



(19)  
**Bundesrepublik Deutschland**  
**Deutsches Patent- und Markenamt**

(10) **DE 11 2005 001 045 T5 2007.04.19**

(12)

## Veröffentlichung

der internationalen Anmeldung mit der  
 (87) Veröffentlichungs-Nr.: **WO 2005/109130**  
 in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)  
 (21) Deutsches Aktenzeichen: **11 2005 001 045.7**  
 (86) PCT-Aktenzeichen: **PCT/US2005/015942**  
 (86) PCT-Anmeldetag: **04.05.2005**  
 (87) PCT-Veröffentlichungstag: **17.11.2005**  
 (43) Veröffentlichungstag der PCT Anmeldung  
 in deutscher Übersetzung: **19.04.2007**

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

(30) Unionspriorität:  
**60/567,980      04.05.2004      US**

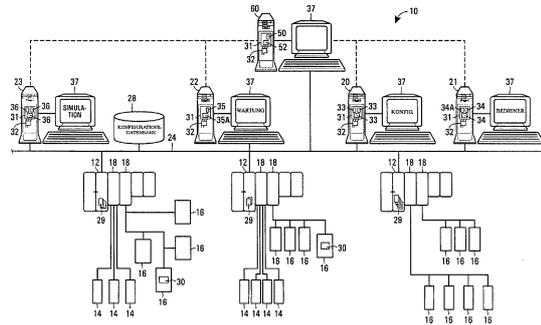
(71) Anmelder:  
**Fisher-Rosemount Systems, Inc., Austin, Tex., US**

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

(72) Erfinder:  
**Gilbert, Stephen, Austin, Tex., US; Hao, Tennyson, Quezon, PH; Guzman, Francis de, , ZZ; Beoughter, Ken J., Round Rock, Tex., US; Campney, Bruce, Manor, Tex., US; Nixon, Mark J., Round Rock, Tex., US**

(54) Bezeichnung: **Skriptgrafik in einem Prozessumfeld**

(57) Hauptanspruch: Anzeigeinstanz zur Verwendung bei einer Darstellung einer optischen Darstellung einer Prozessinstanz einer Prozessanlage für einen Benutzer auf einer Anzeigevorrichtung, mit folgendem:  
 einem computerlesbaren Speicher; und  
 einem im computerlesbaren Speicher gespeicherten Anzeigeobjekt zur Ausführung auf einem Prozessor, wobei das Anzeigeobjekt folgendes enthält:  
 einen Eigenschaftsspeicher zur Speicherung eines Wertes einer der Prozessinstanz zugeordneten Eigenschaft;  
 einer Grafikdarstellung der Prozessinstanz zur Anzeige auf einer Anzeigevorrichtung für einen Benutzer, wenn das Anzeigeobjekt auf einem Prozessor ausgeführt wird; und  
 einer Routine, die in Verbindung mit der für den Benutzer dargestellten grafischen Darstellung der Prozessinstanz arbeitet und dem Wert der Eigenschaft zugeordnet ist.



**Beschreibung**

## Stand der Technik

**[0001]** Die vorliegende Anmeldung ist eine Vorschrift gemäß eingereichter Anmeldung und Ansprüche für Prioritätszwecke zugunsten der vorläufigen US-Anmeldung Serien-Nr. 60/567,980 mit dem Titel "Graphical User Interface for Representing, Monitoring, and Interacting with Process Control Systems" (Grafische Benutzeroberfläche zur Darstellung, Überwachung und Dialogführung mit Prozeßsteuerungssystemen), die am 4. Mai 2004 eingereicht wurde und die von der vorliegenden Anmeldung ausdrücklich durch Bezugnahme in ihrer Gesamtheit aufgenommen wird. Auch bezieht sich die vorliegende Anmeldung auf US-Patentanmeldung Serien-Nr. 10/625,481 mit dem Titel "Integration of Graphic Display Elements, Process Modules and Control Modules in Process Plants" (Integrierung von grafischen Anzeigeelementen, Prozeßmodulen und Steuerungsmodulen in Prozeßanlagen), die am 21. Juli 2003 eingereicht wurde und die als US-Veröffentlichung Nr. 2004/0153804 am 5. August 2004 veröffentlicht wurde und die wiederum eine Teilfortführung der US-Patentanmeldung Serien-Nr. 10/278,469 mit dem Titel "Smart Process Modules and Objects in Process Plants" (Intelligente Prozeßmodulen und Objekte in Prozeßanlagen), die am 22. Oktober 2002 eingereicht wurde und die als US-Veröffentlichung Nr. 2004/0075689 am 22. April 2004 veröffentlicht wurde, deren gesamte Offenbarungen hiermit ausdrücklich durch Bezugnahme darauf in ihrer Gesamtheit aufgenommen werden. Auch bezieht sich die vorliegende Anmeldung auf US-Patentanmeldung Serien-Nr. 10/368,151 mit dem Titel "Module Class Objects in a Process Plant Configuration System" (Modulklassenobjekte in einem Prozeßanlagen-Konfigurationssystem), die am 18. Februar 2003 eingereicht wurde und die als US-Veröffentlichung Nr. 2004/0199925 am 7. Oktober 2004 veröffentlicht wurde, deren gesamte Offenbarung hiermit ausdrücklich durch Bezugnahme darauf in ihrer Gesamtheit aufgenommen wird. Auch bezieht sich die vorliegende Erfindung auf die folgenden Patenanmeldungen, die zum gleichen Datum wie die vorliegende Anmeldung als internationale (PCT-)Anmeldungen eingereicht werden und die die vorliegende Anmeldung ausdrücklich durch Bezugnahme darauf in ihrer Gesamtheit aufnimmt: "Associated Graphic Displays in a Process Environment" (Zusammengehörige Grafikanzeigen in einer Prozeßumgebung) (Aktenzeichen Nr. 06005/41111); "User Configurable Alarms and Alarm Trending for Process Control Systems" (Vom Benutzer konfigurierbare Alarme und Alarm-Trendverfolgung für Prozeßsteuerungssysteme) (Aktenzeichen Nr. 06005/41112); "Integration of Process Modules and Expert Systems in Process Plants" (Integrierung von Prozeßmodulen und Expertensystemen in Prozeßanlagen) (Aktenzeichen Nr. 06005/41113); "A

Process Plant User Interface System Having Customized Process Graphic Display Layers in an Integrated Environment" (Ein Prozeßanlagen-Benutzeroberflächensystem mit kundenspezifischen Prozeß-Grafikanzeigeschichten in einer integrierten Umgebung) (06005/41114); "Graphics Integration into a Process Configuration and Control Environment" (Grafikintegration in eine Prozeßkonfiguration und Steuerungsumgebung) (Aktenzeichen Nr. 06005/41116); "Graphic Element with Multiple Visualizations in a Process Environment" (Grafikelement mit Mehrfachvisualisierungen in einer Prozeßumgebung) (Aktenzeichen Nr. 06005/41117); "System for Configuring Graphic Display Elements and Process Modules in Process Plants" (System zum Konfigurieren von Grafikanzeigeelementen und Prozeßmodulen in Prozeßanlagen) (Aktenzeichen Nr. 06005/41118); "Graphic Display Configuration Framework for Unified Process Control System Interface" (Grafikanzeigen-Konfigurationsrahmen für vereinigte Prozeßsteuerungssystemschnittstelle) (Aktenzeichen Nr. 06005/41124); "Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface" (Auf Textauszeichnungssprache basierende dynamische Prozeßgrafiken in einer Prozeßanlagen-Benutzeroberfläche) (Aktenzeichen Nr. 06005/41127); "Methods and Apparatus for Modifying Process Control Data" (Methoden und Einrichtung zur Abänderung von Prozeßsteuerungsdaten) (Aktenzeichen Nr. 06005/591622 und 20040/59-11622); "Methods and Apparatus for Accessing Process Control Data" (Verfahren und Einrichtung zum Zugreifen auf Prozeßsteuerungsdaten) (Aktenzeichen Nr. 06005/591623 und 20040/59-11623); "Integrated Graphical Runtime Interface for Process Control Systems" (Integrierte grafische Laufzeitschnittstelle für Prozeßsteuerungssysteme) (Aktenzeichen Nr. 06005/591628 und 20040/59-11628); "Service-Oriented Architecture for Process Control Systems" (Dienstorientierte Architektur für Prozeßsteuerungssysteme) (Aktenzeichen Nr. 06005/591629 und 20040/59-11629).

## Technisches Gebiet

**[0002]** Die vorliegende Erfindung bezieht sich allgemein auf Prozeßanlagen und insbesondere auf die Integrierung und Verwendung eines Grafikanzeigeditors und grafische Anzeigenobjekte auf Systemebene eines Prozeßsteuerungs- und Simulationssystems zum Ermöglichen der Erstellung und Verwendung von gemeinsamen Grafikanzeigeelementen in verschiedenen, Anlagenkonfiguration, -steuerung, -wartung, und -simulation zugeordneten Aktivitäten.

## Beschreibung des Standes der Technik

**[0003]** Verteilte Prozeßsteuerungssysteme wie die in chemischen, Petroleum- oder sonstigen Prozessen benutzten enthalten typischerweise eine oder mehrere Prozeßsteuerungen, die kommunikativ an eine

oder mehrere Betriebsvorrichtungen über analoge oder digitale oder kombiniert analoge und digitale Busse angekoppelt sind. Die Betriebsvorrichtungen, die beispielsweise Ventile, Ventilstellvorrichtungen, Schalter und Sender (z. B. Temperatur-, Druck-, Füllgrad- und Durchflusssensoren) sein können, befinden sich in der Prozeßumgebung und führen Prozeßfunktionen wie beispielsweise das Öffnen und Schließen von Ventilen, das Messen von Prozeßparametern usw. durch. Intelligente Betriebsvorrichtungen, wie beispielsweise die den wohlbekannten Fieldbus-Protokollen wie dem FOUNDATION™-Fieldbus-Protokoll entsprechenden Betriebsvorrichtungen können auch Steuerungsberechnungen, Alarmierungsfunktionen und sonstige Steuerungsfunktionen durchführen, die gewöhnlich in der Steuerung implementiert sind. Die Prozeßsteuerungen, die sich ebenfalls typischerweise in der Anlagenumgebung befinden, empfangen Signale, die durch die Betriebsvorrichtungen getätigte Prozeßmessungen und/oder sonstig Informationen betreffs der Betriebsvorrichtung anzeigen und führen eine Steuerungsanwendung aus, die beispielsweise verschiedene Steuerungsmodulen ansteuert, die Prozeßsteuerungsentscheidungen treffen, auf den empfangenen Informationen basierende Steuerungssignale erzeugen und mit den in den Betriebsvorrichtungen, beispielsweise HART- und Fieldbus-Betriebsvorrichtungen, ausgeführten Steuerungsmodulen oder Blöcken koordinieren. Die Steuerungsmodulen in der Steuerung senden die Steuerungssignale über die Kommunikationsleitungen zu den Betriebsvorrichtungen, um dadurch die Funktionsweise des Prozesses zu steuern.

**[0004]** Informationen von den Betriebsvorrichtungen und der Steuerung werden gewöhnlich einer oder mehreren anderen Hardwarevorrichtungen wie beispielsweise Bedienerarbeitsplätzen, Personalcomputern, Datengeschichtsschreibern, Reportgeneratoren, zentralen Datenbanken, usw., die sich typischerweise in Steuerwarten oder an anderen Stellen entfernt von der rauheren Anlagenumgebung befinden, über einen Datenweg zur Verfügung gestellt. Auf diesen Hardwarevorrichtungen laufen Anwendungen ab, die beispielsweise einen Bediener dazu befähigen, Funktionen bezüglich des Prozesses durchzuführen, wie beispielsweise Ändern der Einstellungen der Prozeßsteuerungsroutine, Abändern der Funktionsweise der Steuerungsmodulen in der Steuerung oder den Betriebsvorrichtungen, Betrachten des aktuellen Zustandes des Verfahrens, Betrachten von durch Betriebsvorrichtungen und Steuerungen erzeugten Alarmen, Simulieren der Funktionsweise des Prozesses zwecks Personaltraining oder Prüfen der Prozeßsteuerungssoftware, Unterhalten und Aktualisieren einer Konfigurationsdatenbank usw.

**[0005]** Als Beispiel enthält das von Emerson Process Management vertriebene Steuerungssystem

DeltaV™ darin gespeicherte und von verschiedenen Vorrichtungen, die sich an verschiedenen Plätzen in einer Prozeßanlage befinden, ausgeführte mehrfache Anwendungen. Eine Konfigurationsanwendung, die in einem oder mehreren Bedienerarbeitsplätzen resident ist, ermöglicht den Benutzern, Prozeßsteuerungsmodulen zu erstellen oder zu ändern und diese Prozeßsteuerungsmodulen über einen Datenweg zu fest zugeordneten verteilten Steuerungen herunterzuladen. Typischerweise bestehen diese Steuerungsmodulen aus kommunikativ miteinander verbundenen Funktionsblöcken, die Objekte in einem objektorientierten Programmierungsprotokoll sind, die Funktionen in dem Steuerungsschema auf Grundlage von Eingaben darin durchführen und Ausgaben für andere Funktionsblöcke im Steuerungsschema bereitstellen. Auch kann die Konfigurationsanwendung einem Konstrukteur die Erstellung oder Änderung von Bedieneroberflächen erlauben, die durch eine Betrachtungsanwendung zur Anzeige von Daten für einen Bediener benutzt werden und dem Bediener ermöglichen, Einstellungen wie beispielsweise Sollwerte in der Prozeßsteuerungsroutine zu ändern. Von jeder festzugeordneten Steuerung und in manchen Fällen Betriebsvorrichtungen wird eine Steuerungsanwendung gespeichert und ausgeführt, die dieser zugeordnete und zu dieser heruntergeladene Steuerungsmodulen betreibt, um eigentliche Prozeßsteuerungsfunktionalität zu implementieren. Die Betrachtungsanwendungen, die auf einem oder mehreren Bedienerarbeitsplätzen ablaufen können, empfangen Daten von der Steuerungsanwendung über den Datenweg und stellen diese Daten für Prozeßsteuerungssystem-Konstrukteure, Bediener oder Benutzer dar, die die Benutzeroberflächen benutzen, und können eine beliebige einer Anzahl unterschiedlicher Ansicht, wie beispielsweise eine Bedieneransicht, eine Ingenieursansicht, eine Technikeransicht, usw. bereitstellen. Eine Datengeschichtsschreiberanwendung ist typischerweise in einer Datengeschichtsschreibervorrichtung gespeichert und ausgeführt, die einige oder alle der über den Datenweg bereitgestellten Daten einsammelt und speichert, während eine Konfigurationsdatenbankanwendung in einem noch weiteren, am Datenweg angebrachten Computer ablaufen kann, um die aktuelle Prozeßsteuerungsroutinenkonfiguration und damit verbundene Daten zu speichern. Als Alternative kann sich die Konfigurationsdatenbank im selben Arbeitsplatz wie die Konfigurationsanwendung befinden.

**[0006]** Mit steigender Anzahl und Art von in einer Prozeßsteuerungsumgebung benutzten Steuerungs- und Unterstützungsanwendungen sind verschiedene Grafikanzeigenanwendungen bereitgestellt worden, um Benutzern die wirkungsvolle Konfiguration und Benutzung dieser Anwendungen zu ermöglichen. Beispielsweise sind Grafikanzeigenanwendungen zur Unterstützung von Steuerungskonfigurationsanwendungen benutzt worden, um einem Konfigurati-

onsingenieur die grafische Erstellung von zu den Steuerungsvorrichtungen in einer Prozeßanlage herunterzuladenden Steuerungsprogrammen zu ermöglichen. Zusätzlich sind Grafikanzeigenanwendungen dazu benutzt worden, um Steuerungsbedienern die Betrachtung der aktuellen Funktionsweise der Prozeßanlage oder von Bereichen der Prozeßanlage zu ermöglichen, um Wartungspersonal die Betrachtung des Zustandes von Hardwarevorrichtung in der Prozeßanlage zu ermöglichen, um Simulationen der Prozeßanlage zu ermöglichen, usw. Diese Grafikanzeigenanwendungen sind in der Vergangenheit jedoch als Teil von oder zur Unterstützung der bestimmten Anwendungen erstellt worden, denen sie zugeordnet sind, und sind daher im allgemeinen in ihrer Nützlichkeit auf die bestimmte Prozeßfunktion begrenzt, für die sie erstellt wurden. Beispielsweise ist es schwierig, wenn nicht unmöglich, ein zur Unterstützung eines Steuerungs- oder sonstigen Bedieners erstelltes Grafikprogramm in einer Wartungs-, Konfigurations- oder Simulationsfunktion zu benutzen.

**[0007]** Als bestimmtes Beispiel enthalten einige Prozeßsteuerungskonfigurationsanwendungen gegenwärtig eine Bibliothek von Schablonenobjekten wie beispielsweise Funktionsblock-Schablonenobjekte und in manchen Fällen Steuerungsmodul-Schablonenobjekte, die zur Erstellung einer Steuerungsstrategie für eine Prozeßanlage genutzt werden. Mit den Schablonenobjekten sind vorgegebene Eigenschaften, Einstellungen und Verfahren verbunden und der eine Grafikkonfigurationsanwendung benutzende Ingenieur kann sich diese Schablonenobjekte aussuchen und im wesentlichen Kopien der ausgewählten Schablonenobjekte in ein Konfigurationsschirmbild plazieren, um ein Steuerungsmodul zu entwickeln. Während des Verfahrens der Auswahl und des Plazierens der Schablonenobjekte in das Konfigurationsschirmbild verbindet der Ingenieur die Eingaben und Ausgaben dieser Objekte miteinander und ändert ihre Parameter, Namen, Bezeichnungen oder sonstigen Eigenschaften, um ein bestimmtes Steuerungsmodul für eine bestimmte Verwendung in der Prozeßanlage zu erstellen. Nach Erstellung von einem oder mehreren derartigen Steuerungsmodulen kann der Ingenieur dann das Steuerungsmodul konkretisieren und es zu der entsprechenden Steuerung oder den entsprechenden Steuerungen und Betriebsvorrichtungen zur Ausführung während des Betriebs der Prozeßanlage herunterladen.

**[0008]** Danach kann der Ingenieur eine andere Grafikanzeigenerstellungsanwendung benutzen, um eine oder mehrere Anzeigen für Bediener, Wartungspersonal usw. in der Prozeßanlage durch Auswählen und Aufbauen von Anzeigeobjekten in der Anzeigenerstellungsanwendung zu erstellen. Diese Anzeigen sind typischerweise auf systemweiter Basis in einem oder mehreren der Arbeitsplätze implementiert und stellen vorkonfigurierte Anzeigen für den Bediener

oder die Wartungspersonen hinsichtlich des Betriebszustandes des Steuerungssystems oder der Vorrichtungen in der Anlage bereit. Diese Anzeigen nehmen im allgemeinen die Form von Alarmierungsanzeigen an, die von Steuerungen oder Vorrichtungen in der Prozeßanlage erzeugte Alarme empfangen und anzeigen, den Betriebszustand der Steuerung und anderer Vorrichtungen in der Prozeßanlage anzeigenden Steuerungsanzeigen, den Funktionszustand der Vorrichtung in der Prozeßanlage anzeigenden Wartungsanzeigen, usw. Diese Anzeigen sind jedoch im allgemeinen zur Anzeige auf bekannte Weisen von den Prozeßsteuerungsmodulen oder den Vorrichtungen in der Prozeßanlage empfangenen Informationen oder Daten vorkonfiguriert. In einigen Systemen werden Anzeigen durch eine grafische Darstellung erstellt, die ein physikalisches oder logisches Element darstellt und die kommunikativ an das physikalische oder logische Element angebunden ist, um Daten über das physikalische oder logische Element zu empfangen. Die Grafik auf dem Anzeigeschirm kann sich als Reaktion auf gewisse Ereignisse wie beispielsweise empfangene Daten ändern, um beispielsweise darzustellen, daß ein Tank halbvoll ist, um den durch einen Flußsensor gemessenen Fluß darzustellen, usw. Für die für Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsaktivitäten benutzten Grafikanzeigen werden jedoch allgemein getrennt voneinander unter Verwendung unterschiedlicher Grafikedatoren erstellt. Weiterhin sind auch die begrenzten grafischen Fähigkeiten dieser Anzeigen schwer zu implementieren und nicht als Teil irgendeines grafischen Objekts ausgeführt.

**[0009]** Ähnlich der Steuerungskonfigurationsanwendung kann die Anzeigeneinstellungsanwendung daher grafische Anzeigenelemente der Schablone, wie beispielsweise Behälter, Ventile, Sensoren, Bedienersteuerknöpfe wie Gleitschienen, Ein/Aus-Schalter usw. aufweisen, die in jeder gewünschten Konfiguration auf einem Bildschirm planiert werden können, um eine Bedieneranzeige, Wartungsanzeige und dergleichen zu erstellen. Wenn sie auf dem Bildschirm planiert werden, können einzelne Grafikelemente auf eine Weise auf dem Bildschirm miteinander verbunden sein, die den Benutzern irgendwelche Informationen oder Darstellungen der inneren Funktionsweise der Prozeßanlage bietet. Um jedoch die Grafikanzeige zu animieren, muß der Ersteller der Anzeige jedes der Grafikelemente manuell an in der Prozeßanlage erzeugte Daten wie beispielsweise durch Sensoren gemessene Daten oder Ventilstellungen anzeigende Daten usw. anbinden, indem er eine Kommunikationsverbindung zwischen dem Grafikelement und der entsprechenden Datenquelle in der Prozeßanlage angibt. Dieses Verfahren ist umständlich, zeitaufwendig und kann fehlerbehaftet sein.

**[0010]** Während die Steuerungsschablonenobjekte

in der Steuerungskonfigurationsanwendung und die Anzeigenelemente in der Anzeigenerstellungsanwendung zweckdienlich sind, da sie kopiert und zur Erstellung vieler unterschiedlicher Steuerungsmodulen und Grafikanzeigen benutzt werden können, besteht häufig ein Bedarf, zahlreiche des gleichen Steuerungsmoduls und der gleichen Grafikanzeige für unterschiedliche Geräte in der Prozeßanlage zu erstellen. Beispielsweise weisen viele mittel- bis großdimensionierte Prozeßanlagen zahlreiche Instanzen des gleichen oder ähnlichen Geräts auf, das unter Verwendung des gleichen allgemeinen Grundsteuerungsmoduls und der gleichen allgemeinen Anzeige gesteuert und betrachtet werden können. Zur Erstellung dieser zahlreichen Steuerungsmodulen und Anzeigen wird jedoch ein allgemeines Steuerungsmodul oder Anzeigenmodul erstellt und dieses allgemeine Steuerungs- oder Anzeigenmodul wird dann für jedes der unterschiedlichen Gerätestücke kopiert, für die es anwendbar ist. Nach dem Kopieren muß natürlich jedes der neuen Steuerungs- oder Anzeigenmodulen manuell in der Konfigurationsanwendung geändert werden, um das bestimmte Gerät anzugeben, dem es zugeordnet ist, und alle diese Steuerungs- und Anzeigenmodulen müssen dann konkretisiert und zum Prozeßsteuerungssystem heruntergeladen werden.

**[0011]** Leider sind die oben besprochenen Steuerungsmodulen und Anzeigeelemente auf keine Weise modular. Nach dem Kopieren muß daher jedes der Steuerungsmodulen und Anzeigen manuell und einzeln unter Verwendung der zutreffenden Konfigurationsanwendung geändert werden, um das Gerät in der Anlage anzugeben, dem sie zuzuordnen sind. In einer Anlage mit vielen Kopien der gleichen Geräteart (d. h. duplizierten Geräten) ist dieses Verfahren umständlich, zeitaufwendig und mit vom Bediener eingeführten Fehlern behaftet. Nach ihrer Programmierung haben diese unterschiedlichen Steuerungsmodulen und Anzeigen weiterhin keine Kenntnis voneinander. Um daher nach ihrer Erstellung eine Änderung an den Steuerungsmodulen durchzuführen, muß der Ingenieur oder Bediener die gleiche Änderung manuell an jedem der unterschiedlichen Steuerungsmodulen für das unterschiedliche duplizierte Gerät durchführen, was wiederum zeitaufwendig und umständlich ist. Das gleiche Problem gilt für die für die unterschiedlichen von duplizierten Geräten in der Anlage erstellten grafischen Darstellungen. Anders gesagt, wenn einmal ein bestimmtes Steuerungsmodul oder eine bestimmte grafische Darstellung (einzeln oder durch Kopieren von einem Schablonenobjekt) erstellt worden ist und dann an einem bestimmten Gerätesatz in der Anlage angebunden worden ist, besteht dieses Steuerungsmodul oder die grafische Darstellung als getrennte Instanz oder getrenntes Objekt im System ohne irgendwelche automatische Kenntnis der anderen Steuerungsmodulen oder grafischen Anzeigen, die ihm oder ihr gleich oder ähnlich sind.

Infolgedessen müssen auf jedes der Steuerungsmodulen und Grafikanzeigen anwendbare Änderungen einer bestimmten Art einzeln an diesen Modulen und Anzeigen durchgeführt werden. Bei der Erstellung von grafischen Darstellungen für das gleiche Gerät aber in unterschiedlichen funktionsmäßigen Zusammenhängen in der Anlage wie beispielsweise für Steuerungsbetrachtung, Wartungsbetrachtung und Simulationsfunktionen wird diese Problem noch deutlicher. In diesem Fall werden die grafischen Ansichten getrennt ohne irgendwelche Kenntnis oder irgendein Bewußtsein voneinander erstellt.

**[0012]** Während daher Grafikanzeigen in und im Zusammenhang mit unterschiedlichen Anwendungen bereitgestellt worden sind, die für verschiedene allgemeine, in einer Prozeßanlage durchgeführten Aktivitäten benutzt werden, wurden diese Grafikanzeigen und zugehörigen Grafikanzeigeneditoren im allgemeinen auf der Funktionsebene der Anwendung zugefügt, für deren Unterstützung sie erstellt worden sind. Infolgedessen haben die Grafikeditoren, sofern es sie gab, dem Benutzer nur die Erstellung von Grafiken ermöglicht, die eine durch eine bestimmte Anwendung benötigte Funktionalität unterstützen. Vorherige Prozeßanlagen stellten keinen Grafikanzeigeneditor bereit, der von den grafischen Bedürfnissen verschiedener oder mehrfacher Aktivitäten benutzt werden konnte oder diese unterstützen konnte, die im Rahmen der Anlagenkonfiguration und -unterstützung durchgeführt werden. So ermöglichte beispielsweise ein zur Unterstützung oder zum Ermöglichen von Steuerungskonfigurationsaktivitäten benutzter Grafikanzeigeneditor dem Benutzer nur, Steuerungsprogramme zu erstellen und unterstützte nicht die Bedürfnisse oder Funktionalität von Bediener- oder Wartungsanzeigen. Auf ähnliche Weise unterstützten zur Erstellung von Bedieneransichten, Wartungsansichten usw. zur Bereitstellung für einen Steuerungsbediener oder Wartungstechniker während des Betriebs einer Anlage benutzte Grafikanzeigeneditoren nicht die mit Konfigurationsaktivitäten, Simulationsaktivitäten usw. verbundene Funktionalität. Dadurch, daß Grafikanzeigenbedürfnisse auf den einzelnen Funktionsebenen der Prozeßanlage wie beispielsweise auf den Funktionsebenen der Steuerungskonfiguration, Wartungsunterstützung, Steuerungsbedienerunterstützung, Simulationsunterstützung unterstützt werden, werden durch unterschiedliche der durch diese verschiedenen Editoren erstellten Anzeigen schließlich die gleichen Komponenten in der Anlage modelliert und dargestellt, mit dem Ergebnis der Duplikation von grafischen Anzeigebemühungen durch verschiedene unterschiedliche Mitarbeiter bei der Prozeßanlage. Diese Duplikation von Bemühungen ist nicht nur bei dem Aufwand offenkundig, der zur Erstellung der unterschiedlichem Grafikanzeigen notwendig ist, die das gleiche Prozeßelement für unterschiedliche Verwendungszwecke darstellen, sondern auch bei dem Aufwand, der zur Anbindung der

in unterschiedlichen Anzeigenanwendungen benutzen Grafikelemente an die eigentlichen Hardware- oder Softwareelemente in der Prozeßanlage, denen sie zugeordnet sind, notwendig ist.

**[0013]** Da die Grafikerunterstützung für verschiedene Prozeßanlagenaktivitäten im Nachhinein und als Teil der eigentlichen durchgeführten Aktivität bereitgestellt worden ist, ist die Grafikerunterstützung nicht auf eine Weise in die Anlagenumgebung integriert, die die Erstellung und Verwendung von gemeinsamen Grafiken in der Anlage auf den verschiedenen unterschiedlichen Funktionsebenen der Anlage ermöglicht. Diese Nichtintegration von Grafiken führt dazu, daß die Grafiken, die eigentlich für die verschiedenen Funktionen erstellt worden sind, von Funktion zu Funktion oder von Anwendung zu Anwendung unterschiedlich sind, was zu Verwirrung bei Benutzern führen kann, die, obwohl sie mit einer bestimmten Art von Grafikanzeige vertraut sind, gelegentlich andere Anzeigen betrachten müssen, die mit anderen Operationen oder Funktionen in der Anlage verbunden sind. GleichermäÙen führt die Bereitstellung von Grafikanzeigenunterstützung auf den verschiedenen unterschiedlichen Funktionsebenen der Anlage wie oben bemerkt zur Duplikation von Grafikerunterstützung sowohl bei der Erstellung von Anzeigen als auch ordnungsgemäÙen Verbindung der Elemente in den Anzeigen mit konkreten Hardware- oder Softwareelementen in der Anlage.

**[0014]** Zusätzlich ist Fehlererkennung und sonstige Programmierung zur Erkennung von Zuständen, Fehlern, Alarmen usw., die mit auf den verschiedenen Steuerungen ablaufenden Steuerungsschleifen verbunden sind, und Problemen in den einzelnen Vorrichtungen nützlich. Eine solche Fehlererkennung ist herkömmlicherweise auf den unterschiedlichen Funktionsebenen der Prozeßanlage durchgeführt und auf Grafikanzeigen angezeigt worden, die für diese unterschiedlichen Funktionsaktivitäten erstellt worden sind. Es ist daher schwierig gewesen, das Anlagensteuerungssystem für die Erkennung von Zuständen oder Fehlern auf Systemebene zu programmieren, die durch die Analyse von Daten von verschiedenen, möglicherweise zerstreut angeordneten Vorrichtungen in der Prozeßanlage erkannt werden müssen, und noch schwieriger, diese Fehlerarten auf Bedieneranzeigen darzustellen, die nicht dafür erstellt worden sind, solche Zustandsinformationen auf Systemebene Bedienern oder Wartungspersonal anzuzeigen oder darzubieten. Auch ist es schwierig, Objekte in Bedieneranzeigen mit diesen alternativen Informationsquellen oder Daten für die unterschiedlichen Elemente in der Anzeige zu animieren.

#### Zusammenfassung der Offenbarung

**[0015]** Es werden Grafikelemente und Grafikanzei-

gen zur Verwendung in einer Prozeßumgebung zur Darstellung von Informationen über die Prozeßumgebung wie beispielsweise den aktuellen Zustand von Vorrichtungen in einer Prozeßanlage für einen oder mehrere Benutzer bereitgestellt. Die Grafikelemente und Anzeigen enthalten ein oder mehrere Objekte, von denen jedes eine Visualisierung umfaßt, die eine Instanz bzw. Anzeigeentität in der Prozeßumgebung darstellt, eine der dargestellten Prozeßinstanz zugeordnete Eigenschaft und eine Routine, die in Verbindung mit der Visualisierung und der Eigenschaft fungiert, um eine Änderung der Visualisierung auf Grundlage einer Änderung der Eigenschaft zu veranlassen, oder eine Änderung der Prozeßumgebung auf Grundlage einer Benutzereingabe durch die Visualisierung zu veranlassen. Auf diese Weise können die Grafikobjekte zur Erstellung von Anzeigen benutzt werden, die durch Animierungen und sonstige visuelle Operationen den sich ändernden Zustand des Prozesses oder erkannte Ereignisse in dem Prozeß grafisch darstellen. Diese gleichen Grafikelemente können dazu benutzt werden, Benutzern im Dialog mit einer Visualisierung auf einem Benutzeroberflächenbildschirm wie beispielsweise durch eine Maus oder eine Tastaturvorrichtung zu ermöglichen, eine Änderung der Objekteigenschaft zu veranlassen, die dann eine Eingabe in eine Vorrichtung in der Prozeßanlage bereitstellen kann.

#### Kurze Beschreibung der Zeichnungen

**[0016]** [Fig. 1](#) ist ein Blockdiagramm eines in einer Prozeßanlage befindlichen verteilten Prozeßsteuerungsnetzes mit Bedienerarbeitsplätzen, die Anzeigeroutinen und sonstige, mit verschiedenen Funktionen in der Prozeßanlage verbundene Anwendungen implementieren, wie auch einem Arbeitsplatz, der grafische Unterstützung auf Systemebene bereitstellt, die zur Erstellung von Grafikanzeigeelementen und Grafikanzeigen für jeden der verschiedenen Funktionsbereiche der Anlage benutzt werden kann;

**[0017]** [Fig. 2](#) ist ein logisches Blockdiagramm der Integration von grafischer Unterstützung auf Systemebene in einem Prozeßanlagensteuerungs-, Betrachtungs- und Simulationssystem;

**[0018]** [Fig. 3](#) ist ein Logikdiagramm einer Konfigurationsumgebung, in der Grafikelemente und -anzeigen erstellt werden und einer Laufzeitumgebung, in der Grafikelemente und -anzeigen ausgeführt werden können;

**[0019]** [Fig. 4](#) ist ein vereinfachtes, von einem Grafikeditor erzeugtes Schirmbild, um einem Benutzer die Erstellung oder Aufbereitung eines Grafikelements in der Form eines Pumpenelements zu ermöglichen;

**[0020]** [Fig. 5](#) ist ein weiteres, von einem Grafikeditor erzeugtes Schirmbild bei der Erzeugung eines

Grafikelements in der Form eines Reaktors;

[0021] [Fig. 6](#) ist ein Blockdiagramm einer Weise, auf die verschiedene Visualisierungen an ein Grafikelement angebunden oder diesem zugeordnet werden können;

[0022] [Fig. 7](#) zeigt einen Abschnitt eines Schirmbildes einer ersten, einem Grafikelement zugeordneten Eigenschaftsansicht;

[0023] [Fig. 8](#) zeigt einen Abschnitt eines Schirmbildes einer zweiten, einem Grafikelement zugeordneten Eigenschaftsansicht;

[0024] [Fig. 9](#) zeigt einen Aktions-/Animationsabschnitt eines Schirmbildes mit Aktionen, die Eigenschaften eines Grafikelements zugeordnet sind;

[0025] [Fig. 10](#) ist ein Blockschaltbild einer Weise, auf die visuelle Trigger mit Eigenschaften und Visualisierungen eines Grafikelements integriert werden können;

[0026] [Fig. 11](#) zeigt einen Dialogkasten, der zur Bereitstellung oder Definierung einer Transformationsanimierung für eine Visualisierung eines Grafikelements benutzt werden kann;

[0027] [Fig. 12](#) zeigt einen Dialogkasten, der zur Bereitstellung oder Definierung einer Eigenschaftsanimierung, einschließlich einer Farbanimierung, für eine Visualisierung eines Grafikelements benutzt werden kann;

[0028] [Fig. 13](#) zeigt ein Blockschaltbild, das allgemein die Art und Weise darstellt, auf die Grafikelemente über eine Auflösungstabelle an eine Laufzeitumgebung angebunden werden können;

[0029] [Fig. 14](#) ist ein Blockschaltbild eines an mehrere unterschiedliche Datenquellen in einer Prozeßumgebung angebandenen Anzeigeelements;

[0030] [Fig. 15](#) ist ein durch einen Grafikeditor erzeugtes Schirmbild bei der Erzeugung einer Grafikanzeige aus mehreren Anzeigeelementen und Verbindern;

[0031] [Fig. 16](#) ist ein Schirmbild zur Erstellung einer Grafikanzeige verschiedener Visualisierungen, die einem in der Grafikanzeige platzierten Grafikelement zugeordnet sind;

[0032] [Fig. 17](#) ist ein Schirmbild, das einer aus verschiedenen miteinander verbundenen Grafikelementen und Verbindern bestehenden Grafikanzeige zugeordnet ist;

[0033] [Fig. 18](#) ist ein Satz von Grafikschrmbildern,

auf die über die Grafikanzeige der [Fig. 17](#) zugegriffen werden kann, um ein Steuerfeld und eine Frontplatte für ein Element in der Anzeige der [Fig. 17](#) darzustellen;

[0034] [Fig. 19](#) ist ein Schirmbild einer Grafikanzeige eines Kalkofens aus verschiedenen Anzeigeelementen mit Daten aus verschiedenen anderen Anwendungen und Datenquellen in einer Prozeßanlage;

[0035] [Fig. 20A–Fig. 20E](#) sind Grafikschrmbilder verschiedener Ansichten eines Kalkofens mit dem gleichen Aussehen und Eindruck einschließlich einer Bedieneransicht, einer technischen Ansicht, einer Geschäftsansicht, einer Simulationsansicht und einer Wartungsansicht;

[0036] [Fig. 21A](#) und [Fig. 21B](#) sind Schirmbilder, die einer Steuerungsroutine zum Betreiben des in [Fig. 20A–Fig. 20E](#) gezeigten Kalkofens benutzten Steuerungsroutine zugeordnet sind;

[0037] [Fig. 22](#) ist ein Blockschaltbild von Zuordnungen zwischen verschiedenen unterschiedlichen Grafikanzeigen auf verschiedenen Ebenen einschließlich physikalischen Ebenen und Funktionsebenen;

[0038] [Fig. 23](#) zeigt ein erstes Konfigurationsschrmbild einer einem Konfigurationssystem zugeordneten grafischen Konfigurationshierarchie;

[0039] [Fig. 24](#) zeigt ein zweites Konfigurationsschrmbild einer grafischen Konfigurationshierarchie einer Art und Weise, auf die Grafikelemente und Grafikanzeigen anderen Elementen in dem Konfigurationssystem zugewiesen werden und darin integriert werden können; und

[0040] [Fig. 25](#) zeigt einen Dialogkasten, der für einen Benutzer bereitgestellt werden kann, um die Art und Weise zusammenzufassen, auf die eine oder mehrere Grafikanzeigen in einem Prozeßanlagen-Konfigurationssystem zugewiesen werden.

#### Ausführliche Beschreibung

[0041] [Fig. 1](#) zeigt eine beispielhafte Prozeßanlage **10**, in der grafische Unterstützung auf Systemebene für verschiedene Funktionsbereiche der Anlage **10** bereitgestellt wird. Typischerweise enthält die Prozeßanlage **10** ein verteiltes Prozeßsteuerungssystem mit einer oder mehreren Steuerungen **12**, jeweils mit einer oder mehreren Betriebsvorrichtungen **14** und **16** über Eingangs/Ausgangs(E/A-)Vorrichtungen und **18** über Karten **18** verbunden, die beispielsweise Fieldbus-Schnittstellen, Profibus-Schnittstellen, HART-Schnittstellen, standardmäßige 4–20 ma-Schnittstellen usw. sein können. Die Steuerungen **12** sind auch über einen Datenweg **24**, der beispielsweise eine Ethernet-Verbindung sein kann, an

eine oder mehrere Host- oder Bedienerarbeitsplätze **20–23** angekoppelt. An den Datenweg **24** kann eine Datenbank **28** angeschlossen sein und fungiert als Datengeschichtsschreiber zur Einsammlung und Speicherung an den Steuerungen und Betriebsvorrichtungen in der Anlage **10** zugeordneten Parameter-, Zustands- und sonstigen Daten, und/oder als Konfigurationsdatenbank, die die aktuelle Konfiguration des Prozeßsteuerungssystems in der Anlage **10** speichert, sowie die zu den Steuerungen **12** und Betriebsvorrichtungen **14** und **16** heruntergeladen und darin gespeichert ist. Die Datenbank **28** kann zusätzlich auf hier beschriebene Weise erstellte Grafikobjekte zur Bereitstellung von Grafikunterstützung in der Prozeßanlage **10** speichern. Während sich die Steuerungen **12**, E/A-Karten **18** und Feldbetriebsvorrichtungen **14** und **16** typischerweise unten innerhalb der manchmal rauen Anlagenumgebung befinden und dort verteilt sind, befinden sich die Bedienerarbeitsplätze **20–23** und die Datenbank **28** gewöhnlich in Steuerwarten oder weniger rauen Umgebungen, die leicht durch Steuerungs- oder Wartungspersonal bewertbar sind. In manchen Fällen können jedoch zur Implementierung dieser Funktionen Handvorrichtungen benutzt werden und diese Handvorrichtungen werden typischerweise zu verschiedenen Stellen in der Anlage getragen.

**[0042]** Wie bekannt ist, wird von jeder der Steuerungen **12**, die beispielsweise die von Emerson Process Management vertriebene Steuerung DeltaV™ sein kann, eine Steuerungsanwendung gespeichert und ausgeführt, die unter Verwendung einer beliebigen Anzahl von unterschiedlichen, unabhängig ausgeführten Steuerungsmodulen oder Blöcken **29** eine Steuerungsstrategie implementiert. Jedes der Steuerungsmodule **29** kann aus gewöhnlich als Funktionsblöcken bezeichneten Blöcken bestehen, wobei jeder Funktionsblock ein Teil oder eine Unterroutine einer Gesamt-Steuerungsroutine ist und in Verbindung mit anderen Funktionsblöcken (über Verbindungen genannte Kommunikationen) zur Implementierung von Prozeßsteuerungsschleifen in der Prozeßanlage **10** arbeitet. Wohlbekannterweise führen Funktionsblöcke, die Objekte in einem objektorientierten Programmierungsprotokoll sein können, typischerweise eine von einer Eingabefunktion wie der mit einem Sender, einem Sensor oder einer sonstigen Prozeßparametermeßvorrichtung verbundenen, eine Steuerungsfunktion wie beispielsweise der mit einer Steuerungsroutine verbundenen, die PID-, Fuzzy-Logik- usw. Steuerung durchführt, oder eine Ausgangsfunktion, die die Funktionsweise irgendeiner Vorrichtung wie beispielsweise eines Ventils zur Durchführung irgendeiner physikalischen Funktion in der Prozeßanlage **10** steuert, durch. Natürlich gibt es hybride und andere Arten komplizierter Funktionsblöcke, wie beispielsweise MPC (model predictive controllers), Optimierer usw. Während bei dem Fieldbus-Protokoll und dem DeltaV-Systemprotokoll Steuerungsmodulen

und Funktionsblöcke benutzt werden, die in einem objektorientierten Programmierungsprotokoll entworfen und implementiert sind, könnten die Steuerungsmodulen unter Verwendung eines beliebigen gewünschten Steuerungsprogrammierverfahrens entworfen sein, einschließlich beispielsweise von sequentiellem Funktionsblocks, Leiterlogik usw. und sind nicht darauf begrenzt, unter Verwendung des Funktionsblock oder sonstigen bestimmten Programmierverfahrens entworfen und implementiert zu sein.

