



(12) **Berichtigung der Veröffentlichung**

der internationalen Anmeldung mit der
 (87) Veröffentlichungs-Nr.: **WO 2014/058900**
 in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
 (21) Deutsches Aktenzeichen: **11 2013 004 924.4**
 (86) PCT-Aktenzeichen: **PCT/US2013/063914**
 (86) PCT-Anmeldetag: **08.10.2013**
 (87) PCT-Veröffentlichungstag: **17.04.2014**
 (43) Veröffentlichungstag der PCT Anmeldung
 in deutscher Übersetzung: **02.07.2015**
 (15) Korrekturinformation:
Neue Seite 1 der Beschreibung
 (48) Veröffentlichungstag der Berichtigung: **23.07.2015**

(51) Int Cl.: **G05B 19/042 (2006.01)**
G06F 9/44 (2006.01)
G05B 19/418 (2006.01)
G05B 23/02 (2006.01)

(30) Unionspriorität:
61/711,105 **08.10.2012** **US**
61/711,110 **08.10.2012** **US**

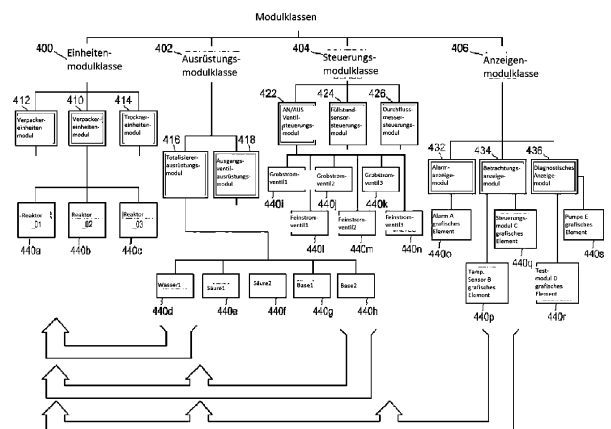
(71) Anmelder:
**Fisher-Rosemount Systems, Inc., Round Rock,
 Tex., US**

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

(72) Erfinder:
**Scott, Cindy, Georgetown, Tex., US; Firkins, Molly
 Marie, Cedar Park, Tex., US; Bellville, Keith R.,
 Round Rock, Tex., US; Jones, Bryan Michael,
 Cedar Park, Tex., US; Ott, Michael George, Austin,
 Tex., US; Marruchella, Dawn, Austin, Tex., US;
 Jundt, Larry Oscar, Round Rock, Tex., US; Lucas,
 John Michael, Leire Lutterworth, Leicestershire,
 GB; Lewis, Marty James, Cedar Park, Tex., US;
 Hammack, Stephen Gerard, Round Rock, Tex.,
 US; Gilbert, Stephen Copps, Austin, Tex., US;
 Sullivan, Matthew Joseph, Austin, Tex., US**

(54) Bezeichnung: **Dynamisch wiederverwendbare Klassen**

(57) Zusammenfassung: Techniken zur flexiblen Konfiguration einer in Betrieb befindlichen Prozessanlage oder eines Prozesssteuerungssystems ermöglichen die selektive Weitergabe einer Änderung an einem übergeordneten Objekt an untergeordnete Objekte, die von dem übergeordneten Objekt abgeleitet sind, sodass eine erste Untergruppe untergeordneter Objekte und ihre jeweiligen Instanziierungen mit der Änderung aktualisiert werden, während eine zweite Untergruppe und ihre jeweiligen Instanziierungen unverändert bleiben. Das übergeordnete Objekt kann ein Klassen- oder Instanzprozessobjekt entsprechend einer Prozessentität sein und die erste und/oder die zweite Untergruppe untergeordneter Objekte kann ausgewählt werden. In einigen Fällen kann die Änderung eine weitergegebene Änderung von einem untergeordneten Objekt, welches von der ersten oder der zweiten Untergruppe untergeordneter Objekte ausgeschlossen ist, an das übergeordnete Objekt sein. In einigen Fällen kann die Änderung zuerst zu einem von dem übergeordneten Objekt abgeleiteten Instanzobjekt weitergegeben werden und dann von dem Instanzobjekt zu einem untergeordneten Objekt weitergegeben werden. ...



Die oben angegebenen bibliographischen Daten entsprechen dem aktuellen Stand zum Zeitpunkt der Veröffentlichung dieser Berichtigung.

BeschreibungQUERVERWEISE ZU
VERWANDTEN ANWENDUNGEN

[0001] Diese Anwendung nimmt die vorläufige US-Patentanmeldung Nr. 61/711,110 mit dem Namen „Process Plant Configurations Using Flexible Objects“ in Anspruch, die am 8. Oktober 2012 eingereicht wurde und deren gesamter Inhalt hiermit durch Verweise eingegliedert wurde. Diese Anwendung nimmt darüber hinaus die vorläufige US-Anmeldung U.S. Provisional Application Nr. 61/711,105 mit dem Namen „Configurable User Displays in a Process Control System“ in Anspruch, die am 8. Oktober 2012 eingereicht wurde und deren gesamter Inhalt hiermit durch Verweis eingegliedert wurde.

[0002] Darüber hinaus ist diese Anwendung mit der US-Anmeldung Nr. 14/048,496, mit dem Namen „Method and Apparatus for Managing Process Control Configuration“ verknüpft und wird damit gemeinsam eingereicht und ihr gesamter Inhalt wurde hiermit durch Verweis eingegliedert. Darüber hinaus steht diese Anwendung mit der US-Anmeldung Nr. 14/048,516, mit dem Namen „Derived and Linked Definitions with Override“ in Verbindung und wird damit gemeinsam eingereicht und deren gesamter Inhalt wurde hiermit durch Verweis eingegliedert. Darüber hinaus steht diese Anwendung mit der US-Anmeldung Nr. 14/048,533, mit dem Namen „Dynamically Reusable Classes“ in Verbindung und wird damit gemeinsam eingereicht und ihr gesamter Inhalt wurde hiermit durch Verweis eingegliedert.

TECHNIK

[0003] Die vorliegende Erfindung bezieht sich im Allgemeinen auf verfahrenstechnische Anlagen und insbesondere auf die Verwendung von flexiblen Objekten beim Konfigurieren und Betrachten des Betriebs einer verfahrenstechnischen Anlage oder eines Prozessleitsystems.

BESCHREIBUNG DER
EINSCHLÄGIGEN TECHNIK

[0004] Verteilte Prozessleitsysteme, wie die in chemischen erdölverarbeitenden oder anderen verfahrenstechnischen Anlagen, umfassen typischerweise einen oder mehrere Prozessregler, die kommunikativ zu einem oder mehreren Feldgeräten über digitale oder analog/digital kombinierte Bus-Geräte verbunden sind. Die Feldgeräte, wie zum Beispiel Ventile, Ventilstellungsregler, Schalter und Transmitter (z. B. Temperatur, Druck, Niveau und Durchflusssensoren) befinden sich innerhalb der Prozessumgebung und führen Prozessfunktionen wie Öffnen oder Schließen der Ventile, Messen von Prozessparametern, usw. aus. Intelligente Feldgeräte, wie Feldgeräte, die mit

dem bewährten Feldbus-Protokoll übereinstimmen, sind auch in der Lage, Kontrollberechnungen, Alarmfunktionen und andere Kontrollfunktionen auszuführen, die üblicherweise im Regler eingebaut sind. Diese Prozessregler, die für gewöhnlich auch in der Anlagenumgebung eingebaut sind, empfangen Signale der Prozessmessungen, der Feldgeräte gemessen und/oder Informationen, die zu den Feldgeräten gehören und eine Regelungsanwendung ausführen, die z. B. verschiedene Steuermodule ausführt, die Prozessregelungsentscheidungen treffen, Regelsignale erzeugen, die auf erhaltenen Informationen beruhen und sich mit den Steuermodulen oder Baugruppen koordinieren, die in den Feldgeräten ausgeführt werden, wie z. B. HART und FOUNDATION® Feldbus Feldgeräte. Die Steuermodule in den Reglern senden die Steuersignale über die Kommunikationskanäle an die Feldgeräte und kontrollieren somit den Betrieb der verfahrenstechnischen Anlage.

[0005] Informationen aus den Feldgeräten und Reglern werden normalerweise über eine Datenautobahn einen oder mehreren Hardwaregeräten zur Verfügung gestellt, wie z. B. Bedienarbeitsplätze, PCs, Datenspeicher, Berichtgeneratoren, zentralisierte Datenbanken, usw., die in der Regel in Kontrollräumen oder anderen Orten, abseits der rauerer Anlagenumgebung, platziert werden. Diese Hardwaregeräte führen Anwendungen aus, die es z. B. einem Betreiber ermöglichen, Funktionen im Hinblick auf den Prozess auszuführen, wie z. B. das Ändern der Einstellungen der Prozessregelroutine, Ändern des Betriebs von Regelungsmodulen innerhalb der Regler oder der Feldgeräte, Ansicht des aktuellen Status des Prozesses, Ansicht der Alarme, die von Feldgeräten und Reglern ausgegeben wurden, Simulation des Betriebs des Prozesses um Personal zu trainieren oder die Prozessregelsoftware zu testen, Erstellen und Aktualisieren einer Konfigurationsdatenbank, usw.

[0006] Das von Emerson Process Management verkaufte DeltaV™-Steuersystem enthält mehrere Anwendungen innerhalb verschiedener Geräte, die auch von mehreren Geräten an unterschiedlichen Stellen innerhalb einer verfahrenstechnischen Anlage platziert sind. Über eine Konfigurationsanwendung, die in einer oder mehreren Bedienarbeitsstationen platziert ist, können Nutzer Prozessregelungsmodule erstellen oder ändern und diese über eine Datenautobahn an bestimmte verteilte Regler herunterzuladen. Diese Regelungsmodule basieren in der Regel auf kommunikativ verbundenen Funktionsblöcken, die Objekte in einem objektorientierten Programmierprotokoll sind und innerhalb des Steuerungsplans auf Eingaben basierenden Funktionen ausführen und Ergebnisse an andere Funktionsblöcke innerhalb des Steuerungsplans liefern. Ebenso können Konfigurationsentwickler mit Hilfe der Konfigurationsanwendung Bedienerchnittstellen erstellen oder ändern, die von einer Anzeigeanwendung

genutzt werden, um dem Bedienpersonal Daten anzuzeigen und dem Bedienpersonal zu ermöglichen, Einstellungen wie Sollwerte innerhalb der Prozesssteuerungsroutinen zu ändern. Jeder zugewiesene Regler und, in manchen Fällen, ein oder mehrere Feldgeräte, speichert und führt eine bestimmte Regelungsanwendung aus, die die zugewiesenen Steuermodi ausführt, die dorthin heruntergeladen wurden, um die tatsächliche Prozessregelungsfunktionalität zu implementieren. Die Anzeigeanwendungen, die auf einer oder mehreren Bedienarbeitsstationen ausgeführt werden können (oder auf einem oder mehreren Fernsteuerungsgeräten in kommunikativer Verbindung mit den Bedienarbeitsstationen und der Datenautobahn), erhalten Daten von der Regelungsanwendung über die Datenautobahn und zeigen diese Daten über Benutzerschnittstellen Bedienpersonal oder Nutzern an und können eine Anzahl verschiedener Ansichten, wie eine Bedienansicht, eine Ingenieursansicht, eine Technikeransicht, usw. bereitstellen. Eine Datenarchivierungsanwendung ist typischerweise auf einem Datenspeichergerät gespeichert, wird dort ausgeführt und sammelt und speichert einige oder alle Daten, die über die Datenautobahn geliefert werden, während eine Konfigurationsdatenbankanwendung auf noch einem weiteren Computer, der an die Datenautobahn angeschlossen ist, um die aktuelle Prozesssteuerungsroutinen-Konfiguration und die damit verbundenen Daten zu speichern, ausgeführt werden kann. Alternativ hierzu kann sich die Konfigurationsdatenbank am gleichen Arbeitsplatz wie die Konfigurationsanwendung befinden.

[0007] Zum gegenwärtigen Zeitpunkt enthalten Konfigurationsanwendungen in der Regel eine Bibliothek mit Vorlagenobjekten oder Elementen wie z. B. Funktionsblockvorlageobjekte sowie in manchen Fällen, Steuermodulmodul-Vorlageobjekte. Diese Konfigurationsanwendungen werden dazu genutzt, eine Regelstrategie für eine verfahrenstechnische Anlage zu konfigurieren sowie um Ansichten auf Benutzerschnittstellen einer verfahrenstechnischen Anlage bereitzustellen. Die Vorlagenobjekte haben alle Standardeigenschaften, Einstellungen und Methoden, die damit verknüpft sind. Die Entwickler, die die Konfigurationsanwendung benutzen, können diese Vorlagenobjekte wählen und im Wesentlichen Kopien der ausgewählten Vorlagenobjekte auf einem Konfigurationsbildschirm platzieren, um ein Modul, z. B. ein Steuermodul, zu entwickeln. Während der Auswahl und Platzierung der Vorlagenobjekte im Konfigurationsbildschirm verbindet der Entwickler die Ein- und Ausgänge dieser Objekte und ändert deren Parameter, Namen, Tags und andere Eigenschaften, um ein spezifisches Steuermodul für einen spezifischen Verwendungszweck in der verfahrenstechnischen Anlage zu erstellen. Nach der Erstellung einer oder mehrerer solcher Steuermodule kann der Entwickler die erstellten Module in der Bibliothek oder an

einem Konfigurationsdatenspeicherort ablegen. Der Entwickler kann die Steuermodule instanziierten (z. B. veranlassen, dass eine ausführbare Datei, die dem Regelungsmodul zugewiesen ist, erstellt wird) und es dann an den oder die entsprechenden Regler, Feldgeräte und anderen Prozesselemente zur Ausführung während des Betriebs der verfahrenstechnischen Anlage herunterladen.

[0008] Danach erstellt der Entwickler in der Regel eine oder mehrere Anzeigen für das Bedienpersonal, Wartungspersonal, usw. innerhalb der verfahrenstechnischen Anlage. Hierzu wählt und erstellt er Anzeigeelemente in einer Anzeigenerstellungsanwendung. Diese Anzeigen sind typischerweise systemweit an einem oder mehreren Arbeitsplätzen implementiert und bieten vorkonfigurierte Anzeigen für das Bedien- oder Wartungspersonal, das den Betriebszustand des Regelungssystems oder der Geräte innerhalb der Anlage kontrolliert. Diese Anzeigen haben normalerweise die Form von Alarmanzeigen, die Alarme von Reglern oder Geräten innerhalb der verfahrenstechnischen Anlage empfangen und anzeigen, die Form von Regelungsanzeigen, die den Betriebszustand der Regler und anderer Geräte, die sich innerhalb der verfahrenstechnischen Anlage befinden, anzeigen, die Form von Wartungsanzeigen, die den Funktionsstand der Geräte innerhalb der verfahrenstechnischen Anlage anzeigen, usw. Diese Anzeigen sind im Allgemeinen so vorkonfiguriert, dass sie auf bestimmte Art und Weise von den Prozesssteuerungsmodulen, Geräten oder anderen Prozesselementen innerhalb der verfahrenstechnischen Anlage erhaltene Informationen oder Daten anzeigen. In manchen Systemen werden Anzeigen über die Verwendung von Objekten erstellt, die jeweils eine Grafik mit einem physischen oder logischen Element assoziiert haben und die jeweils kommunikativ zu dem physischen oder logischen Element gebunden sind, um Daten über das physische oder logische Element zu erhalten. Ein Objekt kann die Grafik ändern, die auf dem Anzeigebildschirm angezeigt wird, je nach den zu illustrierenden Daten, wie z. B., dass ein Tank halbvoll ist oder um den Flüssigkeitsfluss, der von einem Durchflusssensor gemessen wird, anzuzeigen, usw.

[0009] Wie bei einer Regelungskonfigurationsanwendung kann eine Anzeigenerstellungsanwendung über Vorlagen für grafische Anzeigeelemente wie Tanks, Ventile, Bedienknöpfe wie Schieberegler, AN/AUS Knöpfe, usw. verfügen, die auf einem Bildschirm in jeder beliebigen Konfiguration angeordnet werden können, um eine Bedienanzeige, eine Wartungsanzeige o. ä. zu erstellen. Vorlagen für grafische Anzeigeelemente können gemeinsam mit Konfigurationsobjekten in einer Vorlagenbibliothek oder in einer eigenen Vorlagenbibliothek gespeichert werden. Individuelle grafische Elemente können, wenn sie auf dem Bildschirm positioniert werden, miteinander

der so verbunden werden, dass einige Informationen oder Anzeigen der inneren Arbeitsabläufe der verfahrenstechnischen Anlage verschiedenen Nutzern angezeigt werden. Um die grafische Anzeige zu animieren, muss der Ersteller der Anzeige jedoch jedes der grafischen Elemente manuell mit den Daten, die in der verfahrenstechnischen Anlage erzeugt werden, wie z. B. Daten, die von Sensoren gemessen werden oder die Ventilstellungen anzeigen, usw. verbinden indem er eine kommunikative Verbindung zwischen dem grafischen Element und der entsprechenden Datenquelle innerhalb der verfahrenstechnischen Anlage herstellt. Dieser Prozess ist mühsam, langwierig und kann sehr stark fehlerbehaftet sein.

[0010] Darüber hinaus bleibt eine Anzeige, sobald sie einmal erstellt wurde, in ihrer Konfiguration und Anordnung statisch.

[0011] Während die Vorlagen der Regelobjekte innerhalb der Regelkonfigurationsanwendung und die Anzeigeelemente innerhalb der Anzeigenerstellungsanwendung bequem sind, da sie kopiert werden können und verwendet werden können, um viele verschiedene Steuermodule und grafische Anzeigen zu erstellen, besteht doch oft die Notwendigkeit viele gleiche Regelungsmodule und grafische Anzeigen für verschiedene Geräte und Anzeigen innerhalb der verfahrenstechnischen Anlage zu erstellen. Wenn z. B. viele mittelgroße bis große verfahrenstechnische Anlagen über viele Instanzen derselben oder ähnlicher Geräte verfügen, die mit denselben grundlegenden Steuermodulen und Anzeigen gesteuert und betrachtet werden können.

[0012] Um dieses Problem zu lösen, offenbart das U.S. Patent Nr. 7,043,311 (durch die Referenz herein ist hiermit ausdrücklich die gesamte Offenbarung des Patents inkorporiert) ein Konfigurationssystem einer verfahrenstechnischen Anlage, das Klassenobjekte, auch genannt Modulklassenobjekte (und im Allgemeinen hierin als Klassenobjekte oder Klassen bezeichnet), verwendet, um einem Nutzer zu ermöglichen mehrere Regelungsmodule, Einheiten oder Gerätemodule oder Anzeigenmodule aus einem allgemeinen Modulklassenobjekt zu erstellen. Diese Steuer-, Geräte oder Anzeigemodule werden als Instanzen des Modulklassen- oder Klassenobjekts erstellt und enthalten alle Eigenschaften und Funktionen des Modulklassenobjekts, was die Erstellung einer gewissen Anzahl von ähnlichen Regelungs-, Geräte- oder Anzeigeelementen aus einem einzigen allgemeinen Klassenmodulobjekt vereinfacht. Diese Instanzen können wiederum über eigene Kinderobjekte verfügen, so dass Objekte Verwandtschaften über mehrere Generationen oder Stufen haben können. Danach können alle Instanzen, Kinderobjekte oder Multilevel-Kinderobjekte, die aus einem Modulklassenobjekt heraus erstellt wurden, automatisch abgeändert werden, indem die Änderungen am entsprechen-

den Elternobjekt durchgeführt und gespeichert werden. Modulinstanzen bleiben beispielsweise weiterhin mit dem Modulklassenobjekt verknüpft und werden automatisch aktualisiert, wenn das Klassenmodul geändert oder aktualisiert wird. Auf ähnliche Weise können Kinderobjekte und Multilevelobjekte, die aus einem Elternobjekt erstellt wurden, das kein Klassenobjekt ist, auf ähnliche Weise automatisch geändert werden, indem am Elternobjekt Änderungen vorgenommen und gespeichert werden. In einer Verkörperung sind zumindest einige der Kinderobjekte an einem Systemkonfigurationsdatenspeicherort oder einem anderen Speicherort gespeichert, der logisch und/oder physisch von der Bibliothek getrennt ist.

[0013] In einem typischen Steuersystem jedoch, das z. B. in einer verfahrenstechnischen Anlage genutzt wird, können hunderte von ähnlichen Elementen (wie Steuermodule oder Anzeigeelemente) existieren, die definiert werden müssen.

[0014] Diese Elemente umfassen Regelungsstrategien, die z. Bsp. mit der Durchfluss- oder Druckregelung sowie mit den Anzeigekomponenten verknüpft sind, die zur Anzeige dieser Regelungsstrategien in grafischen Anzeigen verwendet werden. Es ist inzwischen ziemlich weit verbreitet, dass ein Konfigurationssystem implementiert wird, das Klassen- oder Modulklassenobjekte verwendet, um diese gemeinsamen Elemente zu konfigurieren, wobei das Konfigurationssystem eine Bibliothek von Modulklassenobjekten enthält und einem Nutzer ermöglicht, viele Kopien oder Instanzen von einem bestimmten Modulklassenobjekt zu erstellen, um dies in der Anlage zu verwenden. In diesen Systemen müssen Änderungen an den Objekten zuerst an den Modulklassenobjekten vorgenommen werden, welche dann diese Änderungen automatisch an alle Instanzen, Kinderobjekte und Multilevel-Kinderobjekte (wenn vorhanden) des Modulklassenobjektes weiterleiten. Diese Konfigurationssysteme sind so konzipiert, dass nur kleine Änderungen direkt an den Modulinstanzen erlaubt sind, wie z. B. Änderungen von Parameterwerten in den Modulinstanzen. Ein typisches Verhalten einer Klasse für Regelungsstrategien ermöglicht z. B. die Änderung einer Instanz einer Klasse nur auf dem Parameterlevel und nur, wenn diesem bestimmten Parameter innerhalb der Modulklassenobjekte ein Änderungszugriff gewährt wurde. Deshalb müssen viele verschiedene Modulklassenobjekte entwickelt werden, um Elemente in verfahrenstechnischen Anlagen zu konfigurieren (wie Geräte in der Anlage), die sich untereinander nur durch kleine Werte unterscheiden. Je mehr dieser Modulklassenobjekte definiert werden, umso weiter verringert sich dadurch leider der ursprüngliche Produktivitätsgewinn, der durch die Verwendung von Konfigurationsklassenobjekten erzielt wurde.

[0015] Darüber hinaus sind aktuelle Konfigurationssysteme, die auf Klassen basierende Konfigurationselemente verwenden, wie oben erwähnt in der Regel so entwickelt, dass, wenn eine Änderung an einem Klasselement vorgenommen wird, diese Änderung automatisch an alle Modulinstanzen dieser Klasse weitergeleitet werden. Diese Funktionalität ist in das Konfigurationssystem eingebettet, da Instanzen tatsächlich für deren Definition deren jeweilige Elternelemente oder -objekte teilen oder darauf zurückverweisen (z. B. das Klasselement oder -objekt). Diese automatische Weiterleitung von Änderungen macht die Entwicklung von Steuerungssystemen zwar in der Anfangsphase einfacher, sobald das Steuerungssystem aber installiert ist und in einer Anlage läuft, kann es aber möglicherweise unzulässig oder unpraktisch sein, alle Instanzen, die mit einem bestimmten Modulklassenobjekt verknüpft sind, gleichzeitig zu ändern. Ebenso sei darauf hingewiesen, dass Klassenobjekte in ihrer Lebenszeit in der Regel viele Änderungen durchlaufen. Nach Erstellung eines Modulklassenobjekts führen folgende Änderungen an der Modulklassenobjekt (die strukturelle Änderungen oder Änderungen von Parameterwerten sein können) zu einer veränderten Modulklassenobjekt, deren Änderungen an die Modulinstanzen, an eines oder mehrere Kindobjekte und/oder an andere Modulklassen, wie z. B. abgeleitete Modulklassen, verteilt werden müssen. In der Praxis jedoch muss möglicherweise jedes Kindobjekt, das in der Anlage betrieben wird, einzeln getestet werden und mit tatsächlichen Prozessgeräten verifiziert werden und deshalb muss der Nutzer die Änderungen am Modulklassenobjekt möglicherweise für Monate oder Jahre hinauszögern, da er auf einen passenden Zeitpunkt wartet, um alle Modulinstanzen dieses Klassenobjekts zu aktualisieren. Da die Änderungen an einem Modulklassenobjekt automatisch an die davon abgeleiteten Objekte weitergeleitet werden, kann eine Änderung an einer Modulklassenobjekt darüber hinaus hunderte und möglicherweise tausende Modulinstanzen und Kindobjekte beeinflussen. Viele Verfahrenstechniken kommen mit den resultierenden Störungen durch benötigte Downloads, die sich während der Betriebszeit der Anlage über mehrere Areale der Anlage erstrecken, nicht klar. Hieraus ergibt sich dass Steuerungssysteme anfänglich mit Modulinstanzen entwickelt werden, die an ein Modulklassenobjekt gebunden sind, nur um diese Module später in klassenlose Regelungsmodule umzuwandeln, bevor das System in die Betriebsphase eintritt.

[0016] Aus den oben genannten Gründen werden Klasseninstanzen in vielen Fällen vom ursprünglichen Klassenobjekt entkoppelt oder entbunden, um Änderungen an der Klasseninstanz zu erlauben, wodurch ein neues Klassenmodul erstellt wird, oder was Änderungen am Klassenobjekt erlaubt, die nicht sofort an alle Instanzen weitergeleitet werden.

[0017] Diese Vorgehensweise macht jedoch die Vorteile der Verwendung von klassenbasierten Konfigurationsobjekten zunichte.

[0018] Sogar in kleineren Systemen, in denen Klassen/Instanzen-Verbindungen nicht getrennt werden, ist es immer noch schwierig, ein neues Änderungs- und Steuerungsobjekt zu definieren, ohne alle Instanzen der Klasse zu beeinflussen, da es einerseits nicht möglich ist, neue Änderungen in einer Instanz vorzunehmen und andererseits jede einem Klassenobjekt vorgenommene Änderung, wird automatisch an alle Instanzen des Klassenobjekts weitergeleitet. Auch hier muss eines der Kindobjekte, z. B. eine Instanz, temporär vom Klassenobjekt getrennt werden, damit die neue Änderung entwickelt und an der Instanz getestet werden kann. In diesem Fall muss die Änderung, wenn sie einmal fertiggestellt ist, manuell am Klassenobjekt vorgenommen werden und der Nutzer muss die Instanz, die für Testzwecke verwendet wurde, reinigen und wieder an sein richtiges Klassenobjekt anbinden. In der Praxis muss der Nutzer also ein komplettes separates Klassenobjekt erstellen, um kleine Variationen zwischen Instanzen handhaben zu können. Hierzu muss er Klassen mehr und mehr zu Testzwecken von deren ursprünglichen Klassenobjekten abtrennen oder er muss das Klassenkonzept komplett aufgeben.

[0019] Die meisten Konfigurationssysteme, die heutzutage auf dem Markt verwendet werden, nutzen keine Klassen, um grafische Anzeigen für das Steuerungssystem zu konfigurieren. Die Grafiken sind in der Regel separat von den Regelungsstrategien definiert und ein Grafikelement wird oft zusammen mit verschiedenen Regelungsstrategieklassen verwendet. Deshalb muss der Nutzer sicherstellen, dass das geänderte Grafikelement mit allen Instanzen der Regelungsstrategie und auf allen Anzeigen funktioniert, wenn eine Änderung an einem bestimmten grafischen Element vorgenommen wird. Sobald die Änderungen an Regelungsstrategien auftreten und die Regelungsstrategieklassen diese Änderungen wie oben erläutert weitergeben, muss jedes grafische Element verifiziert werden, was unpraktisch und zeitraubend ist.

[0020] Grafikprogramme können zwar verwendet werden, um ähnliche grafische Elemente zu aktualisieren. Hierbei würden jedoch alle ausgewählten grafischen Elemente gleich gemacht. Diese Lösung ähnelt demzufolge letzten Endes der Verwendung von Klassen für grafische Elemente. Die Planung von Instanzunterschieden in Grafiken ist darüber hinaus schwierig und erfordert gute Programmierkenntnisse. Da in grafischen Elementen eine große Anzahl an Variationen erlaubt ist und sogar erwartet wird, ist es notwendig unter Verwendung spezieller Formen ein System zu entwickeln, das genau angibt, welche Variationen erlaubt werden oder mit Übersteuerungs-

strukturen in jedem der grafischen Anzeigenelemente, die von einem gemeinsamen grafischen Objekt aus erstellt wurden, versehen sind. Diese Variationen umfassen z. B. die Spezifikation von Änderungen, wie, es dem Nutzer in einer Instanz zu erlauben, die Rotation als Teil des Elements zu definieren oder zu wählen, welche Zeichenketten und Variablen auf der Anzeige gezeigt werden müssen und welche optional sind, usw. Wenn dies nicht von vornherein so entwickelt wurde, können an den grafischen Objekten nicht einmal kleine Änderungen durchgeführt werden. Ein Konfigurationssystem, das versucht, die erlaubten Änderungen in den Grafikelementen zu programmieren oder im Vorfeld festzulegen, wird leider schnell unbrauchbar, da Schwankungen bei grafischen Elementen sehr häufig sind. Daraus ergibt sich, dass die effektive Kostenerhaltung der Grafiken ein permanentes Problem innerhalb von Steuerungssystemen ist und sich nur dann verschärft, wenn der Erhaltungsvorgang für Grafiken mit den Änderungen koordiniert werden muss, die an Regelungsmodulklassen vorgenommen werden, die im Regelungskonfigurationssystem verwendet werden.

[0021] Allgemeiner gesagt wollen Nutzer eine Anzeige aus wiederverwendbaren grafischen Komponenten heraus konfigurieren und die Änderungen, die sie an der ursprünglichen grafischen Definition vorgenommen haben, sehen können. Zusätzlich wollen sie die Änderungen an den Definitionen an einem einzigen Ort vornehmen und diese Änderungen in den Anzeigen eingesetzt bekommen, für die die grafische Definition verwendet wird und danach die spezifischen Änderungen, die sie in verschiedenen Anzeigen gemacht haben, beibehalten. In manchen Anzeigesystemen können heute Anzeigekomponenten in Dynamos gruppiert werden, die eine Kombination aus Grundformen, wie Rechtecken und Textblöcken, sind, die mit den Animationen und Ereignisbehandler-Verhaltensweisen konfiguriert werden. Diese Grundformen können in verschiedenen Kombinationen konfiguriert werden, um Geräte oder Daten aus Teilen der Anlage zu repräsentieren. Manche Dynamos werden in der Regel von Beginn an als Teil des Anzeigenkonfigurationssystems bereitgestellt, während andere Dynamos durch Projektentwickler erstellt und gewartet werden und wieder andere vom Kunden konfiguriert werden. Derzeit wird für jede Instanz, die den ursprünglichen Dynamo verwendet, eine vollständige Kopie des Dynamos auf der Anzeige platziert, wenn dieser in einer Anzeige verwendet wird. Es besteht jedoch keine harte Verknüpfung zurück zum ursprünglichen Dynamo. Jede Instanz kann jedoch mehrere Aspekte des Dynamos ändern, wie z. B. die Pfade des Animationsausdrucks, die Position, die Orientierung oder andere visuelle Aspekte, die notwendig sind, um in das aktuelle Anzeigefeld des Nutzerinterface zu passen, in dem das Anzeigeelement verwendet wird.

[0022] Diese Herangehensweise birgt verschiedene Probleme. Der Nutzer versucht grob gesagt, die finanziellen Kosten zu minimieren und die Qualität und Verlässlichkeit der Anzeigen dabei zu maximieren. Es müssen jedoch unausweichlich eine oder mehrere Änderungen an einer Dynamodefinition vorgenommen werden. Nachdem eine Änderung an einem ursprünglichen oder Master-Dynamo vorgenommen wurde, muss der Kunde diese Änderungen in die Produktion einfügen indem er alle Anzeigen, die den spezifischen Dynamo nutzen, aktualisiert. Da jedoch keine harte Verknüpfung zwischen dem ursprünglichen Dynamo besteht, der zusammen mit seinen Kopien geändert wird, ist die Bestimmung des Einsatzortes des Dynamos problematisch. Jeder erstellte Dynamo enthält in der Regel ein Feld mit einer Zeichenfolge, um den Namen und die Version des Dynamos zu speichern. Nicht alle Kunden nutzen jedoch diese Technik und dieses Feld kann durch einen Nutzer bei einer Bearbeitung unabsichtlich gelöscht werden, womit die Verbindung zurück zum ursprünglichen Dynamo über den Namen verloren geht. Wenn, darüber hinaus, ein neuer Dynamo aus einem bestehenden Dynamo heraus erstellt wird und dieser Wert nicht aktualisiert wird, hat die Zeichenfolge den falschen Bezug zum Namen und der Version des ursprünglichen Dynamos. Wenn ein Unterschied eine Änderung in der Anzeige oder dem Dynamo ist, kann nur der Unterschied im Dynamo, nicht aber der Grund hierfür ermittelt werden. Es ist deshalb für Nutzer schwierig, festzustellen, ob eine Änderung in einem bestimmten Dynamo, der in einer bestimmten Anzeige verwendet wird, aufgenommen werden muss, wenn Änderungen am ursprünglichen Dynamo durchgeführt wurden, oder nicht.

[0023] Der Nutzer kann darüber hinaus die Instanzen zwar mit dem neuen Dynamo aktualisieren und die Änderungen spezifisch für die Instanz beibehalten, er weiß aber in der Regel nicht, welche Änderungen im neuen Master enthalten sind und welche spezifisch für die Instanz sind und kann deshalb nicht feststellen, welche Unterschiede innerhalb des Anzeigeelementes beibehalten werden sollten. Derzeit ist bei einer Aktualisierung eines Dynamos eine komplette neue Kopie des Dynamos erforderlich, um die Werte, die auf der Anzeige konfiguriert wurden, zu ersetzen. Dieser Vorgang überschreibt alle Änderungen auf der Anzeige, die instanz-spezifisch sind. Dadurch muss der Nutzer alle zuvor getätigten Änderungen manuell erneut eingeben. Sollte er dabei einige der Änderungen vergessen, sind diese verloren. Dieser Vorgang beeinträchtigt die Qualität der Bedienanzeigen und erfordert mehr Zeit, da er einen manuellen Vorgang voraussetzt. Darüber hinaus sind die ursprünglichen Dynamo-Stufen nur eine Stufe tief. Dadurch kann ein Dynamo nur aus Grundformen aufgebaut werden und nicht aus anderen Dynamos, was die Erstellung eines Dynamo-Satzes als Grundbausteine für andere Dynamos verhindert.

ZUSAMMENFASSUNG

[0024] Ein Konfigurationssystem nutzt Techniken mit flexiblen oder veränderbaren Objekten (z. Bsp. Modulklassenobjekte, Modulinstanzenobjekte und Kindobjekte), um eine Konfiguration basierend auf Klassen zu ermöglichen, die nützlicher und vorteilhafter für die Entwicklung von neuen Regelungsstrategien oder Anzeigeelementen und beim Vornehmen von Änderungen an diesen Elementen ist, wenn diese Elemente in der Umgebung einer Anlage ausgeführt werden. Ein solches Konfigurationssystem wird hierin entweder als „flexibles Konfigurationssystem“ oder als „Konfigurationssystem“ bezeichnet. Insbesondere verringert die Verwendung von neuen flexiblen Objekten die Zunahme von Klassenobjekten und macht ein einzelnes Klassenobjekt auf eine größere Breite von Anwendungen (Verwendungen) anwendbar, indem sie eine größere Variation zwischen den Instanzen desselben Klassenobjekts zulässt. Das Konfigurationssystem mit den neuen flexiblen Objekten ermöglicht es Nutzern z. B., Elemente zu einer Instanz (oder einem anderen Kindobjekt), die auf einer Klasse basiert, hinzuzufügen, wie dies für ein bestimmtes Bedürfnis erforderlich ist, oder ein Kindobjekt von der Klasse zu entfernen oder zu entkoppeln und dies auf eine Art und Weise, die die anderen Instanzen derselben Klasse nicht beeinflusst, solange der Nutzer diese Änderungen nicht auf die anderen Instanzen oder das Elternobjekt selbst übertragen möchte.

[0025] Ebenso können Elemente, die zu einer Instanz oder einem Kindobjekt hinzugefügt werden, innerhalb der Instanz oder des Kindobjekts markiert werden, damit sie nicht stören, wenn auf der Elternebene etwas hinzugefügt wird und damit eine klare Dokumentation gegeben ist. Das Konfigurationssystem mit den neuen flexiblen Objekten unterstützt darüber hinaus ein Ausschalten von Inhalten aus der Elternebene auf der Kindebene und/oder ein Löschen von Inhalten aus der Elternebene auf der Kindebene. Ein Ausschalten oder Löschen von Inhalten aus der Elternebene auf der Kindebene erlaubt es z. B. Instanzen, die weniger Funktionalität als andere Klasseninstanzen haben, besser zu handhaben, während man immer noch die grundlegende Klassenfunktionalität als Teil der Klasse beibehält. Falls erwünscht, kann der Inhalt, der auf der Kindebene ausgeschaltet wurde, innerhalb der Konfigurationsprogramme, mit denen das Kind angesehen wird, visuell stumm geschaltet oder unsichtbar gemacht werden und vom Nutzer jederzeit wieder eingeschaltet werden. Der ausgeschaltete Inhalt wird nicht in der Laufzeit verwendet, bleibt aber in der Konfigurationsumgebung vorhanden. Wenn der Nutzer entscheidet, einen zuvor ausgeschalteten Teil des Elternobjekts innerhalb eines Kindobjekts zu inkludieren, kann dieses Kindobjekt zum Download markiert werden, damit die Laufzeitversion im Regelungssystem aktualisiert wird.

[0026] Darüber hinaus macht das Konfigurationssystem mit flexiblen Objekten Klassen besser für die Verwendung in laufenden Anlagen nutzbar, indem hierdurch Änderungen und Aktualisierungen phasenweise an die Instanzen und Kindobjekte verteilt werden können. Ein Nutzer oder Konfigurationsentwickler kann z. B. Änderungen an einer Klasse wahlweise an bestimmte Instanzen weiterleiten, indem dem Nutzer erlaubt wird Änderungen an einem Klassenobjekt vorzunehmen und zu entscheiden, welche der Instanzen des Klassenobjekts diese Änderungen erhalten soll und zu welchem Zeitpunkt. Dies führt zu einer besseren Leistung der Anlage und zu einer besseren Laufzeitunterstützung, anstatt alle Änderungen an Klassen auf einmal zwangsweise an alle Instanzen weiterzugeben. Diese Funktion der phasenweisen Weitergabe erlaubt es dem Nutzer, Änderungen an einer Klasse vorzunehmen und diese nur an jene Instanzen weiterzugeben, die diese Änderungen benötigen. Der Nutzer kann später darauf zurückgreifen und die Änderungen an weitere oder die restlichen Instanzen verteilen. Genauer gesagt erlaubt es diese Funktion der phasenweisen Weitergabe dem Nutzer, zu entscheiden, auf welche Instanzen die Konfigurationsaktualisierung, basierend auf aktuellen Überlegungen zur Anlage, heruntergeladen werden und sie kann Änderungen an einer Klasse über einen flexiblen Zeitplan stufenweise ausgeben, der innerhalb des Betriebs der Anlage Sinn macht, anstatt manuell Downloads verwalten zu müssen oder einen automatischen Download an alle Instanzen gleichzeitig durchführen zu müssen.

[0027] Eine weitere Eigenschaft des Konfigurationssystems mit flexiblem Objekten ist es, dass der Nutzer besser schrittweise Änderungen vornehmen und verifizieren kann, indem Änderungen an Instanzen wahlweise auch zurück zum Klassenobjekt übertragen werden. Das Konfigurationssystem ermöglicht es einem Nutzer, z. B. eine Änderung, die auf einer Instanz vorgenommen wurde, auf die höhere Ebene des Klassenobjekts zu übertragen, um sie von dort dann wieder hinunter auf die anderen Instanzen zu übertragen. Diese Funktion bietet dem Nutzer eine eingebaute Testumgebung mit der er eine Instanz ändern und prüfen kann, bevor er diese Änderungen auf alle anderen Instanzen des Klassenobjektes überträgt.

[0028] Das hier beschriebene Konfigurationssystem, hält die Vorteile des klassenbasierten Verhaltens aufrecht, während es gleichzeitig einen Mechanismus bietet, mit dem Nutzer das Übertragen von Änderungen von Modulklassen und anderen Elternobjekten auf die Modulinstanzen und andere Kindobjekte kontrollieren können, die in Reglern, Geräten, Anzeigen oder anderen Prozesselementen innerhalb der verfahrenstechnischen Anlage oder dem Steuerungssystem betrieben werden, z. B. innerhalb der Laufzeitumgebung der verfahrenstechnischen Anla-

ge oder eines Systems. Das hier beschriebene Konfigurationssystem erlaubt es, Änderungen an Modulklassen und anderen Elternobjekten in Prozesselemente einer verfahrenstechnischen Anlage oder einem Prozesssteuerungssystem (die zu den Elternobjekten und den dazugehörigen Kindobjekten zugewiesen sind) zu integrieren, um den Betrieb und/oder das Verhalten der Anlage oder des Systems zu ändern, wobei dies auf kontrollierte und stufenweise Art nach einem Zeitplan geschieht, damit ein Echtzeitbetrieb in Teilen der verfahrenstechnischen Anlage oder im Prozesssteuerungssystem (oder in manchen Fällen der gesamten Anlage oder dem gesamten System) nicht beeinträchtigt wird. In manchen Situationen wird eine Änderung an einem Elternobjekt nicht an die Kindobjekte weitergegeben. Darüber hinaus wird das Vorkommen von unnötigen Verzögerungen bei der Durchführung von Änderungen an Teilen der verfahrenstechnischen Anlage oder des Prozesssteuerungssystems (oder, in manchen Fällen, an der gesamten Anlage oder am gesamten System) verringert, da Änderungen kontrolliert und schrittweise an den Prozesselementen der verfahrenstechnischen Anlage oder des Systems vorgenommen werden (d. h. anstatt zu warten, bis es für alle Prozesselemente, die den Kindobjekten der geänderten Elternobjekte zugehörig sind, günstig ist aktualisiert zu werden). Somit wird die Gesamteffizienz und die Produktivität der verfahrenstechnischen Anlage oder des Systems mit der Zeit gesteigert.

[0029] Die hier beschriebenen Techniken, Systeme und Methoden erlauben dem Nutzer, sicher an temporären Kopien eines Eltern- oder Kindobjekts zu arbeiten und machen das Eltern- oder Kindobjekt nur dann auf einem Regler, Gerät oder anderen Prozesselementen nutzbar (zum Herunterladen), wenn die entsprechenden Zustimmungen eingegangen sind. Die Vorgehensweise, Zustimmungen für Entwürfe oder Änderungen zu verwenden, die in einem Prozesselement in einer Laufzeitumgebung der verfahrenstechnischen Anlage oder des Steuerungssystems angewendet werden (egal ob von einem geänderten Elternobjekt oder einem Kindobjekt aus), verringert die Wahrscheinlichkeit, dass eine unvollständige oder falsche Änderung unabsichtlich in die Anlage oder das System eingeführt wird. Das bedeutet, dass durch die Verwendung der hier beschriebenen Entwurfs- und Zustimmungstechniken das Konfigurationssystem die Genauigkeit der Änderungen verbessert, die in die verfahrenstechnische Anlage oder das System eingeführt werden und somit die Qualität des Betriebs der verfahrenstechnischen Anlage oder des Systems verbessert.

[0030] In einem Fall kann das Konfigurationssystem beispielsweise eine Synchronisierung durchführen, indem es einen Bearbeitungs- und Verteilungsmechanismus ausführt, der entfernt auf einem Modell zur Veröffentlichung von Büchern basiert, in dem

Nutzer an Entwürfen arbeiten, die schlussendlich bestätigt werden, was zu den Versionen führt, die an Orten innerhalb der Anlage verteilt werden. Manche Regler, Anzeigen oder Anlagegeräte erhalten oder nutzen die neueste Version, sobald sie herausgegeben wird, während andere nur auf die neuen Versionen aktualisiert werden, wenn es am günstigsten oder notwendigsten ist. Mit diesem System können Nutzer Änderungen an der Konfigurationsbibliothek und Systemkonfigurationselementen vornehmen, ohne Downloads auszulösen und dadurch möglicherweise das Laufzeitsystem zu beeinflussen. Nutzer können Änderungen an Elementen oder Objekten als Entwurf speichern. Außerdem können sie, sobald die Änderungen bestätigt werden, eine Version erstellen, die auf das Laufzeitsystem verteilt und heruntergeladen werden kann. Um genauer zu sein bedeutet das, dass Modulinstanzen (und andere Kindobjekte) im Konfigurationssystem einer spezifischen Version einer Modulkategorie oder einem spezifischen Elternobjekt zugeordnet sind, wodurch es möglich ist, dass die Modulkategorie oder das Elternobjekt noch weitere Änderungen durchläuft, ohne dass seine Kinder sofort hiervon beeinflusst werden. Als veranschaulichendes Beispiel ist bei dieser Funktion das Herunterladen von Modulinstanzen, jedes Mal, wenn eine Änderung an der Modulkategorie durchgeführt wird, nicht mehr erforderlich. Darüber hinaus kann der Nutzer stattdessen kontrollieren, wann eine Modulinstanz aktualisiert wird, um mit der neuesten Version einer Modulkategorie verbunden zu sein. In der Tat sind Kindobjekte mit einer spezifischen Version ihrer Elternobjekte verknüpft, egal ob das Elternobjekt ein Bibliothekselement ist oder nicht. Hierdurch können Elternobjekte geändert werden, ohne sofort alle Kindobjekte zu beeinflussen.

[0031] Mittels Verwendung von Techniken der schrittweisen Einführung in das Konfigurationssystem, die hier beschrieben wird, können Nutzer wählen, ob sie Änderungen an Bibliothekselementen oder Elternobjekten automatisch an Kindobjekte weitergeben wollen oder ob sie die Verteilung manuell steuern. Die Elternbibliotheks- und Systemkonfigurationselemente können darüber hinaus auch mehrere Stufen von Entwurf, Änderung und (bei Systemkonfigurationselementen) Downloadversionen durchlaufen. Nutzer können hier temporäre Kopien (Entwürfe) von Elternobjekten erstellen und, nur wenn eine angemessene Zustimmung erteilt wurde, die geänderten Versionen der Elemente in das Laufzeitsystem heruntergeladen, was bedeutet, dass Entwürfe getestet werden können, ohne das Laufzeitsystem zu beeinflussen. Nutzer können ebenso, um das Benutzererlebnis zu verbessern und den Nutzern das Gefühl zu geben mehr Kontrolle zu haben, die Unterschiede zwischen den Entwürfen, der aktuellen Version und der heruntergeladenen Version sehen. Im Kontext eines spezifischen Elements kann sich ein Nutzer auf Wunsch die Unterschiede zwischen einem

Entwurf und der aktuellen Version, dem Entwurf und der heruntergeladenen Version, der aktuellen Version und der heruntergeladenen Version, usw. anzeigen lassen. Darüber hinaus können Bibliothekselemente und Systemkonfigurationselemente Referenzen unterstützen, die es dann ermöglichen zu sehen, welche Systemkonfigurationselemente ein Bibliothekselement nutzen und ob das Systemkonfigurationselement die aktuelle Version des Bibliothekselements verwendet.

[0032] Durch Verwendung von Versionen innerhalb der Modulkategorie und der Instanzenstruktur können Pakete erstellt werden, die im Wesentlichen Sammlungen von Versionen von spezifischen Versionen von Elternelementen sind. Darüber hinaus lassen sich diese Pakete zum gleichzeitigen Download verteilen. Diese Funktion ermöglicht es, ein Bibliothekselement, das mit einer bestimmten Lösung oder einem bestimmten Problem verbunden ist, zu entwickeln. Hieraus kann wiederum fast die gesamte oder die gesamte Anlagenkonfiguration erstellt werden. Wenn das Paket dann aktualisiert wird, um Probleme zu korrigieren oder um die Funktionalität zu erweitern, kann die neue Version des Pakets installiert werden, indem die neuen Versionen der Bibliothekselemente, die gleichzeitig im Paket enthalten sind, installiert werden.

[0033] Es steht den Nutzern dann frei, die Systemkonfiguration zu aktualisieren, wie es die Situation in der Anlage erlaubt, was logistische Probleme verringert oder beseitigt, die in aktuellen Konfigurationssystemen entstehen, bei denen eine sofortige Aktualisierung im Elternelement erforderlich ist, um die Kindelemente zu aktualisieren. Beispielsweise könnte das hier beschriebene Konfigurationssystem genutzt werden um Bibliothek- und Systemkonfigurationselemente inklusive der Entwicklereinheiten, Alarmeinstellungen, globaler Werte und Sicherheitsobjekte (wie z. B. Nutzerkonten, Funktions- und Parametersicherheit und Genehmigungsrichtlinien) zu verwalten und auch auf vom Nutzer erstellte Dokumentierung, wie z. B. Standardvorgehensweisen (SOPs), Inbetriebnahme/Abschaltung, Alarmhilfe, usw. angewendet werden.

[0034] Im Kontext der Anzeigeelemente können Nutzer mit Hilfe der flexiblen Objekte des Konfigurationssystems die verlinkten konfigurierbaren grafischen Formen (z. B. Grafikelement-Module oder GEMs) erstellen, die wiederverwendbar sind und andere Formen mit Verhaltensweisen kombinieren. GEMs liefern eine oder mehrere Darstellungsvarianten oder Ansichten einer konfigurierbaren Form. Die Definition eines GEM ist separat von der Verwendung/Instanz dieses GEM in spezifischen Anzeigen und anderen Objekten gespeichert (um z. B. ein Übermitteln der GEM Definition zu ermöglichen). Zusätzlich unterstützen die flexiblen Objekte des Supports für das

Konfigurationssystem das Bestimmen oder Ableiten grafischer Definitionen von anderen grafischen Definitionen (z. B. Ansichten und GEMs). Beispielsweise kann ein GEM unter Verwendung eines anderen GEM erstellt werden.

[0035] Übersteuerungen (overrides) oder Änderungen an einem Elternobjekt resultieren in der Regel in Änderungen an einer Instanz oder einem Kindobjekt. Die flexiblen Objekte des Konfigurationssystems erweitern dieses Override-Konzept, um Änderungen an der internen Struktur eines Anzeigeelements zuzulassen, indem ein einfacher Parameter-Override, der von Regelungsstrategie-Funktionsblöcken unterstützt wird, verwendet wird. Diese zusätzlichen Overrides können zum Beispiel folgende Overrides umfassen: Overrides für Eigenschaften, Animationen und Event Handler; Overrides, die das Hinzufügen einer grafischen Form, einer Animation oder eines Event Handlers unterstützen, Overrides, die Bewegungen einer grafischen Form unterstützen und/oder Overrides, die das Entfernen einer grafischen Form, Animation oder eines Event Handlers unterstützen. Die Overrides für spezifische Anzeigeeinstanzen können separat von den GEM Definitionen mit der Instanz zusammen gespeichert werden, um wissen zu können, welche Änderungen zu einer spezifischen Anzeigeeinstanz gehören, im Gegensatz zu den Änderungen, die in den GEM Definitionen vorgenommen wurden, die in der Anzeigeeinstanz verwendet werden. Overrides für GEMs und andere Anzeigeelemente können an sich (egal ob Eltern oder Kind) Änderungen am Inhalt der GEMs und anderen Anzeigeelementen enthalten.

[0036] Wenn eine Anzeige geladen wird, kann die GEM Definition darüber hinaus verwendet werden, um die anfängliche Repräsentation, Animation und die Event Handling Verhaltensweisen des Anzeigeelements festzustellen. Danach können die Overrides nach Bedarf an einer spezifischen Anzeigeeinstanz eines Anzeigeelements oder Klassenobjekts angewendet werden, um die instanz-spezifischen Änderungen bereitzustellen. Diese Funktion kann unter Verwendung von einfachen Eigenschaftsübersteuerungen (overrides) implementiert werden, um die Orientierung oder die Position einer Form zu ändern oder eine Form zu entfernen oder durch eine andere Form zu ersetzen, wie z. B. das Entfernen eines Textblocks, der einen Textwert zeigt, um dann ein Rechteck einzufügen, das den Wert grafisch als Füllprozentatz zeigt. Alternativ hierzu können Overrides eine Animation oder einen Event Handler hinzufügen oder eine existierende Animation oder einen Event Handler ändern, indem eine oder mehrere Eigenschaften der existierenden Animation oder des Event Handlers übersteuert werden. Das geänderte, übersteuerte GEM kann als Kindobjekt der GEM Definition in einer Verkörperung gespeichert oder erhalten werden.

