Elementkonfigurationen zu erzeugen und um die Modifikationen der mindestens einen der ersten Elementkonfigurationen zur Erzeugung der zweiten Elementkonfigurationen anzuzeigen; und eine Versionskontrollroutine, die in dem computerlesbaren Medium (**20**) gespeichert ist und so konfiguriert ist, dass sie von dem Prozessor (**18**) ausführbar ist, um in der zweiten Datenbank (**102**) dritte Daten zu speichern, welche die Modifikation der mindestens einen der ersten Elementkonfigurationen zur Erzeugung der zweiten Elementkonfigurationen angeben, wobei die Versionskontrollroutine die Modifikation mindestens einer der ersten Elementkonfigurationen zur Erzeugung der zweiten Elementkonfigurationen überwacht und die dritten Daten sammelt, indem ein Ausbuchungs-/Einbuchungsvorgang über die Konfigurationsroutine (**96**) gelegt wird, so dass Konfigurationsdaten zur Verwendung in einer Laufzeit-Umgebung des Prozesses bereitgestellt werden, wobei der Ausbuchungsvorgang dazu ausgebildet ist, festzustellen, welche andere Elementkonfiguration der Vielzahl von ersten Elementkonfigurationen der ersten Konfiguration des Prozesses von einer Folgeveränderung betroffen ist und ausgebucht werden muss, und diese automatisch ausbucht, um die Modifikation zur Verwendung in der Laufzeit-Umgebung des Prozesses zu erstellen.

2. Prozesssteuersystem nach Anspruch 1, dadurch gekennzeichnet, dass der Ausbuchungs-/Einbuchungsvorgang automatisch erfolgt.

3. Prozesssteuersystem nach Anspruch 1, dadurch gekennzeichnet, dass die Modifikation in automatisierter Weise aufgezeichnet und kontrolliert wird.

4. Prozesssteuersystem nach Anspruch 1, dadurch gekennzeichnet, dass das Versionskontrollsystem (**98**) ein Vergleichstool enthält, das eine Benutzerschnittstelle (**94**) zum Anzeigen von Unterschieden zwischen einer aktuellen Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) und einer früheren Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) bietet.

5. Prozesssteuersystem nach Anspruch 4, dadurch gekennzeichnet, dass die Benutzerschnittstelle (**94**) die Unterschiede grafisch darstellt.

6. Prozesssteuersystem nach Anspruch 1, dadurch gekennzeichnet, dass das Versionskontrollsystem (**98**) ein Rückzugstool enthält, um eine aktuelle Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) durch eine frühere Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) zu ersetzen.

7. Prozesssteuersystem nach Anspruch 6, dadurch gekennzeichnet, dass das Rückzugstool automatisch feststellt, ob untergeordnete Elemente des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) gelöscht wurden.

8. Prozesssteuersystem nach Anspruch 1, ferner enthaltend eine Prozesssteuereinrichtung (**12**), auf welche die das Prozesselement (**40, 42, 44, 46, 48, 50, 51, 52, 55**) darstellenden Daten heruntergeladen werden, wobei das Versionskontrollsystem (**98**) Informationen speichert, welche die Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) angeben, um wiederzugeben, wenn die das Prozesselement (**40, 42, 44, 46, 48, 50, 51, 52, 55**) wiedergebenden Daten in die Prozesssteuereinrichtung (**12**) heruntergeladen werden.

9. Prozesssteuersystem nach Anspruch 1, ferner enthaltend eine Prozesssteuereinrichtung, in die das Prozesselement (**40, 42, 44, 46, 48, 50, 51, 52, 55**) wiedergebende Daten zur Implementierung heruntergeladen werden, wobei die Daten Informationen enthalten, die die Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**) angeben, so dass eine Prozessbedienungsperson mit Zugriff auf die Prozesssteuereinrichtung (**12**) über die Version des Prozesselements (**40, 42, 44, 46, 48, 50, 51, 52, 55**), die gegenwärtig implementiert ist, informiert sein kann.

10. Verfahren zum Kontrollieren eines Prozesses mittels eines Prozesssteuersystems, das eine Vielzahl von Prozesselementen (**40, 42, 44, 46, 48, 50, 51, 52, 55**) hat, welches Verfahren die Schritte enthält:

Erstellen und Speichern erster Daten einer Konfiguration des Prozesses, die eine erste Version von Elementkonfigurationen der Prozesselemente (**40, 42, 44, 46, 48, 50, 51, 52, 55**) hat;
Kontrollieren einer Modifikation der Konfiguration des Prozesses, um Daten für eine zweite Version von Elementkonfigurationen der Prozesselemente (**40, 42, 44, 46, 48, 50, 51, 52, 55**) zu erstellen, indem ein Ausbuchungs-/Einbuchungsvorgang implementiert wird, wobei der Ausbuchungsvorgang dazu ausgebildet ist, festzustellen, welche der anderen Elementkonfigurationen der Vielzahl von ersten Elementkonfigurationen der ersten Konfiguration des Prozesses von einer Folgeänderung betroffen sind und ausgebucht werden müssen, wobei diese dann automatisch ausgebucht werden;
Aufzeichnen von Informationen als dritte Daten, die die Modifikation der ersten Version der Elementkonfigurationen zur Erzeugung der zweiten Version der Elementkonfigurationen angeben, wobei die Modifikation überwacht und die Informationen, von dem Prozesssteuersystem gesammelt werden, indem der Ausbuchungs-/Einbuchungsvorgang entsprechend implementiert wird,
Herunterladen von Daten, die die zweite Version der Elementkonfigurationen angeben, in die Laufzeit-Umgebung des Prozesses.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass mindestens einer der Kontroll- und Aufzeichnungsschritte in automatisierter Weise durchgeführt werden.

12. Verfahren nach Anspruch 10, ferner enthaltend den Schritt des Vorsehens einer Benutzerschnittstelle (**94**) zum Anzeigen von Unterschieden zwischen der ersten Version der Elementkonfigurationen und der zweiten Version der Elementkonfigurationen.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass die Benutzerschnittstelle die Unterschiede grafisch darstellt.

14. Verfahren nach Anspruch 10, ferner enthaltend den Schritt des Speicherns von Informationen, die eine herunter geladene Version der Elementkonfigurationen bezeichnen, um wiederzugeben, wenn Daten, welche die Elementkonfigurationen wiedergeben, auf eine Prozesssteuereinrichtung (**12**) heruntergeladen werden.