**[0043]** In der in [Fig. 1](#) dargestellten Anlage **10** können die mit den Steuerungen **12** verbundenen Betriebsvorrichtungen **14** und **16** standardmäßige 4–20 ma-Vorrichtungen oder intelligente Betriebsvorrichtungen wie beispielsweise HART-, Profibus- oder FOUNDATION™-Fieldbus-Betriebsvorrichtungen sein, die einen Prozessor und einen Speicher enthalten, oder können irgendeine andere gewünschte Vorrichtungsart sein. Einige dieser Vorrichtungen wie beispielsweise Fieldbus-Betriebsvorrichtungen (mit Bezugsziffer **16** in der [Fig. 1](#) bezeichnet) können Modulen oder Teilmodulen wie beispielsweise Funktionsblöcke speichern und ausführen, die mit der in den Steuerungen **12** implementierten Steuerungsstrategie verbunden sind. Funktionsblöcke **30**, die in der [Fig. 1](#) als in zwei verschiedenen der Fieldbus-Betriebsvorrichtungen **16** angeordnet dargestellt sind, können in Verbindung mit der Ausführung der Steuerungsmodulen **29** in den Steuerungen **12** zur Implementierung von Prozeßsteuerung, wie wohlbekannt ist, ausgeführt werden. Natürlich können die Betriebsvorrichtungen **14** und **16** eine beliebige Art von Vorrichtungen wie beispielsweise Sensoren, Ventile, Sender, Stellelemente usw. sein und die E/A-Vorrichtungen **18** können beliebige Arten von E/A-Vorrichtungen sein, die einem beliebigen gewünschten Kommunikations- oder Steuerungsprotokoll wie beispielsweise HART, Fieldbus, Profibus entsprechen.

**[0044]** In der Prozeßanlage **10** der [Fig. 1](#) können die Arbeitsplätze **20–23** verschiedene Anwendungen enthalten, die für verschiedene, durch das gleiche oder anderes Personal in der Anlage **10** durchgeführte unterschiedliche Funktionen genutzt werden. Jeder der Arbeitsplätze **20–23** enthält einen Speicher **31**, der verschiedene Anwendungen, Programme, Datenstrukturen usw. speichert, und einen Prozessor **32**, der zur Ausführung einer beliebigen der im Speicher **31** gespeicherten Anwendungen benutzt werden kann. In dem in [Fig. 1](#) dargestellten Beispiel ist der Arbeitsplatz **20** als Konfigurationsarbeitsplatz bezeichnet und enthält eine oder mehrere Konfigurationsanwendungen **33**, die beispielsweise Steuerungsmodulerstellungsanwendungen, Bedienerschnittstellenanwendungen und sonstige Datenstrukturen enthalten können, auf die jeder berechnete Konfigurationsingenieur zugreifen kann, um Steuerungsroutinen oder Modulen wie beispielsweise die Steuerungsmodulen **29** und **30** zu erstellen und zu

den verschiedenen Steuerungen **12** und Vorrichtungen **16** der Anlage **10** herunterzuladen. Der Arbeitsplatz **21** ist allgemein in [Fig. 1](#) als Steuerungsbedienersichtplatz dargestellt und enthält eine Anzahl von Anzeigeanwendungen **34**, die während des Betriebs der Prozeßanlage **10** verschiedene Anzeigen für einen Steuerungsbediener bereitstellen können, und dem Bediener die Betrachtung und Steuerung des Geschehnisses in der Prozeßanlage **10** oder in verschiedenen Teilen der Anlage ermöglicht. Die Anwendungen **34** können Unterstützungsanwendungen **34a** wie beispielsweise Steuerungsdiagnoseanwendungen, Abstimmungsanwendungen, Reportgenerationsanwendungen oder beliebige andere Steuerungsunterstützungsanwendungen umfassen, die zur Unterstützung eines Steuerungsbedieners bei der Durchführung von Steuerungsfunktionen benutzt werden können. Auf ähnliche Weise ist der Arbeitsplatz **22** als Wartungssichtplatz dargestellt und enthält eine Anzahl von Wartungsanwendungen **35**, die von verschiedenem Wartungspersonal zur Betrachtung der Wartungsbedürfnisse der Anlage **10**, zur Betrachtung des Betriebs- oder Arbeitszustandes verschiedener Vorrichtungen **12**, **14**, **16** usw. benutzt werden können. Natürlich können die Anwendungen **35** Unterstützungsanwendungen **35a** wie beispielsweise Wartungsdiagnoseanwendungen, Eichungsanwendungen, Schwingungsanalyseanwendungen, Reportgenerationsanwendungen oder beliebige Wartungsunterstützungsanwendungen sein, die zur Unterstützung einer Wartungsperson bei der Durchführung von Wartungsfunktionen in der Anlage **10** benutzt werden können. Zusätzlich ist der Arbeitsplatz **23** als Simulationsarbeitsplatz angezeigt, der eine Anzahl von Simulationsanwendungen **36** enthält, die zur Simulation der Funktionsweise der Anlage **10** oder verschiedener Teile der Anlage **10** für eine beliebige Anzahl von Zwecken benutzt werden kann, einschließlich von Trainingszwecken, Anlagenmodellierungszwecken zur Hilfe bei Anlagenwartung und Steuerung usw. Typischerweise enthält jeder der Arbeitsplätze **20–23** einen Bildschirm **37** zusammen mit anderen standardmäßigen Peripherievorrichtungen wie beispielsweise einer Tastatur, einer Maus usw.

**[0045]** Während die verschiedenen Konfigurations-, Steuerungs-, Wartungs- und Simulationsanwendungen **33–36** in der [Fig. 1](#) als in einer dieser Funktionen zugeordneten unterschiedlichen Arbeitsplätzen befindlich dargestellt sind, versteht es sich natürlich, daß die verschiedenen, diesen und anderen Anlagenfunktionen zugeordneten Anwendungen **33–36** in Abhängigkeit von den Bedürfnissen und der Einrichtung der Anlage **10** in den gleichen oder unterschiedlichen Arbeitsplätzen oder Computern in der Anlage **10** befindlich und ausgeführt sein können. So kann beispielsweise eine oder mehrere Simulationsanwendungen **36** und Steuerungsanwendungen **33** auf dem gleichen Arbeitsplatz **20–23** ausgeführt werden, während unterschiedliche einzelne Simulationsan-

wendungen **36** oder unterschiedliche einzelne Steuerungsanwendungen **33** auf unterschiedlichen Arbeitsplätzen **20–23** ausgeführt werden können.

**[0046]** In der Vergangenheit wurde die Entwicklung der verschiedenen, in den verschiedenen Funktionsbereichen der Anlage **10** benutzten Anwendungen ziemlich unabhängig durchgeführt. So war die Entwicklung der Konfigurationsanwendungen **33** nicht mit den Simulationsanwendungen **36**, den Wartungsanwendungen **35** oder den Bedienersteuerungsanwendungen **34** integriert. In der Tat kann eine Anlage in vielen Fällen Anwendungen für die verschiedenen Funktionsbereiche enthalten haben, die durch unterschiedliche Firmen oder Softwarelieferanten entwickelt wurden und die in der Tat entwickelt worden sind, um unabhängig von der anderen Software in der Anlage **10** abzulaufen. Infolge dieser unabhängigen Entwicklung und Funktionsweise der den verschiedenen Funktionsbereichen der Anlage **10** zugeordneten unterschiedlichen Anwendungen mußte das Anlagenpersonal im allgemeinen eine Anlage auf jeder der Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulations-Funktionsebene getrennt konfigurieren oder einrichten. Insbesondere mußte das gleiche oder anderes Anlagenpersonal im allgemeinen unterschiedliche Programme zur Herstellung neuer Datenstrukturen und Grafikanzeigen auf jeder der Funktionsebenen benutzen. So enthielt oder benutzte bezüglich der [Fig. 1](#) jede der verschiedenen Anwendungen **33–36**, die Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsfunktionen durchführte, unterschiedliche Grafikanzeigeneditoren und Datenbankstrukturen zur Unterstützung des Anlagenpersonals bei der Durchführung dieser Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsfunktionen. In vielen Fällen wurden diese unterschiedlichen Grafikanzeigeneditoren und Datenbanken zur Erstellung unterschiedlicher Grafikanzeigen, zur Darstellung oder Modellierung der gleichen Abschnitte oder Bereiche der Anlage **10** oder der gleichen Hardware in der Anlage **10** und zur Unterstützung unterschiedlichen Anlagenpersonals benutzt und um das Geschehnis in der Prozeßanlage im Zusammenhang mit Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsaktivität zu visualisieren und zu verstehen.

**[0047]** Da die Anwendungen **33–36** und die zugehörigen Anzeigen für jede verschiedene Funktionen in der Anlage **10** im allgemeinen unabhängig voneinander entwickelt und implementiert wurden, manchmal von unterschiedlichen Personen und sogar unterschiedlichen Firmen, wurden die in den verschiedenen Funktionsbereichen der Prozeßanlage erstellten oder benutzten Grafikanzeigen auf keine konsequente oder leicht verständliche Weise aus dem Standpunkt der Bereitstellung grafischer Visualisierung integriert. Infolgedessen führte die unabhängige Erstellung und Ausführung grafischer Anzeigen auf den

verschiedenen unterschiedlichen Funktionsebenen der Anlage dazu, daß die Grafikanzeigen von Funktion zu Funktion anders aussahen, so daß die Grafikanzeigen über die Funktionsbereiche hinweg kein konsequentes Aussehen und keinen konsequenten Eindruck boten. Zusätzlich führte diese unabhängige Erstellung zur Duplikation des Aufwandes bei der Erstellung grafischer Anzeigen für die gleichen Abschnitte oder Bereiche der Anlage, aber für unterschiedliche Funktionsverwendung, und erforderte, daß die so erstellten Grafikanzeigen in die verschiedenen Vorrichtungen in der Anlage **10** wie beispielsweise die Steuerungen **12** und die Betriebsvorrichtungen **14, 16** getrennt auf den Funktionsebenen der Anlage **10** eingebunden wurden und von ihnen Daten empfangen mußten. Das erforderte wiederum die Duplikation von Datenbankstrukturen zur Verfolgung der gleichen Hardwareelemente für die verschiedenen Anzeigen. So wurde beispielsweise in der Vergangenheit eine erste Anwendung (z. B. eine der Anwendungen **35**) zur Erstellung einer Wartungsanzeige benutzt, die einen Teil der Anlage **10** für Wartungszwecke darstellt, während eine zweite Anwendung (z. B. eine der Anwendungen **34**) zur Erstellung einer Steuerungsbedieneranzeige benutzt wurde, die denselben Teil der Anlage **10** für Steuerungszwecke darstellt. Die getrennt durch diese verschiedenen Anzeigeditoren erstellten Anzeigen konnten in Aussehen und Eindruck ganz unterschiedlich sein, was es für einen Benutzer schwierig machte, zwischen der Wartungsanzeige und der Bedieneranzeige hin und her zu schalten, ohne leicht verwirrt zu werden oder Training an jeder Anzeigart zu erfordern. Auf gleiche Weise wurde der Aufwand bei der unabhängigen Erstellung beider Anzeigen in den verschiedenen Anwendungen **34** und **35** verdoppelt und zusätzlicher Aufwand zur Erstellung von Datenbankstrukturen zur getrennten Anwendung oder Verbindung der Wartungsanzeige und der Steuerungsbedieneranzeige an die gleichen Hardwareelemente in der Anlage **10** aufgewandt, um manchmal die gleichen oder ähnlichen Daten von diesen Hardwareelementen zu empfangen.

**[0048]** Um diese Unzulänglichkeiten zu lindern und in weiterem Umfang nutzbare und verständliche Grafiken in der Anlage **10** zu ermöglichen, wird eine grafische Unterstützungsschicht auf einer Systemebene der Prozeßanlage **10** bereitgestellt, um die Grafikanzeige und Datenstrukturbedürfnisse jedes der verschiedenen Funktionsbereiche der Anlage **10** einschließlich der Konfigurations-, Bedienerbetrachtungs-, Wartungsbetrachtungs-, Simulations- und sonstiger Funktionsbereiche der Anlage **10** zu unterstützen. Diese Systemebene von Unterstützung wird diagrammatisch in [Fig. 2](#) dargestellt, die eine Anlagenbetriebsebene **40**, eine Anlagenfunktionsebene **42** und eine Systemebene **44** darstellt. Wie aus [Fig. 2](#) verständlich sein wird, umfaßt die Anlagenbetriebsebene **40** die Steuerungen **12**, Betriebsvorrich-

tungen **14, 16** usw., die die Steuerungsrountinen oder Modulen **29** und **30** ausführen, wie auch andere in der Anlage **10** ablaufende Software, um Anlagenoperationen während der Laufzeit der Anlage zu implementieren. Die Anlagenfunktionsebene **42** ist als einen Konfigurationsfunktionsblock **46**, einen Steuerungsfunktionsblock **47**, einen Wartungsfunktionsblock **48** und einen Simulationsblock **49** enthaltend dargestellt, obwohl andere oder unterschiedliche Funktionen wie beispielsweise eine technische und eine Geschäftsfunktion ebenfalls bereitgestellt werden könnten. Durch den Konfigurationsfunktionsblock **46** werden die Konfigurationsrountinen **33** implementiert, die mit Komponenten auf der Anlagenbetriebsebene **40** kommunizieren oder verbunden sind, um Steuerungsstrategien oder Steuerungsmodulen dafür bereitzustellen. Der Steuerungsfunktionsblock **47** enthält die Steuerungsbetrachtungs- und andere Anwendungen **34** und **34a**, die ebenfalls typischerweise direkt mit den verschiedenen physikalischen und logischen Komponenten auf der Anlagenbetriebsebene **40** kommunizieren oder verbunden sind, um vom Bediener eingeleitete Änderungen in der Anlage **10** zu implementieren, um über Steuerungsanzeigen **34** Informationen für den Bediener bereitzustellen, um Daten für die Steuerungsanwendungen **34a** zu erfassen usw. Der Wartungsfunktionsblock **48** enthält die Wartungsrountinen und Anwendungen **35** und **35a**, die mit verschiedenen physikalischen und logischen Komponenten auf der Anlagenbetriebsebene **40** kommunizieren oder verbunden sind, um Wartungsprozeduren zu implementieren, Wartungsdaten einzusammeln, Wartungsdaten oder Informationen über Wartungsanzeigen **35** für eine Wartungsperson bereitzustellen, Diagnoseanwendungen **35a** ablaufen zu lassen usw. Auf gleiche Weise enthält der Simulationsfunktionsblock **49** Simulationsrountinen **36**, die Simulationen der Anlage **10** implementieren und die kommunikativ an Komponenten auf der Anlagenbetriebsebene **40** angekoppelt sein können, um Daten bezüglich der Anlage **10** zu erhalten.

**[0049]** Nach der Darstellung in [Fig. 2](#) ist die Unterstützungsschicht **44** auf Systemebene an jeden der Funktionsblöcke **46–49** auf der Anlagenfunktionsebene **42** eingebunden und unterstützt diese, um beispielsweise die Erstellung und Unterhaltung von gemeinsamen Datenbank- und Anzeigenstrukturen wie beispielsweise Softwareobjekte, Grafikelemente und Grafikanzeigen zur Verwendung in den verschiedenen Funktionsbereichen **46–49** zu ermöglichen. Insbesondere enthält die Unterstützungsschicht **44** auf Systemebene Anwendungs-, Datenbank- und Grafikunterstützungselemente, durch die die in jedem der Funktionsblöcke **46–49** durchgeführten grafischen Aktivitäten zusammen integriert werden können oder unter Verwendung gemeinsamer Datenbankstrukturen und auf der Systemunterstützungsschicht **44** erstellten Grafikelementen entwickelt werden können. Zur Bereitstellung dieser Unterstützung auf Systeme-

bene kann die Systemunterstützungsschicht **44** einen Grafikeditor **50** und eine grafische Objektdatenbank **52** umfassen. Der Grafikeditor **50** kann zur Erstellung von Grafikelementen **54** und Grafikanzeigen **56** benutzt werden, während die grafische Objektdatenbank **52** die Elemente **54** und Anzeigen **56** in einem dem Editor **52** und den verschiedenen Anwendungen in den Blöcken **46–49** zugänglichen Speicher speichert. Auch kann die Datenbank **52** andere Objekte **58** und Datenstrukturen speichern, die Grafikelemente **54** mit einzelnen Hardware- und Softwareelementen auf der Anlagenbetriebsebene **40** verbinden. Zusätzlich kann die Datenbank **52** Grafikelement- oder Anzeigenschablonen oder Grundelemente speichern, die zur Erstellung weiterer Grafikelemente oder -anzeigen genutzt werden können. Wie aus [Fig. 2](#) zu verstehen ist, können die Grafikanzeigeelemente **54**, Anzeigen **56** und andere Datenbankstrukturen **58** von einem beliebigen der Funktionsblöcke **46–49** benutzten Grafiken zur Erstellung und Verwendung von diesen Funktionsblöcken zugeordneten Grafiken genutzt werden.

**[0050]** Allgemein gesagt wird vom Unterstützungsblock **44** auf Systemebene eine Art und Weise zum Integrieren der in der Prozeßanlage **10** der [Fig. 1](#) in allen Funktionsbereichen **46–49** benutzten Grafiken bereitgestellt, um dadurch die Notwendigkeit der wiederholten Erstellung von unterschiedlichen Grafikelementen für die gleiche Anlageneinrichtung in unterschiedlichen Funktionszusammenhängen zu verringern oder zu beseitigen und es einem Benutzer in jedem der Funktionsbereiche **46–49** leicht zu machen, sich in Daten einzubinden, die dem Gerät zugeordnet sind, das in diesen Funktionsbereichen zugeordneten grafischen Ansichten dargestellt wird. Es versteht sich, daß die Unterstützungsschicht **44** auf Systemebene zur Bereitstellung von Grafik- und Datenbankunterstützung für mehrere Anwendungen in jedem der Funktionsbereiche **46–49**, für unterschiedliche Anwendungen in unterschiedlichen der Funktionsbereiche **46–49** usw. benutzt werden kann.

**[0051]** Wieder auf [Fig. 1](#) bezugnehmend kann der Unterstützungsblock **44** auf Systemebene unter Verwendung eines zusätzlichen Arbeitsplatzes oder einer zusätzlichen Benutzeroberfläche **60** implementiert werden, die mit jedem der anderen Arbeitsplätze **20–23** verbunden sein kann. Der Arbeitsplatz **60** kann allgemein den Grafikeditor **50** und die Datenbank **52** speichern und gegebenenfalls die anderen Elemente **54**, **56** und **58** speichern. Zusätzlich kann der Arbeitsplatz **60** kommunikativ über getrennte drahtgebundene oder drahtlose Kommunikationsverbindungen (durch gestrichelte Linie in der [Fig. 1](#) dargestellt) oder auf eine beliebige sonstige gewünschte Weise mit den Arbeitsplätzen **20–23** verbunden sein. In der in [Fig. 1](#) dargestellten Konfiguration wird der Anzeigeeditor **50** vom Arbeitsplatz **60** gespeichert und ausgeführt, um einem Benutzer die Erstellung

von Grafikelementen und das Kopieren dieser Elemente in einer oder mehreren Grafikanzeigen zu ermöglichen, die beide hier als Anzeigemodulen bezeichnet werden. Diese Anzeigemodulen können dann in der Datenbank **52** gespeichert werden, damit verschiedene der in [Fig. 2](#) dargestellten und auf den verschiedenen Arbeitsplätzen **20–23** implementierten Funktionsblöcke **46–49** darauf zugreifen und sie benutzen können. Während die Funktionalität des Blocks **44** auf Systemebene und der Blöcke **46–49** auf Systemebene für Darstellungszwecke als auf verschiedenen oder getrennten Arbeitsplätzen **20–23** und **60** in der [Fig. 1](#) implementiert dargestellt ist, versteht es sich, daß eine beliebige oder alle der einem beliebigen dieser verschiedenen Blöcke zugeordneten Anwendungen auf dem gleichen oder anderen Arbeitsplätzen oder sonstigen Computern in der Prozeßanlage **10** oder dieser zugeordnet implementiert sein könnten. So kann der Grafikeditor **50** in einem beliebigen der anderen Arbeitsplätze **20–23** oder einem beliebigen anderen, der Anlage **10** zugeordneten Computer gespeichert sein und darauf ausgeführt werden und muß sich nicht in einem alleinstehenden oder getrennten Computer befinden.

**[0052]** Wie oben besprochen, werden auf der Schicht **44** der Systemebene der [Fig. 2](#) Anzeige- und Datenbankobjekte auf Systemebene implementiert, die in verschiedenen Funktionsumgebungen zur Bereitstellung von Anzeigefähigkeiten auf höherer Ebene benutzt werden können. Allgemein gesagt können die auf Systemebene **44** der [Fig. 2](#) erstellten Anzeigeobjekte als Grafikelemente und Grafikanzeigen kategorisiert werden. Grafikelemente sind allgemein Grundelement- oder niedere Anzeigeobjekte, die einer bestimmten physikalischen Instanz bzw. physische Entität in der Anlage wie beispielsweise einer Hardwarevorrichtung wie einem Ventil, einem Sensor, einer Pumpe, einer Steuerung usw. zugeordnet sind. Grafikanzeigen bestehen allgemein aus einer Menge miteinander verbundener Grafikelemente und werden zur Darstellung und Modulierung von komplizierteren Menge von Hardware in einer Anlage wie beispielsweise einer Einheit, eines Bereichs usw. benutzt und umfassen Zwischenverbindungen zwischen verschiedenen Hardwareeinheiten. Zusätzlich können Grafikanzeigen Graphen, Karten und sonstige von der Anlage, von anderen Anwendungen wie beispielsweise auf den Arbeitsplätzen **20–23** und **60** ablaufenden Diagnose- und Geschäftsanwendungen usw. bereitgestellte Daten umfassen.

**[0053]** [Fig. 3](#) zeigt allgemein die Entwicklung und Verwendung von Grafikelementen und Grafikanzeigen in zwei Umgebungen, in denen diese Elemente und Anzeigen vorkommen können, insbesondere einer Konfigurationsumgebung **70** und einer Laufzeitumgebung **72**. Allgemein gesagt werden Anzeigeobjekte in der Form von Grafikelementen **74** (als getrennte Elementobjekte **74a**, **74b** usw. dargestellt)

und Grafikanzeigen **76** (als getrennte Anzeigeobjekte **76a**, **76b** usw. dargestellt) in der Konfigurationsumgebung **70** unter Verwendung beispielsweise des Anzeigeditors **50** erstellt. Nach ihrer Erstellung können die Objekte **74** und **76** in der Datenbank **52** gespeichert werden. Die Objekte **74** und **76** können als Klassenobjekte, hier als Anzeigemodulklassenobjekte bezeichnet, erstellt werden, die ein nicht an bestimmte Hardware oder Logikelemente in der Prozeßanlage **10** gebundenes generisches Objekt definieren. Klassenobjekte können jedoch zur Erstellung von Laufzeit-Grafikobjekten mit den gleichen Grundeigenschaften wie die Klassenobjekte benutzt werden, die aber an bestimmte Hardware in der Prozeßanlage **10** angebunden sind. Allgemein gesagt bleiben jedoch Klassenobjekte an daraus konkretisierte abgeleitete Objekte angebunden, so daß Änderungen der Klassenobjekte automatisch zu den abgeleiteten Objekten verbreitet werden können, selbst wenn diese abgeleiteten Objekte innerhalb einer Laufzeitumgebung konkretisiert werden.

**[0054]** Wie in der [Fig. 3](#) dargestellt, enthält jedes der Grafikelementobjekte **74** eine Anzahl von Komponenten, die das Grafikelement in vielen verschiedenen Zusammenhängen nützlich machen. Insbesondere enthält Grafikelement **74** eine oder mehrere Visualisierungen **77**, eine beliebige Anzahl von Parametern oder Eigenschaften **78**, eine beliebige Anzahl von Handlungen oder Animationen **79**, die unter Verwendung von Skripten oder Triggern implementiert werden können, und Anbindungen **80**. Allgemein gesagt definiert jede Visualisierung **77** die sichtbaren Eigenschaften oder Elemente, die tatsächlich auch auf einem Bildschirm anzuzeigen sind, wenn das Grafikelement **74** in der Laufzeitumgebung **72** implementiert wird. Typischerweise definieren Visualisierungen eine Darstellung einer physikalischen oder logischen Vorrichtung oder Gruppe von Vorrichtungen, obwohl eine Visualisierung andere Instanzen darstellen könnte. Visualisierungen **77** können in der Laufzeitumgebung **72** unter Verwendung jedes gewünschten Beschreibungs- oder Programmierparadigmas implementiert werden, das die spezifischen Einzelheiten der grafischen Darstellung einer Instanz definiert. In einer Ausführungsform können die Visualisierungen **77** unter Verwendung von PGXML- oder Avalon-Steuerungen implementiert werden, die wohlbekannte, von Microsoft® bereitgestellte Steuerungen sind und die, da sie objektbasiert sind, eine Art und Weise bereitstellen, die Visualisierungen in standardmäßigen Anzeigen der Art Windows® leicht implementierbar und zwischen Anzeigenumgebungen portierbar zu machen. Dieses Merkmal wird ausführlicher unten besprochen und wird in der gleichzeitigen abhängigen Anmeldung mit dem Titel "Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface" (Auf Textauszeichnungssprache basierende dynamische Prozeßgrafiken in einer Prozeßanlagen-Benutzeroberflä-

che) (Aktenzeichen Nr. 06005/41127) erläutert, die durch Bezugnahme hier aufgenommen wird.

**[0055]** Allgemein gesagt definieren die Parameter und Eigenschaften **78** Variablen oder sonstige Eigenschaften wie statische und veränderliche innere Eigenschaften, die mit der durch die Visualisierungen dargestellten Instanz verbunden sind, und diese Eigenschaften sind durch den Ersteller des Elements **74** definierbar. Auf gleiche Weise definieren die Handlungen und Animationen **79** Routinen oder Programme (die als Skripts implementiert sein können, die Transformationen an Eigenschaften durchführen, die Zustände einer Prozeßinstanz auf Grundlage von Eigenschaftswerten erkennen usw.), Animationsroutinen, die alle Routinen enthalten können, die die grafischen Visualisierungen oder Verhaltensweisen für deren Durchführung oder Verwendung der Visualisierung **77** bei Darstellung der Visualisierung **77** auf einem Bildschirm ändern, oder Routinen, die einem Benutzer die Verwendung oder den Dialog mit der Visualisierung **77** ermöglichen, um eine Änderung des Prozesses wie beispielsweise eine Änderung einer Eingabe in den Prozeß zu veranlassen. Diese Handlungen und Animationen stellen interessantere verständlichere oder hilfreichere grafische Eigenschaften für die Visualisierung **77** bereit und erlauben dem Benutzer den Dialog mit der Visualisierung **77**. In einem Fall können diese Handlungen oder Animationen die Form von Änderungen der Farbe, Größe (z. B. Höhe und Breite, Zeilengröße, Schriftarten usw.) verschiedener Komponenten der Visualisierung, Farbfüllbereiche und Animationen wie beispielsweise Änderungen der Farbe, Drehungen, Größen und Skalierungsänderungen, Verzerrung usw. annehmen. Diese Handlungen und Animationen stellen Grafikeigenschaften wie auch Benutzerdialogeigenschaften für das Grafikelement **74** bereit. Die Anbindungen **80**, die statische oder feste Anbindungen oder Anbindungen, die Pseudonyme benutzen, sein können, definieren die Art und Weise, auf die die Parameter oder Eigenschaften **78** an Daten, Etikette oder sonstige Instanzen in der Laufzeitumgebung **72** anzubinden sind, wenn das Grafikelement **74** als teil einer Anzeige in der Laufzeitumgebung **72** implementiert wird. Im Grunde wird durch die Anbindungen **80** für jedes Grafikelement **74** die Art und Weise festgelegt, auf die das Grafikelement **74** an eine oder mehrere Instanzen oder Datenelemente, die sonstwo in der Anlagenumgebung definiert sind, angebunden ist und damit eine Schnittstelle zwischen der eigentlichen Laufzeitumgebung **72** und dem Grafikelement **74** definiert.

**[0056]** Nach der Darstellung in der [Fig. 3](#) enthält jedes der Grafikanzeigeobjekte **76** zahlreiche Komponenten wie beispielsweise einen Verweis auf oder eine Kopie eines oder mehrerer Grafikelemente **81**, Verbinderelemente **82**, Handlungen und Animationen **83**, Eigenschaften **84** und Anbindungen **85**. Allge-

mein gesagt kann eine Grafikanzeige **76** eine Anzeige sein, die die Wechselwirkung verschiedener Grafikelemente **81** darstellt, die optisch mit Rohre, Leitungen, Förderbänder usw. darstellenden Verbinder-elementen **82** miteinander verbunden sein können. Solche Verbinderobjekte sind in der US-Veröffentlichung 2004/0153804 beschrieben. Eine gestrichelte Linie in der [Fig. 3](#) zeigt einen Verweis auf eines der Grafikelemente **74** durch das Grafikanzeigeobjekt **76a**. Es versteht sich, daß die Grafikanzeige **76**, die auf ein Grafikelement **74** Bezug nimmt, alle Eigenschaften, Parameter, Handlungen und Animationen usw. dieses Grafikelements **74** enthält. Ähnlich den Grafikelementen **74** kann jede Grafikanzeige **76** eine oder mehrere damit verbundene zusätzliche Handlungen oder Animationen enthalten, die beispielsweise Animationen auf der Anzeige, Benutzeroberflächendialoge, Datenmanipulationen usw. durchführen. GleichermäÙen kann jeder Grafikanzeige **76** eine beliebige Anzahl von der Anzeige zugeordneten Eigenschaften enthalten und diese Eigenschaften definieren typischerweise Eigenschaften von Einheiten, Bereichen oder sonstigen Gruppen von in der Anzeige dargestellten Elementen. Natürlich definieren die Anbindungen **85** die Art und Weise, auf die die Grafikanzeige **76** an eine oder mehrere anderswo in der Anlagenumgebung definierte Instanzen oder Datenelemente angebinden ist und definieren damit eine Schnittstelle zwischen der eigentlichen Laufzeitumgebung **72** und der Grafikanzeige **76**.

**[0057]** Nach ihrer Erstellung können die Grafikelemente **74** und die Grafikanzeigen **76** auch, beispielsweise einen beliebigen der Arbeitsplätze **20–23** der [Fig. 1](#), an die Laufzeitumgebung **72** angebinden und darin ausgeführt werden. Insbesondere kann nach Erstellung eines Grafikelements **74** oder einer Grafikanzeige **76** als Klassenobjekt und der Speicherung in der Datenbank **52** dieses Element oder diese Anzeige als eigentliches Laufzeitobjekt konkretisiert werden und in der Laufzeitumgebung **72** ausgeführt werden. Wie durch Block **86** dargestellt werden durch das Konkretisierungsverfahren die in den Objekten **74** und **76** definierten Anbindungen aufgefüllt, was unter Verwendung einer oder mehrerer Auflösungstabellen erreicht werden kann, die mit zutreffenden Variablennamen, Etiketten, Pseudonymen usw. in der Prozeßanlage oder dem Prozeßsteuerungssystem beladen werden können, um eine spezifische Verbindung zwischen eigentlichen Instanzen in der Prozeßanlage und den auf einer Anzeigevorrichtung in der Anlage **10** ablaufenden Grafikobjekten bereitzustellen. Als Teil des Anbindungsverfahrens verbinden sich die Objekte **74** und **76** mit Datenquellen in der Prozeßanlage wie durch die Auflösungstabelle definiert und erlangen dadurch Zugang zur Anlage, um logisch und kommunikativ mit der Prozeßanlage **10** verbunden zu sein.

**[0058]** Wie durch die Blöcke **87** dargestellt, kann ein

Anzeigeelement **74** oder eine Grafikanzeige **76** in oder als Teil einer Anzahl verschiedener Funktionen in der Laufzeitumgebung **72** einschließlich einer Konfigurationsanzeige, einer Steuerungsbedieneranzeige, einer Wartungsanzeige und einer Simulationsanzeige, um nur einige zu nennen, ausgeführt werden. Zusätzlich können die Anzeigeobjekte **74** und **76** bei der Durchführung von Funktionen auf Systemebene genutzt werden, z. B. solche, die Daten von verschiedenen der in [Fig. 2](#) dargestellten Funktionsebenen benutzen, einschließlich beispielsweise von prädiktiven Steuerungs- oder prädiktiven Wartungsfunktionen, Fehlererkennung auf Systemebene, Diagnose usw. In der Tat können die Anzeigen **76** nach ihrer Erstellung in der Konfigurationsumgebung **70** und Speicherung in der Datenbank **52** für eine Anzahl unterschiedlicher Aktivitäten benutzt werden. Weiterhin können die Anzeigeobjekte **74** und **76** auch auf jeder gewünschten Anzeige- oder Computerhardware wie beispielsweise einem Arbeitsplatz **90**, einem Laptop-Computer **91**, einer Handvorrichtung **92**, wie beispielsweise einem PDA (personal data assistant), einer Telefonvorrichtung usw. oder einer sonstigen Spezialitätsanzeige **93** wie beispielsweise einer Großschirmanzeige mit mehreren Monitoren usw. ausgeführt werden. Wenn gewünscht kann eine einzelne Grafikanzeige **76** schichtweise angeordnet sein, um eine oder mehrere Ansichten wie beispielsweise eine Konfigurationsansicht, eine Bedieneransicht, eine Wartungsansicht und eine Simulationsansicht zu enthalten. Als Alternative können getrennte Grafikanzeigen **76** konfiguriert sein, um diese getrennten Ansichten unter Verwendung der gleichen oder ähnlicher Grafikelemente **81** bereitzustellen, um ein konsequentes Aussehen und einen konsequenten Eindruck über für diese verschiedenen Funktionen erstellten Anzeigen bereitzustellen.

**[0059]** Wie die durch den Block **95** dargestellt, kann ein zur Laufzeitumgebung **72** zu portierendes Grafikelement **74** oder eine Grafikanzeige **76** kopiert und konkretisiert werden und in die Laufzeitmaschine eingeladen werden. Allgemein gesagt ist es wünschenswert, daß das Anzeigeobjekt **74** oder **76** nur dann an die Laufzeitumgebung **72** angebinden wird, wenn es aufgerufen wird oder tatsächlich auf einer Laufzeitmaschine ausgeführt wird, was hier als Laufzeitanbindung bezeichnet wird. Das bedeutet, die Auflösungstabelle für jedes der konkretisierten Objekte wird nur dann ausgefüllt oder an die Laufzeitumgebung angebinden, wenn das Anzeigeobjekt tatsächlich läuft oder auf einem Laufzeitcomputer ausgeführt wird. Durch dieses Verfahren wird sichergestellt, daß das Anzeigeobjekt einschließlich seiner Visualisierungen, Steuerungen, Skripts, usw. nur dann abläuft und damit Verarbeitungsleistung benutzt, wenn die Visualisierungen) des Objekts tatsächlich auf einem Bildschirm weitergegeben werden. So wird das Anzeigeobjekt vorzugsweise nur dann an die Laufzeitumgebung **72** angebinden, wenn dieses Objekt tat-

sächlich auf einem Laufzeitcomputer abläuft, was bedeutet, daß die Anzeigeobjekte **74** und **76** intermittierend mit der Laufzeitumgebung **72** auf eine Weise verbunden sein können, die durch Aktivitäten der Benutzer definiert wird, die die durch diese Objekte erstellten Visualisierungen betrachten. Insbesondere können diese Objekte zu den Zeiten an eine Laufzeitumgebung **72** angebunden werden, zu denen sie betrachtet werden müssen, und können entbunden oder freigegeben werden, wenn sie nicht von einem Benutzer betrachtet werden, wie beispielsweise wenn ein Benutzer ein Schirmbild minimiert oder schließt, in dem diese Objekte eine Visualisierung bereitstellen.

**[0060]** Die Anzeigeobjekte **74** und **76** sind daher Objekte, die in einer alleinstehenden Umgebung, d. h. der Konfigurationsumgebung **70** erstellt werden können, die aber an andere Objekte oder Datenstrukturen angebunden und damit verbunden werden können, die in der Prozeßanlagenumgebung oder irgendeiner in der Prozeßanlagenumgebung ablaufenden Anwendung definiert sind, einschließlich beispielsweise von Objekten, Datenstrukturen, Anwendungen usw., die in einer beliebigen Steuerungs-, Simulations-, Wartungs- oder Konfigurationsumgebung definiert sind. Weiterhin können die Anzeigeobjekte **74** und **76** nach ihrer Erstellung direkt, über direkte Verweise, Variablen oder in einer Auflösungstabelle definierte Etikette oder indirekt durch Verwendung von Pseudonymen, Variablen und Parametern angebunden sein, die aufgelöst werden können entweder, wenn das Anzeigeobjekt **74** oder **76** heruntergeladen oder in der Laufzeitumgebung **72** konkretisiert wird oder in manchen Fällen wenn das Anzeigeobjekt **74** oder **76** tatsächlich in der Laufzeitumgebung **72** abläuft.

**[0061]** Der Anzeigeeditor **50** der [Fig. 3](#) kann die Erstellung der Anzeigeobjekte **74** und **76** auf verschiedenen Detailebenen ermöglichen, um die Leichtigkeit der Verwendung und die Vielseitigkeit der Anzeigeobjekte **74** und **76** zu verbessern. Beispielsweise können Grafikelemente **74** zuerst erstellt werden, um die Eigenschaften und Operationen von primitiveren physikalischen und logischen Instanzen zu definieren, und Grafikanzeigen **76** können dann durch Verbindung eines oder mehrerer Grafikelemente **74** miteinander zur Erstellung von höheren oder komplizierteren Anzeigen erstellt werden, die kompliziertere physikalische oder logische Instanzen oder Gruppen von physikalischen oder logischen Instanzen darstellen. Natürlich können sowohl Grafikelemente **74** als auch Grafikanzeigen **76** in verschiedenen unterschiedlichen Kategorien gespeichert und darauf zugegriffen werden, um die Erstellung von höheren Anzeigeobjekten einfacher für den Benutzer zu gestalten.

**[0062]** Figur zeigt ein beispielhaftes Schirmbild **100**,

das durch den Anzeigeneditor **50** erstellt werden kann. Das Schirmbild **100**, das im Vorgang der Erstellung eines Grafikelements für eine Pumpe dargestellt ist, enthält einen Haupteditierteil **102**, einen Elementhierarchieteil **104**, einen Eigenschaftsdefinitionsteil **106** und einen Visualisierungsteil **108**. Der Haupteditierteil **102** bietet Arbeitsraum für den Benutzer oder Konstrukteur zum Definieren oder Erstellen einer Visualisierung für das Grafikelement und damit zum Definieren der optischen Eigenschaften des Grafikelements, im vorliegenden Fall eine mit einer Visualisierung **109** dargestellte Pumpe. Allgemein gesagt bietet der Elementhierarchieteil **104** unter Verwendung einer hierarchischen Ansicht oder einer Baumstruktur die der Visualisierung **109** im Haupteditierteil **102** zugeordneten Komponenten. Im Beispiel der [Fig. 4](#) zeigt der Hierarchieteil **104**, daß die im Haupteditierteil **102** definierte Visualisierung **109** Grundelemente oder Teilelemente eines Kreises (mit Zirkel **1** bezeichnet) und zwei Rechtecken (mit **Rect1** und **Rect2** bezeichnet) enthält. Obwohl dies in [Fig. 4](#) nicht dargestellt ist, kann der Hierarchieteil **104** Anzeigen von Animierungen, Handlungen und anderen Anzeigenmerkmalen wie beispielsweise Skripts, optische Trigger usw. enthalten, die für die Visualisierung **109** definiert sind.

**[0063]** Der Eigenschaftsdefinitionsteil **106** zeigt alle Eigenschaften einschließlich innerer Eigenschaften, die gegenwärtig für das der im Editierteil **102** gezeigten Visualisierung **109** zugeordnete Grafikelement definiert sind. Das Beispiel des Schirmbilds **100** der [Fig. 4](#) zeigt zwei Eigenschaften einschließlich einer Eigenschaft **IsOn**, die definiert, ob die der Visualisierung **109** zugeordnete Pumpe ein- oder ausgeschaltet ist, und eine Eigenschaft **Speed**, die die Geschwindigkeit der der Visualisierung **109** zugeordneten Pumpe definiert. Der Benutzer oder Konstrukteur könnte dem Grafikelement andere Eigenschaften und Parameter durch Definieren der Namen, Arten und Anbindungen anderer Variablen, Eigenschaften usw. im Eigenschaftsdefinitionsteil **106** hinzufügen, um damit andere Aspekte des Grafikelements zu definieren. Die im Teil **106** der [Fig. 4](#) gezeigten zwei Eigenschaften sind als Boolesche und Gleitkomma-Variablen aufgeführt. Stattdessen oder zusätzlich könnten andere Arten von Variablen benutzt werden. So könnten die im Teil **106** definierten Eigenschaften beispielsweise Felder, Tabellen, numerierte Listen oder sonstige Arten von Variablen oder Datenstrukturen sein.

**[0064]** Wenn gewünscht könnten dem im Schirmbild **100** definierten Grafikelement mehrfache Visualisierungen zugeordnet sein. Die unterschiedlichen dieser Visualisierungen könnten im Visualisierungsteil **108** dargestellt und einzeln zur Plazierung im Haupteditierteil **102** ausgewählt werden. Beispielsweise sind im Visualisierungsteil **108** der [Fig. 4](#) zwei Visualisierungen **110A** und **110B** dargestellt, aber es könn-

te eine beliebige andere Anzahl und Art von Visualisierungen für das erstellte Grafikelement definiert werden. Während des Vorgangs der Erstellung des Grafikelements könnte jede Visualisierung in den Haupteditierteil **102** plaziert werden, beispielsweise durch rechtes Anklicken oder Doppelanklicken an eine Anzeige der Visualisierung im Teil **108**, Ziehen und Ablegen der Anzeige der Visualisierung auf dem Editierteil **102**, usw. Nach Plazierung im Haupteditierteil **102** kann eine Visualisierung editiert werden, um ihre Anzeigeneigenschaften zu definieren oder neu zu definieren. Allgemein gesagt wird eine der Visualisierungen als Vorgabevisualisierung für das erstellte Grafikelement eingestellt oder definiert. Diese Vorgabeeinstellung kann auf irgendwelche Weise angezeigt werden, wie beispielsweise, daß die Vorgabevisualisierung hervorgehoben ist, mit einer gestrichelten Linie im Visualisierungsteil **108**, wie für die Visualisierung **110A** in der [Fig. 4](#) dargestellt, umgeben ist oder auf irgendwelche andere Weise.