[0037] Zusätzlich können über das Anzeigenkonfigurationssystem GEMs auf anderen GEMs erstellt werden, wodurch allgemeinere Elemente leichter für eine spezifische Funktion, Industriegruppe oder Kunden erstellt werden können. GEMs können zum Beispiel durch Ableitung erweitert werden. Dabei kann dieselbe Override-Struktur an einem abgeleiteten GEM angewendet werden, wodurch ein Anzeigeelement als neues Anzeigeklassenobjekt in der Bibliothek oder als Instanzen, die vom Bedienpersonal in der Laufzeit geöffnet werden, erweitert werden kann. Das Anzeigenkonfigurationssystem kann auch einen Mechanismus enthalten, womit der Nutzer spielend einfach alle der Instanzen (wo verwendet) eines GEM bestimmen kann und außerdem ganz einfach feststellen kann, welche Anzeigen von anderen Anzeigen abgeleitet wurden. Ebenso kann das Anzeigenkonfigurationssystem einen Mechanismus enthalten, womit Nutzer definitiv feststellen können, was bei einer abgeleiteten Definition oder Verwendung geändert wurde, um die interne Struktur einer zusammengesetzten Form (z. B. eines GEM) zu übersteuern (tweaken) während eine verlinkte Verbindung zur ursprünglichen GEM Definition bestehen bleibt, um Formen hinzuzufügen und zu entfernen, um Daten (Animationen) hinzuzufügen und zu entfernen, um Handler hinzuzufügen und zu entfernen sowie um die Eigenschaften der Formen, Animationen, Event Handler und aller anderen Objekttypen in der GEM Definition zu übersteuern.

[0038] Darüber hinaus kann das Anzeigenkonfigurationssystem einen Mechanismus enthalten, womit der Nutzer eine Definition oder eine abgeleitete Definition (ein Anzeigenklassenelement) auf eine neue Version aktualisieren können und diese Änderungen in Instanzen anwenden können, ohne die Overrides dieser Instanzen zu verlieren. Das Anzeigenkonfigurationssystem kann auch einen Mechanismus enthalten, womit der Nutzer GEMs erstellen kann. Hierzu werden andere GEM Definitionen kombiniert. Darüber hinaus ist er hiermit in das Lage, eine Anzeige von einer anderen Anzeige abzuleiten, ein GEM von einem anderen GEM abzuleiten und Tweaks und Overrides anzuzeigen, die an den Anzeigeeinstanzen vorgenommen wurden.

[0039] Die flexible Konfiguration der grafischen Elemente und Anzeigen, die vom Anzeigenkonfigurationssystem bereitgestellt werden, ermöglicht eine sichere und effiziente Überwachung einer verfahrenstechnischen Anlage oder eines Prozesssteuerungssystems und deren Kontrolle und Echtzeitbetrieb. Insbesondere ermöglicht das Anzeigenkonfigurationssystem dem Bedienpersonal grafische Elemente und oder Anzeigen, die spezifisch angepasst oder auf eine bestimmte Überwachung, Steuerung und/oder Betrieb eines oder mehrerer Teile des Prozesssteuerungssystems oder der Anlage zugeschnitten sind, sowohl in Echtzeit als auch über Laufzeitumgebun-

gen der Anlage zu konfigurieren. Die grafischen Elemente und/oder Anzeigen können zusätzlich oder alternativ dazu spezifisch zugeschnitten oder an eine bestimmte Person, Gruppe des Bedienpersonals, Organisation, Unternehmen oder andere Einheit angepasst werden. Über das Anzeigenkonfigurationssystem kann das Bedienpersonal auch grafische Elemente und/oder angepasste Anzeigen für allgemeinen (z. Bsp. systemweiten, Echtzeitumgebungs- oder Konfigurationsumgebungs-)Zugang speichern, verwenden, wiederverwenden und einbetten. Dementsprechend verringern sich Unterbrechungen und Fehler durch das Bedienpersonal, da die Konfiguration von grafischen Elemente(n) und/oder grafischen Anzeige(n) effizient gestaltet und vollständig an verschiedene Umgebungen der verfahrenstechnischen Anlage oder des Systems anpassbar ist. Hierdurch ist das Bedienpersonal in der Lage, die verfahrenstechnische Anlage oder das System effizient und sicher zu bedienen.

[0040] Da die grafischen Elemente und/oder Anzeige(n) auf bestimmte Zwecke für bestimmte Teile oder Einheiten der verfahrenstechnischen Anlage oder des Systems zugeschnitten sind, können Daten, die von den bestimmten Teilen oder Einheiten der verfahrenstechnischen Anlage oder des Systems in Echtzeit geliefert werden (z. Bsp. während einer oder mehrere Prozesse gesteuert werden) und die ein manuelles oder automatisches Eingreifen erfordern, schnell und einfach erkannt werden, indem die angepassten grafischen Elemente und/oder Anzeige(n) verwendet werden. Echtzeitdaten werden zum Beispiel empfangen und auf einem angepassten grafischen Element und/oder einer Anzeige erkannt. Basierend auf dem Inhalt der erkannten Daten wird ein aktualisierter Steuerungsalgorithmus manuell und/oder automatisch erstellt und an die verfahrenstechnische Anlage übermittelt. Ein anderes Beispiel könnte sein, dass, basierend auf dem Inhalt der erhaltenen und erkannten Daten, eine Anweisung für eine Wiederherstellungsaktion manuell oder automatisch an eines oder mehrere Prozesselemente der verfahrenstechnischen Anlage oder des Systems geschickt wird, die durchgeführt werden muss und/oder eine Anweisung zum Herunterfahren oder Hochfahren an eine oder mehrere Prozesselemente der verfahrenstechnischen Anlage oder des Systems geschickt wird, die ausgeführt werden muss. Selbstverständlich ist es auch möglich, basierend auf empfangenen und erkannten Echtzeitdaten andere Daten, Konfigurationen und/oder Anweisungen an die verfahrenstechnische Anlage oder das System zu schicken und/oder davon ausführen zu lassen. Die Daten, Konfigurationen und/oder Anweisungen, die, basierend auf erkannten Echtzeitdaten, an die verfahrenstechnische Anlage oder das System geschickt werden müssen resultieren in einer Änderung an der verfahrenstechnischen Anlage oder dem System (z. B. eine aktualisierte oder neue Konfiguration eines Pro-

zesselements) oder in Operationen davon. In einigen Fällen lassen die verschickten Daten, Konfigurationen und/oder Anweisungen die verfahrenstechnische Anlage oder das Steuerungssystem eine Handlung durchzuführen (z. B. ein bestimmtes Prozesselement aus dem Betrieb nehmen, Daten, die von einer Quelle erstellt wurden zu einer anderen Quelle umleiten, usw.).

[0041] Dementsprechend werden alle erforderlichen Änderungen an der Steuerung und/oder dem Betrieb einer oder mehrerer Teile der verfahrenstechnischen Anlage oder des Prozesssteuerungssystems schneller festgestellt und in die Laufzeitumgebung der verfahrenstechnischen Anlage oder des Systems integriert, weil über das Anzeigekonfigurationssystem grafische Elemente und/oder Anzeigen besser angepasst erstellt und detailliertere Informationen (besonders im Hinblick auf Echtzeitdaten, die von der verfahrenstechnischen Anlage oder dem System erstellt werden) erzeugt werden können. Hieraus ergibt sich, dass die Effizienz und die Sicherheit der verfahrenstechnischen Anlage und des Systems mit den hier beschriebenen Techniken, Methoden und dem System weiter erhöht werden.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0042] Fig. 1 ist ein Blockdiagramm eines verteilten Beispielprozesssteuerungsnetzwerks, das sich in einer verfahrenstechnischen Anlage oder einem Prozesssteuerungssystem befindet und eine Arbeitsstation einschließt, die eine Konfigurationsanwendung einsetzt, die Modulklassenobjekte verwendet, um Regelungs- und Anzeigeaktivitäten der verfahrenstechnischen Anlage oder des Prozesssteuerungssystems zu konfigurieren.

[0043] Fig. 2 ist ein Diagramm einer Reaktoreinheit von Fig. 1;

[0044] Fig. 3 ist ein Diagramm einer Totalisatoreinheit aus der Reaktoreinheit von Fig. 2;

[0045] Fig. 4 ist ein Diagramm eines Ausgangsventilsystems aus der Reaktoreinheit von Fig. 2;

[0046] Fig. 5 ist ein eines logisches Beispieldiagramm, das die Verknüpfungen zwischen Modulklassenobjekten und assoziierten Modulobjekten für Einheiten, Geräte, Regelung und Anzeigetypen von Modulklassenobjekten darstellt;

[0047] Fig. 6 ist ein logisches Beispieldiagramm eines Modulklassenobjekts einer Reaktoreinheit, das für Konfigurationen von Reaktoren innerhalb der Anlage von Fig. 1 eingesetzt werden kann;

[0048] Fig. 7 ist eine Beispielabbildung eines Konfigurationsbildschirms, der von einem Konfigurations-

bediener verwendet werden kann um eine verfahrenstechnische Anlage mithilfe von Modulklassenobjekten zu konfigurieren;

[0049] Fig. 8 ist ein Blockdiagramm eines Beispielszenarios, in dem ein Nutzer ein Modulobjekt ändert, das gespeichert ist in einer Bibliothek des Prozesssteuerungssystems aus Fig. 1;

[0050] Fig. 9 ist ein Blockdiagramm eines zweiten Beispielszenarios, in dem ein Nutzer ein Modulobjekt ändert, das gespeichert ist in einer Bibliothek des Prozesskontrollsystems von Fig. 1;

[0051] Fig. 10 ist ein Blockdiagramm des zweiten Beispielszenarios aus Fig. 9 einschließlich zusätzlicher Aktionen;

[0052] Fig. 11 ist ein Blockdiagramm eines dritten Beispielszenarios, in dem ein Nutzer ein Modulobjekt ändert, das gespeichert ist in einem Systemkonfigurationsspeicher des Prozesskontrollsystems von Fig. 1;

[0053] Fig. 12 stellt drei mögliche Ansichten eines Beispiels für ein grafisches Ventilelement dar;

[0054] Fig. 13 stellt ein Diagramm eines Beispiels für ein Definitionsnutzungsmuster dar;

[0055] Fig. 14 stellt ein Klassendiagramm eines Beispiels für ein geschachteltes Definitionsnutzungsmuster dar;

[0056] Fig. 15 stellt ein Beispiel für eine abstrahierte verschachtelte Definitionsnutzung dar;

[0057] Fig. 16 stellt ein Diagramm eines Beispiels für ein Ableitungsmuster dar;

[0058] Fig. 17 stellt ein Diagramm eines Beispiels für ein geschachteltes Ableitungsmuster dar;;

[0059] Fig. 18 stellt ein Beispiel für eine Verwendung von Formen dar, die auf einer Definition basieren;

[0060] Fig. 19 stellt ein Beispiel für einen Feldwert-Override-Tweak dar, der auf ein Nutzungsmuster angewendet wurde;

[0061] Fig. 20 stellt ein Beispiel für einen Feldwert-Override-Tweak dar, der auf ein Ableitungsmuster angewendet wurde;

[0062] Fig. 21 stellt ein Beispiel für verschiedene Änderungen oder Tweaks an einem Objekt auf mehreren Ebenen dar.

BESCHREIBUNG VON VERKÖRPERUNGEN

[0063] In Bezug auf **Fig. 1**, enthält eine verfahrenstechnische Anlage **10** einen oder mehrere Prozessregler **12**, die mit mehreren Arbeitsstationen **14** über z. Bsp. eine Ethernet-Verbindung oder einen Bus **15** verbunden sind. Die Regler **12** sind innerhalb der verfahrenstechnischen Anlage **10** über Sets an Datenübertragungswegen oder Buse **18** auch mit Geräten oder Equipment verbunden, wobei nur das Set der Datenübertragungswege **18** mit dem Regler **12a** in **Fig. 1** dargestellt ist. Die Datenübertragungswege oder Busse **18** können z. B. Kabelverbindungen, drahtlose Verbindungen oder eine Kombination aus Kabel- und kabellosen Verbindungen sein. Die Regler **12**, die implementiert sein können, indem nur die DeltaV™ Regler von Fisher-Rosemount Systems, Inc. verwendet werden, können mit den Regelungselementen wie Feldgeräten und Funktionsblock innerhalb der Feldgeräte, die über die verfahrenstechnische Anlage **10** verteilt sind, kommunizieren um eine oder mehrere Prozesssteuerungsroutinen **19** durchzuführen und so die gewünschte Kontrolle über die verfahrenstechnische Anlage **10** zu implementieren oder eine oder mehrere Prozesse in der verfahrenstechnischen Anlage **10** zu betreiben. Die Arbeitsstationen **14** (z. B. PCs) können von einem oder mehreren Konfigurationsentwicklern verwendet werden, um die Prozesssteuerungsroutinen **19** zu entfernen, die von den Reglern **12** ausgeführt werden sollen sowie um Routinen anzuzeigen, die von den Arbeitsstationen **14** oder anderen Computern ausgeführt werden sollen und um mit den Reglern **12** zu kommunizieren, damit diese Prozesssteuerungsroutinen **19** auf die Regler **12** heruntergeladen werden können. Die Arbeitsstationen **14** können darüber hinaus Anzeigeroutinen ausführen, die während des Betriebs der verfahrenstechnischen Anlage **10** Informationen empfangen und anzeigen, die zur verfahrenstechnischen Anlage **10** oder deren Elementen gehören.

[0064] Jede der Arbeitsstationen **14** umfasst einen Speicher **20** für die Speicherung von Anwendungen, wie z. B. Konfigurationsentwicklungsanwendungen und Anzeige- oder Betrachteranwendungen und zur Speicherung von Daten, wie den Konfigurationsdaten, die zur Konfiguration der verfahrenstechnischen Anlage **10** gehören. Jede der Arbeitsstationen **14** umfasst auch einen Prozessor **21**, der die Anwendungen ausführt und es einem Konfigurationsentwickler ermöglicht, Prozesssteuerungsroutinen und andere Routinen zu entwickeln und diese Prozesssteuerungsroutinen auf die Regler **12** oder andere Computer herunterzuladen, um Informationen zu sammeln und einem Nutzer während des Betriebs der verfahrenstechnischen Anlage **10** anzuzeigen. In manchen Verkörperungen ist ein Remote-Computing Gerät über einen Datenübertragungsweg mit den Arbeitsstationen **14** verbunden (z. B. über eine Netz-

werk- oder Web-basierte Schnittstelle), so dass ein Konfigurationsentwickler Anwendungen ferngesteuert von den Arbeitsstationen **14** aus ausführen kann.

[0065] Darüber hinaus umfasst jeder der Regler **12** einen Speicher **22**, der Regelungs- und Kommunikationsanwendungen speichert und einen Prozessor **24**, der die Regelungs- und Kommunikationsanwendungen auf jede bekannte Weise ausführt. In einem Fall speichert jeder der Regler **12** eine Regelungsanwendung und führt sie aus, die eine Steuerungsstrategie implementiert, indem sie eine gewisse Anzahl an verschiedenen und unabhängig ausgeführten Regelungsmodulen oder -blöcken **19** verwendet. Die Steuermodule **19** können jeweils aus Funktionsblöcken, wie sie im Allgemeinen bezeichnet werden, aufgebaut sein, wobei jeder Funktionsblock ein Teil oder eine Subroutine einer übergeordneten Steuerungsroutine ist und (über Verbindungen die „Datenübertragung“ (Communications) genannt werden) gemeinsam mit anderen Funktionsblöcken operiert, um Prozesssteuerungsschleifen innerhalb der verfahrenstechnischen Anlage **10** zu implementieren, z. B. um den Betrieb von einem oder mehreren Prozessen, die von der verfahrenstechnischen Anlage **10** ausgeführt werden, zu steuern.

[0066] Bekanntlich führen Funktionsblöcke, die Objekte in einem objektorientierten Programmierprotokoll sein können, typischerweise eine der folgenden Funktionen aus: eine Eingabefunktion, wie z. B. die mit einem Feldgerät, wie einem Transmitter, Sensor oder anderem Messgerät für Prozessparameter, assoziierten Funktionen; eine Steuerungsfunktion, wie z. B. die mit einer Steuerungsroutine, die PID-Steuerung, Fuzzy Logic-Steuerung, usw. ausführt, assoziierte Funktionen; oder eine Ausgabefunktion, die den Betrieb eines Gerätes steuert, wie z. B. eines Ventils oder anderen Feldgerätes, um physische Funktionen innerhalb der verfahrenstechnischen Anlage **10** auszuführen. Es gibt natürlich auch Hybridarten und andere Arten von komplexen Funktionsblöcken, wie z. B. Model Predictive Controllers (MPCs), Optimierer, usw. Während das Feldbus-Protokoll und das DeltaV Systemprotokoll Steuerungsmodule und Funktionsblöcke verwenden, die in einem objektorientierten Programmierprotokoll entwickelt und eingesetzt werden, können die Steuerungsmodule unter Verwendung eines beliebigen Steuerungsprogrammierschemas entwickelt werden, einschließlich beispielsweise einer sequentiellen Funktionstabelle, Kontaktplanlogik, usw. und sind nicht darauf beschränkt unter Verwendung von Funktionsblock oder einer anderen bestimmten Programmiertechnik entwickelt zu werden.

[0067] Die Arbeitsstationen **14** können einem Nutzer über einen Anzeigebildschirm eine grafische Abbildung der Prozesssteuerungsroutinen **19** innerhalb der Regler **12** anzeigen und die Steuerungselemente innerhalb der Prozesssteuerungsroutinen **19** und die

Art darstellen, in der diese Steuerungselemente konfiguriert sind, um die Steuerung der verfahrenstechnischen Anlage **10** bereitzustellen. Im System von **Fig. 1** ist eine Konfigurationsdatenbank **25** mit dem Ethernet-Bus **15** verbunden, um Konfigurationsdaten zu speichern, die von Reglern **12** und Arbeitsstationen **14** verwendet werden und, in manchen Fällen, um als Datenspeicher zu dienen, indem sie Daten, die in der verfahrenstechnischen Anlage **10** generiert werden, sammelt und für zukünftigen Gebrauch speichert. In einer Verkörperung kann die Konfigurationsdatenbank **25** Bibliothekselemente (z. B. Vorlagen und Klassenmodule) und Systemkonfigurationselemente (z. B. Objekte, die aus Bibliothekselementen erstellt wurden), die mit den Konfigurationsdaten in Verbindung stehen, enthalten. Die Konfigurationsdatenbank **25** an sich kann logisch und/oder physisch in einen Bibliotheksdatenspeicher und ein Systemkonfigurationsspeicher getrennt sein.

[0068] In der verfahrenstechnischen Anlage **10**, die in **Fig. 1** abgebildet ist, ist der Regler **12a** kommunikativ über den Bus **18** mit drei Sets von ähnlich konfigurierten Reaktoren (welche replizierte Ausrüstung innerhalb der Anlage **10** sind) verbunden, die hier als Reaktor_01 und Reaktor_02 und Reaktor_03 bezeichnet werden. Reaktor_01 umfasst ein Reaktorgefäß oder einen Reaktortank **100**, drei Eingangsventilsysteme (die Ausrüstungseinheiten sind) **101**, **102** und **103**, die so verbunden sind, dass sie jeweils die Flüssigkeitseingangsleitungen in das Gefäß **100** steuern, die Säure, Base und Wasser liefern, und einem Ausgangsventilsystem **104**, das so verbunden ist, dass es den Flüssigkeitsdurchfluss aus dem Reaktorgefäß **100** steuert. Ein Sensor **105** eines beliebigen Sensortyps, wie z. B. ein Füllstandsensor, ein Temperatursensor, ein Drucksensor, usw., wird in der Nähe oder im Reaktorgefäß **100** angebracht. Zwecks dieser Erläuterungen wird angenommen, dass der Sensor **105** ein Füllstandsensor ist. Darüber hinaus ist ein geteiltes Vorventilsystem **110** flussaufwärts von den Reaktoren Reaktor_01, Reaktor_02 und Reaktor_03 an der Wasserleitung angebracht, um eine Mastersteuerung für die Regelung des Wasserdurchflusses zu jedem dieser Reaktoren zu bieten.

[0069] Ebenso umfasst Reaktor_02 ein Reaktorgefäß **200**, drei Eingangsventilsysteme **201**, **202** und **203**, ein Ausgangsventilsystem **204** und einen Füllstandsensor **205** und Reaktor_03 ein Reaktorgefäß **300**, drei Eingangsventilsysteme **301**, **302** und **303**, ein Ausgangsventilsystem **304** und einen Füllstandsensor **305**. Im Beispiel von **Fig. 1** können die Reaktoren Reaktor_01, Reaktor_02 und Reaktor_03 mit den Eingangsventilsystemen **101**, **201** und **301**, die Säure bereitstellen, den Eingangsventilsystemen **102**, **202**, und **302**, die Base bereitstellen und dem Eingangsventilsystem **103**, **203** und **303**, die zusammen mit dem Wasser-Vorventil **110** dem Reaktorge-

fäß **100** Wasser bereitstellen, Salz produzieren. Die Ausgangsventilsysteme **104**, **204** und **304** können betrieben werden, um das Produkt über eine Abflussleitung in **Fig. 1** nach rechts zu leiten und Abwasser oder ungewolltes Material aus einer Abflussleitung zu leiten, die in den Grund verläuft, wie in **Fig. 1** dargestellt ist.

[0070] Der Regler **12a** ist kommunikativ mit den Ventilsystemen **101–104**, **110**, **201–204** und **301–304** und über den Bus **18** mit den Sensoren **105**, **205** und **305** verbunden, um den Betrieb dieser Elemente zu steuern, in dem eine oder mehrere Operationen mit Bezug zu den Reaktoreinheiten Reaktor_01, Reaktor_02 und Reaktor_03 ausgeführt werden. Solche Operationen, im Allgemeinen Phasen genannt, können z. B. die Befüllung der Reaktorgefäße **100**, **200** und **300** sein, das Heizen des Materials im Reaktorgefäß **100**, **200** und **300**, der Abfluss aus den Reaktorgefäßen **100**, **200** und **300**, das Reinigen der Reaktorgefäße **100**, **200**, **300**, usw.

[0071] Die Ventile, Sensoren und andere Ausrüstung, die in **Fig. 1** dargestellt sind, können beliebige Arten oder Typen von Ausrüstung sein, was z. B. Feldbusdienste, Standard 4-20ma Geräte, HART-Geräte, drahtlose HART-Geräte, usw. einschließt und diese können mit dem Regler **12** kommunizieren (z. B. jedem der Regler **12** oder **12a**), indem sie unter Verwendung jedes bekannten oder gewünschten Kommunikationsprotokolls, wie dem Feldbus-Protokoll, dem HART-Protokoll, einem drahtlosen HART-Protokoll oder anderen drahtlosen Protokollen, dem 4-20ma analogen Protokoll, usw. verwenden. Im Allgemeinen werden Geräte, die sich innerhalb der Prozessumgebung befinden und die eine Funktion ausführen, die die Steuerung des Prozesses direkt beeinflusst (z. Bsp. eine physische Funktion wie das Öffnen oder Schließen eines Ventils, eine Messfunktion, die mit einem Algorithmus oder einer Schleife verwendet wird und/oder andere Funktionen), hier als „Feldgeräte“ bezeichnet.

[0072] Darüber hinaus können andere Gerätetypen in Übereinstimmung mit den hier erläuterten Prinzipien an die Regler **12** angeschlossen und von ihnen gesteuert werden. Ein Regler **12** kann z. B. zu einem oder mehreren Eingangs- oder Ausgangsgeräten (I/O) (nicht dargestellt) angeschlossen sein, die wiederum an ein oder mehrere Feldgeräte angeschlossen sein können. Ein I/O-Gerät wird in der Regel von einem Regler **12** verwendet, um die Kommunikation zwischen einem oder mehreren Feldgeräten, dem Regler **12** und/oder dem Prozesssteuerungssystem zu ermöglichen. Das I/O-Gerät kann an sich auch ein Teilnehmer in der direkten Ausführung eines Steuerungsalgorithmus oder einer -schleife zur Steuerung eines Prozesses sein. Dementsprechend werden Regler, I/O-Geräte und Feldgeräte im Allgemeinen und kategorisch hier als „Prozess-

steuerungsgeräte" bezeichnet. Der Ausdruck „Prozesssteuerungsgerät" ist selbstverständlich nicht nur auf Regler, I/O-Geräte und Feldgeräte beschränkt, sondern kann auch andere Geräte umfassen, die an den Steuerungsalgorithmen und/oder -schleifen teilnehmen, die ausgeführt werden, um einen Prozess in einer verfahrenstechnischen Anlage oder einem Prozesssteuerungssystem zu steuern.

[0073] Zusätzlich kann eine andere Anzahl von Reglern und andere Typen von Reglern innerhalb der verfahrenstechnischen Anlage **10** angeschlossen sein, um andere Geräte oder Bereiche, die mit der verfahrenstechnischen Anlage **10** in Verbindung stehen, zu steuern und der Betrieb solcher Zusatzregler kann, zusammen mit dem Betrieb des Reglers **12a**, der in **Fig. 1** dargestellt ist, auf jede beliebige Weise koordiniert werden. In manchen Verkörperungen enthält die verfahrenstechnische Anlage **10** in **Fig. 1** eine oder mehrere Verbindungsknoten für drahtlose Kommunikation (nicht dargestellt) innerhalb der verfahrenstechnischen Anlage **10**, wie z. B. Zugangspunkte, Gateways zwischen Kabel- und drahtlosen Netzwerken innerhalb der Anlage **10**, Gateways zu anderen Netzwerken, Repeatern, Routern, usw. innerhalb oder außerhalb der Anlage **10** und so weiter. Diese Verbindungsknoten für die drahtlose Kommunikation können kommunikativ (unter Verwendung eines kabelgebundenen Protokolls, eines drahtlosen Protokolls oder einer Kombination aus beiden) an Regler **12**, Arbeitsstationen **14**, die Konfigurationsdatenbank **25**, Feldgeräte, andere Verbindungsknoten mit Drahtlos-Funktion und anderen Datenbanken oder Datenspeichern verbunden sein.

[0074] Im Allgemeinen kann die verfahrenstechnische Anlage **10** aus **Fig. 1** verwendet werden, um Stapelverarbeitungsprozesse einzusetzen, in denen z. B. eine der Arbeitsstationen **14** oder der Regler **12a** eine Stapelverarbeitungsroutine ausführt, die eine hochrangige Steuerungsroutine ist, die den Betrieb einer oder mehrerer Reaktoreinheiten steuert (sowie andere Ausrüstung), um eine Reihe von verschiedenen Schritten auszuführen (üblicherweise Phasen bezeichnet), die benötigt werden um ein Produkt, wie z. B. einen bestimmten Salztyp, herzustellen.

[0075] Um die verschiedenen Phasen zu implementieren, verwendet die Stapelverarbeitungsroutine ein üblicherweise als Rezept bezeichnetes Konzept, das die auszuführenden Schritte, die Beträge und Zeiten, die mit den Schritten verbunden sind und die Reihenfolge der Schritte, spezifiziert. Schritte für ein Rezept können z. B. das Füllen eines Reaktorgefäßes mit den geeigneten Materialien oder Zutaten, das Mischen der Materialien innerhalb des Reaktorgefäßes, das Aufheizen der Materialien innerhalb des Reaktorgefäßes auf eine bestimmte Temperatur für einen bestimmten Zeitraum, das Leeren des Reaktorgefäßes und schließlich die Reinigung des Reaktor-

gefäßes als Vorbereitung für den nächsten Durchgang sein. Jeder dieser Schritte definiert eine Phase des Stapeldurchgangs und die nächste Stapelverarbeitungsroutine innerhalb des Reglers **12a** wird einen anderen Steuerungsalgorithmus für jede dieser Phasen ausführen. Die spezifischen Materialien, die Quantitäten der Materialien, die Temperaturen, Zeiten, usw. können natürlich in verschiedenen Rezepten unterschiedlich ausfallen und daraus folgt, dass sich diese Parameter von Durchgang zu Durchgang ändern können, je nach Produkt, das hergestellt oder produziert wird und je nach Rezept, das verwendet wird. Hierfür versierte Fachleute werden verstehen, dass die Steuerungsroutinen hier für die Stapelverarbeitungsläufe in den Reaktoren aus **Fig. 1** beschrieben werden, aber auch verwendet werden können, um andere beliebige Geräte zu steuern, damit diese beliebige andere Stapelverarbeitungsprozessläufe oder durchgehende Prozessläufe ausführen, wenn dies so gewünscht wird.

[0076] Es ist auch klar, dass dieselben Phasen oder Schritte eines Stapelverarbeitungsprozesses in jeder anderen der verschiedenen Reaktoreinheiten aus **Fig. 1** zur selben Zeit oder zu einer anderen Zeit implementiert werden können. Darüber hinaus kann dieselbe generische Phasensteuerungsroutine einer besonderen Phase auch verwendet werden um jede der verschiedenen Reaktoreinheiten zu steuern, da die Reaktoreinheiten aus **Fig. 1** im Allgemeinen dieselbe Zahl und dieselben Typen an Ausrüstung enthalten, mit der Ausnahme, dass diese generische Phasensteuerungsroutine abgeändert werden muss, um die verschiedenen Hardware- oder Ausrüstungskomponenten, die mit den verschiedenen Reaktoreinheiten in Verbindung stehen, zu steuern. Um zum Beispiel eine Befüllungsphase für Reaktor_01 (in der die Reaktoreinheit befüllt wird) zu implementieren, wird eine Befüllungssteuerungsroutine ein oder mehrere Ventile, die mit dem Eingangsventilsystem **101**, **102** und **103** verbunden sind, für einen gewissen Zeitraum öffnen, z. B. bis der Füllstandsensor **105** erkennt, dass das Gefäß **100** voll ist. Dieselbe Steuerungsroutine kann jedoch auch verwendet werden, um eine Befüllungsphase für den Reaktor_02 zu implementieren, indem einfach die Zuweisung des/der Eingangventile auf jene Ventile geändert wird, die mit den Ventilsystemen **201**, **202** und **203** verbunden sind (anstelle der Ventilsysteme **101**, **102** und **103**) und durch Ändern der Zuweisung des Füllstandsenors von **105** auf **205**.

[0077] **Fig. 2** stellt einen der Reaktoren in **Fig. 1**, insbesondere den Reaktor_01 detailgenauer dar. Wie ebenso in **Fig. 1** dargestellt, schließt Reaktor_01 aus **Fig. 2** den Reaktortank **100**, die Eingangsventilsysteme **101**, **102** und **103** und **110** für den Zulauf von Säure, Base und Wasser in den Tank **100**, das Ausgangsventilsystem **104** zum Auslassen des Materials aus dem Tank **100** und den Füllstandsensor **105**

ein. Wie weiter in **Fig. 2** dargestellt, verwendet jedes der Eingangsventilsysteme **101**, **102** und **110** eine ähnliche Ausrüstungseinheit, die Totalisierer genannt wird, die zwei Ventile, die parallel zueinander ausgerichtet sind, und ein Durchflussmessgerät, das stromabwärts der zwei Ventile angebracht ist, umfassen. Der Totalisierer des Eingangsventilsystems **101**, der in **Fig. 3** genauer dargestellt ist, umfasst ein AN/AUS Ventil, das Grobstromventil **101a** genannt wird, ein AN/AUS Ventil, das Feinstromventil **101b** genannt wird und einen Durchflussmesser **101c**, der stromabwärts der Ventile **101a** und **101b** angebracht ist. Der Totalisierer **101** hat ein oder mehrere Module oder Routinen, die damit assoziiert sind und die verwendet werden, um unter Verwendung der Messungen des Durchflussmessers **101c** den Zufluss an Säure zu steuern. Eine erste solche Steuerungsroutine kann einen schnellen Zufluss durch den Totalisierer **101** durchführen, indem sie das Grobstromventil **101a** und das Feinstromventil **101b** verwendet, während eine zweite solche Steuerungsroutine einen genauen Durchfluss durch den Totalisierer **101** steuert, indem es das Grobstromventil **101a** und das Feinstromventil **101b** verwendet.

[0078] Wie aus **Fig. 2** ersichtlich, umfasst das Basenzuflusssystem **102** einen Totalisierer, der ein Grobstromventil **102a**, ein Feinstromventil **102b** und einen Durchflussmesser **102c** aufweist und das geteilte Wasserzulaufsystem **110** umfasst, ein Grobstromventil **110a**, ein Feinstromventil **110b** und einen Durchflussmesser **110c**. Jeder der Totalisierer **101**, **102** und **110** weist denselben Ausrüstungstyp auf, obwohl sie an unterschiedlichen Orten auf derselben Einheit, d. h. der Reaktor_01-Einheit, verwendet werden. Ebenso verwenden Reaktor_02 und Reaktor_03 Totalisierer im Eingangsventilsystem **201**, **202**, **301** und **302**.

[0079] Das Ausgangsventilsystem **104** ist auf ähnliche Weise ein weiterer Teil einer replizierten Ausrüstung, die drei Ventile enthält. Wie am besten in **Fig. 4** dargestellt, umfasst das Ausgangsventilsystem **104** ein Hauptauslassventil **104a**, das geöffnet werden muss, wenn ein beliebiges Material, das aus dem Tank **100** herausgelassen werden soll, ein Produktventil **104b**, das zusammen mit dem Hauptauslassventil **104a** geöffnet werden muss, um ein Produkt aus dem Tank **100** auszulassen und ein Ablassventil **104c**, das zusammen mit dem Hauptauslassventil **104a** geöffnet werden muss, um Material aus dem Tank **100** in ein Abfluss- oder Sammelsystem abzulassen, wie z. B. Abfallprodukte, Reinigungsflüssigkeiten, usw. Mit dem Ausgangsventilsystem **104** sind selbstverständlich eine oder mehrere Steuerungsroutinen verknüpft, um die Zustände der Ventile **104a**, **104b** und **104c** zu steuern, um den Tank **100** zu schließen, um den Tank **100** zu entleeren oder um das Produkt aus dem Tank **100** abzulassen.

[0080] Um eine Prozesskonfiguration zu erstellen und zu ändern, umfasst eine Konfigurationsanwendung **50**, die in einer der Arbeitsstationen **14** aus **Fig. 1** gespeichert ist, eine Reihe von Modulklassenobjekten **52**, die zur Konfiguration der verfahrenstechnischen Anlage verwendet werden können. Die Modulklassenobjekte sind besonders nützlich, wenn eine Anlage mit mehreren Sets an replizierter Ausrüstung konfiguriert wird. Im Allgemeinen kann für alle verschiedenen Typen an physischen Einheiten oder Ausrüstungen, die repliziert oder innerhalb der verfahrenstechnischen Anlage **10** verwendet werden, für jede Art von Steuerungsaktivität, die repliziert oder innerhalb der verfahrenstechnischen Anlage **10** verwendet wird, für alle verschiedenen Arten von Anzeigeanwendungen, die repliziert oder in der verfahrenstechnischen Anlage **10** verwendet werden, usw., ein anderes Modulklassenobjekt **52** erstellt werden. Sobald es erstellt ist, kann das Modulklassenobjekt **52** verwendet werden, um die Elemente der verfahrenstechnischen Anlage **10** zu konfigurieren, die den Modulklassenobjekten entsprechen.

[0081] Die Modulklassenobjekte **52**, die im Wesentlichen generische Versionen von Prozesseinheiten sind und nicht an eine bestimmte Prozesseinheit gebunden sind, können untergeordnete Objekte oder Instanzen **53**, **54**, **55** und **56** (hier als Modulobjekte oder Modulblocks bezeichnet) mit sich assoziiert haben. Der Ausdruck „Prozesseinheit“, wie er hier verwendet wird, bezieht sich im Allgemeinen auf eine Teilgruppe der verfahrenstechnischen Anlage **10** oder deren Umgebung, die vollständig ermittelt, kategorisiert oder gruppiert sein kann. Eine Prozesseinheit kann z. B. ein physischer Bereich der Anlage, eine Art von Ausrüstung, eine Art von Steuerungsfunktion, eine Gruppe von verwandten Anzeigen oder eine andere Kategorie sein. Prozesseinheiten können auch andere Prozesseinheiten umfassen. Eine Prozesseinheit, die z. B. den „Ventilen“ entspricht, kann untergeordnete Prozesseinheiten, wie z. B. „Gasventile“ oder „Wasserventile“ umfassen und die untergeordnete Prozesseinheit „Wasserventile“ kann noch weiter untergeordnete Prozesseinheiten, wie z. B. „Rückschlagventile“ und „bidirektionale Ventile“ umfassen.

[0082] Wie oben erwähnt, sind Modulklassenobjekte, wie sie hier verwendet werden, im Allgemeinen generisch oder kategorische Hinweise auf Prozesseinheiten. Ein Modulobjekt **53**, **54**, **55**, **56** kann von einem Modulklassenobjekt erstellt oder abgeleitet werden und daher dieselbe Struktur und dieselben Eigenschaften übernehmen, wie das Modulklassenobjekt, von dem es erstellt oder abgeleitet wurde. Jedes Modulobjekt ist jedoch an eine bestimmte Einheit innerhalb der verfahrenstechnischen Anlage **10** gebunden. Deshalb kann ein einzelnes Modulklassenobjekt **52** erstellt werden, um einen bestimmten Typ einer Reaktoreinheit (unabhängig davon wie viele dieser

Reaktoreinheiten innerhalb der Anlage **10** vorhanden sind) zu repräsentieren, während ein anderes Modulobjekt **53** für jeden der verschiedenen Reaktortypen, die tatsächlich in der verfahrenstechnischen Anlage **10** vorhanden sind, existieren oder erstellt werden kann.

[0083] Die Modulobjekte, die von einem Modulklassenobjekt erstellt oder abgeleitet werden, sind mit dem Modulklassenobjekt verknüpft und in dessen Besitz. Dadurch können Änderungen, die am Modulklassenobjekt gemacht werden, in jedem der Modulobjekte, die mit diesem Modulklassenobjekt assoziiert sind, reflektiert oder dorthin weitergegeben werden. Deshalb kann jedes der verschiedenen Modulobjekte verändert werden, indem einfach das Modulklassenobjekt geändert wird und diese Änderungen dann hinunter auf die assoziierten Modulobjekte weitergegeben werden, nachdem eine Anzahl an Modulobjekten von diesem bestimmten Modulklassenobjekt erstellt wurde, wobei jedes der verschiedenen Modulobjekte an verschiedene Prozesseinheiten gebunden ist. Wie später erläutert wird, kann die Weitergabe automatisch erfolgen, wenn das Modulklassenobjekt geändert oder eine bestimmte Zeit für die Weitergabe ausgewählt wird.

[0084] Die Modulklassenobjekte **52** aus **Fig. 1** können das sein, was in einer objektorientierten Programmierumgebung oder Programmiersprache gemeinhin als Objekte bezeichnet wird. Deshalb können diese Objekte andere Objekte besitzen oder darauf hinweisen. Die Modulklassenobjekte **52** sind im allgemeinen übergeordnete Objekte, die Hinweise auf oder Definitionen von einzelnen Elementen enthalten können, wie z. B. Kontrollroutinen, Ausrüstung oder andere Elemente, die mit einer Prozesseinheit zusammen mit einer Definition oder einem Hinweis auf die Art, in der diese einzelnen Elemente miteinander interagieren, verknüpft sind, sowie der Art, auf die physische Elemente untereinander verbunden sind oder der Art, auf die logische Elemente zusammen mit den physischen Elementen operieren. Mit anderen Worten: ein Modulklassenobjekt kann z. B. ein Objekt innerhalb einer objektorientierten Programmiersprache sein, die die Basis für Steuerung und Ansicht eines bestimmten Teils oder einer Gruppe von Ausrüstungen, eines Steuerungselements, einer Ansicht, usw. innerhalb der verfahrenstechnischen Anlage **10** bereitstellt und es kann auch zur Erstellung vieler Instanzen dieses Elements nützlich sein, die verwendet werden, um verschiedene replizierte Ausrüstungen innerhalb der verfahrenstechnischen Anlage **10** zu konfigurieren.

[0085] Jedes Modulklassenobjekt ist im Wesentlichen ein Konfigurationskontainer, der eine generische Definition einer Prozesseinheit in der Form aller auf diese Einheit anwendbaren Steuerungs- und/oder Anzeigenanwendungen, die von den Reglern

12 verwendet werden um diese Einheit zu steuern oder die von den Arbeitsstationen **14** verwendet werden um Anzeigeaktivitäten mit Bezug auf diese Einheit durchzuführen. Das Modulklassenobjekt kann eine Prozesseinheit egal welcher Natur, wie z. Bsp. eine Einheit, ein Ausrüstungsteil, eine Steuerungseinheit, eine Anzeigeanwendung, usw. repräsentieren. Während der Konfiguration der verfahrenstechnischen Anlage **10** kann das Modulklassenobjekt verwendet werden, um Konfigurationsinstanzen der Prozesseinheit für eine beliebig große Zahl an verschiedenen Prozesseinheiten zu erstellen, die mit der Definition, die im Modulklassenobjekt angegeben ist, übereinstimmen, wobei jede Konfigurationsinstanz (das Modulobjekt, das vom Modulklassenobjekt erstellt wurde) mit einer anderen tatsächlichen Prozesseinheit verbunden oder verknüpft ist. Diese verschiedenen Modulobjekte umfassen, unter anderem, Steuerungsrouitinen und/oder Anzeigeroutinen, die an bestimmte Prozesseinheiten gebunden sind, die in der verfahrenstechnischen Anlage **10** vorhanden sind, wobei diese Steuerungsrouitinen auf die Regler **12** aus **Fig. 1** heruntergeladen werden und darauf verwendet werden können, um tatsächliche Steuerungsaktivitäten in den Prozesseinheiten durchzuführen und wobei die Anzeigeroutinen in die Arbeitsstationen **14** heruntergeladen werden und dort tatsächliche Anzeigeaktivitäten im Hinblick auf die Einheiten während des Betriebs der verfahrenstechnischen Anlage **10** durchführen können.

[0086] Verschiedene Typen von Modulklassenobjekte können Prozesseinheiten verschiedenen Ausmaßes widerspiegeln und dadurch auch Steuerungs- und/oder Anzeigeroutinen enthalten, die konfiguriert sind Prozesseinheiten verschiedenen Ausmaßes zu betreiben oder darauf einzuwirken. Je größer das Ausmaß der Prozesseinheit, wie z. B. einer Einheit, ist, desto mehr Steuerungs- und/oder Anzeigeroutinen werden typischerweise mit dem Modulklassenobjekt assoziiert sein und desto einfacher ist es Abschnitte der Anlage zu konfigurieren, indem diese Modulklassenobjekte verwendet werden. Je größer jedoch das Ausmaß der Prozesseinheit ist, die mit einem Modulklassenobjekt verknüpft ist, umso weniger wahrscheinlich ist es auch, dass der Prozess replizierte Ausrüstung in diesem Ausmaß enthält und deshalb ist es auch umso weniger wahrscheinlich, dass das Modulklassenobjekt in großem Ausmaß nützlich ist. Im Gegensatz dazu gilt, dass je kleiner das Ausmaß der Prozesseinheit ist, die mit einem Modulklassenobjekt assoziiert ist, desto wahrscheinlicher kann das Modulklassenobjekt an verschiedenen Orten der Anlage verwendet werden, aber umso weniger Konfiguration wird ausgeführt, wenn dieses Modulklassenobjekt in einer bestimmten Instanz verwendet wird. Mit Modulklassenobjekten können auf jeden Fall verschiedene replizierte Ausrüstungen auf übergeordneten Ebenen der Abstrahierung konfiguriert werden, was das Konfigurieren einer verfahrenstechnischen

Anlage mit replizierten Einheiten und anderer Ausrüstung einfacher und weniger zeitaufwändig macht, wenn Modulklassenobjekte verwendet werden, besonders bei Modulklassenobjekten großen Ausmaßes, wie z. Bsp. auf Einheitsebene.

[0087] Daraus folgt, dass mehrere Objektebenen möglich sind. Objekte, die den Instanzen **53**, **54**, **55**, **56** entsprechen und von den Modulklassenobjekten **52** (z. B. „Instanzobjekte“) erstellt wurden, können selbst Elternobjekte für einen Satz von einem oder mehreren Instanz-Kindobjekten sein (nicht dargestellt). Eines oder mehrere der Instanz-Kindobjekte kann wiederum ein Elternobjekt von einem weiteren Niveau von Kindobjekten sein, usw. „Prozesselementobjekt“ bezieht sich, so wie es hier verwendet wird, im Allgemeinen auf ein niedrigstes Objektniveau, das einer elementaren Prozesseinheit entspricht, auf welche eine Konfiguration heruntergeladen wird, wie z. B. ein Ventil, ein Sensor, eine grafische Form oder ein Regler. So kann ein Prozesselementobjekt demzufolge ein Instanzobjekt sein, das keine Kindobjekte hat.

[0088] Wenn ein Konfigurationsentwickler z. B. ein Prozesssteuerungssystem konfiguriert, kann er ein einzelnes Modulklassenobjekt für verschiedene Elemente, die in der verfahrenstechnischen Anlage repliziert sind, erstellen, wie z. B. für die verschiedenen Reaktoren aus **Fig. 1** dargestellt ist. Danach kann der Konfigurationsentwickler Instanzen des Modulklassenobjekts (Modulobjekte) für jeden der eigentlichen Reaktoren aus **Fig. 1** dargestellt ist. Jedes so erstellte Modulobjekt umfasst Steuerungsroutinen, die vom Regler **12a** verwendet werden, um einen der Reaktoren aus **Fig. 1** zu steuern und ist spezifisch an die Ausrüstung innerhalb dieses Reaktors aus **Fig. 1** dargestellt ist. Diese Steuerungsroutinen können dann auf den Regler **12a** heruntergeladen und während des Betriebs der verfahrenstechnischen Anlage **10** verwendet werden. Einmal erstellt, ist jedes Modulobjekt jedoch immer noch an das Modulklassenobjekt gebunden und kann über das Modulklassenobjekt gesteuert zu werden, um geändert zu werden, um einen Zugriff auf das Modulobjekt zu gewähren oder zu verweigern, usw.

[0089] Während es viele verschiedene mögliche Typen von Modulklassenobjekten gibt, die erstellt oder in einer verfahrenstechnischen Anlage verwendet werden können, um Aktivitäten innerhalb der verfahrenstechnischen Anlage auszuführen, umfassen vier bestimmte, hier beschriebene Typen z. B. Einheitenmodulklassenobjekte, Ausrüstungsmodulklassenobjekte, Steuerungsmodulklassenobjekte und Anzeigemodulklassenobjekte. Im Allgemeinen wurde jeder andere Typ an Modulklassenobjekten für ein anderes Ausmaß an Kontrolle oder eine andere Verwendung innerhalb der verfahrenstechnischen Anlage **10** entwickelt oder vorgesehen. Ein Einheitenmodulklas-

senobjekt ist dafür vorgesehen, verwendet zu werden, um Steuerungsaktivitäten für ein weites Ausmaß an Ausrüstungen innerhalb einer verfahrenstechnischen Anlage zu repräsentieren (und zu konfigurieren). Ein Einheitenmodulklassenobjekt ist besonders als Modell oder dafür bestimmt, um untereinander verbundene Ausrüstungen (typischerweise replizierte Ausrüstungen), wie z. B. die Reaktoren aus **Fig. 1** zu konfigurieren, die einzelne Elemente enthalten, die miteinander auf bekannte Weise zusammenarbeiten.

[0090] Ein Ausrüstungsmodulklassenobjekt ist dafür bestimmt, um Steuerungsaktivitäten für ein weniger breites Ausmaß an physischer Ausrüstung innerhalb der verfahrenstechnischen Anlage zu repräsentieren (und zu konfigurieren). Die Ausrüstungen, die mit einem Ausrüstungsmodulklassenobjekt verknüpft sind, sind im allgemeinen eine oder mehrere physische Einheiten, wie Ventile, Durchflussmesser, usw., die ein Subsystem einer Einheit ausmachen und das Ausrüstungsmodulklassenobjekt kann eines oder mehrere Befehle oder Algorithmen enthalten, die befehlsgesteuerte Algorithmen (CDAs), zustandsgesteuerte Algorithmen (SDAs), sequentielle Funktionstabellenalgorithmen (SFC), Funktionsblockdiagrammalgorithmen (FBD), Phasenalgorithmen, usw. sein können, die auf dem Ausrüstungsteil ausgeführt werden sollen. Ein Ausrüstungsmodulklassenobjekt ist also dahingehend ausgelegt, um die Steuerung von mehreren untergeordneten Komponenten oder Einheiten innerhalb einer Einheit zu steuern, um eine Reihe von Grundfunktionen auf dieser Ausrüstung zu bieten, wie es innerhalb der Einheit verwendet wird. Bekanntlicherweise wird ein befehlsgesteuerter Algorithmus (eine befehlsgesteuerte Steuerungslogik) verwendet, wenn die untergeordneten Komponenten über mehrere Schritte hinweg koordiniert werden müssen, um eine Funktion zu erfüllen. Ein Ventil kann z. B. für einen bestimmten Zeitraum geöffnet werden müssen und dann geschlossen werden müssen, während ein anderes Ventil geöffnet und dann geschlossen wird. Der Totalisierer **101** aus **Fig. 3** nutzt diese Art von befehlsgesteuertem Algorithmus, um das Grobstrom- und das Feinstromventil zuerst zu initiieren und dann zu manipulieren, je nach den Messungen des Durchflussmessers, um den gewünschten Durchfluss durch den Totalisierer zu liefern. Ein zustandsgesteuerter Algorithmus (zustandsgesteuerte Steuerungslogik) kann die Zustände verschiedener untergeordneter Komponenten angeben, die in einem einzigen Schritt manipuliert werden können. Solch ein zustandsgesteuerter Algorithmus kann im Ausgangsventilsystem **104** von **Fig. 4** verwendet werden, in dem die Zustände der verschiedenen Ventile darin unterschiedlich (aber in einem einzigen Schritt) gesteuert werden, je nach dem gewünschten Zustand des Ausgangsventilsystems **104** um den Tank **100** zu schließen, um den Tank **100** zu

entleeren, oder um das Produkt aus dem Tank **100** auszulassen.

[0091] Ein Steuerungsmodulklassenobjekt ist dazu vorgesehen, um ein einzelnes Steuerelement oder Steuermodule innerhalb der verfahrenstechnischen Anlage zu repräsentieren (und zu konfigurieren). Ein Steuerungsmodulklassenobjekt bietet oder spezifiziert einen bestimmten Typ an Steuerung, die in einer Anlageneinheit ausgeführt werden muss, wie z. B. einem Ventil, einem Messgerät, usw., einem Ausrüstungsteil oder sogar an einer Einheit. Im Allgemeinen bietet ein Steuerungsmodulklassenobjekt einen bestimmten Typ an Steuerungsprogrammierung, wie z. B. eine Reihe von kommunikativ verbundenen Funktionsblöcken, die ein Steuerungsmodul definieren, das in einem Regler ausgeführt werden soll, was nützlich ist, um replizierte Steuerungsaktivitäten in einer verfahrenstechnischen Anlage durchzuführen. In den meisten Fällen kann ein Steuerungsmodulklassenobjekt eine generische Steuerungsstrategie anbieten, um ein einzelnes Gerät oder eine verbundene Reihe von Geräten zu manipulieren.

[0092] Ein Anzeigemodulklassenobjekt ist dazu vorgesehen, verwendet zu werden, um Anzeigeaktivitäten, die von einem Nutzer, wie z. B. einem Bediener, während des Betriebs der verfahrenstechnischen Anlage **10** betrachtet werden, zu repräsentieren (und zu konfigurieren). Ein Anzeigemodulklassenobjekt kann also die Programmierung spezifizieren, die benötigt wird, um in einer Bedienarbeitsstation **14** eine Anzeige eines bestimmten Typs zu erzeugen, wie in **Fig. 1** und um die Programmierung spezifizieren, die in einer oder mehreren der Arbeitsstationen **14** (sowie in anderen Geräten der verfahrenstechnischen Anlage **10**) benötigt wird, um es der Anzeige zu ermöglichen während des Betriebs der Anlage **10** geeignete Informationen der Anlage zu erhalten. Zu Anzeigeklassenmodultypen zählen beispielsweise. Alarmanzeigen, Konfigurationsbetrachtungsanzeigen, Betriebsüberwachungsanzeigen, diagnostische Anzeigen, usw. Ein Anzeigenmodulklassenobjekt kann natürlich auch eine Anzeige bereitstellen, die mit jedem beliebigen Ausmaß an physischen Elementen oder Einheiten innerhalb einer verfahrenstechnischen Anlage verbunden ist. Ein Anzeigenmodulklassenobjekt kann z. B. Informationen über einen gesamten Bereich, eine Einheit, ein Ausrüstungsteil, ein Steuerelement oder jede Kombination dieser Elemente innerhalb der verfahrenstechnischen Anlage **10** anzeigen.

