



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

(10) **DE 11 2005 001 030 T5 2007.05.24**

(12)

## Veröffentlichung

der internationalen Anmeldung mit der  
 (87) Veröffentlichungs-Nr.: **WO 2005/109131**  
 in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)  
 (21) Deutsches Aktenzeichen: **11 2005 001 030.9**  
 (86) PCT-Aktenzeichen: **PCT/US2005/015943**  
 (86) PCT-Anmeldetag: **04.05.2005**  
 (87) PCT-Veröffentlichungstag: **17.11.2005**  
 (43) Veröffentlichungstag der PCT Anmeldung  
 in deutscher Übersetzung: **24.05.2007**

(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:  
**Lucas, Michael J., Leicestershire, GB; Hao, Tennyson, Quezon, PH; Guzman, Francis de, Manor, Tex., US; Campney, Bruce, Manor, Tex., US; Campney, Bruce, Manor, Tex., US; Nixon, Mark J., Round Rock, Tex., US; Gilbert, Stephen, Austin, Tex., US**

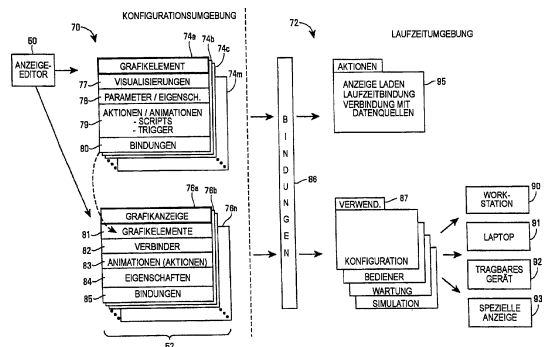
(54) Bezeichnung: **Zugeordnete graphische Anzeigen in einer Prozessumgebung**

(57) Hauptanspruch: Grafikanzeige-Editor zur Verwendung bei der Erzeugung einer Anwenderschnittstelle-Anzeige, welche den Betrieb von einem oder mehr Subjekten in der Verarbeitungsanlage darstellt, wobei der Grafikanzeige-Editor aufweist:

eine Bibliothek von Grafikobjekten, wobei jedes Grafikobjekt eine visuelle Darstellung eines physischen oder eines logischen Subjektes in der Verarbeitungsanlage umfasst;  
 eine grafikbasierte Editor-Canvas-Routine, die es einem Anwender ermöglicht, eine ausführbare grafische Anzeige durch Anordnen von einer oder mehr visuellen Darstellungen der Grafikobjekte aus der Bibliothek von Grafikobjekten auf einem Editier-Canvas zu definieren, um eine Weise zu definieren, auf die die eine oder mehr visuellen Darstellungen der Grafikobjekte einem Anwender auf einer Anzeigevorrichtung während der Ausführung der grafischen Anzeige angezeigt werden;

eine Eigenschaftendefinitions-Canvasroutine, die dazu ausgelegt ist, es einem Anwender zu ermöglichen, ein Eigenschaft zu definieren, die mindestens einem der Vielzahl von Grafikobjekten zugeordnet ist;

eine Bindungsdefinitions-Routine, die dazu ausgelegt ist, es einem Anwender zu ermöglichen, eine Bindung zwischen der Eigenschaft und einer...