Es folgen 14 Blatt Zeichnungen

Anhängende Zeichnungen

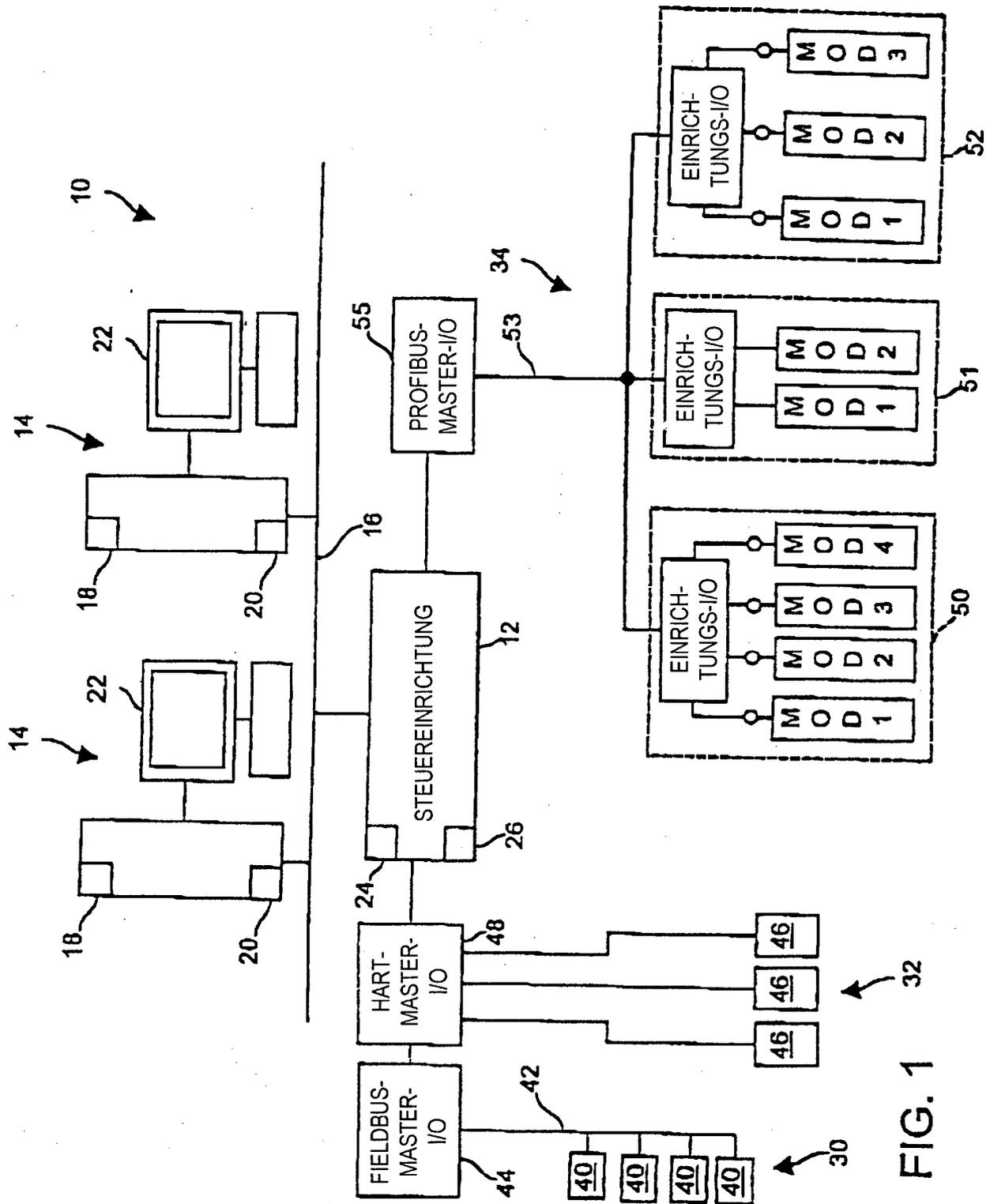


FIG. 1

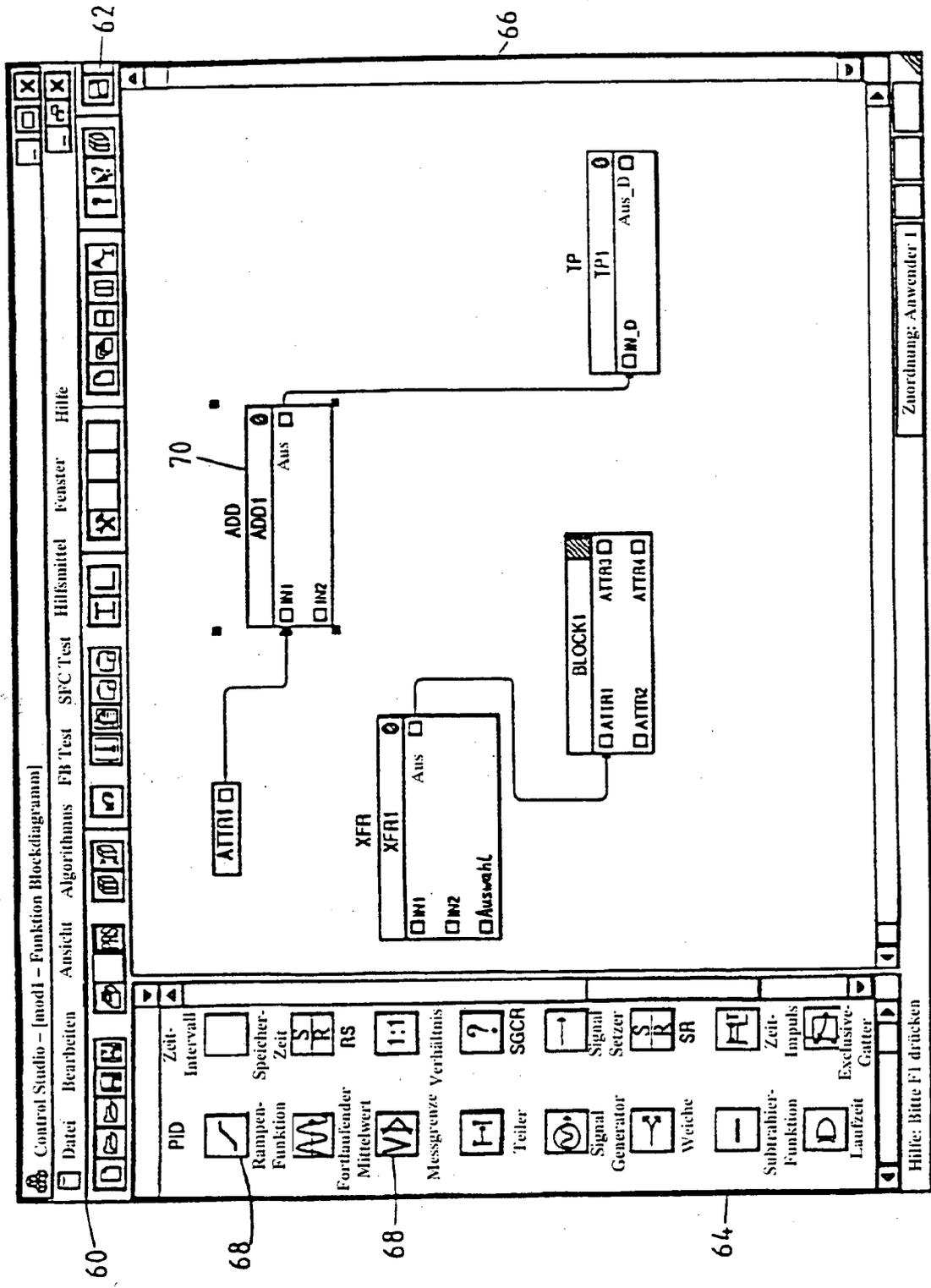
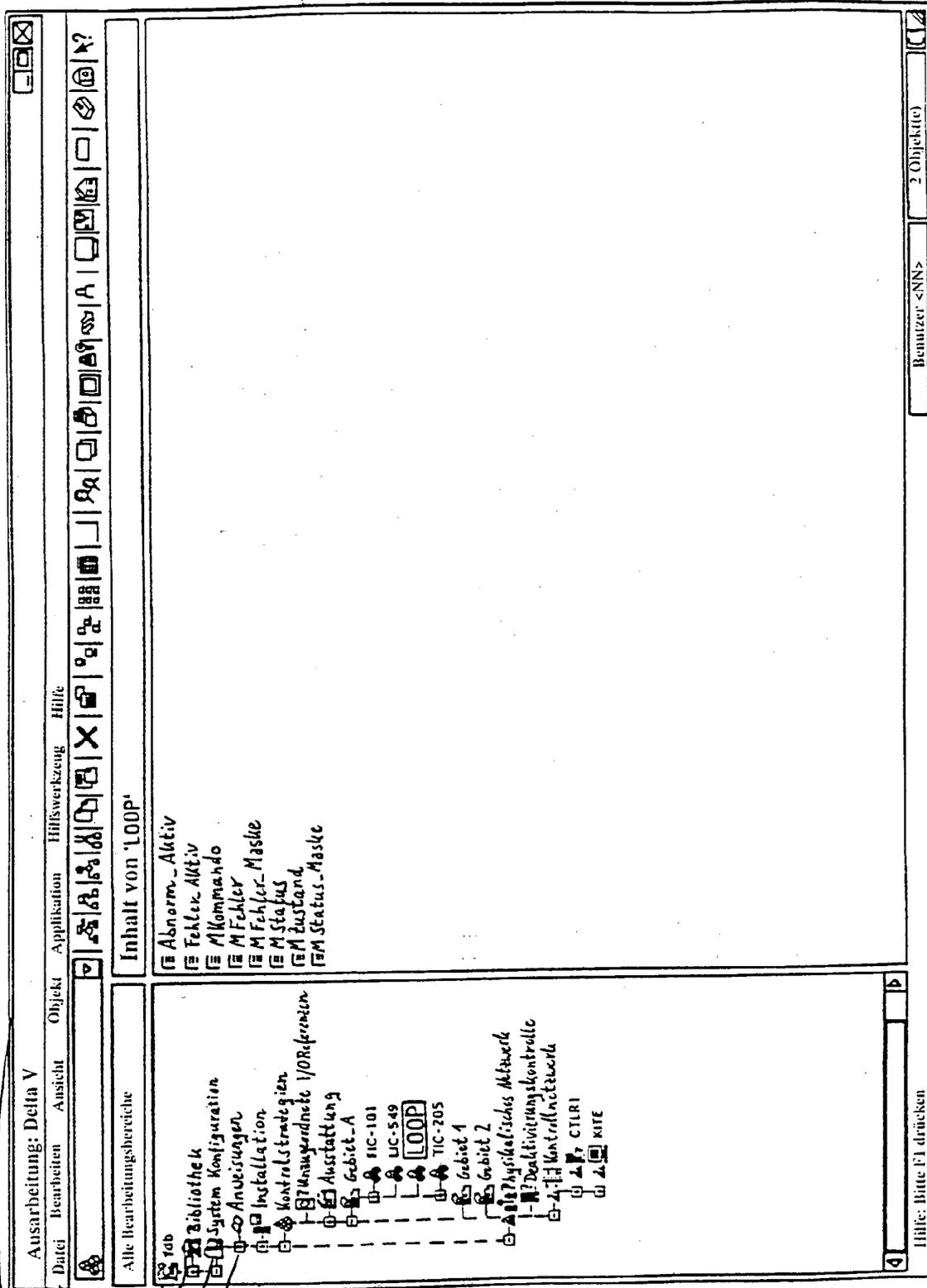