[0093] In Bezug auf **Fig. 5** veranschaulicht eine hierarchische Grafik die Verbindungen zwischen verschiedenen Arten von Modulklassenobjekten, die in der Konfigurationsanwendung **50** von **Fig. 1** angewendet werden sowie auch die Wechselbeziehung zwischen Modulklassenobjekten und den auf der Grundlage dieser Modulklassenobjekte entwickelten Modulobjekten. Wenn wir oben auf der Grafik von

Fig. 5 beginnen, werden die Modulklassenobjekte (oder "Klassenobjekte", wie auf sie hier austauschbar verwiesen wird) nach Modulklassentypen in eines der Einheitsmodule Klassentyp **400**, Gerätemodulklassentyp **402**, Steuerungsmodulklassentyp **404** und Anzeigemodulklassentyp **406** getrennt. Selbstverständlich können andere Typen von Modulklassenobjekten bereitgestellt bzw. eingesetzt werden. Die vier hier veranschaulichten Typen stellen lediglich beispielgebende Modulklassentypen dar. Einzelne Modulklassenobjekte (die Objekte einer oberen Stufe sein können, beispielsweise eine objekt-orientierte Programmierungssprache und die in **Fig. 5** deutlichheitshalber mit einem doppelten Umriss dargestellt sind) fallen unter die verschiedenen Modulklassentypen **400**, **402**, **404** und **406**. Insbesondere kann es zahlreiche verschiedene Einheitsmodulklassenobjekte für verschiedene Einheiten oder Einheitstypen in der Betriebsanlage **10** geben. Beispielsweise kann das Reaktoreinheitklassen-Modulobjekt **410** einen bestimmten Typ bzw. Konfiguration eines Reaktors innerhalb der Betriebsanlage **10** darstellen. Gleichmaßen kann das Packeinheits-Modulklassenobjekt **412** einen bestimmten Typ bzw. Konfiguration einer Packeinheit innerhalb der Betriebsanlage **10** darstellen und das Trocknereinheitsklassen-Modulobjekt **414** kann einen bestimmten Typ bzw. Konfiguration einer Trocknereinheit innerhalb der Betriebsanlage **10** darstellen. Natürlich kann es mehr als ein Reaktoreinheits-Modulklassenobjekt zur Darstellung der Reaktoren geben, die sich physisch untereinander unterscheiden. Des Weiteren wurden keine Versuche unternommen, die verschiedenen Einheitstypen innerhalb eines Betriebs aufzuführen, die mit einem Einheitsmodulklassenobjekt dargestellt oder damit modelliert werden können und der normale Fachmann erkennt, dass es viele verschiedene Einheitstypen von Betrieben gibt, die mit Hilfe der Einheitsmodulklassenobjekten modelliert oder dargestellt werden können.

[0094] Ebenso kann es viele verschiedene Gerätemodulklassenobjekte geben, die zur Darstellung, Modellierung und Konfiguration verschiedener Gerätetypen innerhalb der Betriebsanlage **10** eingesetzt werden. Die in **Fig. 5** veranschaulichten Instanzen weisen das Totalisatorgeräte-Modulklassenobjekt **416** und das Auslassventil-Gerätemodulklassenobjekt **418** auf, wobei jedes davon mit verschiedenen Gerätetypen (und vorzugsweise vervielfältigten Geräten) innerhalb der Betriebsanlage **10** verbunden ist. Auf gleiche Art und Weise kann es viele verschiedene Typen an Steuerungsmodulklassenobjekten geben, die in **Fig. 5** als das Ein-/Aus-Ventilsteuerungs-Modulklassenobjekt **422**, das Niveaufühlersteuerungs-Modulklassenobjekt **424** und das Durchsatzmessersteuerungs-Modulklassenobjekt **426** veranschaulicht sind. Außerdem sind Anzeigemodulklassenobjekte in **Fig. 5** als das Alarmanzeige-Modulklassenobjekt **432**, das Betrachtungsanzeige-Modulklassenobjekt

434 und das Diagnoseanzeige-Modulklassenobjekt **436** veranschaulicht. Natürlich kann jedes sonstige gewünschte Einheits-, Geräte-, Steuerungs- und Anzeigemodulklassenobjekt angelegt und innerhalb der Konfigurationsanwendung **50** der Betriebsanlage **10** in Einklang mit den hier beschriebenen Grundsätzen verwendet werden.

[0095] Jedes Modulklassenobjekt kann Unterobjekte haben, die damit verbunden bzw. ihnen zugeordnet sind. Diese Unterobjekte können Modulklassenobjekte an sich oder, wie in **Fig. 5** veranschaulicht, Modulobjekte sein, die als Instanzen der Modulklassenobjekte erstellt wurden, zu denen sie gehören. Auf dieserart Instanzen wird hier austauschbar als auf die "Modulobjekte", "Instanzen", "Instanzobjekte" oder "Modulinstantzobjekte" des Modulklassenobjekts verwiesen, auf denen sie basieren bzw. von denen sie erstellt wurden. **Fig. 5** veranschaulicht, dass das Reaktoreinheits-Modulklassenobjekt **410** über drei Reaktormodulobjekte namens Reaktor_01 (Verweis **440a**), Reaktor_02 (Verweis **440b**) und Reaktor_03 (Verweis **440c**) verfügt, die ihm zugeordnet wurden, wobei diese Reaktormodulobjekte dem jeweiligen Reaktorklassenobjekt **410** von **Fig. 1** entsprechen (bzw. damit verbunden sind). **Fig. 5** veranschaulicht ebenfalls das Totalisatorgeräte-Modulklassenobjekt **416** als ein Objekt, das fünf verschiedene untergeordnete Modulobjekte hat bzw. besitzt und zwar mit den Namen Wasser1, Säure1, Säure2, Alkali1 und Alkali2 (Verweis **440d–440h**). Zugleich wird das Ein-/Aus-Ventilsteuerungs-Modulklassenobjekt **422** so veranschaulicht, dass die untergeordneten Modulobjekte namens Grob_Ventil1, Grob_Ventil2, Grob_Ventil3, Fein_Ventil1, Fein_Ventil2 und Fein_Ventil3 (Verweis **440i–440n**) aufweist. Außerdem veranschaulicht **Fig. 5** das Alarm_A Grafikelementobjekt **440o** auf der Grundlage des Alarmanzeigemodulobjekts **432**, des Temp_Sensor_B Grafikelements **440p** und des Steuerung_Modul_C Grafikelements **440q** basiert auf dem Betrachtungsanzeigemodulobjekt **434** und dem Test_Modul_D Grafikelement **440r** und dem Pumpen_E Grafikelement **440s** basierend auf dem Diagnosenanzeigemodul **436**. Auf ähnliche Art und Weise kann jede der anderen Einheits-, Geräte-, Steuerungs- und Anzeigemodulklassenobjekte von **Fig. 5** über eine oder mehr damit verbundene Modulobjekte verfügen. Aus Einfachheitsgründen werden diese Modulobjekte nicht in **Fig. 5** veranschaulicht.

[0096] In der Grafik von **Fig. 5** stellen jeder der Reaktor_01, Reaktor_02 und Reaktor_03 Einheitmodulobjekte, der Säure1, Säure2, Alkali1, Alkali2 und Wasser1 Totalisator (Geräte) Modulobjekte (Verweis **440a–h**), der Grob_Ventil1, Grob_Ventil2, Grob_Ventil3, Fein_Ventil1, Fein_Ventil2 und Fein_Ventil3 Steuerungsmodulobjekte (Verweis **440i–440n**), der Alarm_A, Temp_Sensor_B, Steuerung_Modul_C, Test_Modul_D, Pump_E Grafikelementobjekte

(Verweis **440o–440s**) und der anderen Einheit-, Geräte-, Steuerungs- und Anzeigemodulobjekte einzelne Objekte dar, die mit den effektiven Prozesselementen innerhalb der Betriebsanlage **10**, wie beispielsweise Einheiten, Geräten, Steuerungsmodulen, Anzeigeanwendungen und graphischen oder Grafikanzeigeelementen innerhalb der Betriebsanlage **10** verbunden sind. In diesem Sinne wird sich auf die Objekte **440a–440s** austauschbar als auf die "Prozesselementobjekte", "Prozesselement-Modulobjekte" oder die "Elementobjekte" bezogen. Gleichmaßen ist jedes der Objekte **440a–440s** ein "untergeordnetes" oder ein "untergeordnetes Objekt" seiner jeweiligen "übergeordneten" bzw. "übergeordneten Objekte" **410–436**. Beispielsweise gibt es aufgrund vielfacher physischer im Betrieb **10** eingesetzten Säuretotalisatoren mehrfache Säuretotalisatorprozesselement-Modulobjekte, die in der Konfigurationsroutine angelegt werden und wobei es ein separates untergeordnetes Säuretotalisatorprozesselement-Modulobjekt für jeden der einzelnen Säuretotalisatoren gibt, der innerhalb von Betrieb **10** vorhanden ist. Jedoch ist jedes der separaten untergeordneten Totalisatorprozesselement-Modulobjekte an dasselbe übergeordnete Objekttotalisator-Modulklassenobjekt **416** gebunden oder gehört zu ihm. Natürlich veranschaulicht die Grafik von **Fig. 5** nur eine beschränkte Anzahl damit verbundener Modulklassenobjekte, Modulobjekte, Instanzobjekte und Prozesselementobjekte und es versteht sich, dass andere Typen von Modulklassenobjekten vorgesehen werden können und dass eine beliebige gewünschte Anzahl von Modulobjekten, Instanzobjekten und Prozesselementobjekten von jedem der verschiedenen Modulklassenobjekte angelegt werden kann.

[0097] Des Weiteren kann ein Objekt, dass ein untergeordnetes Objekt eines übergeordneten Objekts ist, selbst über untergeordnete Objekte verfügen. Beispielsweise kann das Klassenobjektdurchsatzmesser-Steuerungsmodul **426** zwei untergeordnete Instanzobjekte, beispielsweise "Wasser_Durchsatz_Messermodul" und "Lösemittel_Durchsatz_Messermodul" (nicht veranschaulicht) aufweisen. Das Wasser_Durchsatz_Messermodul kann die jeweiligen untergeordneten Prozesselement-Modulobjekte umfassen, die den jeweiligen effektiven Durchsatzmesserelementen innerhalb der Betriebsanlage **10**, wie beispielsweise der "Wasser_Durchsatz_Messer_1" und "Wasser_Durchsatz_Messer_2" entsprechen. In diesem Sinne basieren die Prozesselementobjekte "Wasser_Durchsatz_Messer_1" und "Wasser_Durchsatz_Messer_2" auf dem "Wasser_Durchsatz_Messer Modul", das auf dem Durchsatzmesser-Steuerungsmodul **426** basiert.

[0098] Jedes der Modulklassenobjekte von **Fig. 5** (und daher jedes der Modulobjekte von **Fig. 5**) kann als Teil eines Objekts eine Definition oder eine Anzei-

ge der physischen oder logischen Prozesselemente enthalten, die das Modul definieren oder ausmachen und, falls gewünscht, die Art enthalten, in der diese Prozesselemente entweder physisch oder logisch aufeinander einwirken, um eine Aktivität innerhalb der Betriebsanlage **10** durchzuführen. Beispielsweise ist in Einheitsmodulklassenobjekt in der Regel eine Anzeige aller physischer und Steuerungselemente darin vorgesehen oder es macht die als die Einheit definierte Prozessfunktionseinheit aus. Das Einheitsmodulklassenobjekt kann ebenfalls die bestimmte Zusammensetzung der einzelnen Teile definieren sowie auch, wie diese Teile physisch miteinander verbunden sind, um als eine Einheit zu funktionieren. Gleichmaßen enthält ein Gerätemodulklassenobjekt üblicherweise die zur Steuerung der Einheit verwendeten Steuerungsroutinen oder Steuerungsmodule, die als der Teil der Geräte definiert sind, und die befehlen, welche die Steuerungsroutinen oder Steuerungsmodule verwenden, um die Art und Weise zu definieren, in der die Teile entweder physisch oder logisch aufeinander einwirken, um als ein Teil der Geräte zu wirken, wenn sie innerhalb von Betrieb **10** angeordnet sind. In ähnlicher Weise definiert jedes Steuerungsmodulklassenobjekt eine Steuerungsaktivität, generell in Gestalt einer Art Steuerungsalgorithmus, die innerhalb des Betrieb auszuführen ist. Des Weiteren kann jedes Anzeigemodulklassenobjekt u. a. die Anzeigebildschirmkonfiguration, die anzuzeigenden Informationen und graphische oder Graphik-elemente definieren, die verschiedene Elemente der Betriebsanlage **10** darstellen und die auf dem Anzeigebildschirm darzustellen sind, sowie auch die zu sammelnden Daten und mittels dieser Daten erforderlichenfalls auszuführenden Datenbearbeitungen für vorgegebene Einheits-, Gerätetypen, Betriebsbereiche oder beliebig andere physische oder logische Einheiten in Betrieb **10**.

[0099] Im Rahmen der Modulklassendefinition kann ein Modulklassenobjekt andere hier aufgenommene oder verwendete Modulklassenobjekte anzeigen oder definieren. Wenn dies der Fall ist, nehmen die von diesem Modulklassenobjekt erstellten Modulobjekte andere von anderen Modulobjekten in Einklang mit den auf der Modulklassenstufe definierten Beziehungen erstellten Modulklassenobjekte auf, nehmen auf sie Bezug oder schließen sie ein. Obwohl es nicht unbedingt erforderlich ist, können Einheitsmodulklassenobjekte andere Einheitsmodulklassenobjekte, Gerätemodulklassenobjekte, Steuerungsmodulklassenobjekte und Anzeigemodulklassenobjekte übernehmen, während Gerätemodulklassenobjekte andere Gerätemodulklassenobjekte, Steuerungsmodulklassenobjekte und Anzeigemodulklassenobjekte übernehmen können. Steuerungsmodulklassenobjekte können andere Steuerungsmodulklassenobjekte und Anzeigemodulklassenobjekte übernehmen und auf sie Bezug nehmen. Jedoch können auf Wunsch auch andere Wechselbeziehungen unter

Modulklassenobjekten angewendet werden. Diese Übernahmebeziehungen sind mit Hilfe der großen Pfeile weiter unten in der Grafik von **Fig. 5** veranschaulicht, die anzeigen, dass beliebige Anzeigemodulklassenobjekte von beliebigen Steuerungs-, Geräte- und Einheitsmodulklassenobjekte übernommen werden können bzw. auf sie Bezug genommen werden kann, dass beliebige Steuerungsmodulklassenobjekte von beliebigen Geräte- und Einheitsmodulklassenobjekten übernommen oder auf sie Bezug genommen werden kann, und dass beliebige Gerätemodulklassenobjekte von beliebigen Einheitsmodulklassenobjekten übernommen bzw. auf sie Bezug genommen werden kann. Es versteht sich, dass Modulklassenobjekte andere Modulklassenobjekte des selben Typs übernehmen können. So kann beispielsweise ein Einheitsmodulklassenobjekt im Rahmen seiner Definition ein anderes Einheitsmodulklassenobjekt übernehmen. Auf ähnliche Art und Weise kann ein Gerätemodulklassenobjekt ein anderes Gerätemodulklassenobjekt aufweisen, ein Steuerungsmodulklassenobjekt kann ein anderes Steuerungsmodulklassenobjekt aufweisen, und ein Anzeigemodulklassenobjekt kann ein anderes Anzeigemodulklassenobjekt aufweisen. Auf Wunsch kann ein Modulklassenobjekt selbstverständlich ein anderes Modulklassenobjekt mehrere Male verwenden oder übernehmen. Beispielsweise kann ein Reaktoreinheits-Modulklassenobjekt ein Totalisatorgeräte-Modulklassenobjekt mehrmals übernehmen oder verwenden, da die von dem Reaktoreinheits-Modulklassenobjekt modellierten Reaktoren mehrfache Instanzen von Totalisatoren aufweisen.

[0100] Des Weiteren versteht sich, dass im Falle der Übernahme oder des Gebrauchs eines zweiten Modulklassenobjekts von einem ersten Modulklassenobjekts, ein beliebiges von oder als eine Instanz des ersten Modulklassenobjekts erstelltes Modulobjekt, ein Modulobjekt übernimmt oder einsetzt, das von oder als eine Instanz des zweiten Modulklassenobjekts erstellt wurde. Wenn daher das Reaktoreinheits-Modulklassenobjekt **410** das Totalisator-Modulklassenobjekt **416** als ein Element oder einen Teil verwendet, verwendet das Reaktor_01 Modulobjekt eines der Totalisator-Modulobjekte, wie beispielsweise Säure1 Modulobjekt **440e**, als ein Element oder als einen Teil davon. Wenn in ähnlicher Weise ein Totalisator-Gerätemodulklassenobjekt ein Auslassventil-Gerätemodulklassenobjekt enthält oder übernimmt, wird ein vom Totalisator-Gerätemodulklassenobjekt erstelltes Modulobjekt eindeutig beispielsweise Totalisator_1 genannt und wird ein vom Auslassventil-Gerätemodulklassenobjekt erstelltes Modulobjekt aufweisen und eindeutig beispielsweise Auslass_Ventil_2 genannt werden. Auf diese Art und Weise werden die Beziehungen zwischen den Modulklassenobjekten gemäß der Definition auf der Modulklassenobjektstufe in den Modulobjekten widergespiegelt, welche von diesen Modulklassenobjekten aus entwi-

ckelt oder erstellt wurden. Diese gegenseitige Verbindung bzw. dieses Herstellen von Bezügen zwischen Modulklassenobjekten (und daher Modulobjekten) ermöglicht eine hohe Variabilität und Übertragbarkeit von Objekten während der Konfigurationsaktivitäten, sodass nach der Erstellung eines Satzes einfacher Modulklassenobjekte, wie beispielsweise Steuerungs- und Gerätemodulklassenobjekte, komplexere Modulklassenobjekte, wie beispielsweise Einheitsmodulklassenobjekte, mühelos durch das Referenzieren der einfachen Modulklassenobjekte angelegt werden können. Obwohl Modulklassenobjekte andere Modulklassenobjekte natürlich referenzieren oder einsetzen können, können sie auch stattdessen einfache Objekte oder Prozesselementobjekte, wie beispielsweise Ventile, Sensoren, usw. definieren oder einsetzen, die über kein zugeordnetes Modulklassenobjekt verfügen. Diese einfachen oder Prozesselementobjekte werden vollständig im Sinne der dafür verwendeten Steuerungsroutinen innerhalb des Modulklassenobjekts selbst definiert.

[0101] Das beispielhafte Reaktoreinheits-Modulklassenobjekt **410** ist in **Fig. 6** veranschaulicht, um die Art und Weise des Beschreibens oder Definierens der Einheiten zu zeigen, die einem Einheitsmodulklassenobjekt zugeordnet wurden oder in diesem vorhanden sind. Wie in **Fig. 6** veranschaulicht, weist das Reaktoreinheits-Modulklassenobjekt **410** eine Anzeige des Tanks **500** auf, was ein einfaches Objekt oder Prozesselementobjekt in der Betriebsanlage **10** ist, für das kein Modulklassenobjekt vorhanden ist. Der Tank **500** ist auf den gepunkteten Linien veranschaulicht, da kein Bedürfnis zur Steuerung oder Durchführung von Niedrigstufenaktivitäten zur Steuerung oder Durchführung von Eingabe-/Ausgabe-Aktivitäten in Bezug auf den Tank besteht. Daher wird der Tank **500** lediglich mit aufgenommen, um die Verbindungen zwischen anderen Objekten zu veranschaulichen, die mit Reaktoreinheits-Modulklassenobjekt **410** in Zusammenhang stehen. Das Reaktoreinheits-Modulklassenobjekt **410** kann ebenfalls drei Totalisatoren **501**, **502** und **510** mit den jeweiligen Namen Säure, Alkali und Wasser aufweisen, die drei verschiedene Verweise auf das Totalisatorgeräte-Modulklassenobjekt **416** von **Fig. 5**. darstellen. Das Wassertotalisator-Modulklassenobjekt **510** veranschaulicht in einen Abschnitt des Einheitsmodulklassenobjekts **410**, dass durch gepunktete Linien getrennt ist, um anzuzeigen, dass es sich hier um ein gemeinsam genutztes Modulklassenobjekt handelt und dass das Einheitsmodulklassenobjekt **410** daher über eine mit anderen Einheitsmodulklassenobjekten gemeinsam verwendete Steuerung über dieses Objekt verfügt. Das Auslassobjekt **504** von **Fig. 6** ist ein Verweis auf das Auslassventilgeräte-Modulklassenobjekt **418** von **Fig. 5**, der Niveaufühler **505** ist ein Verweis auf das Niveaufühler-Steuerungsmodulklassenobjekt **424** von **Fig. 5**, und das Wasser_in Ventil **503** ist ein Verweis auf ein Ventilobjekt, das ein

einfaches Ventilelement (und daher vollständig innerhalb des Einheitsmodulklassenobjekts **410** definiert ist) oder ein Verweis auf ein Ventil-Steuerungsmodulklassenobjekt sein kann, das anderswo in der Konfigurationsstrategie definiert ist. Die physischen Verbindungen zwischen verschiedenen Einheiten oder Teilen des Reaktoreinheits-Modulklassenobjekts **410** sind ebenfalls zur Definition der Verbindungen zwischen diesen verschiedenen Elementen veranschaulicht. Wie weiter oben erwähnt, kann das Einheitsmodulklassenobjekt **410** oder andere Modulklassenobjekte beliebiger Typen einfache Elemente aufweisen, die innerhalb des Modulklassenobjekts vollständig definiert sind (einschließlich beliebiger generischer damit verbundenen Steuerungsroutinen) und/oder auch Verweise auf Modulklassenobjekte aufweisen, die außerhalb des Modulklassenobjekts definiert sind.

[0102] Das Einheitsmodulklassenobjekt **410** weist beispielsweise ebenfalls zwei Anzeigemodulklassenobjekte namens Reaktorbetriebsanzeige **520** und Reaktoralarmanzeige **522** auf, die Verweise auf das Betrachtungsanzeige-Modulklassenobjekt **434** und Alarmanzeige-Modulklassenobjekt **432** von **Fig. 5**. sind. Diese Objekte definieren generische Anzeigeaktivitäten zur Anzeige des Stands (beispielsweise des Füllstands des Tanks) und der Alarme, die mit beliebigen Geräten oder Teilen der Reaktoreinheit in Zusammenhang stehen, die in dem Reaktoreinheits-Modulklassenobjekt **410** definiert sind. In ähnlicher Weise kann das Einheitsmodulklassenobjekt **410** andere Elemente aufweisen, wie beispielsweise Phasenklassenobjekte, die in Kasten **524** als ein Dosis-, ein Mischungs-, ein Abfluss- und ein Spülphasenklassenobjekt veranschaulicht sind, wobei jedes davon eine generische auf der Einheit ausgeführte Steuerungsroutine definiert, die von dem Einheitsmodulklassenobjekt **410** definiert wurde. Das Einheitsmodulklassenobjekt kann Null oder mehr Zuordnungen zu Phasenklassenobjekten haben. Die Phasenklassenobjekte **524** können anderswo definiert werden und in das Einheitsmodulklassenobjekt **410** auf beliebig gewünschte Art und Weise importiert werden. In gewisser Weise stellen die Phasenklassen **524** Befehle oder Routinen dar, die auf einer Einheit ausgeführt werden können, die von dem Einheitsmodulklassenobjekt **410** zur Durchführung verschiedener Funktionen definiert wurde, wie beispielsweise das Auffüllen der Einheit, Anwärmen der Einheit, Ausleeren der Einheit, die Reinigung der Einheit, usw.

[0103] Außerdem kann das Einheitsmodulklassenobjekt **410** einen Speicher oder Abschnitt **526** aufweisen, der Verweise auf Modulklassenobjekte speichert, die von der Konfigurationsanwendung **50** (**Fig. 1**) dieses Einheitsmodulklassenobjekts **410** erstellt wurden. Der Abschnitt **526** ist im Grunde genommen eine Liste der Modulobjekte, die vom Ein-

heitsmodulklassenobjekt **410** erstellt wurden und zu diesem gehören. (Natürlich können diese Liste oder sonstige Anzeigen der im Besitz befindlichen Modulobjekte auf einem Arbeitsplatzgerät oder von Konfigurationsanwendung **50** in beliebig gewünschter Art und Weise gespeichert werden und brauchen nicht physisch im Einheitsmodulklassenobjekt **410** enthalten zu sein). Auf jeden Fall ist in Beispiel von Fig. 6 das Einheitsmodulklassenobjekt **410** in Besitz der Modulklassenobjekte Reaktor_01 **440a**, Reaktor_02 **440b**, Reaktor_03 **440c**, usw., wobei jedes davon vom Reaktoreinheits-Modulklassenobjekt **410** erstellt wurde.

[0104] Des Weiteren weist das Einheitsmodulklassenobjekt **410** einen Satz von Methoden **530** auf, der von dem Einheitsmodulklassenobjekt **410** entweder während oder nach den Konfigurationsaktivitäten ausgeführt werden kann. Die Methoden **530** können eine Veränderungsverwaltungsmethode oder -anwendung aufweisen, die automatisch alle Änderungen propagiert, die am Einheitsmodulklassenobjekt **410** an jedem der Modulobjekte **526** vorgenommen wurden, welche sich im Besitz eines Einheitsmodulklassenobjekts **410** befinden. Sonstige Methoden können Sicherheitssteuerungsmethoden aufweisen, die eine Sicherheits- oder Zugriffssteuerung in Bezug auf das Einheitsmodulklassenobjekt **410** und/oder in Bezug auf beliebige zu ihnen gehörende Einheitsmodulobjekte **526** ausführen, bzw. Methoden, die es dem Anwender oder Konfigurationsingenieur ermöglichen, Änderungsparameter und/oder Sicherheitsparameter für das Modulklassenobjekt oder beliebige davon erstellte Modulobjekte festzulegen. Natürlich können verschiedene Methoden **530** beliebige andere Vorgänge auf dem oder in Bezug auf das Einheitsmodulklassenobjekt **410** ausführen.

[0105] Auf Wunsch kann das Einheitsmodulklassenobjekt **410** die Art und Weise steuern, in der am Modulklassenobjekt **410** vorgenommene Änderungen auf Einheitsmodulobjekte **526** propagiert werden sowie auch die Art und Weise, in der der Sicherheitszugriff in den Einheitsmodulobjekten **526** eingerichtet ist. Eine Art und Weise der Bereitstellung dieser Funktionalität besteht in der Einstellung einer oder mehr Flags bzw. Parameter innerhalb des Einheitsmodulklassenobjekts **410** zur Festlegung der Art und Weise, in der die Änderungen zu propagieren sind und die Sicherheit in den Einheitsmodulobjekten **526** zu behandeln ist. Insbesondere kann ein bzw. können mehr Änderungsweitergabeparameter so eingestellt werden, dass festgelegt wird, ob am Einheitsmodulklassenobjekt **410** vorgenommene Änderungen automatisch auf ein oder mehr der Modulklassenobjekte **526** zu propagieren sind oder nicht. Diese Änderungsweitergabeparameter können in den Einheitsmodulobjekten **526** gespeichert werden und sie können für das gesamte Einheitsmodulobjekt oder auf einer Unterelement-nach-Unterelementgrundla-

ge festlegen, ob am Einheitsmodulklassenobjekt vorgenommene Änderungen in dem Einheitsmodulobjekt widergespiegelt werden sollen. Beispielsweise kann das Einheitsmodulklassenobjekt **410** den globalen Änderungsparameter **534** (mit "C" gekennzeichnet) aufweisen, der in jedem vom Einheitsmodulklassenobjekt **410** erstellten Einheitsmodulobjekt eingestellt werden kann, um am Einheitsmodulklassenobjekt **410** vorgenommene Änderungen zu aktivieren bzw. zu deaktivieren, damit sie im Einheitsmodulobjekt automatisch widergespiegelt werden oder nicht. Gleichmaßen kann jedes Unterelement bzw. jeder Block, wie beispielsweise Blöcke **501–505**, **510**, **520** und **522** einen Änderungsparameter **536** aufweisen, der nur für diesen Block festlegt, ob an diesem Block im Einheitsmodulklassenobjekt **410** vorgenommene Änderungen im Einheitsmodulobjekt widerzuspiegeln sind oder nicht. Natürlich können die verschiedenen Blöcke eines Einheitsmodulobjekts unterschiedlich angelegt werden, sodass beispielsweise an Säureblock **501** des Einheitsmodulklassenobjekts **410** vorgenommene Änderungen an den entsprechenden Säureblock eines bestimmten der Modulobjekte **526** propagiert wird, jedoch derart, dass die am Alkaliblock **502** des Einheitsmodulklassenobjekts **410** vorgenommenen Änderungen nicht zu dem Alkaliblock des bestimmten der Einheitsmodulobjekte propagiert werden. Des Weiteren können die von einem Einheitsmodulklassenobjekt erstellten verschiedenen Einheitsmodulobjekte über Änderungsparameter verfügen, die im Vergleich zueinander unterschiedlich eingestellt sind, sodass Änderungen am Alkaliblock **502** innerhalb des Einheitsmodulklassenobjekts **410** auf den entsprechenden Alkaliblock eines ersten der Einheitsmodulobjekte **526**, aber nicht auf den entsprechenden Alkaliblock eines zweiten der Einheitsmodulobjekte **526** propagiert wird. Natürlich kann die Veränderungsverwaltungsmethode des Einheitsmodulklassenobjekts **410** auf die Änderungsparameter der Einheitsmodulobjekte **526** zugreifen und diese einsetzen, um eine Änderung innerhalb dieser Objekte vorzunehmen oder nicht, wenn die Änderung im Einheitsmodulklassenobjekt **410** durchgeführt wurde.

[0106] Auf ähnliche Art und Weise kann das Einheitsmodulklassenobjekt **410** eine oder mehr Sicherheitsparameter aufweisen, die die Art und Weise festlegen, in der die Sicherheit oder der Zugriff in jeder der Einheitsmodulobjekte **526** gesteuert wird. Das Einheitsmodulklassenobjekt **410** kann einen globalen Sicherheitsparameter **538** (mit "S" markiert) aufweisen, der dem gesamten vom Reaktoreinheits-Modulklassenobjekt **410** erstellten Reaktoreinheits-Modulobjekt beliebige gewünschte Sicherheitsstufen liefert und/oder einen anderen Sicherheitsparameter **540** für jedes Unterelement des Einheitsmodulklassenobjekts **410** aufweisen kann, wie beispielsweise für jeden der Blöcke **501–505**, **510**, **520**, **522**, usw., das eine Sicherheitsstufe für jeden dieser Blöcke auf ei-

ner Block-um-Block-Grundlage festlegt. Der globale Sicherheitsparameter **538** kann einen Verriegelungsparameter aufweisen, der das Einheitsmodulklassenobjekt für alle Anwender ausgenommen derjenigen verriegelt, die über eine im Vorfeld genehmigte Sicherheitszugriffstufe verfügen. Natürlich können die Sicherheitsparameter **538** und **540** beliebige Anzahlen verschiedener Sicherheitsstufen festlegen, wie beispielsweise kein Zugriff, beschränkter Zugriff, Zugriff auf bestimmte Typen oder Identitäten von Anwendern, usw. und die Sicherheitsstufen können so eingestellt werden, dass sie in den verschiedenen Blöcken und in den verschiedenen vom selben Einheitsmodulklassenobjekt erstellten Einheitsmodulobjekten unterschiedlich sind. Auf Wunsch kann ein Teil der Sicherheitsmaßnahmen die Bereitstellung einer Verschlüsselung für eine oder mehr Methoden oder Algorithmen aufweisen, die mit dem Einheitsmodulklassenobjekt in Zusammenhang stehen.

[0107] Es versteht sich, dass die Änderungs- und Sicherheitsparameter des Einheitsmodulklassenobjekts **410** auf einen Vorgabewert eingestellt werden können und dass die entsprechenden Änderungs- und Sicherheitsparameter eines jeden vom Einheitsmodulklassenobjekt **410** erstellten Einheitsmodulobjekts **526** diesen Vorgabewert bei der Erstellung übernehmen können. Jedoch können die Vorgabeänderungs- und Sicherheitsparameter in den Einheitsmodulobjekten **526** nach Erstellung dieser Einheitsmodulobjekte auch einzeln geändert werden (durch einen Anwender mit ordnungsgemäßem Sicherheitszugriff). Obwohl die Änderungs- und Sicherheitsparameter hier in Bezug auf ein Reaktoreinheitsmodulklassenobjekt erörtert werden, können ähnliche Änderungs- und Sicherheitsparameter in anderen Einheitsmodulklassenobjekttypen bereitgestellt werden sowie auch in beliebigen gewünschten Gerätetypen Modulklassenobjekten, Steuerungsmodulklassenobjekten, Anzeigemodulklassenobjekten, usw.

[0108] Auf Wunsch kann das Einheitsmodulklassenobjekt **410** Verweise aufweisen, wie beispielsweise URLs oder sonstige Verweise auf Unterlagen, die für oder in Zusammenhang mit dem Einheitsklassenmodulobjekt gespeichert wurden, einschließlich von Unterlagen in Zusammenhang mit der Einheit oder beliebigen Unterelementen der Einheit, die mit dem Einheitsmodulklassenobjekt **410** in Zusammenhang steht. Dieserart Verweis sind in **Fig. 6** als Verweise **549** veranschaulicht.

[0109] Die in **Fig. 6** dargestellte Ausführungsform zeigt Änderungsparameter und Sicherheitsparameter, wie sie mit dem Einheitsmodulklassenobjekt **410** in Zusammenhang stehen und wie sie die Änderungsweitergabe und Sicherheitsrichtlinien anzeigen, wie sie auf untergeordnete oder abgeleitete Objekte des Modulklassenobjekts **410** Anwendung finden. In gewissen Ausführungsformen können Änderungs-

parameter und/oder Sicherheitsparameter zusätzlich oder wahlweise einem entsprechenden untergeordneten Objekt oder einem entsprechenden abgeleiteten Objekt des Modulklassenobjekts **410** (beispielsweise mit Reaktorobjekt **440a**) zugeordnet werden, welche Richtlinien angeben, ob das entsprechende untergeordnete oder abgeleitete Objekt Änderungen empfängt oder die Sicherheitseinschränkungen eines bzw. mehr übergeordneter Objekts/e berücksichtigt oder nicht.

[0110] **Fig. 7** beschreibt eine Bildschirmanzeige, die von der Konfigurationsanwendung **50** von **Fig. 1** während des Vorgangs eines Konfigurationsingenieurs erstellt werden kann, der Modulklassenobjekte zum Konfigurieren der Betriebsanlage **10** angelegt und verwendet. Im Allgemeinen weist die Bildschirmanzeige eine Explorer-Ansicht auf der linken Seite des Bildschirms auf, die eine organisatorische Baumstruktur mit der Konfiguration der Betriebsanlage **10** darstellt. Gleichermäßen weist die Bildschirmanzeige in der Regel eine oder mehr Informationsansichten auf der rechten Seite davon auf. Diese Informationsansichten liefern weitere Informationen über ausgewählte Elemente in der Explorer-Ansicht. Die Informationen, die einem Anwender angezeigt oder vom Anwender in den Informationsansichten geändert werden können, können von den Steuerungs- und Sicherheitsparametern von **Fig. 6**, die für jedes der verschiedenen Modulklassenobjekte oder diesbezüglichen Unterelementen eingestellt wurden, festgelegt oder gesteuert werden. Daher kann ein bestimmtes Element innerhalb der Explorer-Ansicht darstellbar sein oder für einen Anwender zur Betrachtung und/oder Änderung basierend auf den Sicherheits- und Steuerungsparametern herausgestellt werden, die im Modulklassenobjekt eingestellt und auf die in der Explorer-Ansicht dargestellten Modulobjekte propagiert werden. Natürlich können die Informationen wie schon vorher erläutert, jederzeit ausgeblendet werden; von nur einem Anwender dargestellt oder geändert werden, der ein Passwort oder anderen Sicherheitscode eingibt; jederzeit darstellbar, jedoch nicht änderbar sein; jederzeit darstellbar und änderbar sein oder beliebige sonstige Kombinationen dieser oder anderer Sicherheits- und Änderungsparameter aufweisen. Ferner kann die Darstellbarkeit, Sichtbarkeit oder Veränderlichkeit eines Elements auf Wunsch in der Explorer-Ansicht mittels Hervorhebungen, Abgrauen, Farben oder beliebigen anderen Methoden angezeigt werden, um den Anwender darüber zu informieren, welche Elemente ausführlicher angezeigt oder geändert werden können.

[0111] In **Fig. 7** weist die Bildschirmanzeige **600** einen Teil der Explorer-Konfigurationsansicht **602** auf, die auf der linken Seite der Anzeige dargestellt ist. Der Teil der Explorer-Ansicht **602** veranschaulicht eine Bibliothek, in der zahlreiche Modulklassenobjek-

te gespeichert sind, einschließlich der Einheitsmodulklassenobjekte **604**, Gerätemodulklassenobjekte **606** und Steuerungsmodulklassenobjekte **608**. Das Reaktoreinheits-Modulklassenobjekt **610** (das dem Reaktoreinheits-Modulklassenobjekt **410** von **Fig. 6** entsprechen kann) wird in der Einheitsmodulklassenbibliothek **604** gespeichert und weist Anzeigen zahlreicher Unterelemente auf einschließlich von Dosis-, Mischungs-, Abfluss- und Spülphasenklassenobjekten und Säure-, Alkali-, Wasser- und Auslass-Gerätemodulklassenobjekten, Wasser_Ein und Stufe_Messer-Steuerungs-Modulklassenobjekten und anderen Objekten nach Belieben. Dementsprechend weist, wie in der Einheitmodulklassen-Bibliothek **604** definiert, das Reaktoreinheits-Modulklassenobjekt **610** Anzeigen von Phaseklassen sowie auch Anzeigen von Gerätemodulklassenobjekten und Steuerungsmodulklassenobjekten auf. Da das Reaktoreinheits-Modulklassenobjekt **610** in Bildschirm **600** ausgewählt wurde, sind dessen Elemente ausführlicher auf der rechten Seite **612** des Bildschirms **600** dargestellt.

[0112] Des Weiteren weist die Gerätemodulklassenbibliothek **606** ein Totalisatorgeräte-Modulklassenobjekt **614** auf (das dem Totalisatorgeräte-Modulklassenobjekt **416** von **Fig. 7** entsprechen kann) sowie auch das Reaktor_Auslass Gerätemodulklassenobjekt **616**. Das Totalisatorgeräte-Modulklassenobjekt **614** weist drei verschiedene Abschnitte eines Algorithmus auf (wie beispielsweise einer der Algorithmen **564** von **Fig. 7**) namens Befehl_00001, Befehl_00002 und Befehl_00003. Das Modulklassenobjekt **614** weist ebenfalls Verweise auf Steuerungsmodulobjekte namens Grob_Ventil und Fein_Ventil (die die Steuerungsmodulklassenobjekte des Ein/Aus-Typs sind) und Durchsatz_Messer (das ein Steuerungsmodulklassenobjekt des Durchsatzmesser-Typs ist) auf. Darüber hinaus weist das Reaktor_Auslass Gerätemodulklassenobjekt **616** einen zustandsgetriebenen Steuerungsalgorithmus mit verschiedenen Zuständen namens Zustand_00001, Zustand_00002 und Zustand_00003, ein Ziel-, ein Antriebs-, ein Bildschirm- und ein Wiedergabe-Modul und ein Auslass-, ein Abfluss- und ein Produktventil-Steuerungsmodulobjekt (das Anzeigen von oder Verweise auf Modulblöcke des Typs Ein/Aus-Steuerungsmodulklassenobjekt sein kann, und die Auslass, Abfluss und Produkt genannt werden oder die einfache Objekte sein können) auf. Die zu den Totalisator- und Reaktor_Auslass-Modulklassenobjekten **614** und **616** zugehörigen Befehle und zustandsgetriebenen Algorithmen können beliebige gewünschte Routinen sein und sich auf Steuerungsmodulobjekte innerhalb des Gerätemodulklassenobjekts beziehen, das mit diesen Befehlen verwendet wird. Insbesondere können die CDA- oder SDA-Befehlsalgorithmen der Gerätemodulklassenobjekte Ausdrücke oder Aktionen aufweisen, die sich auf die Steuerungsmodule (oder andere Gerätemodule) beziehen, indem die Namen dieser Module aufgenommen werden, um anzuzeigen, wel-

cher Teil der Geräte bei der Ausführung des Algorithmus zu manipulieren ist. Der Gebrauch des Namens des Steuerungsmoduls (oder eines anderen Gerätemoduls) innerhalb dieser Algorithmen gibt das Steuerungsmodulobjekt an, auf das das Gerätemodulobjekt verweist oder das ihm zugeordnet ist, und in dem sich der Algorithmus befindet, und die spezifischen Namen sind vorherbestimmt oder instanziiert, wenn das Gerätemodulobjekt von dem Gerätemodulklassenobjekt erstellt wird.

[0113] Natürlich kann auf Wunsch der in **Fig. 7** veranschaulichte Bildschirm und ähnliche Bildschirme von einem Konfigurationsingenieur zur Erstellung und Vorgabe von Steuerungsalgorithmen innerhalb einer Dosis oder anderen Phasenklassen oder für beliebige andere Module, wie beispielsweise Einheitsmodulklassenobjekte, Gerätemodulklassenobjekte und Anzeigemodulklassenobjekte verwendet werden, um somit beliebige gewünschte Modulklassenobjekt zu erstellen. Nach der Erstellung eines bzw. mehr Modulklassenobjekten, wie weiter oben beschrieben, kann der Konfigurationsingenieur dann diese Modulklassenobjekte zum Konfigurieren der Elemente innerhalb der Betriebsanlage **10** einsetzen.

[0114] Die Geräte oder Steuerungsmodule könnten innerhalb eines Einheitsmodulklassenobjekts als ein gemeinsam genutztes oder ein nicht gemeinsam genutztes Modulobjekt bestimmt werden. Ein nicht gemeinsam genutztes Modulobjekt gehört vollständig zu einem Modulobjekt einer höheren Stufe, in der das nicht gemeinsam genutzte Modulobjekt erstellt wird. Ein gemeinsam genutztes Modulobjekt gehört zu einem Modulobjekt einer höheren Stufe oder ist mehr als einem zugeordnet. Die gemeinsam genutzte oder nicht gemeinsam genutzte Beschaffenheit des Modulobjekts beeinflusst die Darstellung des Modulobjekts in der Explorer-Ansicht. Insbesondere führt die Bezeichnung eines nicht gemeinsam genutzten Modulobjekts in einem nur unter dem Objekt einer höheren Stufe innerhalb der Steuerungsstrategie dargestellten Modulobjekt, während eine gemeinsam genutzte Modulobjekt-Bezeichnung einen gemeinsam genutzten Modulblock oder ein Modulobjekt zur Folge hat, der/das unter jeder der Modulobjekte einer höheren Stufe dargestellt ist, die das Element sowie auch ein selbstständig ausführbares Modulobjekt in der Explorer-Hierarchie gemeinsam nutzen.

[0115] Es versteht sich, dass der Konfigurationsingenieur auf ähnliche Art und Weise, beliebig andere Einheitsmodulklassenobjekte, Gerätemodulklassenobjekte und Steuerungsmodulklassenobjekte sowie auch Anzeigemodulklassenobjekte zur Erstellung von Konfigurationselementen für Einheiten, Geräte, Steuerungselemente und Anzeigeelemente innerhalb der Prozess-Steuerungsumgebung in Einklang mit den hier beschriebenen Prinzipien einsetzen kann. Ferner kann der Konfigurationsingenieur

Änderungen an den Elementen der Konfiguration verschiedener Prozesseinheiten auf globaler Grundlage durch eine Änderung eines oder mehr Einheitsmodulklassenobjekts/e vornehmen und diese Änderungen auf jede der Modulobjekten propagieren, die von den Einheitsmodulklassenobjekten erstellt wurden und diesen zugeordnet sind. Diese Funktion erleichtert das Vornehmen von Änderungen innerhalb der Konfiguration und macht sie weniger zeitaufwendig, nachdem eine Konfiguration bereits erstellt wurde. Außerdem kann der Konfigurationsingenieur eine Zugriffsstufe für verschiedene Elemente oder Komponenten von Modulobjekten innerhalb des Konfigurationssystems bestimmen, indem er Sicherheitsparameter innerhalb der Modulklassenobjekte festsetzt. Wie weiter oben aufgeführt, kann der Konfigurationsingenieur eine Sicherheit auf einer Modulnach-Modul Grundlage auf jeder beliebigen Stufe definieren, wie beispielsweise auf der Einheitsmodulstufe, Gerätemodulstufe, Steuerungsmodulstufe und Anzeigemodulstufe. Auf diese Art und Weise können bestimmte Elemente des Einheitsmodulobjekts anschaulich sein, während andere nicht betrachtet werden können.

[0116] Natürlich gilt, dass nachdem das Konfigurationssystem abgeschlossen ist und die Modulobjekte an die einzelnen Prozesseinheiten innerhalb der Betriebsanlage **10** gebunden wurden, die diesen Modulen zugeordneten Steuerungs- und Anzeigemodule oder Elemente auf die entsprechenden Controller **12**, Geräte und Arbeitsplatzsysteme **14** von **Fig. 1** zur Ausführung während des Betriebs der Betriebsanlage **10** heruntergeladen werden können.

[0117] Die Methoden, Systeme und hier beschriebenen Verfahren gestatten die Konfiguration von Prozessbetrieben und Prozess-Steuerungssystemen unter Einsatz flexibler Klassenobjekte, Instanzobjekte und Prozesselementobjekte. In einem Beispiel ist die Weitergabe von Änderungen oder Modifikationen an ein übergeordnetes Objekt phasenmäßig gestaltet oder auf eines oder mehr der jeweiligen untergeordneten Objekte verzögert, damit die zeitliche Steuerung der Konfigurationsaktualisierungen in der Betriebsanlage gesteuert werden kann. Die Phasenregelung und/oder Verzögerung wird von dem Anwender in einer Ausführungsform angegeben und kann unterschiedlich für verschiedene untergeordnete Objekte ausfallen. In einem anderen Beispiel werden die an einem übergeordneten Objekt vorgenommenen Änderungen auf ausgewählte untergeordnete Objekte jedoch nicht auf alle der untergeordneten Objekte der übergeordneten Objekte angewendet. Der Anwender kann die Auswahl der gewünschten untergeordneten Objekte angeben, für die die Änderungen gelten sollen. In noch einem weiteren Beispiel werden die an einem untergeordneten Objekt vorgenommenen Änderungen selektiv auf dessen übergeordnetes Objekt und/oder auf ein oder mehr untergeordnete

te Objekte angewendet. Die Auswahl der gewünschten übergeordneten Objekte und/oder des/der untergeordneten Objekts(e) auf die die Änderungen anzuwenden sind, kann vom Anwender angegeben werden. Zusätzlich oder wahlweise kann ein Anwender Änderungsentwürfe oder Modifikationen an verschiedenen Objekten ohne eine automatische Verteilung und/oder Instantiierung (beispielsweise Erstellung einer ablauffähigen Datei) der Änderungen vorzunehmen. Verschiedene Sätze an Änderungsentwürfen können als verschiedene Versionen der verschiedenen Objekte gespeichert und rechnerunabhängig geprüft werden, ohne den live Betrieb der Laufzeit-Betriebsanlage zu beeinträchtigen.

[0118] Des Weiteren gelten die hier beschriebenen Methoden, Systeme und Verfahren für in der Bibliothek gespeicherte Elemente oder Objekte (beispielsweise Vorlagen und/oder vom Anwender erstellte Bibliothekobjekte) des Prozess-Steuerungssystems oder Betriebs **10**. Wahlweise oder zusätzlich gelten die Methoden, Systeme und Verfahren für Elemente oder Objekte, die in der Konfigurationsdatenbank **25** des Prozess-Steuerungssystems oder Betriebs **10** gespeichert sind, die in gewissen Fällen mindestens teilweise von Bibliothekelementen oder -objekten erstellt oder abgeleitet wurden.

[0119] Des Weiteren kann eine oder mehr der hier beschriebenen flexiblen Konfigurationsmethoden und -objekte in einer Betriebsanlage, wie beispielsweise der Betriebsanlage **10** von **Fig. 1** oder in anderen entsprechenden Prozessbetrieben oder Prozess-Steuerungssystemen verwendet werden. In einem Beispiel werden eine oder mehr der hier beschriebenen Methoden der Konfigurationsanwendung **50** auf einem oder mehr Arbeitsplatzsystemen **14** von **Fig. 1** ausgeführt. In einem anderen Beispiel werden eine oder mehr der hier beschriebenen Methoden mindestens teilweise durch eine Fernanwendung (beispielsweise einen Web-Client oder anderes Fernzugriffsmittel) ausgeführt, die auf die Anwendung zugreift. In bestimmten Ausführungsformen werden eine oder mehr der flexiblen Konfigurationsmethoden zusammen mit anderen Konfigurationsmethoden ausgenommen der hier beschriebenen verwendet.

[0120] Überdies werden die hier beschriebenen flexiblen Konfigurationsmethoden und -objekte mit Hilfe der Modulklassen wie beispielsweise den in **Fig. 5** beschriebenen ausgeführt. Jedoch können die hier beschriebenen flexiblen Konfigurationsmethoden auch mit Hilfe anderer entsprechender Modulklassen, Softwarearchitekturen und/oder Programmierungsmethoden ausgeführt werden.

Element- oder Objektentwürfe:

[0121] Wie weiter oben erörtert, ermöglichen die Ausführungsformen der hier beschriebenen flexiblen Konfigurationsmethoden, Systeme und Methoden dem Anwender Änderungen an den Elementen oder Objekten der Bibliothek und an Systemkonfigurationen vorzunehmen, ohne dass dabei Downloads erforderlich wären, die die Laufzeit der Betriebsanlage oder des Prozess-Steuerungssystems nachteilig beeinflussen könnten. Die Elemente oder Objekte der Bibliothek sind in der Regel Vorlageobjekte, die an einem zentral zugänglichen Standort oder in der Bibliothek (beispielsweise in der Konfigurationsdatenbank **25** oder in einem anderen Speicherungsgerät, die den Arbeitsplatzsystemen **14** und anderen Schnittstellen und Rechnergeräte zugänglich ist) gespeichert werden. Systemkonfigurationselemente oder -objekte (und bestimmte Bibliothekselemente oder -objekte) basieren generell auf oder werden von einem oder mehr Bibliothekselementen abgeleitet. Für gewisse Element und Objekte werden mindestens einige Aspekte vom Anwender individuell angepasst.

[0122] Der hier verwendete Begriff "Element" bezieht sich im Allgemeinen auf Objekte, wie beispielsweise ein Klassenobjekt, ein Fallobjekt oder ein Prozesselementobjekt. Ein Element kann in einer Bibliothek (beispielsweise ein "Bibliothekselement" oder "Bibliothekobjekt") gespeichert werden oder auch in einem Systemkonfigurations-Datenspeicherungsbereich (beispielsweise ein "Konfigurationselement" oder "Konfigurationsobjekt"). Des Weiteren kann sich der hier verwendete Begriff "Element" ebenfalls generell auf ein Element beziehen, dass intern in einem Objekt befindlich ist und von diesem definiert wird, beispielsweise mindestens ein Teil des Inhalts eines Objekts. Mögliche interne Elemente von Objekten weisen beispielsweise Methoden, Aktionen, Datenfelder oder andere Attribute, eine Eingabe, eine Ausgabe, eine Art Ein/Aus (beispielsweise einen Typ einer Eingabe/Ausgabe-Karte oder Gerät über den Controller, beispielsweise Controller **12** oder **12a** von **Fig. 1**, der mit dem System kommuniziert), Funktionen oder Nutzungen, Definitionen, Parameterwerte, einen Verweis auf ein übergeordnetes Objekt, einen Verweis auf ein untergeordnetes Objekt, einen Verweis auf ein anderes Objekt, das weder ein übergeordnetes Objekt noch ein untergeordnetes Objekt ist, und andere interne Elemente auf. Ein internes Element kann absolut definiert werden (beispielsweise ein Datenfeld, das einen Konstantwert oder Ausdruck speichert) oder ein internes Element kann relativ definiert werden, wie beispielsweise durch Verweis auf einen absoluten Wert, einen absoluten Ausdruck, ein anderes Objekt oder einen anderen Verweis. Beispielsweise können interne Elemente von einem Grafikanzeige-Elementobjekt definiert werden, das einen Verweis auf ein übergeordnetes Objekt, eine oder mehr Felder, Auslöser, Funktionen, Ansichts-

Definitionen, Ereignishandler, Animationen, Platzhalter, Parameter, Tabellen, usw. aufweist. In einem anderen Beispiel werden interne Elemente von einem Steuerungsmodul-Elementobjekt definiert, das einen Verweis auf ein übergeordnetes Objekt, eine oder mehr Eingaben, Ausgaben, Parameter, Funktionsblöcke, Verbindungen, Feldausdrücke, externe Verweise, Aktionen, Algorithmen, Transitionen, usw. aufweist.

[0123] Der generell hier verwendete Begriff "verlinktes" Element bezieht sich auf ein Objekt oder Element, dessen Struktur und Ausgangswerte von einem übergeordneten Objekt oder Element abgeleitet oder erstellt sind und/oder dessen Struktur und Ausgangswerte einem untergeordneten Objekt oder Element bereitgestellt werden. In diesem Sinne kann ein verlinktes Element ein übergeordnetes Element sein. Zusätzlich oder wahlweise kann ein verlinktes Element ein untergeordnetes Element sein. Dementsprechend bezieht sich ein "unverlinktes" Element, wie hier generell benutzt, auf ein Element oder ein Objekt, das kein übergeordnetes Objekt und keine untergeordneten Objekte hat. Die Verlinkung (beispielsweise die Pflege von Anzeigen von übergeordneten Objekten/untergeordnete Objektbeziehungen) gestattet dem Anwender die Struktur und Ausgangswerte eines übergeordneten Elements oder Objekts zu definieren und dann dieselbe/n Struktur und Ausgangswerte mit Instanz oder untergeordneten Objekten gemeinsam zu nutzen. Wenn beispielsweise ein Anwender eine Modifikation oder Änderung vornehmen möchte, die sich auf alle Instanzen des Klassenobjekts auswirken, modifiziert der Anwender lediglich das Klassenobjekt und die Änderungen werden dann auf die verlinkten Instanzen des Klassenobjekts verteilt oder propagiert.