**[0065]** Allgemein gesagt können die durch den Elementeditor **50** erstellten Visualisierungen aus einer oder mehreren Zusammensetzungen verschiedener Formen bestehen, die auf definierte Weise plaziert oder zusammen gruppiert werden. Diese Visualisierungen können daher als Formzusammensetzungen bezeichnet werden. Beispielsweise können Formzusammensetzungen Kreise, Linien, Punkte, Mehrecke, Quadrate, Rechtecke, Dreiecke und andere einfache Grafikformen umfassen. Wenn sie auf diese Weise definiert sind, können an jede der unterschiedlichen Formen, aus denen eine Formzusammensetzung besteht, getrennte Handlungen oder Animierungen angelegt werden oder diesen zugeordnet werden. Natürlich können Formzusammensetzungen ausführlichere künstlerische Wiedergaben von Elementen umfassen. Zum Definieren oder Aufbauen einer Formzusammensetzung kann ein Benutzer oder Konstrukteur dem Haupteditierteil **102** eine beliebige Anzahl von Grund-Grafikelementen hinzufügen und diese Grundelemente auf jede gewünschte Weise zusammen gruppieren. Nach ihrer Erstellung definieren Formzusammensetzungen die Visualisierung für ein eigentliches Objekt, das in der Laufzeit als XAML-Objekt implementiert sein könnte, das als Visualisierung auf einem Bildschirm oder einer Anzeige angezeigt wird, wenn das Grafikelement in einer Laufzeitumgebung benutzt wird. Die die Formzusammensetzung bildenden Formen oder Grundelemente können in einer Zusammensetzungshierarchie im Hierarchieteil **104** dargestellt sein.

**[0066]** Die Erstellung mehrfacher Visualisierungen für ein einzelnes Grafikelement ermöglicht die Verwendung von unterschiedlichen Visualisierungen in unterschiedlichen Zusammenhängen oder für unterschiedliche Verwendungszwecke, so daß für die verschiedenen Zwecke in der Anlage erstellte Anzeigen aus dem gleichen Grafikelement erstellt werden kön-

nen (d. h. aus dem gleichem grafischen Klassenobjekt erstellt sein können), während sie unterschiedliche Visualisierungen eines gemeinsamen Elements anzeigen. Beispielsweise können verschiedene der Visualisierungen **110A**, **110B** usw. in verschiedenen Funktionszusammenhängen benutzt werden, so daß eine erste Visualisierung **110A** benutzt werden kann, wenn das Grafikelement als Teil der Anzeige eines Steuerungsbedieners benutzt wird. Eine zweite Visualisierung **110B** kann benutzt werden, wenn das Grafikelement als Teil einer Wartungsanzeige benutzt wird, und eine (nichtgezeigte) dritte Visualisierung kann genutzt werden, wenn das Pumpenelement in einer Simulationsanzeige benutzt wird. Andererseits können verschiedene Visualisierungen für Anzeigezwecke auf verschiedenen Arten von Anzeigevorrichtungen benutzt werden. Beispielsweise könnte eine erste Visualisierung eines Grafikelements zur Verwendung auf einem typischen Computer oder Arbeitsplatz geeignet sein, während eine andere Visualisierung erstellt werden könnte, um vorteilhafterweise auf einer Handvorrichtung wie beispielsweise einem PDA oder einer Telefonvorrichtung benutzt zu werden, während eine noch weitere Visualisierung für Großbildanzeigen oder einen Monitor mit mehrfachen Bildschirmen erstellt werden könnte. Es versteht sich, daß unterschiedliche Visualisierungen für ein Grafikelement auf Grundlage der Größe der Anzeige maßgeschneidert werden können, auf der die Visualisierung anzuzeigen ist, so daß eine erste Visualisierung erstellt werden könnte, die für einen größeren Bildschirm geeignet ist, wie beispielsweise einen typischen Computerbildschirm, und eine zweite Visualisierung kann geeigneterweise auf einem Bildschirm erstellt werden, der bedeutend kleiner ist, wie beispielsweise ein Anzeigebildschirm einer Handvorrichtung wie ein PDA oder eine drahtlosen Telefonvorrichtung. So können unterschiedliche Visualisierungen für das gleiche Grafikelement in unterschiedlichen Laufzeitvorrichtungen benutzt werden.

**[0067]** Andererseits können unterschiedliche Visualisierungen zu unterschiedlichen Zeiten im Entwicklungszyklus des Grafikelements dem Grafikelement hinzugefügt werden. Beispielsweise könnte ein Konfigurationsingenieur, der vielleicht nicht die Geschicklichkeit zum Zeichnen oder Erzeugen von ansprechenden Visualisierungen hat, eine erste Grundvisualisierung erzeugen, die vorläufiger und grundlegender Beschaffenheit ist (wie beispielsweise eine strichmännchenartige Zeichnung) bei der ersten Erstellung des Grafikelements und kann diese Visualisierung als Teil des Grafikelements zu dieser Zeit speichern. Später könnte ein erfahrener Zeichner oder Künstler eine ausführlichere und künstlerisch ansprechendere Visualisierung erstellen und diese zweite Visualisierung als alternative Visualisierung dem Grafikelement hinzufügen. Wenn gewünscht, kann die zweite Visualisierung zu dieser Zeit als Vorgabevisualisierung eingestellt werden und diese zweite Visualisie-

rung kann durch alle Laufzeit-Grafikelemente weitergegeben werden, die aus dem Haupt- oder Klassen-Grafikelement konkretisiert worden sind, so daß diese zweite Visualisierung in den Laufzeitanzeigen erscheint oder zur Verfügung steht.

**[0068]** In einem weiteren Beispiel können unterschiedliche Visualisierungen für das gleiche Grafikelement bereitgestellt werden, um unterschiedliche Themen, Grafikstandard, Normen oder Anzeigestile zu unterstützen. Bekannterweise benutzen unterschiedliche Industrien typischerweise unterschiedliche Grafiknormen oder Grafikstandards zur Darstellung von Pumpen, Ventilen, Sensoren und sonstigen Prozeßanlageninstanzen. So benutzt die Öl- und Gasindustrie andere Grafikstandards als die pharmazeutische Industrie. Unter Verwendung des Elementeditors **50** ist es möglich, für jedes Grafikelement unterschiedliche Visualisierungen zur Unterstützung mehrfacher Grafikstandards oder -normen wie beispielsweise der Öl- und Gasnorm und der pharmazeutischen Norm bereitzustellen. Auf diese Weise kann das gleiche Grafikelement in Grafikanzeigen benutzt werden, die zur Unterstützung unterschiedlicher Standards oder Normen oder Anzeigetemen erstellt worden sind, zum Beispiel in unterschiedlichen Industrien annehmbare oder nützliche Anzeigen. Weiterhin können unterschiedliche grafische Stile wie beispielsweise künstlerische Stile in unterschiedlichen Visualisierungen benutzt werden, damit ein Benutzer Anzeigen verschiedener künstlerischer Typen erstellen kann. Dies sind natürlich nur einige Beispiele der Vorteile der Bereitstellung von mehrfachen Visualisierungen für das gleiche Grafikelement und es bestehen auch andere Verwendungen für diese mehrfachen Visualisierungen.

**[0069]** Wenn gewünscht können für das Grafikelement eine oder mehrere grafische Verhaltensweisen definiert oder ihm zugewiesen sein. Insbesondere kann ein Konstrukteur oder Ersteller Animierungen wie beispielsweise Drehungen, Linearverschiebungen, Hintergrundveränderungen, Farbveränderungen, Größenveränderung, Farbgradientenanimierungen, Opazitätsanimierungen, Schriftarteigenschaftsanimierungen, Videos und Videomerkmale wie beispielsweise Start-/Stop-Merkmale, zweidimensionale oder dreidimensionale Änderungen usw. für jede Visualisierung des Grafikelements definieren, wenn die Visualisierung auf einem Bildschirm angezeigt wird. Zur Hinzufügung dieses dynamischen Verhaltens kann der Benutzer ein Grafikelement auswählen und sich entscheiden, der Visualisierung eine Animation (auch Animationsroutine) hinzuzufügen. An dieser Stelle kann der Benutzer die ausgewählte Animierung betreffende Konfigurationsinformation eingeben oder das gewünschte Verhalten unter Verwendung von Skripten definieren, die beispielsweise mit Dialogboxen, Ausdruckseditoren usw. eingegeben werden können. Wenn sie definiert sind, können diese

Skripts in der Hierarchie des Hierarchieteils **104** erscheinen. Im Prinzip ist ein Skript ein Programm oder eine Routine, die als Teil des Anzeigeelements abläuft oder ausgeführt wird, wenn die Visualisierung für das Anzeigeelement während einer Laufzeitaktivität auf einem Bildschirm betrachtet oder dargestellt wird. Obwohl es nicht notwendig ist, können diese Verhaltensroutinen oder Skripts an eines oder mehrere der für das Grafikelement definierten Eigenschaften oder Parameter angebunden sein und diese bearbeiten. Beispielsweise kann ein Skript einem Grafikelement zugeordnet sein, um eine Farbe in der Visualisierung des Grafikelements auf Grund des Wertes eines der im Eigenschaftsteil **106** des Grafikelements definierten Eigenschaften zu ändern. Beispielsweise kann sich die Farbe der Pumpenvisualisierung **109** zu grün ändern, wenn die Eigenschaft IsOn (die an die Laufzeitumgebung angebunden ist) wahr ist, d. h. wenn die Pumpe eingeschaltet ist. Als weiteres Beispiel kann ein Skript für das Pumpenelement definiert sein, das den Wert der Eigenschaft Speed der Pumpe mit einem Sollwert vergleicht und wenn der Wert der Eigenschaft Speed über einem gewissen Pegel liegt, eine Grafikanimation irgendeiner Art in der Visualisierung **109** veranlaßt. Diese Animation kann beispielsweise die Pumpe rot werden lassen, die Pumpe drehen, Drehen eines Motors in der Pumpe anzeigen, Pulsieren oder Vibrieren der Pumpenvisualisierung veranlassen usw. umfassen.

**[0070]** Die für eine Visualisierung, beispielsweise zur Bereitstellung von Animierung und anderen Verhaltensweisen für die Visualisierung, definierten Skripts können so ausgelegt sein, daß sie auf einzelne der Formen oder Grundelemente in einer die Visualisierung bildenden Formzusammensetzung oder auf mehrere der Formen in den Formzusammensetzungen einwirken. Da die Skripts auf die inneren Eigenschaften einwirken oder diese verwenden, die an tatsächliche physikalische Elemente in der Laufzeitumgebung angebunden sind, ermöglichen diese Skripts, daß sich die Visualisierung auf Grundlage tatsächlicher Anlagenoperationen oder auf Grundlage von Daten von anderen Bereichen der Prozeßanlage, die die Eigenschaften einer dem Grafikelement zugeordneten wirklichen Instanz widerspiegeln, ändert.

**[0071]** Es versteht sich, daß unterschiedliche Arten von Grafikelementen unterschiedliche Grafiken aufweisen können, die zur Verwendung mit diesen geeignet sind. So sind die Verhaltensroutinen, die für ein Grafikelement bereitgestellt werden, nicht durch die hier gebotenen Beispiele begrenzt. Beispielsweise können Grafikelemente für rotierende Geräte Skripts enthalten, die Schwingungsgrafiken, Bewegung, Farbveränderungen usw. bereitstellen, während Grafikelement für Geräte wie Sensoren Skripts enthalten können, die einen Grenzüber- oder -unterschreitungszustand darstellen, eine Notwendigkeit

für Eichungszustand usw. Natürlich können beliebige Eigenschaften für Grafikelemente benutzt oder definiert werden und diese Eigenschaften beruhen allgemein auf der Art von Instanz, die dargestellt wird. Auch versteht es sich, daß verschiedene Skripts für jede der unterschiedlichen Visualisierungen bereitgestellt werden können, um unterschiedliche Verhaltensroutinen für die verschiedenen Visualisierungen bereitzustellen. Andererseits können einige Skripts zur Bereitstellung von Verhaltensroutinen für jede der einem Grafikelement zugeordneten Visualisierungen benutzt werden.

**[0072]** Zusätzlich können Skripts oder andere Routinen einem Grafikelement zugeordnet sein, um gewisse Zustände über die zugehörige physikalische Instanz bzw. physische Entität auf Grundlage eines oder mehrerer der Parameter für das Grafikelement zu erkennen. Diese Zustände können die Erkennung von Zuständen der Instanz einschließlich eines physikalischen Zustandes, eines Zustandes wie beispielsweise eines Kommunikationszustandes, eines Vorrichtungszustandes, eines Leerzustandes usw. im Zusammenhang mit der Instanz umfassen. Alle diese erkannte Zustände können unter Verwendung einer Animierung oder sonstigen Handlung oder Verhaltenroutine, die in den durch das Skript erzeugten Zustand oder Wert eingebunden ist, auf der Grafik widergespiegelt werden. Beispielsweise kann ein Skript für ein Pumpengrafikelement bereitgestellt werden, das einen Überhitzungszustand der zugeordneten Pumpe auf Grundlage eines oder mehrerer Parameter für dieses Element, d. h. eines oder mehrere der im Eigenschaftsteil **106** der [Fig. 4](#) definierten oder ausgedrückten Parameter erkennt. Als weiteres Beispiel kann ein Skript für ein Grafikelement bereitgestellt werden, um übermäßige Schwingung oder irgendeinen sonstigen Zustand der Pumpe zu erkennen. Wenn ein Zustand wie beispielsweise Überhitzung oder übermäßige Schwingen erkannt wird, kann eine Animierung oder sonstige, dem Grafikelement zugeordnete Handlung auf den Zustand einwirken, um eine grafische Anzeige des Zustandes innerhalb der oder auf der Visualisierung des Grafikelements bereitzustellen.

**[0073]** [Fig. 5](#) zeigt eine weitere beispielhafte Schirmbildanzeige **112**, die durch den Anzeigeneditor **50** erzeugt werden kann. Ähnlich dem Schirmbild **100** der [Fig. 4](#) enthält die Schirmbildanzeige **112** einen Haupteditierteil **114**, eine Palettenansicht **116**, eine Hierarchieansicht **118** und eine Eigenschaftsansicht **120**. Im Haupteditierteil **114** ist eine erste Visualisierung eines Reaktorelements **122** dargestellt. Wie in der Hierarchieansicht **118** gezeigt, ist der Titel der Elements Reaktor **1** und dieses Element enthält drei Visualisierungen mit der Bezeichnung Visual1 (die Vorgabevisualisierung), Visual2 und Visual3. Wie unter der Rubrik Visual1 in der Hierarchieansicht **118** angezeigt, besteht die erste Visualisierung aus

einem Entwurfshintergrund Canvas, der ein Rechteckelement und ein Ellipseelement enthält. Die gegenwärtig definierten Eigenschaften, im vorliegenden Fall ein Name, eine Höhe und eine Breite der Visualisierung, sind in der Eigenschaftsansicht **120** aufgeführt. Wenn eine Visualisierung in der Hierarchieansicht **118** ausgewählt wird, werden alle der Visualisierung zugeordneten abgeleiteten Grundelemente oder Elemente in der Editieransicht **114** dargestellt und die Eigenschaften des gegenwärtig ausgewählten Elements werden in der Eigenschaftsansicht **120** angezeigt.

**[0074]** Im Schirmbild **112** enthält die Palettenansicht **116** eine Anzahl von Grundelementen, die zur Erstellung einer Visualisierung benutzt werden können. Beispielsweise enthält die Palettenansicht **116** eine Menge von Grund-UI-Elementen (user interface, Benutzeroberfläche) wie beispielsweise Knöpfen, Textkästen, Schiebern, usw., eine Menge von Grundfeldern und eine Menge von Grundformen. Die definierten Felder können ein Entwurfssfeld enthalten, das einen Bereich definiert, in dem der Benutzer explizit Elemente durch Koordinaten bezüglich des Entwurfsbereichs positionieren kann, ein Andockfeld, das einen Bereich definiert, in dem der Benutzer Elemente entweder horizontal oder senkrecht zueinander anordnen kann und ein Flußfeld, das zum Aufbrechen, Umbrechen und Ausrichten seines Inhalts innerhalb eines Flußfeldbereichs mit angezeigter Flußrichtung genutzt werden kann. Flußrichtungen können beispielsweise jede Kombination von oben, rechts, links und unten wie beispielsweise links nach rechts und oben nach unten oder rechts nach links, oben nach unten usw. sein. Weiterhin können die Grundformen in der Palettenansicht **116** ISA-Symbole (Instrument Society of America), Sendersymbole, Ventilsymbole, PI & D-Diagrammsymbole oder sonstige Steuerungssymbole usw. umfassen die alle zum Aufbauen eines Grafikelements benutzt werden können.

**[0075]** Natürlich können andere Grundelemente wie beispielsweise Grundsteuerungselemente, Vorrichtungen usw. als Grundelemente in der Palettenansicht **116** zur Verwendung zur Erstellung von Visualisierungen für ein definiertes Grafikelement bereitgestellt werden. Auch kann die Palettenansicht **116** eine Liste von vom Benutzer definierten Kategorien oder Elementen bereitstellen, die erstellt werden können, damit Benutzer beliebige andere nützliche Formen aus der Palettenansicht **116** in die Editieransicht **114** ziehen können. Wenn gewünscht kann Zugang zu solchen vom Benutzer definierten Kategorien auf die die erstellenden Benutzer begrenzt sein und diese Kategorien und Elemente können in einer einem bestimmten Benutzer zugeordneten Benutzervorzugsdatei gespeichert sein. Jedoch können eingebaute Kategorien und Palettenegegenstände in der Datenbank gespeichert sein und global für alle Benutzer zur Verfügung stehen. Auf alle Fälle kann die Palet-

tenansicht **116** dazu benutzt werden, Bibliotheken von Grundelementen anzuzeigen oder zugänglich zu machen, die zum Aufbauen eines Grafikelements benutzt werden können, und diese Bibliotheken können nach Wunsch verriegelt, nach Version geordnet, auf bestimmte Benutzer begrenzt usw. sein.

**[0076]** Wie oben bemerkt können beliebigen der Visualisierungen für ein Grafikelement Animierungen und/oder Handlungen zugeordnet werden und solche Animierungen oder Handlungen können in einer Handlungs-/Animierungsansicht **123** des Schirmbildes **112** gezeigt sein. Wenn Visualisierungen Animierungen oder Handlungen enthalten, können diese Animierungen oder Handlungen auch in der Hierarchie mit Sondersymbolen wie beispielsweise Sternen usw. angezeigt sein. Wenn sie in der Hierarchieansicht **118** angewählt werden, werden alle für eine Visualisierung oder ein Teilelement einer Visualisierung definierten Handlungen oder Animierungen in der Handlungs-/Animierungsansicht **123** dargestellt. Handlungen oder Animierungen können durch Definieren solcher Handlungen oder Animierungen in der Ansicht **123** oder Zufügen solcher Handlungen oder Animierungen zu der Hierarchieansicht **118** erstellt und zugewiesen werden. Wenn ein Benutzer wünscht, eine Handlung oder Animierung zu erstellen oder zu editieren, kann der Editor **50** einen Dialog- oder Editierkasten zur Verfügung stellen, damit dieses Merkmal voll spezifiziert oder definiert werden kann. Natürlich können Handlungen oder Animierungen unter Verwendung von Skripts, optischen Triggern oder sonstigen Programmen definiert werden.

**[0077]** Während der Verwendung des Schirmbildes **112** kann ein Benutzer oder Konstrukteur eine Visualisierung eines Elements durch Ziehen und Ablegen oder sonstwie Anwählen unterschiedlicher Gegenstände in der Palettenansicht **116** und Anordnen dieser Gegenstände in der Editieransicht **114** zum Erstellen der gewünschten Visualisierung erstellen. Zur Bereitstellung beliebiger standardmäßiger Editierfunktionen wie beispielsweise Zufügung neuer Visualisierungen oder Animierungen, Löschen, Entfernen, Editieren von Animierungen, Anordnen von Grundelementen oder Elementen beispielsweise von hinten nach vorne, Ermöglichen, daß Verbindungselemente die Herstellung von Verbindungen zu der erstellten Visualisierung zeigen oder ermöglichen, Zusammengruppieren unterschiedlicher Grundelemente, so daß sie ihre Position zueinander halten, Zufügen von statischen Elementen wie Linien und Text usw. kann eine oder mehrere Werkzeugleisten **124** benutzt werden.

**[0078]** Nach der Darstellung in [Fig. 6](#) kann ein Grafikelement **130** mehrere Visualisierungen **132** oder damit verbundene optische Darstellungen aufweisen, wobei es sich versteht, daß die verschiedenen Visualisierungen **132** zu unterschiedlichen Zeiten oder in

unterschiedlichen Anzeigen benutzt werden können, in denen das Grafikelement **130** benutzt wird. Wie oben erläutert können beliebige der Visualisierungen **132** aus einer beliebigen Anzahl von Grundelementen **134** wie auch Animierungen und Handlungen **136** bestehen. Weiterhin kann das Grafikelement **130** eine beliebige Anzahl von Eigenschaften oder Parametern **138** enthalten, die in die Handlungen und Animierungen **136** eingebunden oder von diesen benutzt werden können, um Änderungen an den Visualisierungen **132** durchzuführen und den Visualisierungen **132** zugeordnete optische Trigger zu betätigen. Weiterhin können für die Visualisierungen **132** oder einzelnen Grundelemente, die eine Visualisierung **132** bilden, Handlungen für vordefinierte Ereignisse, beispielsweise Mausfahrereignisse, Mausclickereignisse usw. aufweisen. Diese Handlungen (die auch als Routinen bezeichnet werden) ermöglichen die Einstellung oder Definition von Ereignishandhabungen zur weiteren kundenspezifischen Einstellung des Verhaltens des Grafikelements **130** und um einem Benutzer den Dialog mit der Visualisierung **132** zu erlauben, um beispielsweise eine Änderung in der Laufzeitumgebung zu veranlassen. Insbesondere kann der Benutzer einen Dialog mit der Visualisierung **132** durch Eingeben von Werten oder anderen Nummern oder Informationen darin, Bewegen eines Elements auf der Visualisierung wie beispielsweise eine Schiebeleiste, oder Unternehmen irgendeiner sonstigen Handlung zur Veränderung beispielsweise einer Eigenschaft in der Visualisierung führen. Diese Visualisierungs-Eigenschaftsänderung kann über ein Skript oder direkt an eine Prozeßeingabe wie beispielsweise eine Prozeßlaufzeitvariable angebinden sein, um die Änderung dieser Variablen zu bewirken. Insbesondere kann die Visualisierungseigenschaft mit einer für das Grafikelement definierten Eigenschaft verbunden sein, die wiederum an eine Prozeßeingabe angebinden sein kann. Auf diese Weise kann ein Benutzer über eine Handlung oder Animierungsroutine mit der Visualisierung einen Dialog führen, um eine Änderung zu veranlassen oder eine Eingabe in den Prozeß oder sonstige Laufzeitumgebung wie beispielsweise eine Simulationsumgebung bereitzustellen.

**[0079]** Wenn gewünscht kann eine Visualisierung **132** mit dem Grafikelement **130** verbunden sein, indem sie entweder durch einen Benutzer angegeben oder programmatisch angegeben wird. Insbesondere kann das Grafikelement **130** eine Numerierungseigenschaft freigeben, die die programmatische Änderung der Visualisierung erlaubt, beispielsweise auf Grundlage der Darstellung des Herstellers des Elements durch das Grafikelement oder sonstige dem Grafikelement **130** zugeordnete Parameter wie beispielsweise den Zustand der dem Grafikelement zugeordneten Vorrichtung.

**[0080]** Wie oben bemerkt liefert oder zeigt die Ei-

genschaftenansicht **120** der [Fig. 5](#) die für den ausgewählten Gegenstand in der hierarchischen Ansicht **118** definierten Eigenschaften und Ereignisse und damit für den in der Haupteditieransicht **114** gezeigten Gegenstand. Ein Benutzer kann zwischen Eigenschaften, inneren Eigenschaften und Ereignissen in der Eigenschaftsansicht **120** unter Verwendung von beispielsweise Werkzeugleistenknöpfen umschalten. [Fig. 7](#) zeigt eine Eigenschaftsansicht **120A**, in der die inneren Eigenschaften gezeigt werden, im vorliegenden Falle mit einer Eigenschaft `IsOn` und einer Eigenschaft `Speed`. Im vorliegenden Fall zeigt die Eigenschaftsansicht **120A** die Datenart dieser Variablen und alle Vorgabeeinstellungen für diese Variablen. Wie oben bemerkt kann ein Benutzer zu dieser Eigenschaftsliste hinzufügen, aus ihr löschen oder sie editieren, um Eigenschaften mit einer beliebigen gewünschten Datenart einschließlich Aufstellungen, Tabellenstrukturen usw. zu definieren. Wenn gewünscht kann die Eigenschaftsansicht **120** auch irgendwelche Laufzeitanbindungen zeigen, die für diese Eigenschaften definiert sind, und diese Laufzeitanbindungen können feste Variablen oder Etikette sein oder Etikette unter Verwendung von Pseudonymen sein, die zur Laufzeit oder beim Herunterladen des Grafikelements zu einer Laufzeitmaschine ausgefüllt werden.

**[0081]** Die Werte der für ein Grafikelement definierten Eigenschaften können als Aktivierungszustände für Animierungen, Handlungen usw. benutzt werden und diese Verhaltensroutinen können durch ein oder mehrere dem Grafikelement zugeordnete Skripts definiert werden. Wenn beispielsweise ein Motor eine Eigenschaft `On` aufweist, die wahr ist, dann kann eine Animierung des arbeitenden Motors in der Grafikanzeige aktiviert werden. Weiterhin kann eine Grundelementeigenschaft, beispielsweise eine Rechteckfüllungseigenschaft eines Grundelements in der Form eines Rechtecks an eine Grafikelementeigenschaft angebunden sein, so daß Änderungen der Grafikelementeigenschaft die Grundelementeigenschaft bewirken. GleichermäÙen können Änderungen an einer Visualisierung an einer Eigenschaft angebunden sein, so daß Änderungen der Visualisierung Änderungen am Eigenschaftswert verursachen können.

**[0082]** Wenn gewünscht kann der Editor **50** einem Benutzer ermöglichen, eine oder mehrere Transformationsfunktionen anzugeben, um mehr wünschenswerte Anbindung zwischen Grafikelementeigenschaften und Animationen oder Handlungen bereitzustellen.

**[0083]** Beispielsweise könnte ein Benutzer wünschen, eine Grafikelementeigenschaft `TankLevel` (die die Höhe einer Flüssigkeit in einem Tank definiert) an eine Grundelementeigenschaftsrechteckfüllung anzubinden, um dadurch die Tankfüllhöhe grafisch mit einer Füllfarbe für ein als Teil der Visualisierung defi-

nirtes Grundelement darzustellen. In diesem Fall kann der Benutzer jedoch eine Transformationsfunktion definieren, die die Eigenschaft (`TankLevel`) in eine aufgezählte Menge oder einen aufgezählten Zustand umformt, so daß, wenn die Tankfüllhöhe zwischen einer ersten und zweiten Höhe liegt, die Rechteckfüllung auf grün, wenn die Tankfüllhöhe zwischen der zweiten und dritten Höhe liegt, die Rechteckfüllung auf gelb und wenn die Tankfüllhöhe größer als die dritte Höhe ist, die Rechteckfüllung auf rot eingestellt ist. Diese Transformationsfunktion kann als ein Skript oder sonstiges anderes Programm definiert sein, das mit dem Grafikelement abläuft und dazu benutzt werden kann, jede gewünschte Eigenschaftstransformation wie beispielsweise Ändern eines Eigenschaftswerts in eine Länge, einen Schriftartnamen, eine lokalisierte Kette, eine Dauer, eine Drehung, einen Farbgradienten, eine Opazität, ein Pinselmuster usw. zu veranlassen. GleichermäÙen können Transformationsroutinen Eingaben von einem Benutzer über eine Visualisierung in beliebige gewünschte Eigenschaftswerte umformen.

**[0084]** Als weiteres Beispiel können Grafikelementeigenschaften oder Parameter unter Verwendung von Transformationsfunktionen an externe Referenzen angebunden sein. Hier kann der Editor **50** einem Benutzer erlauben, eine oder mehrere Transformationsfunktionen anzugeben, die automatisch angewandt werden, wenn das Grafikelement in einer Laufzeitumgebung benutzt wird, um Quellwerte (Laufzeitvariablen) in Zielwerte (grafische Variablen) oder umgekehrt umzuwandeln. Beispielsweise könnte eine Geschwindigkeits-Zielvariable, die an eine Datenquelle angebunden sein könnte, die eine Geschwindigkeit als numerischen Wert bereitstellt, in eine Kettenvariable umgewandelt werden, wie beispielsweise eine Kette, die eines von "langsam", "mäÙig" und "schnell" auf Grundlage des Werts der Quellvariable anzeigt. Solche Transformationsfunktionen könnten auch zur Umwandlung von Einheiten oder Bereitstellung anderer Transformationen benutzt werden. Auf alle Fälle können diese Transformationsfunktionen als Skripts oder sonstige Programme implementiert werden und können für beliebige Zwecke zur Bereitstellung von Handlungen oder Animierungen oder zur sonstigen Beeinflussung von Anzeigeneigenschaften für ein Grafikelement benutzt werden.

**[0085]** Natürlich können alle gewünschten Eigenschaften für unterschiedliche Arten von Grafikelementen definiert werden, die für unterschiedliche physikalische Elemente in einer ProzeÙsteuerung erstellt wurden. Beispielsweise kann ein Pumpenelement einen Pumpenzustand (z. B. ein oder aus), eine Eingangsdruckeigenschaft, eine Ausgangsdruckeigenschaft, eine EinflueÙeigenschaft und eine AusflueÙeigenschaft umfassen. GleichermäÙen kann ein mit einem Stellglied benutztes Segelventil beispielsweise eine Nameneigenschaft, eine Eingangsdruckeigen-

schaft, eine Ausgangsdichteigenschaft, eine Einflußigenschaft, eine Ausflußigenschaft, eine Eingangsdruckeigenschaft, eine Ausgangsdruckeigenschaft, eine Eingangstemperatüreigenschaft, Ausgangstemperatüreigenschaft, eine Ventilstellungseigenschaft, eine Eigenschaft Ventil offen und Ventil geschlossen (die definieren kann, ob das Ventil vollständig offen oder geschlossen ist), einen Sollwert, einen Prozeßwert, eine Ventilart wie beispielsweise linear, schnellöffnend, gleichprozentual usw. umfassen. Natürlich soll diese Liste nicht umfassend sein. Weiterhin können beliebige dieser Eigenschaften mit Animierungen oder Handlungen für das Grafikelement verbunden sein.

**[0086]** [Fig. 8](#) zeigt eine zweite Eigenschaftsansicht **120B**, die zum Betrachten der Animierungen und Anbindungen benutzt werden kann, die den Grafikelementeigenschaften für ein ausgewähltes Element (im vorliegenden Fall allgemein das Reaktorelement) in der hierarchischen Ansicht **118** ([Fig. 5](#)) zugeordnet sind. Die Eigenschaftsansicht **120B** der [Fig. 8](#) zeigt die Art und Weise, auf die die Eigenschaft IsOn mit verschiedenen Komponenten der Visualisierungen für das Grafikelement in Verbindung steht. Insbesondere gibt es, wie durch den Tabelleneintrag **140** angezeigt, eine Animierung, die mit dem Hintergrund der Visualisierung in Verbindung steht und die auf Grundlage des Wertes der Eigenschaft IsOn fungiert. Ein Tabelleneintrag **142** zeigt die Anbindungen zwischen der Eigenschaft IsOn und kann zum Zugreifen auf Animierungen, Skripts und Transformationen benutzt werden, die die Eigenschaften IsOn mit dem Hintergrundelement oder Grundelement in der Visualisierung in Verbindung bringen. Im vorliegenden Fall kann der Benutzer zusätzliche Informationen durch Anwählen der Knöpfe in den Kabeleintragskästen **140** und **142** erhalten. Beispielsweise kann der Benutzer durch Anwählen des Knopfes im Tabelleneintrag **140** auf eine Animierung zugreifen, die bewirkt, daß der Hintergrund der Visualisierung aufgrund des Wertes des Parameters IsOn die Farbe ändert. Anklicken eines solchen Animierungsknopfes kann auch bewirken, daß der Editor **50** einen Eigenschaftsanimierungsdialogkasten öffnet, der die Verwaltung und Konfiguration der Animierungen ermöglicht. Weiterhin kann Anbindung durch Anklicken eines Anbindungsknopfes im Kasten **142** durchgeführt und verwaltet werden, der den Editor **50** veranlaßt, einen Dialogkasten zu öffnen, um Erstellung, Betrachtung und Editierung von Anbindungen zu ermöglichen. Natürlich könnten ähnliche Schirmbilder für andere Eigenschaften des Grafikelements bereitgestellt werden und die Liste von Eigenschaften, die in dem Eigenschaftsschirmbild **120B** gezeigt werden, werden von dem gegenwärtig ausgewählten Gegenstand in der Hierarchieansicht **118** abhängig sein.

**[0087]** Wiederum auf [Fig. 5](#) Bezug nehmend, wird die Eigenschaftsansicht **120** bei Anwahl einer Visua-

lisierung in der Hierarchieansicht **118** die Visualisierungseigenschaften wie beispielsweise einen Namen, eine ID, die eine einmalige Kennung für die Visualisierung bereitstellt, eine Beschreibung und ob diese Visualisierung als Vorgabevisualisierung für das Grafikelement eingestellt ist, anzeigen. Auch kann die Hierarchieansicht **118** oder die Eigenschaftsansicht **120** anzeigen, ob die Visualisierung verriegelt ist, d. h. ob sie abgeändert werden kann.

**[0088]** Zusätzlich wird die Eigenschaftsansicht **120**, wenn ein Verbinderelement aus dem Hierarchieteil **118** angewählt wird, die Eigenschaften dieses Verbinderelements zeigen, die eine Anzeige der Verbinderelementart (z. B. Flüssigkeitsleitung, elektrische Leitung, Gasleitung, kreisförmiger oder quadratischer Kanal, Förderband usw.) enthalten, ob die Verbindung in die oder aus der Vorrichtung führt (d. h. die Flußrichtung von Material bezüglich der durch das Grafikelement dargestellten Instanz), die Mindest- und Höchstzahl von erforderlichen oder zulässigen Verbindungen, die Breite und Höhe der Verbindung, die Lage der Verbindung bezüglich des Grafikelements wie beispielsweise oben, links usw.

**[0089]** Wenn in der Hierarchieansicht **118** ein Grundelement ausgewählt wird, zeigt die Eigenschaftsansicht **120** die Eigenschaften dieses Grundelements, zu denen beispielsweise eine ID, eine Kategoriebeschreibung, ob das Grundelement auswählbar und/oder sichtbar ist, eine Größe und Länge oder sonstige Definition der Form oder Beschaffenheit des Grundelements, eine Hintergrundfarbe und der Füllraum für das Grundelement usw. gehören. Natürlich wird jeder andere ausgewählte Gegenstand in der Hierarchieansicht **118** die Anzeige der Eigenschaften dieses Gegenstands in der Eigenschaftsansicht **120** veranlassen, und die Art und Beschaffenheit der Eigenschaften wird von der Beschaffenheit des ausgewählten Gegenstands abhängig sein.

**[0090]** Wenn gewünscht kann ein Ereignisblatt bereitgestellt werden, um die mit jedem oder allen der ausgewählten Hierarchiegegenstände in der Hierarchieansicht **118** verbundenen Ereignisse zu zeigen. Ein solches Ereignisblatt könnte Ereignisse enthalten, die eintreten, wenn ein Benutzer eine Handlung bezüglich eines Grafikelements durchführt wie beispielsweise ein "Anklick"-Ereignis und "Mausfahrereignis", die definieren, was geschieht, wenn ein Benutzer eine Visualisierung für das Grafikelement oder einen Teil derselben anklickt oder was geschieht, wenn der Benutzer den Mauscursor über die Visualisierung des Grafikelements oder einen Teil derselben positioniert. Als Teil dieses Ereignisblatts kann einem Benutzer Zugang zu einem Skripteditor zum Definieren oder Zugreifen auf ein Skript geboten werden, das bei Eintreten des Ereignisses (das hier als Routine bezeichnet wird) laufen wird.

**[0091]** Zusätzlich können, sofern gewünscht, kundenspezifische Ereignisse Grafikelementen zugewiesen werden. Kundenspezifische Ereignisse sind allgemein Ereignisse, die als Ergebnis irgendeines äußeren oder externen Ereignisses definiert sind oder Ereignisse, die Kommunikation mit einer externen Anwendung oder Datenquelle als Ergebnis einer von einem Benutzer des Grafikelements unternommenen Handlung erfordern. Im wesentlichen ist ein Grafiker-eignis eine Nachricht, die vom Grafikelement gesendet wird, um das Eintreten einer Handlung bezüglich der Visualisierung des Grafikelements zu signalisieren. Die das Grafikelement implementierende enthaltende oder Laufzeitanwendung kann diese Ereignisse unter Verwendung beispielsweise einer C#-Syntax oder Registrierung der Ereignisbenachrichtigung auf irgendeine bekannte Weise einfangen. In einem Beispiel kann die enthaltende Anwendung ein kundenspezifisches Überhitzungsereignis registrieren und eine Funktion meine Handhabung bereitstellen, die ein Skript oder sonstige Routine fährt, um einem Benutzer die Handhabung des Ereignisses bei seiner Aktivierung zu ermöglichen. Diese kundenspezifischen Ereignisse sind besonders nützlich, wenn das Grafikelement als Teil einer Steuerungsbedieneranzeige implementiert wird. In diesem Fall kann das Grafikelement selbst einen Ereigniszustand bestimmen, wie beispielsweise durch Vergleichen einer Pumpenhitzeeigenschaft (die an eine externe Messung angebunden ist) mit einem Sollwert oder sonstigen Grenze, und kann eine Ereignisreaktion durch Benachrichtigen des Benutzers über das Ereignis und Implementieren einer Anwendung, eines Skripts, eines Dialogs usw. aktivieren, damit der Benutzer auf das Ereignis, im vorliegenden Fall eine überhitzende Pumpe, reagieren kann. Um diesen Ereignisbehandlungsvorgang zu ermöglichen, erlaubt der Elementeditor **50** einem Benutzer die Definition von Ereignisargumenten, d. h. was als Ereignis erkannt wird und was sich aus dem Eintritt eines Ereignisses ergibt.

**[0092]** Natürlich können Grafikelemente, sofern gewünscht, standardmäßige Ereignisse bereitstellen, die von einer Avalon-Steuerungsklasse übernommen worden sind. Wie bekannt ist, sind die Avalon-Objekte oder Steuerungen Microsoft-definierte Grafiken für die Avalon-Benutzeroberflächenarchitektur für das Betriebssystem Microsoft Longhorn, das die Wiedergabe von Vektorgrafiken auf einer Benutzeroberfläche unterstützt. Diese Standardereignisse können beispielsweise das Anbinden der Betrachtung einer Frontplattenanzeige oder einer Vorrichtungsdetailanzeige an ein Klick- oder Doppelklick-(Maus-)Ereignis umfassen. Beispielsweise kann ein Klick- oder Doppelklick-Ereignis, das ein Ereignis ist, das einen Mausklick oder Doppelklick innerhalb der Grenzen der Visualisierung des Grafikelements berichtet, die Anzeige einer Frontplatte oder Detailanzeige für die durch die Visualisierung dargestellte Vorrichtung oder Instanz aktivieren und bietet dem Benutzer voll-

ständigere oder umfassendere Informationen über den Zustand, die Konstruktion, Herstellung usw. der dargestellten Instanz. Natürlich können andere Ereignisse, einschließlich Standardereignissen, für ein Ereignis Taste auf/ab definiert sein, das ein Ereignis ist, das meldet, daß eine Taste gedrückt wurde, Mausereignisse, die Mauseingabe, Maus schwebend, Mausbewegung, Mausrad, und so weiter Handlungen oder sonstige vom Benutzer eingeleitete Ereignisse umfassen.

**[0093]** So können gegebenenfalls Grafikelementereignisse im Skript einer Ereignisbehandlung wie beispielsweise in der bei Ereignisbehandlung für ein Grundereignis oder in einer Transformationsfunktion bei Änderungen in einer Eigenschaft des Grafikelements aktiviert werden. Weiterhin können Grundelemente eines Grafikelements Ereignisse freilegen, die durch Benutzerdialoge mit ihnen beispielsweise über die Maus und die Tastatur aktiviert werden können. Diese Ereignisse bieten dem Benutzer Dialogmöglichkeiten mit dem Grafikelement auf den Ebenen von Formen und Grundelementen. Entwickler können die Ereignisse intern innerhalb des Grafikelements handhaben, indem sie Ereignisbehandlungen beispielsweise unter Verwendung von C#-Verfahren angeben.

**[0094]** Wieder auf [Fig. 5](#) Bezug nehmend kann die Aktions-/Animierungsansicht **123** eine Liste von Transformationsanimierungen und Eigenschaftsanimierungen bereitstellen oder zeigen, die für die gegenwärtig in der Editierungsansicht **14** angezeigten Visualisierung definiert sind. Ein Benutzer kann beispielsweise eine Zeile der Ansicht **123** doppelt anklicken, um den Editor **50** zur Darstellung eines Dialogkastens zu veranlassen, der ein Transformationsanimierungsdialogkasten oder ein Eigenschaftsanimierungsdialogkasten sein könnte, womit der Benutzer die ausgewählte Animierung editieren kann. Typischerweise werden nur die Animierungen der gegenwärtig ausgewählten Visualisierung gezeigt, jedoch könnten durch Anwahl eines Kastens Show-All **144** Animierungen und Aktionen für alle Visualisierungen gezeigt werden. Angenommen, daß Animierungen mit bedeutsamen Namen oder Beschreibungen ausgestattet sind, beispielsweise "Hebel animieren", "Motor drehen" usw. ermöglicht die Aktions-/Animierungsansicht **123** eine leichte Betrachtung und Zugriff zu für eine Visualisierung definierten Animierungsverhaltensweisen. [Fig. 9](#) zeigt eine beispielhafte Aktions-/Animierungsansicht **123A**, die die für die Eigenschaft IsOn definierten Aktionen darstellt. In diesem Fall wird, wenn die Eigenschaft IsOn "wahr" ist, durch das RectangleI genannte Grundelement der Visuell genannten Visualisierung eine Füllaktion mit dem Füllwert "rot" implementiert. Gleicherweise wird, wenn die Eigenschaft IsOn "wahr" ist, durch das Ellipse1 genannte Grundelement der Visuell genannten Visualisierung eine sichtbare Aktion mit dem Wert falsch implementiert (z. B. wird unsichtbar). Natürlich

können Aktionen und Animierungen getrennt für getrennte Grundelemente einer Visualisierung definiert werden, aber diese Aktionen und Animierungen können gleichzeitig auf das gleiche Ereignis, Trigger oder Eigenschaftsänderung einwirken, um eine kompliziertere aber ansprechende Animierung darzustellen. Weiterhin können Aktionen und Animierungen das Durchführen von Einzeloperationen wie beispielsweise Vergrößern, Farbe füllen usw. oder wiederholte Operationen, die fortlaufend stattfinden, bis sie ausgeschaltet werden, umfassen.