FIG. 2



80

86

88

90

92

82

84

FIG. 3

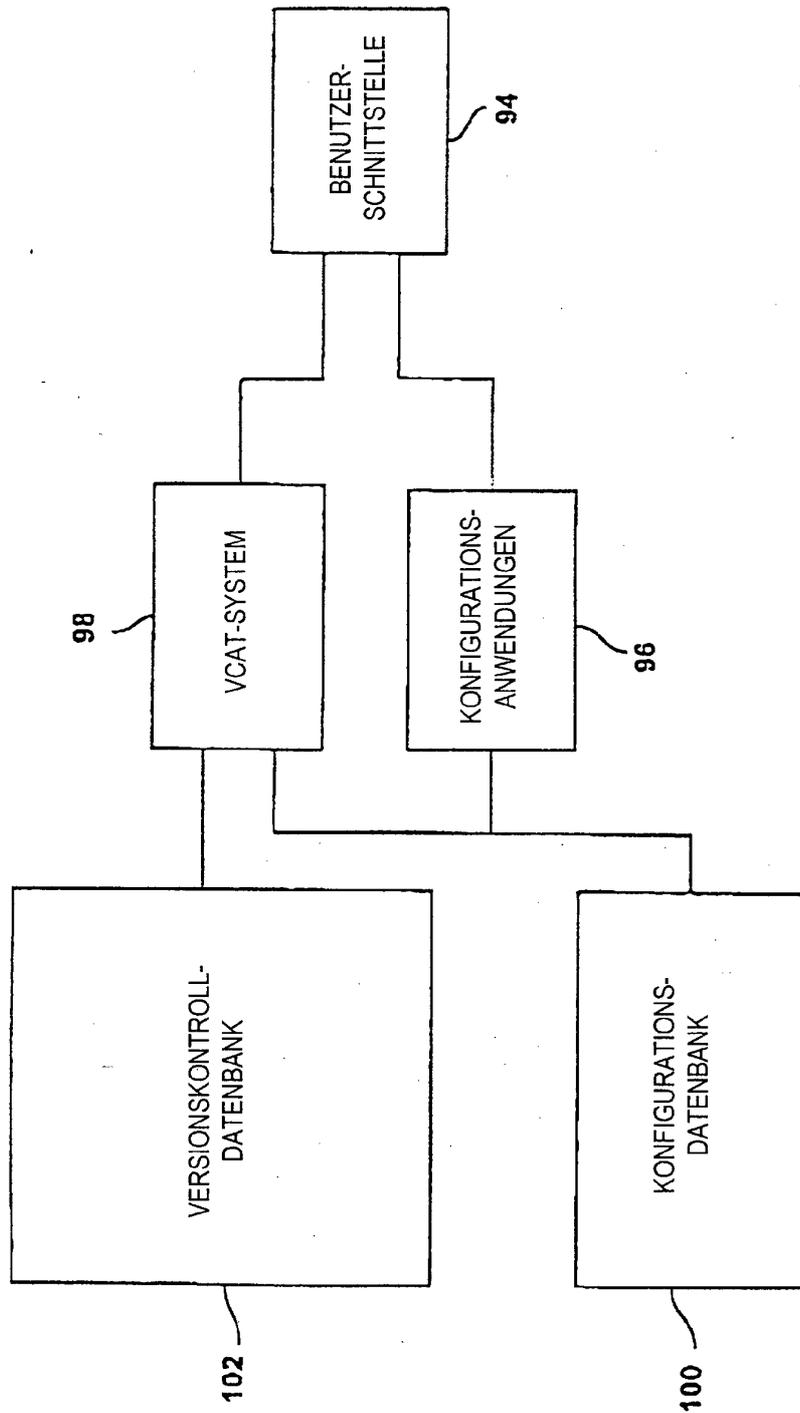
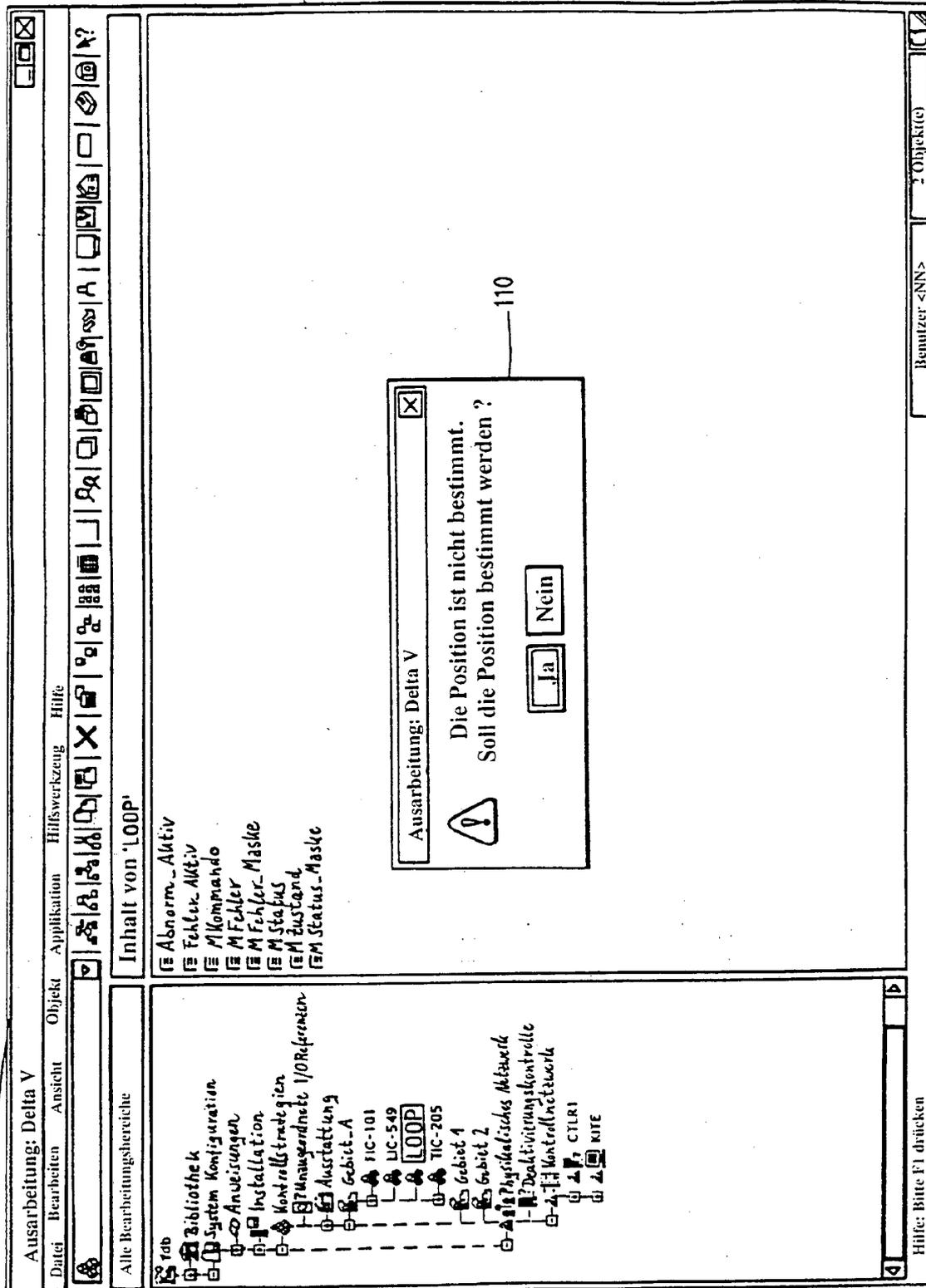