[0124] Der hier verwendete Begriff "aktuelles Element" oder "aktuelles Objekt" bezeichnet ein Element oder Objekt, das instanziiert (und in bestimmten Fällen heruntergeladen) wurde, und das während der Laufzeit des entsprechenden Prozesselements in der Betriebsanlage **10** ausgeführt werden kann. Wenn beispielsweise ein aktuelles Prozesselement-Steuerungsobjekt instanziiert wird, wird der ablauffähige Konfigurationcode, der dem aktuellen Prozesselement-Steuerungsobjekt entspricht, auf ein Prozesselement heruntergeladen, wie beispielsweise auf einen Controller **12** oder **12a** von **Fig. 1**, wobei der Controller **12** oder **12a** so konfiguriert ist, dass er während der Laufzeit in Einklang mit den Funktionen, Eingaben, Ausgaben und anderen Bedingungen, wie von dem instanziierten aktuellen Prozesselement-Steuerungsobjekt definiert, läuft. In einem anderen Beispiel wird ein aktuelles Grafikanzeigeansichtobjekt auf einer Anwenderschnittstelle instanziiert, wenn eine Ansicht aufgebaut wird, die der Definition entspricht, die in dem aktuellen Grafikanzeigeansichtobjekt eingeschlossen ist, und auf der Anwenderschnittstelle prä-

sentiert wird. In der Regel wirkt sich eine Änderung, die an einem bestimmten internen Element vorgenommen wurde, das von einem Objekt definiert wurde, nicht auf andere interne Element auf, die von dem Objekt definiert wurden.

[0125] Die Begriffe "Modifikation" und "optimieren" werden hier austauschbar verwendet, um eine oder mehr Änderungen des Inhalts eines Objekts während der Beibehaltung eines Links zu seinem übergeordneten Objekt zu bezeichnen. Ein Beispiel einer Modifikation oder Änderung an einem aktuellen Element oder Objekt ist die Addition des neuen internen Elements zum aktuellen Element oder Objekt, wie beispielsweise die Addition des neuen Parameters oder der neuen Aktion. Zusätzlich oder wahlweise kann eine Modifikation das Löschen eines vorhandenen Elements des aktuellen Prozesselementobjekts und/oder die Änderung eines Werts, Ausdrucks oder den Verweis eines vom Objekt definierten Elements aufweisen. In bestimmten Szenarien weist eine Modifikation die Deaktivierung des bestimmten Elements auf, sodass das bestimmte Element während der Instantiierung ausgeklammert wird. Beispielsweise kann ein Anwender in einem bestimmten Instanzobjekt ein Element deaktivieren, das im übergeordneten Klassenobjekt definiert ist. Diese Fähigkeit gestattet dem Anwender, die Instanzen zu definieren, die über eine geringere Funktionalität im Vergleich zum Rest der Klassen verfügen, jedoch dennoch eine Kernklassenfunktionalität aufweisen. Der deaktivierte Inhalt kann visuell stumm geschaltet werden, unsichtbar gemacht werden, oder auf sonstige Weise leistungsunfähig gemacht werden, und wird nicht während der Laufzeit verwendet. Der deaktivierte Inhalt kann vom Anwender jederzeit re-aktiviert werden. Im Allgemeinen kann eine Modifikation die Lösung eines Verweises auf einen Konstantwert oder Ausdruck aufweisen, wie beispielsweise, wenn ein Verweis in einem übergeordneten Objekt auf einen Konstantwert in einem jeweiligen untergeordneten Objekt gelöst wird.

[0126] Alle Modifikationen, Änderungen und diesbezügliche Kombinationen können an Klassen-Elementen oder -objekten, Instanzelementen oder -objekten, Prozesselement-Elementen oder -objekten oder internen Elementen vorgenommen werden. Im Fall von Instanzobjekten und anderen untergeordneten Objekten haben die Änderungen oder Modifikationen an untergeordneten Objekten nicht zur Folge, dass das untergeordnete Objekt von seinen übergeordneten Objekten in einer Ausführungsform entfernt wird. Beispielsweise entfernen die Modifikationen an einem Instanzobjekt nicht das Instanzobjekt von seiner Klasse und wirken sich auch nicht auf andere Instanzen derselben Klassen aus, solange dies nicht von dem Anwender angezeigt wird. In einer Ausführungsform werden die von einem untergeordneten Objekt definierten Änderungen oder Modifikationen von Ele-

mente mittels einer Flag oder anderen Zeichen im untergeordneten Objekt angezeigt, und können als solches bezeichnet werden, damit nicht mit Elementen auf der Klassenstufe interferiert wird. In diesem Sinne könnte diese Flexibilität die Proliferation von übergeordneten Objekten verringern und ein einziges übergeordnetes Element oder Objekt für eine breitgefächerte Vielfalt von Anwendungen verwendbar machen, indem mehr Variation unter seinen untergeordneten Elementen oder -objekten gewährt wird.

[0127] Des Weiteren kann der Anwender einen Satz an Modifikationen auf einem Element oder Objekt als einen Entwurf speichern. Im Allgemeinen können Entwurfobjekte ohne Auswirkungen auf die Laufzeit der Betriebsanlage oder des Systems getestet werden. (Das "Testen" des Entwurfobjekts, wie hier durchgeführt, bezieht sich in der Regel auf die Fähigkeit, eine Anfangsprüfung auf dem Entwurfobjekt selbst vorzunehmen, um bestimmen zu können, ob die Änderungen oder Modifikationen stimmen, jedoch typischerweise impliziert dies keine komplette Werksannahmeprüfung, bei der das Entwurfobjekt in einer Umgebung mit anderen Entwürfen zu funktionieren hat.)

[0128] In einer Ausführungsform kann nur ein Entwurf für ein bestimmtes Objekt auf einmal vorhanden sein. In einer Ausführungsform können vielfache verschiedene Entwürfe für ein bestimmtes Objekt vorhanden sein. Der Anwender kann ein Entwurfobjekt hinzufügen, löschen oder ändern. In bestimmten Ausführungsformen steht die Versionsteuerung von Entwurfobjekten zur Verfügung. Ein Entwurfobjekt kann als ein Entwurf mit einer Flag, einem Feld oder anderen Anzeigen gekennzeichnet werden, die im Objekt aufgenommen oder ihm zugeordnet wurden. In einer Ausführungsform wird ein Entwurfobjekt als ein untergeordnetes Objekt des zur Zeit instanziierten Objekts, als ein untergeordnetes Objekt des übergeordneten Objekts des aktuellen Objekts oder als ein untergeordnetes Objekt des Bibliothekobjekts, von dem das aktuelle Objekt abgeleitet wurde, gespeichert. In gewissen Szenarien können andere Informationen, die dem Entwurfobjekt (beispielsweise Autor, Zeitpunkt der Speicherung, usw.) entsprechend, ebenfalls gespeichert werden.

[0129] Verschiedene Entwürfe, die verschiedene Modifikationen an einem aktuellen Objekt aufweisen, können erstellt und gespeichert werden (beispielsweise gleichzeitig). Das verschiedenen Entwürfe können aufeinanderfolgend sein. Beispielsweise kann ein erster Entwurf dem Konfigurieren eines Altprozessgeräts und ein zweiter Entwurf dem Konfigurieren eines neueren Modells des Prozessgeräts entsprechen. Verschiedene Entwürfe können einen unterschiedlichen Inhalt aufweisen. So könnte beispielsweise ein erster Entwurf der Konfiguration eines Ventil entsprechen, das von Hersteller A fabri-

ziert wurde, um eine bestimmte Funktion zu erfüllen, während ein zweiter Entwurf der Konfiguration eines Ventils entspricht, dass von Hersteller B zur Erfüllung einer ähnlichen Funktion hergestellt wurde. Beliebige Kombinationen aufeinanderfolgender und/oder Änderungen des Inhalts können durch einen Satz von Entwürfen aufgewiesen werden, wie beispielsweise vom Anwender gewünscht.

[0130] In der Regel ist nicht gestattet, dass Entwürfe instanziiert oder heruntergeladen werden. Sie dürfen jedoch getestet werden, um sicherzustellen, dass sie ordnungsgemäß funktionieren, bevor sie in Editionen veröffentlicht werden, die verteilt und heruntergeladen werden dürfen. Auf diese Art und Weise kann ein Anwender Modifikationen an einem aktuellen Element oder Objekt vornehmen und die Modifikationen als einen Entwurf speichern. Sollte der Anwender den Entwurf als annehmbar erachten, kann er den Entwurf veröffentlichen, um eine neue Edition des Element oder Objekts zu erzeugen, wie beispielsweise einen "veröffentlichtes" oder "genehmigtes" Element oder Objekt. Die neue Edition des Elements oder Objekts wird gespeichert und steht zur Instanziierung oder zum Herunterladen in die Betriebsanlage oder System zur Verfügung.

Editionen:

[0131] In diesem Sinne helfen die Editionen dem Anwender die Änderungen der Prozesselemente zu verfolgen und zu steuern. Insbesondere kann ein Element oder Objekt Änderungen im Rahmen seiner Lebenszeit durchgehen. Zunächst wird das Objekt erstellt und jede folgende Änderung des Objekts (entweder der Struktur, wie beispielsweise das Hinzufügen/Löschen eines Funktionsblocks oder die Änderung eines Parameterwerts) wird in der Regel verteilt oder auf seine untergeordneten Objekte propagiert. Während des live Betriebs der Betriebsanlage sollte es jedoch verhindert werden, dass Änderungen an übergeordneten Elementen oder -objekten (und insbesondere unvollständige, ungenehmigte oder nicht geprüfte Änderungen) in das Laufzeit-System ungehindert eingespeist werden. Dementsprechend weisen die flexiblen hier beschriebenen Konfigurationsmethoden und Objekte "Editionen" auf, die dem Anwender die Steuerung der Verteilung und Weitergabe der Änderungen ermöglichen. Die hier verwendete "Edition" bezieht sich in der Regel auf eine Entwurfsmodifikation für ein aktuelles Objekt, das genehmigt oder veröffentlicht wurde und zur Instanziierung in eine Laufzeit-Umgebung bereitsteht. Editionen oder Veröffentlichungen können in eine Betriebsanlage oder in Prozess-Steuerungssysteme instanziiert werden, wogegen Entwürfe nicht instanziiert werden dürfen.

[0132] In einer Ausführungsform werden die Entwürfe in Editionen veröffentlicht, wenn der Anwen-

der dies angibt (beispielsweise nur nach einem vom Anwender genehmigten Entwurf). Dementsprechend gilt, wenn eine Entwurfsmodifikation für ein Prozesselement-Element eines Objekts genehmigt ist, kann sich der Zustand des Prozesselement-Elements oder Objekts vom ersten Zustand, in dem die Instanziierung verhindert oder nicht gestattet wurde (beispielsweise ein Zustands-"Entwurf"), in einen zweiten Zustand verändern, in dem eine Instanziierung gestattet ist (beispielsweise ein "veröffentlichter" Zustand). In einer Ausführungsform wird nur befugten Anwendern oder Anwendern, die über eine entsprechende Unbedenklichkeitsbescheinigung verfügen, gestattet, Entwürfe in Editionen zu veröffentlichen (beispielsweise zur Autorisierung von Entwurfsmodifikationen zur Erstellung von veröffentlichten Modifikationen oder Editionen). Sollten mehrere Entwürfe zur Verfügung stehen, kann der Anwender auswählen, welcher der vielfachen Entwürfe in einer Edition veröffentlicht wird.

[0133] Die Editionen von Objekten können mittels einer Flag, eines Felds oder anderer Anzeigen erkenntlich gemacht werden, die das Objekt aufweist oder die ihm zugeordnet wurden. In einer Ausführungsform wird die Edition oder das veröffentlichte modifizierte Objekt als ein separates Objekt gespeichert, das als sein übergeordnetes Objekt einen Entwurf des modifizierten Objekts, des aktuellen Objekts, einer anderen Edition, eines anderen Objekts oder eines Bibliothekobjekts anzeigt oder damit verlinkt ist. In gewissen Szenarien können andere dem Entwurf des Objekts entsprechende Informationen (beispielsweise Autor, Zeitpunkt der Veröffentlichung, usw.) ebenfalls gespeichert werden. In einer Ausführungsform kann ein untergeordnetes Element oder Objekt mit der Edition des übergeordneten Objekts und mit dem übergeordneten Objekt selbst verlinkt werden.

[0134] Wenn mehrere Editionen eines Objekts zur Verfügung stehen, kann der Anwender bestimmen oder auswählen, welche der mehrfachen Editionen in einer Ausführungsform instanziiert werden soll. So kann ein Anwender beispielsweise angeben, dass verschiedene Editionen des Objekts zu verschiedenen Prozesselementen innerhalb der Betriebsanlage instanziiert werden sollen. Wahlweise oder zusätzlich kann ein Anwender verschiedene Instanzierungszeiten für eine Edition oder für verschiedene Editionen angeben. Ferner kann der Anwender die Edition in gewissen Fällen löschen, deaktivieren oder bearbeiten.

[0135] In bestimmten Ausführungsformen darf nur ein Entwurf (und nicht mehrere Entwürfe) gespeichert werden. In diesen Ausführungsformen darf nur die aktuelle Edition des aktuellen Objekts vom Anwender bearbeitet werden. In diesem Sinne speichert der Anwender nach der Bearbeitung das Objekt, dessen Entwurf modifiziert wurde, ab, und auf Anweisung des Anwenders wird dieses Objekt dann als eine neue ak-

tuelle Edition des Objekts veröffentlicht. In einer Ausführungsform kann eine zuvor aktuelle Edition automatisch gelöscht werden, falls keine untergeordneten Objekte mit ihr verlinkt sind, und falls sie nicht die einzige Edition darstellt, wie beispielsweise im Fall von Bibliothekselementen oder -objekten. Sollte also eine aktuelle Edition keine untergeordneten Objekte haben, kann eine nachfolgende Veröffentlichung der Modifikationen aktueller Editionen die aktuelle Edition einfach ersetzen oder überschreiben (beispielsweise wird keine zusätzliche aktuelle Edition erstellt). In einer Ausführungsform können eine oder mehr vorherige aktuelle Editionen ausgeblendet oder nicht für Anwender zugänglich gemacht werden.

[0136] In einer Ausführungsform bewirkt die Veröffentlichung eines übergeordneten Elements oder -objekts, das mindestens ein untergeordnetes Objekt hat, automatisch die neue aktuelle Edition des untergeordneten Objekts. In einer Ausführungsform bewirkt die Neubenennung eines Objekt nicht die Erstellung einer neuen Edition des Objekts; jedoch erfordert die Neubenennung eines vom Objekt definierten internen Elements die Abspeicherung des umbenannten internen Elements als ein Entwurf sowie auch die Veröffentlichung des umbenannten internen Elemententwurfs als eine neue Edition des internen Elements.

[0137] In einer Ausführungsform kann die Erstellung einer neuen Edition des innersten Elements oder Objekts automatisch neue Editionen eines jeden beinhalteten Elements oder Objekts auf allen Verschachtelungsstufen erstellen, wenn ein veröffentlichtes Element oder Objekt auf ein anderes veröffentlichtes Element oder Objekt verwiesen oder dort aufgenommen wird (und wahlweise in diesem Sinne in mehreren Stufen der Objekte verschachtelt wird).

Verteilung oder Weitergabe von
Änderungen und Modifikationen:

[0138] Die Begriffe "Verteilung" und "Weitergabe" werden hier austauschbar verwendet und beziehen sich in der Regel auf die Übertragung von Struktur (wie beispielsweise Funktionsblocknutzungen, Dynamonutzungen, Formen, Parametertypen, Drähte, usw. für Steuerungsobjekte; und auch beispielsweise Animationen, Ansichtsdefinitionen, Formen, Anschlüsse, usw. für Grafikelementobjekte) und Parameterwerten des übergeordneten Objekts auf ein untergeordnetes Objekt und/oder von einem untergeordneten Objekt auf sein übergeordnetes Objekt. In der Regel werden alle Modifikationen oder Änderungen in einer Edition oder in einem veröffentlichtem Objekt propagiert. In einer Ausführungsform werden nur die geänderten in der veröffentlichten Edition des Objekts enthaltenen Elemente propagiert. In einer Ausführungsform wird die gesamte Edition des Objekts propagiert.

[0139] Eine Richtlinieneinstellung kann den Verteilungs- oder Weitergabeprozess auf einer pro Objekt- oder pro Element-Grundlage steuern. In einer Ausführungsform kann die Richtlinieneinstellung vom Anwender, wie beispielsweise dem Konfigurationsdesigner oder dem Systemadministrator, geändert werden. Der Zugriff zur Änderung der Richtlinieneinstellungen ist in bestimmten Konfigurationssystemen sicher gesteuert. Die Verteilung oder Weitergabe der Richtlinieneinstellung weist mindestens zwei mögliche Werte auf: automatisch und vom Anwender verwaltet. In einem Beispiel werden Änderungen oder Modifikationen an in der Bibliothek gespeicherten Geräteelementen, die in einer neuen Edition veröffentlicht wurden, automatisch und umgehend auf beliebig verlinkte Elemente dieses Geräteelements propagiert, was zu neu veröffentlichten Editionen der verlinkten Elements führt, wenn der Richtliniensatz für ein in der Bibliothek definiertes Geräteelement auf automatisch eingestellt wird.

[0140] Die neu veröffentlichten Änderungen werden nicht automatisch und umgehend auf verlinkte Elemente propagiert, falls die Richtlinien des Objekts auf anwenderverwaltet eingestellt wurden. Stattdessen kann der Anwender die Weitergabe (oder ein Skript oder ähnliches zur Einleitung der Weitergabe) auf ausgewählte verlinkte Elemente zu bestimmten Zeiten manuell einleiten. In dem Beispiel des in der Bibliothek gespeicherten Geräteelements weiter oben kann der Anwender bestimmte untergeordnete Systemkonfigurationselemente auswählen, damit diese die veröffentlichten Änderungen zu bestimmten Zeitpunkten erhalten, wenn ihre Richtlinien auf anwenderverwaltet eingestellt sind. Des Weiteren kann der Anwender bestimmen, welche der mehrfachen Editionen zu welchen Zeitpunkten zu propagieren sind, sollte es mehrere gespeicherte Editionen von in der Bibliothek gespeicherten Geräteelementen geben.

[0141] Dementsprechend gestatten die flexiblen hier beschriebenen Konfigurationsmethoden und Objekte dem Anwender Änderungen an einem übergeordneten Objekt (beispielsweise einem Klassenobjekt) vorzunehmen, untergeordnete Objekte (beispielsweise Instanzobjekte) anzugeben, an die Änderungen zu verteilen oder zu propagieren sind sowie auch anzugeben, wann die Weitergabe stattfinden soll. Daher können nach der Veröffentlichung die an den übergeordneten Objekten vorgenommenen Änderungen umgehend nur auf die die Änderung benötigenden untergeordneten Objekte propagiert werden, während die anderen untergeordneten Objekte mit der Änderung zu einem späteren Zeitpunkt aktualisiert werden können. Die Auswahl der Empfänger der Änderung oder Modifikation kann auf der Grundlage einer pro Prozesseinheit oder eines pro Elements vorgenommen werden, oder auch auf der Grundlage einer Gruppe (beispielsweise ein Bereich des Betriebs, alle Geräte eines bestimmten Herstellers, alle Gerä-

te, die Version X betreiben, alle Messer in einer bestimmten Testfertigungsanlage, usw.). Dieserart Selektivität hinsichtlich der Verteilung und Weitergabe kann die Betriebsleistung während der Laufzeit erhöhen, da nicht alle untergeordneten Objekte auf einmal aktualisiert werden. Ferner können die Änderungen und Modifikationen in einer Betriebsanlage auf optimierte und gesteuert Art und Weise allmählich eingeführt werden.

[0142] In einer Ausführungsform für ein übergeordnetes Objekt, das mindestens ein untergeordnetes Element sowie auch mehrfache vorhandene Editionen hat, wirkt sich eine Änderung der Verteilungs- oder Weitergaberrichtlinieneinstellung nicht auf veröffentlichte, zur Zeit instanziierte Editionen des übergeordneten Objekts aus. Sollte jedoch der Inhalt des übergeordneten Elements oder -objekts im Folgenden geändert werden, werden die Inhaltsänderungen auf alle seiner untergeordneten Elemente oder -objekte propagiert, sodass die untergeordneten Objekte die hierdurch entstandene aktuelle Edition des übergeordneten Objekts verwenden, und vorherige Editionen gelöscht bzw. deaktiviert werden.

[0143] In einer Ausführungsform findet die Weitergabe oder Verteilung von Parameterwerten von einem übergeordneten Objekt nur dann statt, wenn der Wert nicht im untergeordneten Objekt überschrieben wurde.

[0144] Ähnlich wie bei den übergeordneten Objekten können an einem untergeordneten Objekt (beispielsweise einem Instanzobjekt oder einem Prozesselementobjekt) vorgenommene Änderungen oder Modifikationen ebenfalls auf ein jeweiliges übergeordnetes Objekt (beispielsweise ein Klassenobjekt oder ein Instanzobjekt) selektiv propagiert oder verteilt werden. In einer Ausführungsform bestimmt der Anwender die zeitliche Steuerung und/oder den Inhalt der Weitergabe. Auf diese Art und Weise können an einem untergeordneten Objekt vorgenommene Änderungen auf ein übergeordnetes Objekt propagiert werden, das wiederum die Änderung auf andere untergeordnete Objekte propagieren kann.

[0145] In einer Ausführungsform kann der Anwender bestimmen, dass ein bestimmtes von einem untergeordneten Objekt definiertes Element dem übergeordneten Objekt gegenüber auszublenden ist (es wird beispielsweise deaktiviert). Wenn beispielsweise ein bestimmter Parameterwert einem untergeordneten Objekt hinzugefügt wird, jedoch nicht auf das übergeordnete Objekt und andere untergeordnete Objekte des übergeordneten Objekts generell anwendbar ist, kann der bestimmte Parameterwert beflagged oder auf sonstige Weise während Weitergabe auf das übergeordnete Objekt ausgeblendet oder deaktiviert werden.

[0146] In mit einer Ausführungsform kann eine Edition zur Instantiierung durch einen Anwenders ausgewählt werden. Die Instantiierung der ausgewählten Edition führt zu einer entsprechenden Ausführung der Prozesselemente während der Laufzeit in Einklang mit den internen von der Edition definierten Elementen. Beispielsweise wünscht ein Anwender im Fall eines Steuerungsobjekts, dass eine bestimmte veröffentlichte Edition des Steuerungsobjekts auf ein entsprechendes Gerät im Laufzeit-System heruntergeladen wird. Der Anwender weist das System an, eine Download-Edition von der ausgewählten Edition zu erstellen und diese auf das Prozesselement im Laufzeit-System zu übertragen. In diesem Sinne führt das Prozesselement eine ausführende Edition des Steuerungsobjekts während der Laufzeit aus, wobei die ausführende Edition eine Konfiguration aufweist, die die in der Edition enthaltenen Modifikationen enthält. In einem anderen Beispiel weist der Anwender das Konfigurationssystem an, ein veröffentlichtes in einer Anzeigeansicht enthaltenes Grafikanzeigeelementobjekt zu instantiieren. Das Konfigurationssystem erstellt eine ausführende Edition des Grafikanzeigeelementobjekts. Wenn die entsprechende Anzeigeansicht während der Laufzeit erstellt wird, wird die ausführende Edition des Grafikanzeigeobjekts ausgeführt, was dazu führt, dass das jeweilige Grafikelement in die Anzeigeansicht aufgenommen wird. Wenn mehrfache Editionen zur Verfügung stehen, kann der Anwender bestimmen, welche der mehrfachen Editionen instanziiert werden sollen.

[0147] In den weiter oben aufgeführten Instantiierungsveispielen findet die Instantiierung zu einem vom Anwender bestimmten Zeitpunkt oder nach einem vom Anwender bestimmten oder angegebenen Ereignis statt. In bestimmten Ausführungsformen kann der Anwender bestimmen, dass die Edition eines Objekts umgehend nach der Veröffentlichung instanziiert wird. Beispielsweise kann der Anwender bestimmen, dass die Instantiierung eines ersten Satzes an der untergeordneten Objekten entsprechenden Edition umgehend nach der Veröffentlichung durchgeführt wird, dass die Instantiierung eines zweiten Satzes von der untergeordneten Objekten entsprechenden Edition eine bestimmte Zeit lang verzögert wird, und dass die Instantiierung eines dritten Satzes von der untergeordneten Objekten entsprechenden Edition solange verzögert wird, bis die Instantiierung ausdrücklich vom Anwender anfordert wird. In diesem Sinne kann der Anwender mit Hilfe der flexiblen hier beschriebenen Konfigurationssystem-Methoden und -Objekte steuern, wenn und welche Modifikationen für verschiedene Prozess-Steuerungselemente instanziiert werden.

[0148] Die hier beschriebenen Methoden und Objekte bieten dem Anwender die Funktionalität und Fä-

higkeit, auf flexiblere Weise den Inhalt von Objekten oder Elementen zu ändern und die Verteilung und zeitliche Steuerung der Modifikationen innerhalb der Betriebsanlage oder des Prozess-Steuerungssystems flexibler zu verwalten. So kann der Anwender beispielsweise Instanzen und Nutzungen von Objekten erstellen, Kopien von Objekten erstellen, abgeleitete Elemente von Objekten erstellen, interne Elemente (beispielsweise Nutzungen und Instanzen) so modifizieren, dass zusätzliche Strukturen hinzugefügt werden, in Objekten und Nutzungen und Instanzen (beispielsweise Parameter und Eigenschaftswerte) enthaltene interne Elemente modifizieren, ausgewählte interne Elemente ausblenden oder deaktivieren, Objekte verlinken und trennen und/oder die Weitergabe und Verteilung von Änderungen auf Elemente oder Objekte steuern.

Beispielszenarien:

[0149] Als Nächstes werden mehrere Beispielszenarien dargestellt, die die Funktionen, Fähigkeiten und den Betrieb flexibler Konfigurationen in Prozessbetrieben oder Prozess-Steuerungssystemen veranschaulichen. Diese Beispielszenarien können von einem oder mehr Bestandteilen der Methoden umgesetzt werden, die von **Fig. 1–Fig. 7** oder Kombinationen davon veranschaulicht sind. Diese Beispielszenarien sollen nicht einschränkend sein, und werden zur Veranschaulichung einiger der hier beschriebenen Konzepte, Vorteile und Nutzungen der Methoden, Systeme und Verfahren geliefert. Des Weiteren stellen diese Szenarien keinen erschöpfenden Satz an möglichen Szenarien dar. Ferner können diese Beispielszenarien mühelos auf andere Elemente und Objekte innerhalb der Betriebsanlage oder des Prozess-Steuerungssystems, wie beispielsweise Systemkonfigurationsobjekte, Grafik- und Anzeigeobjekte, Instanzobjekte, Prozesselementobjekte, Klassenobjekte und ähnliches angewendet werden, obwohl sich diese Beispielszenarien in der Regel auf ein Element oder Objekt der Bibliothek beziehen.

[0150] In einem ersten in **Fig. 8** veranschaulichten Beispielszenario verrichtet ein Anwender Aufgaben oder führt einen oder mehr Befehle zur Erstellung, Änderung oder Modifizierung eines Bibliothekelements oder -Objekts **1000** aus. In einer Ausführungsform werden die Befehle auf einem Konfigurationssystem, beispielsweise über die Arbeitsstation **14** oder eine andere Anwender-Schnittstelle empfangen. Wenn der Anwender das Bibliothekelement oder -Objekt **1000** erstellt oder bearbeitet und es speichert, wird ein Entwurf **1002** des Elements gespeichert. Der Anwender kann zusätzliche Bearbeitungen am Entwurf **1002** vornehmen. Irgendwann genehmigt der Anwender die Änderungen und veröffentlicht **1005** das Element. Die Veröffentlichung **1005** führt zu einer neuen Edition **1008** des Bibliothekelements **1000**, da das Bibliothekelement **1000** in diesem ers-

ten Beispielszenario keine untergeordneten Objekte hat. Die neue Edition **1008** steht zur Instantiierung in einer Laufzeit-Umgebung der Betriebsanlage oder Prozess-Steuerungsumgebung zur Verfügung.

[0151] In einem zweiten in **Fig. 9** veranschaulichten Beispielszenario kann ein Bibliothekelement **1010** ein untergeordnetes Element oder Objekt **1012** haben; beispielsweise ist das Bibliothekelement **1010** mit einem anderen Bibliothekelement oder einem Systemkonfigurationselement **1012** verlinkt. In diesem zweiten Szenario nimmt der Anwender Änderungen an Bibliothekelement **1010** vor und speichert es als Entwurf eines modifizierten Bibliothekelements **1013**. Irgendwann werden die Änderungen am Entwurf **1013** genehmigt und der Anwender veröffentlicht Entwurf **1013** zur Generierung einer aktuellen Edition der Bibliotheksedition **1014**, die die Änderungen aufweist. Der Anwender möchte steuern, wie die Änderungen am Bibliothekelement **1010** an seine untergeordneten Objekte **1012** verteilt oder propagiert werden, beispielsweise angesichts der Betriebsphase der Betriebsanlage, Personalbesetzung, prozess-technischer Fragen und/oder anderen Interessen. Die anwenderverwaltete Verteilung kann den Betrieb in Prozessbetrieben (wie beispielsweise in Raffinerien) verbessern, da Änderungen oder Modifikationen zu optimalen Zeiten verteilt werden können (beispielsweise ordnungsgemäße Unterstützungsteams sind zur Blockierung der Ventile vorhanden, usw.).

[0152] Zur Durchführung der gesteuerten Weitergabe oder Verteilung der aktuellen Edition **1014** von Bibliothekelement **1010** stellt der Anwender eine Richtlinieneinstellung ein, die den Steuerungsverteilungsprozess für Bibliothekelement **1010** steuert. In einer Ausführungsform kann die Richtlinieneinstellung dem Bibliothekelement **1010** entsprechen. Wahlweise kann die Richtlinieneinstellung dem Gebrauch oder untergeordneten Objekt **1012** des Bibliothekelements **1010** entsprechen. In einer Ausführungsform kann die Richtlinieneinstellung zwei Optionen aufweisen: automatisch **1018** und anwenderverwaltet **1015**. Wenn die Richtlinien auf automatisch **1018** eingestellt sind, wird die veröffentlichte, aktuelle Edition **1014** des Bibliothekelements **1010** (oder in der veröffentlichten Edition **1014** aufgenommene Änderungen) von dem Bibliothekelement **1010** auf das untergeordnete Element **1012** automatisch verteilt **1018**. Wenn die Richtlinieneinstellung auf anwenderverwaltet **1015** eingestellt ist, kann der Anwender einen Entwurf **1022** der aktuellen Edition **1014** erstellen. Der Entwurf **1022** kann wiederum als eine neue aktuelle Edition **1025** des untergeordneten Elements **1012** veröffentlicht werden.

[0153] Wie in **Fig. 10** dargestellt, kann die neue aktuelle Edition **1025** des Systemkonfigurationselements **1012** instanziiert sein. Wenn beispielsweise das Systemkonfigurationselement **1012** einem

Steuerungs- oder Funktionsblock entspricht, kann die aktuelle Edition **1025** in eine Download-Edition **1028** umgewandelt werden, die (beispielsweise automatisch oder gemäß Anweisungen vom Anwender) in das entsprechende Laufzeit-Element **1030** als eine ausführende Edition **1032** heruntergeladen werden kann. In einem anderen Beispiel, in dem das Systemkonfigurationselement **1012** einem Grafikelement entspricht, ist die aktuelle Edition des Grafikelements auf Laufzeit instanziiert, wenn eine Anzeigeansicht einschließlich des Grafikelements erstellt wurde.

[0154] Auf der anderen Hand, wenn wir uns wieder zurück auf **Fig. 9** beziehen, kann der Anwender die an dem Bibliothekselement **1010** vorgenommenen Änderungen auf jedes seiner untergeordneten Elemente **1012** wie gewünscht manuell propagieren oder verteilen, wenn die Verteilungsrichtlinieneinstellung anwenderverwaltet **1018** angibt. Im Grunde genommen wird die aktuelle Edition **1014** des Bibliothekselements **1010** aus der Bibliothek für ein untergeordnetes Element **1012** mittels eines Anwenderbefehls **1018** "gezogen" (der ausdrücklich von einem Anwender eingegeben oder in ein Skript eingefügt werden kann). In einem Beispiel wählt der Anwender ein untergeordnetes Zielelement (beispielsweise ein Systemkonfigurationselement **1012**) aus und fordert **1018** an, dass die aktuelle Edition **1014** des Bibliothekselements **1010** (oder die darin enthaltenen Modifikationen) auf das untergeordnete Zielelement **1012** propagiert werden. Nach der Weitergabe wird der Entwurf **1022** des Systemkonfigurationselements **1012** erstellt und gespeichert. Dem Anwender steht es frei, zusätzliche Bearbeitungen am Entwurfssystemkonfigurationselement **1022** vorzunehmen. Zu einem gewünschten Zeitpunkt veröffentlicht der Anwender das Entwurfssystemkonfigurationselement **1022**, was die neue Edition **1025** des Systemkonfigurationselements **1012** zur Folge hat, das dann instanziiert werden kann.

[0155] Die vorherigen Szenarien beschreiben die Modifikationen von Bibliothekselementen **1000** und **1010**. Jedoch können nicht nur Bibliothekselemente **1010** modifiziert werden. **Fig. 11** veranschaulicht ein Szenario, in dem der Anwender Änderungen direkt an dem Systemkonfigurationselement **1040** vornimmt, das keine Bibliotheksvorlage ist. Der Anwender speichert die Änderungen am Konfigurationselement **1040** als Entwurf **1042** ab, und nach der Autorisierung von Entwurf **1042**, veröffentlicht der Anwender den Entwurf **1042** in einer aktuellen Edition **1045** des Elements **1040**. In dem in **Fig. 11** dargestellten Beispiel fordert der Anwender die Erstellung der Download-Edition **1048** (die beispielsweise ablauffähig ist) an, und fordert danach an, dass die Download-Edition **1048** des Elements **1040** auf Laufzeit-Element **1050** übertragen wird, sodass eine ausführende Edition **1052** übernommen und vom entsprechenden Laufzeit-Element **1050** ausgeführt wird.

[0156] Sonstige mögliche Szenarien, die die flexiblen hier beschriebenen Methoden und Objekte veranschaulichen, sind:

1. Anwenderkopien eines Bibliothekselements, das über mehrere Editionen verfügt.

[0157] In diesem Szenario wird die Einstellung der Verteilungsrichtlinien auf anwenderverwaltet eingestellt. Ein Bibliothekselement mit mehreren Editionen ist in der Bibliothek vorhanden.

[0158] Der Anwender kopiert das Bibliothekselement zur Erstellung eines neuen Bibliothekselements mit einem anderen Namen. Die hierdurch entstandene Kopie des Bibliothekselements verfügt über eine einzige Edition, die der aktuellen Edition des ursprünglichen Bibliothekselements entspricht. Jedoch ist das neue Bibliothekselement lediglich eine Kopie und darauffolgende Änderungen am ursprünglichen Bibliothekselement wirken sich nicht auf das Element aus.

2. Der Anwender leitet ein Bibliothekselement von einem anderen Bibliothekselement ab.

[0159] Ein Element (wie beispielsweise eine Anzeige oder Abdeckung) ist in der Bibliothek vorhanden. Das Element hat einen einzigen Tank, der beispielsweise einen Transmitter und zwei Ventile aufweist.

[0160] Der Anwender wählt einen Befehl zur Erstellung des Entwurfs eines neuen Bibliothekselements aus, der von dem ersten Bibliothekselement abgeleitet wurde. Der Anwender ändert daraufhin den Entwurf, um einen zweiten Transmitter hinzuzufügen, speichert den Entwurf ab und veröffentlicht ihn als erste Edition des abgeleiteten Bibliothekselements.

[0161] Im Folgenden modifiziert der Anwender das ursprüngliche Bibliothekselement so, dass ein drittes Ventil hinzugefügt wird, speichert den Entwurf des ursprünglichen Bibliothekselements ab, und veröffentlicht den Entwurf in einer Edition. Der Anwender verteilt daraufhin die Edition des modifizierten, ursprünglichen Bibliothekselements. In Folge der Verteilung enthält das abgeleitete Bibliothekselement jetzt drei Ventile.

3. Der Anwender erstellt eine Instanz des Bibliothekselements, das mehrere Editionen hat.

[0162] Die Verteilungseinstellung ist auf anwenderverwaltet eingestellt und in diesem Szenario steht nur eine aktuelle Edition auf einmal zur Verfügung. Der Anwender erstellt ein Bibliothekselement sowie auch eine erste Instanz des Bibliothekselements. Der Anwender modifiziert daraufhin das ursprüngliche Bibliothekselement, speichert die Modifikation als einen Entwurf ab und veröffentlicht das modifizierte ursprüngliche Bibliothekselement. Die Veröffentlichung hat eine neue Edition des ursprünglichen Bibliothekselements zur Folge.

[0163] Als Nächstes erstellt der Anwender eine zweite Instanz des Bibliothekelements. Die zweite Instanz basiert auf der aktuellen Edition des Bibliothekelements (beispielsweise, auf dem modifizierten, ursprünglichen Bibliothekelement), wobei jedoch die erste Instanz unberührt bleibt. Sprich, die erste Instanz basiert weiterhin auf dem ursprünglichen nicht modifizierten Bibliothekelement.

[0164] In mit einer Ausführungsform bezieht sich die neue Instanz oder Nutzung auf die aktuelle, verfügbare Edition des Bibliothekelements, wenn eine neue Instanz oder Nutzung vom über mehrere Editionen verfügbenden Bibliothekelement erstellt wird.

4. Der Anwender erstellt die Nutzung eines Bibliothekelements, das mehrere Editionen hat.

[0165] Dieses Szenario ähnelt dem vorherigen Szenario, geht jedoch mit der Nutzung eines Bibliothekelements anstelle einer Instanz einher. In diesem Szenario ist nur eine aktuelle Edition auf einmal verfügbar. Sollte eine neue Nutzung von einem Element erstellt worden sein, die mehreren Editionen hat, basiert die neue Nutzung auf der aktuellen verfügbaren Edition des Bibliothekelements.

5. Der Anwender modifiziert das über eine Nutzung verfügbende Bibliothekelement.

[0166] In mit diesem Szenario basiert das Systemkonfigurationselement auf einem Bibliothekelement. Das Systemkonfigurationselement weist die Nutzung (wie beispielsweise eine zusammengesetzte Funktionsblocknutzung innerhalb eines Moduls) des Bibliothekelements auf. Die Verteilung von Änderungen ist auf anwenderverwaltet eingestellt.

[0167] Der Anwender ändert das Bibliothekelement, auf dem das Systemkonfigurationselement basiert und beobachtet, dass das Bibliothekelement jetzt einen visuellen Indikator hat (wie beispielsweise ein gestapeltes Symbol) um anzuzeigen, dass mehrere Editionen des Bibliothekelements vorhanden sind. Der Anwender öffnet eine Verweissansicht im Bibliothekelement und sieht, dass eine Nutzung vorliegt (die zusammengesetzte Funktionsblocknutzung im Modul), die nicht die neue (jetzt aktuelle) Edition des Bibliothekelements verwendet, die beispielsweise vom Systemkonfigurationselement übernommen wurde.

6. Der Anwender modifiziert das Bibliothekelement, das eine Nutzung in einem anderen Bibliothekelement hat.

[0168] Dieses Szenario ähnelt dem vorherigen Szenario, jedoch ist dabei die Nutzung in einem anderen Bibliothekelement anstelle im Systemkonfigurationselement mit eingeschlossen. In diesem Szenario steuern die Verteilungsrichtlinien nur die Verteilung auf Systemkonfigurationselemente und bleiben auf anwenderverwaltet eingestellt. An einem Bibliothekelement vorgenommene Änderungen werden auto-

matisch auf Nutzungen innerhalb anderer Bibliothekelemente ungeachtet der Richtlinieneinstellung verteilt. Sprich, für die Verteilung auf Bibliothekelemente wird der Standardwert "automatisch" verwendet, und sie kann dann nicht mehr modifiziert werden. Die Verteilung ist rekursiv, sollten die betroffenen Bibliothekelemente selbst von anderen Bibliothekelementen verwendet werden; neue Editionen werden für alle Bibliothekelemente erstellt, die von der Änderung eines anderen Bibliothekelements betroffen sind.

[0169] In bestimmten Ausführungsformen stehen änderbare Richtlinieneinstellungen für Bibliothekelemente zur Verfügung und verschiedene änderbare Richtlinieneinstellungen stehen für nicht-Bibliothekelemente, wie beispielsweise Konfigurationselemente, zur Verfügung.

7. Der Anwender modifiziert das über eine Instanz verfügbende Bibliothekelement.

[0170] In diesem Szenario wird eine Instanz (wie beispielsweise eine Modulinstanz) des Bibliothekelements als ein Systemkonfigurationselement gespeichert. Die Verteilung ist auf anwenderverwaltet eingestellt.

[0171] Der Anwender ändert das Bibliothekelement und beobachtet, dass das Bibliothekelement jetzt einen visuellen Indikator hat (wie beispielsweise ein gestapeltes Symbol), um anzuzeigen, dass mehrere Editionen vorliegen. Der Anwender öffnet eine Verweissansicht im Bibliothekelement und sieht, dass es eine Systemkonfigurationsinstanz gibt, die nicht die neue (jetzt aktuelle) Edition des Bibliothekelements benutzt.

8. Der Anwender modifiziert das Bibliothekelement, das ein abgeleitetes Bibliothekelement hat.

[0172] In diesem Szenario wird ein übergeordnetes Bibliothekelement von einem abgeleiteten Bibliothekelement benutzt, dass wiederum von einem Systemkonfigurationselement verwendet wird. Die Verteilung auf Systemkonfigurationselemente ist auf anwenderverwaltet eingestellt, die Verteilung auf Bibliothekelemente ist nicht änderbar und ihr Standardwert ist auf automatisch eingestellt.

[0173] Der Anwender modifiziert das übergeordnete Bibliothekelement. Da die Verteilung des Bibliothekelements auf automatisch eingestellt wurde, wird das abgeleitete Bibliothekelement automatisch aktualisiert. Jedoch wird dabei das auf der abgeleiteten Bibliothek basierende Systemkonfigurationselement nicht aktualisiert, da die Verteilung für Konfigurationselemente auf anwenderverwaltet eingestellt ist. In diesem Sinne verwendet das Systemkonfigurationselement weiterhin die vorherige Edition des abgeleiteten Bibliothekelements. Der Anwender kann dies sehen, wenn er die Verweissansicht des übergeordneten Bibliothekelements öffnet und beobachtet,

dass das abgeleitete Bibliothekelement, jedoch nicht das Systemkonfigurationselement, die aktuelle Edition des übergeordneten Bibliothekelements benutzt.

9. Der Anwender aktualisiert eine Nutzung im Kontext des Bibliothekelements.

[0174] Wenn wir auf dem vorherigen Szenario aufbauen, sieht der Anwender ein gestapeltes Symbol, das dem Bibliothekelement entspricht, und anzeigt, dass nicht alle Instanzen des Bibliothekelement aktualisiert worden sind. Die Verteilung auf Systemkonfigurationselemente ist auf anwenderverwaltet eingestellt.

[0175] Der Anwender öffnet die Verweisansicht im Bibliothekelement und sieht zwei Reihen, die zwei Anzeigen entsprechen, wobei jede das Bibliothekelement benutzt. Der Anwender ändert daraufhin eine der Anzeigen zum Gebrauch der aktuellen aktualisierten Edition des Bibliothekelements. Wenn der Anwender die Anzeige öffnet, sieht er die aktuelle aktualisierte Edition des Bibliothekelements; so zeigen beispielsweise die Anzeige sowie auch beliebige Nutzungen des Bibliothekelements an, dass sie die aktuelle aktualisierte Edition benutzen.

[0176] Später öffnet der Anwender die Verweisansicht auf der zweiten Anzeige und ändert die Nutzung, um die aktuelle aktualisierte Edition des Bibliothekelements zu verwenden. Der Anwender speichert die zweite Anzeige und die Darstellung der Nutzung gibt an, dass die aktuelle aktualisierte Edition des Bibliothekelements zur Zeit verwendet wird.

[0177] In Folge der vom Anwender vorgenommenen Änderungen existieren keine mit der ersten Edition des Bibliothekelements verlinkte Elemente im System. Die erste Edition wird daher automatisch vom System gelöscht. In anderen Ausführungsformen kann die erste Edition auf deaktiviert oder inaktiv eingestellt sein.

[0178] Der Anwender kann nicht aktuelle Nutzungen mehrfach auswählen und sie auf die aktuelle Edition aktualisieren. Der Indikator des Bibliothekelements kann verschwinden wenn alle Instanzen die neueste Edition verwenden. Falls eine Nutzung oder Instanz nicht während eines Mehrfach-Auswahlbetriebs aktualisiert werden kann, wird die Nutzung oder Instanz übersprungen und die Aktualisierung setzt mit dem nächsten Element fort. Zum Abschluss der Aktualisierung wird dem Anwender angezeigt, welche Element nicht erfolgreich aktualisiert werden konnten.

10. Der Anwender aktualisiert die Nutzung eines Bibliothekelements im Zusammenhang mit dem Bibliothekelement.

[0179] In mit einer Ausführungsform werden die an einem Bibliothekelement vorgenommenen Änderungen automatisch an Nutzungen innerhalb ande-

rer Bibliothekelemente ungeachtet der Richtlinien-einstellungen für die Systemkonfigurationsverteilung verteilt. In bestimmten Ausführungsformen können verschiedene Richtlinieneinstellungen für die Bibliotheksverteilung verfügbar sein, die von einem über eine Autorisierung verfügenden Anwender geändert werden können.

11. Der Anwender aktualisiert eine Instanz in Zusammenhang mit dem Bibliothekelement.

[0180] Dieses Szenario erweitert sich auf ein herkömmliches klassenbasiertes Modulverhalten. Ein Bibliothekelement ist vorhanden und eine Instanz des Bibliothekelements ist als ein Systemkonfigurationselement gespeichert. Die Instanz benutzt jedoch nicht die aktuelle Edition des Bibliothekelements. Die Verteilung auf Systemkonfigurationsobjekte ist auf anwenderverwaltet eingestellt.

[0181] Der Anwender wählt das Bibliothekelement aus und öffnet die Verweisansicht. Der Anwender sieht, dass die Systemkonfigurationsinstanz nicht die aktuelle Edition des Bibliothekelements benutzt, wählt daher die Instanz aus und führt den Befehl "Als aktuell festlegen" aus. Dieser Befehl führt zu einen neuen Entwurf der Instanz, die die aktuelle Edition des Bibliothekelements verwendet. Der Anwender kann daraufhin Maßnahmen ergreifen, um den Entwurf der Systemkonfigurationsinstanz zu bearbeiten und/oder zu veröffentlichen.

12. Der Anwender aktualisiert ein abgeleitetes Element in Zusammenhang mit dem Bibliothekelement.

[0182] Dem Anwender wird gestattet, auf anderen Bibliothekelementen basierende Bibliothekelemente zu erstellen. Beispielsweise kann der Anwender ein übergeordnetes Bibliothekmodul ohne Interlocks (Sperrungen) und danach ein abgeleitetes Bibliothekmodul erstellen, das über alle Fähigkeiten des übergeordneten Bibliothekmoduls samt Interlocks verfügt. Am Algorithmus des übergeordneten Bibliothekmoduls vorgenommene Änderungen könnten zum abgeleiteten Bibliothekmodul durchfließen, sodass der Anwender nützen kann, dass der Algorithmus an nur einer Stelle definiert wurde.

13. Der Anwender aktualisiert eine Nutzung im Kontext der Nutzung.

[0183] In diesem Szenario möchte der Anwender die Nutzungen aktualisieren, um die aktuelle Edition verwenden zu können. Beispielsweise wird ein Steuerungsmodulobjekt als ein Systemkonfigurationsobjekt gespeichert, das eine Funktionsblocknutzung eines einzigen zusammengesetzten Editionsblocks mit einem einzigen Eingabekonnektor aufweist. Der einzige zusammengesetzte Editionsblock ist ein Bibliothekelement und die Verteilung auf Systemkonfigurationsobjekte ist auf anwenderverwaltet eingestellt. Der Anwender modifiziert den zusammengesetzten

Block in der Bibliothek, um einen anderen Eingabekonnektor hinzuzufügen, um somit eine neue aktuelle Edition für den zusammengesetzten Bibliotheksblock zu erstellen. Der Anwender kann eine Grafik sehen, die angibt, dass der zusammengesetzte Bibliotheksblock mehrere Editionen hat.

[0184] Als Nächstes wählt der Anwender die Funktionsblocknutzung auf einem bestimmten Steuerungsmodul sowie auch den "Als aktuell festlegen" Befehl aus. Die Nutzung des bestimmten Steuerungsmoduls wird aktualisiert, sodass die aktuelle Edition eingesetzt werden kann und so den zusätzlichen Eingabekonnektor anzeigt. Sollte die vorherige zusammengesetzte Blockedition nicht im System verwendet worden sein, ändert sich die Grafik, um anzuzeigen, dass der zusammengesetzte Bibliotheksblock nicht mehr über mehrere Editionen verfügt.

14. Der Anwender aktualisiert die Nutzung des Bibliothekelements in Zusammenhang mit der Nutzung des Bibliothekelements.

[0185] Wie die abgeleiteten Elemente können auch die Nutzungen (beispielsweise Funktionsblock) von Bibliothekelementen in anderen Bibliothekelementen ungeachtet der Verteilungsrichtlinieneinstellung auf Konfigurationselemente auf automatische Aktualisierungen standardmäßig eingestellt werden. Jedoch in Fällen, in denen eine Änderung nicht automatisch auf ein anderes Bibliothekelement aufgrund eines Konflikts verteilt werden konnte, kann der Anwender die Nutzung des Bibliothekelements manuell berichtigen.

15. Der Anwender aktualisiert eine Instanz in Zusammenhang mit der Instanz.

[0186] In diesem Szenario hat ein Bibliothekmodul eine einzigen Edition und eine Instanz des Bibliothekmoduls wurde als ein Systemkonfigurationselement gespeichert. Die Verteilung auf Systemkonfigurationselemente ist auf anwenderverwaltet eingestellt. Der Anwender modifiziert das Bibliothekmodul und veröffentlicht die Änderungen. Eine Grafik auf der mit dem Bibliothekmodul verbundenen Anzeige gibt an, dass der zusammengesetzte Bibliotheksblock mehrere Editionen hat.

[0187] Der Anwender wählt die Modulinstanz aus sowie auch den Befehl "Als aktuell festlegen". Ein Entwurf der Instanz wird erstellt, der die aktuelle modifizierte Edition des Bibliothekmoduls verwendet. Der Anwender kann daraufhin den Entwurf veröffentlichen, um eine neue aktuelle Edition der Modulinstanz zu erstellen. Sollte die vorherige Bibliotheksmoduledition nirgendwo im System mehr verwendet werden, wird der mit der Bibliothek verbundene mehrere Editionen-Indikator von der Anzeige entfernt.

16. Der Anwender aktualisiert alle in einem Geräteelement enthaltenen Instanzen.

[0188] Dieses Szenario ähnelt vorherigen Szenarios, außer dass die Aktualisierung im Kontext eines Geräteelements vorgenommen wird, wie beispielsweise eines Bereichs, einer Zelle oder einer Einheit. Ein Bibliothekmodul mit einer einzigen Edition ist vorhanden und zwei Instanzen des Moduls sind in der Systemkonfiguration unter einem Bereich vorhanden. Die Verteilung auf die Systemkonfiguration ist auf anwenderverwaltet eingestellt. Der Anwender modifiziert das Bibliothekmodul und eine Grafik wird geliefert, die angibt, dass der zusammengesetzte Bibliotheksblock mehrere Editionen hat.

[0189] Auf einer dem Kontext entsprechenden Anzeige (beispielsweise, ein Bereich) ist der Bereich ebenfalls als ein Bereich mit nicht aktuellen Instanzen markiert. Nach der Erweiterung der Bereichsansicht werden zwei Modulinstanzen als nicht aktuell markiert. Der Anwender wählt den Bereich und den Befehl "Als aktuell festlegen" aus. Entwürfe der Instanzen, die die aktuelle Edition des Bibliothekmoduls einsetzen, werden erstellt. Der Anwender kann die Entwürfe veröffentlichen, um neue aktuelle Editionen der Modulinstanzen zu erstellen und um somit zu bewirken, dass die Kennzeichnung "nicht aktuell" von der Anzeige verschwindet. Sollte die vorherige Bibliotheksmoduledition nirgendwo im System mehr verwendet werden, kann der mit dem Bibliothekmodul verbundene mehrere Editionen-Indikator ebenfalls aus der Anzeige verschwinden.

17. Der Anwender aktualisiert ein abgeleitetes Element in Zusammenhang mit dem abgeleiteten Element.

[0190] Die Bibliothek speichert zwei Elemente, die Bibliotheksanzeige DISP1 und die von DISP1 abgeleitete Bibliotheksanzeige DISP2. Der Anwender ändert DISP1, um ein Textfeld hinzuzufügen, veröffentlicht DISP1 und verteilt DISP1. Dementsprechend wird DISP2 automatisch mit dem Textfeld aktualisiert.