**[0095]** [Fig. 10](#) zeigt eine Art und Weise, auf die ein optischer Trigger **148** an einer Visualisierung **150** eines Grafikelements **152** auf Grundlage der Eigenschaften wie beispielsweise inneren Eigenschaften des Grafikelements **152** implementiert werden kann. Insbesondere wird eine oder mehrere der inneren Eigenschaften des Grafikelements **152** durch einen oder mehrere, für den optischen Trigger **148** definierte Eigenschaftstrigger **154** überwacht (von denen drei in der [Fig. 10](#) dargestellt sind). Jeder Eigenschaftstrigger, der unter Verwendung eines Skripts implementiert werden kann, kann, wie durch die gestrichelte Linie **155** gezeigt, gewisse Grafikelement-Eigenschaftswerte überwachen. Danach kann jeder Eigenschaftstrigger, wie durch die gestrichelten Linien **156** dargestellt, einen oder mehrere Grundeigenschaftswerte der Visualisierung **150** einstellen, wenn der überwachte Grafikelementeigenschaftswert erfüllt ist oder einem angegebenen Zustand entspricht. So kann beispielsweise einer der Eigenschaftstrigger eine oder mehrere Grafikelementeigenschaften überwachen, um zu bestimmen, wann ein Wert einer oder mehrerer dieser Eigenschaften in einen bestimmten Bereich fällt. Wenn dieser Zustand erfüllt ist, kann einer der Eigenschaftstrigger **154** das Ablaufen einer Animierung oder eines sonstigen optischen Skripts beispielsweise auf einem Grundelement oder sonstigem Element der Visualisierung **150** veranlassen, um den optischen Trigger **148** bereitzustellen. Natürlich können mehrere der Eigenschaftstrigger **154** zusammenarbeiten, um mehrfache gleichzeitige Änderungen oder Animierungen als Teil des optischen Triggers **148** bereitzustellen, oder es können unterschiedliche der Eigenschaftstrigger **154** unabhängig auf beispielsweise unterschiedliche innere Eigenschaften oder auf Grundlage unterschiedlicher Werte der gleichen inneren Eigenschaft einwirken, um unterschiedliche Operationen des optischen Triggers **148** zu unterschiedlichen Zeiten oder als Reaktion auf unterschiedliche Prozeßzustände bereitzustellen. Auf diese Weise können in der Visualisierung **150** Farbänderungen, Animierungen usw. auf Grundlage der Werte der inneren Eigenschaften des Grafikelements **152** bereitgestellt werden.

**[0096]** Wenn gewünscht kann der Editor **50** ein optisches Triggerfeld bereitstellen oder anzeigen, in dem alle Eigenschaftstrigger für eine gegenwärtig

ausgewählte Visualisierung aufgeführt sind. Ein solches Feld ist in der [Fig. 9](#) dargestellt, in dem in der Elementeeigenschaftsspalte der Eigenschaftsname des Grafikelements aufgeführt ist, die erste Wert-Spalte den beobachteten Eigenschaftswert des Grafikelements anzeigt, die Ziel-Spalte eine Visualisierung oder eine Grundelementkennung bereitstellt, die geändert wird, die Weg-Spalte die geänderte Eigenschaft des Grundelements ist und die zweite Wert-Spalte die Grundelementeigenschaft ist, die angewandt werden soll, wenn der beobachtete Eigenschaftswert des Grafikelements erfüllt ist. Natürlich können Animierungen und sonstige Visualisierungsänderungen unter Verwendung dieser Struktur bereitgestellt werden.

**[0097]** [Fig. 11](#) zeigt einen Transformationsanimierungs-Dialogkasten **160**, der dazu benutzt werden kann, einem Benutzer die Angabe einer Transformationsanimierung zu erleichtern oder zu ermöglichen. Nach der Darstellung enthält der Transformationsanimierungsdialogkasten **160** einen Bewegungsteil **162**, der dem Benutzer die Angabe einer Richtung und einer Entfernung als Pixel erlaubt, die eine Bewegung in einer Animierung definieren, einen Drehungsteil **164**, der einem Benutzer die Angabe einer Drehrichtung und eines Winkels für die Animierung erlaubt, und einen Skalenteil **166**, der dem Benutzer die Angabe eines Skalierungsgrades für eine Animierung sowohl in der horizontalen als auch senkrechten Richtung erlaubt und anzugeben, ob das Seitenverhältnis verriegelt werden sollte. Auch enthält der Dialogkasten **160** einen Verschiebungsteil **168**, der einem Benutzer erlaubt, eine während einer Animierung anzulegende Verschiebung sowohl in horizontaler als auch senkrechter Richtung anzugeben. Ein Einstellungsteil **170** erlaubt einem Benutzer zu definieren, ob die Animierung fortlaufend oder nicht sein soll und die jeweilige der anderen Aktionen Bewegen, Rotieren, Skalieren und Verschieben freizugeben. Weiterhin kann ein Vorschaukasten **172** eine Vorschau der Animierung darstellen.

**[0098]** Wie zu verstehen ist, ist eine Animierung ein Objekt, dessen Wert sich über eine Zeitperiode hinweg ändert. Eine Eigenschaftsanimierung kann durch Assoziieren einer Animierung mit einer Grundelementeigenschaft erreicht werden. Eigenschaftsanimierungen können so subtil sein, daß sie eine Änderung der Farbe des Texts bewirken oder eine Element wie eine Linie zum Blinken veranlassen. Andererseits können Eigenschaftsanimierungen komplizierter sein, wie beispielsweise Animieren der Punkte einer Polygraphlinie usw. Dies sind natürlich nur einige Beispiele von Animierungen und andere Animierungen können das Zufügen oder Ändern von Farben, das Ändern der Größe (wie beispielsweise die Breite, Länge oder Punktgröße) eines Grundelements, Bewegen, Drehen, Verschieben, Skalieren von Grundelementen usw. umfassen. Weiterhin

könnten andere Animierungen auf jeder Ebene der Visualisierung bereitgestellt werden, wie beispielsweise auf der Grundelementebene oder als Teil der gesamten Visualisierung. Natürlich können mehrfache Animierungen bereitgestellt oder an jede bestimmte Visualisierung oder an jedes bestimmte Grundelement einer Visualisierung angelegt werden, wenn dies erwünscht ist. Weiterhin können einer bestimmten Visualisierung oder einem Grundelement einer Visualisierung zugeordnete mehrfache Animierungen gleichzeitig oder zu unterschiedlichen Zeiten in Abhängigkeit von Änderungen der Eigenschaftswerte oder aufgrund von vom Benutzer eingeleiteten Aktivierungsereignissen wie beispielsweise Mausereignissen operieren.

**[0099]** [Fig. 12](#) zeigt einen Eigenschaftsanimierungsdialogkasten **180**, der vom Editor **50** erzeugt werden kann, um einem Benutzer zu ermöglichen, eine Eigenschaft der Animierung zu definieren oder zu ändern, um die Animierung dadurch zu definieren. Der Dialogkasten **180** enthält einen Grenzendefinitionsteil **182** und einen Zeitskaladefinitionsteil **184**. Der Grenzendefinitionsteil **182** bietet oder definiert die Grenzen der Animierung.

**[0100]** Insbesondere definiert eine Eigenschaft "von" den Startwert der Animierung und eine Eigenschaft "zu" definiert den Endwert der Animierung. Der Zeitskaladefinitionsteil **184** definiert eine Dauer für die Animierung, eine Beginnzeit und eine Endezeit für die Animierung. Die Dauereigenschaft definiert die Zeitlänge, die die Animierung zur Vollendung benötigt, die Beginnzeiteigenschaft definiert einen Zeitversatz bezüglich der Startzeit der Animierung und die Endezeiteigenschaft definiert die Endezeit der Animierung bezüglich der Startzeit. Ein Geschwindigkeitsdefinitionsteil **186** erlaubt einem Benutzer die Angabe einer Geschwindigkeit, einer Beschleunigung und einer Verlangsamung unter Verwendung von beispielsweise Gleitschienen. Wie offenbar sein wird, definiert die Geschwindigkeitseigenschaft der Animierung, die Beschleunigungseigenschaft beschleunigt die Animierung bezüglich zunehmender Zeit, während die Verlangsamungseigenschaft die Animierung bezüglich zunehmender Zeit, verlangsamt. Ein Wiederholungsdefinitionsteil **188** erlaubt einem Benutzer die Definition der Art und Weise, auf die sich die Animierung wiederholt, z. B. für eine angegebene Anzahl von Malen, für eine angegebene Zählung oder Dauer oder fortlaufend bis zum Ausschalten.

**[0101]** Ein weiterer Animierungsdialogkasten **190**, auf den über den Dialogkasten **180** zugegriffen werden kann, kann zur Angabe unterschiedlicher Farben benutzt werden, die bei Verwendung des Grenzendefinitionsteils **182** als Grenzen in der Animierung benutzt werden sollen. Ähnliche Dialogkästen können dazu benutzt werden, dem Benutzer die Auswahl an-

derer nichtnumerischer Werte als Eigenschaften einer Animierung auszuwählen. Obgleich sie nicht dargestellt sind, können andere Animierungseigenschaften ebenfalls unter Verwendung dieser oder anderer Dialogkästen ausgewählt oder angegeben werden. Beispielsweise kann eine Selbstumkehrseigenschaft zum Definieren eines Booleschen Werts benutzt werden, der anzeigt, ob die Animierung rückwärts abläuft, nachdem sie ihre Vorwärtsrichtung vollendet hat, und eine Eigenschaft "um" kann dazu benutzt werden, den Gesamtbetrag zu definieren, um den die Animierung ihren Startwert ändert. Eine Relativgeschwindigkeitseigenschaft kann zum Definieren einer Relativgeschwindigkeit benutzt werden, mit der die Zeit für die Animierung im Vergleich zu Hauptanimierungen verläuft, d. h. Hauptelementen der Visualisierung zugeordnete Animierungen. Ein Wert von 1 kann beispielsweise anzeigen, daß die Animierung mit derselben Geschwindigkeit wie die Hauptanimierung fortschreitet, ein Wert von 2 kann anzeigen, daß die Animierung zweimal so schnell wie ihre Hauptanimierung fortschreitet, und so weiter.

**[0102]** Obgleich dies nicht dargestellt ist, können Dialogkästen auch zur Einstellung der Schriftarteneigenschaften einer Animierung benutzt werden, wie beispielsweise des Stils, der Größe, der Schriftart und der Farbe von Text in der Animierung oder Änderungen darin. Zusätzlich kann ein Stricheigenschaftendialogkasten zum Einstellen des Randes oder Liniestils, der Dicke und Farbe von Linien und zur Änderung dieser Eigenschaften benutzt werden.

**[0103]** Grafikdialogkästen können auch zum Editieren anderer Merkmale eines Grafikelements benutzt werden. Beispielsweise können Grafikdialogkästen zur Erstellung neuer Prozeßgrafiken benutzt werden, die das Zufügen einer Grafikelementeigenschaft zu einem Grafikelement und das Anbinden einer Grafikelementeigenschaft an eine Laufzeitumgebung umfassen. Ein solcher Anbindungsdialogkasten kann einen Browser bereitstellen, der einem Benutzer das Browsen unter unterschiedlichen Etiketten oder Variablen ermöglicht, die im Steuerungssystem oder einer sonstigen Laufzeitumgebung definiert sind, um gewünschte Etikette, Variablen, Namen usw. zur Durchführung von Anbindung zu finden. Gleicherweise können andere Dialogkästen zur Zufügung von Visualisierungen, zur Zufügung von optischen Triggern, zum Browsen in Grafikelementen oder Anzeigen und zur Zufügung von Ereignissen benutzt werden.

**[0104]** Wie zu verstehen ist, stellt daher der Grafikeditor **50** Unterstützung zum Definieren von Grafikelement-Grundeigenschaften, Unterstützung für Grafikelementvisualisierungen einschließlich der Erstellung und Handhabung von Grafikgrundelementen oder Formen, Zufügung von Dynamiken, insbesondere Transformationsanimierungen (Drehung, Fortbewegung, Skala und Verschiebung) und Eigenschaftsani-

mierungen (Längenanimierung, Farbanimierung usw.) und zum Definieren von Zuständen, die die Ausführung von Dynamiken aktivieren, bereit. Weiterhin bietet der Grafikeditor **50** Unterstützung zum Speichern und Abrufen von Grafikelementen in der Datenbank, einschließlich von Unterstützung für Grafikelement-Seriellumsetzung in einen XML-Fleck, und Unterstützung zum Kategorisieren von Grafikelementen in einer Grafikelementebibliothek. Weiterhin ermöglicht das System die Speicherung von Grafikelement in vom Benutzer definierten Kategorien oder Mengen, Verriegeln oder Bereitstellung von Sicherheit für solche Grafikelemente, Speichern der Grafikelemente in vom Benutzer definierten oder sonstigen nützlichen Kategorien, Versionsführung von Grafikelementen usw.

**[0105]** Auf alle Fälle wird ein Grafikelement nach seiner Erstellung so in einer Grafikelementedatenbank gespeichert, daß es nicht an Prozeßvariablen oder sonstige Daten in der Laufzeitumgebung der Prozeßanlage gebunden ist. Dieses gespeicherte Grafikelement wird nicht unbedingt in irgendwelchen Anzeigen benutzt, steht aber für eine solche Verwendung zur Verfügung, da dieses Element nunmehr in eine Laufzeitumgebung heruntergeladen wird und an bestimmte wirkliche oder simulierte physikalische Elemente in der Prozeßanlage oder dem Prozeßsteuerungssystem angebunden werden kann. Wenn es angebunden ist, sind die Grundeigenschaften des Grafikelements wie beispielsweise die Prozeßvariable, der Sollwert, die aktuelle Geschwindigkeit usw. mit Datenreferenzen in der Laufzeitumgebung verbunden und an diese angebunden.

**[0106]** [Fig. 13](#) zeigt eine Art und Weise, auf die Grafikelemente an wirkliche physikalische Komponenten oder Elemente in einer Prozeßanlage oder einem in einer Prozeßanlage benutzten Prozeßsteuerungssystem angebunden werden können. Insbesondere enthält jedes Grafikelement **192** eine XAML **194**, die im wesentlichen die Visualisierungen für das Element definiert, und zugehörige Skripts **196**, Animierungen **198**, Trigger **200**, Aktionen **202** und Ereignisse **204**. Auch enthält das Grafikelement **192** eine Bezugstabelle **206**, die alle der XAML **194** zugeordneten Verweise aufführt oder enthält, die in der XAML **194** verfügbar oder offen und freigelegt sind. Die Bezugstabelle **206** besteht im wesentlichen aus oder enthält die Eigenschaften und Parameter des Grafikelements wie auch beliebige andere von den Skripts **196**, Animierungen **198**, Triggern **200**, Aktionen **202** und Ereignissen **204** benutzen Variablen oder Verweise. Die Variablen oder Instanzen in der Bezugstabelle **206** können auf Variablen, Tabellen, Tabelleneinträge in anderen Programmen oder beliebige andere Typen oder Arten von Daten, die sonstwo in dem Prozeßsteuerungssystem definiert sind, verweisen oder daran angebunden sein.

**[0107]** Nach der Darstellung in [Fig. 13](#) wird eine Auflösungstabelle **208** zum Anbinden der Verweise oder Variablen in der Bezugstabelle **206** an die eigentliche Prozeßsteuerungsumgebung oder sonstige Laufzeitumgebung benutzt. Allgemein gesagt wird die Auflösungstabelle **208**, die Verweise entweder direkt oder über Verwendung von Pseudonymen definieren kann, für das Grafikelement **192** bereitgestellt oder erstellt, wenn das Grafikelement **192** tatsächlich zur Verwendung in einer bestimmten Anzeige in einer Laufzeitumgebung konfiguriert wird und auf eine Laufzeitmaschine heruntergeladen wird. Vor oder während der Operation des Grafikelements **192** in einer Anzeige werden durch die Auflösungstabelle **208** die Pseudonyme und sonstige Parameter aufgelöst und diese aufgelösten Datenverbindungen an die Bezugstabelle **206** angebunden, um eine Anbindung zwischen den Variablen in der Bezugstabelle **206** und den eigentlichen Datenquellen in dem Prozeßsteuerungssystem oder sonstigen Laufzeitumgebung bereitzustellen.

**[0108]** Um unnötigen Verbrauch von Verarbeitungsleistung im Prozeßsteuerungssystem während der Laufzeit zu verhindern, müssen Grafikelemente und die Grafikanzeigen, in denen sie benutzt werden, nicht an die Laufzeitumgebung angebunden bleiben, wenn sie nicht eigentlich angezeigt oder auf irgendeinem Anzeigeschirm benutzt werden. Stattdessen kann die Auflösungstabelle **208** nur dann an die Bezugstabelle **206** des Grafikelements **192** angebunden sein, wenn das Grafikelement **192** abläuft oder auf einem Bildschirm in der Laufzeitumgebung angezeigt wird.

**[0109]** Da jedes Grafikelement an das Prozeßsteuerungssystem oder die Laufzeitumgebung durch Verwendung einer Bezugstabelle **206** und einer Auflösungstabelle **208** angebunden ist und da Anbindung stattfindet, nachdem das Grafikelement auf Systemebene erstellt und kopiert und in eine Laufzeitumgebung oder -maschine eingeladen wird, kann das Grafikelement **192** getrennt an unterschiedliche Datenquellen für unterschiedliche Verwendungen zu unterschiedlichen Zeiten angebunden sein. Weiterhin kann das Grafikelement **192** an Daten angebunden sein, die von einer beliebigen Anzahl unterschiedlicher Quellen erzeugt werden oder durch diese verfügbar sind, einschließlich von Datenquellen in den unterschiedlichen Funktionsbereichen **42** der [Fig. 2](#) wie beispielsweise Steuerungsaktivitäten, Wartungsaktivitäten, Modellierungsaktivitäten, Simulationsaktivitäten, Konfigurationsaktivitäten usw. zugeordnete Datenquellen.

**[0110]** Beispielsweise kann nach der Darstellung in [Fig. 14](#) ein Anzeigeelement **192** an Daten angebunden und zum Anzeigen und oder Verarbeiten von Daten benutzt werden, die in weit unterschiedlichen Arten von Datenquellen erzeugt wurden oder und die-

sen zur Verfügung gestellt wurden. Diese Datenquellen können Steuerungsumgebung-Datenquellen **210** wie Steuerungsprogramme wie das wohlbekannte Steuerungssystem DeltaV, OPC-Datenquellen **212**, die Verbindungen mit anderen Systemen über die wohlbekannte OPC-Verbindungsschnittstelle bereitstellt, Wartungsdatenquellen **214** wie das wohlbekannte AMS-System, höhere oder Geschäftssysteme **216** wie das wohlbekannte Ovation-System und sogar Datenquellen, die konkurrierende Systeme **218** wie konkurrierende Steuerungsanwendungen benutzen, enthalten. Auf diese Weise kann das Grafikelement **192** zur Anzeige von Daten und oder Visualisierungen von physikalischen Elementen auf einer beliebigen Systemebene unter Verwendung von Daten von einer beliebigen Anwendung im System benutzt werden, selbst wenn die Daten von weit unterschiedlichen Anwendungsarten kommen oder ihren Ursprung haben, einschließlich von konkurrierenden Systemen zugeordneten Anwendungen, die niemals für eine Zusammenarbeit vorgesehen waren. Da die Grafikelemente und die von Grafikelementen erstellten Grafikanzeigen auf einer Systemebene erstellt werden, können sie daher zur Bereitstellung von Visualisierungen des Geschehnisses in der Anlage für einen beliebigen Zweck benutzt werden, selbst wenn unterschiedliche Arten von Software auf unterschiedliche Hardware- und Softwarekomponenten in der Anlage zugreifen und diese ablaufen lassen.

**[0111]** Aufgrund der modularen Beschaffenheit der Grafikelemente ist es möglich, Grafikelemente automatisch oder halbautomatisch von Gerätebeschreibungen (DD – device descriptions) zu erstellen, die in der wohlbekannten und weitverbreiteten Gerätebeschreibungssprache (DDL – device description language) geschrieben wurden. Insbesondere stellen Gerätehersteller typischerweise eine DD für jedes von ihnen hergestellte Gerät bereit, die in der DDL die dem Gerät zugeordneten Parameter definiert, wie man mit dem Gerät kommunizieren kann, Grenzen für das Gerät usw. Im Ergebnis kann eine Grafikelement-Erstellungsanwendung die DD für ein Gerät in der DDL lesen, um die Geräteart und die wichtigen Parameter, Grenzen usw. zu bestimmen, die diesem Gerät zugeordnet sind, und kann dann diese Parameter als die Grundeigenschaften oder Parameter des Grafikelemente für das Gerät definieren. Auch kann das Programm eine Grundformzusammensetzung als Visualisierung für das Gerät auswählen oder definieren und kann ein oder mehrere generische Skripts zur Verwendung zur Bereitstellung von Grundaktionen und Animierungen für das Gerät entweder auf Grundlage von Informationen von der DD oder auf Grundlage von für Geräte der durch die DD für das Gerät definierten Geräteart gespeicherten Schablonen auswählen. Gegebenenfalls kann das Programm während dieses Vorgangs einen Benutzer fragen, Informationen über das Gerät zu liefern oder Auswahlen betreffs welche Animierungen, Visualisie-

rungen, Grundelemente usw. für das Grafikelement zu verwenden sind, zu treffen.

**[0112]** Für ein vollständigeres oder wohldefiniertes Grafikelement kann das Programm verschiedene generische Grafikschemen für unterschiedliche Arten von Vorrichtungen wie beispielsweise für Sensoren, Ventile, Motoren, Behälter usw. speichern. Das Programm kann dann das zu verwendende Schablonengrafikelement auf Grundlage der durch die DD für die Vorrichtung definierten Geräteart bestimmen. Gegebenenfalls kann die Schablone verschiedene Wahlmöglichkeiten zur Verwendung im Grafikelement bereitstellen oder verfügbar haben und diese Wahlmöglichkeiten können auf Grundlage von Informationen in der DD für die Vorrichtung oder auf Grundlage von Benutzereingabe bestimmt werden. So kann die Schablone beispielsweise verschiedene, verschiedenen Untertypen von Geräten wie beispielsweise Sensoren zugeordnete Grundparameter bereitstellen und das Programm kann die für das Grafikelement zu definierenden Grundparameter auf Grundlage der Informationen in der DD bestimmen.

**[0113]** Als Teil der Schablone können verschiedene Elementarskripts bereitgestellt werden und zur Bereitstellung von Verhaltensweisen für die Visualisierung benutzt werden. Die im Grafikelement zu verwendenden Skripts können wiederum automatisch auf Grundlage der Informationen in der DD wie beispielsweise Gerätetyp usw. oder wenn gewünscht auf Grundlage von Fragen an den Benutzer gewählt werden. Zusätzlich können die im Skript benutzten verschiedenen Sequenzen oder Variablen aus Informationen in der DD bestimmt werden. So kann beispielsweise ein Skript wie das, das eine rotierende Grafik bereitstellt, als im Grafikelement zu benutzen oder bereitzustellen gewählt werden, wenn die DD anzeigt, daß die Vorrichtung ein rotierendes Gerät ist und gewisse Aspekte der Rotationsgrafik wie beispielsweise die Geschwindigkeit, wann die Grafik Farbe wechselt usw. können auf Grenzen beruhen, die dem in der DD definierten Gerät zugeordnet sind. Diese Grenzen können beispielsweise normale oder Nennbetriebsgeschwindigkeit, die definierten Übergeschwindigkeits- oder Untergeschwindigkeitszustände oder Grenzen usw. sein. Als weiteres Beispiel können, wenn die Vorrichtung ein Sensor ist, die hohen und niedrigen Werte des Sensors zur Bereitstellung von dem gegenwärtigen Meßwert des Sensors zugeordneten Grafiken, Grafiken, die zeigen, ob der Sensor fehlerhaft ist usw. benutzt werden.

**[0114]** Auf diese Weise kann aus einer DD für ein Gerät, wenn das Gerät einer Prozeßanlage zugeordnet oder in einer Prozeßanlage erkannt wird, durch Definieren gewisser Grundskripts, Grafikanimierungen, Visualisierungen und Grundparameter für das Grafikelement auf Grundlage der Geräteart von bekannten Eigenschaften des Geräts, sowie sie in der

DD für das Gerät definiert sind, automatisch ein Grundgrafikelement erstellt werden. Diese automatische Erstellung eines Grafikelements bietet einem Benutzer einige Möglichkeit, ein neu zugefügtes Gerät automatisch und ohne viel Grafikeditorungen für das Gerät in Grafiken zu integrieren, so daß Geräte zumindest auf einer elementaren Ebene automatisch in Grafikanzeigen unterstützt werden können, wenn die DD für diese Geräte in das System eingeladen wird. Anders gesagt kann ein Benutzer automatisch ein Grafikelement aus einer DD für ein Gerät erstellen, indem er eine DD für ein Gerät für das System bereitstellt und dann das Programm ablaufen läßt, um ein Grafikelement für das Gerät aus der DD zu erstellen. Danach kann dieses Grafikelement in einer oder mehreren Grafikanzeigen zum Modellieren oder zur Bereitstellung einer Visualisierung für das Gerät benutzt werden, wodurch dem Benutzer die Möglichkeit gegeben wird, das Gerät zu modellieren oder das Gerät in Grafikanzeigen darzustellen, ohne von Hand ein Grafikelement für das Gerät erstellen zu müssen.

**[0115]** Wie oben bemerkt kann, sobald eine Anzahl von Grafikelementen **74** der [Fig. 3](#) erstellt worden, der Grafikeditor **50** zum Erstellen von einer oder mehreren Grafikanzeigen wie beispielsweise den Anzeigen **76** der [Fig. 3](#) benutzt werden. In der Tat kann, sofern gewünscht, der Anzeigeneditor **50** für einen Benutzer oder Käufer mit einer Anzahl vorkonfigurierter Grafikelemente bereitgestellt werden, denen verschiedene Visualisierungen zugeordnet sein können, wie beispielsweise Visualisierungen für unterschiedliche Industrien, unterschiedliche Funktionsverwendungen usw. Dann erlaubt der Grafikeditor **50** dem Benutzer die Erstellung von kundenspezifischen Elementen wie auch die Erstellung oder den Aufbau von Grafikanzeigen **76**.

**[0116]** Allgemein gesagt wird der Benutzer zum Erstellen einer Anzeige **76** aus einer Bibliothek von Grafikelementen **74** und sonstiger optischer Elemente auswählen und sie zusammenlegen, um ein Anzeige aufzubauen. Nach Vervollständigung einer Grafikanzeige kann die sich ergebende Datenstruktur oder das sich ergebende Objekt in einer Datenbank wie beispielsweise einer Konfigurationsdatenbank als Anzeigeklassenobjekt mit allen verschiedenen Elementen, Grundeigenschaften und Visualisierungen definiert als einzelne Definitionsinstanz gespeichert werden. Dieses Anzeigeklassenobjekt wird jedoch nicht an Prozeßvariablen angebunden sein und nicht unbedingt in irgendwelchen Laufzeitanzeigen benutzt. Gegebenenfalls kann dieses Klassenobjekt in der Datenbank als ein XML-Fleck gespeichert sein, bei dem Visualisierungen, Skripts usw. alle zusammen als eine XML-Instanz gespeichert und zusammengeketet sind. Danach können einzelne Grafikanzeigen aus diesem Klassenobjekt erstellt werden und diese einzelnen Grafikanzeigen können Bedienerarbeitsplätzen oder anderen Laufzeitumgebungen zu-

gewiesen und dorthin heruntergeladen werden.

**[0117]** Wenn eine Anzeige heruntergeladen wird, werden die darin enthaltenen Grafikelementdefinitionen in Avalon-Steuererelemente umgewandelt, in Gruppen kompiliert und auf der Ziehmaschine, d. h. der Laufzeitmaschine eingesetzt. Die Anzeige, die selbst eine Avalon-Instanz ist, wird auf die kompilierten Steuergruppen Bezug nehmen und zusätzlich kann eine kundenspezifische Datenquelle erzeugt werden, die als Datenadapter wirkt, der die Avalon-Steuererelemente mit den Backend-Datenquellen wie beispielsweise der Steuerungs- oder Wartungs-Laufzeitanwendungen verbindet. Auf diese Weise können die Grafikanzeigen (und wenn gewünscht einzelne Grafikelemente) in einer Sprache erstellt und editiert werden, in einer anderen Sprache oder Form (z. B. als XML-Fleck) gespeichert und in einer dritten Sprache oder Form (z. B. als an in einer beliebigen verschiedener ausführbarer Sprachen angebundene Avalon-Steuererelemente) ablaufen gelassen werden.

**[0118]** Nunmehr auf [Fig. 15](#) Bezug nehmend kann ein Schirmbild **220** durch den Grafikeditor **50** erzeugt werden, um einem Benutzer die Erstellung einer oder mehrerer Grafikanzeigen zu ermöglichen. Allgemein gesagt besteht eine Grafikanzeige aus miteinander verbundenen Elementen, die physikalische Geräte in einer Anlage darstellen, und kann zusätzlich diesen Geräten zugeordnete Informationen enthalten oder anzeigen. Jedes der Grafikelemente in einer Grafikanzeige enthält eine Menge Grundeigenschaften, die Prozeßvariablen, Konstanten oder sonstigen externen Werten gleichzusetzen sind und wie oben erläutert kann jedes Grafikelement mehrere optische Darstellungen aufweisen, die dynamisches Verhalten, beispielsweise Farbänderungen oder Animierung enthalten können. Zusätzlich kann die Grafikanzeige statische Elemente wie beispielsweise Text, Kästen usw. Benutzerdynamos enthalten, die dem Benutzer erlauben, mit der Grafikanzeige auf irgendwelche Weise, allen möglichen Kästen, die Prozeß- oder sonstige Informationen für den Benutzer darstellen können, usw. zu interagieren.

**[0119]** Ähnlich dem Editierschirmbild **112** der [Fig. 5](#) für ein Grafikelement enthält das beispielhafte Grafikanzeigeneditorschirmbild **220** der [Fig. 15](#) eine Haupteditierentwurfsläche **224**, auf der eine Grafikanzeige aufgebaut wird und einen Palettenteil **226**, wo Schablonen-Grafikelemente, Grundelemente oder sonstige Bibliothekselemente angezeigt werden können und aus dem diese verschiedenen Elemente ausgewählt und auf die Haupteditierentwurfsläche **224** gezogen und darauf abgelegt werden können. Die Haupteditierentwurfsläche **224** bietet eine Diagrammansicht, in der der Benutzer in Dialog tritt, um Grafiken, entweder Grundelemente oder Zusammensetzungen zu erstellen und zu editieren und bietet

Vektorgrafikeditier- und Betrachtungsmerkmale. Der Editor **50** ermöglicht einem Benutzer, Grafikobjekte wie beispielsweise Grafikelemente irgendwo in einen grenzenlosen Raum zu setzen, der innerhalb oder außerhalb des gegenwärtig in der Editierentwurfsfläche **224** dargestellten Anzeigerahmens liegen kann. So könnte die Haupteditierentwurfsfläche **224** nur einen Teil der gegenwärtig erstellten Anzeige darstellen und ein Benutzer könnte die Ansicht verschieben, um andere Teile einer Anzeige darzustellen, ein- und auszuzoomen, um das Vergrößerungsniveau der Ansicht zu ändern und die Ansicht drehen, um die zweckentsprechendste Orientierung zu erhalten, mit der er zu jeder gegebenen Zeit an der Anzeige arbeiten kann. Es versteht sich jedoch, daß jegliche Zeichnung und Manipulierung von Grafikobjekten auf der Haupteditierentwurfsfläche **224** durchgeführt wird. Als Hilfe beim Zeichnen kann die X-Y-Lage der Ansicht von Gliedmaßstäben **227** abgelesen werden, der Vergrößerungsgrad kann in einer Topdownliste an einer Werkzeuggeste **228** angezeigt und durch Auswählen von Einträgen in einem Zoom-Popup geändert werden, usw.

**[0120]** Auch enthält das Schirmbild **220** einen Hierarchieteil **230**, der eine hierarchische Anzeige oder Auflistung der Elemente in der Haupteditierentwurfsfläche **224** darstellt, einen Eigenschaftenteil **232**, der die der in der Entwurfsfläche **224** entstellten oder hervorgehobenen Elemente in der Entwurfsfläche **224** zugeordneten Eigenschaften oder Parameter auführt, und einen Anbindungsteil **234**, der die Art und Weise darstellt oder auführt, auf die verschiedene Parameter oder Merkmale an Elemente in dem Prozeßsteuerungssystem oder irgendeiner sonstigen Laufzeitumgebung angebinden sind. Zusätzlich enthält das Schirmbild **220** eine Werkzeuggeste **236**, die die verschiedenen Ansichten oder Schichten auflistet, die der in der Haupteditierentwurfsfläche **224** erstellten Grafikanzeige zugeordnet sind. Insbesondere kann jede bestimmte Anzeige verschiedene Schichten oder Ansichten aufweisen, die von verschiedenen Leuten in unterschiedlichen Zusammenhängen benutzt werden können, wie beispielsweise eine Bedieneranzeige, eine Wartungsanzeige, eine Simulationsanzeige, eine Modellierungsanzeige, eine technische Anzeige, eine Geschäftsanzeige usw. In dem beispielhaften Schirmbild **220** der **Fig. 15** enthält die Werkzeuggeste **236** drei, als Operationsansicht dargestellte Ansichten oder Ebenen (gegenwärtig in der Haupt-Editierentwurfsfläche **102** dargestellt), die eine typische Steuerwartenbedieneransicht, eine Eng/Maintenance-Ansicht, die eine typische Wartungs- oder Technikeransicht bereitstellt, und eine Training-Ansicht, die eine Simulationsansicht bereitstellt, die beim Simulieren der Prozeßanlage oder an eines in der erstellten Anzeige dargestellten Teils der Prozeßanlage benutzt wird. Da in der Haupteditierentwurfsfläche **224** der **Fig. 15** keine Elemente ausgewählt sind, zeigt der Eigenschaften-

teil **232** den aktuellen Namen der erstellten Anzeige zusammen mit Merkmalen der Anzeige wie beispielsweise Breite und Höhe, eine vom Ersteller bereitgestellte Beschreibung, einen Namen, eine Hintergrundbeschreibung und sonstige Informationen, die ein Ersteller gegebenenfalls für die Anzeige speichern möchte. Da keine eigentlichen Elemente in der Haupteditierentwurfsfläche **224** der **Fig. 15** planiert sind, zeigt der Hierarchieteil **230** auf ähnliche Weise nur den Namen der Anzeige und im Anbindungsteil **234** sind keine Anbindungen dargestellt.

**[0121]** Während der Erstellung einer Anzeige kann ein Benutzer ein Element wie beispielsweise ein Grafikelement auf die Haupteditierentwurfsfläche **224** durch Zugreifen auf dieses Element über beispielsweise eine aus einer Menge von im Palettenteil **226** definierten Elementen plazieren. In dem Beispiel der **Fig. 15** zeigt der Palettenteil **226** verschiedene Kategorien von Elementen, die in die Haupteditierentwurfsfläche **224** plaziert werden können und miteinander verbunden sind, um eine vollständige Anzeige zu erstellen, einschließlich von Stellgliedelementen, Berechnungs- und Steuerungselementen, Verarbeitungselementen, Eigenschaften und Messungen, Formen, Benutzeroberflächensteuerelementen und vom Benutzer definierten Elementen. Natürlich können in der Palettenansicht **226** beliebige andere Arten und Kategorien oder Unterkategorien von vordefinierten Elementen bereitgestellt oder darauf zugegriffen werden. Im vorliegenden Beispiel können Stellglieder Ventile und sonstige Stellgliedelemente umfassen, während Berechnungs- und Steuerungselemente beliebige steuerungsbezogene Elemente wie beispielsweise Anzeigen von Steuerungen, Regelschleifen wie beispielsweise PID-Regelschleifen und sonstige Arten von Regelschleifen, Funktionsblöcke, Steuerungsmodulen usw. umfassen. Bearbeitungselemente nach der Darstellung in **Fig. 15** können Behälter, Reaktoren, Mischgeräte und sonstige Elemente umfassen, die Materialien auf irgendwelche Weise verarbeiten, wie auch beliebige andere Arten von Vorrichtungen, Einheiten usw. Eigenschaften und Messung kann Kästen oder Anzeigeelemente umfassen, die dafür ausgelegt sind, Eigenschaften oder Messungen oder sonstige Daten in der Laufzeitumgebung zu zeigen wie beispielsweise Prozeßvariablen, Alarme usw. Formen können Grundelemente oder sonstige vordefinierte Formen sein, während Benutzeroberflächensteuerungen verschiedene Benutzeroberflächensteuerelemente wie beispielsweise Knöpfe, Gleitregler, Werkzeugkästen usw. umfassen können, die der Benutzer auf einem Anzeigeschirm handhaben kann, um eine Eingabe in die Anzeige zu bewirken. Natürlich können vom Benutzer definierte Elemente beliebige vordefinierte Elemente wie beispielsweise beliebige andere Grafikelemente oder aus Grafikelementen hergestellte höhere Elemente umfassen. In einem Fall können vom Benutzer definierte Elemente Prozeßeinheiten, Prozeßanla-

genbereiche oder sonstige höhere Prozeßinstanzen umfassen. Wie oben bemerkt kann der Zugriff auf vom Benutzer definierte Elemente auf der Basis der Identität des den Editor **50** benutzenden Benutzers begrenzt sein oder global zugänglich sein. Man wird natürlich erkennen, daß beliebige andere Elemente, Formen usw. in den Palettenteil **226** unter beliebigen zutreffenden Rubriken plaziert werden können, um diese Elemente organisiert und leicht durch einen Benutzer zugänglich zu halten.

**[0122]** Wenn ein Benutzer ein Element wie beispielsweise einen Mischbehälter in die Haupteditierentwurfsfläche **224** legt, kann diese Element in der Entwurfsfläche **224** unter Verwendung der Vorgabebildschirmauswahl für dieses Element wiedergegeben werden. In diesem Fall wird der Grafikhierarchieelement **230** das Element zeigen und eine Hierarchie von Teilelementen für dieses Element bereitstellen wie beispielsweise Visualisierungen, Animierungen, Grundelemente usw., die diesem Grafikelement zugeordnet sind, auf ähnliche Weise wie in [Fig. 5](#) gezeigt. Wenn weiterhin ein Grafikelement in der Entwurfsfläche **224** gezeigt wird, das Visualisierungen oder Animierungen enthält, die an die Laufzeitumgebung angehängt sein könnten, zeigt der Anbindungsteil **234** die gegenwärtig definierten Anbindungen.

**[0123]** Wenn gewünscht kann der Benutzer die Visualisierung eines Grafikelements in der Haupteditierentwurfsfläche **224** oder im Palettenteil **226** durch Auswählen einer unterschiedlichen Visualisierung auf jede gewünschte Weise auswählen oder ändern. Vom Benutzer kann diese Visualisierungsauswahlfunktion unter Verwendung einer Dropdown-Liste oder eines Kastens durchgeführt werden, auf den beispielsweise durch rechtes Anklicken mit dem Mauszeiger über dem Element oder Auswählen einer unterschiedlichen Visualisierung in dem Hierarchieelement **230** oder auf sonstige gewünschte Weise zugegriffen werden kann. [Fig. 16](#) zeigt einen Teil der Haupteditierentwurfsfläche **224** und des Palettenteils **226** des Schirmbildes **220**, in dem ein senkrecht Behälterelement **240** als in die Haupteditierentwurfsfläche **224** vom Palettenteil **226** aus plaziert dargestellt ist. Das senkrechte Behälterelement **240** ist eine Kopie oder Konkretisierung einer Schablone oder eines im Palettenteil **226** dargestellten klassenmäßigen senkrechten Behälterelements **241**. Jedoch können zusätzliche Visualisierungen für den senkrechten Behälter **240** einschließlich von in einer Seitenleiste **242** dargestellten Visualisierungen **1-8**, auf die durch rechtes Anklicken mit der Maus zugegriffen wird, wenn der Mauszeiger sich über der Visualisierung **240** befindet, als die für das senkrechte Behälterelement **240** zu verwendende Visualisierung ausgewählt werden, wenn es sich in der in der Editierentwurfsfläche **224** erstellten Anzeige befindet. Nach der Darstellung im Palettenteil **226** kann eine ähnliche Art von Seitenleistenanzeige **243** durch rechtes

Anklicken auf dem schablonenmäßigen senkrechten Behälterelement **241** in Palettenteil **226** erhalten werden, um die möglichen Visualisierungen des Schablonen-Behälterelements **241** zu erhalten oder zu betrachten. Das Auswählen einer unterschiedlichen Visualisierung in der Seitenleiste **242** ändert die in der Entwurfsfläche **224** benutzte Visualisierung des Behälterelements **240**, während das Auswählen einer unterschiedlichen Visualisierung in der Seitenleiste **243** die Vorgabeeinstellung oder Visualisierung für das im Palettenteil **226** gespeicherte Schablonen-Behälterelement **241** ändert.