**Beschreibung**

## Verwandte Anmeldungen

**[0001]** Die vorliegende Anmeldung ist eine regulär hinterlegte Anmeldung von, und die Priorität der am 04.05.2004 eingereichten Vorläufigen US-Anmeldung mit dem dem amtlichen Aktenzeichen 60/567,980 und der Bezeichnung "Graphical User Interface for Representing, Monitoring, and Interacting with Process Control Systems", auf die hiermit ausdrücklich vollinhaltlich Bezug genommen wird. Die vorliegende Anmeldung ist auch mit der am 21.07.2003 hinterlegten US-Patentanmeldung mit dem amtlichen Aktenzeichen 10/625,481 und der Bezeichnung "Integration of Graphic Display Elements, Process Modules and Control Modules in Process Plants" verwandt, die am 05.08.2004 als US-Veröffentlichung Nr. 2004/0153804 veröffentlicht wurde, die wiederum eine Teilfortführung der US-Patentanmeldung mit dem amtlichen Aktenzeichen 10/278,469 mit der Bezeichnung "Smart Process Modules and Objects in Process Plants" ist, die am 22.10.2002 hinterlegt und am 22.04.2004 als US-Veröffentlichung Nr. 2004/0075689 veröffentlicht wurde, auf deren Offenbarungsgehalt hiermit ausdrücklich vollinhaltlich Bezug genommen wird. Die vorliegende Anmeldung ist auch mit der am 18.02.2003 hinterlegten US-Patentanmeldung mit dem amtlichen Aktenzeichen 10/368,151 und der Bezeichnung "Module Class Objects in a Process Plant Configuration System" verwandt, die am 07.10.2004 als US-Veröffentlichung Nr. 2004/0199925 veröffentlicht wurde, auf deren Offenbarungsgehalt hiermit ausdrücklich vollinhaltlich Bezug genommen wird. Die vorliegende Anmeldung ist auch mit den folgenden Patentanmeldungen verwandt, die am gleichen Datum wie die vorliegende Anmeldung als Internationale (PCT) Anmeldungen eingereicht werden und auf deren Offenbarungsgehalt hiermit ausdrücklich vollinhaltlich Bezug genommen wird: "User Configurable Alarms and Alarm Trending for Process Control Systems" (Anwaltsaktenzeichen 06005/41112); "Integration of Process Modules and Expert Systems in Process Plants" (Anwaltsaktenzeichen 06005/41113); "A Process Plant User Interface System Having Customized Process Graphic Display Layers in an Integrated Environment" (06005/41114); "Scripted Graphics in a Process Environment" (Anwaltsaktenzeichen 06005/41115); "Graphics Integration into a Process Configuration and Control Environment" (Anwaltsaktenzeichen 06005/41116); "Graphic Element with Multiple Visualizations in a Process Environment" (Anwaltsaktenzeichen 06005/41117); "System for Configuring Graphic Display Elements and Process Modules in Process Plants" (Anwaltsaktenzeichen 06005/41118); "Graphic Display Configuration Framework for Unified Process Control System Interface" (Anwaltsaktenzeichen 06005/41124); "Markup Language-Based, Dynamic Process Graphics in a

Process Plant User Interface" (Anwaltsaktenzeichen 06005/41127); "Methods and Apparatus for Modifying Process Control Data" (Anwaltsaktenzeichen 06005/591622 und 20040/59-11622); "Methods and Apparatus for Accessing Process Control Data" (Anwaltsaktenzeichen 06005/591623 und 20040/59-11623); "Integrated Graphical Runtime Interface for Process Control Systems" (Anwaltsaktenzeichen 06005/591628 und 20040/59-11628); "Service-Oriented Architecture for Process Control Systems" (Anwaltsaktenzeichen 06005/591629 und 20040/59-11629).

## Technisches Gebiet

**[0002]** Die vorliegende Erfindung betrifft im Allgemeinen Verarbeitungsanlagen, und genauer gesagt die Integration und Verwendung eines Grafikanzeige-Editors und von grafischen Anzeigeobjekten auf einer Systemebene eines Prozesssteuerungs- und Simulationssystems, um die Erzeugung und Verwendung von gemeinsamen grafischen Anzeigeelementen bei verschiedenen Aktivitäten zu ermöglichen, die mit der Konfiguration, Steuerung, Wartung und Simulation einer Anlage in Zusammenhang stehen.

## Beschreibung des einschlägigen Standes der Technik

**[0003]** Verteilte Prozesssteuerungssysteme wie diejenigen, die in chemischen, Erdöl verarbeitenden oder anderen Prozessen verwendet werden, weisen für gewöhnlich einen oder mehr Prozess-Controller auf, die kommunikationsmäßig mit einer oder mehr Feldvorrichtungen über analoge, digitale oder kombinierte Analog- und Digitalbusse gekoppelt sind. Die Feldvorrichtungen, bei denen es sich beispielsweise um Ventile, Ventilstellglieder, Schalter und Sender (z.B. Temperatur-, Druck-, Füllstand- und Volumensensoren) handeln kann, befinden sich innerhalb der Prozessumgebung und erfüllen Prozessfunktionen wie etwa das Öffnen oder Schließen von Ventilen, das Messen von Prozessparametern, usw. Smart-Feldvorrichtungen, wie etwa die mit den allgemein bekannten Feldbus-Protokollen wie dem FOUNDATION™ Feldbus Protokoll konformen Feldvorrichtungen, können auch Steuerberechnungen, Warnfunktionen und andere Steuerfunktionen erfüllen, die üblicherweise in dem Controller implementiert sind. Die Prozess-Controller, die sich für gewöhnlich auch innerhalb der Anlagenumgebung befinden, empfangen Signale, welche von den Feldvorrichtungen gemachte Prozessmessungen angeben, und/oder andere Informationen, welche die Feldvorrichtungen betreffen, und führen eine Controller-Anwendung aus, die beispielsweise verschiedene Steuermodule betreibt, welche Prozesssteuerentscheidungen treffen, Steuersignale auf der Grundlage der empfangenen Informationen erzeugen und mit den Steuermodulen oder Blöcken koordinieren, die in den

Feldvorrichtungen wie etwa HART- und Fieldbus-Feldvorrichtungen ausgeführt werden. Die Steuermodule in dem Controller senden die Steuersignale über die Kommunikationsleitungen an die Feldvorrichtungen, um dadurch die Operation des Prozesses zu steuern.