[0191] In ähnlicher Weise speichert die Bibliothek ein Bibliotheksgerät des Typs VENTIL, dem u. U. eine Abdeckung zugeordnet wurde. Die Bibliothek speichert ebenfalls ein Bibliotheksgerät des Typs DVC, das von VENTIL abgeleitet ist und zusätzliche Informationen zur von VENTIL benutzten Abdeckung aufweist. Ferner speichert die Bibliothek ein anwendungsspezifisches Bibliotheksgerät des Typs MfctrDVC, das wiederum noch mehr Informationen der Abdeckung für DVC hinzufügt. Wenn der Anwender einen Parameterwert für VENTIL ändert, wird der Parameterwert in DVC sowie auch in MfctrDVC automatisch aktualisiert, wenn DVC und MfctrDVC den Parameterwert nicht überschreiben.

[0192] Als eine Nachverfolgung des obigen kann der Anwender ein anwendungsspezifisches Gerät des Typs MfctrDVC-CriticalService erstellen, indem er diesen Gerätetyp vom MfctrDVC Gerätetyp ablei-

tet. Somit überschreibt der Anwender einen der Parameter. Wenn der Anwender in diesem Nachverfolgungsszenario denselben Parameterwert in VENTIL ändert, wird der Parameterwert in DVC und MfctrDVC automatisch aktualisiert, da DVC und MfctrDVC den Parameterwert nicht überschreiben. Der Parameterwert wird jedoch nicht in MfctrDVC-CriticalService geändert, da der Anwender den Parameterwert überschrieben hat.

[0193] In diesem Sinne werden in einer Ausführungsform die von anderen Elementen abgeleiteten Bibliothekelemente ungeachtet der Verteilungsrichtlinieneinstellung für Systemkonfigurationselemente stets aktualisiert, wenn das übergeordnete Bibliothekelement geändert wird. Im Falle von Konflikten kann jedoch die aktuelle Edition eines übergeordneten Bibliothekelements nicht automatisch auf ein abgeleitetes Bibliothekelement propagiert werden, an dem strukturelle Änderungen vorgenommen wurden. Nachdem der Anwender das Problem behoben hat, kann der Anwender ein abgeleitetes Bibliothekelement manuell aktualisieren.

18. Der Anwender aktualisiert eine Nutzung, was zu einer Verletzung der Regeln führt.

[0194] In diesem Szenario ist die Verteilung auf Systemkonfigurationsobjekte auf automatisch eingestellt. Jedoch kann die Verteilung einer an einem Bibliothekelement vorgenommenen Änderung einige oder alle der untergeordneten Systemkonfigurationsinstanzen nicht aktualisieren, sollte eine Regel (beispielsweise eine Datenbankregel) verletzt worden sein. Diese Diskrepanz fällt dem Anwender bei der Speicherung eines Bibliothekelements u. U. nicht auf, wenn die Verteilung mit einem zu einem späteren Zeitpunkt stattfindenden Veröffentlichungsschritt verbunden ist. In der Tat könnte die neue Edition bereits von anderen verlinkten Elementen verwendet worden sein, bevor die Diskrepanz entdeckt wurde. In dieser Situation kann der Anwender das Bibliothekelement bearbeiten, um die Diskrepanz zu korrigieren, und um somit eine neue aktuelle Edition des Bibliothekelements zu erstellen, das für alle Instanzen benutzt werden kann, ohne dass dabei gegen Regeln verstoßen wird.

[0195] Zur Veranschaulichung wurde einer Instanz eine Aktualisierung auf eine neue Edition mit einem neuen Nutzungskonflikt mit dem Namen der Nutzung hinzugefügt. Die Bibliothek speichert beispielsweise das Bibliothekmodul LM mit einer Nutzung namens NUTZUNG1 ab. Die Systemkonfigurationsinstanzen MA und MB des Bibliothekmoduls LM sind im Bereich BereichA vorhanden. Der Anwender fügt daraufhin der Instanz MA die Nutzung NUTZUNG2 hinzu.

[0196] Der Anwender fügt eine Nutzung NUTZUNG2 dem Bibliothekmodul LM hinzu, veröffentlicht das Bibliothekmodul LM und verteilt es. Da die Instanz MA

bereits über eine Nutzung NUTZUNG2 verfügt, wurde Instanz MA nicht aktuell und der Indikator "nicht aktuell" entsprechend der Instanz MA gelassen. Instanz MB hat jedoch keine solche Nutzung und wurde erfolgreich aktualisiert.

19. Der Anwender speichert ein Bibliothekelement nach dem Umschalten von anwenderverwaltet auf automatische Systemkonfigurationsverteilung.

[0197] In diesem Szenario wird ein Bibliothekelement von mehreren Anzeigen benutzt. Eine Anzeige verwendet eine aktuelle Edition, während die andere Anzeige die vorherige Edition benutzt. Der Anwender schaltet von anwenderverwaltet auf automatische Verteilung um. Die Änderung der Verteilung von anwenderverwaltet auf automatische Verteilung wirkt sich nicht auf die Anzeigen aus.

[0198] Der Anwender modifiziert das Bibliothekelement, während die Systemkonfigurationsverteilung auf automatisch eingestellt ist. Infolgedessen werden alle Instanzen des Bibliothekelements aktualisiert, um die aktuelle modifizierte Edition zu verwenden. Da alle Systemkonfigurationsinstanzen aktualisiert wurden, wird kein Bezug auf ältere Editionen des Bibliothekelements mehr genommen und sie können automatisch gelöscht werden. Das Symbol für das Bibliothekelement ändert sich, um anzugeben, dass nur eine Edition vorhanden ist.

20. Der Anwender lädt eine Instanz herunter, die nicht die aktuelle Edition des Bibliothekelements verwendet.

[0199] Der Anwender kann eine Systemkonfigurationsinstanz auf ein Prozesselement herunterladen, die nicht die aktuelle Edition eines Bibliothekelements benutzt. Die Download-Anforderung kann dazu führen, dass eine Download-Warnung angezeigt wird.

21. Der Anwender lädt ein Objekt herunter, das nicht die aktuelle Edition eines Bibliothekelements verwendet.

[0200] Der Anwender kann eine Systemkonfigurationsinstanz auf ein Prozesselement herunterladen, die nicht die aktuelle Edition einer Nutzung eines Bibliothekelements umfasst. Die Download-Anforderung kann dazu führen, dass eine Download-Warnung angezeigt wird.

22. Der Anwender importiert eine Datei, die ein Bibliothekelement enthält.

[0201] In diesem Szenario importiert der Anwender eine Datei und dem Anwender wird eine Liste von Elementen in der Datei vorgelegt. Der Anwender kann von einem Importverhalten für jedes Element auswählen (wie beispielsweise überspringen, überschreiben, usw.).

[0202] Wie durch die weiter oben aufgeführten Beispielszenarien veranschaulicht wurde, kann der An-

wender mindestens die folgenden Aktionen oder Befehle mit Hilfe der hier beschriebenen flexiblen Konfigurationsmethoden, Systeme und Verfahren durchführen:

Bearbeiten – Öffnen und eventuelles Modifizieren eines Elements oder Objekts. Der Anwender kann einen Entwurf oder eine Edition öffnen.

Als Entwurf speichern – Speicherung eines geöffneten (und eventuell modifizierten) Elements als Entwurf.

Testen – Ein Entwurfobjekt ohne Herunterladung oder Instanziierung des Entwurfobjekts laufen.

Genehmigen – Autorisierung eines Entwurfs zur Veröffentlichung.

Veröffentlichen – Einen Entwurf auswählen und ihn in eine aktuelle Edition umwandeln. In der Regel wird die Veröffentlichung nach der Autorisierung des Entwurfs durchgeführt.

Erstellen – Erstellung eines Downloads oder einer Laufzeit-Edition der veröffentlichten oder aktuellen Edition.

Übertragen – Übertragung einer Edition auf die Laufzeit.

Herunterladen – Eine Ausführungsform des Übertragungsbefehls, der eine Download-Edition zur Laufzeit schickt. Generell wird ein Download im Kontext eines Systemkonfigurationselements durchgeführt.

Verteilen/Propagieren/Aktualisieren – Aktualisierung eines anderen verlinkten Objekts mit Modifikationen, die an einer bestimmten Objektedition vorgenommen wurden. Die Aktualisierung, Verteilung oder Weitergabe von Modifikationen auf ein bestimmtes Objekt kann zur Aktualisierung mehrerer anderer verlinkter Objekte führen, wie beispielsweise untergeordneter Objekte und/oder eines übergeordneten Objekts.

Aktualisierung von der Bibliothek aus – Aktualisierung (beispielsweise Verteilen oder Propagieren) von Modifikationen an einer Bibliothekobjektedition an untergeordnete Objekte, die von dem Bibliothekobjekt abgeleitet wurden. Eine Aktualisierung oder Weitergabe des Bibliothekobjekts kann zu Aktualisierungen mehrerer Stufen verlinkter untergeordneter Objekte führen. In diesem Sinne kann der Anwender beispielsweise eine Instanz oder eine abgeleitete Elementnutzung von einer aktuellen Edition eines Bibliothekelements generieren.

Löschen – Löschung eines Objekts oder Elements von dem Speicher.

Überprüfen – Überprüfung eines Elements oder Objekts auf Fehler. Entwürfe wie auch Editionen können überprüft werden.

Unterschiede – Festlegung der Unterschiede zwischen verschiedenen Editionen eines Objekts oder Elements.

Verweise – Festlegung der Verweise, die mit einem Element oder Objekt verbunden sind.

[0203] In bestimmten Ausführungsformen sind eine oder mehr der weiter oben beschriebenen Aktionen oder Befehlen ausgeblendet oder nicht für den Endanwender verfügbar. Beispielsweise können die Befehle Generieren und Übertragen im Download Befehl zusammengelegt werden oder es kann dem Anwender nicht erlaubt werden, ein Element zu Löschen, solange er nicht die ordnungsgemäße Autorisierung dazu hat.

[0204] In einer Ausführungsform kann der Anwender vom flexiblen Konfigurationssystem anfordern, dass alle Unterschiede zwischen einem Entwurf, einer aktuellen veröffentlichten Edition und einer Download-Edition für ein Element oder Objekt angezeigt werden. Beispielsweise kann der Anwender anfordern, das ihm der Unterschied zwischen einem Entwurf und der aktuellen veröffentlichten Edition, einem Entwurf und beliebigen Editionen in Versionskontrolle, einem Entwurf und der Download-Edition, zwei verschiedenen Entwürfen, zwei verschiedenen aktuellen Editionen, einer aktuellen Edition und der Download-Edition und anderen Kombinationen angezeigt wird.

[0205] In einer Ausführungsform kann der Anwender das Konfigurationssystem auffordern, alle direkt-verlinkten untergeordneten Objekte eines bestimmten Objekts anzuzeigen, ohne Rücksicht darauf, ob ein direkt-verlinktes untergeordnetes Objekt die aktuelle Edition des bestimmten Objekts benutzt oder nicht. Im umgekehrten Fall kann der Anwender in einer Ausführungsform eine Ansicht anfordern, um zu bestimmen, ob ein untergeordnetes Objekt die aktuelle Edition seines übergeordneten Objekts benutzt.

Pakete:

[0206] In einigen Ausführungsformen können die oben beschriebenen Techniken von Entwürfen, Veröffentlichung und Instanziierung auf Gruppen von Objekten angewandt werden. Eine Gruppe oder Mehrzahl von Objekten wird in diesem Dokument im Allgemeinen als ein „Paket“ bezeichnet. In einer Ausführungsform werden die Elemente eines Pakets von einem Benutzer ausgewählt. Mindestens einige der Elemente eines Pakets werden Out-of-the-box (OOB) in einigen Fällen von einem Hersteller oder Provider bereitgestellt.

[0207] Beispielsweise kann ein OOB-Paket einen Satz von „Starter“-Bibliothekselementen (z. B. Module, Verbundelemente, Anzeigen, Vorrichtungen usw.) umfassen. Ein Benutzer kann unterschiedliche Elemente aus einer Starter-Sammlung von Elementen auswählen, um eine Prozesslösung zu schaffen, wie z. B. eine Regelschleife oder ein Prozessmodell. Wenn ein Benutzer angepasste Objekte zu einem OOB-Paket hinzufügt, dann kann der Benutzer das erweiterte OOB-Paket als ein Paket benennen, welches sich von dem Starterpaket unterscheidet.

[0208] Ein weiteres beispielhaftes Paket kann eine Konfigurationsimplementierung umfassen, die darauf zugeschnitten ist, um ein spezifisches Prozessproblem zu lösen, z. B. ein Boilerpaket oder ein Reaktorpaket. Solche Pakete werden typischerweise als Komplettlösungen für Prozesssteuerungsprobleme verwendet, und werden als solche in diesem Dokument als „Lösungspakete“ bezeichnet. In einer Ausführungsform umfassen Pakete optionale Instanzierungsregeln die angewandt werden, wenn ein Benutzer eine Instanz des Pakets herstellt. Wenn z. B. das Boilerpaket instantiiert wird, kann eine Regel festlegen, wie alle Elemente in dem Paket entsprechend einer Benennungskonvention unter Verwendung der Ersetzung von Textelementen benannt werden.

[0209] Ähnlich wie bei einem Element oder Objekt kann einer oder mehrere Entwürfe für ein aktuelles Paket erstellt oder bearbeitet werden. Entwurfspakete können in Ausgaben veröffentlicht werden und Ausgaben können zur Instantiierung ausgewählt werden. Wenn ein Benutzer bei einer Ausführungsform eine neue Ausgabe eines Pakets speichern möchte, dann kann die neue Ausgabe die vorherige Ausgabe ersetzen oder die neue Ausgabe kann gleichzeitig mit der vorherigen Ausgabe gespeichert werden.

[0210] Diese Techniken von Paketentwürfen, Paketveröffentlichung und Paketinstantiierung bieten eine weitere Ebene an Flexibilität für den Benutzer. In einem beispielhaften (jedoch nicht begrenzenden) Szenario wird ein Bibliothekspaket erstellt und als ein erster Entwurf des Pakets gespeichert. Beispielsweise entspricht das Bibliothekspaket einer Vorlagenkonfiguration eines Boilers oder eines Reaktors. Der erste Entwurf des Pakets wird veröffentlicht und als eine erste Ausgabe gespeichert, und in die Prozessanlage oder das -system instantiiert. Die erste Ausgabe wird nachfolgend modifiziert (z. B. um Probleme zu korrigieren und/oder um Funktionen zu dem Boiler oder zu dem Reaktor hinzuzufügen), und wird daraufhin als eine zweite Ausgabe des Pakets veröffentlicht. Nach dem Empfang einer Benutzerangabe zur Instantiierung der zweiten Ausgabe des Pakets werden nur die in der zweiten Ausgabe des Pakets enthaltenen Elemente instantiiert, die von der ersten Version des Pakets aktualisiert sind. Weiterhin kann ein Benutzer einen Zeitpunkt der Instanti-

ierung unterschiedlicher aktualisierter Elemente angeben, wie dies aufgrund von Logistik-, Leistungs-, Bandbreiten- und/oder anderen Bedingungen erforderlich ist. Wenn die Verwendung der zweiten Ausgabe aus irgendeinem Grund beendet werden muss, kann die erste Ausgabe erneut in die Prozessanlage oder das -system instantiiert werden.

Grafik- oder Anzeigeelemente oder -objekte

[0211] Indem wir uns nun einer Abhandlung flexibler Grafik- oder Anzeigenelemente oder -objekte zuwenden, ist eine „Anzeige“, wie sie im Allgemeinen in diesem Dokument bezeichnet wird, eine Grafikentität (z. B. eine sehr granulare Entität), die von einem Bediener in Laufzeit einzeln geöffnet und betrachtet werden kann. Typischerweise sind flexible Grafik- oder Anzeigenelemente oder -objekte von anderen Grafikobjekten konfiguriert und können in einer Objektbibliothek oder in einer Systemkonfigurationsdatenspeicherungsentität gespeichert werden. Anzeigen können durch Herunterladen einer veröffentlichten Ausgabe zur Ausführung während der Laufzeit des Prozesselements instantiiert werden oder können in Echtzeit instantiiert werden, wie z. B. wenn ein Bediener die Erstellung einer bestimmten Anzeigenansicht anfordert. Typen von Anzeigen, die durch flexible Grafikanzeigen oder -objekte definiert sein können, umfassen z. B.:

1) Betriebsanzeigen: Betriebsanzeigen stellen im Allgemeinen ein Fenster in einen Prozess oder eine Darstellung eines Prozesses bereit, während er in einer Prozessanlage ausgeführt wird. Typischerweise werden Anzeigenansichten vom GEM und anderen Bibliotheks- und/oder Systemkonfigurationsobjekten konfiguriert und können auf einer lokalen oder entfernten Benutzerschnittstelle instantiiert werden.

2) Dashboard-Anzeigen: Dashboard-Anzeigen stellen Schlüsselkennzahlen für Bediener und Anlagenpersonal bereit. Dashboard-Anzeigen können von Bibliotheks- und/oder Systemkonfigurationsobjekten konfiguriert sein und auf einer lokalen oder entfernten Benutzerschnittstelle instantiiert werden. Bei einigen Ausführungsformen können begrenzte Kapazität, Merkmale oder auf der Dashboard-Anzeige enthaltene Grafiken in der Laufzeit durch einen Bediener konfiguriert und instantiiert werden. Fähigkeiten, Merkmale und Grafikelemente sowie Formen, die auf einem Dashboard enthalten sind, werden in diesem Dokument im Allgemeinen als „Gadgets“ bezeichnet. Im Allgemeinen ist ein Gadget ein GEM, welches als während der Laufzeit verfügbar markiert wurde und auf Dashboard-Anzeigen positioniert werden kann. Als solcher bezieht sich der Begriff „GEMs“, wie er in diesem Dokument verwendet wird, gattungsgemäß sowohl auf GEMs die Gadgets sind, als auch auf GEMs die keine Gadgets sind. Beispiele von Dashboards und Gadgets sind in der

ebenfalls anhängigen vorläufigen U.S.-Patentanmeldung Nr. 61/711,105, eingereicht am 8. Oktober 2012, mit dem Titel „Configurable User Displays in a Process Control System“ (Konfigurierbare Benutzeranzeigen in einem Prozesssteuerungssystem) vorhanden, deren gesamte Inhalte hiermit durch Bezugnahme enthalten sind.

3) Layoutanzeigen: Layoutanzeigen stellen Regionen oder Bereiche auf einer Benutzerschnittstelle bereit, um einen Benutzer in die Lage zu versetzen, andere Anzeigen darauf zu organisieren. Ohne eine Layoutanzeige ist ein Benutzer in der Lage, eine Anordnung von Anzeigen auf einem einzelnen Überwachungsbildschirm oder über mehrere Überwachungsbildschirme zu erstellen.

4) Formanzeigen: Formanzeigen stellen eine Benutzerschnittstelle zur Dateneingabe während der Konfiguration oder der Laufzeit bereit. Benutzer können Formanzeigen zur Erstellung oder Modifikation von Objekten verwenden, um Konfigurations- oder Laufzeitdaten einzugeben, um Werte einzugeben und um im Allgemeinen Anzeigenelemente und -objekte zu handhaben.

[0212] Eine Anzeige enthält typischerweise eine oder mehrere Ansichten, wovon jede eine andere visuelle Darstellung der Anzeige ist. Jede Ansicht einer Anzeige kann Formen (z. B. Rechtecke, Ellipsen und andere Formen) und/oder Steuerungen (z. B. Etiketten oder Textboxen) umfassen und kann ein entsprechendes Objekt aufweisen, welches in der Objektbibliothek oder in einer Systemkonfigurationsdatenspeicherungsentität gespeichert ist. Einige Formen, wie z. B. ein Rechteck, ein Textfeld oder eine Stapelplatte können in das Konfigurationssystem eingebaut und dort gespeichert werden (z. B. sind kompiliert und im System registriert). Andere Formen können vom Benutzer erstellt werden, und entsprechende Formobjekte können in der Objektbibliothek oder als ein Systemkonfigurations-GEM oder -Gadget gespeichert werden.

[0213] Wie zuvor abgehandelt, ist ein GEM oder ein Gadget eine wiederverwendbare Form, die eine oder mehrere Formen mit Verhalten kombiniert. GEMs und Gadgets können von einem Benutzer erstellt werden, in Ausgaben veröffentlicht und in dem Konfigurationssystem als Objekte gespeichert werden, wie z. B. in der Bibliothek oder in der Systemkonfiguration. GEMs können mit anderen Objekten verbunden werden, und nachfolgende Änderungen an den GEM-Objekten können an alle Verwendungen des GEMs in anderen GEMs und Anzeigen weitergegeben werden, z. B. auf eine Art und Weise, wie zuvor beschrieben. In einer Ausführungsform werden Out-of-the-box(OOB)-Formen, GEMs, Gadgets und Anzeigen (die in einigen Fällen der bestimmten Industrie eines Kunden entsprechen) in einer Bibliothek eines Konfigurationssystems, z. B. als ein Satz von primiti-

ven Objekten oder als ein „Startersatz“ von Objekten bereitgestellt.

[0214] Wie Anzeigen kann ein GEM eine oder mehrere visuelle Darstellungen oder Ansichten enthalten. Jede der GEM-Ansichten umfasst eine oder mehrere integrierte und/oder erstellte Formen. **Fig. 12** veranschaulicht drei unterschiedliche Ansichten eines beispielhaften Ventil-GEM, d. h. eine Vertikalansicht **1060a**, eine Horizontalansicht **1060b** und eine alternative Horizontalansicht **1060c**. Ansichten von GEMs können verwendet werden, um unterschiedliche Ebenen von Details, Anordnungen von Formen, oder einen anderen Schwerpunkt von zu dem GEM gehörenden Informationen zu präsentieren.

[0215] Zusätzlich zu den relativ statischen Formen und Ansichten kann eine Anzeige, ein GEM oder ein Gadget auch dynamisches Verhalten umfassen um dem Bediener zu ermöglichen, die Prozessdaten zu sehen, durch die Prozessdaten zu navigieren oder die Prozessdaten zu ändern. Dynamisches Verhalten kann z. B. Animationen und Ereignishandhabung umfassen, die beide nachfolgend abgehandelt werden.

[0216] Animationskarten verarbeiten Daten von Informationsquellen innerhalb der Prozessanlage zu den unterschiedlichen Formen auf der Anzeige, um die Prozessdaten in Form von Text oder Grafiken darzustellen. Das Animationsergebnis kann auf ein Feld einer Form (z. B. auf ein Textfeld oder auf ein Sichtbarkeitsfeld) angewandt werden, sodass sich die Animation ändert, wenn sich ein Feldwert einer Informationsquelle ändert. Demnach lässt es Animation, dass Prozessdaten mit Funktionen (z. B. mathematische Funktionen) manipuliert werden können, um die Rohprozessdaten in eine aussagekräftige visuelle Darstellung umzuwandeln. Beispielsweise kann eine Animation in eine Rechteckform umgewandelt werden, um den Füllungsprozentsatz eines Rechtecks für ein Tankniveau darzustellen, oder die Animation kann auf eine Textform angewandt werden, um den Füllungsprozentsatzwert in Textform anzuzeigen.

[0217] Ereignishandler sind konfiguriert, um eine definierte Reaktion auf Ereignisse wie z. B. einen Mausklick, eine Tastaturbetätigung und eine Bewegung eines Tastaturkürzels auszuführen. Bei der definierten Reaktion kann es sich z. B. um die Navigation zu einer anderen Anzeige, Bestätigung von Alarmen, Änderung eines Prozesseinstellungspunkts, Eingabe von Daten und andere Aktionen dieser Art handeln.

[0218] Da ein GEM in mehreren visuellen Darstellungen oder Ansichten enthalten sein kann, können die mit dem GEM zusammenhängenden Animationen und Ereignishandler einmal erstellt und in unterschiedlichen Ansichten wiederverwendet werden, um unterschiedliche Ausrichtungen oder Darstellungen der Daten bereitzustellen. Als solcher kann der

Betrag eines benutzerdefinierten Codes, der für die Erstellung des GEM und zur Modifikation der Verwendung des GEM für jede von dessen unterschiedlichen Ansichten benötigt wird, reduziert werden.

[0219] Um das GEM über viele Objekte und Informationsquellen wiederverwendbar zu gestalten, sind die Animationen und Ereignishandler-Pfade in einer Ausführungsform teilweise oder vollständig mit Platzhaltern tokenisiert. Ein Wert eines Platzhalterwerts kann bereitgestellt werden, wenn spezifische Verwendungen der GEMs konfiguriert werden, oder der Platzhalterwert kann bei Laufzeit programmatisch gesetzt werden. Animationen und Ereignishandler können auf einen Platzhalter verweisen, sodass der Wert des Platzhalters in der Referenz bei Laufzeit ersetzt wird, um einen Pfad zu dem spezifischen Objekt aufzulösen. Außerdem können Anzeigen ähnlich wie GEMs auch Platzhalter unterstützen, sodass eine Anzeige wiederverwendbar sein kann. Anzeigeeinstanzen können als Teil der Konfiguration erstellt werden, oder dynamisch in Laufzeit durch Bereitstellung von Werten für den/die unterschiedlichen Platzhalter erstellt werden.

[0220] In einigen Ausführungsformen werden Wiederverwendung und gemeinsame Nutzung von Informationen durch Verwendung globaler Sätze beeinflusst. Globale Sätze ermöglichen es einem Konfigurationsdesigner, Zustände und Verhalten zu konfigurieren, die über viele nicht miteinander in Beziehung stehende Anzeigen und GEMs gemeinsam genutzt werden, z. B. für die Lebensdauer einer Human Machine Interface(HMI)-Client-Sitzung. Elemente in einem globalen Satz können referenziert werden und können durch beliebige andere Anzeigen oder GEMs verwendet werden, womit sie die Wiederverwendung und gemeinsame Nutzung unter vielen Elementen zulassen. In einer Ausführungsform ist jedes Element in einem globalen Satz in einem Objekt gespeichert, welches mit einem Objekt verbunden ist, welches dem globalen Satz entspricht.

[0221] In einigen Ausführungsformen werden Wiederverwendung und gemeinsame Nutzung von Informationen durch Verwendung von Formatvorlagen und/oder Formatvorlagensätzen beeinflusst. Eine Formatvorlage ist eine Sammlung von Feldnamen und Wertpaaren und ein Formatvorlagensatz ist eine Sammlung von Formatvorlagen. In einer Ausführungsform unterstützen alle Formen eine Formatvorlage. Wenn eine Form einer Ansicht einer Anzeige oder GEM durch einen Benutzer hinzugefügt wird, kann das Formatvorlagenfeld auf eine Formatvorlage eines Formatvorlagensatzes eingestellt werden. Folglich verwendet die hinzugefügte Form die Werte, die in jeder Formatvorlage für jedes Feld definiert sind. Formatvorlagen und Formatvorlagensätze können in dem Konfigurationssystem durch jeweilige Objekte dargestellt werden.

[0222] In einigen Ausführungsformen ermöglichen benannte Konstanten Benutzern Flexibilität bei der Erstellung von Namen-Wertpaaren. Beispielsweise werden Felder innerhalb eines Grafikelementobjekts einer benannten Konstante als Wert zugewiesen, wie z. B. TitleFont oder TankBodyColor. Namenskonstanten ermöglichen die Erstellung von Projektstandards und werden Feldwerten zugewiesen. Wenn die Projektstandards geändert werden, dann kann der benannte Konstantenwert an einer zentralen Stelle aktualisiert werden. Eine benannte Konstante kann z. B. einen Namen, Titel, Beschreibung und Datentyp enthalten. Jeder beliebige Feldwert kann auf einen Wert oder auf eine benannte Konstante gesetzt werden, wie z. B. Felder auf Formeinsätzen, Animationen, Konvertern, Tabellen, globalen Sätzen und Formatvorlagen.

Muster:

[0223] Anzeigenobjekte können sich durch eines oder mehrere Muster wie z. B. Einsatzmuster oder ein abgeleitetes Muster aufeinander und auf andere Objekte beziehen. Zusätzlich oder alternativ können sich Konfigurationsobjekte durch eines oder mehrere Muster aufeinander und auf andere Objekte beziehen. Die Verwendung unterschiedlicher Typen von Mustern muss nicht gegenseitig ausschließlich sein. Beispielsweise können verschiedene Aspekte mehrerer Muster kombiniert werden.

[0224] Ein Einsatzmuster ist im Allgemeinen ein Einschluss oder eine Gruppierung von Objekten basierend auf Definitionen. **Fig. 13** veranschaulicht ein Diagramm eines Beispiels eines Definitionsverwendungs-Designmusters. Bei einem Verwendungsmuster umfasst eine Definition eines Grafikobjekts **1062a** eine oder mehrere Verwendungen **1062b** eines oder mehrerer anderer Grafikelemente **1062c**. Wenn sie instantiiert werden, stellen diese Verwendungen **1062b** eine Grafikform dar, die von dem Grafikobjekt **1062a** definiert wird, welches auf einer Ansicht einer Anzeige dargestellt wird. Jede Verwendung **1062b** kann eine Definition aufweisen, die Vorgabewerte für interne Elemente wie z. B. Felder, Ereignisse und Verhalten umfasst. Solche internen Elemente können in einer Ausführungsform konfiguriert sein.

[0225] Das Definitionsverwendungsmuster kann aus mehreren verschachtelten Ebenen bestehen, wenn gewünscht. **Fig. 14** veranschaulicht ein Klassendiagramm für ein beispielhaftes verschachteltes Definitionsverwendungs-Designmuster. Beispielsweise kann eine Anzeige ein GEM enthalten, welches ein weiteres GEM oder eine Form enthalten kann.

[0226] **Fig. 15** veranschaulicht ein Beispiel einer abstrahierten verschachtelten Definitionsverwendung. In **Fig. 15** basiert die Verwendung C1 (Bezugszei-

chen **1065a**) auf der Definition der Verwendung B1 (Bezugszeichen **1065b**). Ein Benutzer kann eines oder mehrere Elemente von einer anderen Verwendung modifizieren (z. B. außer Kraft setzen oder optimieren). Beispielsweise ist in **Fig. 15** eine Änderung an Verwendung C1 (Bezugszeichen **1065c**) eine Optimierung.

[0227] Ein anderes Muster, wodurch Objekte verwandt sein können, ist ein abgeleitetes Muster. Ein Diagramm eines beispielhaften abgeleiteten Musters ist dargestellt in **Fig. 16**. Das abgeleitete Muster ermöglicht die Erweiterung einer Definition ohne Verwendung von Aggregation (wie im Falle des oben abgehandelten Definitionsverwendungsmusters). Das abgeleitete Muster ermöglicht insbesondere die Erstellung einer neuen Definition **1068a** ohne Erstellung einer anderen Ebene in der Hierarchie. Als solche sind alle internen Elemente in der „basierend auf“-Definition **1068a** in der abgeleiteten Definition **1068b** enthalten.

[0228] Das abgeleitete Muster kann in mehreren Ebenen verschachtelt sein, z. B. kann eine abgeleitete Definition auf einer anderen abgeleiteten Definition basieren, die schlussendlich auf einer basierend auf einer nicht abgeleiteten Definition basieren kann. Jede beliebige Anzahl verschachtelter Ebenen kann möglich sein. **Fig. 17** veranschaulicht ein Beispiel eines verschachtelten abgeleiteten Musters, bei dem eine neue Definition **1070a** von einer ersten abgeleiteten Definition **1070b** erstellt wird, die aus einer zweiten abgeleiteten Definition **1070c** erstellt wurde, die aus einer nicht abgeleiteten Definition (nicht dargestellt) erstellt wurde.

Flexible Elemente und Objekte:

[0229] Wie zuvor abgehandelt, können Objekte flexibel definiert und in dem Configurationssystem verwendet werden. In einer Ausführungsform kann ein flexibles Objekt eines oder mehrere interne Elemente umfassen, die wiederum von einem anderen Objekt definiert werden. Es folgt eine Beschreibung flexibler Elemente, Objekte und interner Elemente, die in dem Configurationssystem enthalten sein können. Obwohl diese Beschreibungen im Kontext von Grafik- oder Anzeigeobjekten dargestellt werden, sind die in diesem Dokument abgehandelten Konzepte und Attribute auch auf Konfigurationsobjekte anwendbar.

[0230] Felder sind interne Elemente, die Zustände oder Attribute eines gegebenen Objekts darstellen. Beispielsweise ist eine Rechteckgröße in einem Breitenfeld und in einem Höhenfeld gespeichert. Jedes Feld kann einen Namen und Wert aufweisen. In einer Ausführungsform geben mit einem Feld zusammenhängende zusätzliche Metadaten andere Attribute des Feldes an, wie z. B., ob das Feld animiert sein kann oder nicht, ob das Feld Nur-Lesen ist, sowie ei-

nen Datentypen des Felds. Die Metadaten können z. B. bei der Konfiguration des gegebenen Objekts verwendet werden. Ein Wert eines Felds kann ein einfacher Datentyp wie z. B. eine Zeichenfolge sein oder kann ein Verweis auf ein anderes Objekt sein, wie z. B. ein Bibliotheksobjekt, ein Formobjekt oder ein Prozesselementobjekt. In einer Ausführungsform kann ein Wert eines Feldes von einem Benutzer modifiziert oder außer Kraft gesetzt werden.

[0231] Platzhalter sind verallgemeinerte interne Elemente, welche die Verwendung definierter Elemente über mehrere andere Objekte zulassen. Beispielsweise kann für Anzeigenobjekte ein Platzhalter ein Grafikelement sein, welches die Verallgemeinerung und Wiederverwendung von Animationen und Ereignishandlern über unterschiedliche Grafik- oder Anzeigenobjekte und Informationsquellenpfade ermöglicht. Als solche stellen Platzhalter eine Ebene eines Umwegs bereit und werden mit einem Wert bei Laufzeit aufgelöst. In einigen Ausführungsformen können Platzhalter verschachtelt sein.

[0232] Parameter sind interne Elemente, die Zustände oder Attribute eines Feldes darstellen. Beispielsweise kann ein Parameter für Anzeigen- und Grafikobjekte ein Grafikelement mit einem Wert und einem Status sein. Parameter können zu Grafikelementen wie z. B. GEMs, Gadgets, Anzeigen und Dashboards, Layouts und Formen, globalen Sätzen usw. hinzugefügt werden. Parameter sind typischerweise während der Laufzeit nicht sichtbar. Parameter können jedoch einen sichtbaren Aspekt eines Objekts während der Laufzeit (wie z. B. eine Animation) durch Erstellung von Ausdrücken beeinflussen, die auf den Wert verweisen. Ein Benutzer kann Parameter zu Objekten hinzufügen, doch in einigen Objekten wird einem Benutzer möglicherweise nicht erlaubt, Parameter zu bestimmten Bibliothekselementen oder primitiven Objekten hinzuzufügen.

[0233] Funktionen sind interne Elemente, die Verhalten für ein spezifisches Objekt bereitstellen. Beispielsweise kann bei Anzeigen- und Grafikobjekten eine Funktion ein Grafikelement oder ein allgemeines Verhalten sein, welches von einer Gruppe oder einem globalen Satz bereitgestellt wird. Beispielsweise kann ein visuelles Gruppenelement eine AddShape()-Funktion bereitstellen oder ein globaler Satz kann eine OpenDisplay()-Funktion bereitstellen. Einige Funktionen können integriert sein und sind daher möglicherweise nicht durch einen Benutzer modifizierbar. Es können weitere Funktionen vom Benutzer erstellt werden. In einer Ausführungsform können Funktionen während der Laufzeit über ein Programmierungs-API oder ein Skript ausgeführt werden. Außerdem können Funktionen eines oder mehrere Argumente annehmen, z. B. eine Konstante, einen Wert eines Feldes, einen Verweis auf ein anderes Objekt oder einen Verweis auf einen Parameter. Eine Funk-

tion kann einen Wert ergeben, z. B. eine Konstante, einen Wert eines Feldes, einen Verweis auf ein anderes Objekt oder in einigen Fällen einen Verweis auf einen Parameter.

[0234] Animationen sind interne Elemente, die Daten von einer Informationsquelle oder einem Objekt auf einem Feld eines internen Elements abbilden. Beispielsweise kann eine Animation Daten auf das FillPercent-Feld eines Rechtecks oder auf die Text-Eigenschaft eines Etiketts abbilden. Eine Animation umfasst einen Ausdruck, der erneut bewertet und/oder neu berechnet wird, wenn sich ein referenzierter Wert des Ausdrucks ändert. Außerdem steht eine Animation mit einem Zielfeld in Zusammenhang, z. B. dem Feld auf dem übergeordneten Objekt, welches mit den Ergebnissen des neu berechneten Ausdrucks zu aktualisieren ist. Als solches wird das Zielfeld mit dem neu berechneten Wert aktualisiert, wenn sich der berechnete Wert der Animation ändert, wodurch eine Änderung der Bildschirmdarstellung verursacht wird. In einer Ausführungsform umfasst eine Animation eine Aktualisierungsfrequenz und kann einen Konverter zur Umwandlung von Ausdrucksergebnissen umfassen.

[0235] Konverter sind interne Elemente, die das Ausdrucksergebnis einer Animation in ein endgültiges Animationsergebnis umwandeln. Beispielsweise kann die Umwandlung den neu berechneten Wert (Ergebnis) des Ausdrucks, den Datentyp oder -format des Ergebnisses, das Ergebnis und den Datentyp ändern, oder die Umwandlung kann einen anderen Typ von Konvertierung des Ergebnisses des Ausdrucks ausführen. Jeder Konvertertyp kann spezifische Felder umfassen, die von einem Benutzer konfigurierbar sind.

[0236] Ereignisse sind interne Elemente eines Objekts, die eine Aktion und eine Reaktion definieren. Die Definition eines Ereignisses umfasst typischerweise einen Ereignisauslöser wie z. B. den Klick eines Benutzers, eine Mausbewegung, Ansteuerung eines bestimmten Punkts mit der Maus oder eine Datenänderung. Optional werden Ereignisse auch mit einem Ereignishandler definiert, der eine oder mehrere Reaktionsfunktionen oder Verhalten ausführt, wenn der Auslöser feuert. Der Ereignishandler kann von dem Benutzer konfiguriert sein, kann als ein OOB-Verhalten auf vorkonfigurierten Definitionen bereitgestellt sein oder kann als integriertes Verhalten auf spezifischen Formen integriert sein. Das Hinzufügen von Ereignishandlern für spezifische Ereignisse ermöglicht dem Kunden, Verhalten basierend auf spezifischen Auslösern zu erzeugen.

[0237] Jeder Ereignisauslöser kann eines oder mehrere spezifische Argumente unterstützen. Im Allgemeinen umfasst ein Ereignisauslöser ein Senderargument, welches das Objekt angibt, das den Ereignis-

auslöser gefeuert hat. Zusätzlich kann der Ereignisauslöser optionale Argumente umfassen, die für den Typ von Ereignis angemessen sind, wie z. B. Informationen, die nach dem Feuern zu senden sind. Beispielsweise kann ein Mausklickauslöser nur das Senderargument aufweisen, während ein Mausbewegungsauslöser ein Senderargument und Mausstandortkoordinaten (z. B. X, Y) aufweisen kann, die zu senden sind, wenn der Mausbewegungsauslöser gefeuert wird. In einigen Fällen werden Ereignisse nicht basierend auf einer Benutzeraktion gefeuert, sondern werden gefeuert, wenn sich der Feldwert oder Systemstatus eines Objekts ändert, z. B. von Initialisierung in stabilen Zustand.

[0238] In einigen Ausführungsformen ist ein Ereignishandler für einen Ereignisauslöser konfiguriert. Bei diesen Ausführungsformen wird der Ereignishandler aufgerufen, wenn der Ereignisauslöser feuert, und führt eine oder mehrere Funktionen als Reaktion auf das Feuern des Auslösers aus. Bei einer Ausführungsform definiert ein Benutzer einen Ereignishandler dahingehend, dass er benutzerdefiniertes Verhalten oder Logik zusammen mit verfügbaren Funktionen ausführt (z. B. für die Anzeige, das Fenster oder einen globalen Satz verfügbare Funktionen).

[0239] Tabellen sind interne Elemente, die eine oder mehrere Datenstrukturen mit Eingabewerten und Ausgabeergebnissen umfassen. In einer Ausführungsform kann eine Tabelle mit einem Tabellenkonverter zur Umwandlung des Eingabewerts von einem Datentypen in einen anderen verwendet werden. Beispielsweise kann die Tabellenverwendung für ein Grafikobjekt die Konvertierung eines Prozesswerts in eine Farbe oder die Konvertierung eines Prozesswerts in eine Zeichenfolge umfassen. Als solche stellt eine Tabelle eine nichtprogrammatische Möglichkeit bereit, um Ergebnisse für einen Bereich von Eingabewerten zu konfigurieren. Tabellen können mit einer genauen Übereinstimmung eines Eingabewerts oder mit einem Bereich von Eingabewerten arbeiten.

[0240] Formobjekte sind grundlegende verallgemeinerte, visuelle Bausteine, die miteinander kombinierbar sind, um visuelle Darstellungen komplexer und spezialisierter visueller Elemente zu erzeugen, wie z. B. GEMs und Anzeigen. Einige Formobjekte sind integriert (weisen z. B. ausführbare installierte OOBs mit dem Konfigurationssystem auf) und werden in diesem Dokument als primitive Formen bezeichnet. Primitive Formen können zu einer von mehreren Kategorien wie z. B. geometrischen Formen (z. B. Rechteck, Linie, Ellipse) gehören, Behälter, in denen andere Formen angeordnet werden können (z. B. Gruppe, Panels, Gitter) oder Steuerungen (z. B. Text-Box, Dropdown-Kombinationsfeld oder Stapellistensteuerung, Trends).

[0241] Jede Form (ob primitiv oder nicht), verfügt über eine Definition. Die Definition kann z. B. eines oder mehrere Felder, Animationen, Ereignisse mit Ereignisauslösern, Händler und/oder Verbindungspunkte umfassen. In einer Ausführungsform kann eine Form ein Formatvorlagenfeld unterstützen. Typischerweise sind die meisten Formen konfigurierbar und modifizierbar. Beispielsweise wird ein Rechteck als ein integriertes Element des Konfigurationssystems bereitgestellt und ein Benutzer kann das Rechteckobjekt durch Setzen von Feldwerten und Hinzufügen von Ereignissen oder Animationen zur Verwendung beim Rechteck modifizieren. Der Benutzer kann dann die modifizierte Rechteckform als ein neues Objekt speichern und nach Genehmigung veröffentlichen.

[0242] Fig. 18 veranschaulicht ein Beispiel eines Formverwendungs-Rect 1 (Bezugszeichen **1075**), welches basierend auf der Definition **1078** eines Rechtecks erstellt ist. Der Benutzer hat die Werte der Breiten- **1080a** und Höhen-Felder **1080b** von den Standardwerten der Definition des Rechtecks modifiziert. Außerdem hat der Benutzer eine Animation **1082a** hinzugefügt, die auf das Breiten-Feld abzielt und hat einen Bereichskonverter **1082b** hinzugefügt, um den Wert entsprechend zu skalieren. Der Benutzer hat auch einen Ereignishandler **1085a** bei der Verwendung **1075** für das Klickereignis **1085b** hinzugefügt und der Ereignishandler wurde konfiguriert, um die Funktion `OpenDisplay()` aufzurufen. Bei einer Ausführungsform bleiben nur die Felder bestehen, die spezifisch außer Kraft gesetzt werden (z. B. **1080a**, **1080b**). Das heißt, dass die Formverwendung **1075** mit der Definition **1078** rückverbunden ist und jedes Feld, dessen Wert nicht außer Kraft gesetzt wurde, von der Definition **1078** (oder in einigen Fällen von der Formatvorlage) kommt. Wenn z. B. der Vordergrundpinsel der Rechteckdefinition **1078** von blau in grau geändert wird, werden alle Verwendungen, welche den Vordergrundpinsel (z. B. Verwendung `Rect1` **1075**) nicht außer Kraft setzen, einen blauen Vordergrundpinsel aufweisen, nachdem die Änderung von der Ausgabe der Rechteckdefinition **1078** weitergegeben wurde.

[0243] In Bezug auf Geometrieformdefinitionen ermöglicht eine Gruppenform, dass mehrere Formen als eine Form für Aktionen wie z. B. Sichtbarkeit, Größenänderung, Platzierung/Ausrichtung und Ereignisse behandelt werden. Als solche sind Gruppenformen Kopien von mehreren Formen, die ihre jeweiligen (X, Y) Koordinatenausrichtungen relativ beibehalten. Das Verhalten der Felder und Ereignisse einer Gruppe, und wie sie die Felder und Ereignisse enthaltener Formen innerhalb der Gruppe beeinflussen, kann konfigurierbar sein. Beispielsweise ist ein Dynamo ein Typ einer Gruppenform.

[0244] Eine Panel-Form kann einer Gruppenform ähnlich sein jedoch eine spezifischere Steuerung der Wiedergabe enthaltener Formen zulassen (z. B. Größe, Position, Anordnung usw.). Elemente innerhalb eines Panels sind allgemein angeordnet oder an einen spezifischen Bereich des Panels angedockt, und sie weisen im Gegensatz zu Gruppen keine relative Koordinateneinstellung auf. Panels versetzen den Benutzer in die Lage zur Steuerung von Elementen, wenn die genaue Anzahl von Elementen unbekannt ist, und in die Lage zur visuellen Anordnung von Elementen, ohne jedes Element einzeln bewegen zu müssen, wenn das Panel oder das enthaltene Fenster in der Größe verändert wird. Panels stellen auch Verhalten bereit, wenn Elemente innerhalb des Panels hinzugefügt, entfernt oder verborgen werden, durch Anpassung der Position anderer Elemente, um Leeräume zu entfernen und Überlappung zu vermeiden.

[0245] Eine Verbinderverwendung stellt ein physikalisches Verbindungselement in der Prozessanlage dar, wie z. B. ein Kabel, einen Kanal oder einen Förderer. Während der Konfiguration bezeichnet der Verbindungspunkt eine Angabe einer Verbindung der Verbindungsverwendung mit einer Formverwendung. Die Verbinderverwendung ermöglicht dem Verbinder die Nachverfolgung, Neuverlegung und Größenänderung, wenn die Formverwendung während der Konfiguration bewegt wird. Der Verbinder folgt und routet entsprechend seiner definierten Algorithmen und Einstellungen. Als solche sind Verbinder Formen, die andere Anzeigenobjekte auf Anzeigen miteinander verbinden, und werden durch jeweilige Eigenschaften und Verhalten definiert. Bei Platzierung auf einer Anzeige, und wenn er angeschlossen ist, kann ein Verbinder ein Routing um unterschiedliche Formverwendungen vornehmen. Ein Konfigurationsanwender kann die Platzierung oder das Routing des Verbinders modifizieren, Standard-Visuals definieren und/oder Verbindungspunkte einer Verbindungsverwendung modifizieren.

[0246] Ebenen eines anderen Typs von Anzeigenobjekten können in der Konfigurationszeit verwendet werden, um die Editierungserfahrung eines Benutzers durch Verbergen und Arretieren von Formen zu verbessern, die mit einer aktuellen Aufgabe kollidieren können, oder die eine unbeabsichtigte Modifikation von Formverwendungen verhindern können. Schichten bieten eine Möglichkeit zur Organisation der Formverwendungen, um die Konfiguration einer Anzeigendefinition und/oder einer GEM-Anzeigendefinition zu unterstützen. Schichten können auch bei der Organisation von Formen verwendet werden, um Laufzeitmerkmale wie z. B. Zeigen und Verbergen von Details wie einer Hilfe-, Wartungs- oder Diagnoseschicht bereitzustellen.

[0247] Insbesondere können Ansichten eine oder mehrere Schichten aufweisen. Formen innerhalb einer Ansicht können mit einer bestimmten Schicht in Zusammenhang stehen (z. B. können Formobjekt und Schichtobjekt miteinander verbunden sein). Eine bestimmte Schicht kann nur als Konfiguration oder sowohl als Konfiguration als auch als Laufzeit bezeichnet werden. Eine Schicht kann sichtbar oder verborgen sein. Die Schicht kann gesperrt und entsperrt sein, um unbeabsichtigte Auswahl oder Modifikation zu verhindern. Wenn die Sichtbarkeit der Schicht geändert wird, kann ein Wert rekursiv zu allen Schichten weitergegeben werden, die vom Namen her zu den enthaltenen GEM-Verwendungen passen. Beispielsweise gilt die Sichtbarkeit in der „Hilfe“-Schicht für alle GEM-Verwendungen, die mit der Hilfe-Schicht in Zusammenhang stehen.

Vom Benutzer erstellte Elemente und Objekte:

[0248] Mindestens einige der oben abgehandelten flexiblen Objekte können von einem Benutzer erstellt werden, beispielsweise:

Anzeigen – z. B. ein visuelles Element, welches während der Laufzeit von Benutzern geöffnet werden kann. Eine Anzeige stellt typischerweise eine Ansicht des Betriebsprozesses dar oder stellt Details über einen spezifischen Bereich des Prozesses bereit.

[0249] GEMs und Gadgets – z. B. eine konfigurierte Form, die ein Prozesselement oder andere Entität innerhalb des Prozesses darstellt und auf Anzeigen oder anderen GEMs enthalten sein kann (z. B. platziert). Beispielsweise kann eine Vorrichtung, ein Steuerungsmodul, ein Dynamo, eine Anzeigenansicht oder ein anderes Prozesselement von einem GEM dargestellt werden. GEMs können so konfiguriert sein, dass sie wiederverwendbar sind, um unterschiedliche Instanzen desselben Typs von Ausrüstung oder eines Teils des Prozesses innerhalb der Anlage oder des Prozesses darzustellen. Als solches kann ein GEM einmal konfiguriert und getestet sein, doch viele Male in anderen GEMs und Anzeigen verwendet werden. Typischerweise umfasst ein GEM einen Namen, einen Titel, eine Beschreibung, eine ausgewählte Ansicht und andere Felder wie z. B. Tabellen, Ereignisse, Parameter und Funktionen. GEMs können z. B. in anderen GEMs oder innerhalb von Anzeigen verschachtelt sein. Typischerweise weist ein GEM eine oder mehrere Definitionen auf, die eine oder mehrere entsprechende Darstellungen des GEMs bereitstellen. Eine Standardansicht kann für ein GEM bereitgestellt werden, und in einigen Ausführungsformen kann der Benutzer in der Lage sein, die Standardansicht zu ändern. In der Tat kann der Benutzer in der Lage sein, jedes der von einem GEM definierten Felder zu ändern.

[0250] In einer Ausführungsform ist ein Gadget ein GEM, welches zur Laufzeit verfügbar ist und auf Da-

shboard-Anzeigen platziert werden kann. Ein Bediener kann in der Lage sein, Gadgets z. B. während der Laufzeit zu erstellen und zu modifizieren.

[0251] Eine GEM-Definition kann von einer anderen GEM-Definition abgeleitet werden. Wenn ein GEM von einem anderen GEM abgeleitet wird, dann umfasst das GEM alle Ansichten von dem basierend-auf-GEM. Ein Benutzer kann diese Ansichten des abgeleiteten GEMs modifizieren oder zusätzliche Anzeigen dazu hinzufügen. Die zusätzlichen Ansichten können unabhängig sein, können von einer vorhandenen Ansicht abgeleitet sein, die in der vorliegenden GEM-Definition definiert ist, oder können basierend auf der GEM-Definition erstellt sein.

[0252] Bei einer Ausführungsform werden dann, wenn GEMs in einer Anzeigenansicht oder der Anzeigenansicht eines anderen GEMs platziert werden, aus den platzierten GEMs Formverwendungen (ähnlich wie das Hinzufügen einer integrierten Formverwendung wie z. B. eines Rechtecks oder einer Gruppe). Die platzierten GEMs bleiben jedoch mit ihrer jeweiligen Definition rückverbunden und Änderungen an den Verwendungen werden basierend auf den Weitergabeeinstellungen und -strategien weitergegeben.

[0253] In einer Ausführungsform ist die Modifikation einer GEM-Verwendung (z. B. Größenveränderung) ähnlich wie die Modifikation einer Gruppenverwendung. Außerdem kann ein GEM oder Gadget unter Verwendung einer standardisierten oder teilweise standardisierten Form konfiguriert werden, die einem Konfigurationsdesigner präsentiert wird.

[0254] Ansichten – z. B. eine visuelle Darstellung innerhalb einer Anzeige oder eines GEMs. Jede Anzeige oder GEM kann eine oder mehrere definierte Ansichten aufweisen. Visuelle oder Grafikelemente (z. B. Formen und andere GEMs) können einer Ansichtsdefinition hinzugefügt werden. Eine Ansicht kann unabhängig sein oder kann von einer anderen Ansicht innerhalb derselben übergeordneten Definition oder übergeordneten Basis-Definition abgeleitet sein.

[0255] Unabhängige Ansichtsdefinitionen definieren typischerweise ihren gesamten Inhalt und Informationen wie z. B. Formen, statisch konfigurierte Feldwerte, Animationen und Ereignishandler. Als solche basieren unabhängige Ansichtsdefinitionen nicht auf irgendeiner anderen Ansichtsdefinition innerhalb der enthaltenden Anzeigen- oder GEM-Definition. Eine Anzeigen- oder GEM-Definition kann jedoch mehrere Ansichten aufweisen, die unabhängig, und nicht von einer gemeinsamen Ansicht abgeleitet sind.