**[0124]** Zusätzlich zum Auswählen von Grafikelementen und anderen Elementen aus dem Palettenteil **226** zur Erstellung einer Anzeige kann ein Benutzer auch Gegenstände aus der in [Fig. 15](#) gezeigten Werkzeugleiste **228** auswählen oder benutzen, um Grundzeichnungswerkzeuge wie beispielsweise Linien, Formen wie Quadrate, Rechtecke, Kreis, Fünfecke usw., Text usw. zu erhalten und kann diese einfachen Zeichnungswerkzeuge oder -elemente zum Zeichnen von Linien oder zum Hinzufügen von Text in der Anzeige benutzen. Zusätzlich kann der Benutzer der Anzeige in der Entwurfsfläche **224** unter Verwendung von in der Werkzeugleiste **228** gezeigten Werkzeugleisten-Verbinderelementen **245** Verbinderelemente zufügen. Wenn es ausgewählt wird, kann das Verbinderelement **245** dem Benutzer eine Liste von Verbindern zur Verwendung in der Anzeige beispielsweise unter Verwendung eines Dropdown-Menüs, eines Dialogkastens usw. bieten. Mögliche Verbinderelemente umfassen Rohrleitungen, Förderbänder, elektrische Leitungen, Flüssigkeitsleitung oder sonstige Arten von Verbindern, die tatsächlich ein Hardwareelement wie beispielsweise einen Behälter, einen Mischer, eine Pumpe usw. mit einem anderen Hardwareelement wie beispielsweise einem Ventil, einem Sensor usw. verbinden. Solche Verbinderelemente und die Bereitstellung von Verbinderelementen zwischen unterschiedlichen Darstellungen von physikalischen Geräten wie beispielsweise Ventilbehältern, Pumpen usw. wird ausführlicher in der US-Veröffentlichung Nr. 2004/0153804 erläutert, die ausdrücklich durch Bezugnahme hier aufgenommen wird. Die Verbinderelemente können einem Benutzer das Zusammenschalten von unterschiedlichen Elementen in der erstellten Anzeige und Anpassen an die für die verschiedenen Elemente, wie oben für Grafikelemente erläutert, definierten Verbinderpunkte erlauben. Beispielsweise können Rohrverbinder zum Anbringen unterschiedlicher Elemente mit Rohrverbinderpunkten benutzt werden, während ein Kanalverbinder zum Anbringen von Elementen benutzt werden kann, die mit Kanalverbinderpunkten definiert worden sind. Wenn gewünscht kann der Editor **50** Verbindungsregeln implementieren, die einem Benutzer nur das Verbinden unterschiedlicher Grafikelemente wie beispielsweise eines Ventils und eines Behälters mit der zutreffenden Verbindertyp, wie

durch die Verbindungspunkte des Ventils und des Behälters definiert, erlauben. Natürlich kann sich das Aussehen der Verbindungselemente in Abhängigkeit von der Verbindungsart unterscheiden, um der mit Verbindungselementen erstellten Anzeige ein besseres Aussehen und einen besseren Eindruck zu verleihen.

**[0125]** Natürlich kann die Werkzeugleiste **228** andere Vektorzeichenwerkzeuge und Standardbefehle oder -funktionen wie beispielsweise typische Dateioptionen (Neu, Öffnen, Sichern, Sichern als, Schließen, Verlassen usw.), Editieroptionen (wie beispielsweise Löschen, Wiederholen, Ausschneiden, Kopieren, Einfügen, Entfernen), Suchoptionen usw. enthalten. Zusätzlich kann die Werkzeugleiste **228** Menügegenstände enthalten, die grafikbezogene Befehle, wie beispielsweise eine Funktion Eigenschaft zufügen, enthalten, die zum Zufügen einer Grafikelement-Grundeigenschaft benutzt werden kann, eine Funktion Ereignis zufügen, die zum Zufügen eines Grafikeignisses benutzt werden kann, eine Funktion Visualisierung zufügen, die einem Benutzer die Zufügung einer Visualisierung erlauben kann, eine Funktion optischen Trigger zufügen, die einen Dialogkasten öffnen könnte, um dem Benutzer die Eingabe von Informationen bezüglich des zu erstellenden Triggers zu erlauben usw. Weiterhin kann ein Menü Format-Funktionen (wie beispielsweise Einstellung von zugehörigen Text-, Linien- und Füll-Eigenschaften für Text, Linien und zweidimensionale und dreidimensionale Bildungen) und Form-Funktionen erlauben, zu denen die Gruppierung oder Entgruppierung von Objekten, die Auswahl der Reihenfolge von Objekten von vorne nach hinten, die Ausrichtung von Objekten sowohl senkrecht als auch horizontal usw. gehören kann. Auch kann das Werkzeugleistenmenü **228** Funktionen bereitstellen, die einem Benutzer die Anwendung von Animierungen auf Objekte ermöglichen kann und solche Animierungen können an ein Grundelement eines Elements oder an ein oder mehrere Elemente eines Objekts wie beispielsweise durch Animierung einer Elementbreite, -farbe, Form usw. angewandt werden, um Verschiebungen, Drehungen oder Skalierung der Form bereitzustellen, Elemente zu drehen oder überwerfen usw. Natürlich kann ein Benutzer Zugang zu den jedem Element in der Anzeige zugeordneten Animierungen und Aktionen erlangen und diese Animierungen und Aktionen unter Verwendung zutreffender Dialogkästen oder sonstiger Programmierwerkzeuge ändern.

**[0126]** Es versteht sich, daß ein Benutzer oder Konstrukteur den Anzeigeneditor **50** benutzen und insbesondere mit dem Schirmbild **220** der **Fig. 50** in Dialog treten kann, um schnell und leicht Grafikanzeigen zu erstellen, die aus standardmäßigen zweidimensionalen oder sogar dreidimensionalen Grafikelementen bestehen, die Messungen, Stellglieder und Prozeß-

geräte dynamisch zeigen können. Auch werden Steuerungen und Berechnungen zugeordnete statische Elemente und Benutzerdynamiken unterstützt und können in den Anzeigen bereitgestellt werden. Zusätzlich kann der Benutzer Schichten zum Adressieren der Schnittstellenerfordernisse unterschiedlicher Benutzer der Grafikanzeigen definieren, indem er beispielsweise Bediener-, Wartungs- und Simulationsanzeigen unter Verwendung der gleichen Elemente mit etwas unterschiedlicher Ansicht oder umgebenden Informationen erstellt.

**[0127]** **Fig. 17** zeigt ein Anzeigeschirmbild **300** einer beispielhaften Grafikanzeige in der Form einer Steuerwartenbedieneranzeige, die unter Verwendung des Grafikeditors **50** erstellt werden kann. Das Anzeigeschirmbild **300** enthält ein Reaktorelement **310**, das mit einem Ventilelement **312** und einem oder mehreren Pumpenelementen **314** über Verbindungselemente **316** verbunden ist, die in diesem Fall Flüssigkeitsleitungen sind. Flüssigkeitsfluß- oder Stromelemente **318** bieten Verweise auf die Materialströme, die in den durch das Schirmbild **300** dargestellten Teil der Anlage hereinkommen und ihn verlassen. Es versteht sich, daß die Grafik im Anzeigeschirmbild **300** durch Auswählen und Verbinden von verschiedenen Grafikelementen miteinander erstellt werden kann, und so kann das Anzeigeschirmbild **300** hochauflösende Grafiken, Animierungen, Aktionen, optische Trigger usw. enthalten, die oben bezüglich einzelner Grafikelemente auf beschriebene Weise entwickelt wurde. Natürlich können zusätzliche Grafiken einschließlich von Animierungen, Aktionen, optischen Triggern, Flüssigkeitsflußelementen usw. auf der Grafikanzeigeebene hinzugefügt werden. Auf diese Weise können Animierung und Benutzeraktionen in die Grafikanzeige aufgenommen werden. Beispielsweise können unter Verwendung dieser Fähigkeit statische Grafikkomponenten für Prozeßgeräte abgeändert werden, um den Zustand des Geräts anzuzeigen, z. B. ob ein Motor eingeschaltet oder aktiviert ist, den Betriebszustand oder die Position eines Ventiles usw. Zusätzlich kann Animierung zur Darstellung von dem Gerät zugeordneten dynamischen Daten benutzt werden, wie beispielsweise die Höhe eines Behälters, der gefüllt wird (z. B. unter Verwendung eines Füllverfahrens) zeigen oder den Zustand eines Rührwerks durch Bewegung anzeigende Anzeigenänderungen (Animierungen) zeigen. Auf gleiche Weise können Datenanzeigeelemente **319** in das Schirmbild **300** plaziert werden, um Prozeßdaten oder sonstige außerhalb der Anzeige **300** entwickelte, aber mit der Funktionsweise der Komponenten in der Anzeige **300** in Beziehung stehende Daten zu zeigen.

**[0128]** Wenn gewünscht kann die Grafikanzeige **300** auch Dynamos oder Benutzeroberflächenknöpfe **320A** und **320B** enthalten, die einem Benutzer den Dialog mit der Anzeige **300** auf eine Weise ermögli-

chen, die dem Benutzer die Betrachtung zusätzlicher Informationen oder das Unternehmen von Handlungen bezüglich der Laufzeitumgebung über den Dialog mit dem Anzeigeschirmbild **300** erlaubt. In manchen Fällen können diese Dialoge, wie oben bezüglich Grafikelementen beschrieben, mit optischen Triggern oder Aktionen implementiert werden. Beispielsweise kann der Knopf **320A** dem Benutzer eine weitere Ansicht von Informationen über den Kühlturm-Aufbau bieten, während der Knopf **320B** eine Frontplattenanzeige für den Reaktor **310** bereitstellen kann. Während die Anzeige **300** eine Menge von Elementen enthält, die die Gerätestücke, die einen Prozeß oder einen Teil eines Prozesses bilden, unter Verwendung dreidimensionaler Komponenten darstellt, die wie in der Vergangenheit auf von einer Anzahl unterschiedlicher Quellen wie beispielsweise In-Tools, Auto-Cad, Windows-Metadateien wie beispielsweise Visio, Vektorzeichnungen, JPEG und Bitmap-Bildformaten importierten Grafiken beruhen können, kann das Schirmbild **300** daher nunmehr Animierungen enthalten, um beispielsweise Drehungen, Dimensionierung, Skalierung, Verschiebung, Farbänderung usw. auf Elementebene durchzuführen, um interessantere und realistischere Animierungen und daher verständlichere Anzeigen bereitzustellen.

**[0129]** Weiterhin können in der Anzeige **300** Grund-Benutzeroberflächenkomponenten wie beispielsweise Knöpfe, Skalen, Zeigebalken und Knöpfe in der Anzeige **300** dargestellt werden und dynamisch mit Informationen oder Steuerelementen im Steuersystem oder einer sonstigen Laufzeitumgebung verknüpft sein. Datenbetrachtungselemente oder Dynamos können auch Messungen in Steuerfunktionen, Alarmen, Ereignissen usw. zugeordnete Schlüsselparameter bereitstellen oder darstellen. Beispielsweise kann ein Dynamo zum Anzeigen eines weiteren Schirmbildes mit Informationen bezüglich eines Regelschleifenparameters und technischen Einheit benutzt werden und kann neben dem Dynamo dargestellt werden, um Zusammenhang mit dem durch den Dynamo angezeigten Wert bereitzustellen. Der Zustand von einer Regelschleife zugeordneten Prozeßalarmen kann im Dynamo durch Farbänderung, beispielsweise eine Änderung der Hintergrundfarbe des Steuerparameterwerts widerspiegelt werden. Auch kann zur Unterdrückung von Störzeichen in der Anzeige die Tatsache, daß sich eine Schleife nicht in einer entworfenen normale Betriebsart befindet, durch Farbänderung angezeigt werden. Natürlich können solche Dynamos dazu veranlaßt werden, einem beliebigen Standard zu entsprechen.

**[0130]** In einem Beispiel kann, wenn ein Bediener auf eine Benutzerdialogkomponente oder einen Dynamo zugreift, ein Skript oder sonstiges Programm ein weiteres Schirmbild oder eine weitere Anzeige

wie beispielsweise eine Frontplattenanzeige oder eine Steuerfeldanzeige hochziehen, von denen Beispiele in der [Fig. 18](#) dargestellt sind. Wenn zum Beispiel auf einen Benutzeroberflächenknopf wie beispielsweise den in dem Schirmbild **300** gezeigten Knopf **320B** zugegriffen wird, kann dem Bediener eine Frontplatte für den Reaktor **310** dargestellt werden und der Bediener kann dann diese Frontplatte zur Abänderung oder Betrachtung von Einzelheiten über den Reaktor **310** benutzen. Im Beispiel der [Fig. 18](#) ist die Frontplatteninformation **350** einer Regelschleife (mit der Bezeichnung FIG2\_28/TC2-1) für den Reaktor **310** zugeordnet, auf die der Benutzer über den Knopf **320A** in dem Schirmbild **300** zugreifen kann. Unter Verwendung des Teils **350A** der Frontplatte **350** kann der Benutzer die Betriebsart (von Kaskade zu beispielsweise manuell oder auto) unter Verwendung der Knöpfe **352** ändern (deren Funktionsweise durch Aktionsroutinen definiert sind), kann die aktuellen Werte der Betriebsparameter auf den Schieberanzeigen **354** betrachten, einen der Regelschleife zugeordneten Sollwert unter Verwendung eines Pfeils **355** ändern, usw. Zusätzlich können dem Benutzer Informationen über die Grenzen und Abstimmparameter der Regelschleife im Anzeigeteil **350B** geboten werden und Simulationsfähigkeiten im Teil **356** ermöglicht werden. Wenn gewünscht kann der Teil **350B** durch Anwahl eines des Knöpfe **358** auf dem Teil **350A** aus der Ansicht des Teils **350A** erhalten werden. Auf gleiche Weise kann der Benutzer auf weitere Informationen über die Regelschleife wie beispielsweise Tendenzdaten, Diagnosedaten usw. zugreifen oder kann Steuerungs- und Diagnoseprogramme wie beispielsweise Schleifenabstimmprogramme über die anderen Knöpfe **358** fahren und darauf zugreifen. So kann auf beliebige andere Aktivitäten, Schirmbilder und Aktionen über die Anzeige **300** als Reaktion auf auf dem Schirmbild **300** unternommene oder zugelassene Benutzeraktionen zugegriffen werden.

**[0131]** Wo das Steuerungssystem die Verwendung von Pseudonymen in der Definition ähnlicher Geräte unterstützt, können die dynamischen Anzeigekomponenten auch zur Unterstützung von dynamischer Bezugnahme auf Grundlage des im Anzeigeschirmbild **300** ausgewählten Geräts ausgelegt sein. In solchen Fällen können vorkonfigurierte Pseudonyme und Attribute anstelle eines Objektetiketts oder grafischen Attributen benutzt werden, die normalerweise als Teil eines Anzeigeobjekts definiert werden. Diese Pseudonymfähigkeit unterstützt ein hohes Maß an Flexibilität und Wiederverwendbarkeit, da ähnliche Anzeigeobjekte mit unterschiedlichen E/A-Pumpen verbunden sein können und unterschiedliche grafische Attribute, Erscheinungsformen und Sicherheit darstellen. Durch eine solche Fähigkeit kann sich die Notwendigkeit zum Neubauen ähnlicher Anzeigeobjekte oder ähnlicher Anzeigen für unterschiedliche Stücke duplizierten Geräts in einer Anlage erübrigen. Auf

diese Weise kann die gleiche Grafikanzeige zur Betrachtung unterschiedlicher Hardwareeinheiten benutzt werden, die sich in Aufmachung und Nutzung in einer Anlage gleich sind.

[0132] Natürlich kann das Anzeigeschirmbild **300** der [Fig. 17](#) zur Unterstützung von Werkzeugleisten einschließlich von sowohl horizontalen (unterhalb der Anzeige) und senkrechten (rechts von der Anzeige) Werkzeugleisten zusätzlich zu oder anstelle der oben im Schirmbild **300** dargestellten Werkzeugleiste ausgelegt sein. Wenn gewünscht können Vorgabe-Werkzeugleisten zur Unterstützung der Zeit- und Datumanzeige, Betrachtung einer Alarmliste mit direktem Zugriff auf die zur Bestätigung des Alarms oder Ruhelegung des Alarms erforderliche Alarmanzeige, Navigation zu einer Alarmzusammenfassungsveranstaltung oder -menü, einem Hauptmenü oder sonstigen Standardmenüs oder Anzeigen, Systemzustandsanzeigen usw. bereitgestellt werden.

[0133] [Fig. 19](#) zeigt ein Anzeigeschirmbild **400**, das einer fortgeschritteneren Grafikanzeige, d. h. einer mit mehr Elementen, Verbindungen, Benutzeroberflächenaktionen, Dynamos und sonstigen Datenverweisen zugeordnet ist. Insbesondere zeigt das Schirmbild **400** die Funktionsweise einer Kalkofeneinheit, in der Luft und sonstige Verbrennungsbrennstoffprodukte durch Pumpen **412**, Ventile **404** und zugehörige Verbinderelemente in einen Eingang **406** eines Ofentrommeltrockners **408** eingepumpt oder eingespeist werden. Auf gleiche Weise wird Prozeßmaterial in einem Behälter **409** von einer Schichtförderervorrichtung **410**, die den Kalk durch den Trommeltrockner **408** transportiert, gespeist. Natürlich können jede dieser Elemente und sonstigen im Schirmbild **400** dargestellten Geräte Grafikelemente sein, die einzeln erstellt und wie oben beschrieben in das Schirmbild **400** plaziert werden können. Der Eingang **406** des Trommeltrockners **408** kann ein animiertes Grafikelement mit einem Feuer oder einer Flamme **415** als Animierung enthalten, wenn der Trockner **408** in Betrieb ist, um einem Betrachter des Schirmbildes **400** deutlich die Funktionsweise der Ofeneinheit anzuzeigen. Zusätzlich können die Temperaturen an verschiedenen Stellen im Trockner **408** mit Dynamos oder Temperaturanzeigekästen **41G** angezeigt werden und gegebenenfalls können Temperaturen an diesen verschiedenen Punkten über die Farbe der Feueranimierung, des Trockners **408** oder auf sonstige Weise angezeigt werden. Gleicherweise können andere Prozeßparameter wie beispielsweise Haubendruck, Ofengeschwindigkeit, Sandluft, Methanol, Eingabe, Primär- und Sekundärlufteingaben, Öl- und Gaseingabe usw. wie in [Fig. 19](#) dargestellt mit Parameterkästen dargestellt werden, wobei die Werte der Variablen in den Kästen an bestimmte Verweise im Prozeßsteuerungssystem angebunden sind und daraus erhalten werden. Natürlich sind im Schirmbild **400** andere physikalische Komponenten

der Kalkofeneinheit dargestellt.

[0134] Auf dem Schirmbild **400** können jedoch verschiedene andere Informationen bereitgestellt werden, die aus anderen Datenquellen in der Laufzeitumgebung erhalten werden können, wie beispielsweise andere Anwendungen, die Daten von dem Steuersystem oder von den Geräten im Steuersystem verarbeiten. Solche Anwendungen können beispielsweise Steueranwendungen, Wartungsanwendungen, Diagnoseanwendungen, Geschäftsanwendungen usw. umfassen. Beispielsweise ist oben in der Mitte des Anzeigeschirmbildes **400** ein Tendenzgraph **420** mit der Temperatur (in Grad) des Systems über Zeit angezeigt und ist an einen Datenhistoriker oder eine Tendenzanwendung angebunden und zeichnet diese Daten automatisch auf. Auf gleiche Weise ist eine Karte **422** mit einer Zusammenfassung der Funktionsweise des Ofens **412**, die durch eine Geschäftsanwendung bereitgestellt werden kann, die verschiedene Geschäftsaspekte wie beispielsweise Profit, Energieverbrauch, Produktionsrate usw. des Ofens verfolgt, oben links auf dem Schirmbild **400** dargestellt. Weiterhin kann auf sonstige Informationen, die für den Benutzer nützlich sein könnten, wie beispielsweise historische Tendenzen, Hilfsinformationen, Ofeninformationen, Steuerungsbeschränkungen, Energie und sonstige Informationen durch Anwählen der Benutzeroberflächenknöpfe **424** im oberen Teil des Anzeigeschirmbildes **400** zugegriffen werden. Weiterhin kann unten im Schirmbild **400** eine Alarmfläche **430** oder sonstige Fläche bereitgestellt werden und kann an eine Alarmierungsanwendung angebunden sein. Natürlich können alle diese Elemente und Merkmale des Anzeigeschirmbildes **400** unter Verwendung des oben beschriebenen Anzeigeditors **50** bereitgestellt und in die Anzeige einprogrammiert werden. Zusätzlich können diese und sonstige Merkmale auf gewünschte Weise in einer Grafikanzeige kombiniert werden, um jede gewünschte Art von Anzeige zu erstellen. Weiterhin könnten alle gewünschten Animierungen und Grafikaktionen im Schirmbild **400** bereitgestellt werden, um nützlichere oder verständlichere optische Informationen für einen Benutzer bereitzustellen, um dem Benutzer eine unmittelbare Betrachtung anderer verwandter Informationen in dem Schirmbild **400** zu erlauben, usw.

[0135] So können wie in der [Fig. 19](#) dargestellt Informationen von verschiedenen unterschiedlichen Datenquellen einschließlich von Prozeßelementen, auf die durch das Steuerungsteilsystem zugegriffen wird, Anwendungen in der Form von Wartungs-, Steuerungs-, Diagnose-, Abstimmungs- und Geschäftsanwendungen, historische Daten oder Tendenzdaten von Datenbanken wie beispielsweise einem Datenhistoriker oder beliebigen anderen Datenquellen im Anzeigeschirmbild **400** angezeigt werden. Weiterhin kann auf andere Informationen oder Anzei-

gen über Benutzerlinks mit solchen Anzeigen direkt aus dem Anzeigeschirmbild **400** zugegriffen werden. Auf diese Weise bietet die durch das Schirmbild **400** dargestellte Grafikanzeige einem Benutzer mehr Informationen auf eine Weise, die nützlicher sein könnte.

**[0136]** Da der Editor **50** zum Erstellen beliebiger Grafikanzeigen unter Verwendung einer gemeinsamen Menge von Grafikelementen, Verbindungselementen usw. benutzt werden kann, kann der Anzeigeditor **50** zur leichten Erstellung ähnlicher oder verwandter Anzeigen benutzt werden. Solche Anzeigen können auf den gleichen Satz Anlagenhardware bezogen sein und diesen darstellen, aber unterschiedliche Informationen für diese Hardware für unterschiedliche Zwecke darstellen, wie beispielsweise für Steuerwartenbedienerzwecke, Simulationszwecke, Wartungszwecke usw. In diesem Fall kann eine gemeinsame oder Grundanzeige geschaffen werden, die die einer Anlage oder einen Teil einer Anlage zugeordneten Hardwareelemente zeigt und es können unterschiedliche Anzeigen unter Verwendung dieser Grundanzeige erstellt werden, die aber unterschiedliche Informationen für unterschiedliche Benutzer oder Benutzerarten bereitstellen.

**[0137]** Beispielsweise zeigen die [Fig. 20A–Fig. 20E](#) beispielhafte Anzeigen einschließlich einer Bedieneransicht, einer Ingenieursansicht, einer Manageransicht, einer Simulationsansicht und einer Wartungsansicht, erstellt für eine Kalkofeneinheit mit den gleichen Grundhardwareelementen in der gleichen Konfiguration und unter Verwendung der gleichen Visualisierungen aber mit für unterschiedliche Funktionszwecke zugefügten unterschiedlichen Informationen. So besitzt jede der Anzeigen der [Fig. 20A–Fig. 20E](#) das gleiche Aussehen und fühlt sich genauso an, da sie mit dem gleichen Anzeigeditor unter Verwendung der gleichen Grund-Grafikelemente hergestellt sind und es daher leicht ist, zwischen ihnen zu navigieren und dabei ein Verständnis der dargestellten Informationen zu behalten und wie sie auf die Hardware in der Anlage bezogen sind.

**[0138]** Insbesondere zeigt die [Fig. 20A](#) eine Bedieneransicht **500** einer Ofeneinheit mit einer Kalk- und Schlammzufuhrquelle **502** (die ein Stromelement sein kann), die Rohmaterial durch eine Pumpe **504** zu einem kalten Ende **506** eines Ofentrockners **508** speist. Von einem Förderer **510** wird der verarbeitete Kalk aus einem heißen Ende **512** des Ofentrockners **508** und der Kalk in einen Behälter **514** abgelegt. Eine Produktlinie oder ein Stromelement **516** zeigt die Menge an Produkt (Kalk), die aus dem Behälter **514** austritt. Auf gleiche Weise wird Brennstoff von einem Brennstoffquellenstromanzeiger **520** durch ein Ventil **522**, dessen Farbe die Funktionsweise des Ventils grafisch darstellen kann, in das heiße Ende

**512** des Ofentrockners **508** zugeführt. Von einem Lüfter **524** wird Luft durch ein Kanalverbindererelement **526** zum heißen Ende **512** des Ofentrockners **508** gepumpt, wo sie mit dem Brennstoff vermischt wird. Im heißen Ende **512** des Ofentrockners **508** können Animierungen wie beispielsweise Feuer oder Flamme **528** gezeigt werden, um den Betriebszustand des Ofentrockners **508** darzustellen. Auf gleiche Weise wird von einem Induktionslüfter **530** Luft aus dem kalten Ende **506** des Ofentrockners **508** durch zusätzliches Kanalwerk **532** gezogen und diese Luft zu einem Kamin gesendet wie durch ein Kaminstromelement **534** angezeigt. Weiterhin sind die Werte verschiedener Prozeßparameter wie beispielsweise Temperaturen an verschiedenen Teilen des Systems, die Geschwindigkeit der Trommel des Ofentrockners **508**, Luft- und Brennstoffflüsse usw. darstellende Variablen oder Parameterkästen dargestellt. Wie ersichtlich sind diese Hardware und diese Variablenkästen allen Schirmbildern der [Fig. 20A–Fig. 20E](#) gemeinsam und geben die Grundelemente der Anzeigen, die der Grund dafür sind, daß diese Anzeigen das gleiche Aussehen aufweisen und sich gleich anfühlen.

**[0139]** Jedes der Schirmbilder der [Fig. 20A–Fig. 20E](#) enthält jedoch zusätzliche Informationen, Benutzeroberflächenknöpfe und Aktionen, die auf unterschiedliche Funktionen in der Anlage zugeschnitten sind. Beispielsweise ist das Schirmbild **500** der [Fig. 20A](#) einer Bedieneransicht und zeigt durch einen Labortest gemessenes, geschätztes und in einem Kasten **540** bereitgestelltes Restcarbonat, die Gesamtenergie und die spezifische Energie des Ofens in einem Kasten **542** wie auch einen Tendenzgraphen **544**, der die vergangenen und zukünftigen Tendenzen der Temperaturen zeigt, die durch eine prädiktive Modell-Steuerroutine bereitgestellt werden, die der Schlüssel zum Erreichen eines optimalen Betriebs sind. Zusätzlich werden durch das Schirmbild **500** verschiedene Benutzeroberflächenknöpfe oder -dynamos **548** bereitgestellt, die einem Benutzer die Betrachtung zusätzlicher Informationen über einen zugehörigen Gegenstand wie beispielsweise Regelschleifeninformationen, weitere Betriebsinformationen über ein Element usw. erlauben.

**[0140]** Das Schirmbild **550** der [Fig. 20B](#) zeigt eine technische Ansicht, die der der [Fig. 20A](#) ähnlich ist, aber die über gestrichelte Linien **552** die Messungsorte zeigt, die für die Regelung von Wichtigkeit sind und die Signalregel, um zu zeigen, wie das Grundregelsystem und das prädiktive Modell-Steuersystem zusammenarbeiten, um den Betrieb zu optimieren. Das Schirmbild **560** der [Fig. 20C](#) bietet eine Managementansicht die, während sie die Grundoperation der Ofeneinheit zeigt, keine Benutzeroberflächenknöpfe bereitstellt, die das Erlangen zusätzlicher Informationen über in einzelne Komponenten oder Regelschleifen des Systems erlauben. Weiterhin enthält die Ma-

nagementansicht **560** eine Karte **562**, die die Gesamtenergie, spezifische Energie und das Restcarbonat (unumgesetzte Einspeisung) von Labormessungen über den vergangenen Tag und Monat zeigt. Diese Informationen können von einer Geschäftsanwendung bereitgestellt werden, die beispielsweise auf einem Geschäftsrechner abläuft.

**[0141]** Weiterhin bietet die [Fig. 20D](#) eine Simulationsansicht **570**, die einem Simulator die Veränderung von Parametern in einem simulierten System und das Betrachten der simulierten Ergebnisse erlaubt. Die Ansicht **570** kann beispielsweise zum Schulen von Bedienern, zum Prüfen unterschiedlicher zukünftiger Betriebsarten usw. benutzt werden.

**[0142]** Nach der Darstellung in [Fig. 20D](#) kann der Simulator Parameter über einen oder mehrere Dialogkästen **570** ändern, die für den Simulationsbediener über einen oder mehrere Benutzeroberflächenknöpfe auf dem Schirmbild **570** bereitgestellt werden. Die Prozeßsimulationsansicht **570** kann in einer Off-line-Situation zum Schulen oder in einer On-line-Situation zur Bereitstellung zusätzlicher Informationen, die zur Erkennung zukünftiger Probleme nützlich sein können, benutzt werden. Die Prozeßsimulation kann aus den Grafiken erzeugt werden, da die Gerätearten und ihre Verbindungen als Ergebnis der Verwendung von intelligenten Objekten oder Grafikelementen mit Simulationsfähigkeiten beim Aufbau der Bedienergrafiken bekannt sind. Diese intelligenten Objekte sind ausführlich in US-Veröffentlichung Nr. 2004/0153804 beschrieben.

**[0143]** Gleicherweise zeigt die [Fig. 20E](#) eine Wartungsansicht **580**, die Informationen über oder in den Zustand der Geräte in der Ofeneinheit bereitstellt. In der Ansicht **580** kann Geräteausfall unter Verwendung von Anzeigen **582** und **584** von Gerätegesundheit gezeigt werden. In diesem Fall sind Anzeigen **582** und **584** (von denen nicht alle in der [Fig. 20E](#) etikettiert sind) Halbkreise, wobei eine Füllfarbe den aktuellen überwachten Gesundheitszustand des Geräts zeigt oder angibt. In der Ansicht **580** zeigen die Anzeigen **584** ein Gerät mit weniger als optimaler Gesundheit an. Natürlich können diese Anzeigen **582** und **584** aus Wartungs- oder Diagnoseanwendungen heraus bereitgestellt werden. Weiterhin kann ein Wartungstechniker verständlicherweise bei Erkennung eines Prozeßproblems das Schirmbild **580** für mehr Details und zum Auffinden der empfohlenen Prozeduren zur Problembeseitigung benutzen. Als Beispiel muß ein verstopfter Stammfilter gewaschen werden, was dann Ausschalten der Zuführung und Versetzen des Ofens in Leerlaufzustand erfordert, was dann aus der Bedieneransicht **500** der [Fig. 20A](#) aus geschehen kann. Die Betriebstechnik wird auch wissen, daß sie die Güte des Beschickungsmaterials auf übermäßigen Grobstaub überprüfen muß, was durch ein stromaufwärtiges Prozeßproblem verur-

sacht sein könnte. Wie durch dieses einfache Beispiel gezeigt, können verschiedene Wechselwirkungen zwischen den unterschiedlichen Funktionen wie beispielsweise Erkennung eines Problems und Beseitigung desselben einfach und leicht durch Umschalten zwischen unterschiedlichen Ansichten durchgeführt werden, die das gleiche Aussehen und Beschaffenheit aufweisen und daher leicht zu navigieren sind. Natürlich können die Informationen in den verschiedenen Schirmbildern der [Fig. 20A–Fig. 20E](#) auf der Anzeige geschichtet sein und können sofern gewünscht nur in Abhängigkeit von dem Benutzer des Schirmbildes gezeigt werden. Auch können sich die für jedes Element oder intelligente Objekt dargestellten Informationen mit dem Benutzer oder der Identität des Benutzers ändern.

**[0144]** Weiterhin können zusätzliche Anzeigen für die Anzeigen der [Fig. 20A–Fig. 20E](#) bereitgestellt und diesen zugeordnet sein. Zu diesen Anzeigen können beispielsweise Steuerungskonfigurationsanzeigen wie die der [Fig. 21A](#) und [Fig. 21B](#) gehören. Eine Steuerungsanzeige **585** der [Fig. 21A](#) zeigt die Art und Weise, auf die verschiedene Steuerungssignale zu und von den verschiedenen Hardwareelementen der in [Fig. 20A–Fig. 20E](#) dargestellten Ofeneinheit zu einer Steuerung gesendet werden, wie auch die diesen Steuerungselementen zugeordneten Etiketten oder Variablennamen. Eine Steuerungsanzeige **590** der [Fig. 21B](#) zeigt ein Steuermodul mit den Steuerprozeduren, die die Steuerung der Ofeneinheit der [Fig. 20A–Fig. 20E](#) implementieren. Obwohl sie nicht besonders als solche dargestellt sind, können die Steuerungsanzeigen **585** der [Fig. 21A](#) unter Verwendung der gleichen Elemente der Schirmbilder **20A–20E** (mit den gleichen oder unterschiedlichen, diesen Elementen zugeordneten Visualisierungen) hergestellt werden oder können unterschiedliche Elemente benutzen. Auf gleiche Weise können, da die Anzeigen **585** und **590** der [Fig. 21A](#) und [Fig. 21B](#) unter Verwendung des Anzeigeneditors **50** hergestellt werden können, für diese Anzeigen beliebige der oben besprochenen Grafik- und Animierungsfähigkeiten bereitgestellt werden. Weiterhin kann auf die Steuerungsanzeigen **585** und **590** beispielsweise über die Bedieneransicht **500** der [Fig. 20A](#) über einen der Benutzeroberflächenknöpfe **584** zugegriffen werden, die darauf bereitgestellt sind, um einem Bediener zu ermöglichen, leicht Zugang zu der aktuellen, aus der Bedieneransicht **500** aufgebauten Steuerung zu erhalten.

**[0145]** So können, wie aus der oben gebotenen Besprechung zu verstehen ist, verschiedene verwandte Anzeigen durch den Editor **50** erstellt werden und diese Anzeigen können auf verschiedene Weise geschichtet sein. Insbesondere können die Anzeigen wie oben beschrieben geschichtet sein, um verwandte oder ähnliche Bediener-, Geschäfts-, Simulations-, Wartungs- und Ingenieuransichten bereitzustellen

und diese Ansichten können leicht voneinander aus zugänglich sein. Weiterhin können verschiedene Anzeigen auf eine Weise geschichtet oder miteinander verbunden sein, die eine bestimmte hierarchische, logische oder physikalische Struktur der Anlage widerspiegelt. So können beispielsweise Anzeigen für unterschiedliche physikalische oder geographische Bereiche einer Anlage hergestellt werden. So kann eine einzige Anzeige erstellt werden, die die Hauptbereiche der Anlage zeigt, und zusätzliche Anzeigen können erstellt werden, die die Grundstruktur zeigen, beispielsweise auf einer einheitsindividuellen Basis, in jedem Bereich der Anlage, während weitere Anzeigen für jede Einheit erstellt werden können. Auf ähnliche Weise kann ein Benutzer durch die Anzeigen gehen, um immer größere Einzelheiten zu erhalten, die immer kleineren Teilen der Anlage zugeordnet sind. Als weiteres Beispiel können Anzeigen für verschiedene Abschnitte oder Teile der Anlage über Benutzeroberflächenknöpfe verkettet oder aneinander angebracht sein, so daß der Bediener leicht vorwärts und rückwärts durch die verschiedenen Anzeigen durchrollen kann, um die unterschiedlichen, aber physikalisch miteinander verbundenen Teile der Anlage oder Komponenten der Anlage auf logische Weise zu betrachten.

[0146] Diese Arten von Anzeigenschichtung sind ausführlicher durch das Diagramm der [Fig. 22](#) dargestellt. Insbesondere kann eine allgemeine oder Gesamt-Anlagenübersichtsanzeige **600** eine Anzeige bereitstellen, die die grundlegende oder höhere Struktur der gesamten Anlage darstellt, obwohl sie möglicherweise nicht viele Einzelheiten über jeden bestimmten Abschnitt oder Teil der Anlage bereitstellt. Von der Anzeige **600** könnte ein Benutzer in der Lage sein (unter Verwendung von Benutzeroberflächenknöpfen) jeden bestimmten Abschnitt oder Bereich der Anlage auszuwählen oder dorthin zu gehen, um eine oder mehrere Anzeigen **602** des Bereichs A oder eine oder mehrere Anzeigen **604** des Bereichs B der Anlage hochzuziehen. Wie in der [Fig. 22](#) angezeigt, können dem Bereich A n getrennte Anzeigen zugeordnet sein, die alle logisch in Reihe miteinander verbunden sind, um den Prozeßfluß durch diesen Bereich der Anlage oder irgendeine andere, dem Bereich A der Anlage zugeordnete logische Struktur widerzuspiegeln. Auf die n Anzeigen des Bereichs A mit den Bezeichnungen **602a**, **602b**, ... **602n** kann unter Verwendung einer Aktionsart Vorwärtsblättern oder Rückwärtsblättern zugegriffen werden, bei der der Benutzer von einer Anzeige zur nächsten rollen kann. Auf diese Weise kann ein Benutzer die Anzeigen von einem Abschnitt des Bereichs A zu einem anderen Abschnitt des Bereichs A leicht auf eine Weise durchrollen, die für den Bediener einen Sinn ergibt. Weiterhin kann der Bediener wie in [Fig. 21](#) dargestellt vordefinierte Informationen oder sonstige Anzeigeninformationen aus getrennten der Anzeigen **602a–602n** erhalten. So kann der Bediener bei Betrachtung der

Anzeige **602a** des Bereichs A in der Lage sein, eine weitere Anzeige **610** zu erhalten, die eine vordefinierte Tendenz von Schlüsselparametern in der Anzeige **602a** zeigt. Auf gleiche Weise kann der Bediener bei Betrachtung der Anzeige **602b** in der Lage sein, auf eine Liste oder Schrift **612** mit Hochlauf- und Herunterfahrprozedurinformationen zuzugreifen. Natürlich könnte der Benutzer vorwärts oder rückwärts zwischen den Anzeigen **602a**, **602b** usw. rollen.

[0147] Auf gleiche Weise kann der Benutzer bei Betrachtung des Bereichs B, der mit m Anzeigen **604a–604m** dargestellt ist, zwischen Anzeigen auf dem gleichen Detailniveau unter Verwendung von Vorwärts- und Rückwärtsknöpfen (nächster und zurück) in den Anzeigen rollen, so daß es einem Bediener oder anderem Benutzer möglich ist, auf Anzeigen zuzugreifen, die Informationen stromaufwärts oder stromabwärts des angezeigten Abschnitts des Prozesses enthalten. Zusätzlich können Dynamos oder sonstige Benutzeroberflächenknöpfe bereitgestellt werden, damit auf eine andere Anzeige zugegriffen werden kann, um andere, Elementen in der aktuellen Anzeige zugeordnete Informationen zu erhalten. Durch Verwendung dieser Werkzeuge ist es möglich, eine Anzeigehierarchie zu erstellen, aus der eine Übersichtsanzeige zum Zugreifen auf die Schlüsselanzeige in jedem Prozeßbereich benutzt werden kann.

[0148] Zusätzlich können nach der Darstellung in [Fig. 20A–Fig. 20E](#) Anzeigen so beschichtet sein, daß die zahlreichen Anzeigen für den gleichen Abschnitt oder Teil einer Anlage bestehen, aber für unterschiedliche Funktionszwecke wie beispielsweise Bedieneraktionen, Wartungsaktionen, Simulationsaktionen, Geschäftsaktionen, technische Aktionen usw. benutzt werden. Diese getrennten Funktionsanzeigen sind in der [Fig. 22](#) als zwischen den Anzeigen **602a**, **602b** usw. geschichtet dargestellt und es kann aus sie sofern gewünscht voneinander aus zugegriffen werden. Wenn daher ein Bediener oder Benutzer die Anzeige **602a** betrachtet, kann der Benutzer zwischen den anderen Funktionsanzeigen für diesen Teil der Anlage, wie beispielsweise der Wartungsansicht, Simulationsansicht, Geschäftsansicht usw. umschalten oder darauf zugreifen, wenn ein solcher Zugriff erlaubt ist. Natürlich ermöglicht die schichtweise Anordnung der Wartungsansicht, der Bedieneransicht, der Simulationsansicht usw. auf diese Weise unter Verwendung der gleichen Grund-Anzeigeelemente ein leichtes Umschalten zwischen diesen Ansichten und ein besseres Verständnis des Geschehnisses in der Anlage bezüglich der unterschiedlichen Funktionen in der Anlage.

[0149] Zur Implementierung dieser und anderer Funktionalität können Grafikanzeigen erstellt werden, um eine Anzeige der für die Grafikanzeige beabsichtigten Rolle (oder Funktion) einzuschließen. Sol-

che Rollen können beispielsweise eine Frontplattenanzeige, eine Einzelheitenanzeige, eine Primärsteuerungsanzeige, eine schematische Anzeige, eine Wartungsanzeige, eine Geschäftsanzeige, eine Simulationsanzeige oder eine beliebige andere vom Benutzer definierte Rolle umfassen. Die Rolle ist Teil der Grafikanzeige und kann, wenn die Grafikanzeige einem Steuerungsmodul oder einem Hardwaregerät zugeordnet ist, zum Definieren der Verwendung und des Zugriffs auf diese Anzeige während der Laufzeit benutzt werden. Die Anzeige der der Grafikanzeige zugewiesenen Rolle kann dazu benutzt werden, anzuzeigen, wo und welchen Teilen des Konfigurationssystems oder der Anlage eine bestimmte Anzeige zugewiesen werden kann. Zusätzlich können diese Rolleninformationen zur Bestimmung benutzt werden, auf welche Anzeigen ein bestimmter Benutzer zugreifen kann, auf Grundlage der vom Benutzer durchgeführten Arbeit. Beispielsweise können Steuerwartenbediener möglicherweise nicht Grafikanzeigen betrachten oder darauf zugreifen, die als Simulations- oder Geschäftsanzeigen definiert sind.

**[0150]** Die oben beschriebenen Grafikanzeigenfähigkeiten können auch zur Erstellung von Sonderanzeigen benutzt werden, damit der Zustand von kritischen Geräten leicht überwacht werden kann. Einige Beispiele dieser Arten von Anzeigen umfassen eine First-out-Anzeige bei einer Prozeßstilllegung, Schwingungsüberwachung, Brennerverwaltung, Rußgebläsebetrieb und Sicherheitssystemzustand. Natürlich können die zugehörigen Anzeigen so strukturiert sein, daß sie die Informationen zusammenfassen und in Fällen, wo sich bewegende Geräte Animierung umfassen, beispielsweise ein Rußgebläse, kann Animierung wirkungsvoll dafür benutzt werden, einem Bediener schnell Zugang oder Verständnis der Funktionsweise des Systems zu vermitteln. Zusätzlich können die Berechnungsfähigkeiten der meisten Steuerungssysteme zum Implementieren von Online-Berechnung von Betriebskosten, Wirkungsgrad usw. benutzt werden, und diese Informationsart kann leicht in die Grafikanzeige des Bediener eingebaut werden, so daß der Bediener diese Informationen zur Verbesserung der Prozeßoperation benutzen kann.

**[0151]** Auch können verschiedene Verfahren zum Integrieren von Untersysteminformationen im Steuerungssystem benutzt werden, so daß standardmäßige Grafiken und Dynamos zur Erstellung von Bedieneranzeigen benutzt werden können, um Zugang zu Untersysteminformationen von einer höheren Anzeige aus zu ermöglichen. In manchen Fällen kann 3D-Auswertung von Matrixwerten zur Darstellung von Informationen (z. B. Blechlehreninformationen) durchgeführt werden.