FIG. 4



80

112

82

84

110

FIG. 5

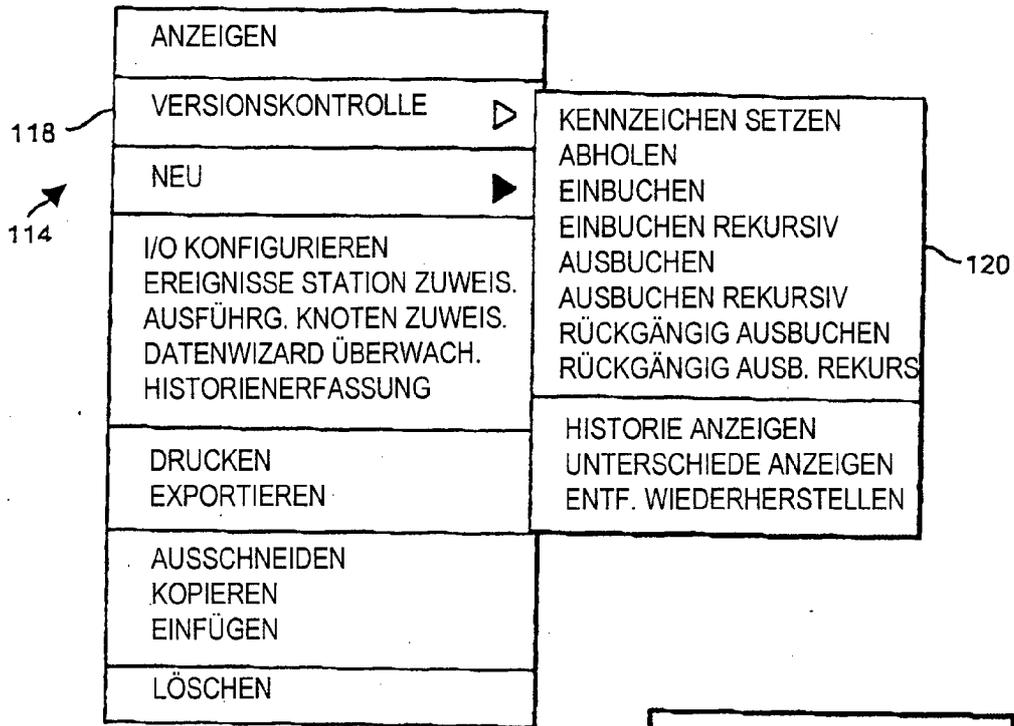


FIG. 6

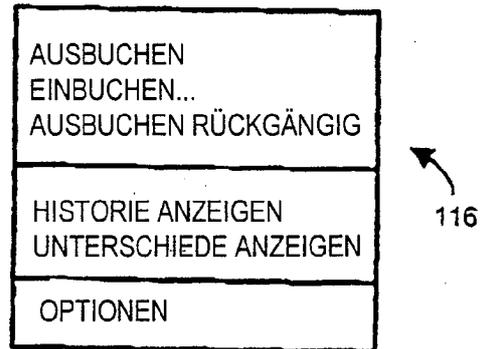


FIG. 7

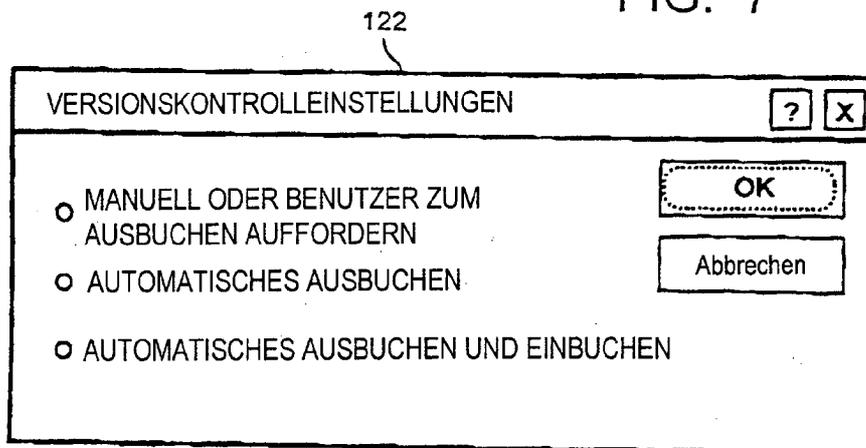


FIG. 8