[0256] Abgeleitete Ansichtsdefinitionen basieren andererseits typischerweise auf einer anderen Ansichtsdefinition innerhalb derselben Behälterdefiniti-

on. Wenn die Behälterdefinition abgeleitet ist, dann kann eine Ansichtsdefinition von einer Ansicht der basierend-auf-Definition der Behälterdefinition abgeleitet sein. Beispielsweise wenn Anzeige AnsichtA und AnsichtB definiert. Wenn AnzeigeB von AnzeigeA abgeleitet wird, dann wird AnzeigeB standardmäßig AnsichtA und AnsichtB haben, wird jedoch auch eine AnsichtC haben, wie sie von AnsichtB abgeleitet ist.

[0257] Eine Ansichtsdefinition kann einen Satz unbekannter Felder umfassen, die der Benutzer konfigurieren kann oder zu denen der Benutzer Animationen hinzufügen kann, die auf diese Felder abzielen. Die Ansichtsdefinition kann auch eines oder mehrere Ereignisse umfassen, die bekannte Auslöser sind, zu denen ein Benutzer Ereignishandler hinzufügen kann. Außerdem umfasst eine bestimmte Ansichtsdefinition Formverwendungen, z. B. alle Verwendung von der basierend-auf-Ansicht sowie alle beliebigen Formverwendungen, die dieser bestimmten Ansichtsdefinition hinzugefügt wurden. Ansichten können einen Namespace aufweisen, sodass Elemente innerhalb der Ansichtsdefinition einheitlich benannt sind.

[0258] Global Sätze – z. B. ein konfiguriertes Element, welches Zustand (z. B. Werte) und Verhalten (z. B. Funktionen) umfasst, kann von nicht verwandten Anzeigen, Gadgets und GEMs referenziert werden. Im Allgemeinen weisen globale Sätze keinerlei visuelle Darstellungen in der Laufzeit auf, sind jedoch während der Laufzeit verfügbar. Ein globaler Satz kann als Singleton markiert werden oder ein globaler Satz kann einer Instanz bzw. einer Gruppe von Objekten entsprechen. Als solche stellen globale Sätze eine Möglichkeit bereit, Informationen zwischen vielen Objekten gemeinsam zu nutzen und wiederzuverwenden.

[0259] Formatvorlagen und Formatsätze – z. B. eine Sammlung von Feldnamen und Werten (z. B. Feld-/Wertpaare), die sich an einer Stelle geändert haben können und über einen Satz von Formverwendungen oder andere visuelle Elemente angewandt worden sein können, um Konsistenz aufrechtzuerhalten. Ein Formatvorlagensatz ist eine Sammlung von Formatvorlagen und umfasst typischerweise einen Namen, Titel, Beschreibung und möglicherweise andere Felder. Bei einer Ausführungsform gelten die Werte für die Felder in der Formatvorlage dann, wenn eine Form einen Formatvorlagensatz aufweist, für die Form, wenn das Feld nicht außer Kraft gesetzt wurde und nicht animiert ist. Formatvorlagensätze können bearbeitbar und versionierbar sein.

[0260] Daher unterstützt das Konfigurationssystem wie vorstehend abgehandelt einen robusten Satz von visuellen oder Grafikelementen, wovon jedes durch ein jeweiliges Objekt definiert ist. Der Satz von visuellen oder Grafikelementen kann z. B. eine oder

mehrere Anzeigendefinitionen, Dashboards, Layouts, GEM-Definitionen, Gadget-Definitionen, globale Sätze, integrierte Formdefinitionen, Gruppendefinitionen, Ansichtsdefinitionen, Formverwendungen, Gruppenverwendungen, Dynamos, GEM-Verwendungen, Gadget-Verwendungen, Chromverwendungen, Bereichsverwendungen, Panel-Verwendungen, Verbindungspunkte, Verbinder, Animationen, Konverter, Ereignisauslöser, Ereignishandler, Platzhalter, Parameter, Funktionen, Tabellen, Namenssätze, Formatvorlagen und/oder Formatvorlagensätze umfassen. Mindestens ein Teil des Satzes von visuellen oder Grafikelementen kann von einem Benutzer erstellt oder modifiziert sein.

[0261] Weiterhin kann das Objekt, welches jedes visuelle oder Grafikelement definiert, z. B. eines oder mehrere Elemente wie z. B. Felder, Ereignisauslöser, Funktionen, Ansichtsdefinitionen, Ereignishandler, Animationen, Platzhalter, Parameter, Tabellen, Namenssätze, Konverter, Verwendungen integrierter Form(en), Verwendung(en) von GEM(s), Verwendungen von Gadget(s), Chromverwendungen, Bereichsverwendungen, Gruppenverwendungen, Dynamos, Panel-Verwendungen, Verbindungspunkte, Verbinder, globale Sätze, Formatvorlagensätze und/oder Formatvorlagen umfassen. Jedes der in dem Objekt des visuellen oder Grafikelements definierten internen Elemente kann in dem Konfigurationssystem integriert sein, von einem Benutzer hinzugefügt sein, durch eine Definition konfiguriert sein oder ein Verweis auf ein anderes Element sein.

Änderungen, Modifikationen und Optimierungen:

[0262] Wie zuvor abgehandelt, kann ein Benutzer den Inhalt einer Objektdefinition ändern, modifizieren oder optimieren. Das heißt, dass ein Benutzer eines oder mehrere interne Elemente eines Objekts ändern kann, während eine Verbindung zu dessen jeweiliger basierend-auf-Definition aufrechterhalten wird. Diese Änderungen, Modifikationen oder Optimierungen können z. B. eine Hinzufügung, eine Löschung und/oder eine Modifikation an den Inhalten (z. B. ein Feld oder ein anderes internes Element) des Objekts umfassen. Weiterhin erhalten diese Änderungen, wenn sie in Ausgaben veröffentlicht werden, im Allgemeinen Vorrang vor der basierend-auf-Definition des Objekts und können es zukünftigen Änderungen der modifizierten Definition ermöglichen angewandt zu werden, wenn neue Ausgaben weitergegeben werden. Wenn z. B. ein in der Definition enthaltenes Element (z. B. ein GEM) in einer Verwendung oder einer abgeleiteten Ansicht modifiziert und optimiert wird, dann erhält die Optimierung Vorrang. Optimierungen einer Verwendungs- oder abgeleiteten Definition gehen verloren, wenn das Element aus der basierend-auf-Definition gelöscht wird. Beispielsweise wird ein in einer basierend-auf-Definition enthaltenes Element (z. B. ein GEM) gelöscht, wurde jedoch zuvor in ei-

ner Verwendungs- oder abgeleiteten Definition optimiert. In diesem Beispiel geht die Optimierung verloren. Weiterhin wird dann, wenn ein Element, welches in einer basierend-auf-Definition (z. B. ein GEM) enthalten war, modifiziert wird, jedoch in einer Verwendungs- oder abgeleiteten Definition gelöscht wurde, die Änderung an der basierend-auf-Definition nicht an die Verwendungs- oder abgeleitete Definition weitergegeben.

[0263] In einer Ausführungsform können Optimierungen eines spezifischen untergeordneten Objekts oder Verwendung in die basierend-auf-Definition zurückgedrängt werden. In einem Beispiel wird die basierend-auf-Definition auf eine neue Ausgabe aktualisiert, welche die an dem untergeordneten Element vorgenommenen Optimierungen umfasst, wobei die Optimierungen von dem untergeordneten Element entfernt werden.

[0264] Fig. 19–Fig. 21 veranschaulichen Beispiele von Modifikationen oder Optimierungen, die an unterschiedlichen visuellen Elementen und Objekten vorgenommen worden sein könnten. Insbesondere veranschaulicht Fig. 19 ein Beispiel einer Außerkraftsetzung einer Feldwertoptimierung, die auf ein Verwendungsmuster angewandt wird. In Fig. 19 werden der Breiten-Feldwert **1090a** und der Höhen-Feldwert **1090b** außer Kraft gesetzt.

[0265] Fig. 20 veranschaulicht ein Beispiel einer Außerkraftsetzung einer Feldwertoptimierung, die auf ein abgeleitetes Muster angewandt wird. In Fig. 20 werden der Breiten-Feldwert **1092a** und der Höhen-Feldwert **1092b** außer Kraft gesetzt.

[0266] Fig. 21 veranschaulicht ein Beispiel von mehreren Modifikationen oder Optimierungen an einem Objekt in mehreren Ebenen. Insbesondere veranschaulicht Fig. 21, wie eine Optimierung, die auf eine niedrigere Definition angewandt wird, ein Standardwert für eine nächste Ebene (z. B. Bezugszeichen **1095a–1095c**) wird, wie ein Wert in mehreren Ebenen außer Kraft gesetzt werden kann (z. B. Bezugszeichen **1098a**, **1098b**), und wie eine äußerste Definition Vorrang erhält (z. B. Bezugszeichen **1100**).

[0267] Wie veranschaulicht in Fig. 19–Fig. 21 kann die Optimierung in einer Ausführungsform bei Verwendungsmustern und abgeleiteten Mustern ausgeführt werden. Weiterhin können Optimierungen oder Modifikationen an Konfigurationselementen oder -objekten, und/oder an visuellen oder Grafikelementen oder -objekten vorgenommen werden. Beispielsweise kann der Satz von visuellen oder Grafikelementen kann z. B. eine oder mehrere Anzeigendefinitionen, Dashboards, Layouts, GEM-Definitionen, Gadget-Definitionen, globale Sätze, integrierte Formdefinitionen, Gruppendifinitionen, Ansichtsdefinitionen, Formverwendungen, Gruppenverwendungen, Dyna-

mos, GEM-Verwendungen, Gadget-Verwendungen, Chromverwendungen, Bereichsverwendungen, Panel-Verwendungen, Verbindungspunkte, Verbinder, Animationen, Konverter, Ereignisauslöser, Ereignishandler, Platzhalter, Parameter, Funktionen, Tabellen, Namenssätze, Formatvorlagen und/oder Formatvorlagensätze umfassen. Weiterhin können Optimierungen an einem oder mehreren Objekten ausgeführt werden, die in anderen Objektdefinitionen enthalten sind, wie z. B. Felder, Ereignisauslöser, Funktionen, Ansichtsdefinitionen, Ereignishandler, Animationen, Platzhalter, Parameter, Tabellen, Namenssätze, Konverter, Verwendungen integrierter Form(en), Verwendungen von GEM(s), Verwendungen von Gadget(s), Chromverwendungen, Bereichsverwendungen, Gruppenverwendungen, Dynamos, Panel-Verwendungen, Verbindungspunkte, Verbinder, globale Sätze, Formatvorlagensätze und/oder Formatvorlagen.

[0268] Wenn implementiert, kann jedes der in diesem Dokument beschriebenen Verfahren und Techniken oder Teile davon durch die Ausführung von Software ausgeführt werden, die in nicht vorübergehenden, materiellen, computerlesbaren Speichermedien oder Speichern wie z. B. Magnetdatenträgern, Laser Disks, optischen Datenträgern Halbleiterspeichern, anderen Speichervorrichtungen oder anderen Speichermedien, in einem RAM oder ROM eines Computers oder Prozessors usw. gespeichert sind. In einer Ausführungsform kann jedes der in diesem Dokument beschriebenen Verfahren und Techniken durch einen Konfigurator oder einen flexiblen Konfigurator (z. B. eine oder mehrere Konfigurationsanwendungen oder andere geeignete Anwendungen) ausgeführt werden. Der Konfigurator kann kommunikativ mit einer Datenspeichervorrichtung und mit einem oder mehreren konfigurierbaren Elementen gekoppelt sein, die in der Prozessanlage oder in dem Prozesssteuerungssystem enthalten sind. Der Konfigurator kann computerausführbare Anweisungen umfassen, die in einem oder mehreren materiellen, nicht vorübergehenden Speichern oder computerlesbaren Speichermedien gespeichert sind, wobei die computerausführbaren Anweisungen ausführbar sein können, um jedes der in diesem Dokument beschriebenen Verfahren oder Techniken oder Teile davon auszuführen.

[0269] Obwohl der vorangehende Text eine detaillierte Beschreibung zahlreicher unterschiedlicher Ausführungsformen darstellt, sollte verstanden werden, dass der Umfang des Patents durch die Worte in den Ansprüchen definiert ist, die am Ende dieses Patents beschrieben sind, sowie durch ihre Entsprechungen. Die detaillierte Beschreibung soll nur beispielhaft sein und nicht jede mögliche Ausführungsform beschreiben, da die Beschreibung jeder möglichen Ausführungsform unpraktisch, wenn nicht sogar unmöglich wäre. Es könnten zahlreiche alternati-

ve Ausführungsformen entweder unter Verwendung aktueller Technologie oder von Technologie implementiert werden, die nach dem Anmeldedatum dieses Patents entwickelt wurde, die immer noch in den Umfang der Ansprüche und aller ihrer Entsprechungen fallen würde. Beispielhaft und nicht begrenzend werden im Rahmen der Offenbarung in diesem Dokument mindestens die folgenden Aspekte betrachtet: Verfahren zur Konfiguration einer Prozessanlage, welches Folgendes umfasst:

Empfang einer ersten Benutzereingabe, die eine Modifikation an einem aktuellen Prozesselementobjekt für ein spezifisches Element der Prozessanlage angibt, wobei das spezifische Element einem oder mehreren, in der Prozessanlage gesteuerten Prozessen entspricht;

Modifikation des aktuellen Prozesselementobjekts basierend auf der ersten Benutzereingabe zur Erzeugung eines modifizierten Prozesselementobjekts für das spezifische Element;

Verzögerung einer Instantiierung des modifizierten Prozesselementobjekts bis zu einer spezifizierten Zeit, die von einer zweiten Benutzereingabe angegeben ist;

Veranlassung der Instantiierung des modifizierten Prozesselementobjekts an dem spezifizierten Zeitpunkt, sodass das spezifische Element während der Laufzeit entsprechend dem modifizierten Prozesselementobjekt arbeitet, um Steuerungs- oder Anzeigefunktionen bereitzustellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören.

[0270] Verfahren nach dem vorangegangenen Aspekt, wobei:

der Empfang einer ersten Benutzereingabe, die eine Modifikation an dem aktuellen Prozesselementobjekt angibt, den Empfang einer Benutzereingabe umfasst, die eine Modifikation an einem aktuellen Instanzobjekt angibt, wovon ein Satz der aktuellen Prozesselementobjekte erstellt wird, wobei der Satz aktueller Prozesselementobjekte das aktuelle Prozesselementobjekt für das spezifische Element umfasst, und das von einem Klassenobjekt erstellte aktuelle Instanzobjekt; und

das Verfahren weiterhin die Modifikation des aktuellen Instanzobjekts basierend auf der Benutzereingabe umfasst, damit das aktuelle Instanzobjekt ein modifiziertes Instanzobjekt erzeugt, und Modifikation des Satzes aktueller Prozesselemente basierend auf dem modifizierten Instanzobjekt.

[0271] Verfahren nach einem der vorangegangenen Aspekte, wobei:

das aktuelle Instanzobjekt ein erstes aktuelles Instanzobjekt ist,

das modifizierte Instanzobjekt ein erstes modifiziertes Instanzobjekt ist, und

das Verfahren weiterhin die Modifikation des ersten aktuellen Instanzobjekts basierend auf der Benutzer-

eingabe umfasst, welche die Modifikation an demselben angibt, um ein zweites modifiziertes Instanzobjekt zu erzeugen, wobei das zweite aktuelle Instanzobjekt von dem Klassenobjekt erstellt ist.

[0272] Verfahren nach einem der vorangegangenen Aspekte, wobei das zweite aktuelle Instanzobjekt auf dem ersten aktuellen Instanzobjekt basiert.

[0273] Verfahren nach einem der vorangegangenen Aspekte, wobei:

die spezifizierte Zeit eine erste spezifizierte Zeit ist; der Satz aktueller Prozesssteuerungselementobjekte ein erster Satz aktueller Prozesssteuerungselemente ist; und

das Verfahren weiterhin Folgendes umfasst:

Veranlassung der Instantiierung des ersten Satzes aktueller Prozesssteuerungselementobjekte am ersten spezifizierten Zeitpunkt, und

Veranlassung der Instantiierung eines zweiten Satzes aktueller Prozesssteuerungselementobjekte entsprechend dem zweiten modifizierten Instanzobjekt an einem zweiten spezifizierten Zeitpunkt, der sich von dem ersten spezifizierten Zeitpunkt unterscheidet.

[0274] Verfahren nach einem der vorangegangenen Aspekte, wobei mindestens ein Punkt von:

dem ersten aktuellen Instanzobjekt von einem Benutzer angegeben wird, dem zweiten aktuellen Instanzobjekt von dem Benutzer angegeben wird, der erste spezifizierte Zeitpunkt von dem Benutzer angegeben wird, oder der zweite spezifizierte Zeitpunkt von dem Benutzer angegeben wird.

[0275] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Aufrechterhaltung eines anderen aktuellen Prozesselementobjekts einschließlich der Nicht-Modifikation des anderen aktuellen Prozesselementobjekts basierend auf der Benutzereingabe umfasst, welche die Modifikation des aktuellen Instanzobjekts angibt, wobei das andere aktuelle Prozesselementobjekt auf dem Klassenobjekt basiert und von dem Satz aktueller Prozesselementobjekte entsprechend dem aktuellen Instanzobjekt ausgeschlossen wird.

[0276] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Weitergabe der Modifikation des aktuellen Instanzobjekts an das Klassenobjekt umfasst.

[0277] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der ersten Benutzereingabe, die eine Modifikation an dem aktuellen Prozesselementobjekt angibt, den Empfang einer Benutzereingabe umfasst, die eine Modifikation an einem

Klassenobjekt angibt, auf dem das aktuelle Prozesselementobjekt basiert.

[0278] Verfahren nach einem der vorangegangenen Aspekte, wobei:
das aktuelle Prozesselementobjekt ein erstes aktuelles Prozesselementobjekt ist,
das modifizierte Prozesselementobjekt ein erstes modifiziertes Prozesselementobjekt ist, und
das Verfahren weiterhin die Modifikation eines zweiten aktuellen Prozesselementobjekts basierend auf der ersten Benutzereingabe umfasst, um ein zweites modifiziertes Prozesselementobjekt zu erzeugen, wobei das zweite aktuelle Prozesselementobjekt auf dem Klassenobjekt basiert.

[0279] Verfahren nach einem der vorangegangenen Aspekte, wobei der spezifizierte Zeitpunkt ein erster spezifizierter Zeitpunkt ist und das Verfahren weiterhin die Veranlassung einer Instantiierung des zweiten modifizierten Prozesselementobjekts an einem zweiten spezifizierten Zeitpunkt umfasst, der sich von dem ersten spezifizierten Zeitpunkt unterscheidet.

[0280] Verfahren nach einem der vorangegangenen Aspekte,
wobei mindestens eines der ersten aktuellen Prozesselementobjekte aus einem ersten Instanzobjekt erstellt wird oder das zweite aktuelle Prozesselementobjekt aus einem zweiten Instanzobjekt erstellt wird, und
wobei das erste Instanzobjekt und das zweite Instanzobjekt aus dem Klassenobjekt erstellt werden.

[0281] Verfahren nach einem der vorangegangenen Aspekte, wobei:
das aktuelle Prozesselementobjekt ein erstes aktuelles Prozesselementobjekt ist,
das modifizierte Prozesselementobjekt ein erstes modifiziertes Prozesselementobjekt ist, und
das Verfahren weiterhin die Nicht-Modifikation eines zweiten aktuellen Prozesselementobjekts basierend auf der ersten Benutzereingabe umfasst, wobei das zweite aktuelle Prozesselementobjekt auf dem Klassenobjekt basiert.

[0282] Verfahren nach einem der vorangegangenen Aspekte, wobei das zweite aktuelle Prozesselementobjekt von einem Benutzer angegeben wird.

[0283] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der ersten Benutzereingabe, welche die Modifikation des aktuellen Prozesselementobjekts angibt, mindestens einen der folgenden Punkte umfasst:
Hinzufügung zu einem Inhalt des aktuellen Prozesselementobjekts,
Löschung mindestens eines ersten Teils des Inhalts des aktuellen Prozesselementobjekts,

Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen Prozesselementobjekts oder mindestens eines zweiten Teils des Inhalts des aktuellen Prozesselementobjekts,
Aktivierung eines deaktivierten Inhalts des aktuellen Prozesselementobjekts,
Änderung eines in dem aktuellen Prozesselementobjekt enthaltenen Werts,
Änderung eines in dem aktuellen Prozesselementobjekt enthaltenen Verweises,
Umbenennung des aktuellen Prozesselementobjekts, oder
Auflösung eines in dem aktuellen Prozesselementobjekt enthaltenen Verweises.

[0284] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der ersten Benutzereingabe, die eine Modifikation an dem aktuellen Prozesselementobjekt angibt, den Empfang der ersten Benutzereingabe umfasst, welche die Modifikation an dem aktuellen Prozesselementobjekt für eines von einem Steuerungsmodul oder einem Anzeigenmodul angibt.

[0285] Verfahren nach einem der vorangegangenen Aspekte,
welches weiterhin Folgendes umfasst:
Speicherung des modifizierten Prozesselementobjekts als ein entwurfsmodifiziertes Prozesselementobjekt, und
Empfang einer Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten Prozesselementobjekts zur Erzeugung eines veröffentlichten modifizierten Prozesselementobjekts; und
wobei die Veranlassung der Instantiierung des modifizierten Prozesselementobjekts die Veranlassung der Instantiierung des veröffentlichten modifizierten Prozesselementobjekts umfasst.

[0286] Verfahren nach einem der vorangegangenen Aspekte, wobei das modifizierte Prozesselementobjekt ein erstes modifiziertes Prozesselementobjekt ist und das Verfahren weiterhin Folgendes umfasst:
Empfang einer dritten Benutzereingabe, die eine andere Modifikation an einem von dem aktuellen Prozesselementobjekt oder dem ersten modifizierten Prozesselementobjekt angibt, und
Modifikation des einen von dem aktuellen Prozesselementobjekt oder dem ersten modifizierten Prozesselementobjekt entsprechend der dritten Benutzereingabe zur Erzeugung eines zweiten modifizierten Prozesselementobjekts.

[0287] Verfahren nach einem der vorangegangenen Aspekte, wobei:
die Speicherung des modifizierten Prozesselementobjekts als ein entwurfsmodifiziertes Prozesselementobjekt die Speicherung des ersten modifizierten Prozesselementobjekts als ein erstes entwurfsmodifiziertes Prozesselementobjekt umfasst, und

das Verfahren weiterhin die Speicherung des zweiten modifizierten Prozesselementobjekts als ein zweites entwurfsmodifiziertes Prozesselementobjekt umfasst.

[0288] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten Prozesselementobjekts den Empfang der Benutzerangabe zur Veröffentlichung eines ausgewählten entwurfsmodifizierten Prozesselementobjekts umfasst.

[0289] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin Folgendes umfasst: Empfang einer Benutzeranforderung zum Vergleich von mindestens zwei der folgenden Punkte: des aktuellen Prozesselementobjekts, des entwurfsmodifizierten Prozesselementobjekts und des veröffentlichten modifizierten Prozesselementobjekts; und Anzeige eines Vergleichs entsprechend einer Benutzeranfrage an einer Benutzerschnittstelle.

[0290] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin Folgendes umfasst: Empfang einer Angabe einer Auswahl von einem in einem Entwurfspaket einzubindenden Satz von Objekten; Speicherung einer Angabe des Entwurfspakets; und Empfang einer Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung eines veröffentlichten Pakets; und Veranlassung der Instantiierung eines Satzes von Prozesselementobjekten entsprechend den in dem veröffentlichten Paket enthaltenen Objekten.

[0291] Verfahren nach einem der vorangegangenen Aspekte, wobei der Satz von Objekten mindestens eines von einem Klassenobjekt, einem Instanzobjekt, einem Bibliotheksobjekt oder einem Prozesselementobjekt umfasst.

[0292] Verfahren nach einem der vorangegangenen Aspekte, wobei:
das Entwurfspaket ein erstes Entwurfspaket ist;
das Verfahren weiterhin Folgendes umfasst:
Empfang einer dritten Benutzereingabe, die eine Modifikation an dem ersten Entwurfspaket angibt, und Modifikation des ersten Entwurfspakets entsprechend der dritten Benutzereingabe; und
Speicherung des modifizierten ersten Entwurfspakets als ein zweites Entwurfspaket; und
wobei der Empfang der Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung des veröffentlichten Pakets den Empfang einer Benutzerangabe zur Veröffentlichung eines ausgewählten Entwurfspakets umfasst, um das veröffentlichte Paket zu erzeugen.

[0293] Verfahren zur Konfiguration einer Prozessanlage, welches Folgendes umfasst:

Empfang einer Benutzereingabe, die eine Modifikation an einem aktuellen Klassenobjekt angibt, welches einem oder mehreren, in der Prozessanlage gesteuerten Prozessen entspricht;

Weitergabe der Modifikation an jedes aktuelle Objekt einer ersten Untergruppe eines Satzes aktueller Objekte, die aus dem Klassenobjekt erstellt sind, zur Erstellung einer ersten Untergruppe modifizierter Objekte;

Nicht-Weitergabe der Modifikation an eine zweite Untergruppe des Satzes aktueller Objekte, die aus dem Klassenobjekt erstellt sind;

Veranlassung der Instantiierung eines ersten Satzes modifizierter Prozesselementobjekte entsprechend der ersten Untergruppe modifizierter Objekte, sodass ein jeweiliger erster Satz von Prozesselementen in Laufzeit entsprechend dem ersten Satz modifizierter Prozesselementobjekte arbeitet, um Steuerungs- oder Anzeigefunktionen bereitzustellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören; und

Aufrechterhaltung einer Instantiierung eines zweiten Satzes aktueller Prozesselementobjekte entsprechend der zweiten Untergruppe aktueller Objekte, sodass ein jeweiliger zweiter Satz von Prozesselementen in Laufzeit entsprechend der zweiten Untergruppe aktueller Objekte arbeitet, um Steuerungs- oder Anzeigefunktionen bereitzustellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören.

[0294] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin den Empfang von mindestens einer von einer Angabe einer Benutzerauswahl der ersten Untergruppe aktueller Objekte oder einer Benutzerauswahl der zweiten Untergruppe aktueller Objekte umfasst.

[0295] Verfahren nach einem der vorangegangenen Aspekte, wobei mindestens eine von:
der ersten Untergruppe aktueller Objekte ein erstes aktuelles Instanzobjekt umfasst, welches aus dem aktuellen Klassenobjekt erstellt ist,
der zweiten Untergruppe aktueller Objekte ein zweites aktuelles Instanzobjekt umfasst, welches aus dem aktuellen Klassenobjekt erstellt ist,
der ersten Untergruppe aktueller Objekte ein erstes aktuelles Prozesselementobjekt umfasst, welches eine Definition aufweist, die auf dem aktuellen Klassenobjekt basiert, oder
der zweiten Untergruppe aktueller Objekte ein zweites aktuelles Prozesselementobjekt umfasst, welches eine Definition aufweist, die auf dem aktuellen Klassenobjekt basiert.

[0296] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Aufrechterhaltung einer Angabe der Modifikation umfasst.

[0297] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der ersten Benutzereingabe, welche die Modifikation des aktuellen Klassenobjekts angibt, mindestens einen der folgenden Punkte umfasst:

Hinzufügung zu einem Inhalt des aktuellen Klassenobjekts,
 Löschung mindestens eines ersten Teils des Inhalts des aktuellen Klassenobjekts,
 Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen Klassenobjekts oder mindestens eines zweiten Teils des Inhalts des aktuellen Klassenobjekts,
 Aktivierung eines deaktivierten Inhalts des aktuellen Klassenobjekts,
 Änderung eines in dem aktuellen Klassenobjekt enthaltenen Werts,
 Änderung eines in dem aktuellen Klassenobjekt enthaltenen Verweises,
 Umbenennung des aktuellen Klassenobjekts, oder
 Auflösung eines in dem aktuellen Klassenobjekt enthaltenen Verweises.

[0298] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Weitergabe der Modifikation an jedes aktuelle Objekt einer dritten Untergruppe des Satzes aktueller Objekte umfasst, die aus dem Klassenobjekt erstellt sind, zur Erstellung einer dritten Untergruppe modifizierter Objekte. In einer Ausführungsform kann jedes aktuelle Objekt, welches die weitergegebene Modifikation umfasst, umbenannt werden.

[0299] Verfahren nach einem der vorangegangenen Aspekte, wobei die Benutzereingabe eine erste Benutzereingabe ist, und wobei das Verfahren weiterhin die Verzögerung der Instantiierung des ersten Satzes modifizierter Prozesselementobjekte bis zu einem spezifizierten Zeitpunkt umfasst, der von einer zweiten Benutzereingabe spezifiziert ist.

[0300] Verfahren nach einem der vorangegangenen Aspekte, wobei der spezifizierte Zeitpunkt ein erster spezifizierter Zeitpunkt ist und wobei das Verfahren weiterhin die Verzögerung der Instantiierung eines dritten Satzes modifizierter Prozesselementobjekte entsprechend der dritten Untergruppe des Satzes aktueller Objekte bis zu einem zweiten spezifizierten Zeitpunkt umfasst, der sich von dem ersten spezifizierten Zeitpunkt unterscheidet.

[0301] Verfahren nach einem der vorangegangenen Aspekte, wobei der jeweilige erste Satz von Prozesselementen mindestens eines von einem Anzeigenmodul oder einem Steuerungsmodul umfasst.

[0302] Verfahren zur Konfiguration einer Prozessanlage, welches Folgendes umfasst:
 Empfang einer Benutzereingabe, die eine Modifikation an einem aktuellen Instanzobjekt angibt, welches

einem oder mehreren, in der Prozessanlage gesteuerten Prozessen entspricht;
 Modifikation des aktuellen Instanzobjekts basierend auf der Benutzereingabe zur Erzeugung eines modifizierten Instanzobjekts;
 Modifikation eines ersten Satzes aktueller Prozesselementobjekte, die von dem aktuellen Instanzobjekt erstellt wurden, zur Erzeugung eines ersten Satzes modifizierter Prozesselementobjekte, basierend auf dem modifizierten Instanzobjekt;
 Veranlassung der Instantiierung des ersten Satzes modifizierter Prozesselementobjekte, sodass ein jeweiliger erster Satz von Prozesselementen in Laufzeit entsprechend dem ersten Satz modifizierter Prozesselementobjekte arbeitet, um Steuerungs- oder Anzeigefunktionen bereitzustellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören;
 Weitergabe der Modifikation an einen Satz anderer Objekte zur Erzeugung eines Satzes modifizierter anderer Objekte, wobei der Satz anderer Objekte auf dem Klassenobjekt basiert; und
 Veranlassung der Instantiierung eines zweiten Satzes modifizierter Prozesselementobjekte entsprechend dem Satz modifizierter anderer Objekte, sodass ein jeweiliger zweiter Satz von Prozesselementen in Laufzeit entsprechend dem zweiten Satz modifizierter Prozesselementobjekte arbeitet, um Steuerungs- oder Anzeigefunktionen bereitzustellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören.

[0303] Verfahren nach einem der vorangegangenen Aspekte, wobei das aktuelle Instanzobjekt ein erstes aktuelles Instanzobjekt ist und wobei die Weitergabe der Modifikation an den Satz anderer Objekte die Weitergabe der Modifikation an ein zweites aktuelles Instanzobjekt umfasst, welches von dem Klassenobjekt erstellt ist.

[0304] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Nicht-Weitergabe der Modifikation an ein drittes aktuelles Instanzobjekt umfasst, welches von dem Klassenobjekt erstellt ist.

[0305] Verfahren nach einem der vorangegangenen Aspekte, wobei die Weitergabe der Modifikation an den Satz anderer Objekte die Weitergabe der Modifikation an das Klassenobjekt umfasst.

[0306] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Aufrechterhaltung einer Angabe der Modifikation umfasst.

[0307] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Nicht-Modifikation eines dritten Satzes aktueller Prozesselementobjekte basierend auf dem modifizierten Klassenobjekt umfasst, wobei der dritte Satz aktueller Prozesselement-

objekte auf dem Klassenobjekt aufgebaut und von einem Benutzer angegeben ist.

[0308] Verfahren nach einem der vorangegangenen Aspekte, wobei die Weitergabe der Modifikation an den Satz anderer Objekte die Weitergabe der Modifikation an ein anderes Objekt umfasst, welches basierend auf dem aktuellen Instanzobjekt erstellt ist.

[0309] Verfahren nach einem der vorangegangenen Aspekte, wobei das andere Objekt eines von einem anderen Instanzobjekt oder einem Prozesselementobjekt ist, welches von dem aktuellen Instanzobjekt erstellt ist.

[0310] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin den Empfang einer Angabe einer Benutzerauswahl des Satzes anderer Objekte umfasst.

[0311] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Nicht-Modifikation eines dritten Satzes aktueller Prozesselementobjekte basierend auf dem modifizierten Instanzobjekt umfasst, wobei der dritte Satz aktueller Prozesselementobjekte auf dem aktuellen Instanzobjekt basiert und von einem Benutzer angegeben ist.

[0312] Verfahren nach einem der vorangegangenen Aspekte, wobei die Benutzereingabe eine erste Benutzereingabe ist, und wobei das Verfahren weiterhin die Verzögerung der Instantiierung des ersten Satzes modifizierter Prozesselementobjekte bis zu einem spezifizierten Zeitpunkt umfasst, der von einer zweiten Benutzereingabe spezifiziert ist.

[0313] Verfahren nach einem der vorangegangenen Aspekte, wobei der spezifizierte Zeitpunkt ein erster spezifizierter Zeitpunkt ist und das Verfahren weiterhin die Verzögerung des zweiten Satzes modifizierter Prozesselementobjekte bis zu einem zweiten spezifizierten Zeitpunkt umfasst, der sich von dem ersten spezifizierten Zeitpunkt unterscheidet.

[0314] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der ersten Benutzereingabe, welche die Modifikation des aktuellen Instanzobjekts angibt, mindestens einen der folgenden Punkte umfasst:

Hinzufügung zu einem Inhalt des aktuellen Instanzobjekts,
 Löschung mindestens eines ersten Teils des Inhalts des aktuellen Instanzobjekts,
 Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen Instanzobjekts oder mindestens eines zweiten Teils des Inhalts des aktuellen Instanzobjekts,
 Aktivierung eines deaktivierten Inhalts des aktuellen Instanzobjekts,

Änderung eines in dem aktuellen Instanzobjekt enthaltenen Werts,

Änderung eines in dem aktuellen Instanzobjekt enthaltenen Verweises,

Umbenennung des aktuellen Instanzobjekts, oder

Auflösung eines in dem aktuellen Instanzobjekt enthaltenen Verweises.

[0315] Verfahren nach einem der vorangegangenen Aspekte, wobei der jeweilige erste Satz von Prozesselementen mindestens eines von einem Steuerungsmodul oder einem Anzeigenmodul umfasst.

[0316] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin die Umbenennung beliebiger modifizierter Objekte umfasst:

Verfahren zur Konfiguration einer Prozessanlage, welches Folgendes umfasst:

Empfang einer Benutzereingabe, die eine Modifikation an einem aktuellen Prozesselementobjekt für ein spezifisches Element der Prozessanlage angibt, wobei das spezifische Element einem oder mehreren, in der Prozessanlage gesteuerten Prozessen entspricht;

Modifikation des aktuellen Prozesselementobjekts basierend auf der Benutzereingabe zur Erzeugung eines modifizierten Prozesselementobjekts für das spezifische Element;

Speicherung des modifizierten Prozesselementobjekts als ein entwurfsmodifiziertes Prozesselementobjekt;

Empfang einer Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten Prozesselementobjekts zur Erzeugung eines veröffentlichten modifizierten Prozesselementobjekts; und

Veranlassung einer Instantiierung des veröffentlichten modifizierten Prozesselementobjekts, sodass das spezifische Element während der Laufzeit entsprechend dem veröffentlichten modifizierten Prozesselementobjekt arbeitet, um Steuerungs- oder Anzeigefunktionen bereitzustellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören.

[0317] Verfahren nach einem der vorangegangenen Aspekte, wobei die Benutzereingabe eine erste Benutzereingabe ist und das modifizierte Prozesselementobjekt ein erstes modifiziertes Prozesselementobjekt ist, und das Verfahren weiterhin Folgendes umfasst:

Empfang einer zweiten Benutzereingabe, die eine andere Modifikation an einem von dem aktuellen Prozesselementobjekt oder dem ersten modifizierten Prozesselementobjekt angibt, und

Modifikation des einen von dem aktuellen Prozesselementobjekt oder dem ersten modifizierten Prozesselementobjekt entsprechend der zweiten Benutzereingabe zur Erzeugung eines zweiten modifizierten Prozesselementobjekts.

[0318] Verfahren nach einem der vorangegangenen Aspekte, wobei:
die Speicherung des modifizierten Prozesselementobjekts als ein entwurfsmodifiziertes Prozesselementobjekt die Speicherung des ersten modifizierten Prozesselementobjekts als ein erstes entwurfsmodifiziertes Prozesselementobjekt umfasst, und
das Verfahren weiterhin die Speicherung des zweiten modifizierten Prozesselementobjekts als ein zweites entwurfsmodifiziertes Prozesselementobjekt umfasst.

[0319] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten Prozesselementobjekts den Empfang der Benutzerangabe zur Veröffentlichung eines ausgewählten entwurfsmodifizierten Prozesselementobjekts umfasst.

[0320] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin Folgendes umfasst:
Empfang einer Benutzeranfrage zum Vergleich von mindestens zwei von: dem aktuellen Prozesselementobjekt, dem entwurfsmodifizierten Prozesselementobjekt und dem veröffentlichten modifizierten Prozesselementobjekt; und
Anzeige eines Vergleichs entsprechend der Benutzeranfrage an einer Benutzerschnittstelle.

[0321] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin Folgendes umfasst:
Speicherung einer Vielzahl entwurfsmodifizierter Objekte der Prozessanlage als Entwurfspaket, wobei die Vielzahl entwurfsmodifizierter Objekte das entwurfsmodifizierte Prozesselementobjekt umfasst;
Empfang einer Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung eines veröffentlichten Pakets;
Veröffentlichung mindestens der Untergruppe der Vielzahl entwurfsmodifizierter Objekte zur Erzeugung eines Satzes veröffentlichter modifizierter Objekte, die in dem veröffentlichten Paket enthalten sind; und
Veranlassung der Instantiierung des Satzes veröffentlichter modifizierter Prozesselementobjekte, so dass Prozesselementobjekte entsprechend dem Satz veröffentlichter modifizierter Objekte während der Laufzeit entsprechend dem Satz veröffentlichter modifizierter Objekte Steuerungs- oder Anzeigefunktionen bereitstellen, die zu dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen gehören.

[0322] Verfahren nach einem der vorangegangenen Aspekte, wobei die Speicherung der Vielzahl entwurfsmodifizierter Objekte die Speicherung der Vielzahl entwurfsmodifizierter Objekte einschließlich mindestens von einem von folgenden Punkten umfasst:
ein Entwurfsklassenobjekt oder ein Entwurfsinstanzobjekt, welches von dem Entwurfsklassenobjekt oder von einem anderen Klassenobjekt erstellt wurde.

[0323] Verfahren nach einem der vorangegangenen Aspekte, welches weiterhin den Empfang einer Angabe einer Auswahl der Vielzahl entwurfsmodifizierter Objekte umfasst.

[0324] Verfahren nach einem der vorangegangenen Aspekte, wobei:
das Entwurfspaket ein erstes Entwurfspaket ist;
das Verfahren weiterhin Folgendes umfasst:
Empfang einer weiteren Benutzereingabe, die eine Modifikation an dem ersten Entwurfspaket angibt, und
Modifikation des ersten Entwurfspakets entsprechend der anderen Benutzereingabe; und
Speicherung des modifizierten ersten Entwurfspakets als ein zweites Entwurfspaket; und
wobei der Empfang der Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung des veröffentlichten Pakets den Empfang einer Benutzerangabe zur Veröffentlichung eines ausgewählten Entwurfspakets umfasst, um das veröffentlichte Paket zu erzeugen.

[0325] Verfahren nach einem der vorangegangenen Aspekte, wobei der Empfang der Benutzereingabe, welche die Modifikation an dem aktuellen Prozesselementobjekt angibt, mindestens einen der folgenden Punkte umfasst:
Empfang einer Benutzereingabe, die eine Modifikation an einem aktuellen Instanzobjekt angibt, worauf das aktuelle Prozesselementobjekt basiert, oder
Empfang einer Benutzereingabe, die eine Modifikation an einem aktuellen Klassenobjekt angibt, worauf das aktuelle Prozesselementobjekt basiert.

[0326] Vorrichtung, die zur Ausführung eines der vorangegangenen Verfahren oder einer Kombination der vorangegangenen Verfahren konfiguriert ist.

[0327] Daher ermöglichen die in diesem Dokument abgehandelten Techniken, Systeme, Verfahren, Vorrichtungen und Geräte angesichts des Vorangegangenen die Integration von Änderungen an Modulklassen und anderen übergeordneten Objekten in einer Prozessanlage oder ein Prozesssteuerungssystem, um dessen Betrieb und/oder Verhalten auf eine kontrollierte Art über die Zeit zu ändern, damit Echtzeitvorgänge in Teilen der Prozessanlage oder des Prozesssteuerungssystems (oder in einigen Fällen der Gesamtheit der Anlage oder des Systems) nicht nachteilig beeinflusst werden. Außerdem wird das Auftreten von unnötigen Verzögerungen bei der Anwendung von Änderungen auf Teile der Prozessanlage oder des Prozesssteuerungssystems (oder in einigen Fällen der Gesamtheit der Anlage oder des Systems) verringert, da diese Änderungen inkrementell oder phasenförmig auf Prozesselemente der Prozessanlage oder des -systems angewandt werden können (z. B. anstatt bis zu einem Zeitpunkt zu warten, der für alle Prozesselemente geeignet ist, um

aktualisiert zu werden, die untergeordneten Objekten des geänderten übergeordneten Objekts entsprechen), wodurch die Gesamteffizienz und -produktivität der Prozessanlage oder des -systems über die Zeit erhöht wird.

[0328] Weiterhin ermöglichen die in diesem Dokument abgehandelten Techniken, Systeme, Verfahren, Vorrichtungen und Geräte die Genehmigung von Entwürfen von Modifikationen oder Änderungen an übergeordneten Objekten und/oder an untergeordneten Objekten, bevor sie auf ein Prozesselement in einer Laufzeitumgebung der Prozessanlage oder das Steuerungssystem angewandt oder instantiiert werden (ob von einem geänderten übergeordneten Objekt oder einem geänderten untergeordneten Objekt). Daher wird die Möglichkeit einer unvollständigen oder unkorrekten Modifikation oder die Möglichkeit einer unabsichtlich in die Anlage oder das System integrierten Änderung bedeutend verringert, und die Genauigkeit der in die Prozessanlage oder das -system integrierten Änderungen wird erhöht, wodurch die Qualität, Effizienz und Sicherheit von Vorgängen der Prozessanlage oder des -systems erhöht werden.

[0329] Darüber hinaus ermöglichen die in diesem Dokument in Bezug auf die Grafikelemente und -anzeigen abgehandelten Techniken, Systeme, Verfahren, Vorrichtungen und Geräte, dass eine Prozessanlage oder ein Prozesssteuerungssystem sicherer und effizienter in Echtzeit überwacht, gesteuert und/oder betrieben werden kann. Insbesondere sind Bediener in der Lage, sowohl in einer Echtzeit- als auch in einer Laufzeitbetriebsumgebung der Anlage und in einer Konfigurationsumgebung speziell auf die spezifische Überwachung und/oder den Betrieb eines oder mehrerer Teile des Prozesssteuerungssystems oder der -anlage zugeschnittene oder angepasste Grafikelemente und/oder -anzeigen zu konfigurieren. Zusätzlich oder alternativ können eines oder mehrere der Grafikelemente und/oder -anzeigen auf einen bestimmten Bediener, eine Gruppe von Bedienern, eine organisatorische Entität, ein Unternehmen oder andere Entität spezifisch zugeschnitten oder angepasst sein. Bediener sind in der Lage, diese angepassten Grafikelemente und/oder -anzeigen für allgemeinen (z. B. anlagenweit oder systemweit, Echtzeit- oder Konfiguration) Zugriff, Verwendung, Wiederverwendung und Integration zu speichern. Dementsprechend werden Unterbrechungen und Fehler von Bedienern verringert, da die Konfiguration von Grafikelement(en) und/oder Grafikanzeige(n) in mehreren Umgebungen verschlankt und anpassbar werden, wodurch die Bediener in die Lage versetzt werden, die Prozessanlage oder das -system effizient und sicher zu bedienen.

[0330] Weiterhin können dann, wenn das/die Grafikelement(e) und/oder -anzeige(n) für spezifische Zwecke von spezifischen Teilen oder Entitäten der Pro-

zessanlage oder des -systems angepasst werden, Echtzeitdaten, die von den spezifischen Teilen oder Entitäten der Prozessanlage oder des -systems erzeugt werden (z. B. bei der Steuerung eines oder mehrerer Prozesse), und die einen notwendigen manuellen und/oder automatischen Eingriff erfordern, leicht und schnell unter Verwendung des/der Grafikelement(e) und/oder -anzeige(n) unterschieden werden. In einigen Fällen ergeben die mit dem Eingriff in Zusammenhang stehenden Daten, Konfigurationen und/oder Anweisungen, die zu der Prozessanlage oder dem -system geliefert wurden, eine Veränderung an der Prozessanlage oder dem -system (z. B. eine aktualisierte oder neue Konfiguration an einem Prozesselement) oder an Vorgängen davon. In einigen Fällen veranlassen die gelieferten Daten, Konfigurationen und/oder Anweisungen die Prozessanlage oder das Steuerungssystem zur Ausführung einer Aktion (z. B. Entfernung eines bestimmten Prozesselements vom Betrieb, erneutes Routing der von einer Quelle erzeugten Daten zu einer anderen Quelle usw.). Dementsprechend werden alle notwendigen Modifikationen an der Steuerung und/oder an den Vorgängen des einen oder der mehreren Teile der Prozessanlage oder des Prozesssteuerungssystems schneller bestimmt und in die Laufzeitumgebung der Prozessanlage oder des Steuerungssystems integriert, da die in diesem Dokument beschriebenen Techniken, Verfahren und Systeme ermöglichen, dass Grafikelemente und/oder -anzeigen besser angepasste und detaillierte Informationen erzeugen (und insbesondere in Bezug auf von der Prozessanlage oder dem -system erzeugte Echtzeitdaten). Folglich werden Effizienz und Sicherheit der Prozessanlage oder des -systems mit den in diesem Dokument beschriebenen Techniken, Verfahren und Systemen weiter erhöht.

Patentansprüche

1. Verfahren zur flexiblen Konfiguration einer Prozessanlage, welches Folgendes umfasst:
Empfang einer Angabe einer Modifikation an einem aktuellen übergeordneten Objekt, welches einer Entität der Prozessanlage entspricht;
Weitergabe der Modifikation an jedes aktuelle untergeordnete Objekt einer ersten Untergruppe eines Satzes aktueller untergeordneter Objekte zur Erstellung einer ersten Untergruppe modifizierter untergeordneter Objekte; wobei:
jedes aktuelle untergeordnete Objekt, welches in dem Satz aktueller untergeordneter Objekte enthalten ist, von dem aktuellen übergeordneten Objekt abgeleitet ist und einem jeweiligen spezifischen Element der Prozessanlage entspricht, und
dass das jeweilige spezifische Element entsprechend einer Instantiierung von jedem aktuellen untergeordneten Objekt arbeitet, um eine Funktion entsprechend einem oder mehreren, in der Prozessanlage gesteuerten Prozessen bereitzustellen;

Nicht-Weitergabe der Modifikation an eine zweite Untergruppe des Satzes aktueller untergeordneter Objekte;

Veranlassung der Erzeugung einer Instantiierung des ersten Satzes modifizierter aktueller untergeordneter Objekte und Veranlassung der Ausführung der Instantiierung der ersten Untergruppe modifizierter aktueller untergeordneter Objekte in einem ersten Satz jeweiliger spezifischer Elemente, sodass der erste Satz jeweiliger spezifischer Elemente in Laufzeit entsprechend der Instantiierung der ersten Untergruppe modifizierter aktueller untergeordneter Objekte arbeitet; und

Aufrechterhaltung einer Instantiierung der zweiten Untergruppe des Satzes aktueller untergeordneter Objekte, sodass ein zweiter Satz jeweiliger spezifischer Elemente entsprechend der zweiten Untergruppe aktueller untergeordneter Objekte in Laufzeit entsprechend der Instantiierung der zweiten Untergruppe aktueller untergeordneter Objekte arbeitet.

2. Verfahren nach Anspruch 1, welches weiterhin den Empfang mindestens eines der folgenden Punkte umfasst: Angabe einer Auswahl der ersten Untergruppe des Satzes aktueller untergeordneter Objekte oder eine Angabe einer Auswahl der zweiten Untergruppe des Satzes aktueller untergeordneter Objekte.

3. Verfahren nach einem der vorangegangenen Ansprüche, wobei mindestens eine von der Weitergabe der Modifikation an die erste Untergruppe des Satzes aktueller untergeordneter Objekte oder Nicht-Weitergabe der Modifikation an die zweite Untergruppe des Satzes aktueller untergeordneter Objekte auf einer Verteilungsrichtlinie basiert.

4. Verfahren nach Anspruch 3, wobei die Verteilungsrichtlinie angibt, ob Änderungen an mindestens eines der folgenden weiterzugeben sind oder nicht: ein bestimmtes Klassenobjekt oder ein bestimmtes Prozesselementobjekt.

5. Verfahren nach einem der vorangegangenen Ansprüche, umfassend mindestens einen der folgenden Punkte:

dass die Weitergabe der Modifikation an jedes aktuelle untergeordnete Objekt der ersten Untergruppe des aktuellen untergeordneten Objekts die Weitergabe der Modifikation an ein erstes aktuelles Instanzobjekt umfasst, welches von dem aktuellen übergeordneten Objekt abgeleitet ist, wobei mindestens eines der in der ersten Untergruppe aktueller untergeordneter Objekte enthaltenen aktuellen untergeordneten Objekte von dem ersten aktuellen Instanzobjekt abgeleitet ist; oder

dass die Nicht-Weitergabe der Modifikation an die zweite Untergruppe die Nicht-Weitergabe der Modifikation an ein zweites aktuelles Instanzobjekt umfasst, welches von dem aktuellen übergeordneten

Objekt abgeleitet ist, wobei mindestens eines der in der zweiten Untergruppe aktueller untergeordneter Objekte enthaltenen aktuellen untergeordneten Objekte von dem zweiten aktuellen Instanzobjekt abgeleitet ist.

6. Verfahren nach einem der vorangegangenen Ansprüche, wobei der Empfang der Angabe der Modifikation an dem aktuellen übergeordneten Objekt den Empfang einer Angabe von mindestens einem der folgenden Punkte umfasst:

Hinzufügung zu einem Inhalt des aktuellen übergeordneten Objekts,

Löschung mindestens eines ersten Teils des Inhalts des aktuellen übergeordneten Objekts,

Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen übergeordneten Objekts oder mindestens eines zweiten Teils des Inhalts des aktuellen übergeordneten Objekts,

Aktivierung eines deaktivierten Inhalts des aktuellen übergeordneten Objekts,

Änderung eines in dem aktuellen übergeordneten Objekt enthaltenen Werts,

Auflösung eines in dem aktuellen übergeordneten Objekt enthaltenen Verweises, oder

Auflösung eines in dem aktuellen übergeordneten Objekt enthaltenen Verweises.

7. Verfahren nach einem der vorangegangenen Ansprüche, welches weiterhin die Verzögerung der Ausführung der Instantiierung der ersten Untergruppe modifizierter aktueller untergeordneter Objekte an dem ersten Satz jeweiliger spezifischer Elemente bis zu einem spezifizierten Zeitpunkt umfasst.

8. Verfahren nach Anspruch 7, wobei der spezifizierte Zeitpunkt ein erster spezifizierter Zeitpunkt ist und das Verfahren weiterhin Folgendes umfasst.

Weitergabe der Modifikation an jedes aktuelle Objekt einer dritten Untergruppe des Satzes aktueller untergeordneter Objekte zur Erstellung einer zweiten Untergruppe modifizierter untergeordneter Objekte; und Verzögerung einer Ausführung einer Instantiierung der zweiten Untergruppe modifizierter untergeordneter Objekte an einem dritten Satz jeweiliger spezifischer Elemente bis zu einem zweiten spezifizierten Zeitpunkt, der sich von dem ersten spezifizierten Zeitpunkt unterscheidet.

9. Verfahren nach einem der vorangegangenen Ansprüche, wobei der erste Satz jeweiliger spezifischer Elemente mindestens eines von einem Anzeigemodul, einem Steuerungsmodul oder einem Modul umfasst, welches Kommunikationen unter Verwendung eines in der Prozessanlage enthaltenen Netzwerks routet.

10. Verfahren nach einem der vorangegangenen Ansprüche, wobei der Empfang der Angabe der Modifikation an dem aktuellen übergeordneten Objekt

den Empfang einer Angabe einer Modifikation an einem aktuellen Klassenprozessobjekt umfasst.