**[0004]** Informationen von den Feldvorrichtungen und dem Controller werden üblicherweise über einen Datenbus einer oder mehr anderen Hardware-Vorrichtungen wie etwa Bediener-Workstations, Personalcomputern, Datenhistorieneinrichtungen, Meldungsgeneratoren, zentralisierten Datenbanken usw. verfügbar gemacht, die für gewöhnlich in Steuerzentralen oder an anderen Stellen von der rauerer Anlagenumgebung entfernt angeordnet sind. Diese Hardware-Vorrichtungen betreiben Anwendungen, die beispielsweise einen Bediener in die Lage versetzen können, Funktionen im Hinblick auf den Prozess durchzuführen, wie etwa das Ändern von Einstellungen der Prozess-Steueroutine, Modifizieren des Betriebs der Steuermodule in dem Controller oder den Feldvorrichtungen, Betrachten des aktuellen Zustands des Prozesses, Betrachten von Warnungen, die von Feldvorrichtungen und Controllern erzeugt wurden, Simulieren des Betriebs des Prozesses zum Zweck der Schulung von Personal oder Testen der Prozesssteuerungs-Software, Führen und Aktualisieren einer Konfigurations-Datenbank usw.

**[0005]** Als ein Beispiel weist das von Emerson Process Management vertriebene DeltaV™ -Steuersystem eine Vielzahl von Anwendungen auf, die in verschiedenen Vorrichtungen gespeichert sind und ausgeführt werden, welche sich an unterschiedlichen Orten innerhalb einer Verarbeitungsanlage befinden. Eine Konfigurationsanwendung, die sich in einer oder mehr Bediener-Workstations befindet, ermöglicht es Anwendern, Prozesssteuer-Module zu erzeugen oder zu ändern und diese Prozesssteuer-Module über einen Datenbus auf dedizierte verteilte Controller herunterzuladen. Für gewöhnlich bestehen diese Steuermodule aus kommunikationsmäßig untereinander verbundenen Funktionsblöcken, die Objekte in einem objektorientierten Programmierprotokoll sind, welche Funktionen innerhalb des Steuerverfahrens auf der Grundlage von Eingaben erfüllen und Ausgänge an andere Funktionsblöcke innerhalb des Steuerverfahrens liefern. Die Konfigurationsanwendung kann es auch einem Designer ermöglichen, Bedieneransichten zu erzeugen oder zu ändern, die von einer Betrachtungsanwendung verwendet werden, um Daten für einen Bediener anzuzeigen und es dem Bediener zu ermöglichen, Einstellungen wie etwa Sollwerte in der Prozesssteuer-Routine zu ändern. Jeder dedizierte Controller, und in einigen Fällen Feldvorrichtungen, speichert und führt eine Controller-Anwendung aus, welche die ihm zugeteilten und auf ihn heruntergeladenen Steuermodule betreibt, um eine tatsächliche Prozesssteuer-Funktio-

nalität zu verwirklichen. Die Betrachtungsanwendungen, die auf einer oder mehr Bediener-Workstations betrieben werden können, empfangen Daten von der Controller-Anwendung über den Datenbus und zeigen diese Daten Prozesssteuerungssystem-Designern, Bedienern oder Anwendern unter Verwendung der Anwenderschnittstellen an, und können jegliche einer Anzahl von verschiedenen Ansichten zur Verfügung stellen, wie etwa eine Bedieneransicht, eine Ingenieursansicht, eine Technikeransicht usw. Eine Datenhistorienanwendung wird für gewöhnlich in einer Datenhistorienvorrichtung gespeichert und ausgeführt, die einige oder alle der über den Datenbus gelieferten Daten sammelt und speichert, während eine Konfigurationsdatenbank-Anwendung auf wieder einem anderen Computer laufen kann, der an den Datenbus angeschlossen ist, um die aktuelle Prozesssteueroutine-Konfiguration und damit in Zusammenhang stehende Daten zu speichern. Als Alternative kann sich die Konfigurationsdatenbank in der gleichen Workstation wie die Konfigurationsanwendung befinden.