FIG. 9

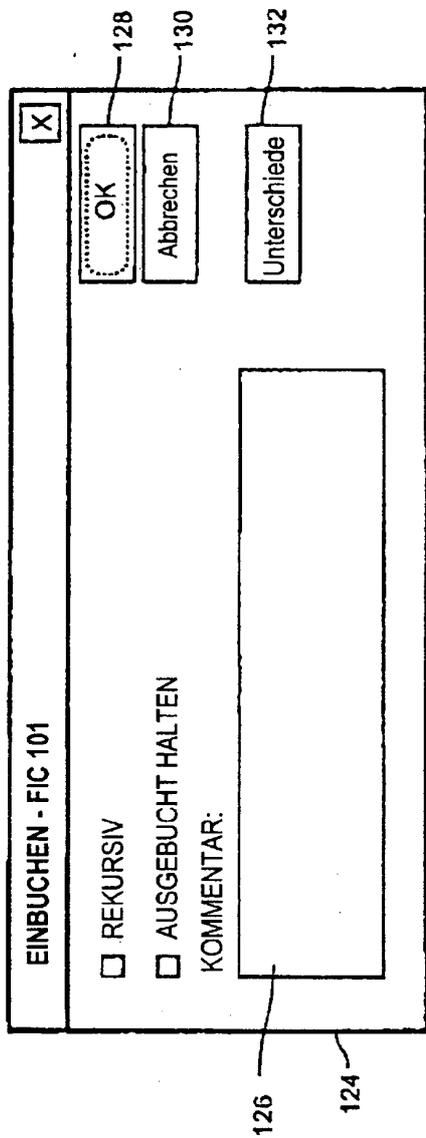
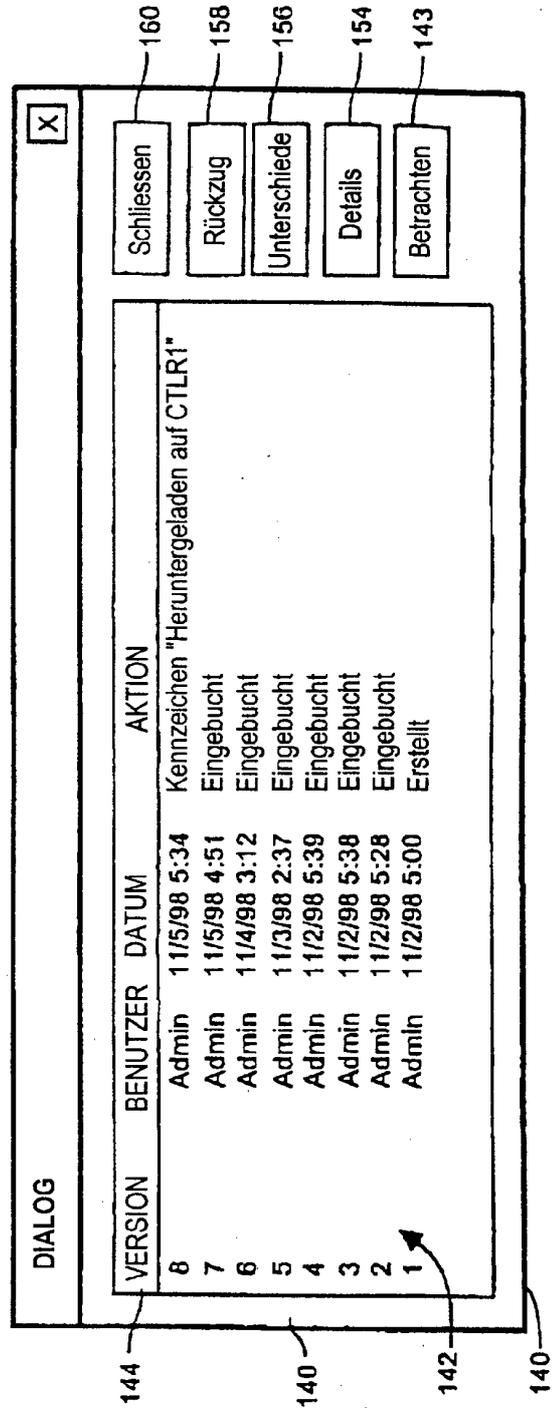


FIG. 10



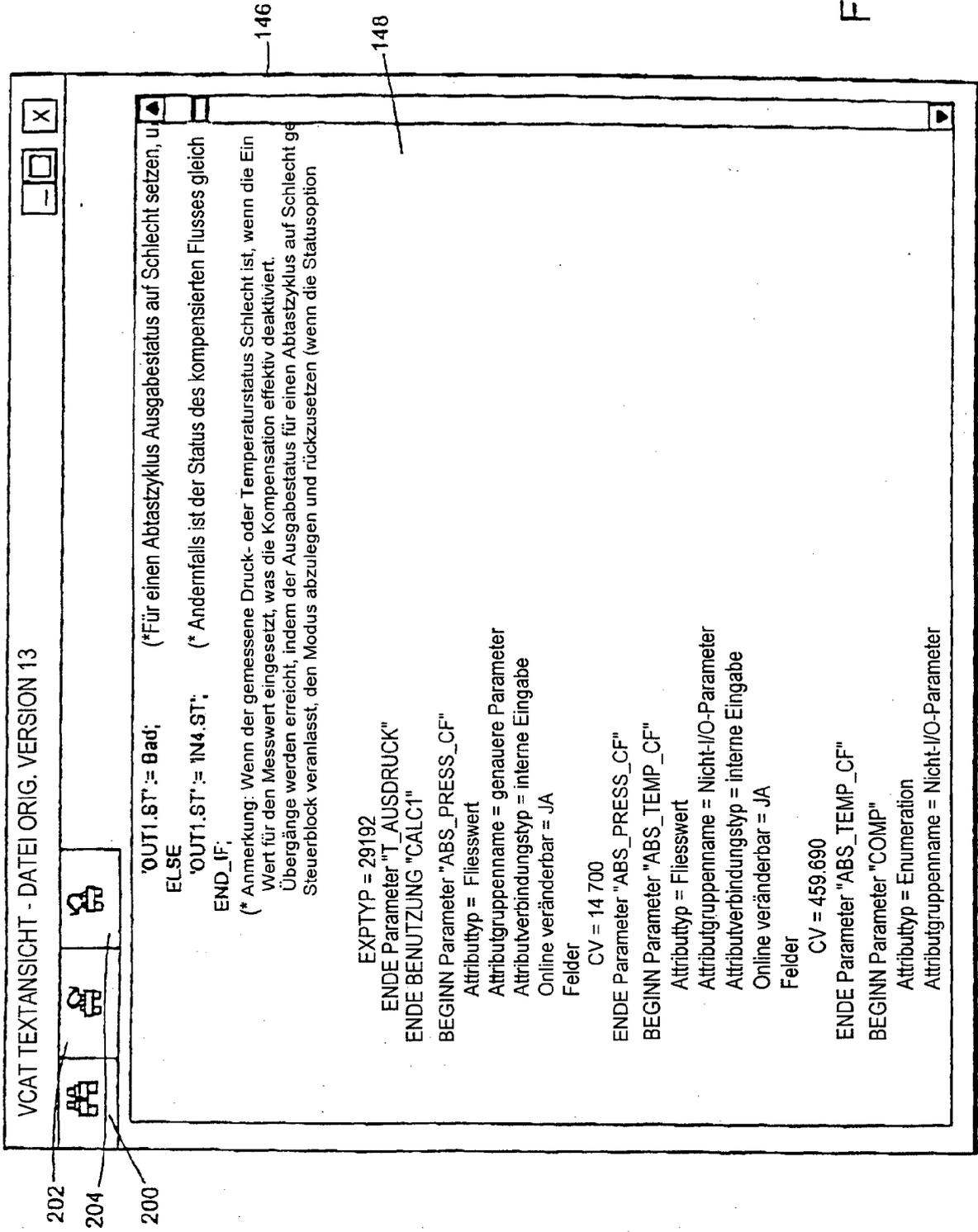
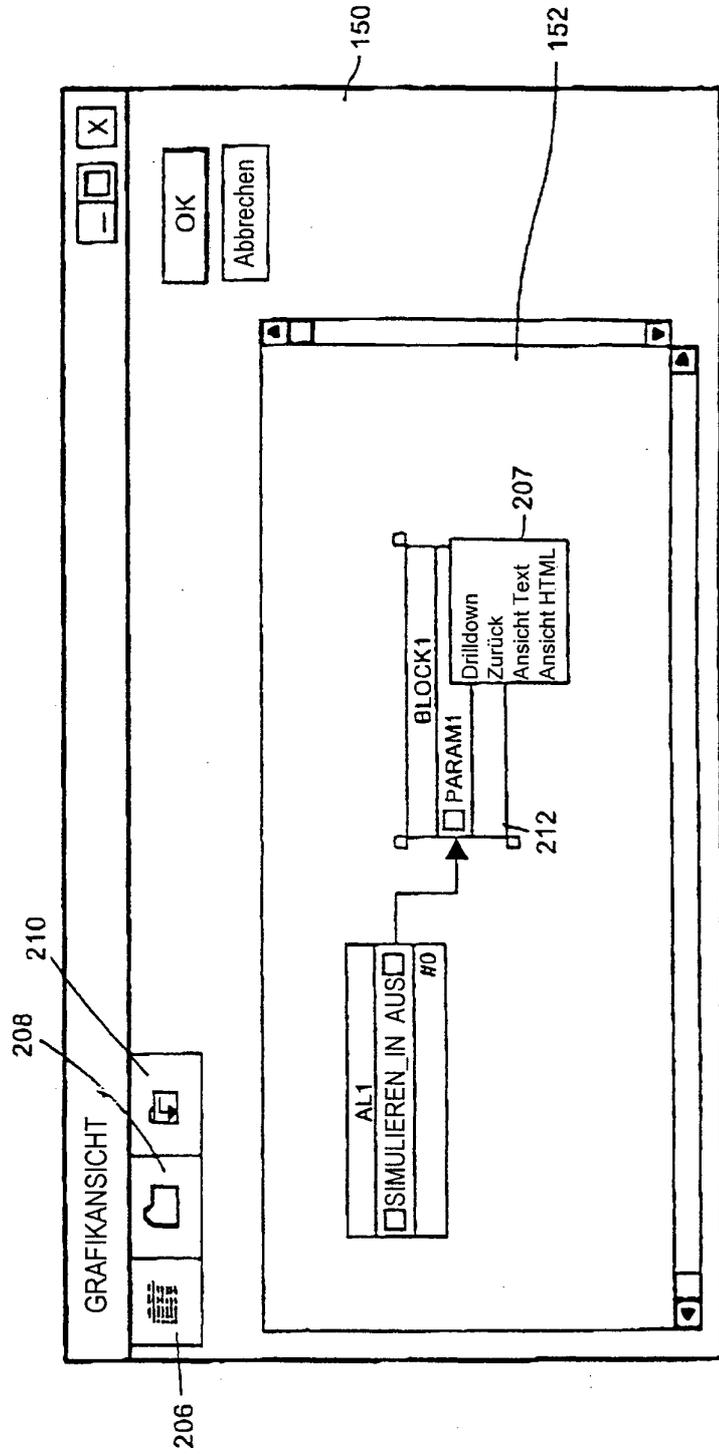