11. Verfahren nach einem der vorangegangenen Ansprüche,

welches weiterhin Folgendes umfasst:

Empfang einer Angabe einer Modifikation an einem bestimmten aktuellen untergeordneten Objekt, wobei das aktuelle untergeordnete Objekt in dem Satz von dem übergeordneten Objekt abgeleiteter aktueller untergeordneter Objekte enthalten ist, wobei das bestimmte aktuelle untergeordnete Objekt kein Element der ersten Untergruppe des Satzes aktueller untergeordneter Objekte ist und das bestimmte aktuelle untergeordnete Objekt kein Element der zweiten Untergruppe aktueller untergeordneter Objekte ist; und Weitergabe der Modifikation des bestimmten aktuellen untergeordneten Objekts an das aktuelle übergeordnete Objekt; und

wobei der Empfang der Angabe der Modifikation an dem aktuellen übergeordneten Objekt den Empfang einer Angabe der weitergegebenen Modifikation des bestimmten aktuellen untergeordneten Objekts für das aktuelle übergeordnete Objekt umfasst.

12. Vorrichtung zur flexiblen Konfiguration einer Prozessanlage, welche Folgendes umfasst:

einen flexiblen Konfigurator, der kommunikativ mit einem oder mehreren, in der Prozessanlage enthaltenen spezifischen Elementen gekoppelt ist, wobei jedes des einen oder der mehreren spezifischen Elemente einem oder mehreren, von der Prozessanlage gesteuerten Prozessen entspricht, wobei der flexible Konfigurator computerausführbare Anweisungen umfasst, die in einem oder mehreren materiellen, nicht vorübergehenden Speichern gespeichert sind, wobei die computerausführbaren Anweisungen von einem oder mehreren Prozessoren ausführbar sein können für:

den Empfang einer Angabe einer Benutzermodifikation entsprechend einem aktuellen übergeordneten Objekt entsprechend einer Entität der Prozessanlage;

die Weitergabe der Benutzermodifikation an jedes aktuelle untergeordnete Objekt einer ersten Untergruppe eines Satzes aktueller untergeordneter Objekte zur Erstellung eines Satzes modifizierter untergeordneter Objekte, wobei:

jedes aktuelle untergeordnete Objekt, welches in dem Satz aktueller untergeordneter Objekte enthalten ist, von dem aktuellen untergeordneten Objekt abgeleitet ist und einem jeweiligen spezifischen Element der Prozessanlage entspricht, und das jeweilige spezifische Element entsprechend einer Instantiierung von jedem aktuellen untergeordneten Objekt arbeitet, um eine Funktion entsprechend dem einem oder den mehreren, in der Prozessanlage gesteuerten Prozessen bereitzustellen;

die Verhinderung der Weitergabe der Benutzermodifikation an eine zweite Untergruppe des Satzes aktueller untergeordneter Objekte;

Veranlassung der Übertragung einer Instantiierung des Satzes modifizierter untergeordneter Objekte an einen ersten Satz jeweiliger spezifischer Elemente, sodass die erste Untergruppe jeweiliger spezifischer Elemente in Laufzeit entsprechend der Instantiierung des ersten Satzes modifizierter untergeordneter Objekte arbeitet;

Aufrechterhaltung einer Instantiierung einer zweiten Untergruppe des Satzes aktueller untergeordneter Objekte in einem zweiten Satz jeweiliger spezifischer Elemente entsprechend der zweiten Untergruppe aktueller untergeordneter Objekte, sodass der zweite Satz jeweiliger spezifischer Elemente in Laufzeit entsprechend der Instantiierung der zweiten Untergruppe aktueller untergeordneter Objekte arbeitet; und Durchführung des Verfahrens nach Null oder mehreren der vorangegangenen Ansprüche.

13. Verfahren nach Anspruch 12, wobei mindestens eine von: der ersten Untergruppe des Satzes aktueller untergeordneter Objekte eine ausgewählte erste Untergruppe aktueller untergeordneter Objekte ist; oder die zweite Untergruppe des Satzes aktueller untergeordneter Objekte eine ausgewählte zweite Untergruppe aktueller untergeordneter Objekte ist.

14. Verfahren nach Anspruch 12 oder Anspruch 13, wobei die Weitergabe der Benutzermodifikation oder der Verhinderung der Weitergabe der Benutzermodifikation auf mindestens einem Verteilungsparameter entsprechend dem einen oder mehreren Objekten basiert, wobei das eine oder die mehreren Objekte mindestens eines umfassen von: dem aktuellen übergeordneten Objekt, der ersten Untergruppe des Satzes aktueller untergeordneter Objekte, der zweiten Untergruppe des Satzes aktueller untergeordneter Objekte, einem Instanzobjekt, welches ein übergeordnetes Objekt von mindestens einem aktuellen untergeordneten Objekt ist, welches sich in der ersten oder der zweiten Untergruppe aktueller untergeordneter Objekte befindet und welches von dem aktuellen übergeordneten Objekt abgeleitet ist, oder jedem aktuellen untergeordneten Objekt.

15. Vorrichtung nach Anspruch 14, wobei der mindestens eine Verteilungsparameter modifizierbar ist.

16. Vorrichtung nach Anspruch 14 oder Anspruch 15, wobei der dem einen oder den mehreren Objekten entsprechende Verteilungsparameter eine Richtlinie zur Annahme einer oder mehrerer weitergegebener Modifikationen an dem einen oder den mehreren Objekten angibt.

17. Vorrichtung nach einem der Ansprüche 14–16, wobei der dem einen oder den mehreren Objekten entsprechende Verteilungsparameter eine Richtlinie

zur Weitergabe einer oder mehrerer Modifikationen an dem einen oder den mehreren Objekten angibt.

18. Vorrichtung nach einem der Ansprüche 12–17, wobei:

mindestens eines der in der ersten Untergruppe aktueller untergeordneter Objekte enthaltenen aktuellen untergeordneten Objekte von einem aktuellen Instanzobjekt abgeleitet ist;
das aktuelle Instanzobjekt vom dem aktuellen übergeordneten Objekt abgeleitet ist;
die Benutzermodifikation entsprechend einem aktuellen übergeordneten Objekt eine Benutzermodifikation an dem aktuellen übergeordneten Objekt ist;
die Weitergabe der Benutzermodifikation an das mindestens eine der aktuellen untergeordneten Objekte auf einer Weitergabe der Benutzermodifikation an das aktuelle Instanzobjekt basiert; und
die Weitergabe der Benutzermodifikation an das aktuelle Instanzobjekt auf der Benutzermodifikation an dem aktuellen übergeordneten Objekt basiert.

19. Vorrichtung nach einem der Ansprüche 12–18, wobei:

die Benutzermodifikation entsprechend dem aktuellen übergeordneten Objekt eine Benutzermodifikation ist, die von einer Benutzermodifikation zu einem bestimmten aktuellen untergeordneten Objekt weitergegeben wird;
das bestimmte aktuelle untergeordnete Objekt von dem aktuellen übergeordneten Objekt abgeleitet ist;
das bestimmte aktuelle untergeordnete Objekt kein Element der ersten Untergruppe aktueller untergeordneter Objekte ist; und
das bestimmte aktuelle untergeordnete Objekt kein Element der zweiten Untergruppe aktueller untergeordneter Objekte ist.

20. Vorrichtung nach einem der Ansprüche 12–19, wobei mindestens ein in der zweiten Untergruppe aktueller untergeordneter Objekte enthaltenes aktuelles untergeordnetes Objekt von einem aktuellen Instanzobjekt abgeleitet ist, und das aktuelle Instanzobjekt von dem aktuellen übergeordneten Objekt abgeleitet ist.

21. Verfahren nach einem der vorangegangenen Ansprüche 12–20, wobei die Benutzermodifikation an dem ersten aktuellen übergeordneten Objekt mindestens einen der folgenden Punkte aufweist:

Hinzufügung zu einem Inhalt des aktuellen übergeordneten Objekts,
Löschung mindestens eines ersten Teils des Inhalts des aktuellen übergeordneten Objekts,
Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen übergeordneten Objekts oder mindestens eines zweiten Teils des Inhalts des aktuellen übergeordneten Objekts,
Aktivierung eines deaktivierten Inhalts des aktuellen übergeordneten Objekts,

Änderung eines in dem aktuellen übergeordneten Objekt enthaltenen Werts,
Änderung eines in dem aktuellen übergeordneten Objekt enthaltenen Verweises, oder
Auflösung eines in dem aktuellen übergeordneten Objekt enthaltenen Verweises.

22. Vorrichtung nach einem der Ansprüche 12–21, wobei das aktuelle übergeordnete Objekt eines von einem Klassenprozessobjekt oder von dem Klassenprozessobjekt abgeleiteten Instanzobjekt ist.

23. Vorrichtung nach einem der Ansprüche 12–22, wobei die Funktion entsprechend dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen eine Funktion zur Darstellung einer Anzeige auf einer Benutzeroberfläche umfasst, wobei die Anzeige Echtzeitinformationen umfasst, die von dem einen oder den mehreren, von der Prozessanlage gesteuerten Prozessen erzeugt sind.

24. Vorrichtung nach einem der Ansprüche 12–23, wobei die Funktion entsprechend dem einen oder den mehreren, von der Prozessanlage gesteuerten Prozessen eine zur Steuerung des einen oder der mehreren Prozesse ausgeführte Steuerungsfunktion umfasst.

25. Vorrichtung nach einem der Ansprüche 12–24, wobei die Funktion entsprechend dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen eine Ausgabefunktion umfasst, die zum Routen einer Kommunikation ausgeführt wird, deren Inhalte zur Steuerung des einen oder der mehreren Prozesse verwendet werden.

26. Vorrichtung nach einem der Ansprüche 12–25, wobei die Übertragung des Satzes modifizierter untergeordneter Objekte zu dem ersten Satz jeweiliger spezifischer Elemente erst nach der Genehmigung oder Veröffentlichung des Satzes modifizierter untergeordneter Objekte erfolgt.

27. Vorrichtung nach Anspruch 26, wobei mindestens ein in dem Satz genehmigter oder veröffentlichter modifizierter untergeordneter Objekte enthaltenes untergeordnetes Objekt umbenannt wird.

28. Verfahren zur flexiblen Konfiguration einer Prozessanlage, welches Folgendes umfasst:

Empfang einer Angabe einer Aktualisierung an einem übergeordneten Objekt, wobei jedes untergeordnete Objekt einer Vielzahl untergeordneter Objekte von dem übergeordneten Objekt abgeleitet ist und eine Instantiierung von jedem untergeordneten Objekt in einem jeweiligen Element ausgeführt wird, um eine Funktion entsprechend der Steuerung eines oder mehrerer Prozesse in der Prozessanlage auszuführen;

Aktualisierung des übergeordneten Objekts zur Erzeugung eines aktualisierten übergeordneten Objekts basierend auf der Angabe;

Weitergabe der Aktualisierung zur Erzeugung eines Satzes aktualisierter untergeordneter Objekte an eine erste Untergruppe der Vielzahl untergeordneter Objekte;

Verhinderung der Weitergabe der Aktualisierung an eine zweite Untergruppe der Vielzahl untergeordneter Objekte;

Veranlassung der Aktualisierung jeweiliger Konfigurationen jeweiliger Elemente entsprechend dem Satz aktualisierter untergeordneter Objekte entsprechend den Instantiierungen des Satzes aktualisierter untergeordneter Objekte, sodass die jeweiligen Elemente in Echtzeit entsprechend den Instantiierungen des Satzes aktualisierter untergeordneter Objekte arbeiten; und

Durchführung des Verfahrens nach Null oder mehr der vorangegangenen Ansprüche.

29. Verfahren nach Anspruch 28, welches weiterhin die Bestimmung von mindestens einem der Elemente der ersten Untergruppe oder der Elemente der zweiten Untergruppe basierend auf einer Verteilungsrichtlinie umfasst.

30. Verfahren nach Anspruch 29, wobei die Verteilungsrichtlinie einen Verteilungsparameter umfasst entsprechend:

dem mindestens einen der Elemente der ersten Untergruppe oder der Elemente der zweiten Untergruppe, oder

einem Instanzobjekt, von dem das mindestens eine der Elemente der ersten Untergruppe oder der Elemente der zweiten Untergruppe abgeleitet ist, wobei das Instanzobjekt von dem übergeordneten Objekt abgeleitet ist.

31. Verfahren nach einem der Ansprüche 28–30, wobei die Weitergabe der Aktualisierung an die erste Untergruppe der Vielzahl aktueller untergeordneter Objekte die Weitergabe der Aktualisierung an ein von dem aktuellen übergeordneten Element abgeleitetes aktuelles Instanzobjekt umfasst, und Weitergabe der Aktualisierung an mindestens ein Element der ersten Untergruppe aktueller untergeordneter Objekte basierend auf der Aktualisierung des aktuellen Instanzobjekts.

32. Verfahren nach einem der Ansprüche 28–31, welches weiterhin Folgendes umfasst:

Empfang einer Angabe einer Aktualisierung eines bestimmten untergeordneten Objekts, wobei das bestimmte untergeordnete Objekt von dem aktuellen übergeordneten Objekt abgeleitet ist, wobei das bestimmte untergeordnete Objekt kein Element der ersten Untergruppe der Vielzahl aktueller untergeordneter Objekte ist, und das bestimmte untergeordnete

Objekt kein Element der zweiten Untergruppe der Vielzahl aktueller untergeordneter Objekte ist; und Weitergabe der Aktualisierung des bestimmten untergeordneten Objekts an das aktuelle übergeordnete Objekt; und

wobei die Angabe der Aktualisierung an das aktuelle übergeordnete Objekt eine Angabe der Aktualisierung des bestimmten untergeordneten Objekts umfasst, welche an das aktuelle übergeordnete Objekt weitergegeben wird.

33. Verfahren nach einem der Ansprüche 28–32, wobei der Empfang der Angabe der Aktualisierung an das aktuelle übergeordnete Objekt den Empfang einer Angabe von mindestens einem der folgenden Punkte umfasst:

Hinzufügung zu einem Inhalt des aktuellen übergeordneten Objekts,

Löschung mindestens eines ersten Teils des Inhalts des aktuellen übergeordneten Objekts,

Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen übergeordneten Objekts oder mindestens eines zweiten Teils des Inhalts des aktuellen übergeordneten Objekts,

Aktivierung eines deaktivierten Inhalts des aktuellen übergeordneten Objekts,

Änderung eines in dem aktuellen übergeordneten Objekt enthaltenen Werts,

Änderung eines in dem aktuellen übergeordneten Objekt enthaltenen Verweises, oder

Auflösung eines in dem aktuellen übergeordneten Objekt enthaltenen Verweises.

34. Verfahren nach einem der Ansprüche 28–33, welches weiterhin den Empfang einer Angabe einer Genehmigung oder einer Veröffentlichung des Satzes aktualisierter untergeordneter Objekte umfasst, und wobei die Veranlassung der Aktualisierung der jeweiligen Konfigurationen der jeweiligen Elemente entsprechend den Instantiierungen des Satzes aktualisierter untergeordneter Objekte die Veranlassung der Aktualisierung der jeweiligen Konfigurationen der jeweiligen Elemente entsprechend den Instantiierungen des Satzes aktualisierter untergeordneter Objekte nach der Genehmigung oder Veröffentlichung des Satzes aktualisierter untergeordneter Objekte umfasst.

35. Verfahren nach Anspruch 34, wobei der Empfang der Angabe der Genehmigung oder Veröffentlichung des Satzes aktualisierter untergeordneter Objekte die Umbenennung von mindestens einem in dem Satz aktualisierter untergeordneter Objekte enthaltenen untergeordneten Objekt umfasst.

36. Verfahren nach einem der Ansprüche 28–35, wobei:

die zweite Untergruppe der Vielzahl von untergeordneten Objekten ein Satz aufrecht erhaltener untergeordneter Objekte ist,

eine aktualisierte Vielzahl untergeordneter Objekte den Satz aktualisierter untergeordneter Objekte und den Satz aufrecht erhaltener untergeordneter Objekte umfasst, und

die Angabe der Aktualisierung zu dem aktuellen übergeordneten Objekt eine erste Aktualisierung ist; und das Verfahren weiterhin Folgendes umfasst:

Empfang einer Angabe einer zweiten Aktualisierung des aktualisierten übergeordneten Objekts;

Aktualisierung des aktualisierten übergeordneten Objekts zu einem weiteren aktualisierten übergeordneten Objekt; und

Weitergabe der zweiten Aktualisierung zu einer Untergruppe der aktualisierten Vielzahl untergeordneter Objekte, wobei die Untergruppe der aktualisierten Vielzahl untergeordneter Objekte mindestens eines von einem Element des Satzes aktualisierter untergeordneter Objekte, einem Element des Satzes aufrecht erhaltener untergeordneter Objekte oder einem anderen untergeordneten Objekt umfasst.

37. Verfahren zur flexiblen Konfiguration einer Prozessanlage, welches Folgendes umfasst:

Empfang einer Benutzereingabe, die eine Modifikation an einem aktuellen übergeordneten Prozessobjekt angibt, an einer Rechnervorrichtung, wovon ein aktuelles untergeordnetes Prozessobjekt abgeleitet wird,

dass das aktuelle untergeordnete Prozessobjekt einem spezifischen Element der Prozessanlage entspricht, und

dass das spezifische Element entsprechend einer Instantiierung des aktuellen untergeordneten Objekts arbeitet, um eine Funktion entsprechend einem oder mehreren, in der Prozessanlage gesteuerten Prozessen bereitzustellen;

Modifikation des aktuellen übergeordneten Prozessobjekts durch die Rechnervorrichtung und basierend auf der Benutzereingabe zur Erzeugung eines modifizierten übergeordneten Prozessobjekts, einschließlich Modifikation des aktuellen untergeordneten Prozessobjekts zur Erzeugung eines modifizierten untergeordneten Prozessobjekts entsprechend dem spezifischen Element;

Speicherung des modifizierten übergeordneten Prozessobjekts als ein entwurfsmodifiziertes übergeordnetes Prozessobjekt und des modifizierten untergeordneten Prozessobjekts als entwurfsmodifiziertes untergeordnetes Prozessobjekt entsprechend dem entwurfsmodifizierten übergeordneten Prozessobjekt in einer mit der Rechnervorrichtung gekoppelten Datenspeicherungsentität;

Verhinderung der Instantiierung des entwurfsmodifizierten untergeordneten Objekts durch die Rechnervorrichtung;

Empfang einer Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten Prozesselementobjekts zur Erzeugung eines veröffentlichten modifizierten Prozesselementobjekts an der Rechnervorrichtung;

Veranlassung (i) der Instantiierung des veröffentlichten modifizierten Prozesselements und (ii) der Ausgabe der Instantiierung des veröffentlichten modifizierten Prozesselements durch die Rechnervorrichtung unter Verwendung einer Kommunikationsverbindung zu einem spezifischen Element der Prozessanlage, sodass das spezifische Element während der Laufzeit arbeitet, entsprechend der Instantiierung des veröffentlichten modifizierten untergeordneten Prozessobjekts; und

Durchführung des Verfahrens nach Null oder einem der vorangegangenen Ansprüche.

38. Verfahren nach Anspruch 37, wobei der Empfang der Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten Prozesselementobjekts den Empfang einer Benutzerangabe zur Veröffentlichung des ausgewählten entwurfsmodifizierten übergeordneten Prozesselementobjekts umfasst.

39. Verfahren nach Anspruch 37 oder 38, wobei der Empfang der Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten übergeordneten Prozessobjekts den Empfang einer Angabe einer Benutzerauswahl des entwurfsmodifizierten übergeordneten Prozessobjekts aus einer Vielzahl übergeordneter Entwurfsprozessobjekte umfasst, wobei jedes übergeordnete Entwurfsprozessobjekt der Vielzahl übergeordneter Entwurfsprozessobjekte eine jeweilige Modifikation an dem aktuellen übergeordneten Prozessobjekt umfasst.

40. Verfahren nach einem der Ansprüche 37–39, wobei der Empfang der Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten untergeordneten Prozessobjekts den Empfang einer Angabe einer Benutzerauswahl des entwurfsmodifizierten untergeordneten Prozessobjekts aus einer Vielzahl untergeordneter Entwurfsprozessobjekte umfasst, wobei jedes untergeordnete Entwurfsprozessobjekt der Vielzahl untergeordneter Entwurfsprozessobjekte eine jeweilige Modifikation an dem aktuellen untergeordneten Prozessobjekt umfasst.

41. Verfahren nach einem der Ansprüche 37–40, wobei:

die Benutzereingabe eine erste Benutzereingabe ist, die Modifikation an dem aktuellen übergeordneten Prozessobjekt eine erste Modifikation an dem aktuellen übergeordneten Prozessobjekt ist, das modifizierte übergeordnete Prozessobjekt ein erstes modifiziertes übergeordnetes Prozessobjekt ist, das modifizierte untergeordnete Prozessobjekt ein erstes modifiziertes untergeordnetes Prozessobjekt ist, das übergeordnete Entwurfsprozessobjekt ein erstes übergeordnetes Entwurfsprozessobjekt ist und das modifizierte untergeordnete Prozessobjekt ein erstes entwurfsmodifiziertes untergeordnetes Prozessobjekt ist; und

das Verfahren weiterhin Folgendes umfasst:

Empfang einer zweiten Benutzereingabe, die eine zweite Modifikation an dem aktuellen übergeordneten Prozessobjekt angibt;

Modifikation des aktuellen übergeordneten Prozessobjekts durch die Rechnervorrichtung und basierend auf der zweiten Benutzereingabe zur Erzeugung eines zweiten modifizierten übergeordneten Prozessobjekts, einschließlich Modifikation des aktuellen untergeordneten Prozessobjekts zur Erzeugung eines zweiten modifizierten untergeordneten Prozessobjekts; und

Speicherung des zweiten modifizierten übergeordneten Prozessobjekts als ein zweites entwurfsmodifiziertes übergeordnetes Prozessobjekt und des zweiten modifizierten untergeordneten Prozessobjekts als ein zweites entwurfsmodifiziertes untergeordnetes Prozessobjekt entsprechend dem zweiten entwurfsmodifizierten übergeordneten Prozessobjekt in der Datenspeicherungsentität.

42. Verfahren nach den Ansprüchen 37–41, wobei die einem oder mehreren, in der Prozessanlage gesteuerten Prozessen entsprechende Funktion eine ist von (i) einer Funktion, die in einem zur Steuerung mindestens eines Teils des einen oder der mehreren Prozesse verwendeten Algorithmus enthalten ist, oder (2) eine Funktion, die Informationen anzeigt, die basierend auf der Steuerung des mindestens einen Teils des einen oder der mehreren Prozesse erzeugt sind.

43. Verfahren nach den Ansprüchen 37–42, welches weiterhin Folgendes umfasst:

Empfang einer Benutzeranfrage an der Rechnervorrichtung zum Vergleich von mindestens zwei der folgenden Punkte: des aktuellen untergeordneten Prozessobjekts, des entwurfsmodifizierten untergeordneten Prozessobjekts, des veröffentlichten modifizierten untergeordneten Prozessobjekts, des aktuellen übergeordneten Prozessobjekts, des entwurfsmodifizierten untergeordneten Prozessobjekts oder des veröffentlichten modifizierten untergeordneten Prozessobjekts; und

Veranlassung der Darstellung eines Vergleichs entsprechend der Benutzeranfrage an einer Benutzerschnittstelle.

44. Verfahren nach den Ansprüchen 37–43, wobei das aktuelle übergeordnete Prozessobjekt eines von einem aktuellen Instanzobjekt oder von einem aktuellen Klassenobjekt ist.

45. Verfahren nach den Ansprüchen 37–44, welches weiterhin Folgendes umfasst:

Bestimmung durch die Rechnervorrichtung, dass ein Entwurfspaket eine Vielzahl entwurfsmodifizierter Objekte umfasst, wobei die Vielzahl entwurfsmodifizierter Objekte einer Vielzahl unterschiedlicher Prozessentitäten der Prozessanlage entspricht und die

Vielzahl entwurfsmodifizierter Objekte das entwurfsmodifizierte untergeordnete Prozessobjekt umfasst; Empfang einer Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung eines veröffentlichten Pakets an der Rechnervorrichtung;

nach Empfang der Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung des veröffentlichten Pakets, Veröffentlichung mindestens einer Untergruppe der Vielzahl entwurfsmodifizierter Objekte an der Rechnervorrichtung zur Erzeugung eines Satzes veröffentlichter modifizierter Objekte entsprechend dem veröffentlichten Paket, wobei der Satz veröffentlichter modifizierter Objekte das veröffentlichte modifizierte untergeordnete Prozessobjekt umfasst; und

nach Erzeugung des Satzes veröffentlichter modifizierter Objekte Veranlassung (i) einer Instantiierung des Satzes veröffentlichter modifizierter Objekte und (ii) Ausgabe des instantiierten Satzes veröffentlichter modifizierter Objekte zu der Vielzahl unterschiedlicher Prozessentitäten, sodass die Vielzahl unterschiedlicher Prozessentitäten während Laufzeit arbeitet, entsprechend der Instantiierung des Satzes veröffentlichter modifizierter Objekte zur Bereitstellung jeweiliger Funktionen entsprechend dem einen oder den mehreren, in der Prozessanlage gesteuerten Prozessen.

46. Verfahren nach Anspruch 45, wobei die Bestimmung, dass das Entwurfspaket die Vielzahl entwurfsmodifizierter Objekte umfasst, die Bestimmung umfasst, dass das Entwurfspaket das entwurfsmodifizierte untergeordnete Prozessobjekt und mindestens eines umfasst von: (i) dem entwurfsmodifizierten übergeordneten Prozessobjekt, (ii) einem anderen übergeordneten Prozessobjekt oder (iii) einem anderen untergeordneten Prozessobjekt.

47. Verfahren nach Anspruch 45 oder Anspruch 46, welches weiterhin den Empfang einer Angabe einer Auswahl der Vielzahl entwurfsmodifizierter Objekte an der Rechnervorrichtung umfasst.

48. Verfahren nach einem der Ansprüche 45–47, wobei:

(i) das Entwurfspaket ein erstes Entwurfspaket ist;

(ii) das Verfahren weiterhin Folgendes umfasst:

Empfang einer weiteren Benutzereingabe an der Rechnervorrichtung, die eine Modifikation an dem ersten Entwurfspaket angibt,

Modifikation des ersten Entwurfspakets durch die Rechnervorrichtung und basierend auf der anderen Benutzereingabe entsprechend der anderen Benutzereingabe, und

Speicherung des modifizierten ersten Entwurfspakets als ein zweites Entwurfspaket in der Datenspeicherungsentität; und

(iii) Empfang der Benutzerangabe zur Veröffentlichung des Entwurfspakets zur Erzeugung des veröffentlichten Pakets, der den Empfang einer Benutzer-

angabe zur Veröffentlichung des ausgewählten ersten Entwurfspakets oder des zweiten Entwurfspakets umfasst, um das veröffentlichte Paket zu erzeugen.

49. Vorrichtung zur flexiblen Konfiguration einer Prozessanlage, welche Folgendes umfasst:

einen flexiblen Konfigurator, der kommunikativ mit einer oder mehreren Datenspeichervorrichtungen gekoppelt ist, und kommunikativ mit einem in der Prozessanlage enthaltenen spezifischen Element gekoppelt ist, wobei der flexible Konfigurator computerausführbare Anweisungen umfasst, die in einem oder mehreren materiellen, nicht vorübergehenden Speichern gespeichert sind, wobei die computerausführbaren Anweisungen von einem oder mehreren Prozessoren ausführbar sein können für:

den Empfang einer vom Benutzer angegebenen Modifikation an einem aktuellen übergeordneten Prozessobjekt, wovon eines oder mehrere untergeordnete Objekte abgeleitet sind,

dass das eine oder die mehreren untergeordneten Objekte ein aktuelles untergeordnetes Prozessobjekt entsprechend dem spezifischen Element der Prozessanlage umfassen,

dass das spezifische Element der Prozessanlage eine Instantiierung des aktuellen untergeordneten Prozessobjekts umfasst, und

dass die Instantiierung des aktuellen untergeordneten Prozessobjekts den Betrieb des spezifischen Elements in Laufzeit veranlasst, um eine zu einem in der Prozessanlage gesteuerten Prozess gehörende Funktion bereitzustellen;

die Modifizierung des aktuellen übergeordneten Prozessobjekts basierend auf der Angabe der vom Benutzer angegebenen Modifikation zur Erzeugung eines modifizierten übergeordneten Prozessobjekts und eines modifizierten untergeordneten Prozessobjekts für das spezifische Element;

Speicherung des modifizierten übergeordneten Prozessobjekts als ein entwurfsmodifiziertes übergeordnetes Prozessobjekt und des modifizierten untergeordneten Prozessobjekts als ein entwurfsmodifiziertes untergeordnetes Prozessobjekt in der einen oder den mehreren Datenspeichervorrichtungen;

Verhinderung der Instantiierung des entwurfsmodifizierten untergeordneten Prozessobjekts;

Empfang einer Benutzerangabe zur Veröffentlichung des entwurfsmodifizierten übergeordneten Prozessobjekts zur Erzeugung eines veröffentlichten modifizierten übergeordneten Prozessobjekts und eines veröffentlichten modifizierten untergeordneten Prozessobjekts;

Veranlassung (i) der Instantiierung des veröffentlichten modifizierten übergeordneten Prozessobjekts und (ii) der Instantiierung des veröffentlichten modifizierten untergeordneten Prozessobjekts zum Herunterladen zu dem spezifischen Element, sodass das spezifische Element während der Laufzeit arbeitet, entsprechend der Instantiierung des veröffentlichten modifizierten untergeordneten Prozessobjekts; und

Durchführung des Verfahrens nach Null oder mehreren der vorangegangenen Ansprüche.

50. Vorrichtung nach Anspruch 49, wobei die von dem spezifischen Element in Laufzeit bereitgestellte Funktion eine Funktion ist, die in einem zur Steuerung des Prozesses verwendeten Algorithmus enthalten ist.

51. Vorrichtung nach Anspruch 49, wobei die von dem spezifischen Element in Laufzeit bereitgestellte Funktion eine Anzeigefunktion zur Bereitstellung von Echtzeitinformationen an einer Benutzerschnittstelle zur Steuerung des Prozesses in der Prozessanlage ist.

52. Vorrichtung nach einem der Ansprüche 49–51, wobei das aktuelle übergeordnete Objekt eines von folgenden ist: ein Klassenobjekt, ein Instanzobjekt oder ein Bibliotheksobjekt.

53. Vorrichtung nach einem der Ansprüche 49–52, wobei:

das entwurfsmodifizierte übergeordnete Prozessobjekt ein bestimmtes entwurfsmodifiziertes übergeordnetes Prozessobjekt ist, welches in einer Vielzahl entwurfsmodifizierter übergeordneter Prozessobjekte entsprechend dem aktuellen übergeordneten Prozessobjekt enthalten ist; und

die Benutzerangabe zur Veröffentlichung des bestimmten entwurfsmodifizierten übergeordneten Prozessobjekts auf einer Auswahl des bestimmten entwurfsmodifizierten übergeordneten Prozessobjekts aus der Vielzahl entwurfsmodifizierter übergeordneter Prozessobjekte basiert.

54. Vorrichtung nach einem der Ansprüche 49–53, wobei der flexible Konfigurator weiterhin computerausführbare Anweisungen umfasst, die von dem einen oder den mehreren Prozessoren betätigbar sind für:

den Empfang einer Benutzerangabe zur Veröffentlichung eines Entwurfspakets, wobei das Entwurfspaket eine Vielzahl von Entwurfsprozessobjekten umfasst, die Vielzahl von Entwurfsprozessobjekten das entwurfsmodifizierte übergeordnete Prozessobjekt umfasst und die Vielzahl von Entwurfsprozessobjekten einer Vielzahl unterschiedlicher Prozessentitäten der Prozessanlage entspricht;

die Veröffentlichung des Entwurfspakets nach Empfang der Benutzerangabe,

die Veröffentlichung der Vielzahl von Entwurfsprozessobjekten entsprechend dem Entwurfspaket zur Bildung einer Vielzahl veröffentlichter Prozessobjekte; und

Veranlassung der Instantiierung der Vielzahl veröffentlichter Prozessobjekte in der Prozessanlage.

55. Vorrichtung nach Anspruch 54, die mindestens einen der folgenden Punkte umfasst:

dass die Vielzahl der in dem Entwurfspaket enthaltenen Entwurfsprozessobjekte mindestens teilweise von einem Benutzer ausgewählt wird;
 dass die Vielzahl der in dem Entwurfspaket enthaltenen Entwurfsprozessobjekte weiterhin das entwurfsmodifizierte untergeordnete Prozessobjekt umfasst; und
 dass die Benutzerangabe zur Veröffentlichung des Entwurfspakets eine Benutzerangabe zur Veröffentlichung eines ausgewählten Entwurfspakets von einer Vielzahl von Entwurfspaketen umfasst.

56. Verfahren zur flexiblen Konfiguration einer Prozessanlage, die Folgendes umfasst:
 den Empfang einer Eingabe über eine Schnittstelle einer Rechnervorrichtung, die eine Modifikation an einem aktuellen übergeordneten Prozessobjekt angibt, dass das aktuelle übergeordnete Prozessobjekt ein davon abgeleitetes untergeordnetes Prozessobjekt aufweist,
 dass das spezifische Element einen oder mehrere Prozesse in der Prozessanlage entsprechend einer Instantiierung des aktuellen untergeordneten Prozessobjekts steuert, und
 dass das spezifische Element kommunikativ mit der Rechnervorrichtung verbunden ist;
 Modifikation des aktuellen übergeordneten Prozessobjekts durch die Rechnervorrichtung und basierend auf der Eingabe zur Erzeugung eines modifizierten übergeordneten Prozessobjekts, einschließlich Modifikation des aktuellen untergeordneten Prozessobjekts zur Erzeugung eines modifizierten untergeordneten Prozessobjekts;
 Veranlassung einer Instantiierung des modifizierten untergeordneten Prozessobjekts durch die Rechnervorrichtung, welches zu dem spezifischen Element der Prozessanlage erst nach Empfang einer Genehmigung des modifizierten untergeordneten Prozessobjekts zu liefern ist, wobei:
 die Genehmigung des modifizierten untergeordneten Prozessobjekts eine Änderung des Status des modifizierten untergeordneten Prozessobjekts von einem ersten Status mit der Angabe veranlasst, dass Instantiierung nicht zulässig ist, in einen zweiten Status mit der Angabe, dass Instantiierung zulässig ist, und
 die Instantiierung des modifizierten untergeordneten Prozessobjekts in dem spezifischen Element das spezifische Element zur Steuerung des einen oder mehreren Prozesse in der Prozessanlage entsprechend der Instantiierung des modifizierten untergeordneten Prozessobjekts veranlasst; und
 Durchführung des Verfahrens nach Null oder einem der vorangegangenen Ansprüche.

57. Verfahren nach Anspruch 56, wobei die Genehmigung des modifizierten untergeordneten Prozessobjekts in einer Genehmigung des modifizierten übergeordneten Prozessobjekts enthalten ist.

58. Verfahren nach Anspruch 56 oder 57, wobei:

die Eingabe eine erste Eingabe ist, die Modifikation eine erste Modifikation ist, das modifizierte übergeordnete Prozessobjekt ein erstes modifiziertes übergeordnetes Prozessobjekt ist und das modifizierte untergeordnete Prozessobjekt ein erstes modifiziertes untergeordnetes Prozessobjekt ist; und
 das Verfahren weiterhin Folgendes umfasst:
 Empfang einer zweiten Benutzereingabe, die eine zweite Modifikation an dem aktuellen übergeordneten Prozessobjekt angibt;
 Modifikation des aktuellen übergeordneten Prozessobjekts zur Erzeugung eines zweiten modifizierten übergeordneten Prozessobjekts, einschließlich Modifikation des aktuellen untergeordneten Prozessobjekts zur Erzeugung eines zweiten modifizierten untergeordneten Prozessobjekts, und
 Empfang einer Genehmigung des zweiten modifizierten untergeordneten Prozessobjekts um das zweite modifizierte untergeordnete Prozessobjekt zu veranlassen, dass der zweite Status angibt, dass Instantiierung zulässig ist.

59. Verfahren nach Anspruch 58, welches weiterhin den Empfang einer Auswahl von einem des ersten modifizierten untergeordneten Prozessobjekts oder des zweiten modifizierten untergeordneten Prozessobjekts umfasst, und Veranlassung der Ausgabe einer Instantiierung des ausgewählten modifizierten untergeordneten Objekts zu dem spezifischen Element.

60. Verfahren nach Anspruch 58 oder Anspruch 59, wobei der Empfang einer Auswahl des einen von dem ersten modifizierten untergeordneten Prozessobjekt oder dem zweiten modifizierten untergeordneten Prozessobjekt den Empfang einer Auswahl des ersten modifizierten übergeordneten Prozessobjekts oder des zweiten modifizierten übergeordneten Prozessobjekts umfasst.

61. Verfahren nach einem der Ansprüche 56–60, wobei der Empfang der Eingabe, welche die Modifikation an dem aktuellen übergeordneten Prozessobjekt angibt, den Empfang einer Angabe von mindestens einem der folgenden Punkte umfasst:
 Hinzufügung zu einem Inhalt des aktuellen übergeordneten Prozessobjekts,
 Löschung mindestens eines ersten Teils des Inhalts des aktuellen übergeordneten Prozessobjekts,
 Deaktivierung mindestens des ersten Teils des Inhalts des aktuellen übergeordneten Prozesselementobjekts oder mindestens eines zweiten Teils des Inhalts des aktuellen übergeordneten Prozessobjekts,
 Aktivierung eines deaktivierten Inhalts des aktuellen übergeordneten Prozessobjekts,
 Änderung eines in dem aktuellen übergeordneten Prozessobjekt enthaltenen Werts,
 Änderung eines in dem aktuellen übergeordneten Prozessobjekt enthaltenen Verweises,

Auflösung eines in dem aktuellen übergeordneten Prozessobjekt enthaltenen Verweises.

62. Grafikelement, welches auf einer Prozesssteuerungsanzeigenansicht entsprechend einer Prozessanlage dargestellt ist, wobei das Grafikelement Folgendes umfasst:

eine Ausführung einer Instantiierung eines Grafikelementobjekts in einer laufzeitbetriebenen Umgebung der Prozessanlage entsprechend einer in der Prozessanlage enthaltenen Prozessentität, wobei das Grafikelement konfigurierbar ist, wobei das Grafikelementobjekt eine Verbindung zu einem Anzeigenobjekt umfasst, wovon die Prozesssteuerungsanzeigenansicht instantiiert wird, und wobei die Ausführung der Instantiierung des Grafikelementobjekts Folgendes umfasst:

eine visuelle Darstellung der Prozessentität, und eine Verbindung zu dem Prozesselement zum Empfang von Echtzeitdaten entsprechend einem in der Prozessanlage gesteuerten Prozess, wobei die Echtzeitdaten mindestens einem Attribut des Grafikelements entsprechen, wobei das Grafikelement in Null oder mehreren der vorangegangenen Ansprüche enthalten ist.

63. Grafikelement nach Anspruch 62, wobei das Grafikelementobjekt in der Laufzeitbetriebsumgebung der Prozessanlage konfigurierbar ist.

64. Grafikelement nach Anspruch 63, wobei das Grafikelementobjekt in einer Konfigurationsumgebung der Prozessanlage konfigurierbar ist.

65. Grafikelement nach einem der Ansprüche 62–64, wobei eine Konfiguration des Grafikelements, welches mindestens teilweise in der Laufzeitbetriebsumgebung erzeugt wird, an einer Stelle gespeichert wird, die für die Laufzeitbetriebsumgebung und die Konfigurationsumgebung der Prozessanlage zugänglich ist.

66. Grafikelement nach einem der Ansprüche 62–65, wobei das Grafikelementobjekt ein erstes Grafikelementobjekt ist, und wobei das erste Grafikelementobjekt weiterhin eine Verbindung zu einem zweiten Grafikelement umfasst.

67. Grafikelement nach Anspruch 66, wobei eine Instantiierung des zweiten Grafikelementobjekts eine Darstellung eines anderen Grafikelements auf der Prozesssteuerungsanzeigenansicht umfasst.

68. Grafikelement nach Anspruch 66, wobei das Anzeigenobjekt ein erstes Anzeigenobjekt ist, die Prozesssteuerungsanzeigenansicht eine erste Prozesssteuerungsanzeigenansicht ist, und wobei das zweite Grafikelementobjekt, womit das erste Grafikelementobjekt verbunden ist, mit einem zweiten Anzei-

genobjekt entsprechend einer zweiten Prozesssteuerungsanzeigenansicht verbunden ist.

69. Grafikelement nach einem der Ansprüche 62–65, wobei das Grafikelementobjekt ein erstes Grafikelementobjekt ist, und wobei das erste Grafikelementobjekt von einem zweiten Grafikelementobjekt abgeleitet ist.

70. Grafikelement nach Anspruch 69, wobei eine Instantiierung des zweiten Grafikelementobjekts ein anderes Grafikelement oder eine andere Anzeigenansicht umfasst.

71. Grafikelement nach einem der Ansprüche 62–65, wobei das Grafikelementobjekt ein erstes Grafikelementobjekt ist, und wobei das zweite Grafikelementobjekt von dem ersten Grafikelementobjekt abgeleitet ist.

72. Grafikelement nach Anspruch 71, wobei eine Instantiierung des zweiten Grafikelementobjekts ein anderes Grafikelement oder eine andere Anzeigenansicht umfasst.

73. Grafikelement nach einem der Ansprüche 62–72, wobei das mindestens eine Attribut des Grafikelements mindestens eine von einer Eigenschaft, einer Animation oder einem in dem Grafikelementobjekt definierten Ereignishandler entsprechend der visuellen Darstellung der Prozessentität ist.

74. Grafikelement nach Anspruch 73, wobei: die Animation ein dynamisches Verhalten mindestens eines Teils der visuellen Darstellung der Prozessentität definiert, die sich basierend auf den Echtzeitdaten ändert; der Ereignishandler ein Verhalten definiert, welches auszuführen ist, wenn ein Auslöser auftritt; und die Eigenschaft eine Charakteristik der visuellen Darstellung der Prozessentität definiert.

75. Grafikelement nach einem der Ansprüche 62–74, wobei: eine Modifikation des Grafikelementobjekts an einer Benutzerschnittstelle der Laufzeitbetriebsumgebung der Prozessanlage und an einer Benutzerschnittstelle einer Konfigurationsumgebung der Prozessanlage empfangbar ist, und die Modifikation an dem Grafikelementobjekt ein modifiziertes Grafikelementobjekt erzeugt, welches für die Laufzeitbetriebsumgebung der Prozessanlage und für die Konfigurationsumgebung der Prozessanlage zugänglich ist.

76. Grafikelement nach Anspruch 75, wobei das mindestens eine Attribut ein erstes Attribut ist, und wobei die Modifikation an dem Grafikelementobjekt mindestens eines von einer Modifikation an dem ers-

ten Attribut, eine Löschung des ersten Attributs oder eine Hinzufügung eines zweiten Attributs umfasst.

77. Grafikelement nach Anspruch 75 oder Anspruch 76, wobei die Modifikation des Grafikelementobjekts mindestens eine von einer Modifikation an einer Definition der visuellen Darstellung der Prozessentität umfasst.

78. Grafikelement nach einem der Ansprüche 75–77, wobei die Verbindung zu dem Anzeigenobjekt eine erste Anzeigenverbindung ist, und wobei die Modifikation an dem Grafikelementobjekt mindestens eines von einer Modifikation an der ersten Anzeigenverbindung, eine Löschung der ersten Anzeigenverbindung oder eine Hinzufügung einer zweiten Anzeigenverbindung zu einem anderen Anzeigenobjekt umfasst.

79. Grafikelement nach einem der Ansprüche 75–78, wobei:
die Modifikation an dem Grafikelementobjekt eine Außerkraftsetzung einer in dem Grafikelement enthaltenen Definition umfasst,
eine Quelle der Definition ein anderes Grafikelementobjekt ist, wovon das Grafikelementobjekt abgeleitet ist, und
die Definition mindestens eine umfasst von: einer Definition der visuellen Darstellung der Prozessentität, einer Definition der Verbindung zum Empfang der Echtzeitdaten, einer Definition der Verbindung zu dem Anzeigenobjekt oder einer Definition des mindestens einen Attributs.

80. Grafikelement nach Anspruch 79, wobei die Außerkraftsetzung der in dem Grafikelementobjekt enthaltenen Definition eine erste Modifikation ist, und wobei eine zweite Modifikation auf das modifizierte Grafikelementobjekt angewandt wird.

81. Grafikelement nach einem der Ansprüche 62–80, wobei das Grafikelementobjekt eine oder mehrere Ansichten definiert, wobei jede der einen oder mehreren Ansichten einer jeweiligen Anordnung von Teilen der visuellen Darstellung des Grafikelements entspricht.

82. Grafikelement nach einem der Ansprüche 62–81, wobei das Grafikanzeigenobjekt aus einer Kombination von mindestens zwei anderen Grafikanzeigenobjekten erzeugt wird.

83. Mindestens ein Teil der Prozessanlage, die zur Verwendung des Grafikelements nach einem der Ansprüche 62–82 und zur Steuerung des Prozesses unter Verwendung mindestens einer physikalischen Funktion konfiguriert ist.

84. Verfahren, welches von mindestens einem Teil der Prozessanlage unter Verwendung des Grafikele-

ments nach einem der Ansprüche 62–82 ausgeführt wird, wobei die Prozessanlage zur Steuerung des Prozesses unter Verwendung mindestens einer physikalischen Funktion konfiguriert ist.

85. Vorrichtung der Prozessanlage, wobei die Vorrichtung mindestens das Grafikelement nach einem der Ansprüche 62–82 umfasst, wobei die Prozessanlage zur Steuerung des Prozesses unter Verwendung mindestens einer physikalischen Funktion konfiguriert ist.

86. Verfahren von flexibel konfigurierbaren Grafikelementen zur Verwendung in einer Prozessanlage, wobei das Verfahren Folgendes umfasst:
Empfang einer Angabe einer Modifikation an einem bestimmten Grafikelementobjekt, welches in einem Satz von Grafikelementobjekten enthalten ist, in einer Laufzeitbetriebsumgebung, wobei:
eine Instantiierung von jedem Grafikelementobjekt des Satzes von Grafikelementobjekten ein jeweiliges Grafikelementobjekt umfasst, welches auf mindestens einer Prozesssteuerungsanzeigenansicht eine jeweilige Prozessentität der Prozessanlage darstellt; jedes Grafikelementobjekt des Satzes von Grafikelementobjekten mit mindestens einem Anzeigenobjekt verbunden ist, wobei eine Instantiierung von jedem Grafikelementobjekt, welches in der Laufzeitbetriebsumgebung der Prozessanlage ausgeführt wird, die mindestens eine Prozesssteuerungsanzeigenansicht erzeugt, und
die mindestens eine Prozesssteuerungsanzeigenansicht Echtzeitdaten entsprechend der Steuerung eines Prozesses durch die Prozessanlage darstellt; Weitergabe der Modifikation an jedes Grafikelementobjekt einer Untergruppe des Satzes von Grafikelementobjekten in der Laufzeitbetriebsumgebung der Prozessanlage, wobei jedes in der Untergruppe enthaltene Grafikelementobjekt von dem bestimmten Grafikelementobjekt abgeleitet ist, und die Weitergabe der Modifikation an die Untergruppe von Grafikelementobjekten einen Satz modifizierter Grafikelementobjekte erzeugt;
Veranlassung einer Instantiierung eines oder mehrerer Elemente des Satzes modifizierter Grafikelementobjekte, die in der Laufzeitbetriebsumgebung auszuführen ist, sodass eines oder mehrere modifizierte Grafikelemente entsprechend dem einen oder mehreren Elementen des Satzes modifizierter Grafikelementobjekte auf der mindestens einen Prozesssteuerungsanzeigenansicht dargestellt werden; und Durchführung des Verfahrens nach Null oder einem der vorangegangenen Ansprüche.

87. Verfahren nach Anspruch 86, welches weiterhin die Speicherung des Satzes modifizierter Grafikelementobjekte umfasst, damit sie für eine Konfigurationsumgebung der Prozessanlage zugänglich sind.

88. Verfahren nach Anspruch 86 oder 87, wobei der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang einer Angabe einer Modifikation an einer in dem bestimmten Grafikelementobjekt enthaltenen Definition umfasst, wobei die Definition mindestens eine umfasst von: einer Definition einer visuellen Form oder eines Inhalts eines von dem bestimmten Grafikelementobjekt instantiierten bestimmten Grafikelements, einer Definition der Verbindung zu dem mindestens einen Anzeigenobjekt oder einer Definition eines Attributs des bestimmten Grafikelements.

89. Verfahren nach Anspruch 88, wobei das von dem bestimmten Grafikelementobjekt definierte Attribut eines von folgenden ist:
eine Animation, die ein dynamisches Verhalten mindestens eines Teils des bestimmten Grafikelements definiert, welches sich basierend auf den Echtzeitdaten ändert;
ein Ereignishandler, der ein Verhalten definiert, welches von dem bestimmten Grafikelement auszuführen ist, wenn ein Auslöser auftritt; und
die Eigenschaft eine Charakteristik des bestimmten Grafikelements definiert.

90. Verfahren nach Anspruch 88 oder 89, wobei der Empfang der Angabe der Modifikation an der in dem bestimmten Grafikelementobjekt enthaltenen Definition den Empfang einer Angabe einer Modifikation an einer in dem bestimmten Grafikelementobjekt enthaltenen Definition umfasst, die von einem anderen Grafikelementobjekt bereitgestellt ist, wovon das bestimmte Grafikelementobjekt abgeleitet ist.

91. Verfahren nach einem der Ansprüche 88–90, wobei der Empfang der Angabe der Modifikation an der in dem bestimmten Grafikelementobjekt enthaltenen Definition den Empfang einer Angabe einer Löschung der Definition oder einer Änderung der Definition umfasst.

92. Verfahren nach einem der Ansprüche 88–91, wobei der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang einer Angabe einer dem bestimmten Grafikelementobjekt hinzugefügten zusätzlichen Definition umfasst, wobei die zusätzliche Definition mindestens einem von folgenden Punkten entspricht: einer anderen visuellen Form oder Inhalt des bestimmten Grafikelements, einer anderen Verbindung zu einem anderen Anzeigenobjekt oder einem anderen Attribut des bestimmten Grafikelements.

93. Verfahren nach einem der Ansprüche 86–92, welches weiterhin die Speicherung einer Angabe umfasst, dass die Modifikation an jedes Grafikelementobjekt der Untergruppe weitergegeben wurde.

94. Verfahren nach einem der Ansprüche 86–93, wobei die Modifikation an jedem Grafikelementobjekt eine erste Modifikation ist, und wobei das Verfahren weiterhin die Bestimmung eines Satzes mehrerer Modifikationen an jedem Grafikelementobjekt der Untergruppe umfasst, wobei die mehreren Modifikationen die erste Modifikation umfassen.

95. Verfahren nach einem der Ansprüche 86–94, wobei der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang einer Angabe einer Modifikation an einem anderen Grafikelementobjekt umfasst, von dem das bestimmte Grafikelementobjekt abgeleitet wurde.

96. Verfahren nach einem der Ansprüche 86–95, welches weiterhin die Bestimmung der Untergruppe des Satzes von Grafikelementobjekten basierend auf einer Verteilungsrichtlinie umfasst.

97. Verfahren nach Anspruch 96, wobei die Bestimmung der Untergruppe der Grafikelementobjekte basierend auf der Verteilungsrichtlinie die Bestimmung der Untergruppe von Grafikelementobjekten basierend auf einer Verteilungsrichtlinie entsprechend dem bestimmten Grafikelementobjekt umfasst.

98. Verfahren nach Anspruch 96 oder Anspruch 97, wobei die Bestimmung der Untergruppe der Grafikelementobjekte basierend auf der Verteilungsrichtlinie die Bestimmung der Untergruppe von Grafikelementobjekten basierend auf einer Verteilungsrichtlinie entsprechend mindestens einiger Elemente der Untergruppe von Grafikelementobjekten umfasst.

99. Verfahren von flexibel konfigurierbaren Grafikelementen zur Verwendung in einer Prozessanlage, wobei das Verfahren Folgendes umfasst:
Empfang einer Angabe einer Modifikation an einem bestimmten Grafikelementobjekt, welches in einem Satz von Grafikelementobjekten enthalten ist, wobei:
eine Ausführung einer Instantiierung von jedem Grafikelementobjekt des Satzes von Grafikelementobjekten (i) ein jeweiliges Grafikelementobjekt erzeugt, welches auf jeweils einer oder mehreren Prozesssteuerungsanzeigenansichten dargestellt wird und (ii) eine jeweilige Prozessentität der Prozessanlage darstellt,
die Erzeugung der jeweils einen oder mehreren Prozesssteuerungsanzeigenansichten (a) von einer Ausführung einer Instantiierung von jeweils einem oder mehreren Anzeigenobjekten und (b) Darstellung von Echtzeitdaten entsprechend der Steuerung eines Prozesses durch die Prozessanlage erfolgt, und
jedes Grafikelementobjekt des Satzes von Grafikelementobjekten mit dem jeweils einen oder mehreren Anzeigenobjekten verbunden ist;
Weitergabe der Modifikation an jedes Grafikelementobjekt einer ersten Untergruppe des Satzes von Gra-

fikelementobjekten, wobei jedes in der ersten Untergruppe enthaltene Grafikelementobjekt von dem bestimmten Grafikelementobjekt abgeleitet ist, und die Weitergabe der Modifikation an die erste Untergruppe von Grafikelementobjekten einen Satz modifizierter Grafikelementobjekte erzeugt;

Veranlassung der Darstellung eines oder mehrerer Grafikelemente entsprechend einem oder mehreren Elementen des Satzes modifizierter Grafikelementobjekte auf einer oder mehreren entsprechenden Prozesssteuerungsanzeigenansichten, wenn die Instantiierungen des einen oder der mehreren Elemente des Satzes modifizierter Grafikelementobjekte ausgeführt werden;

Nicht-Weitergabe der Modifikation an jedes Grafikelementobjekt einer zweiten Untergruppe des Satzes von Grafikelementobjekten, wobei jedes in der zweiten Untergruppe enthaltene Grafikelementobjekt von dem bestimmten Grafikelementobjekt abgeleitet ist; und

Durchführung des Verfahrens nach Null oder einem der vorangegangenen Ansprüche.