**[0006]** Da die Anzahl und Art von in einer Prozesssteuerungs-Umgebung verwendeten Steuer- und Unterstützungsanwendungen mehr geworden sind, wurden verschiedene Grafikanzeige-Anwendungen zur Verfügung gestellt, um es Anwendern zu ermöglichen, diese Anwendungen effektiv zu konfigurieren und zu verwenden. Beispielsweise wurden Grafikanzeige-Anwendungen zum Unterstützen von Steuerungskonfiguration-Anwendungen verwendet, um einen Konfigurierungstechniker in die Lage zu versetzen, Steuerprogramme, die dann auf die Steuervorrichtungen innerhalb einer Verarbeitungsanlage heruntergeladen werden, grafisch zu erzeugen. Zusätzlich wurden Grafikanzeige-Anwendungen verwendet, um es Steuerungsbedienern zu ermöglichen, den aktuellen Betrieb der Verarbeitungsanlage oder Bereiche der Verarbeitungsanlage zu betrachten, um es Wartungspersonal zu ermöglichen, den Zustand von Hardware-Vorrichtungen innerhalb der Verarbeitungsanlage zu betrachten, um eine Simulation der Verarbeitungsanlage zu ermöglichen, usw. Diese Grafikanzeige-Anwendungen wurden jedoch in der Vergangenheit als Teil oder zur Unterstützung der konkreten Anwendungen erzeugt, denen sie zugeordnet sind, und besitzen daher im Allgemeinen beschränkt ist, für die sie erzeugt wurden. Beispielsweise ist es schwierig oder sogar unmöglich, ein Grafikprogramm, das als Unterstützung für eine Steuerung oder andere Bediener erzeugt wurde, in einer Wartungs-Konfigurations- oder Simulationsfunktion zu verwenden.

**[0007]** Als ein konkretes Beispiel weisen einige Prozesssteuer-Konfigurationsanwendungen gegenwärtig eine Bibliothek von Schablonenobjekten wie etwa Funktionsblock-Schablonenobjekte, und in einigen

Fällen Steuermodul-Schablonenobjekte auf, die dazu verwendet werden, eine Steuerungsstrategie für eine Verarbeitungsanlage zu erzeugen. Den Schablonenobjekten sind Vorgabeeigenschaften, -eigenschaften und -verfahren zugeordnet, und der Techniker, der eine Grafik-Konfigurationsanwendung anwendet, kann diese Schablonenobjekte wählen und im Wesentlichen Kopien der ausgewählten Schablonenobjekte auf einem Konfigurationsbildschirm anordnen, um ein Steuermodul zu entwickeln. Während des Vorgangs des Auswählens und Anordnens der Schablonenobjekte auf dem Konfigurationsbildschirm verbindet der Techniker die Eingänge und Ausgänge dieser Objekte und ändert ihre Parameter, Bezeichnungen, Etiketten und andere Eigenschaften, um ein konkretes Steuermodul für eine konkrete Verwendung in der Verarbeitungsanlage zu erzeugen. Nach der Erzeugung von einem oder mehr solchen Steuermodulen kann der Techniker dann das Steuermodul erstellen und auf den/die geeigneten Controller und Feldvorrichtungen herunterladen, damit sie während des Betriebs der Verarbeitungsanlage ausgeführt werden.

**[0008]** Danach kann der Techniker eine verschiedene Grafikanzeige-Erzeugungsanwendung verwenden, um eine oder mehr Anzeigen für Bediener, Wartungspersonal usw. innerhalb der Verarbeitungsanlage zu erzeugen, indem er Anzeigeobjekte in der Anzeigeerzeugung-Anwendung auswählt und aufbaut. Diese Anzeigen werden für gewöhnlich auf einer systemweiten Basis in einer oder mehr der Workstations implementiert und stellen dem Bediener oder dem Wartungspersonal vorkonfigurierte Anzeigen im Hinblick auf den Betriebszustand des Steuersystems oder der Vorrichtungen innerhalb des Betriebs zur Verfügung. Diese Anzeigen haben im Allgemeinen die Form von Warnanzeigen, die von Controllern oder Vorrichtungen innerhalb der Verarbeitungsanlage erzeugte Warnungen empfangen und anzeigen; Steuerungsanzeigen, die den Betriebszustand der Controllern und anderer Vorrichtungen innerhalb der Verarbeitungsanlage angeben, Wartungsanzeigen, welche den Funktionszustand der Vorrichtungen innerhalb der Verarbeitungsanlage angeben; usw. Diese Anzeigen sind jedoch im Allgemeinen vorkonfiguriert, um auf eine bekannte Weise Informationen oder Daten anzuzeigen, die von den Prozesssteuer-Modulen oder den Vorrichtungen innerhalb der Verarbeitungsanlage empfangen wurden. Bei einigen Systemen werden Anzeigen von einer grafischen Abbildung erzeugt, die ein physisches oder ein logisches Element darstellt und kommunikationsmäßig in das physische oder logische Element eingebunden ist, um Daten über das physische oder logische Element zu erhalten. Die Grafik auf dem Anzeigebildschirm kann sich im Ansprechen auf bestimmte Ereignisse wie etwa auf empfangene Daten ändern, um beispielsweise zu veranschaulichen, dass ein Tank halb voll ist, um die von einem Strömungssensor gemessene