FIG. 11

FIG. 12



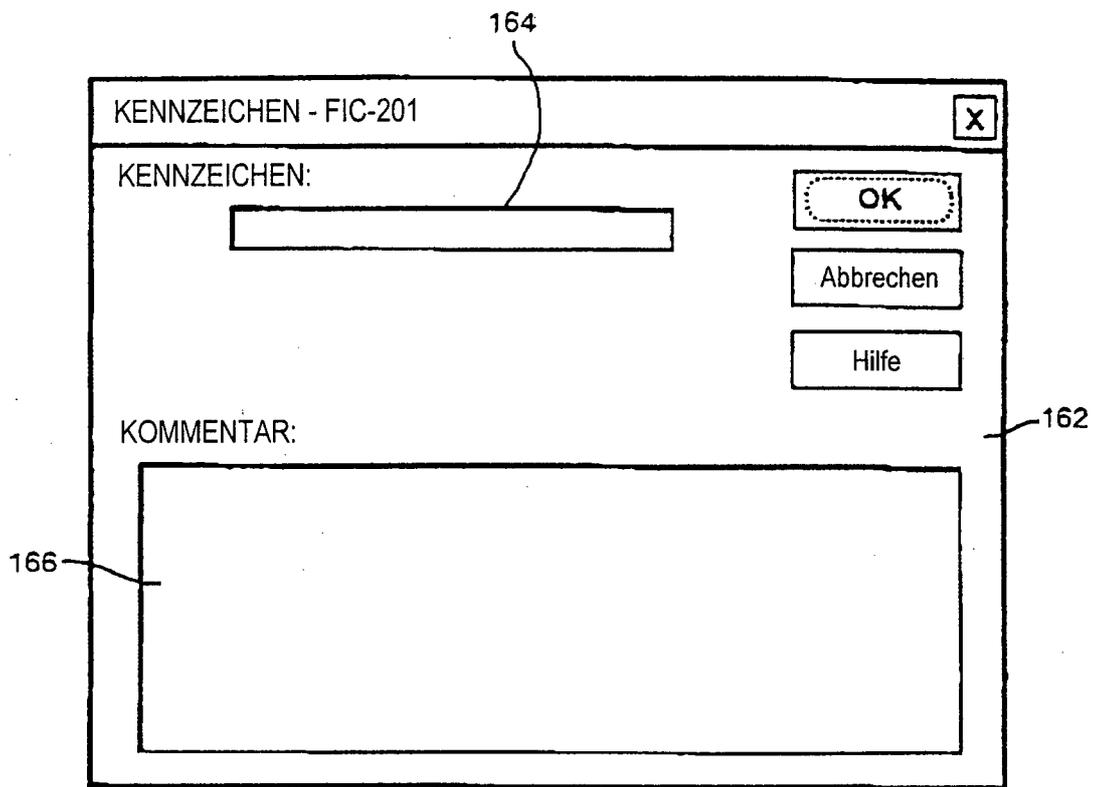


FIG. 13

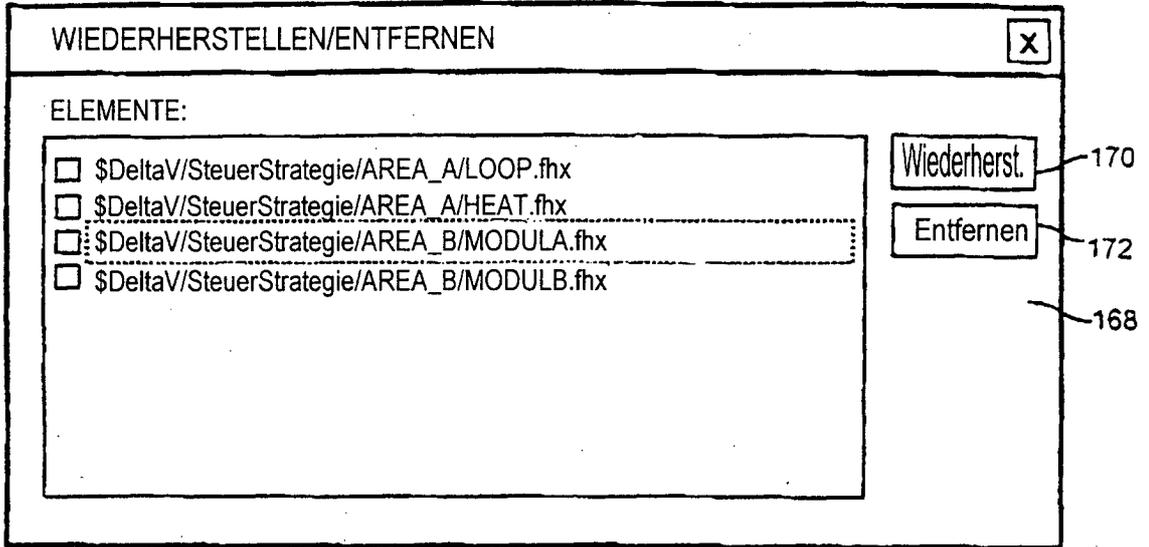


FIG. 14