100. Verfahren nach Anspruch 99, wobei der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt an einer Benutzerschnittstelle in einer Laufzeitbetriebsumgebung der Prozessanlage umfasst.

101. Verfahren nach Anspruch 99 oder Anspruch 100, wobei das Verfahren weiterhin die Speicherung des Satzes modifizierter Grafikelementobjekte an einer Stelle umfasst, die für die Laufzeitbetriebsumgebung und die Konfigurationsumgebung der Prozessanlage zugänglich ist.

102. Vorrichtung nach einem der Ansprüche 99–101, wobei:

der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang einer Angabe einer Modifikation an einer in dem bestimmten Grafikelementobjekt enthaltenen Definition umfasst, und

die Definition mindestens eine umfasst von: einer Definition einer visuellen Form oder eines Inhalts eines bestimmten Grafikelementobjekts entsprechend dem bestimmten Grafikelementobjekt, einer Definition einer Verbindung zu dem mindestens einen Anzeigenobjekt oder einer Definition eines Attributs des bestimmten Grafikelements.

103. Verfahren nach Anspruch 102, wobei das Attribut des bestimmten Grafikelementobjekts eines ist von:

einer Animation, die ein dynamisches Verhalten mindestens eines Teils der visuellen Form oder des Inhalts des bestimmten Grafikelements definiert, die sich basierend auf den Echtzeitdaten ändert.

einem Ereignishandler, der ein Verhalten definiert, welches von dem bestimmten Grafikelement auszuführen ist, wenn ein Auslöser auftritt; und einer Eigenschaft, die eine Charakteristik der visuellen Form oder des Inhalts des bestimmten Grafikelements definiert.

104. Verfahren nach Anspruch 102 oder 103, wobei der Empfang der Angabe der Modifikation an der in dem bestimmten Grafikelementobjekt enthaltenen Definition den Empfang einer Angabe einer Modifikation an einer in dem bestimmten Grafikelementobjekt enthaltenen Definition umfasst, die von einem anderen Grafikelementobjekt bereitgestellt ist, wovon das bestimmte Grafikelementobjekt abgeleitet ist.

105. Verfahren nach einem der Ansprüche 102–104, wobei der Empfang der Angabe der Modifikation an der Definition den Empfang einer Angabe einer von einer Löschung der Definition oder einer Änderung der Definition umfasst.

106. Verfahren nach einem der Ansprüche 102–105, wobei der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang einer Angabe einer dem bestimmten Grafikelementobjekt hinzugefügten zusätzlichen Definition umfasst, wobei die zusätzliche Definition mindestens einem von folgenden Punkten entspricht: einer anderen visuellen Form oder Inhalt des bestimmten Grafikelements, einer anderen Verbindung zu einem anderen Anzeigenobjekt oder einem anderen Attribut des bestimmten Grafikelements.

107. Verfahren nach einem der Ansprüche 99–106, welches weiterhin die Speicherung einer Angabe umfasst, dass die Modifikation an jedes Grafikelementobjekt der ersten Untergruppe weitergegeben wurde.

108. Verfahren nach einem der Ansprüche 99–107, wobei die Modifikation an jedem Grafikelementobjekt eine erste Modifikation ist, und wobei das Verfahren weiterhin die Bestimmung eines Satzes mehrerer Modifikationen an jedem Grafikelementobjekt der ersten Untergruppe umfasst, wobei die mehreren Modifikationen die erste Modifikation umfassen.

109. Verfahren nach einem der Ansprüche 99–108, wobei der Empfang der Angabe der Modifikation an dem bestimmten Grafikelementobjekt den Empfang einer Angabe einer Modifikation an einem anderen Grafikelementobjekt umfasst, von dem das bestimmte Grafikelementobjekt abgeleitet wurde.

110. Verfahren nach einem der Ansprüche 99–109, welches weiterhin die Bestimmung mindestens einer von (i) der ersten Untergruppe von Grafikelementobjekten oder (ii) der zweiten Untergruppe von Grafikelementobjekten basierend auf einer Verteilungsrichtlinie umfasst.

111. Verfahren nach Anspruch 110, wobei die Bestimmung mindestens einer von der ersten oder der zweiten Untergruppe von Grafikelementobjekten basierend auf der Verteilungsrichtlinie die Bestimmung der mindestens einen von der ersten oder der zweiten Untergruppe von Grafikelementobjekten basierend auf einer Verteilungsrichtlinie entsprechend dem bestimmten Grafikelementobjekt umfasst.

112. Verfahren nach Anspruch 110 oder Anspruch 111, wobei die Bestimmung mindestens einer von der ersten oder der zweiten Untergruppe von Grafikelementobjekten basierend auf der Verteilungsrichtlinie die Bestimmung der mindestens einen von der ersten oder der zweiten Untergruppe von Grafikelementobjekten basierend auf einer Verteilungsrichtlinie entsprechend einer von der ersten oder der zweiten Untergruppe von Grafikelementobjekten umfasst.

113. Verfahren zur Verwendung von flexibel konfigurierten Grafikelementen in Prozessanlagen, wobei das Verfahren Folgendes umfasst:

Initiierung des Ladens einer Anzeigenansicht in einer Rechnervorrichtung,

dass die Anzeigenansicht die Ausführung einer Instantiierung eines Anzeigenobjekts mit einer Verbindung zum Empfang von Echtzeitdaten entsprechend der Steuerung eines Prozesses in der Prozessanlage umfasst, wobei die auf der Anzeigenansicht dargestellten Informationen auf den empfangenen Echtzeitdaten basieren;

dass das Anzeigenobjekt mit einem bestimmten Grafikelementobjekt verbunden ist, und

dass das bestimmte Grafikelementobjekt (i) eine Definition einer visuellen Darstellung einer Prozessentität der Prozessanlage umfasst, wobei die Definition der visuellen Darstellung der Prozessentität von einem anderen Grafikelementobjekt bereitgestellt ist, wovon das bestimmte Grafikelementobjekt abgeleitet ist und (ii) eine Modifikation an der Definition der visuellen Darstellung von dem anderen Grafikelementobjekt bereitgestellt wird;

nach der Initiierung des Ladens der Anzeigenansicht, (a) Bestimmung der Definition der von dem anderen Grafikelementobjekt bereitgestellten visuellen Darstellung;

(b) Anwendung der Modifikation auf die bestimmte Definition zur Erzeugung einer modifizierten Definition; und

(c) Instantiierung der modifizierten Definition als ein bestimmtes Grafikelement entsprechend dem bestimmten Grafikelementobjekt; und Durchführung des Verfahrens nach Null oder mehr der vorangegangenen Ansprüche.

114. Verfahren nach Anspruch 113, wobei eine Angabe der Modifikation getrennt von der Definition der visuellen Darstellung gespeichert wird, die von dem anderen Grafikelementobjekt bereitgestellt wird.

115. Verfahren nach Anspruch 113 oder 114, wobei:

die Bestimmung der Definition der von dem anderen Grafikelementobjekt bereitgestellten visuellen Darstellung die Bestimmung einer Definition einer Eigenschaft der visuellen Darstellung umfasst, wobei die Eigenschaft der visuellen Darstellung eine visuelle Charakteristik mindestens eines Teils der visuellen Darstellung umfasst; und

die Anwendung der Modifikation auf die bestimmte Definition mindestens einen der folgenden Punkte umfasst: Entfernung der Eigenschaft von der bestimmten Definition, Ersetzung der Eigenschaft durch eine andere Eigenschaft oder Änderung des mindestens einen Teils der visuellen Darstellung entsprechend der Eigenschaft.

116. Vorrichtung nach einem der Ansprüche 113–115, wobei:

die Bestimmung der Definition der von dem anderen Grafikelementobjekt bereitgestellten visuellen Darstellung die Bestimmung einer Definition einer Form mindestens eines Teils der visuellen Darstellung umfasst; und

die Anwendung der Modifikation auf die bestimmte Definition mindestens einen der folgenden Punkte umfasst: Entfernung der Form von der bestimmten Definition, Ersetzung der Form durch eine andere Form oder Änderung der Form.

117. Verfahren nach Anspruch 116, wobei die Änderung der Form die Änderung von mindestens einem Punkt von einer Ausrichtung, einer Größe oder einer Position der Form umfasst.

118. Vorrichtung nach einem der Ansprüche 113–117, wobei:

die Bestimmung der Definition der von dem anderen Grafikelementobjekt bereitgestellten visuellen Darstellung die Bestimmung einer Definition einer Eigenschaft der visuellen Darstellung umfasst, wobei die Animation der visuellen Darstellung ein Verhalten mindestens eines Teils der visuellen Darstellung umfasst, der sich basierend auf den empfangenen Echtzeitdaten dynamisch ändert; und

die Anwendung der Modifikation auf die bestimmte Definition mindestens einen der folgenden Punkte umfasst: Entfernung der Animation von der bestimmten Definition, Ersetzung der Animation durch eine andere Animation oder Änderung des mindestens einen Teils der visuellen Darstellung entsprechend der Animation.

119. Vorrichtung nach einem der Ansprüche 113–118, wobei:

die Bestimmung der Definition der von dem anderen Grafikelementobjekt bereitgestellten visuellen Darstellung die Bestimmung einer Definition eines Ereignishandlers umfasst, wobei der Ereignishandler ein

bei Auftreten eines Auslösers auszuführendes Verhalten angibt; und
die Anwendung der Modifikation auf die bestimmte Definition mindestens einen der folgenden Punkte umfasst: Entfernung des Ereignishandlers, Änderung des von dem bestimmten Grafikelement bei Auftreten des Auslösers auszuführenden Verhaltens oder Änderung des Auslösers, der die Ausführung des Verhaltens veranlasst.

120. Verfahren nach Anspruch 119, wobei der Auslöser auf den empfangenen Echtzeitdaten basiert.

121. Verfahren nach einem der Ansprüche 113–120, wobei die Anwendung der Modifikation auf die bestimmte Definition das Hinzufügen von mindestens einem Punkt von einer Eigenschaft, einer Animation oder einem Ereignishandler umfasst.

122. System zur Unterstützung flexibler Grafikelemente in einer Prozessanlage, wobei das System Folgendes umfasst:

eine oder mehrere in einer Echtzeitbetriebsumgebung der Prozessanlage erzeugte Anzeigenansichten, wobei die eine oder mehreren Anzeigenansichten Informationen entsprechend anhand von Steuerung eines Prozesses in der Echtzeitdatenbetriebsumgebung der Prozessanlage erzeugten Echtzeitdaten darstellen; und

eine Vielzahl von auf der einen oder den mehreren Anzeigenansichten dargestellten Grafikelementen, wobei:

jedes Grafikelement der Vielzahl von Grafikelementen eine visuelle Darstellung einer in der Prozessanlage enthaltenen jeweiligen Prozessentität umfasst, jedes Grafikelement durch eine Ausführung der Instantiierung eines jeweiligen Grafikelementobjekts erzeugt wird,

das jeweilige Grafikelementobjekt mit einem jeweiligen Anzeigenobjekt entsprechend jeder Anzeigenansicht verbunden ist, worauf jedes Grafikelement dargestellt wird, und

jedes jeweilige Grafikelementobjekt in der Echtzeitbetriebsumgebung der Prozessanlage konfigurierbar ist, und in einer Konfigurationsumgebung der Prozessanlage konfigurierbar ist,

wobei das System zur Durchführung des Verfahrens nach Null oder mehreren der vorangegangenen Ansprüche konfiguriert ist.

123. System nach Anspruch 122, wobei das jeweilige Anzeigenobjekt ein bestimmtes Grafikelementobjekt ist.

124. System nach Anspruch 122 oder 123, wobei ein erstes Grafikelementobjekt von einem zweiten Grafikelementobjekt abgeleitet ist.

125. System nach Anspruch 124, welches weiterhin eine Modifikation an dem ersten Grafikelementobjekt umfasst.

126. System nach Anspruch 125, wobei die Modifikation an dem ersten Grafikelementobjekt von dem zweiten Grafikelementobjekt weitergegeben wird.

127. System nach Anspruch 125, wobei die Modifikation an dem ersten Grafikelementobjekt nicht von dem zweiten Grafikelementobjekt weitergegeben wird.

128. System nach einem der Ansprüche 125–127, wobei die Modifikation an dem ersten Grafikelementobjekt auf einer Benutzereingabe basiert.

129. System nach einem der Ansprüche 125, 127 oder 128, wobei die Modifikation an dem ersten Grafikelementobjekt von einem anderen Grafikelementobjekt weitergegeben wird, welches sich von dem zweiten Grafikelementobjekt unterscheidet.

130. System nach einem der Ansprüche 125–129, wobei die Modifikation an dem ersten Grafikelementobjekt eine Außerkraftsetzung einer von dem zweiten Grafikelementobjekt bereitgestellten Definition ist.

131. System nach Anspruch 130, wobei die Außerkraftsetzung der Definition mindestens einen Punkt von einer Entfernung mindestens eines Teils der Definition, eines Austauschs mindestens eines Teils der Definition, einer Änderung an mindestens einem Teil der Definition oder eine Hinzufügung zu der Definition umfasst.

132. System nach Anspruch 130 oder Anspruch 131, wobei die Instantiierung des ersten Grafikelementobjekts (i) eine Bestimmung der von dem zweiten Grafikelementobjekt bereitgestellten Definition und (ii) eine Anwendung der Außerkraftsetzung auf die bestimmte Definition umfasst.

133. System nach einem der Ansprüche 130–132, welches Folgendes umfasst:

Empfang einer Anfrage durch eine Benutzerschnittstelle nach einer Identifikation eines Satzes von Außerkraftsetzungen entsprechend dem ersten Grafikelementobjekt, wobei der Satz von Außerkraftsetzungen die Außerkraftsetzung der von dem zweiten Grafikelementobjekt bereitgestellten Definition umfasst; und

einen Mechanismus zur Bestimmung des Satzes von Außerkraftsetzungen und zur Bereitstellung einer Antwort auf die Anfrage.

134. System nach einem der Ansprüche 130–133, wobei die Definition mindestens einen Punkt von einer Form, einer Eigenschaft, einer Animation oder ei-

nem Ereignishandler mindestens eines Teils der visuellen Darstellung der Prozessentität definiert.

135. System nach einem der Ansprüche 124–134, welches Folgendes umfasst:

Empfang einer Anfrage durch eine Benutzerschnittstelle nach einer Identifikation eines Satzes von Grafikelementobjekten, der von dem zweiten Grafikelementobjekt abgeleitet ist, wobei der Satz von Grafikelementobjekten das erste Grafikelementobjekt umfasst; und

einen Mechanismus zur Bestimmung des Satzes von Grafikelementobjekten, der von dem zweiten Grafikelementobjekt abgeleitet ist, und zur Bereitstellung einer Antwort auf die Anfrage.

136. System nach einem der Ansprüche 124–135, wobei das erste Grafikelementobjekt von mehreren Grafikelementobjekten abgeleitet ist, wobei die mehreren Grafikelementobjekte das zweite Grafikelementobjekt umfassen.

137. System nach einem der Ansprüche 124–136, wobei ein drittes Grafikelementobjekt von dem zweiten Grafikelementobjekt abgeleitet ist.

138. System nach einem der Ansprüche 124–137, wobei das zweite Grafikelementobjekt von einem vierten Grafikelementobjekt abgeleitet ist.

139. System nach einem der Ansprüche 124–138, wobei ein fünftes Grafikelementobjekt von dem ersten Grafikelementobjekt abgeleitet ist.

140. System nach einem der Ansprüche 122–139, wobei ein erstes Grafikelementobjekt eine Kombination von mehr als einem anderen Grafikelementobjekt umfasst.

141. System nach einem der Ansprüche 122–140, welches weiterhin eine Datenspeicherungsentität umfasst, die zur Speicherung der Vielzahl von Grafikelementobjekten konfiguriert ist, wobei die Datenspeicherungsentität für die Laufzeitbetriebsumgebung der Prozessanlage und die Konfigurationsumgebung der Prozessanlage zugänglich ist.

142. System nach einem der Ansprüche 122–141, wobei die Prozessentität mindestens einen von folgenden Punkten umfasst:

von einem Klassenmodul, einer Instanz des Klassenmoduls, einem Prozesselementmodul, einem Bereich, einer Einheit, einem Teil einer Ausrüstung, einem Steuerungsmodul, einem Routingmodul oder einem Anzeigenmodul.

143. Einer der vorangegangenen Ansprüche in Kombination mit einem oder mehreren von einem beliebigen anderen der vorangegangenen Ansprüche.

Es folgen 13 Seiten Zeichnungen

Anhängende Zeichnungen

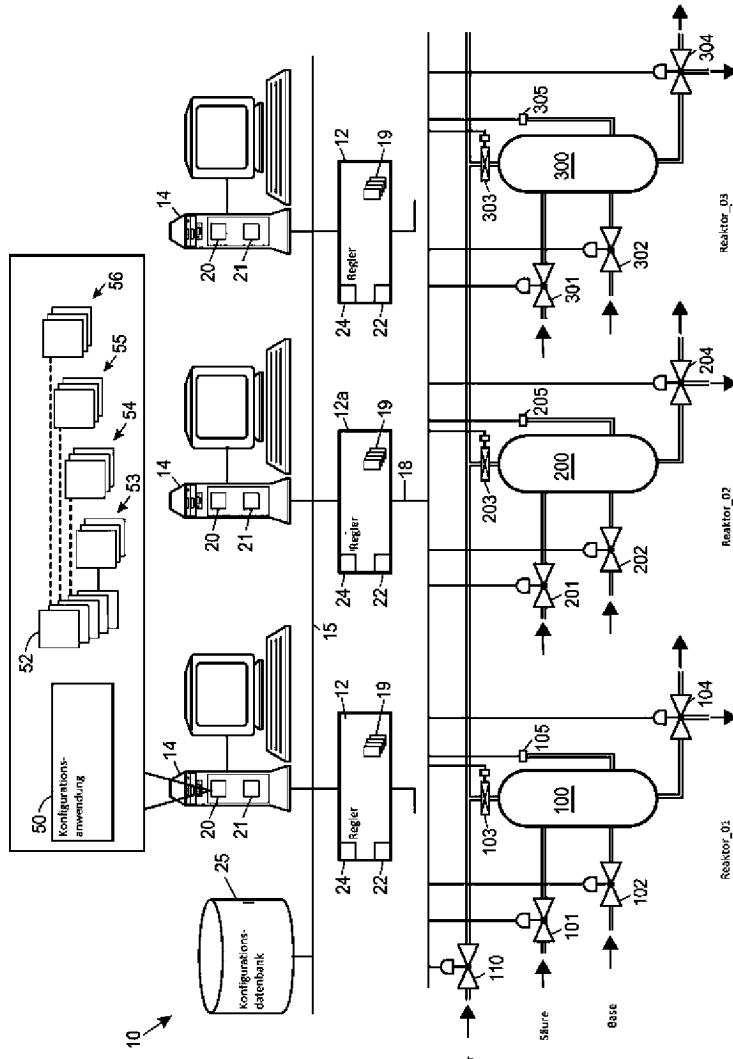


FIG. 1

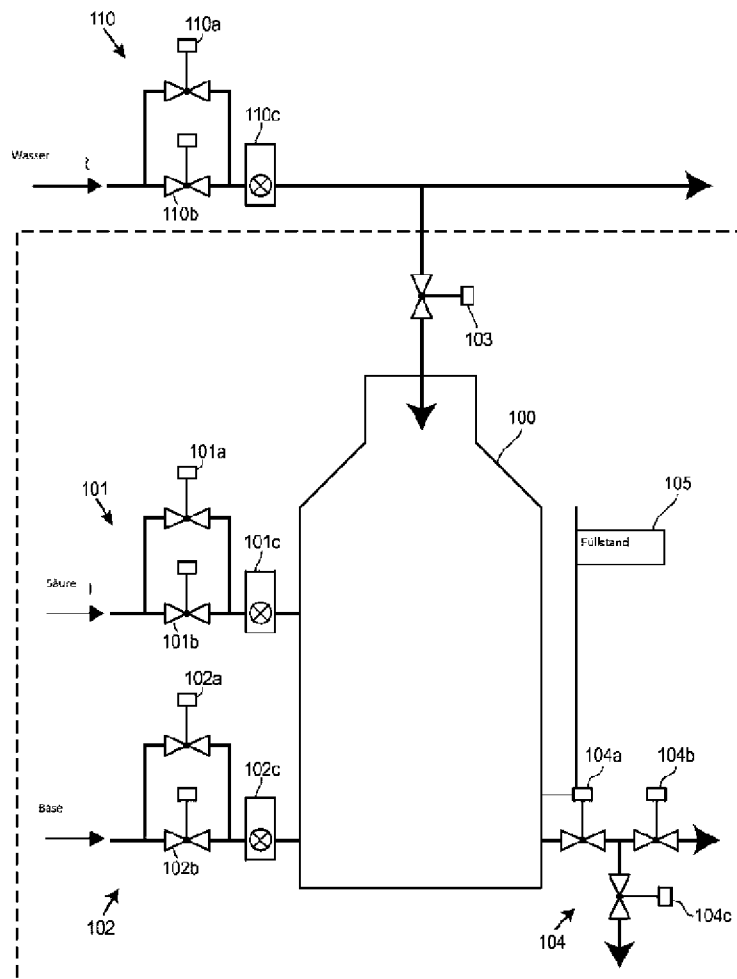


FIG. 2

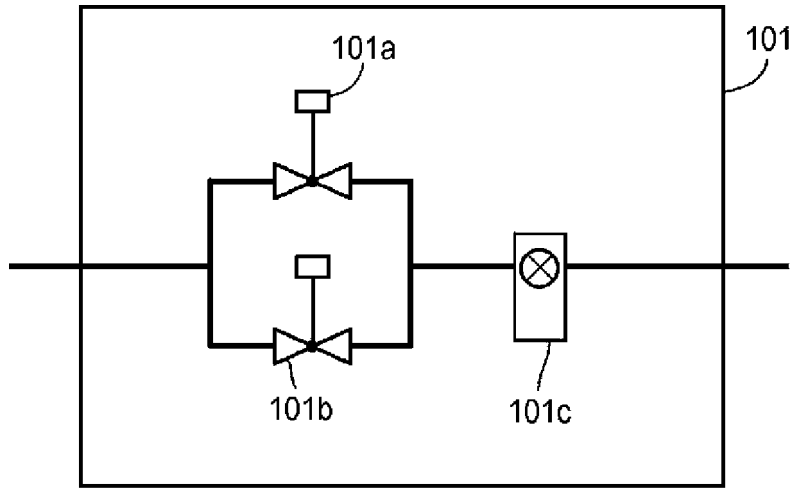


FIG. 3

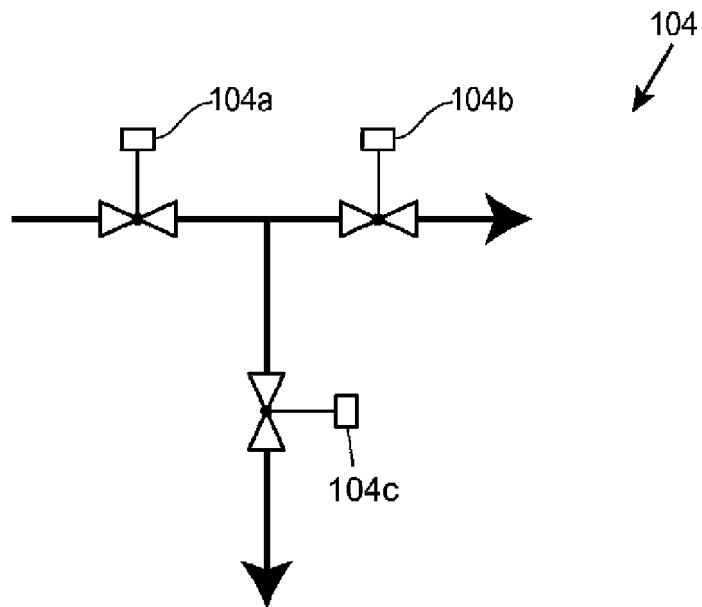


FIG. 4

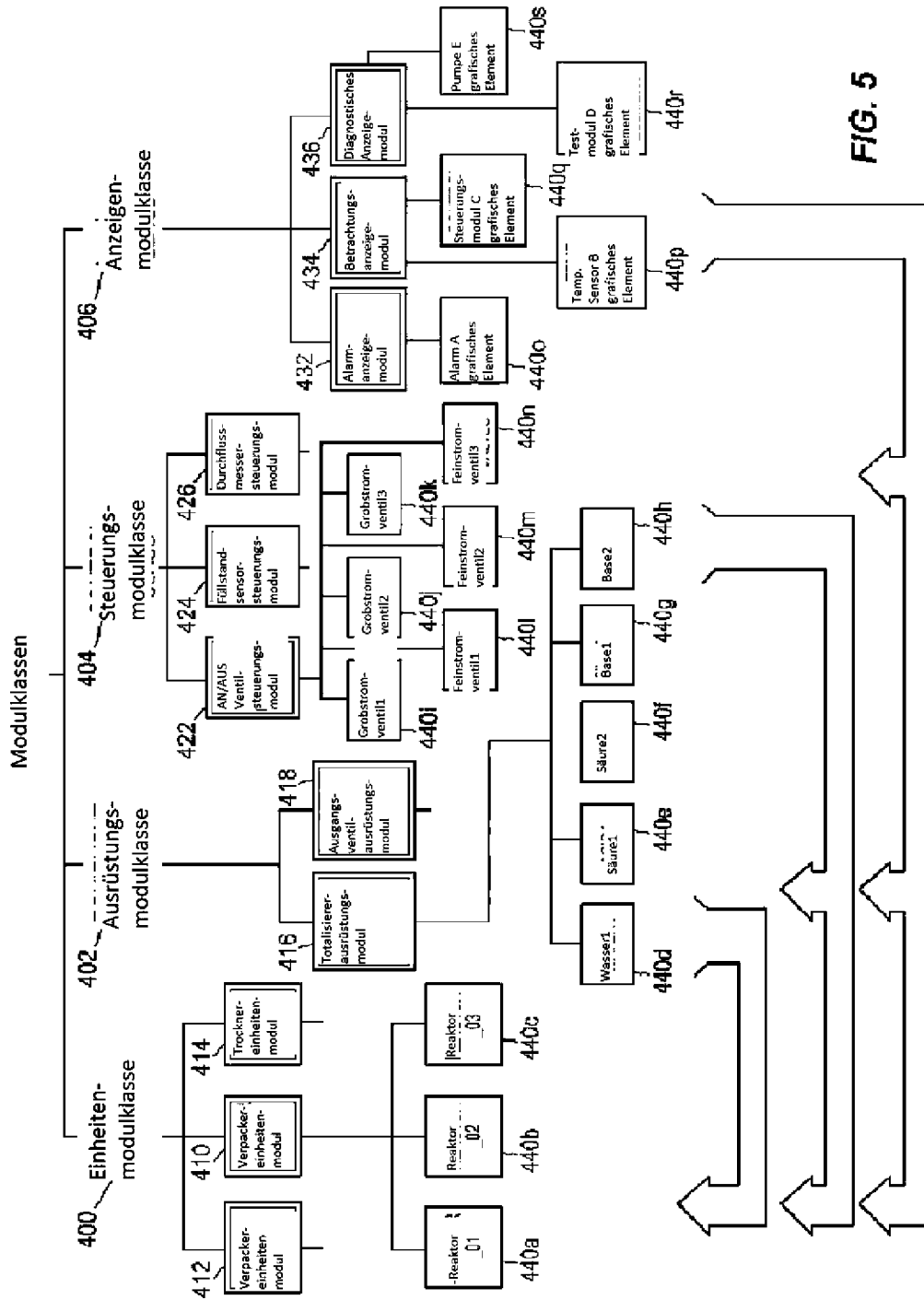


FIG. 5

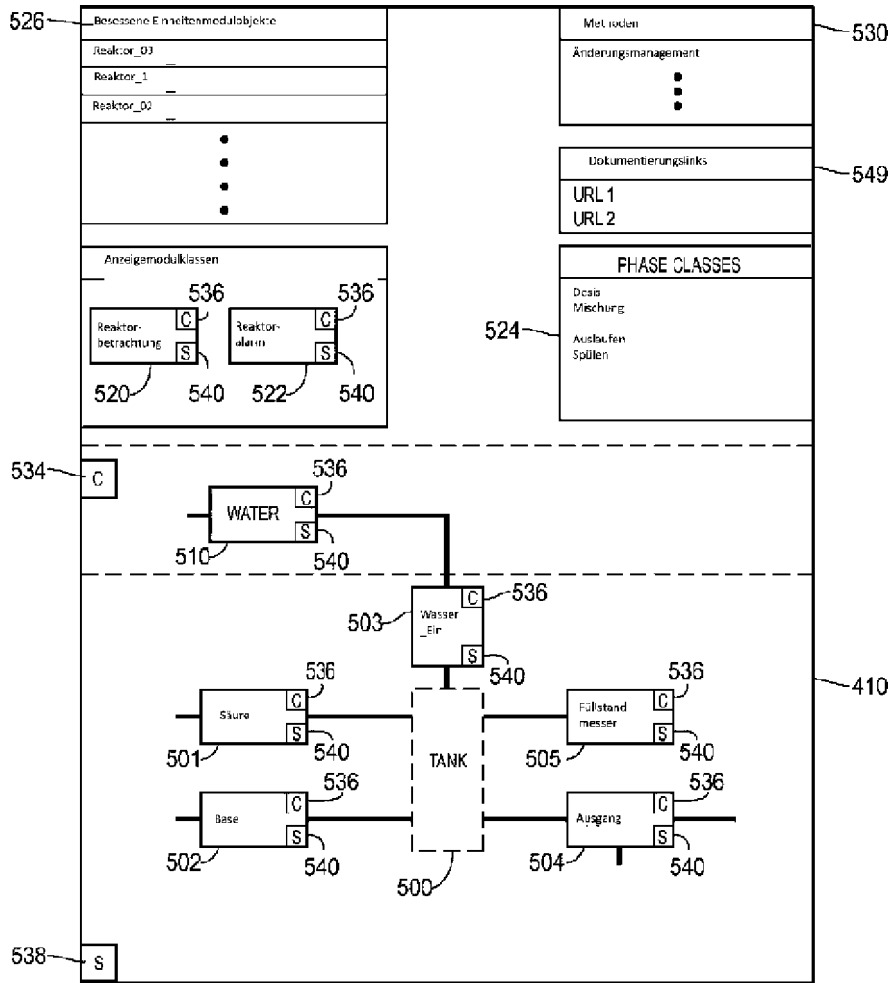


FIG. 6

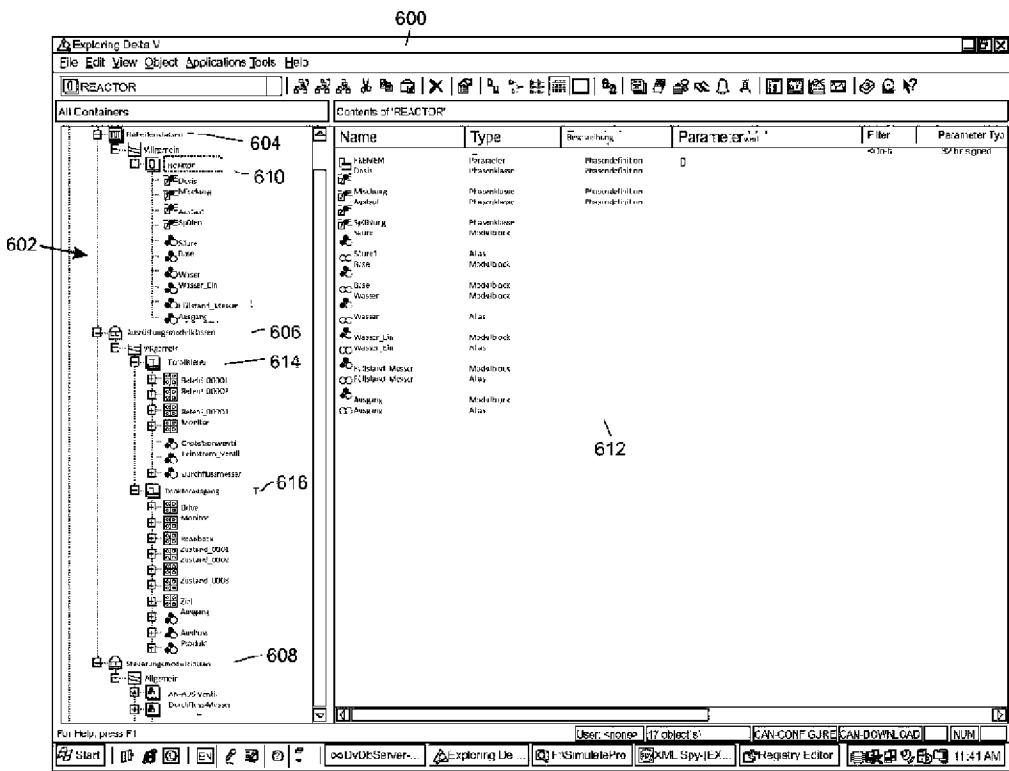


FIG. 7

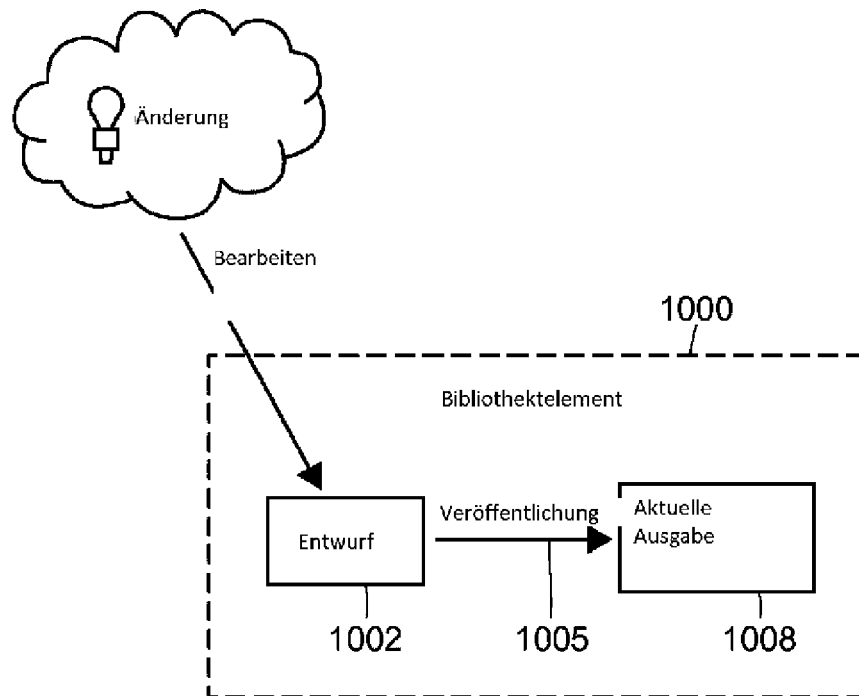


FIG. 8

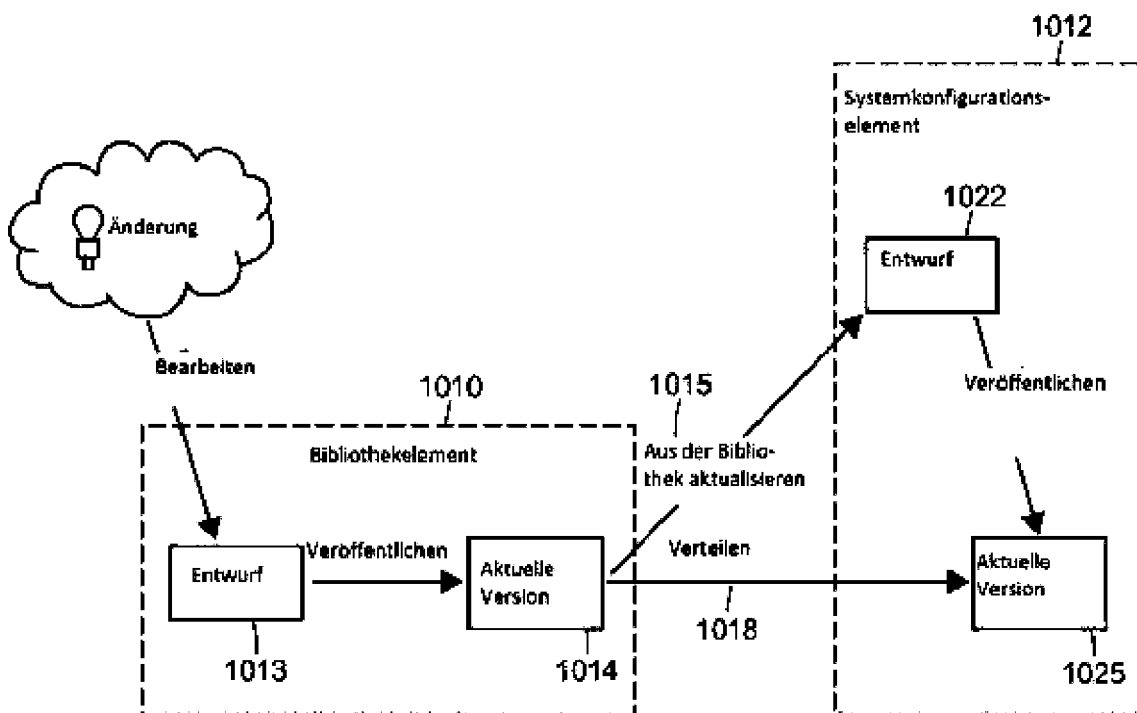


FIG. 9

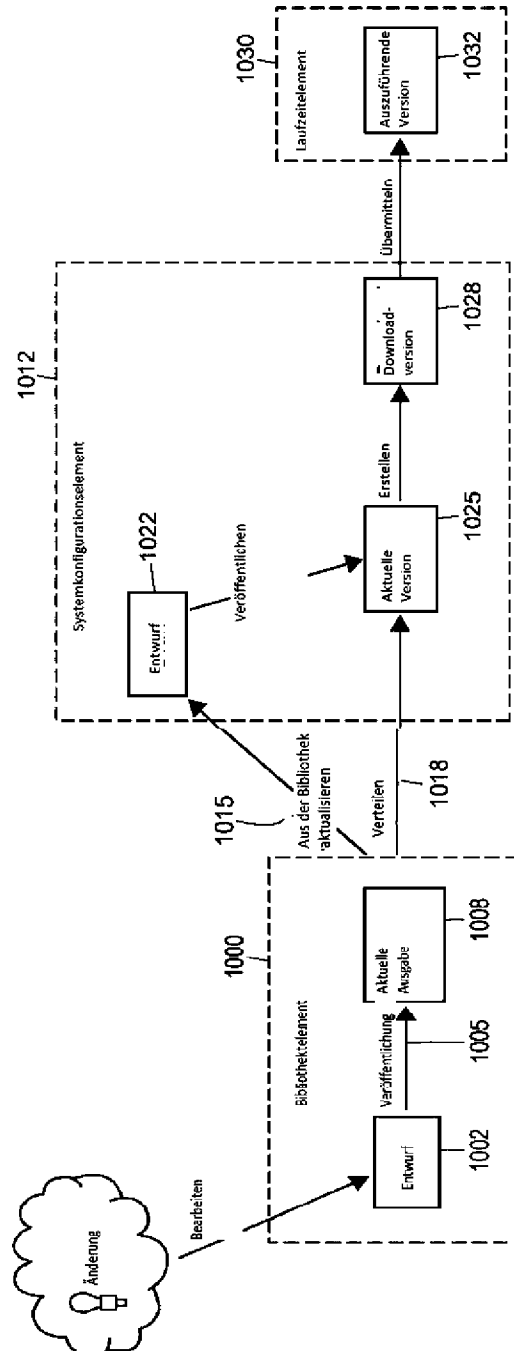


FIG. 10

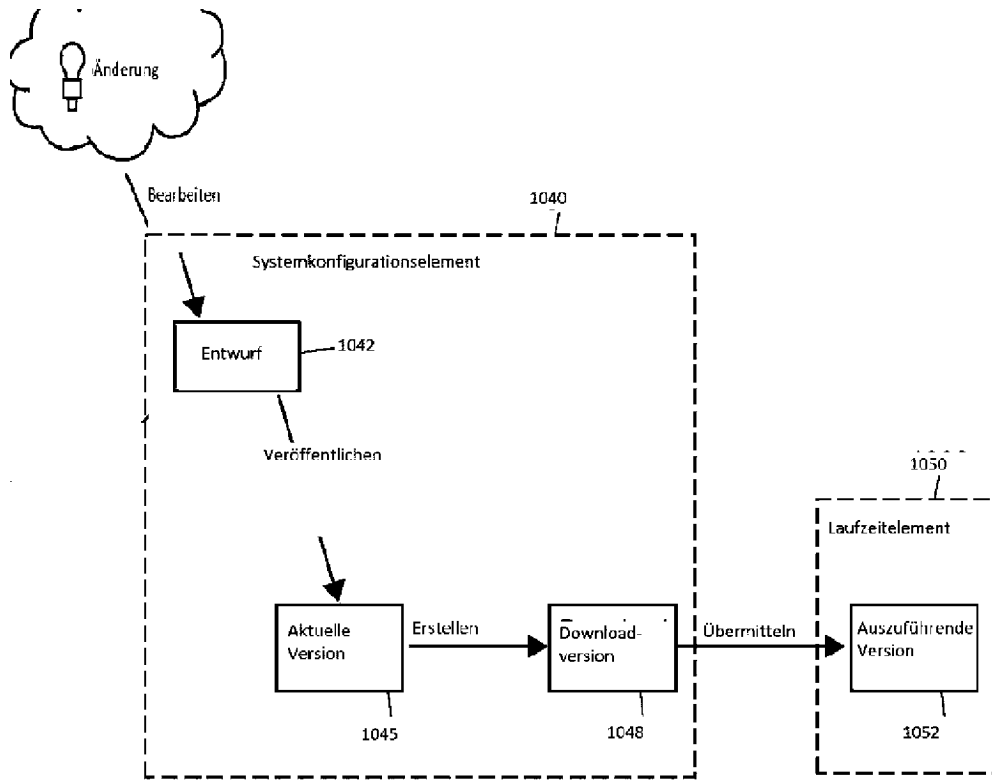


FIG. 11

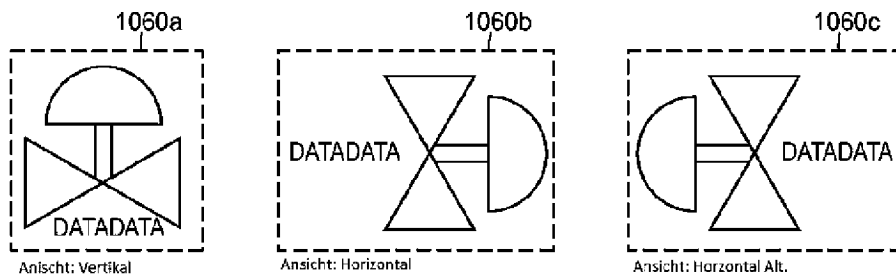


FIG. 12

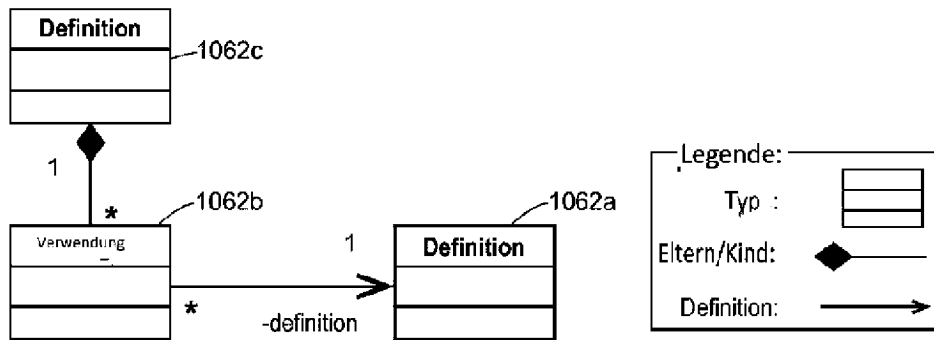


FIG. 13

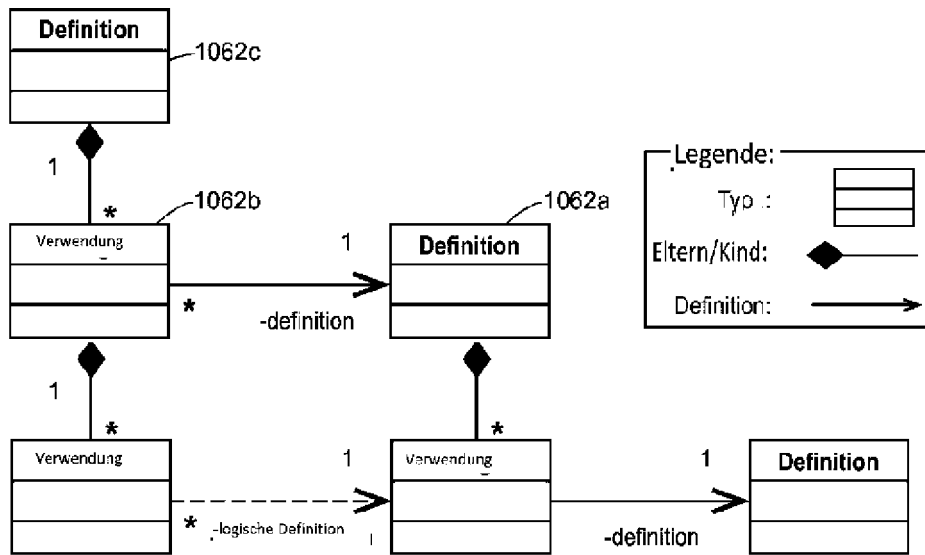


FIG. 14

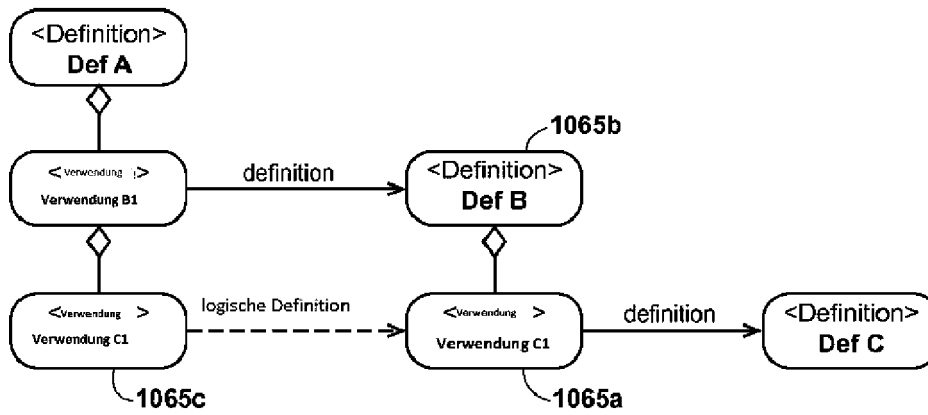


FIG. 15

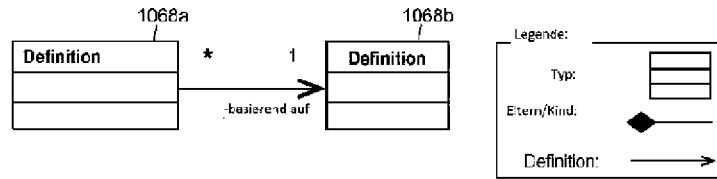


FIG. 16

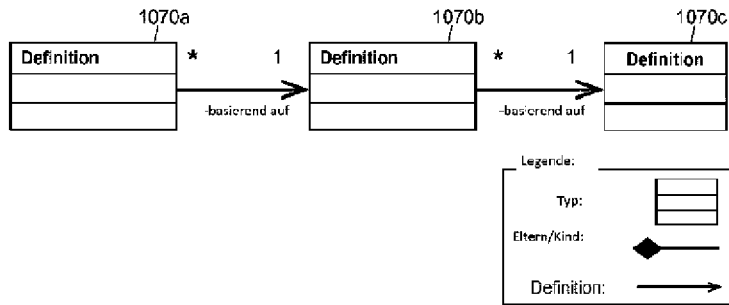


FIG. 17

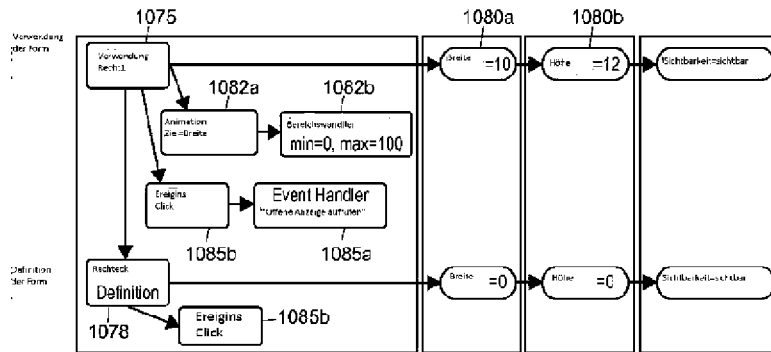


FIG. 18

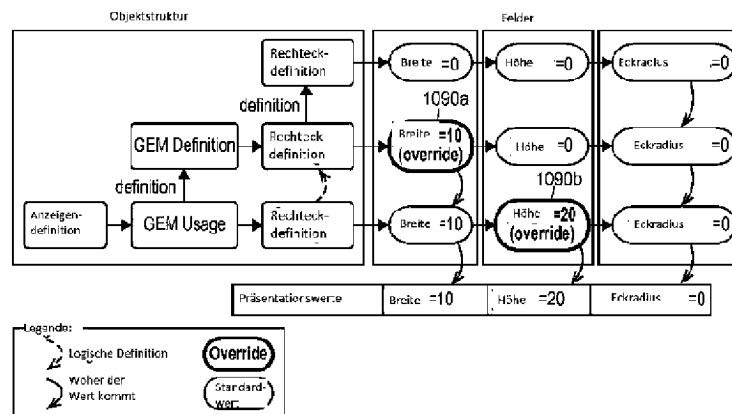


FIG. 19

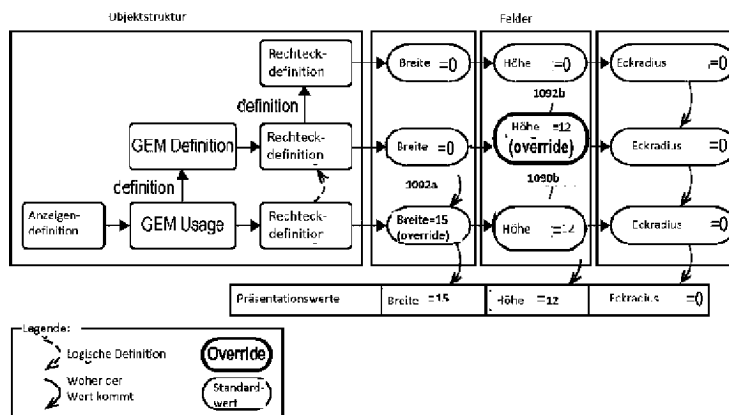


FIG. 20

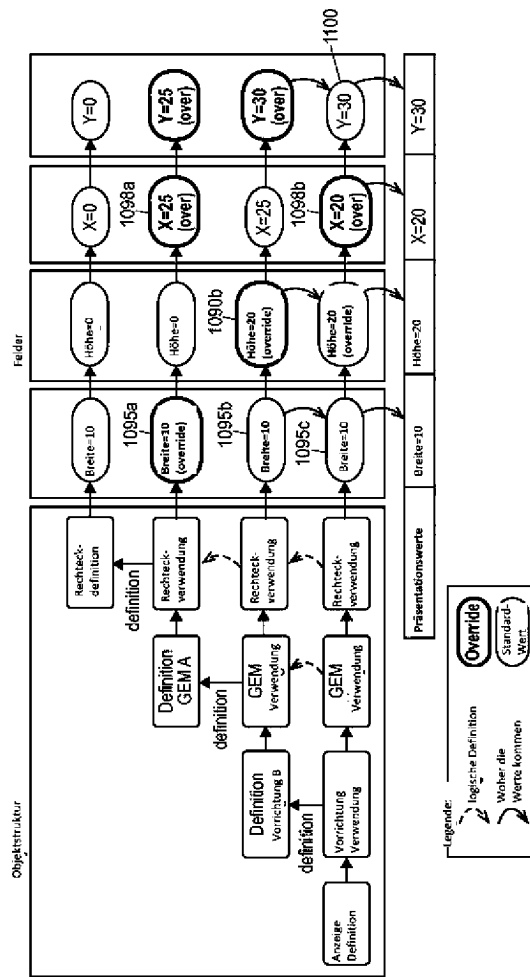


FIG. 21