**[0152]** Zusätzlich ist zu bemerken, daß die Grafikelemente und Anzeigen vorteilhafterweise unter Verwendung von Vektorgrafik wie der durch Micro-

soft-Avalon-Steuerungen bereitgestellten zur Bereitstellung von Flexibilität und Geschwindigkeit implementiert werden können. Die Verwendung von Vektorgrafik bietet die Fähigkeit zur Verwendung von skalierbaren Grafiken, damit die erstellte Anzeige skaliert und dimensioniert werden kann, um zu einer bestimmten Anzeigemaschine zu passen, und um Animierungen bereitzustellen, die Skalierung benutzen.

**[0153]** Allgemein gesagt und wie ausführlicher in der gleichzeitig anhängigen Anmeldung mit dem Titel Erstellung von jeder Prozeß-Grafikanzeige und darin enthaltenen Grafikanzeigeelementen beschrieben, ist in einer in einem ersten beschreibenden Format aufgeführten jeweiligen Textbeschreibung aufgezeichnet. Durch Skriptbefehle wird trotz der Kompliziertheit der widerzugebenden Grafiken ein wirkungsvoller nichtspeicherintensiver Mechanismus zum Definieren der Anzeigen bereitgestellt. Das beschreibende Format und daher die Skriptbefehle können auf einer beliebigen einer Anzahl unterschiedlicher Auszeichnungssprachen beruhen. Insbesondere kann eine auf XML basierende Auszeichnungssprache verlässlicherweise einen (auch XML-Fleck genannte) Widergabedefinition für jede Anzeige und jedes Anzeigeelement aufführen und dieser XML-Fleck kann zum Speichern dieser Anzeigen und Elemente in einer Konfigurationsbibliothek oder -datenbank vor ihrer Herunterladung in einer Prozeß-Laufzeitumgebung benutzt werden. Zum Unterstützen fortgeschrittener Grafiken wie beispielsweise Animierung kann die Auszeichnungssprache auch solche Grafiken entsprechend einem Vektorgrafikschema definieren.

**[0154]** Wie oben angezeigt und insbesondere in der gleichzeitig anhängigen Anmeldung mit dem Titel "Auszeichnungssprachen basierende dynamische Prozeßgrafiken in einer Prozeßanlagen-Benutzeroberfläche" beschrieben, die durch Bezugnahme aufgenommen worden ist, soll die dynamische Beschaffenheit der Prozeßgrafik aktuelle Werte der Prozeßanlagenparameter mit sich ändernden On-line- oder Simulationszuständen widerspiegeln. Dahingehend können die Prozeßgrafiken mit Quellen für die diese Änderungen widerspiegelnden Daten verbunden sind. Jede XML-basierende Beschreibung kann dementsprechend einen oder mehrere Datenquellenverweise enthalten, die im allgemeinen einen Datenquellenort für jeden entsprechenden dynamischen Grafikparameter (z. B. die sich ändernde Farbe eines Behälterinneren) identifizieren, der entsprechend den Daten abzuändern ist. Der Datenquellenort kann auch für spätere Angabe während der Konfiguration über die Editoren offengelassen werden, so daß das Skript ein Pseudonym oder einen Platzhalter zur Bezugnahme auf die später anzugebende Datenquelle oder Weginformation identifiziert. Da die Datenquelleninformationen und sonstige Eigenschaften der

Prozeßgrafikanzeigen (z. B. Verhaltensweisen wie beispielsweise Ereignisbehandlung) über die XML-basierende Beschreibung angegeben werden, kann die XML-basierende Sprache als PGXML bzw. Prozeßgrafik-XML bezeichnet werden.

**[0155]** Nach Beendigung der Konfigurations- und Entwurfsarbeit mit dem Ziel zum Definieren einer Prozeßgrafikanzeige und ihrer Bestandteile entscheidet sich der Konfigurationsingenieur oder sonstige Benutzer gegebenenfalls, die PGXML-Beschreibung als Vorbereitung zum Herunterladen der Prozeßgrafiken zu den Bedienerarbeitsplätzen oder sonstigen Benutzeranzeigevorrichtung zu verarbeiten. Allgemein gesagt wird jede PGXML-Beschreibung der Grafikanzeigen und Anzeigenelementen verarbeitet, um (i) Skript in einem zu einer zu benutzenden Grafikwiedergabemaschine kompatiblen Vektorgrafikformat zu erzeugen und (ii) Code mit Anweisungen, die die Datenquellenverweise und sonstige Nichtgrafikfunktionalität (z. B. Verhaltensweisen) der Anzeige angeben. Das Vektorgrafikformat des Skripts kann auch eine beschreibende bzw. XML-basierende Sprache sein. In Ausführungsformen mit einer Microsoft-Avalon-Benutzeroberflächenarchitektur kann das Vektorgrafikskript in Microsoft XAML ausgeführt sein. Andere Ausführungsformen können das offene Quellformat SVG (Scalable Vector Graphics) benutzen. Der Code kann in C# oder sonstiger geeigneter Programmiersprache aufgeführt sein.

**[0156]** In manchen Ausführungsformen wird das Vektorgrafikskript und der zugehörige Code dann kombiniert und in Dateien kompiliert, die ausführbare Befehle für den Bedienerarbeitsplatz oder sonstige Benutzeranzeigevorrichtung aufführen. Dahingehen kann für jede Prozeßgrafikanzeige und jedes Grafikanzeigeelement eine jeweilige DDL (dynamic-link library)-Datei erstellt werden. Auf alle Fälle kann eine solche Kompilierung des Vektorgrafikskripts und zugehörigen Codes vor dem Herunterladen durchgeführt werden, um Netzdatenübertragungserfordernisse zu minimieren.

**[0157]** Nach ihrer Erstellung können Grafikelemente und Grafikanzeigen als generische oder Schablonenobjekte in einer Konfigurationsdatenbank gespeichert werden und können als klassenbasierende oder nichtklassenbasierende Objekte oder Elemente gespeichert werden, bevor sie in einer Laufzeitumgebung benutzt werden. Allgemein gesagt können die hier besprochenen Grafikelemente und Grafikanzeigen Modulen in der Form von Anzeigemodulen sein, so wie sie in US-Veröffentlichung Nr. 2004/0199925 beschrieben sind, die hier durch Bezugnahme aufgenommen wird. Ein Klassenmodul (-objekt) ist eines, das nicht an irgendwelche Hardware oder Vorrichtung in einer Prozeßanlage oder einem Prozeßsteuerungssystem angebunden ist, sondern ein Objekt ist, aus dem andere Objekte, die an die Prozeßanlage

und das Prozeßsteuerungssystem angebunden sind, konkretisiert werden können. Allgemein gesagt kann zum Konfigurieren eine Prozeßanlage und insbesondere einer Laufzeitumgebung einer Prozeßanlage eine Konfigurationsmaschine wie beispielsweise eine der in dem Konfigurationsarbeitsplatz **20** der [Fig. 1](#) gezeigten Konfigurationsanwendungen **33** dazu benutzt werden, die Grafikobjekte (einschließlich von Grafikelementen und Grafikanzeigen) anderen logischen und physikalischen Instanzen in der Prozeßanlage einschließlich von logischen Instanzen wie beispielsweise Kontrollmodulen, Gerätemodulen (z. B. Einheitsmodulen), Prozeßmodulen usw. oder physikalischen Instanzen wie beispielsweise Bedienerarbeitsplätzen oder sonstigen Anzeigevorrichtungen zuzuordnen. In manchen Fällen können die Grafikobjekte Klassenobjekte sein und können anderen Klassenobjekten wie Gerätemodul-Klassenobjekten oder Steuermodul-Klassenobjekten zugeordnet sein oder können die Grafikobjekte einzelner Objekte sein und können konkretisierten Objekten zugeordnet sein, wie beispielsweise Objekten, die an Geräte in der Prozeßanlage angebunden worden sind und beispielsweise zu einer Steuerung, einem Arbeitsplatz oder einem sonstigen Gerätstück in der Prozeßanlage heruntergeladen worden sind.

**[0158]** So können die oben identifizierten Grafikelemente und Grafikanzeigen in einem Prozeßanlagenkonfigurationssystem konfiguriert und zur Hardware in einer Prozeßanlagenlaufzeitumgebung heruntergeladen werden, auf ähnliche Weise wie die Weise, auf die andere Klassenobjekte in einer Bibliothek gespeichert und darin konfiguriert sind und dann zu verschiedenen Abschnitten oder Teilabschnitten des Prozeßsteuersystems heruntergeladen werden, wie in US-Veröffentlichung Nr. 2004/0199925 beschrieben. Insbesondere wird durch die Konfigurationsmaschine, die an Daten in einer Konfigurationsdatenbank wie der Datenbank **28** der [Fig. 1](#) operiert, eine Konfigurationsdatenbankhierarchie wie beispielsweise der in [Fig. 23](#) und [Fig. 24](#) dargestellten benutzt, um einem Benutzer die Betrachtung und Änderung der Konfiguration der Prozeßanlage zu ermöglichen. Das Konfigurationshierarchieschirmbild **700** der [Fig. 23](#) zeigt, daß die Konfigurationsdaten für das (als DeltaV-Steuersystem bezeichnete) Steuersystem **702** einen Bibliotheksteil **704**, einen Systemkonfigurationsteil **706** und einen Suchergebnisteil **708** enthält.

**[0159]** Der Bibliotheksteil **704** enthält Klassenobjekte und andere Objektdefinitionen, die nicht zugewiesen oder zur Laufzeitumgebung heruntergeladen worden sind, sondern stattdessen als Schablonen oder generische und ungebundene Objekte gespeichert sind. Wie in [Fig. 23](#) angezeigt, enthält der Bibliotheksteil **704** einen Steuermodulteil **710**, einen Geräteteil **712**, einen Prozeßmodulteil **714** und einen Bedieneroberflächenteil **716** wie auch einen Char-

genteil und einen Sicherheitsinstrumentsystemteil (die nicht durch eine Zahl angezeigt sind).

**[0160]** Wenngleich der Steuermodulteil **710** nicht erweitert ist, enthält dieser Teil der Konfigurationsdatenbank im allgemeinen die unterschiedlichen Steuermodulen wie beispielsweise Steuermodulschablonen und Steuermodulklassenobjekte, die für die Prozeßanlage erstellt worden sind, um in ihren verschiedenen Steuerungen und sonstigen Steuervorrichtungen in der Prozeßanlage benutzt zu werden. Weiterhin enthält der Geräteteil **712** Anzeigen von Geräteobjekten, die Anzeigen von tatsächlichen, in der Prozeßanlage benutzten Geräten und Gerätearten enthalten könnten, wie beispielsweise Ventile, Sensoren, Steuerungen, Arbeitsplätze, wie auch Anzeigen von Geräteklassenobjekten, Einrichtungsklassenobjekten usw., die unterschiedlichen Geräten oder Hardware in der Prozeßanlage zugeordnete logische Elemente definieren. Weiterhin enthält der Prozeßmodulteil **714** die für die Prozeßanlage erstellten verschiedenen Prozeßmodulen einschließlich von Prozeßmodulschablonen und für diese Prozeßmodulen definierte Prozeßblockdefinitionen. Solche Prozeßmodulen sind ausführlich in der US-Veröffentlichung Nr. 2004/0199925 beschrieben und werden daher hier nicht weiter beschrieben.

**[0161]** Wie in [Fig. 23](#) angezeigt speichert jedoch der Bedieneroberflächenteil **716** der Konfigurationshierarchie **700** den oben beschriebenen Grafikobjekten zugeordnete Informationen. Insbesondere enthält ein Grafikteil **720** verschiedene zusammengesetzte Prozeßgrafikdefinitionen, Grafikklassenobjekte und Grafikschemata, die die verschiedenen Grafikelemente definieren, und Grafikanzeigen, die für das System erstellt worden sind. Es ist zu bemerken, daß der Teil für zusammengesetzte Prozeßgrafikdefinitionen im allgemeinen die für das System erstellten Grafikelemente enthält, der Prozeßgrafikklassenteil die auf die oben beschriebene Weise erstellten Grafikanzeigen als Klassenobjekte enthält, während die Prozeßgrafikschemata nichtklassenbasierende Grafikanzeigen und sonstige Schablonenobjekte enthalten kann. Die in diesem Teil gespeicherten Grafikobjekte der Konfigurationsdatenbank sind im allgemeinen ungebundene Objekte, die zur Erstellung anderer Objekte benutzt werden können, und die während eines Konfigurationsverfahrens zum Definieren der Art und Weise benutzt werden können, auf die Grafikelemente und Grafikanzeigen an die verschiedene Hardware und Software in der Prozeßanlage anzubinden sind. Weiterhin definiert ein Layoutteil **722** unter dem Bedieneroberflächenteil **716** verschiedene Layouts für die Bedieneroberflächen, während ein Rollenteil **724** verschiedene Rollen definiert, die auf Bedieneroberflächen implementiert werden können, wie auch Rollen, die durch verschiedene der Grafikanzeigen und -elemente erfüllt werden können.

**[0162]** So kann die durch die Konfigurationshierarchie **700** dargestellte Konfigurationsdatenbank einen Teil enthalten, der Grafikobjekte mit Grafikelementen und Grafikanzeigen als generische Schablonen, als Klassenobjekte oder als spezifische oder einzelne Elemente oder Anzeigen speichert, die, obwohl sie nicht an bestimmte Prozeßinstanzen angebunden sind, zum Definieren von Elementen oder Anzeigen benutzt werden können, die an spezifische Prozeßinstanzen angebunden werden. Allgemein gesagt kann ein Konfigurationsingenieur oder sonstiger Benutzer die Hierarchie **700** zum Konfigurieren einer Prozeßanlage auf eine ausführlicher unten besprochene Weise benutzen, so daß sie Grafikanzeigefähigkeiten enthält.

**[0163]** [Fig. 24](#) zeigt ein Hierarchieschirmbild **730**, in dem der Systemkonfigurationsteil **706** erweitert worden ist, um darzustellen, wie Prozeßgrafikelemente und -grafikanzeigen verschiedenen logischen und physikalischen Instanzen der Prozeßanlage als Teil einer Systemkonfiguration zugeordnet werden können. Insbesondere definiert ein Prozeßgrafikteil **734** unter dem Systemkonfigurationsteil **706** verschiedene Grafikanzeigen, die allgemein zum Herunterladen zur Prozeßanlage zur Verwendung in einer Laufzeitumgebung definiert worden sind. Allgemein gesagt werden die Grafikanzeigen und Elemente in dem Prozeßgrafikteil **734**, die beispielsweise Fontdatenanzeigen, Bedieneranzeigen, Detailanzeigen, Wartungsanzeigen, Simulationsanzeigen usw. enthalten können, zu jeder Anzeigeeinheit (z. B. jedem Arbeitsplatz oder sonstigen Anzeigevorrichtung) in der Prozeßanlage heruntergeladen, da der Prozeßgrafikteil **734** für die gesamte Systemkonfiguration gilt. Grafikelemente und -anzeigen können stattdessen jedoch bestimmten Teilen der Systemkonfiguration einschließlich eines Steuerungsstrategieteils **736** und eines physikalischen Netzteils **740** zugeordnet sein, wodurch die Anzeigevorrichtungen begrenzt sein können, auf denen diese Grafikelemente und -anzeigen zur Verfügung stehen.

**[0164]** Allgemein gesagt definiert der Steuerungsstrategieteil **736** unterschiedlichen physikalischen und logischen Teilen der Prozeßanlage zugewiesenen Steuerungsroutrinen oder dahingehend implementierten Strategien. Verschiedene der Grafikanzeigen (die wiederum beliebige Arten von Anzeigen wie beispielsweise Frontplattenanzeigen, Steuerungsanzeigen, Wartungsanzeigen usw.) sein können, können bestimmten Teilbereichen oder Unterkategorien des Steuerungsstrategieteils **736** zugewiesen sein, einschließlich beispielsweise Bereichen (wie beispielsweise den Bereichen mit Bezeichnungen Area\_A742 und Area\_B744), zu Bereichen zugewiesenen Steuerungsmodulen wie beispielsweise einem Steuerungsmodulteil **746** und Bereichen zugewiesenen Prozeßmodulen wie beispielsweise einem Prozeßmodul **748**. So ist wie in

[0165] **Fig. 24** dargestellt ein Anzeigeteil **750** dem Area\_A-Teil **742** zugewiesen, der alle Anzeigen definiert, die für und im Bereich Area A der Anlage benutzt werden. Die nicht geöffneten Anzeigen unter dem Ordner **750** könnten verschiedene, unterschiedlichen Mengen von Hardware im Area A zugeordnete Anzeigen, wie beispielsweise die unterschiedlichen Anzeigen **602a–602n** der **Fig. 22**, wie auch verschiedene, aus diesen Anzeigen zugängliche Unteranzeigen, wie beispielsweise Anzeigen **610** und **612** der **Fig. 22**, umfassen. Die für die Anzeigen definierten Rollen könnten die Art und Weise anzeigen, auf die diese Anzeigen voneinander aus zugänglich oder für einen eine Anzeigevorrichtung benutzenden Bediener zugänglich sind. Wie bezüglich der **Fig. 22** beschrieben könnte daher ein Benutzer die Anzeigen im Teil **750** durchrollen, um verschiedene Teile des Area\_A der Anlage zu betrachten oder weitere Einzelheiten bezüglich eines bestimmten Bereichs zu betrachten, einschließlich ausführlicher Anzeigen bestimmter Teile einer höheren Anzeige, die nicht mit solchen Einzelheiten in der höheren Anzeige dargestellt ist.

[0166] Weiterhin können Anzeigen, wie durch das Displays-Icon **752** unter dem Steuerungsmodul-Teil **746** angezeigt, bestimmten Steuerungsmodulen im Area A-Teil **742** zugewiesen werden. In diesem Fall sind die Anzeigen **752** einer (mit LOOP bezeichneten) Steuerungsmodulschleife zugeordnet und können spezifisch dieser bestimmten Regelschleife zugeordnete Steuerwartenbedieneranzeigen sein. Eine für eine Anzeige definierte Rolle kann wiederum ihre Rolle in einem Bedienerplatz als Teil einer Steuerungsanzeige anzeigen. Wie bezüglich des Area B-Teils **744** dargestellt, können Anzeigen **754** weiterhin allgemein dem Bereich zugeordnet sein, während Anzeigen **756** einem Bereich zugewiesenen bestimmten Prozeßmodulen zugeordnet sein können. In diesem Fall ist ein Display-Icon **756** einem mit PMOD2 bezeichneten Prozeßmodul zugeordnet, das ein dem Gerät im Area\_B zugeordnetes Prozeßmodul ist. Das PMOD2 kann beispielsweise ein Einheitsmodul oder ein Gerätemodul oder eine beliebige andere Art von Prozeßmodul sein.

[0167] So können, wie allgemein unter den Steuerungsstrategieteilen **736** angezeigt, Anzeigen bestimmten Steuerungsstrategien oder Steuerungsdefinitionen zugeordnet sein, da diese Steuerungsdefinitionen innerhalb des Konfigurationssystems erstellt werden. Wie in dem Beispiel der **Fig. 24** gezeigt, können Anzeigen Steuerungsdefinitionen zugeordnet sein, die auf bereichsweiser Basis definiert sind, wie beispielsweise für die Bereichsabschnitte **742** und **744**, wie auch auf einer steuerungsmodulebenen Basis wie beispielsweise für Steuerungsmodulen **750** und auf einer Prozeßmodulenbasis wie durch den Anzeigenteil **756** angezeigt. Zusätzlich können gegebenenfalls Geräte Gegenstand der Rollenzuordnung

sein, wodurch auf die diesen Rollen zugeordneten Anzeigen dynamisch von Geräteelementen aus zugegriffen werden kann. Solche Geräteelementen und -anzeigen können im Prozeßgrafikteil **734** gespeichert sein.

[0168] Zusätzlich können Anzeigen auch eigentlichen Hardwareelementen in der Prozeßanlage einschließlich von Bedienerarbeitsplätzen und sonstigen Anzeigevorrichtungen zugewiesen oder diesen zugeordnet sein. Wie unter dem Physical Network-Teil **740** angezeigt, können daher spezifische Anzeigen spezifischen Bedienerarbeitsplätzen oder sonstigen Laufzeitmaschinen zugewiesen sein. In dem bestimmten Beispiel der **Fig. 24** enthält der Physical Network-Teil **740** ein Steuerungsnetz **760**, das zugehörige Hardware wie beispielsweise eine Steuerung mit der Bezeichnung CTLR1, eine Pro Plus-Station, die eine Konfigurationsstation sein kann, und einen Bedienerarbeitsplatz mit der Bezeichnung Operator Station **762** aufweist. Die Operator Station **762**, die einen bestimmten Arbeitsplatz oder eine bestimmte Benutzerschnittstelle in der Anlage definiert oder dieser zugeordnet ist, enthält Alarmer und Ereignisfunktionen, Bedieneraktivitäten, eine kontinuierliche Historikerfunktion und damit verbundene Simulationsaktivitäten, obwohl einer bestimmten Benutzerschnittstelle andere Aktivitäten oder Funktionen wie beispielsweise Wartungsaktivitäten, Geschäftsaktivitäten usw. zugeordnet sein oder darauf durchgeführt werden könnten. Auf diese Weise können bestimmten Benutzerschnittstellen Rollen in der Prozeßanlage zugewiesen werden und die Anzeigen für diese Rollen in der Anzeigevorrichtung gespeichert oder dorthin heruntergeladen werden.

[0169] Wie in **Fig. 24** angezeigt, werden Anzeigen **764** und **766** der Bedienerfunktion oder den Bedieneraktivitäten wie auch der oder den durch den Arbeitsplatz **762** durchgeführten Simulationsfunktion oder Aktivitäten zugewiesen. Obwohl dies nicht in **Fig. 24** dargestellt ist, können Anzeigen anderen Anzeigevorrichtungen, z.B. den anderen Knoten in der Prozeßanlage zugeordneten zugewiesen werden, um auf diesen Anzeigevorrichtungen implementiert zu werden. Weiterhin kann die Zuweisung von Anzeigen zu bestimmten Funktionen in den Bedienerarbeitsplätzen oder anderen Laufzeit-Anzeigemaschinen die Rollen oder Funktionen bestimmen, die diese Anzeigen in der Laufzeitumgebung durchführen können, wie beispielsweise eine Bedienerfunktion oder Simulationsfunktion, eine Wartungsfunktion usw. Natürlich können diese verschiedenen Funktionen auf eine durch das Konfigurationssystem eingestellte Weise auf der gleichen Anzeigevorrichtung oder auf unterschiedlichen Anzeigevorrichtungen durchgeführt werden. Auch können, während die Konfigurationshierarchie **730** einem Bedienerarbeitsplatz zugewiesene Anzeigen darstellt, Anzeigen anderen Arten von Anzeigevorrichtungen einschließlich

von-Hand-Computeranzeigevorrichtungen wie PDAs, Telefonvorrichtungen, Geschäftsarbeitsplätzen oder sonstigen erwünschten Arten von Anzeigevorrichtungen zugewiesen werden.

**[0170]** Mit der Konfigurationsmaschine kann ein Benutzer wie beispielsweise ein Konfigurationsingenieur die Konfigurationshierarchie zum Definieren und Verwalten der Art und Weise benutzen, auf die die Prozeßgrafikanzeigen an die Prozeßanlage angebunden und darin heruntergeladen werden. Allgemein gesagt kann ein Benutzer wie beispielsweise ein Konfigurationsingenieur Anzeigeobjekte innerhalb des Konfigurationssystems auf die gleiche Weise wie ein Benutzer es bezüglich anderer Objekte tun kann erstellen, löschen, umbenennen, zuweisen oder herunterladen. Insbesondere ermöglicht die Konfigurationsmaschine die Integration von Grafikanzeigen in die Konfiguration der Prozeßanlage indem definiert wird, wohin (zu welchen Vorrichtungen) diese Anzeigen zur Ausführung zuzuweisen sind und die (logischen und physikalischen) Prozeßinstanzen, an die diese Anzeigen während der Laufzeit angebunden sein sollten. Wenn gewünscht kann ein Konfigurationshierarchieschirmbild wie die der [Fig. 23](#) und [Fig. 24](#) zur Anzeige von Herunterladungen benutzt werden, die benötigt werden und zum Implementieren von Versionssteuerung auf den Anzeigen.

**[0171]** Konfiguration kann auf mehreren Ebenen im Konfigurationssystem ausgeführt werden. An erster Stelle kann ein Konfigurationsingenieur zuerst ein Anzeigeklassenobjekt einem Modulklassenobjekt im Bibliotheksteil **704** zuordnen, wodurch ein bestimmtes Anzeigeklassenobjekt an ein bestimmtes Modulklassenobjekt wie beispielsweise ein Gerätemodulklassenobjekt, ein Bereichsmodulklassenobjekt usw. angebunden wird. Danach werden, wenn das Modulklassenobjekt konkretisiert und einem bestimmten Teil der Prozeßanlage zugewiesen (oder sonstwie konfiguriert) ist, die dem konkretisierten Modulobjekt zugeordneten Anzeigeobjekte (z.B. Grafikanzeigen) zusammen mit dem Modulobjekt an die gleiche Hardware angebunden und so ist kein zusätzlicher Aufwand erforderlich, eine Grafikanzeige der zutreffenden Hardware in der Anlage zuzuweisen, oder dieses Grafikanzeigenobjekt zu der zutreffenden Hardware herunterzuladen. Stattdessen folgt das Grafikanzeigenobjekt in diesem Fall der Konfiguration des Modulobjekts, an das es angebunden ist. Wenn weiterhin ein Anzeigeklassenobjekt einem Modulklassenobjekt wie oben beschrieben zugeordnet ist, können am Modulklassenobjekt durchgeführte Änderungen die automatische Ausbreitung dieser Änderungen zu diesen gleichen Elementen in den zugeordneten Anzeigeklassenobjekt veranlassen und diese Änderungen können dann zu tatsächlichen konkretisierten Versionen von Grafikanzeigen herunterverbreitet werden. Wenn in einem Beispiel eine Grafikanzeige einen Mischer (Geräteklassenobjekt) enthält, der ein LOOP

genanntes Steuerungsmodul enthält, und dieses Steuerungsmodul ausgewählt und umbenannt wird, kann das Konfigurationssystem automatisch die Verweise zur LOOP in der Grafikanzeige auf den neuen Namen ändern um damit sicherzustellen, daß die Grafikanzeigeanbindungen mit dem neuen Namen aktualisiert sind. Die eigentlichen neu geänderten Anzeigen müssen jedoch gegebenenfalls immer noch zur Laufzeitmaschine heruntergeladen werden, damit die Änderungen im Laufzeitsystem stattfinden. Wenn gewünscht können Anzeigen, daß eine Herunterladung erforderlich ist, den in der Konfigurationshierarchie gespeicherten Anzeigen benachbart platziert werden, wie beispielsweise durch Platzieren eines blauen Dreiecks neben den zutreffenden des Grafikanzeigenikons.

**[0172]** Andererseits können Grafikanzeigen getrennt oder direkt unterschiedlichen Komponenten in der Prozeßanlage oder dem Konfigurationssystem zugewiesen werden. Insbesondere können wie in [Fig. 24](#) dargestellt einzelne Grafikanzeigen in den Prozeßgrafikteil **734** unter Systemkonfigurationsteil **734** platziert werden. Diese Anzeigen können in diesen platziert werden, indem sie beispielsweise im Bibliotheksteil **702** ausgewählt und zum Teil **734** gezogen und dort abgelegt werden. Natürlich können auch andere Weisen zum Bewegen von Grafikanzeigen zu bestimmten Teilen der Hierarchie **730** benutzt werden. Auf alle Fälle könnten die Grafikanzeigen im Teil **734** beispielsweise Frontplattenanzeigen oder sonstige klassenbasierende oder nicht klassenbasierende Anzeigen enthalten und können allgemein zu allen Arbeitsplätzen in der Anlage oder dem durch den Systemkonfigurationsteil **706** abgedeckten Teil der Anlage heruntergeladen werden.

**[0173]** Ein Benutzer kann jedoch Anzeigen aus dem Prozeßgrafikteil **734** heraus und in bestimmte Unterabschnitte verlegen, um die Identität der Arbeitsplätze (oder anderen Anzeigevorrichtungen) in dem Prozeßsteuerungssystem zu steuern, die Zugriff auf diese Anzeigen haben werden, und daher die Identität der Vorrichtungen, aus denen diese Anzeigen benutzt werden können, steuern. Zur Zuweisung einer bestimmten Anzeige zu einem logischen oder physikalischen Anlagenteil kann der Benutzer beispielsweise eine Grafikanzeige auswählen, die Grafikanzeige zu dem bestimmten Teil der Konfigurationshierarchie **730** ziehen und dort ablegen, um dadurch die Zugehörigkeit zwischen dieser Grafikanzeige und den logischen oder physikalischen Instanzen zu definieren, denen diese Anzeige zuzuordnen ist. Die Auflösungstabelle für eine Anzeige kann automatisch zur Zeit dieses Konfigurationsschritts auf Grundlage des Teils der Konfiguration, in der eine Anzeige abgelegt oder zugeordnet wird, ausgefüllt werden.

**[0174]** Beispielsweise können Grafikanzeigen aus dem Bibliotheksteil **702** auf den Prozeßgrafikteil **706**

gezogen werden, um die Anzeige der Laufzeitumgebung zuzuordnen. Auch können Grafikanzeigen beispielsweise im Process Graphics-Teil **734** in andere Teile wie beispielsweise in den Area\_A-Displays-Teil **750** verlegt werden (wodurch diese Anzeige für die Area\_A-Instanz und -Funktionalität spezifisch aber allgemein auf jeder Area\_A-Anzeigevorrichtung verfügbar wird), die Area\_B-Displays-Teil **754** usw. Gleichweise können Grafikanzeigen Teilabschnitten von Logikinstanzen zugeordnet werden, wie beispielsweise dem mit LOOP bezeichneten Steuerungsmodul (wie durch den Anzeigenordner **752** angezeigt) zugeordnet werden, wodurch diese Anzeigen für diese Logikinstanz spezifisch werden und nur auf Maschinen verfügbar sind, die diese Regelschleifenfunktion durchführen oder als hier zugeordnet definiert sind. Zusätzlich können Anzeigen spezifischen Anzeigevorrichtungen oder funktionsmäßigen Teilabschnitten von Anzeigevorrichtungen wie der durch die Bedienerfunktion in der Operator Station **762** oder der Simulationsfunktion **770** in der Operator Station **762** definierten zugewiesen werden. Obwohl dies nicht besonders im einzelnen gezeigt ist, kann der Benutzer auch Grafikanzeigen einem Bereich, einer Zelle, einer Einheit, einem Gerätemodul wie auch unterschiedlichen Steuerungsabschnitten zuweisen, um die Art und Weise zu definieren, auf die die Grafikanzeigen der Prozeßanlage zugeordnet sind und darin benutzt werden können.

**[0175]** Wenn eine Grafikanzeige einer logischen Instanz wie beispielsweise einer Steuerungsstrategie wie der durch einer Bereichsbezeichnung wie Area\_A und Area\_B definierten zugewiesen ist, ist die Grafikanzeige den physikalischen Schnittstellen (z.B. Arbeitsplätzen) zugewiesen, denen diese logischen Instanzen zugewiesen sind. Auf ähnliche Weise wird, wenn eine Anzeige einer Einheit, einer Zelle oder einem Gerätemodul zugewiesen ist, die Anzeige dem Arbeitsplatz oder sonstigen Schnittstellenvorrichtungen zugewiesen, denen diese Einheit, Zelle oder Gerät zugewiesen ist.

**[0176]** Wenn ein Modul mehrere Grafikanzeigen enthält, kann das Modul die Rollen für diese Anzeigen oder Rollen definieren, die jede Anzeige erfüllt, wie durch bei Erstellung der Anzeigen den Grafikanzeigen zugewiesene Anzeigen einer Rolle definiert. Zugang zu diesen Grafikanzeigen in jeder Grafikvorrichtung kann dann der für die Anzeige definierten Rolle zugeordnet oder durch diese Rolle begrenzt werden. Weiterhin kann, wenn eine Grafikanzeige einem Bereich oder einem Modul zugewiesen wird, ein Verweis auf diese Grafikanzeige im Displays-Ordner des Bereichs oder Moduls auftreten, zu dem sie gehört. Wenn einmal eine Grafikanzeige einem Bereich oder Modul zugewiesen ist, folgen die diesem Bereich oder Modul, und daher diese Anzeige, eigenen Anzeigeverweise dem Bereich oder Modul während jeder Zuweisung oder Verlegung des Bereichs oder

Moduls. Wenn andererseits ein Bereich oder Modul einem Arbeitsplatz zugewiesen ist, wird die zu diesem Bereich oder Modul gehörende Anzeige automatisch dem Bedienerteilsystem oder der Bedienerfunktion dieses Arbeitsplatzes nach Definition durch die Rolle der Anzeige zugewiesen werden. Natürlich können einzelne Grafikanzeigen direkt dem Bedienerteilsystem einer Anzeigevorrichtung zugewiesen werden, indem sie direkt in oder unter die bestimmte Schnittstellenvorrichtung oder das bestimmte Teilsystem der Schnittstellenvorrichtung im Physical Network-Teil **740** der Konfigurationshierarchie geplant werden.

**[0177]** Es versteht sich daher, daß Grafikanzeigen einzelnen Knoten, Steuerungsbereichen, Steuerungsmodulen, Prozeßmodulen oder sonstigen logischen Steuerungsinstanzen wie auch einzelnen Anzeigevorrichtungen und Teilsystemen derselben zugewiesen werden, indem sie zu diesen Instanzen im Konfigurationsschirmbild **730** gezogen und darin abgelegt werden. Natürlich kann die gleiche Grafikanzeige gegebenenfalls mehreren Stationen zugewiesen werden und der Zustand jeder Anzeige (heruntergeladen oder nur zugewiesen) kann durch ein für jeden Anzeigenamen in den verschiedenen Ordnern in der Hierarchie **730** bereitgestelltes Icon angezeigt werden. Wenn gewünscht können andere Icons in der Hierarchie **730** benutzt werden, um anzuzeigen, wann Grafikanzeigen im Speicher verriegelt sind oder ob sie im virtuellen Speicher auf der Platte unterhalten werden können.

**[0178]** Allgemein gesagt bestehen Grafikanzeigen aus einem Anzeigehalt (z.B. einem XAML-Skript, das die Weise definiert, auf die die Anzeige optisch auf einem Anzeigeschirm erscheint), lokalen Tabellen und Verweisen, die Verweise auf andere Anzeigen wie auch Verweise auf Laufzeitparameter wie beispielsweise Steuerungsparameter usw. sein können, Verweise auf Anzeigesteuern, Laufzeitpseudonyme (von denen einige zur Laufzeit anbinden können), lokale Tabellenverweise und globale Tabellenverweise. Allgemein gesagt behandelt die Konfigurationsmaschine den Grafikteil des Grafikelements (das XAML) getrennt von den Verweisen, wodurch es leichter wird, Grafikanzeigen zu behandeln und zu verwenden. Insbesondere kann das Konfigurationssystem eine Grafikanzeige während des Konfigurationsvorgangs anbinden, indem es die Verweise ausfüllt, die keine dynamischen Pseudonyme benutzen, aufgrund der in der Konfigurationshierarchie durchgeführten Zuweisungen. Wenn beispielsweise ein Benutzer eine Grafikanzeige aus dem Prozeßgrafikteil **734** oder aus dem Bibliotheksteil **704** der Konfigurationshierarchie **730** zu einem bestimmten Bereich oder Steuerungsmodul zieht, wird Anbindung automatisch in der Grafik durchgeführt, um die Verweise in der Grafikanzeige an die diesem Bereich oder Modul zugeordneten spezifischen Elemente anzubin-

den. Auf diese Weise können Herunterladungsanbindungen automatisch auf Grundlage der vom Benutzer durchgeführten Konfiguration durchgeführt werden, wodurch der Aufwand verringert wird, den der Benutzer von Hand zum Angeben dieser Anbindungen durchführen muß.

**[0179]** Wenn es weiterhin unaufgelöste Anbindungen im Konfigurationsschritt gibt, kann die Konfigurationsmaschine den Benutzer auffordern, Informationen zur Auflösung der Anbindungen wie beispielsweise zum Ändern der Anbindungen oder der Verweise bereitzustellen. Weiterhin können wenn gewünscht Volumeneditierwerkzeuge wie beispielsweise ein Kalkulationstabellenprogramm wie Microsoft Excel<sup>®</sup> dazu benutzt werden, einen Konfigurationsingenieur beim Ausfüllen oder Definieren von Auflösungen oder Anbindungsparametern während des Konfigurationsvorgangs zu unterstützen. Auf alle Fälle wird durch diese Konfigurationsschritte die bezüglich der [Fig. 13](#) besprochene Auflösungstabelle **208** ausgefüllt, wodurch die Grafikanzeigen in einer Laufzeitumgebung benutzt werden können. So kann der Benutzer Anbindungen und Herunterladen von Grafikanzeigen (und Grafikelementen) allgemein auf die gleiche Weise durchführen, wie es der Benutzer bezüglich der in US-Veröffentlichung Nr. 2004/0199925 beschriebenen Prozeßmodulen tut.

**[0180]** Natürlich kann der Benutzer oder Konfigurationsingenieur die Zuweisung grafischer Anzeigen durch Anklicken der Grafikanzeige und Entfernen derselben aus der Hierarchie **730** oder sonstiges Betätigen eines Dialogkastens zur Aufhebung der Zuweisung dieser Grafikanzeige aufheben. Weiterhin kann ein Benutzer die Art und Weise betrachten, auf die eine bestimmte Grafikanzeige oder Menge von Grafikanzeigen zugewiesen ist, und diese Informationen können beispielsweise durch einen in [Fig. 25](#) gezeigten Dialogkasten **800** bereitgestellt werden. Der Dialogkasten **800** der [Fig. 25](#) zeigt die Weise an, auf die jede von zwei Anzeigen Display 1 und Display 2 in der Prozeßanlage konfiguriert ist. Insbesondere zeigt der Dialogkasten **800** an, wo jede der Anzeigen benutzt wird (z.B. wird Display 1 in einem mit Area\_A bezeichneten Anlagenbereich und in dem Mod1 genannten Steuerungsmodul benutzt), die physikalische Station, der sie zugewiesen ist (z.B. ist Display 1 einem Bedienerplatz mit der Bezeichnung Oper1 zugewiesen), und die Rolle die die Anzeige erfüllt (z.B. ist Display 1 eine Hauptanzeige in jeder ihrer Verwendungsarten).

**[0181]** Es versteht sich, daß die Anzeigerollen auch auf jede gewünschte Weise geändert werden können und daß die Rollen zum Anzeigen eines Teils des Konfigurationssystems benutzt werden können, dem die Grafikanzeige zugewiesen werden kann oder die die Grafikanzeige benutzen kann, d.h. für welche Funktion. Grafikanzeigen können zum Erfüllen von

mehreren Rollen benutzt werden, sofern gewünscht. Wenn ein Benutzer einen Bereich oder ein Modul irgendeiner Art einem Arbeitsplatz oder einer sonstigen Anzeigevorrichtung zuweist, werden natürlich wie oben bemerkt alle diesem Bereich oder Modul zugeordneten Anzeigen ebenfalls dem Arbeitsplatz zugewiesen, um sicherzustellen, daß Grafikanzeigeunterstützung für diesen Bereich oder sonstiges Modul auf dem Arbeitsplatz bereitgestellt wird. Auch kann, wenn eine Grafikanzeige einem Arbeitsplatz oder einer sonstigen Anzeigevorrichtung zugewiesen wird, die Grafikanzeige einem Teilsystem des Arbeitsplatzes auf Grundlage der Rolle dieser Grafikanzeige zugewiesen werden, wie beispielsweise, ob die Grafikanzeige eine Simulationsanzeige, eine Bedieneranzeige usw. ist. Zusätzlich kann der Benutzer Grafikanzeigen auf bestimmte Teilsysteme eines Arbeitsplatzes oder einer sonstigen Anzeigevorrichtung ziehen und dort ablegen und wenn die Rolle nicht richtig zugewiesen ist, kann das Konfigurationssystem einen Dialogkasten darstellen, um dem Benutzer anzuzeigen, daß eine falsche Zuweisung stattfindet oder um dem Benutzer erlauben, die Rollen der Grafikanzeige zu ändern.

**[0182]** Nach Durchführung von zutreffenden oder gewünschten Zuweisungen unter Verwendung der Konfigurationsmaschine und der Konfigurationshierarchie **730** der [Fig. 23](#) und [Fig. 24](#) kann der Benutzer/Bediener eine tatsächliche Herunterladung der Grafikanzeigen durchführen, wodurch die Grafikanzeigen tatsächlich zu den zutreffenden Bedienerarbeitsplätzen nach Definition in der Konfigurationshierarchie **730** für die im Konfigurationssystem definierten Zwecke und Rollen heruntergeladen werden. Während dieses Herunterladens werden die Anbindungen in den Anzeigen in Abhängigkeit von den Zuweisungen der Bereiche, Schleifen, Steuerungsmodulen, Prozeßmodulen usw. wie auch zu den Bedienerarbeitsplätzen oder anderen Anzeigevorrichtungen aufgelöst.

**[0183]** Wenn gewünscht kann der (in [Fig. 23](#) gezeigte) Search-Results-Teil **708** der Konfigurationshierarchie einem Benutzer die Durchführung von Suchen nach irgendwelchen einzelnen Grafikanzeigen oder Grafikelementen ermöglichen, die in den Grafikanzeigen in der gesamten Prozeßanlage oder dem Konfigurationssystem benutzt werden. Insbesondere kann die Konfigurationsmaschine eine Suchmaschine enthalten, die, wenn sie vom Benutzer betätigt wird, Suchoperationen nach bestimmten Grafikanzeigen oder Grafikelementen durchführen kann und die Ergebnisse dieser Suchoperationen im Search-Results-Teil **708** speichern kann, um eine vollständige Auflistung der Verwendungszwecke jedes bestimmten Grafikelements oder jeder bestimmten Grafikanzeige für den Benutzer bereitzustellen. Solche Suchoperationen sind nützlich, damit Benutzer, die Änderungen durchführen, betrachten können, welche Gra-

fikanzeigen geändert werden müssen oder welche Herunterladungen stattfinden müssen, um diese Änderungen zu implementieren.