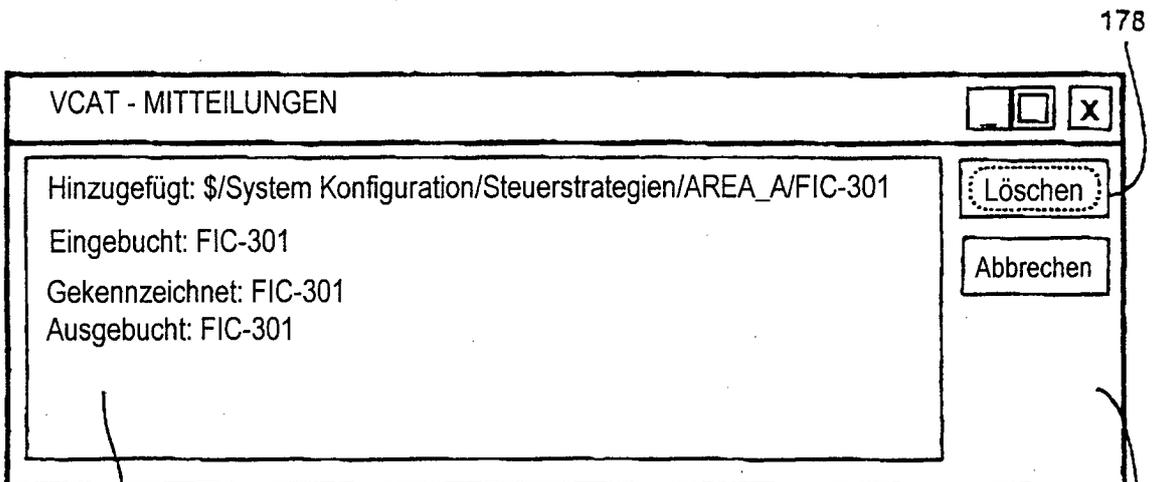


FIG. 15

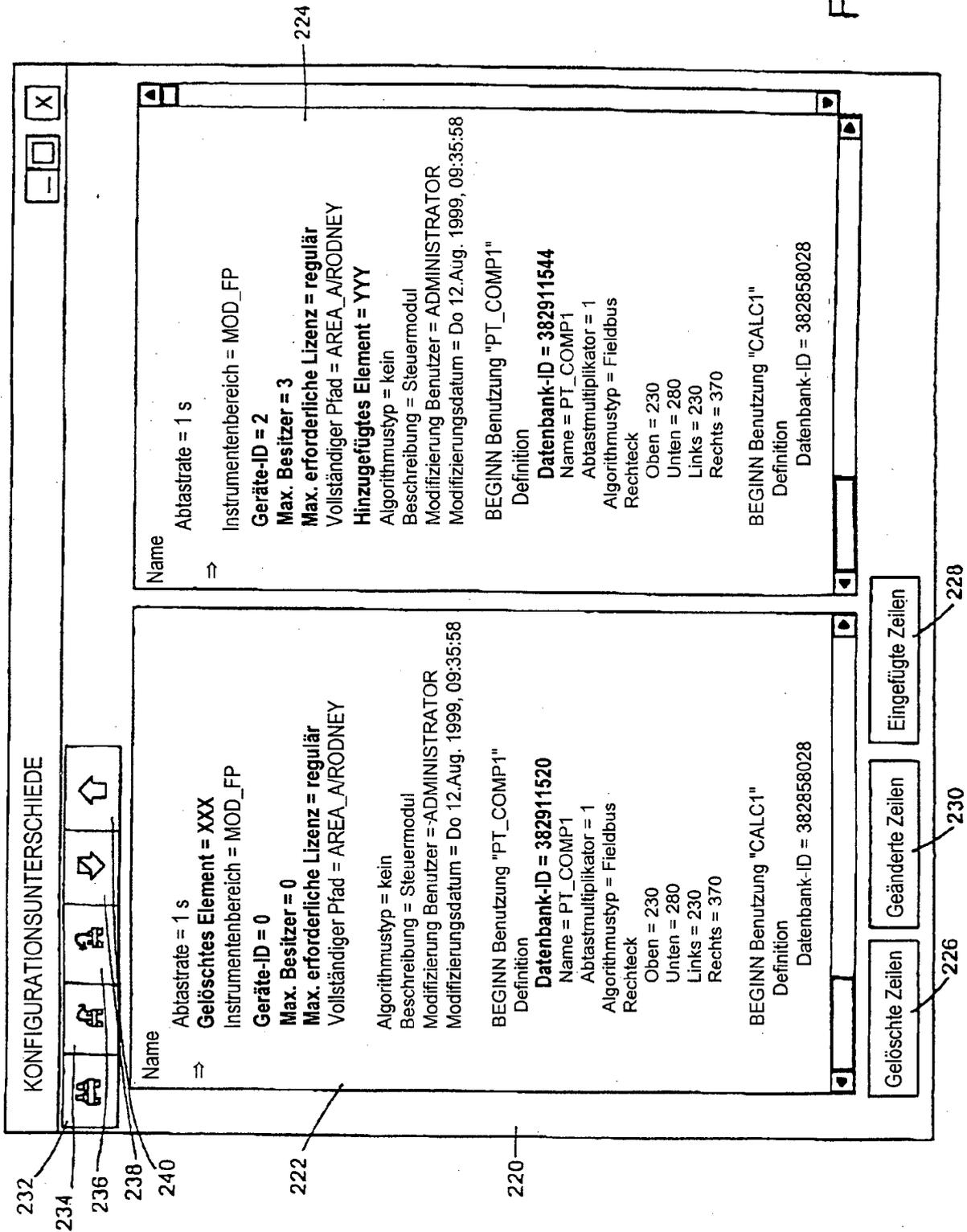
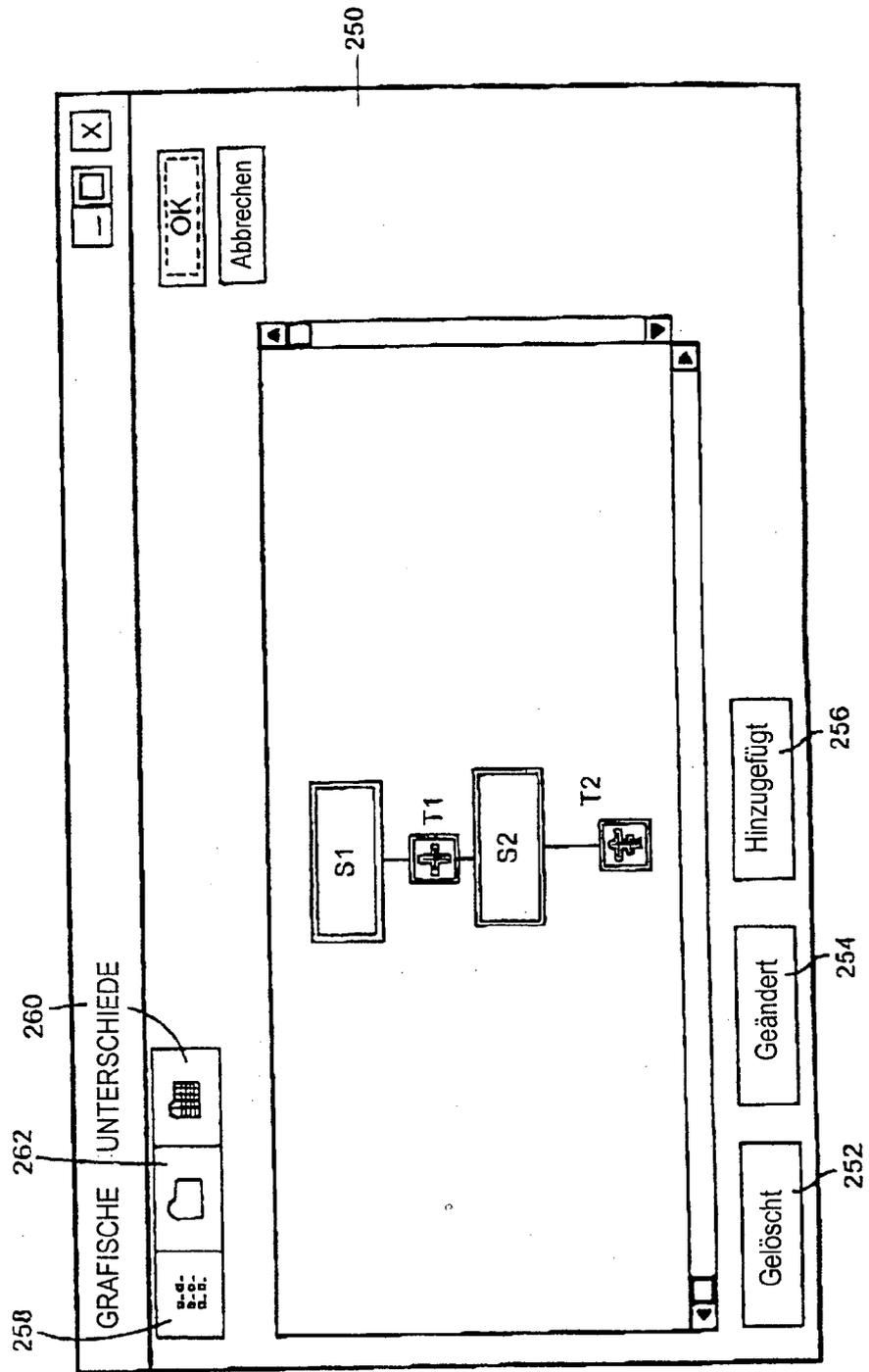