**[0184]** Nach Implementierung kann jede hier beschriebene Software in jedem vom Computer lesbaren Speicher wie beispielsweise auf einer Magnetplatte, auf einer Laserplatte oder sonstigem Speichermedium, in einem RAM oder ROM eines Computers oder Prozessors usw. gespeichert werden. Auf gleiche Weise kann diese Software einem Benutzer, einer Prozeßanlage oder einem Bedienerarbeitsplatz unter Verwendung jedes bekannten oder gewünschten Zuführungsverfahrens, einschließlich beispielsweise einer von einem Computer lesbaren Platte oder sonstigem transportierbaren Computer-Speichermechanismus oder über einen Kommunikationskanal, wie beispielsweise eine Telefonleitung, das Internet, das World Wide Web, ein beliebiges sonstiges Ortsnetz oder Weitverkehrsnetz usw. zugeführt werden (wobei diese Zuführung der Bereitstellung dieser Software über ein transportierbares Speichermedium ebenbürtig oder dafür austauschbar betrachtet wird). Weiterhin kann diese Software direkt ohne Modulation oder Verschlüsselung bereitgestellt werden oder kann vor Übertragung über einen Kommunikationskanal unter Verwendung einer beliebigen geeigneten Modulationsträgerwelle und/oder eines beliebigen Verschlüsselungsverfahrens moduliert und/oder verschlüsselt werden.

**[0185]** Obgleich die vorliegende Erfindung unter Bezugnahme auf bestimmte Beispiele beschrieben worden ist, die nur beispielhaft und nicht die Erfindung begrenzend sein sollen, wird es dem gewöhnlichen Fachmann offenbar sein, daß Änderungen, Zufügungen oder Löschungen an den offenbarten Ausführungsformen durchgeführt werden können, ohne aus dem Sinn und Rahmen der Erfindung zu weichen.

#### Zusammenfassung

**[0186]** Es werden Grafikelemente und Grafikanzeigen zur Verwendung in einer Prozeßumgebung zur Darstellung von Informationen über die Prozeßumgebung wie beispielsweise den aktuellen Zustand von Vorrichtungen in einer Prozeßanlage für einen oder mehrere Benutzer bereitgestellt. Die Grafikelemente und Anzeigen enthalten ein oder mehrere Objekte, von denen jedes eine Visualisierung umfaßt, die eine Instanz in der Prozeßumgebung darstellt, eine der dargestellten Prozeßinstanz zugeordnete Eigenschaft und eine Routine, die in Verbindung mit der Visualisierung und der Eigenschaft fungiert, um eine Änderung der Visualisierung auf Grundlage einer Änderung der Eigenschaft zu veranlassen, oder eine Änderung der Prozeßumgebung auf Grundlage einer Benutzereingabe durch die Visualisierung zu veranlassen. Auf diese Weise können die Grafikobjekte zur Erstellung von Anzeigen benutzt werden, die durch

Animierungen und sonstige visuelle Operationen den sich ändernden Zustand des Prozesses oder erkannte Ereignisse in dem Prozeß grafisch darstellen. Diese gleichen Grafikelemente können dazu benutzt werden, Benutzern im Dialog mit einer Visualisierung auf einem Benutzeroberflächenbildschirm wie beispielsweise durch eine Maus oder eine Tastaturvorrichtung zu ermöglichen, eine Änderung der Objekteigenschaft zu veranlassen, die dann eine Eingabe in eine Vorrichtung in der Prozeßanlage bereitstellen kann.

#### Patentansprüche

1. Anzeigeinstanz zur Verwendung bei einer Darstellung einer optischen Darstellung einer Prozeßinstanz einer Prozeßanlage für einen Benutzer auf einer Anzeigevorrichtung, mit folgendem:  
einem computerlesbaren Speicher; und  
einem im computerlesbaren Speicher gespeicherten Anzeigeobjekt zur Ausführung auf einem Prozessor, wobei das Anzeigeobjekt folgendes enthält:  
einen Eigenschaftsspeicher zur Speicherung eines Wertes einer der Prozeßinstanz zugeordneten Eigenschaft;  
einer Grafikdarstellung der Prozeßinstanz zur Anzeige auf einer Anzeigevorrichtung für einen Benutzer, wenn das Anzeigeobjekt auf einem Prozessor ausgeführt wird; und  
einer Routine, die in Verbindung mit der für den Benutzer dargestellten grafischen Darstellung der Prozeßinstanz arbeitet und dem Wert der Eigenschaft zugeordnet ist.

2. Anzeigeinstanz nach Anspruch 1, wobei die Routine eine Animierungsroutine umfaßt, die die Grafikdarstellung animiert.

3. Anzeigeinstanz nach Anspruch 2, wobei die Animierungsroutine die Grafikdarstellung fortlaufend animiert.

4. Anzeigeinstanz nach Anspruch 2, wobei die Animierungsroutine die Grafikdarstellung durch Anwenden von mindestens einer einer Verschiebung, einer Drehung, einer Umsetzung und einer Neudimensionierung der Grafikdarstellung animiert.

5. Anzeigeinstanz nach Anspruch 2, wobei die Animierungsroutine die Grafikdarstellung durch Ändern einer einer Farbanimierung oder einer Farbgradientenanimierung oder einer Opazitätsanimierung oder einer Schriftarteigenschaftsanimierung oder einer Videoeigenschaft der Grafikdarstellung animiert.

6. Anzeigeinstanz nach Anspruch 1, wobei die grafische Darstellung der Prozeßinstanz zwei oder mehr Grundelemente enthält und wobei die Routine eine Eigenschaft eines der Grundelemente ändert.

7. Anzeigeanstanz nach Anspruch 6, wobei die Eigenschaft eines der Grundelemente eine Fülleigenschaft ist.

8. Anzeigeanstanz nach Anspruch 1, weiterhin mit einem Verweis, der den Eigenschaftswert mit einer Datenquelle in der Prozeßanlage verbindet.

9. Anzeigeanstanz nach Anspruch 8, wobei die Routine eine ablauffähige Routine ist, die den von der Datenquelle in der Prozeßanlage empfangenen Eigenschaftswert umformt.

10. Anzeigeanstanz nach Anspruch 9, wobei die ablauffähige Routine den Eigenschaftswert in eine Farbe oder in einen einer aufgezählten Liste von Werten oder in einer Länge, oder in einen Schriftartnamen oder in eine lokalisierte Kette oder in eine Dauer oder in eine Drehung umformt.

11. Anzeigeanstanz nach Anspruch 1, wobei die Routine eine ablauffähige Routine ist, die einen der Prozeßinstanz zugeordneten Zustand erkennt und den erkannten Zustand mit der Änderung der grafischen Darstellung der Prozeßinstanz anzeigt.

12. Anzeigeanstanz nach Anspruch 11, wobei der erkannte Zustand auf einen Kommunikationszustand oder einen Vorrichtungszustand oder einen Wertzustand bezogen ist.

13. Anzeigeanstanz nach Anspruch 1, wobei die Routine eine ablauffähige Routine ist, die eine Eingabe von einem Benutzer über die grafische Visualisierung annimmt.

14. Anzeigeanstanz nach Anspruch 13, wobei die ablauffähige Routine die Eingabe benutzt, um zu veranlassen, daß eine Änderung des Eigenschaftswerts eine Laufzeitumgebung außerhalb der Anzeigeanstanz bewirkt.

15. Grafikanzeigeeditor zur Verwendung in einer Prozeßanlage zur Erstellung einer Grafikanzeige, die die Funktionsweise einer oder mehrerer Instanzen in der Prozeßanlage darstellt, wobei der Grafikanzeigeeditor folgendes umfaßt:

eine Bibliothek von Grafikobjekten, wobei jedes Grafikobjekt eine optische Darstellung einer physikalischen oder logischen Instanz in der Prozeßanlage umfaßt;

eine grafisch basierende Editorentwurfsflächenroutine, die einem Benutzer die Definition einer ablauffähigen Grafikanzeige durch Planieren einer oder mehrerer optischen Darstellungen der Grafikobjekte aus der Bibliothek von Grafikobjekten auf eine Editorentwurfsfläche ermöglicht, um eine Art und Weise zu definieren, auf die die eine oder mehreren optischen Darstellungen der Grafikobjekte auf einer Anzeigevorrichtung für einen Benutzer während der Ausführung

der Grafikanzeige dargestellt werden; eine Eigenschaftsdefinitionsentwurfsflächenroutine, um einem Benutzer die Definition einer mindestens einem der Mehrzahl von Grafikobjekten zugeordneten Eigenschaft zu ermöglichen; eine Anbindungsdefinitionsroutine, um einem Benutzer die Angabe einer Anbindung zwischen der Eigenschaft und einer Laufzeitumgebung in der Prozeßanlage zu ermöglichen; und eine Definitionsroutine, um einem Benutzer die Definition einer Routine zu ermöglichen, die in Verbindung mit der optischen Darstellung eines der Grafikobjekte und/oder der Eigenschaft während der Ausführung der Grafikanzeige arbeitet.

16. Grafikanzeigeeditor nach Anspruch 15, wobei die Definitionsroutine einem Benutzer die Definition einer Routine als Animierungsroutine ermöglicht, die die optische Darstellung des einen der Grafikobjekte animiert.

17. Grafikanzeigeeditor nach Anspruch 16, wobei die Definitionsroutine einem Benutzer die Definition einer Routine als Animierungsroutine ermöglicht, die die optische Darstellung eines der Grafikobjekte fortlaufend animiert.

18. Grafikanzeigeeditor nach Anspruch 16, wobei die Definitionsroutine einem Benutzer die Definition einer Routine als Animierungsroutine ermöglicht, die die optische Darstellung eines der Grafikobjekte durch Anwenden mindestens einer einer Verschiebung, einer Drehung, einer Umsetzung und einer Neudimensionierung der optischen Darstellung des einen der Grafikobjekte animiert.

19. Grafikanzeigeeditor nach Anspruch 16, wobei die Definitionsroutine einem Benutzer die Definition der Routine als Animierungsroutine ermöglicht, die die optische Darstellung des einen der Grafikobjekte durch Ändern einer der optischen Darstellung eines der Grafikobjekte zugeordneten Farbe animiert.

20. Grafikanzeigeeditor nach Anspruch 15, wobei die Definitionsroutine einem Benutzer die Definition der Routine als ablauffähige Transformationsroutine ermöglicht, die einen für die Eigenschaft von der Datenquelle in der Prozeßanlage empfangenen Eigenschaftswert umformt.

21. Grafikanzeigeeditor nach Anspruch 20, wobei die Definitionsroutine einem Benutzer die Definition der Routine als ablauffähigen Transformationsroutine als eine Transformation ermöglicht, die den Eigenschaftswert in eine Farbe oder in einen einer aufgezählten Liste von Werten oder in eine Länge oder in einen Schriftartnamen oder in eine lokalisierte Kette oder in eine Dauer oder in eine Drehung umformt.

22. Grafikanzeigeditor nach Anspruch 15, wobei die Definitionsroutine einem Benutzer die Definition der Routine als ablauffähige Routine ermöglicht, die einen der Prozeßinstanz zugeordneten Zustand erkennt und die den erkannten Zustand mit der Änderung der optischen Darstellung eines der Grafikobjekte anzeigt.

23. Grafikanzeigeditor nach Anspruch 15, wobei die Definitionsroutine einem Benutzer die Definition der Routine als Routine ermöglicht, die die optische Darstellung des einen der Grafikobjekte unter Verwendung unterschiedlicher, auf Grundlage eines Wertes der Eigenschaft ausgewählten Animierungen ändert.

24. Grafikanzeigeditor nach Anspruch 15, wobei die Definitionsroutine einem Benutzer die Definition der Routine als eine Routine ermöglicht, die eine Eingabe von einem Benutzer über die grafische Visualisierung annimmt.

25. Grafikanzeigeditor nach Anspruch 24, wobei die Definitionsroutine einem Benutzer die Definition der Routine als ablauffähige Routine ermöglicht, die eine Eingabe benutzt, um zu veranlassen, daß eine Änderung der Eigenschaft eine Variable in einer Laufzeitumgebung außerhalb der Anzeigeinstanz bewirkt.

26. Grafikanzeige zur Verwendung in einer Prozeßanlage zur Darstellung einer oder mehrerer Instanzen in der Prozeßanlage, mit folgendem:  
einer Mehrzahl von zusammenschalteten Grafikobjekten zum Bilden einer optischen Darstellung mindestens eines Teils der Prozeßanlage;  
einem Eigenschaftsspeicher zum Speichern eines Wertes für eine mindestens einem der Mehrzahl von Grafikobjekten zugeordneten Eigenschaft;  
einer Anbindung, die den Eigenschaftsspeicher an eine Laufzeitumgebung in der Prozeßanlage anbindet, um der Eigenschaft zugeordnete Daten zum Bestimmen des Werts der Eigenschaft zu empfangen; und  
einer Routine, die in Verbindung bezüglich zur optischen Darstellung von mindestens einem der Mehrzahl von Grafikobjekten und/oder dem Wert der Eigenschaft arbeitet.

27. Grafikanzeige nach Anspruch 26, wobei die Routine eine Animierungsroutine umfaßt, die die optische Darstellung eines der Mehrzahl von Grafikobjekten auf Grundlage des Werts der Eigenschaft animiert.

28. Grafikanzeige nach Anspruch 27, wobei die Animierungsroutine die optische Darstellung fortlaufend animiert.

29. Grafikanzeige nach Anspruch 27, wobei die

Animierungsroutine die optische Darstellung durch Anwenden mindestens einer einer Verschiebung, einer Drehung, einer Umsetzung und einer Neudimensionierung der optischen Darstellung animiert.

30. Grafikanzeige nach Anspruch 27, wobei die Animierungsroutine die optische Darstellung durch Ändern einer Farbe der optischen Darstellung animiert.

31. Grafikanzeige nach Anspruch 26, wobei die optische Darstellung des einen der Mehrzahl von Grafikobjekten ein oder mehrere Grundelemente enthält und wobei die Routine eine Eigenschaft eines der Grundelemente ändert.

32. Grafikanzeige nach Anspruch 26, wobei die Routine eine ablauffähige Routine ist, die eine Transformation an den Eigenschaftswert anlegt.

33. Grafikanzeige nach Anspruch 32, wobei die ablauffähige Routine den Eigenschaftswert in eine Farbe oder in einen einer aufgezählten Liste von Werten umformt.

34. Grafikanzeige nach Anspruch 26, wobei die Routine eine ablauffähige Routine ist, die einen einer Prozeßinstanz auf Grundlage des Werts der Eigenschaft zugeordneten Zustand erkennt und den erkannten Zustand mit einer Änderung der optischen Darstellung eines der Mehrzahl von Grafikobjekten anzeigt.

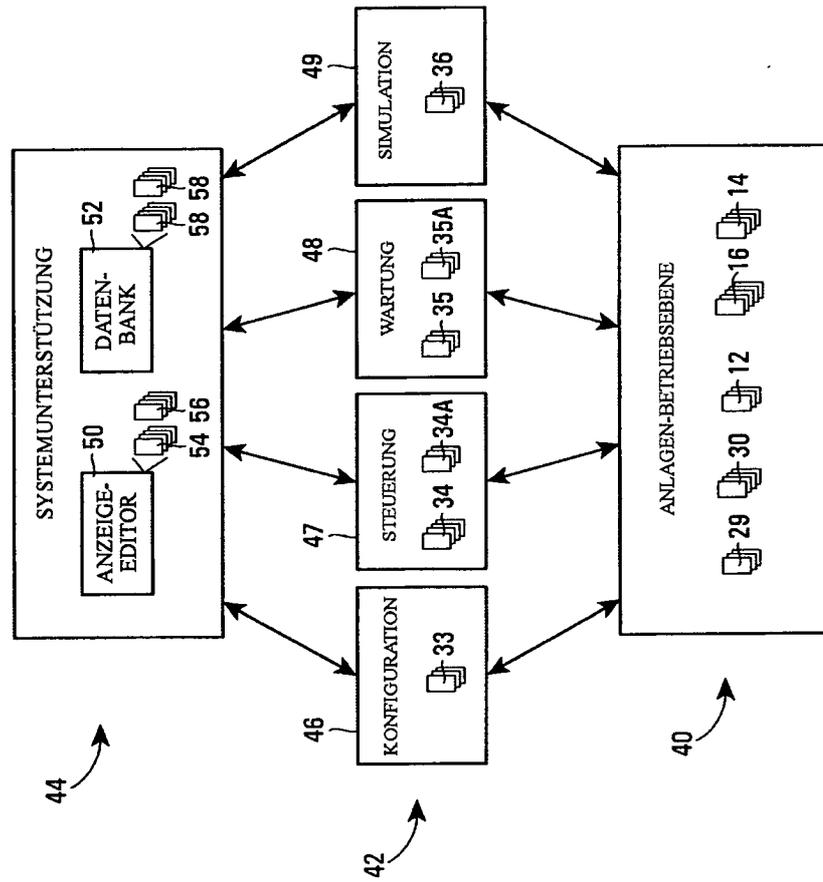
35. Grafikanzeige nach Anspruch 26, wobei die Routine die optische Darstellung eines der Mehrzahl von Grafikobjekten unter Verwendung unterschiedlicher Animierungen ändert, die auf Grundlage des Eigenschaftswerts ausgewählt werden.

36. Grafikanzeige nach Anspruch 26, wobei die Routine eine ablauffähige Routine ist, die eine Eingabe von einem Benutzer über die optische Darstellung des einen der Mehrzahl von Grafikobjekten annimmt.

37. Grafikanzeigeinstanz nach Anspruch 36, wobei die ablauffähige Routine die Eingabe benutzt, um einer Änderung einer Laufzeitumgebung außerhalb der Anzeigeinstanz zu veranlassen.

Es folgen 27 Blatt Zeichnungen





**FIG. 2**

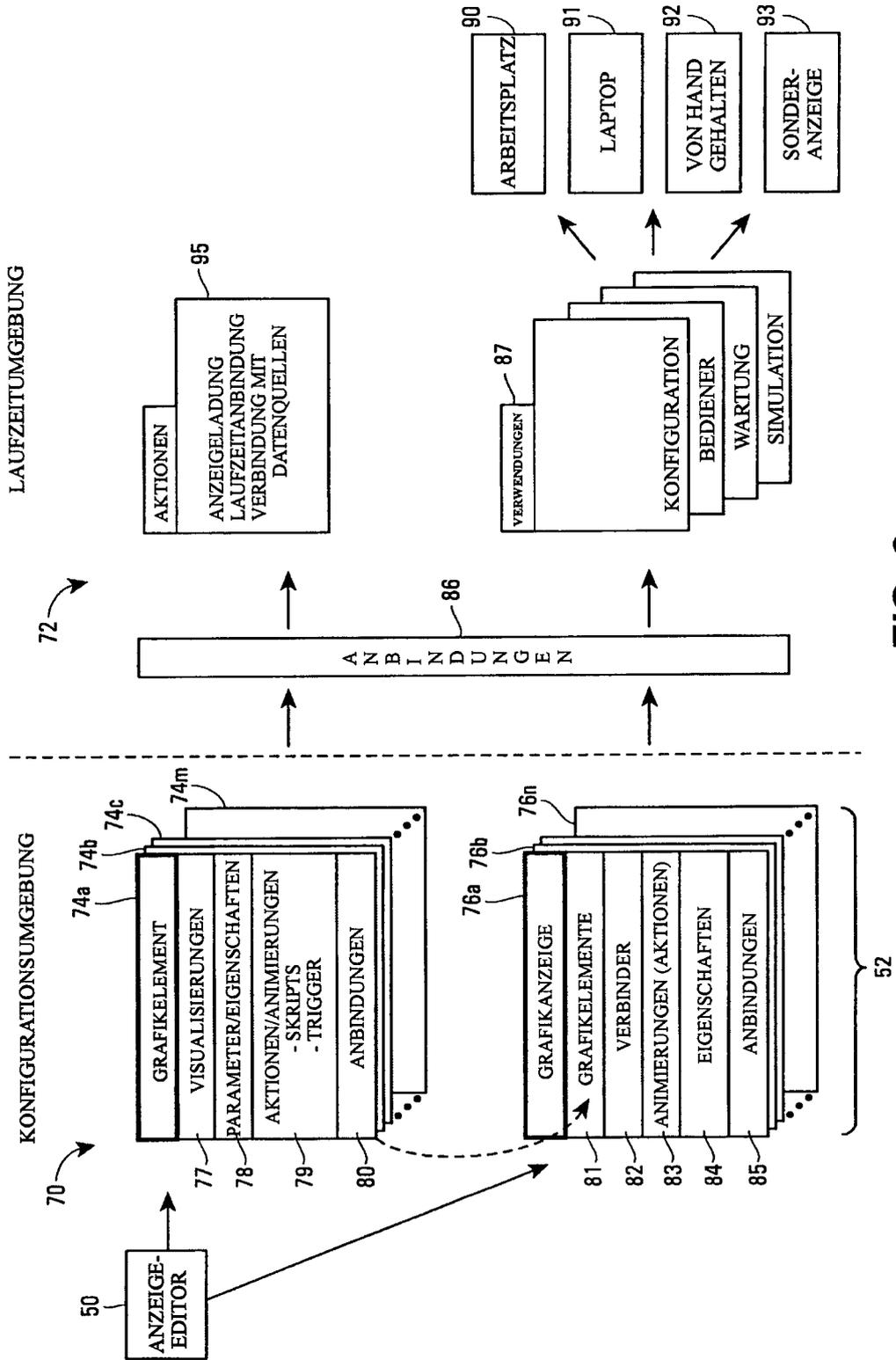
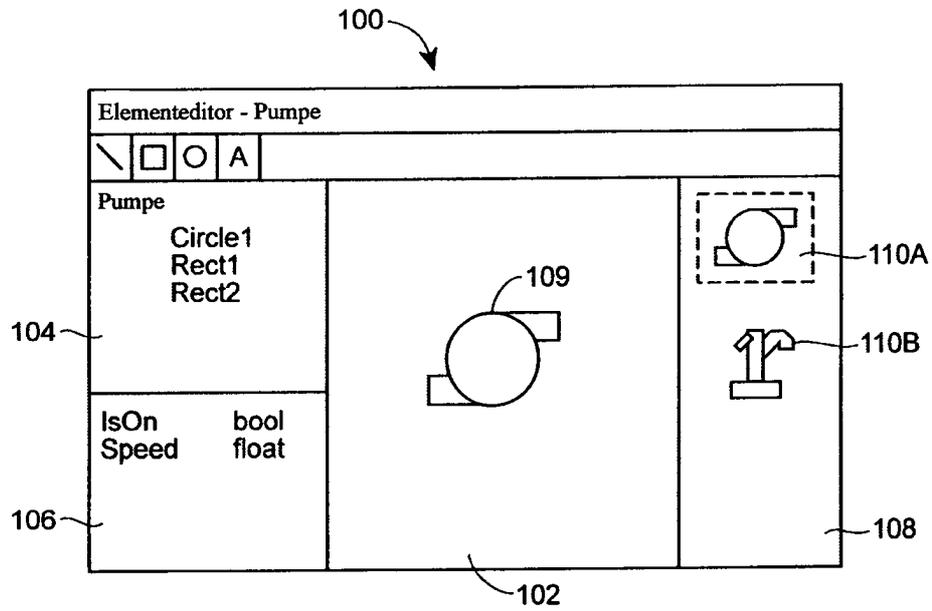
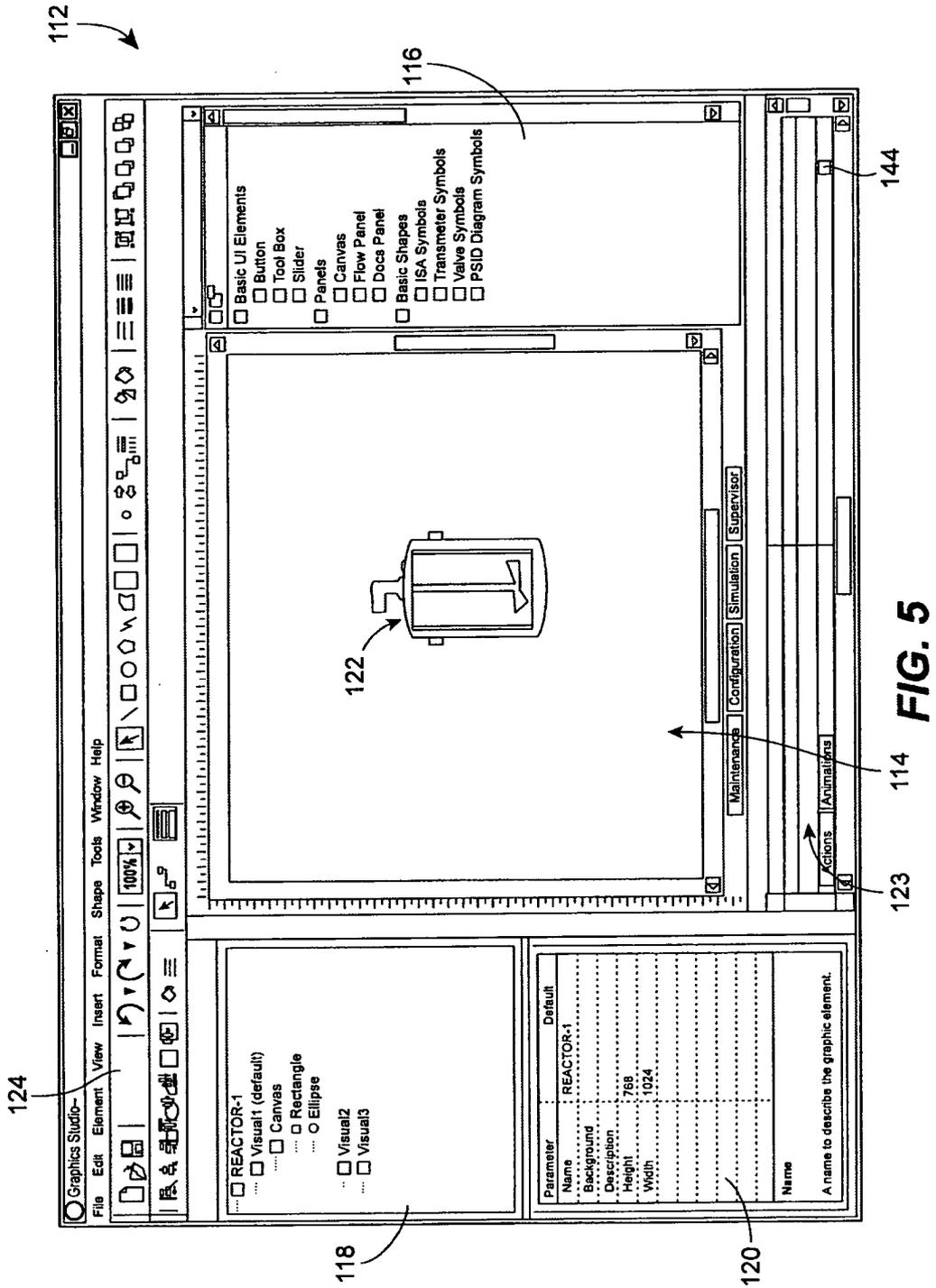


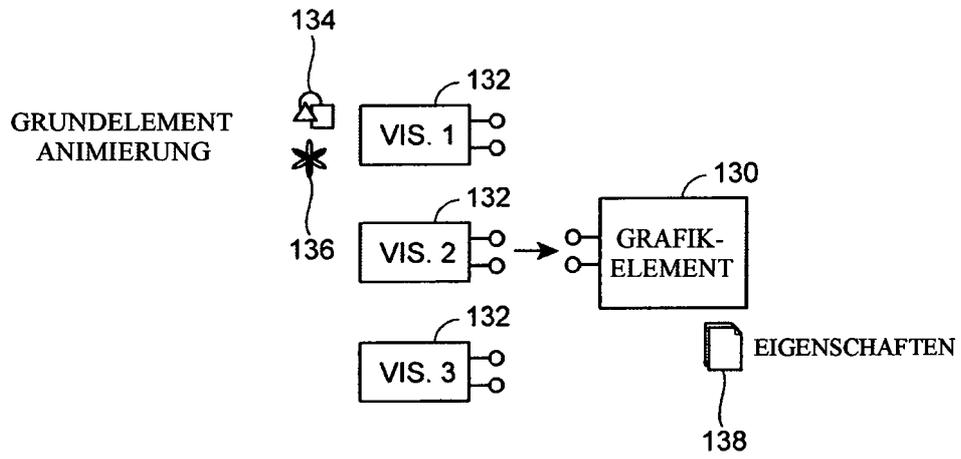
FIG. 3



**FIG. 4**



Parameter	Default
Name	REACTOR-1
Background	
Description	
Height	768
Width	1024
Name	
A name to describe the graphic element.	



**FIG. 6**

120A

Eigenschaften		
Name	Datenart	Vorgabewert
IsOn	Bool	True
Speed	String	Fast
<b>IsOn</b> Zeigt an, ob die Pumpe an oder aus ist		

FIG. 7

120B

Name	Wert	Animierung	Anbindung
Name	1		
Hintergrund		1	<input type="checkbox"/> Spe... <input type="checkbox"/>
Beschreibung			
Höhe	768		
Breite	1024		
<b>IsOn</b> Zeigt an, ob die Pumpe an oder aus ist			

140

142

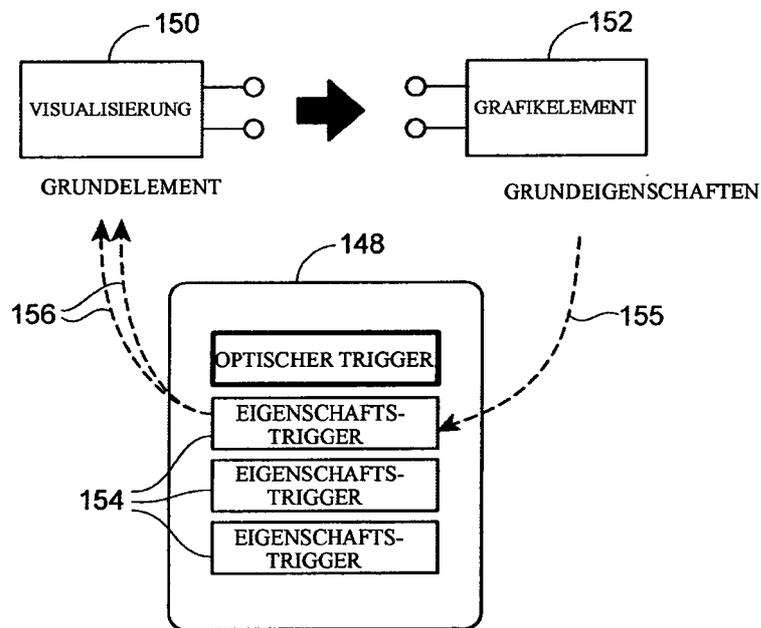
FIG. 8

123A

Element Property	Value	Target	Path	Value
IsOn	True	Visual1/Rectangle1	Fill	Red
IsOn	True	Visual1/Ellipse1	Visible	False
Actions		Animations		Show All

144

**FIG. 9**



**FIG. 10**

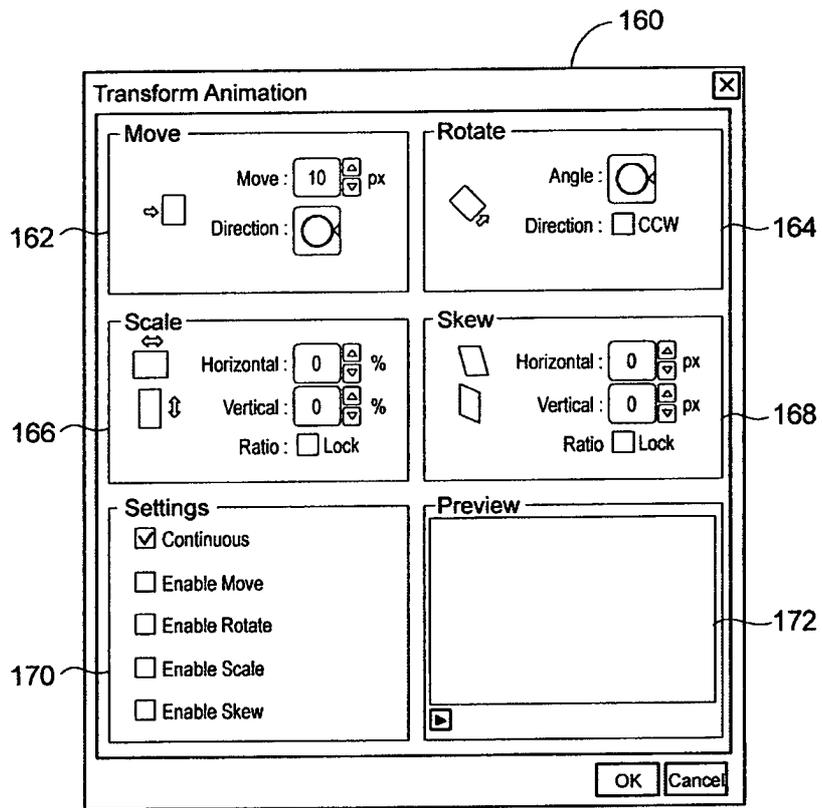


FIG. 11

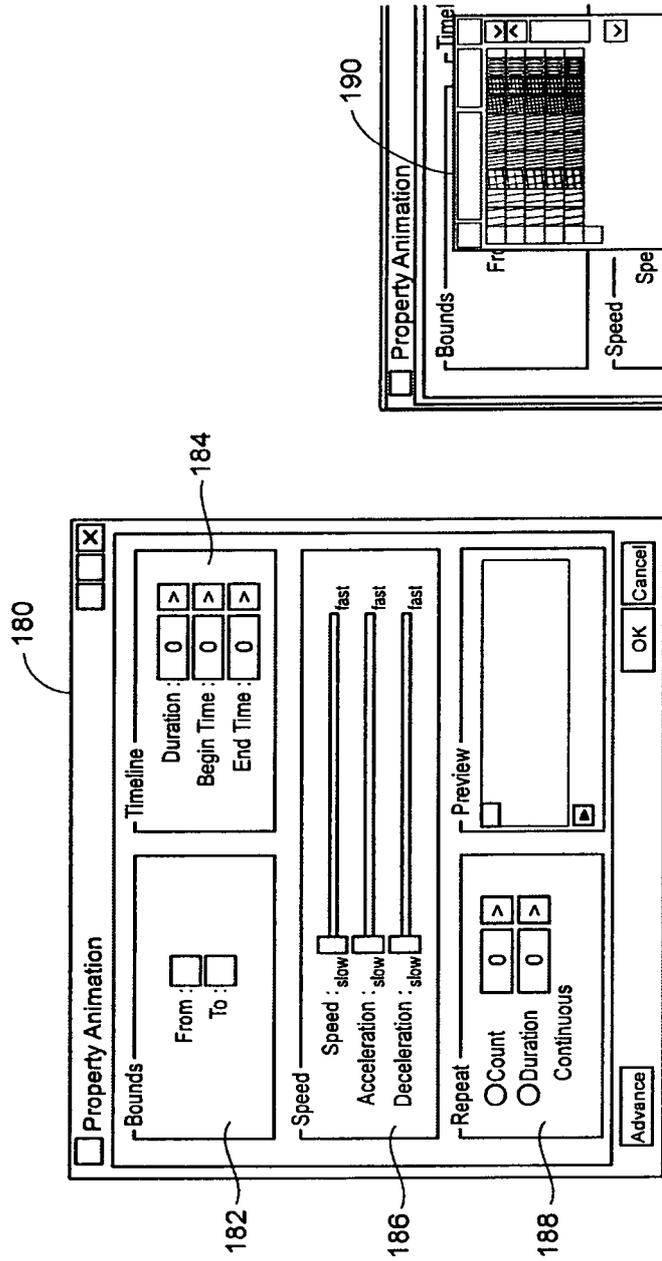


FIG. 12

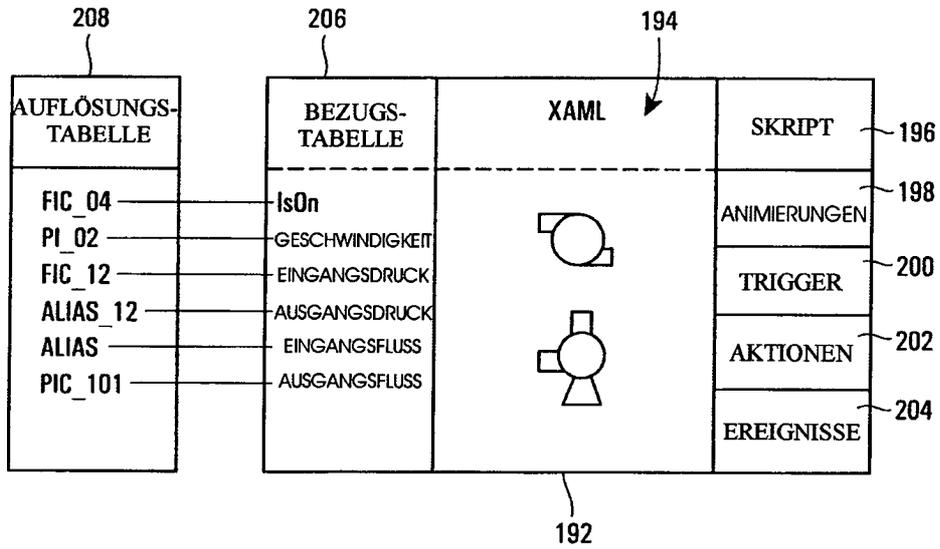


FIG. 13

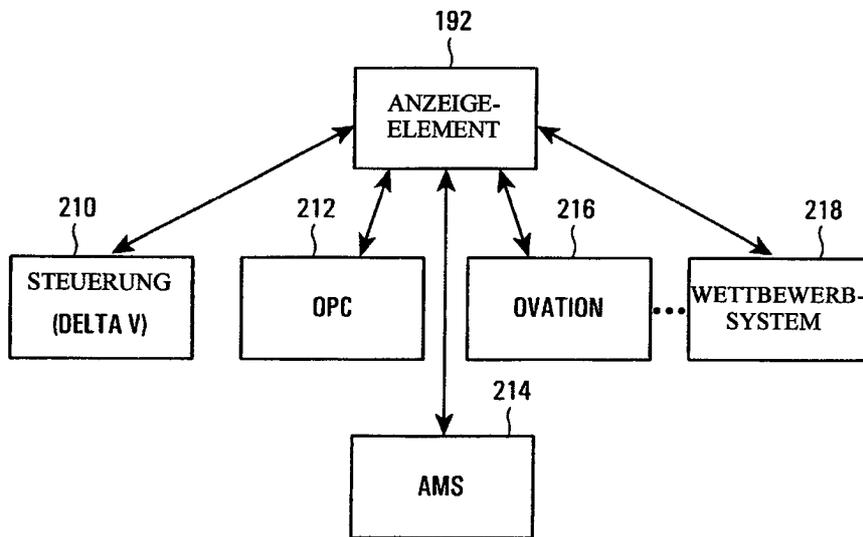


FIG. 14

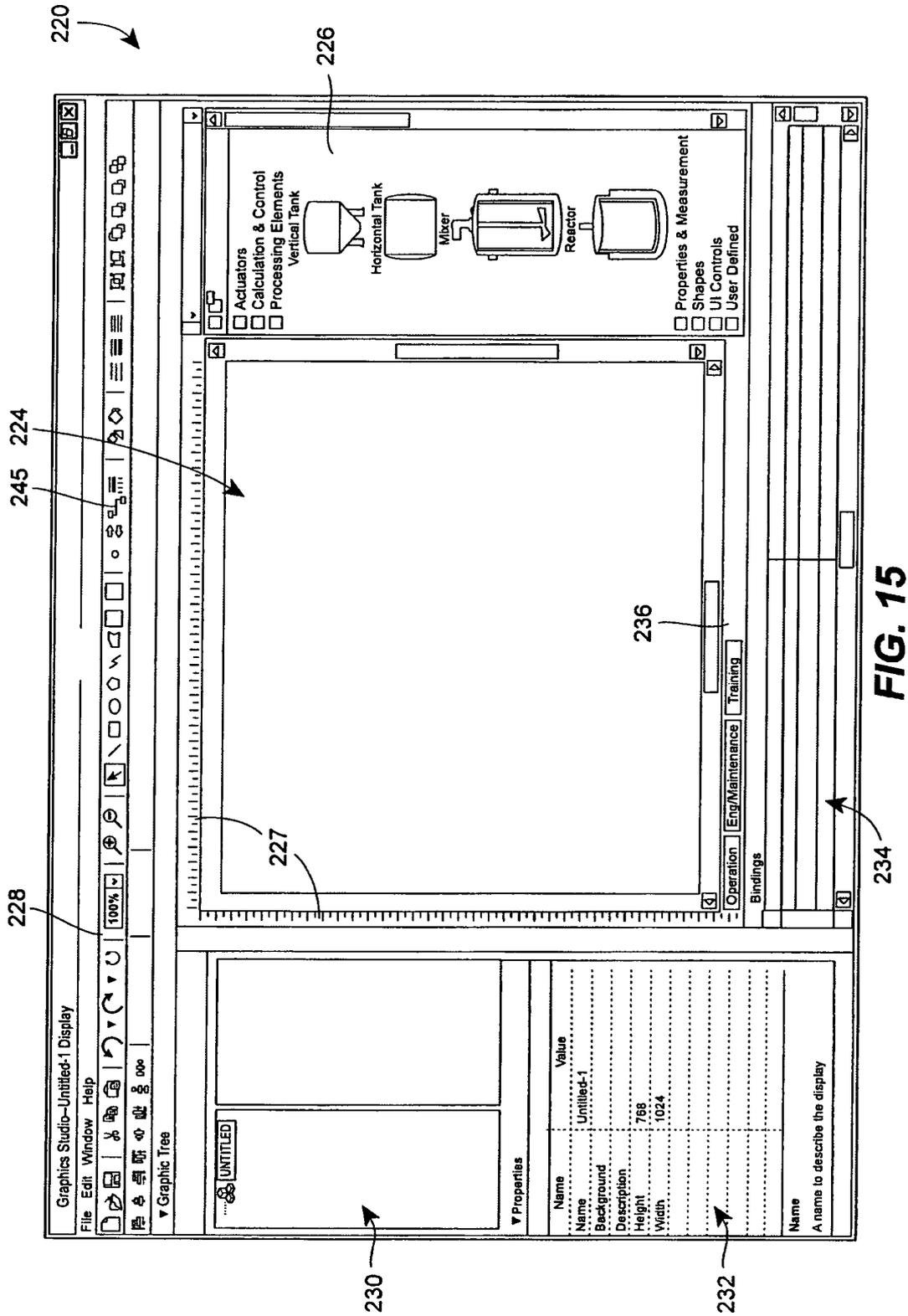


FIG. 15

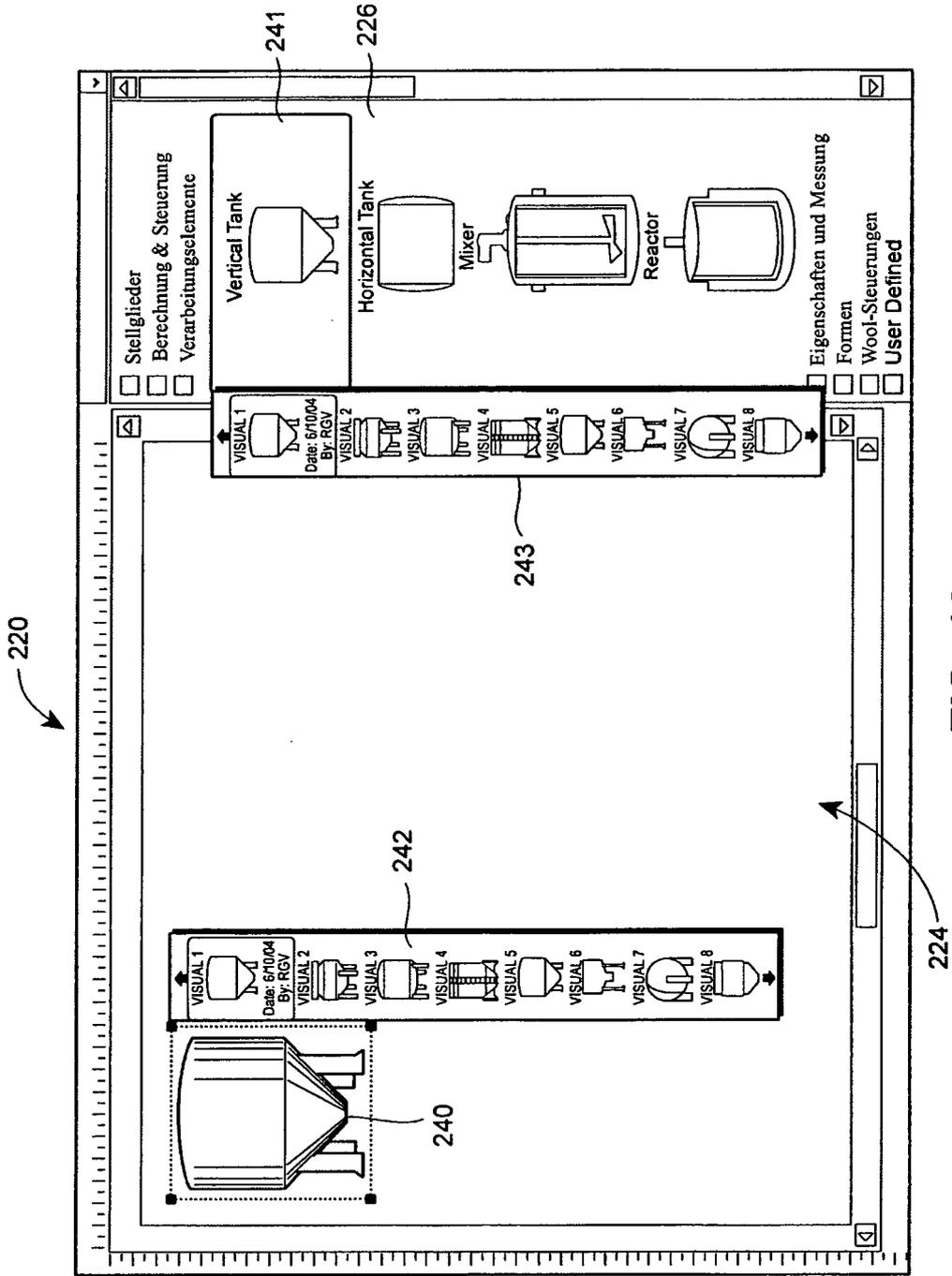


FIG. 16

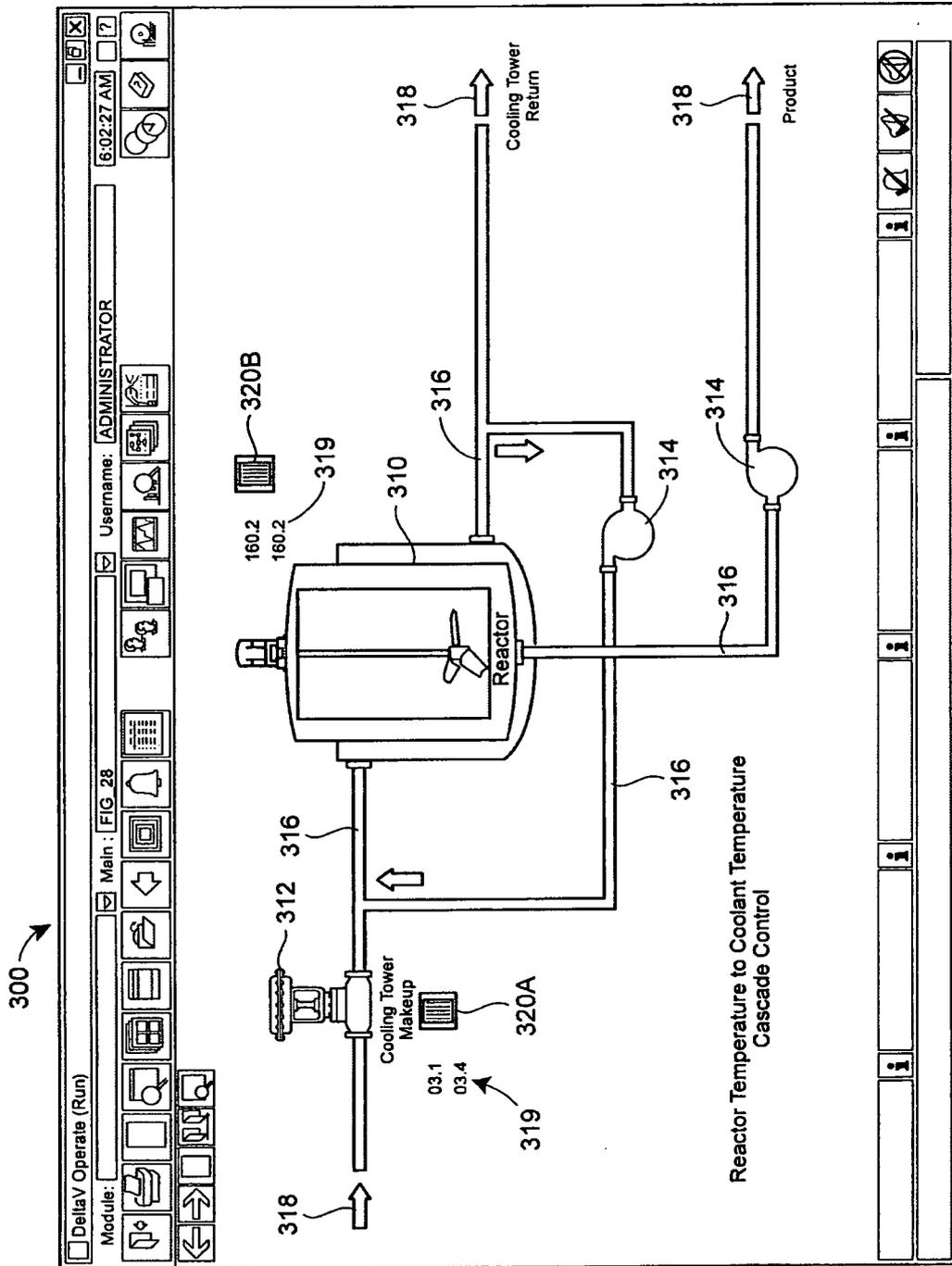


FIG. 17

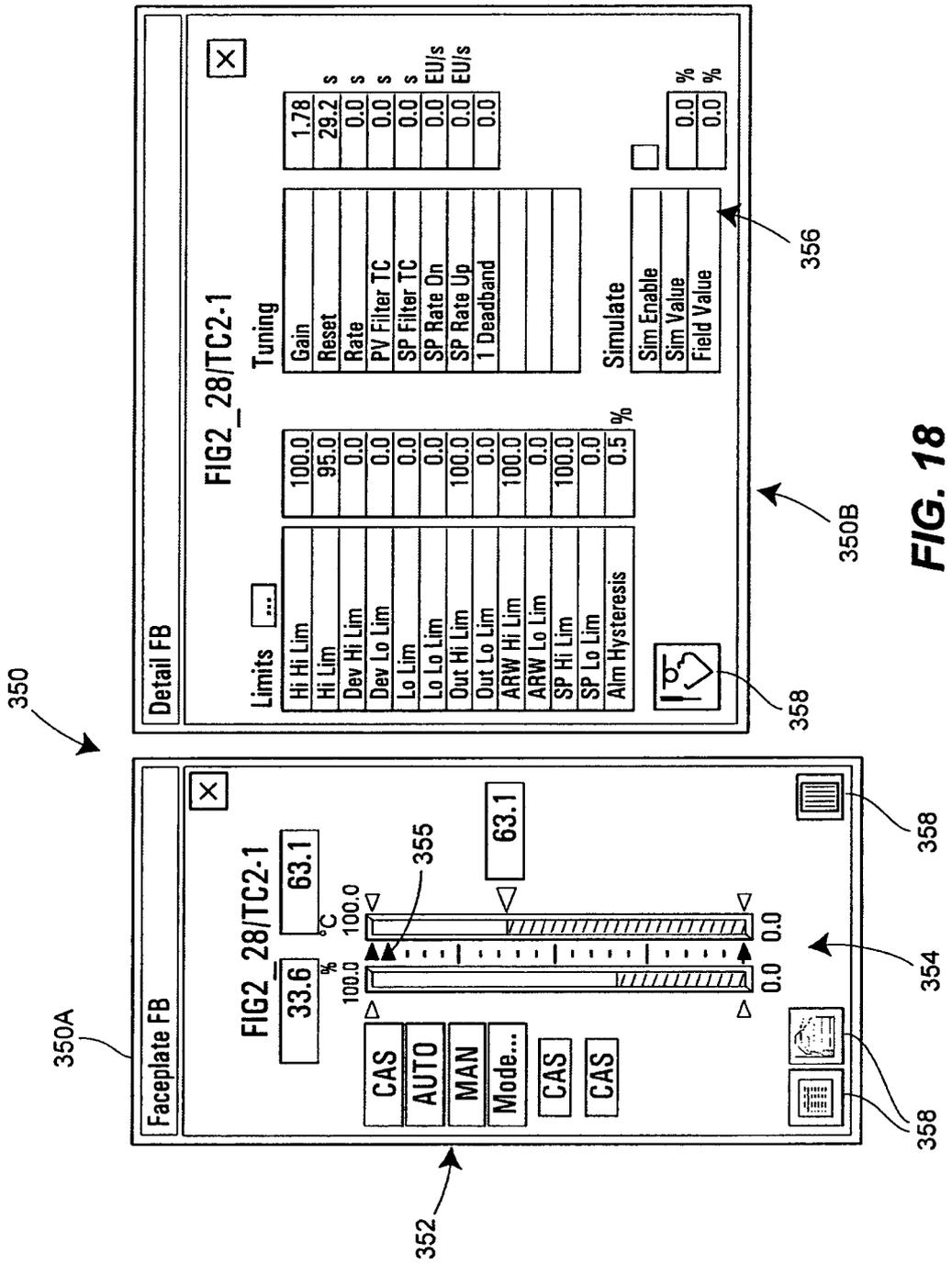


FIG. 18

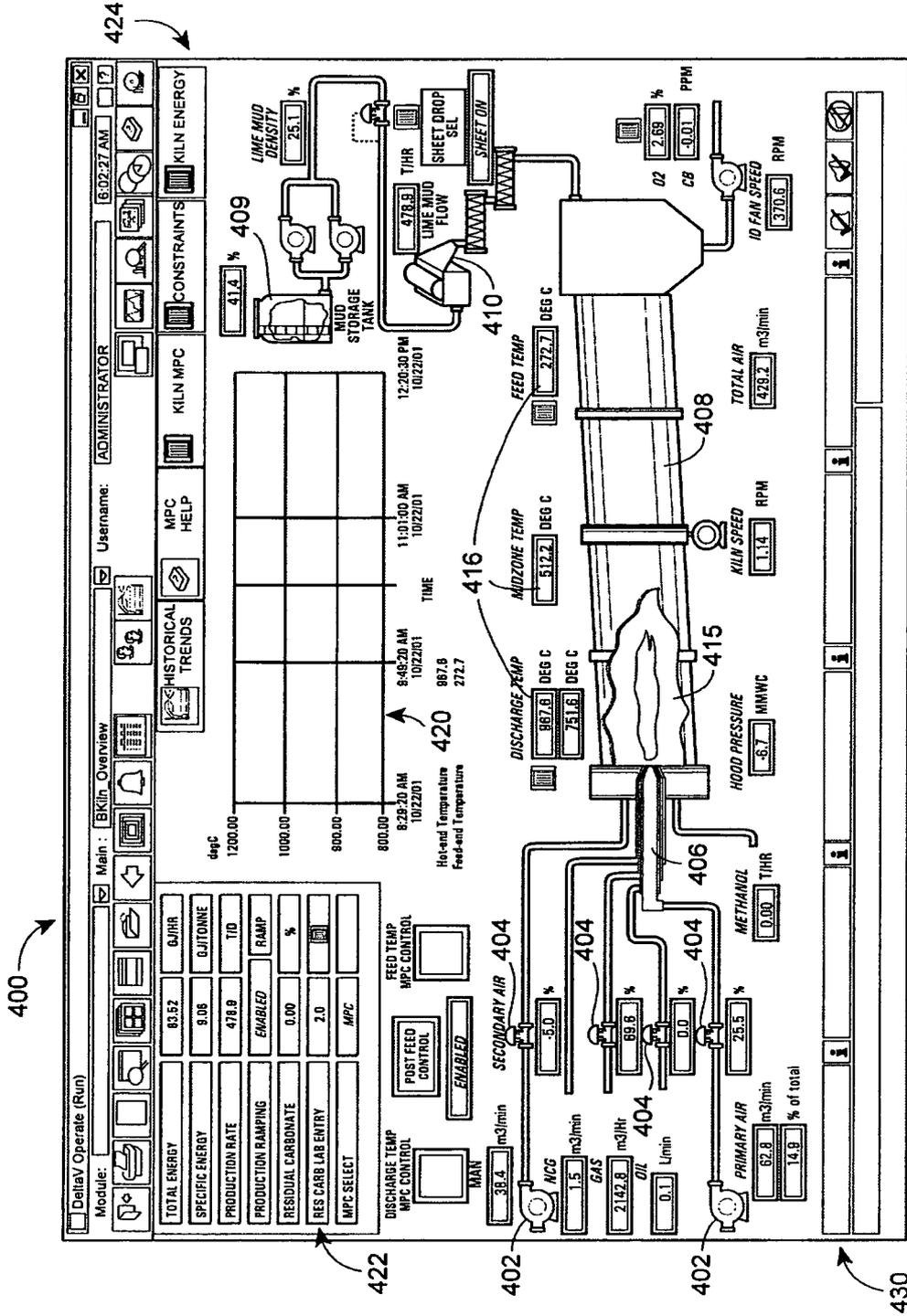


FIG. 19



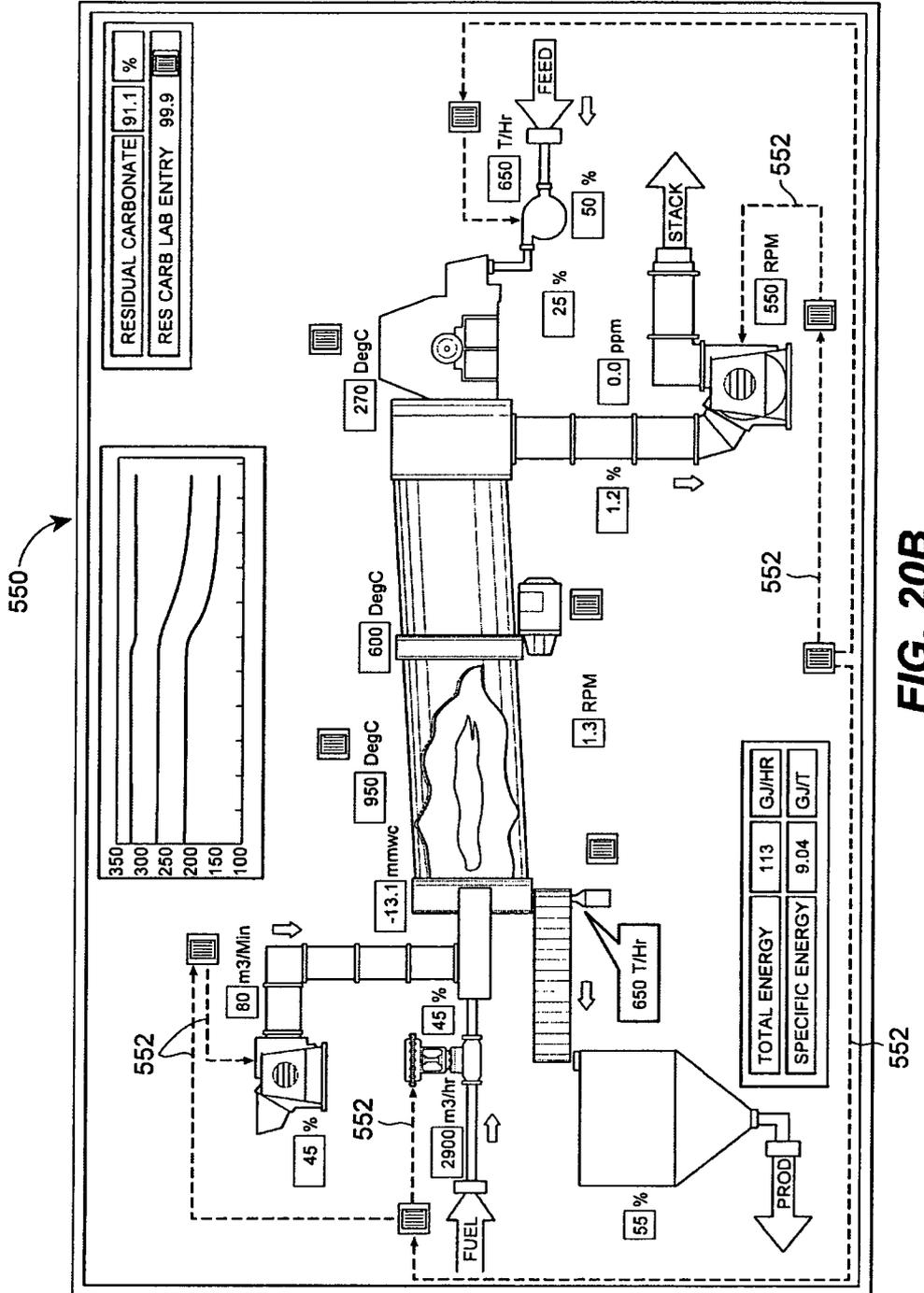


FIG. 20B

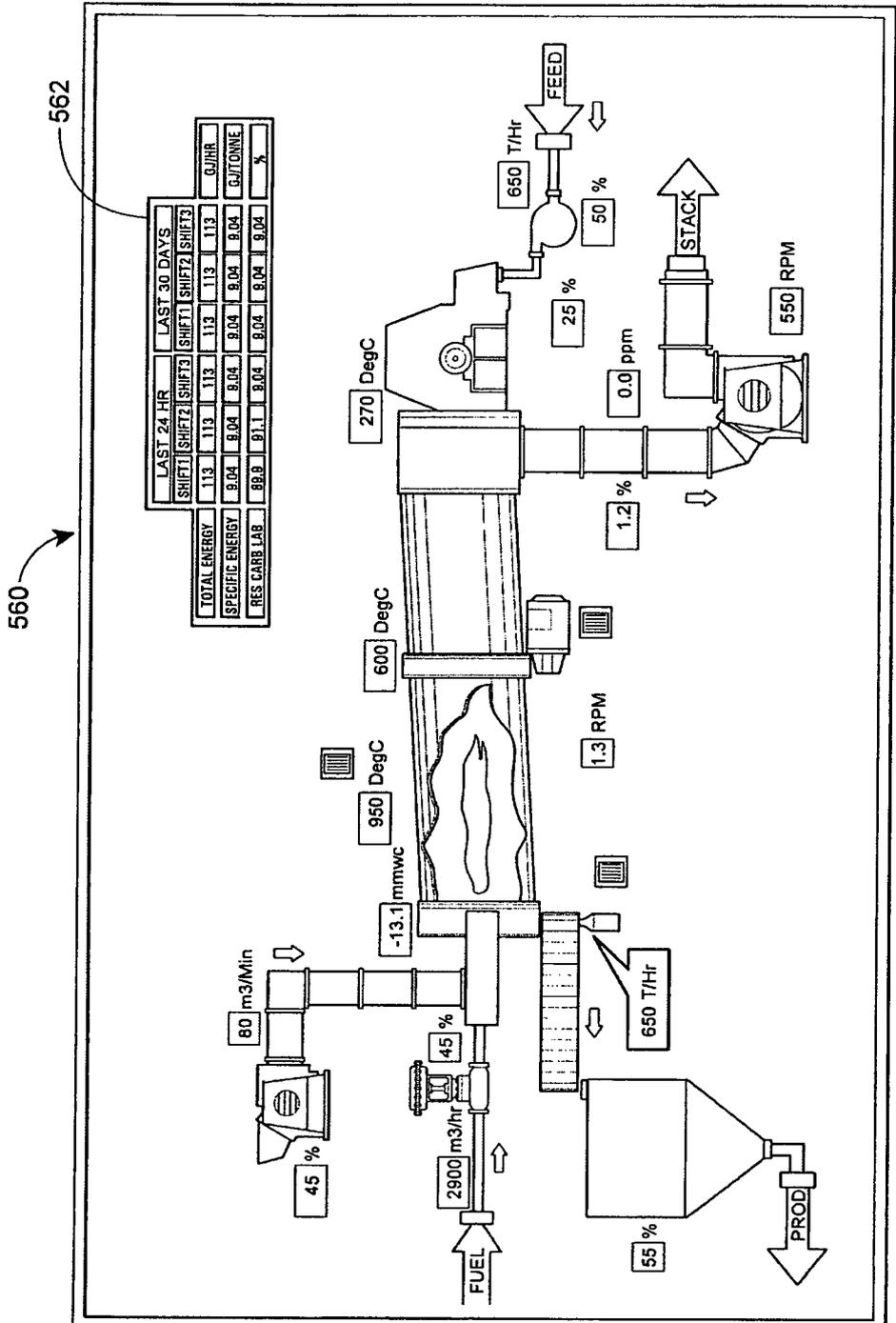


FIG. 20C

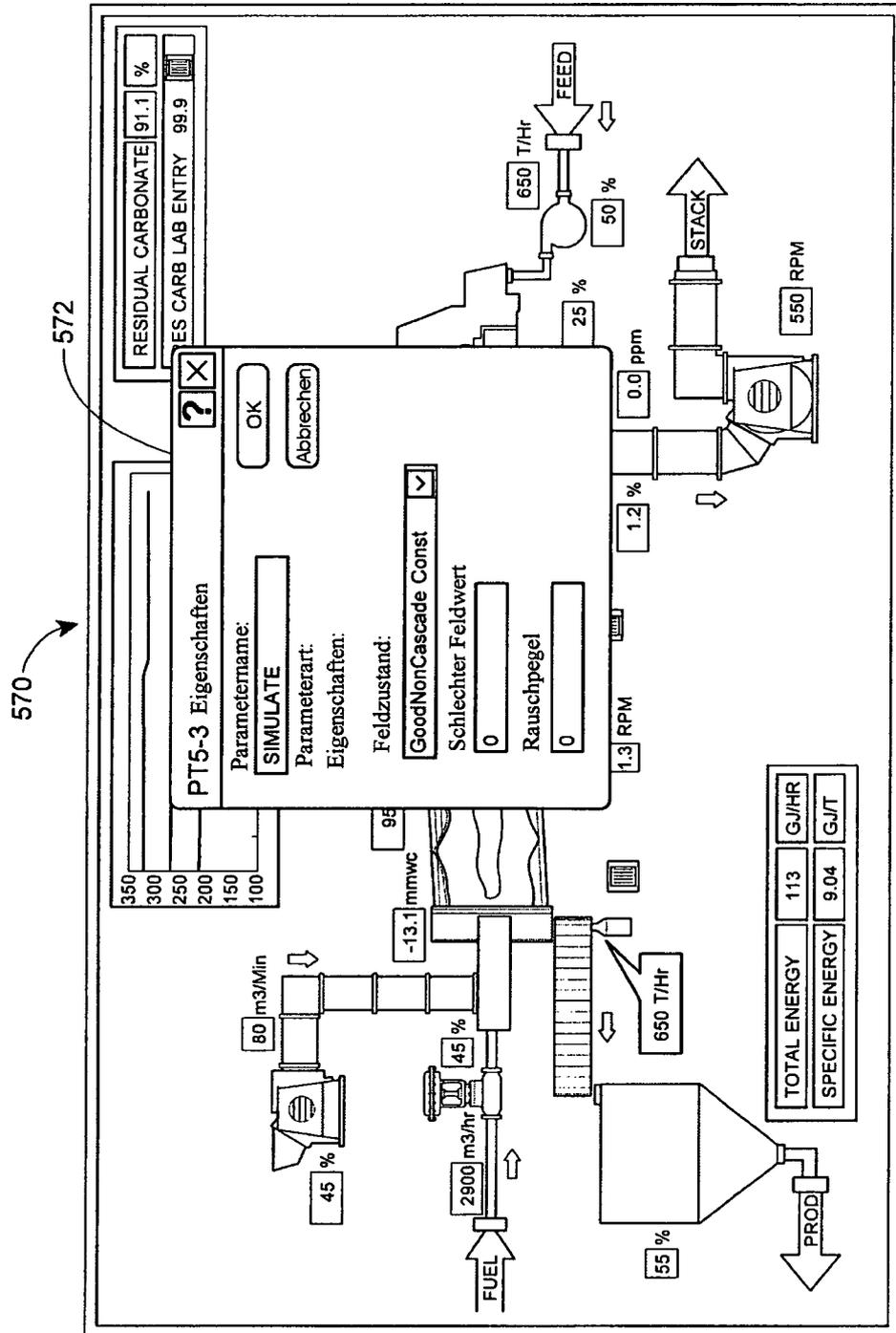


FIG. 20D

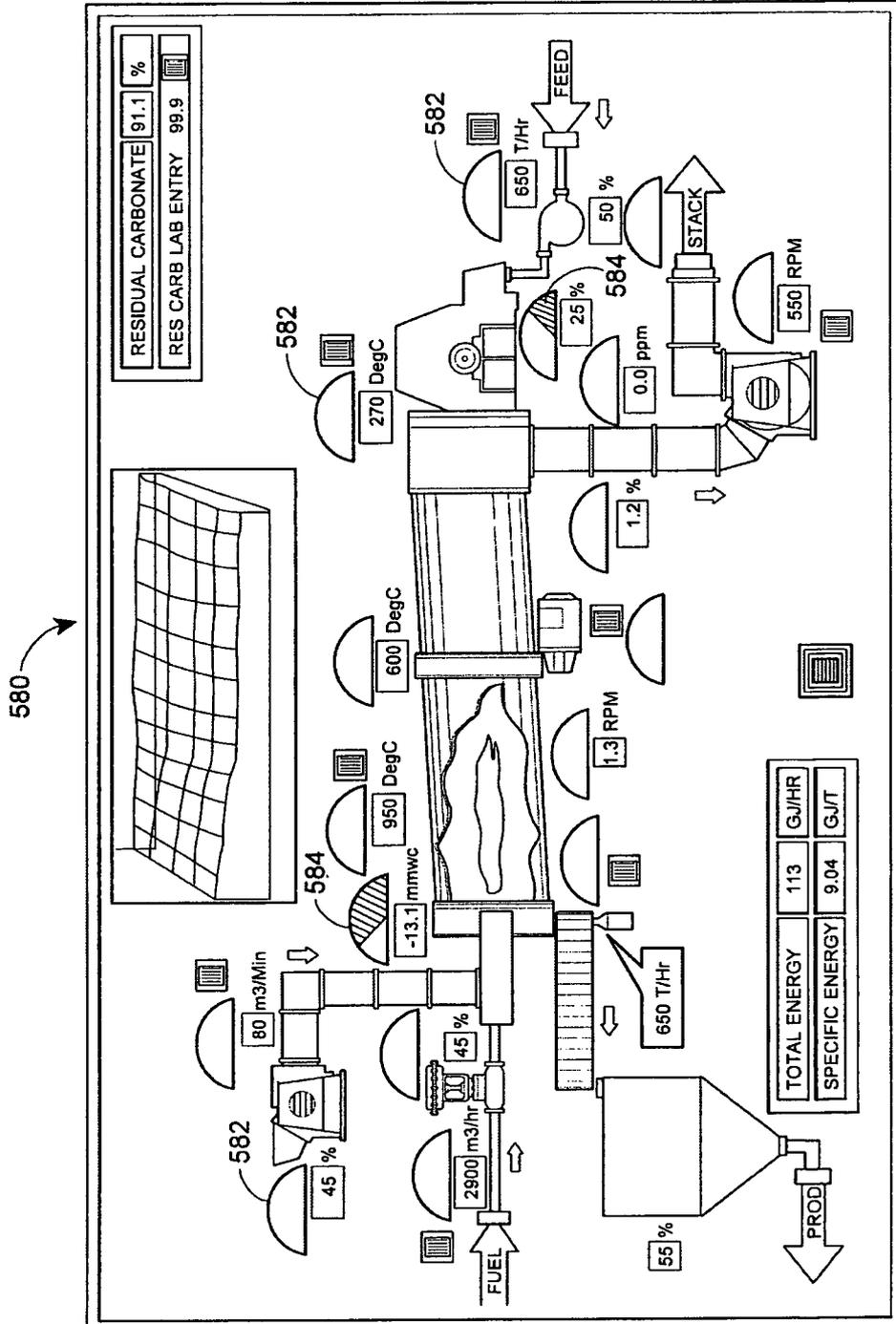


FIG. 20E

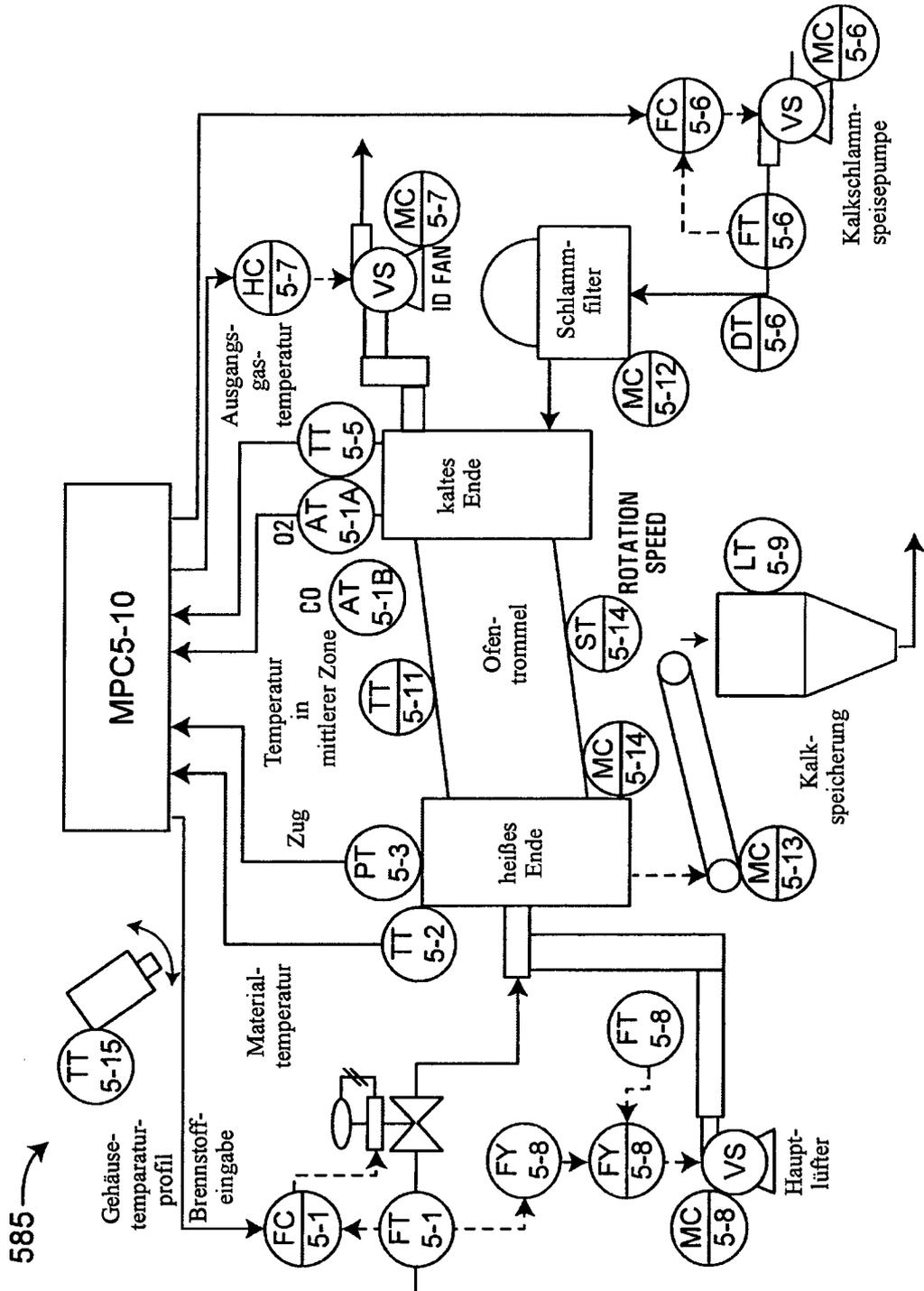


FIG. 21A



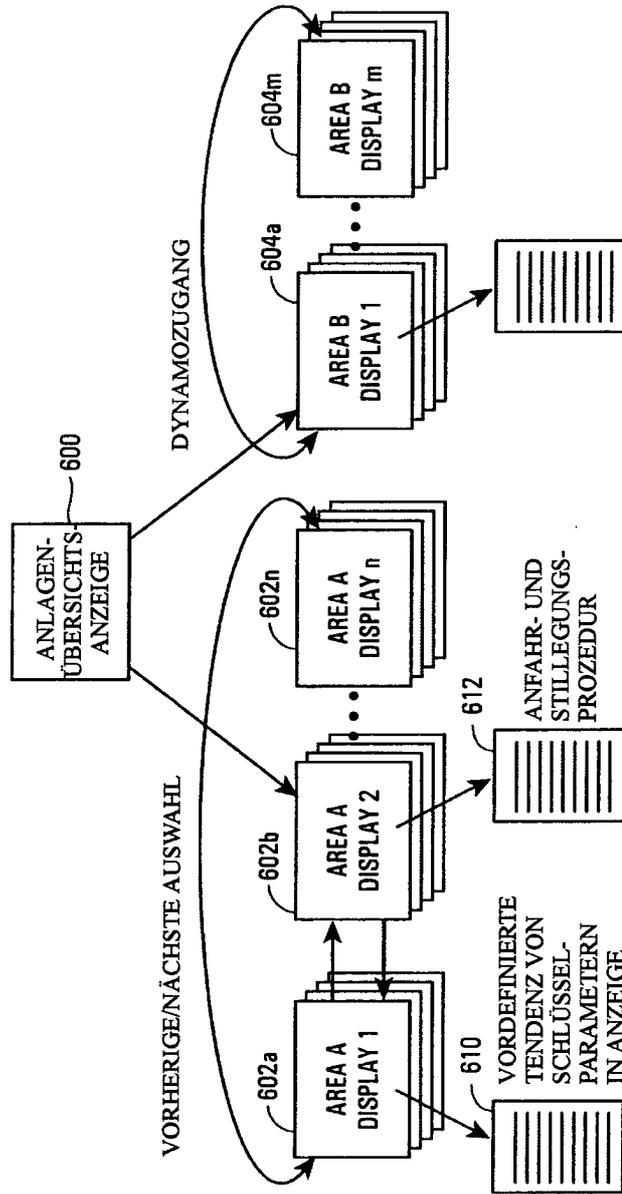


FIG. 22

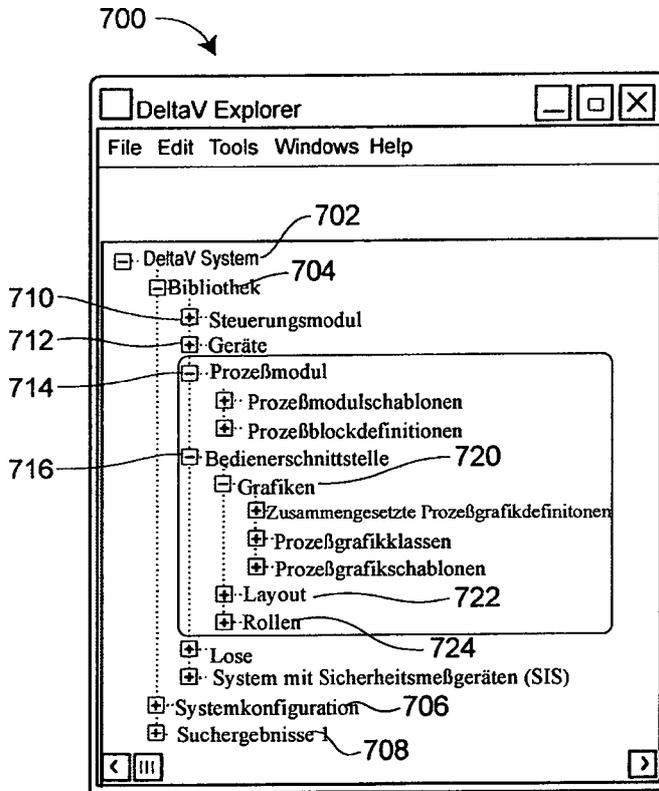


FIG. 23

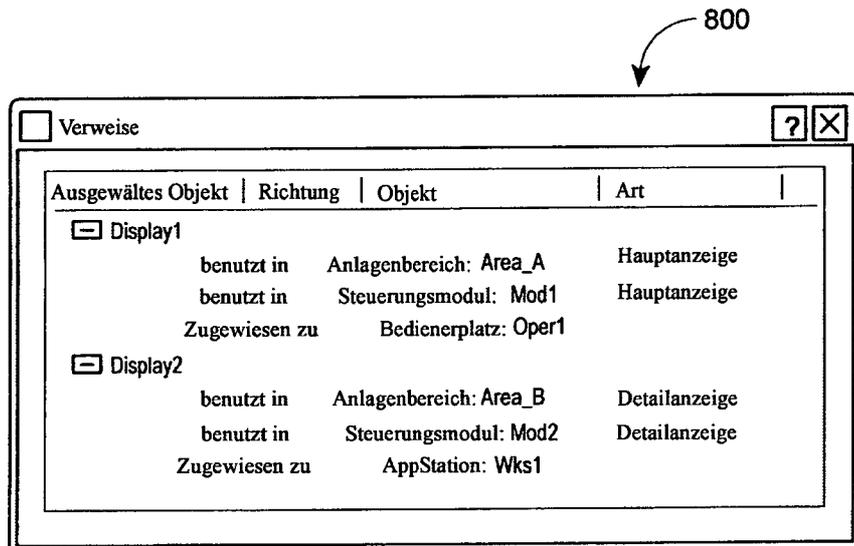


FIG. 25

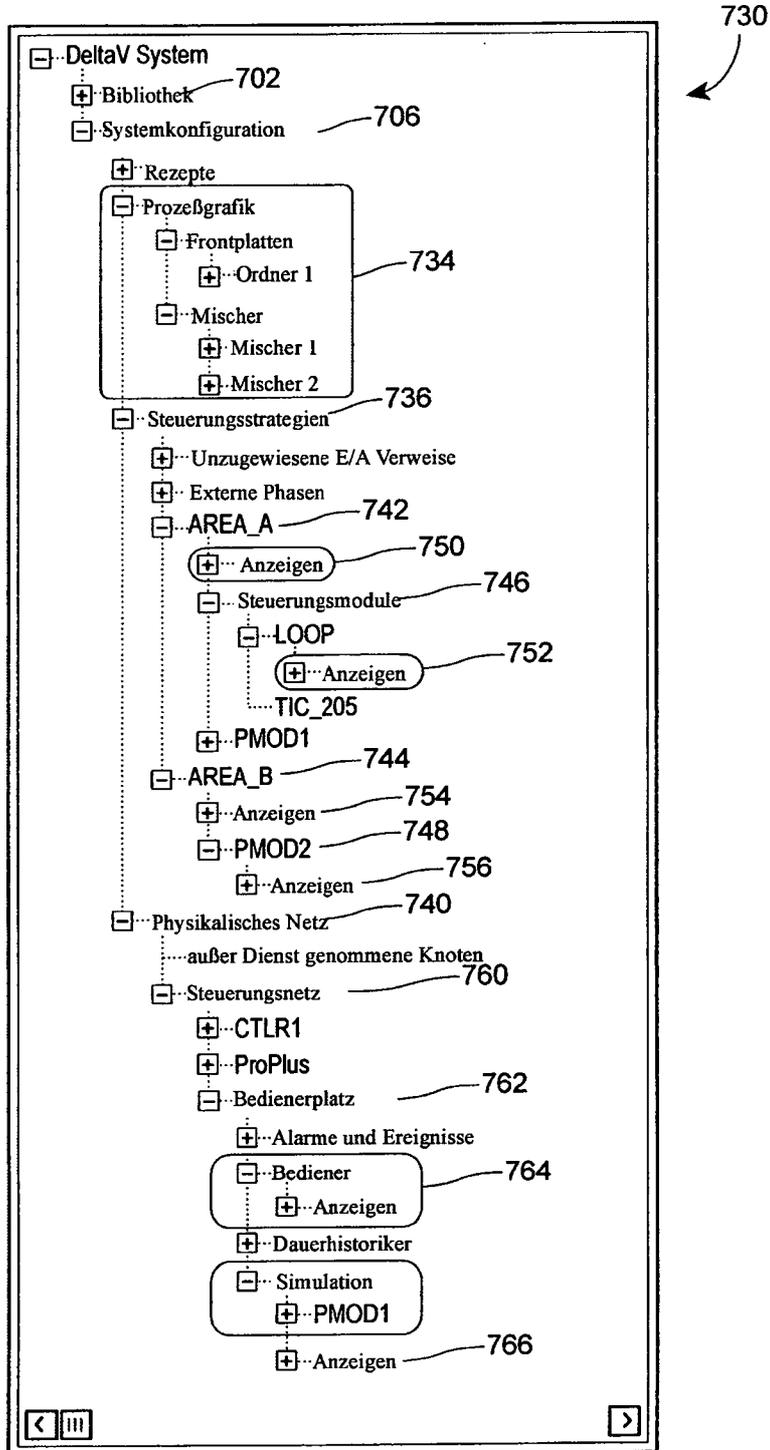


FIG. 24