FIG. 16

FIG. 17



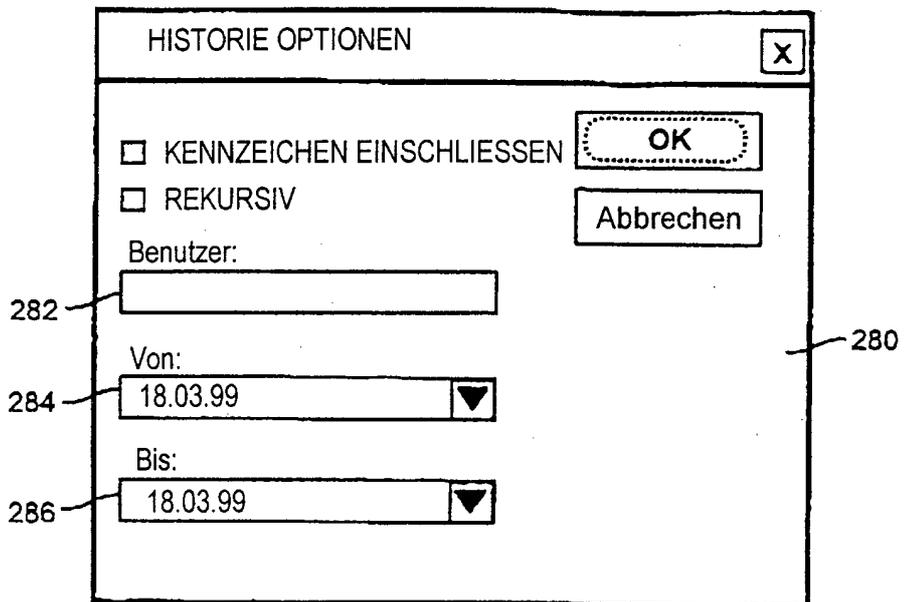


FIG. 18

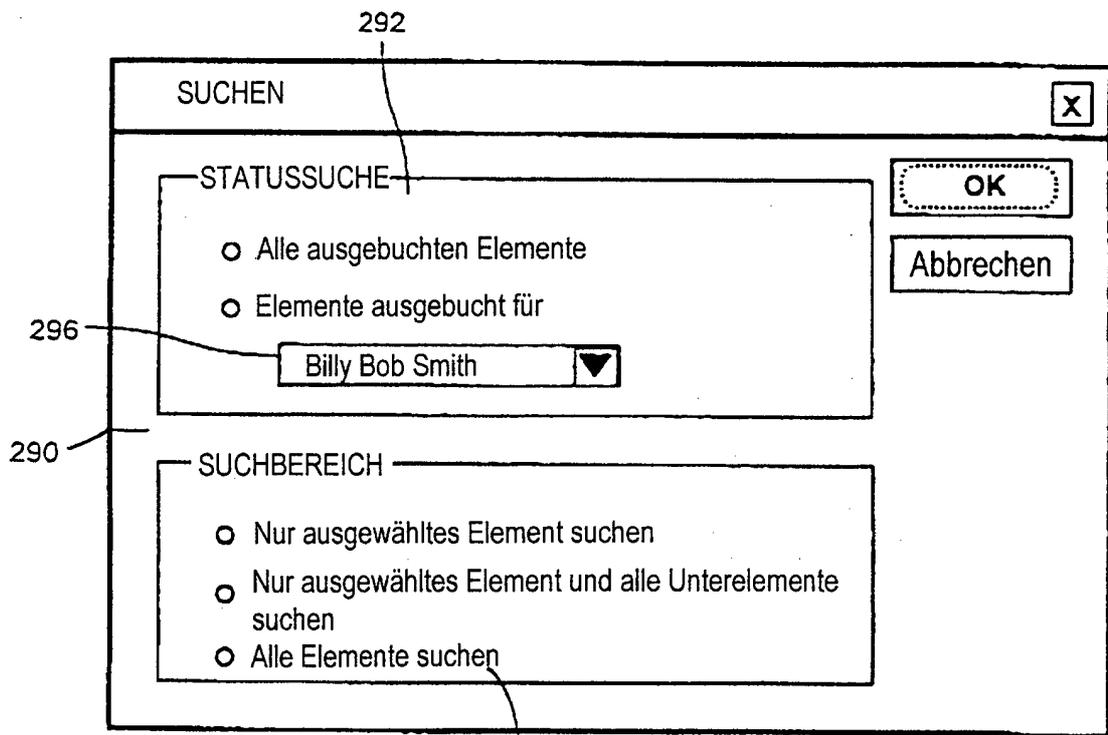


FIG. 19