Strömung zu veranschaulichen, usw. Die grafische Anzeigen, die für Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsaktivitäten verwendet werden, werden jedoch im Allgemeinen separat von einander unter Verwendung verschiedener Grafik-Editoren erzeugt. Ferner sind die begrenzten Grafikfähigkeiten dieser Anzeigen schwer zu implementieren und werden auch nicht als Teil eines Grafikobjektes implementiert.

**[0009]** Somit kann die Anzeigeerzeugung-Anwendung, ähnlich wie die Steuerungskonfiguration-Anwendung, grafische Schablonenanzeige-Gegenstände wie etwa Tanks, Ventile, Sensoren, Bediener Steuerknöpfe wie etwa Schiebeleisten, Ein/Aus-Schalter usw. aufweisen, die auf einem Bildschirm in einer beliebigen Konfiguration angeordnet werden können, um eine Bedieneranzeige, Wartungsanzeige und dergleichen zu erzeugen. Bei ihrem Anordnen auf dem Bildschirm können individuelle grafische Gegenstände auf dem Bildschirm mit einander auf eine Weise verbunden werden, die Anwendern einige Informationen über bzw. eine Darstellung des inneren Zusammenhangs der Verarbeitungsanlage zur Verfügung stellt. Um die grafische Anzeige zu animieren, muss jedoch der Anzeigengenerator manuell jeden der grafischen Gegenstände mit Daten einbinden, die innerhalb der Verarbeitungsanlage erzeugt werden, wie etwa Daten, die von Sensoren gemessen werden oder die Ventilstellungen angeben usw., indem eine Kommunikationsverbindung zwischen dem grafischen Gegenstand und der relevanten Datenquelle innerhalb der Verarbeitungsanlage angegeben wird. Dieser Vorgang ist mühsam, zeitraubend, und kann mit Fehlern behaftet sein.

**[0010]** Während die Steuerungsschablonenobjekte in der Steuerungskonfiguration-Anwendung und die Anzeigegegenstände in der Anzeigeerzeugung-Anwendung nützlich sind, weil sie kopiert und verwendet werden können, um viele verschiedene Steuermodule und grafische Anzeigen zu erzeugen, besteht häufig ein Bedarf danach, zahlreiche des gleichen Steuermoduls und der gleichen grafischen Anzeige für verschiedene Einrichtungen innerhalb der Verarbeitungsanlage zu erzeugen. Beispielsweise weisen viele mittelgroße bis große Verarbeitungsanlagen mehrere Exemplare einer gleichen oder ähnlichen Einrichtung auf, die unter Verwendung des gleichen grundlegenden, allgemeinen Steuermoduls und der gleichen grundlegenden, allgemeinen Anzeige gesteuert und betrachtet werden können. Um diese zahlreichen Steuermodule und Anzeigen zu erzeugen, wird jedoch ein allgemeines Steuermodul oder Anzeigemodul erzeugt, und dieses allgemeine Steuer- oder Anzeigemodul wird dann für jedes Exemplar der Einrichtung, auf das es angewendet werden kann, kopiert. Natürlich muss jedes der neuen Steuer- oder Anzeigemodule nach dem Kopieren in der Konfigurationsanwendung manuell abgewandelt

werden, um die konkrete Einrichtung zu bezeichnen, der es zugeordnet ist, und alle diese Steuer- und Anzeigemodule müssen dann erstellt und auf das Prozesssteuerungssystem heruntergeladen werden.

**[0011]** Unglücklicherweise sind die oben erörterten Steuermodule und Anzeigegegenstände keineswegs modular. Somit müssen alle Steuermodule und Anzeigen nach ihrem Kopieren manuell und individuell unter Verwendung der geeigneten Konfigurationsanwendung abgewandelt werden, um die Einrichtung innerhalb des Betriebs anzugeben, der sie zugeordnet sein sollen. Bei einer Anlage mit vielen Kopien eines gleichen Einrichtungstyps (d.h. einer replizierten Einrichtung) ist dieser Vorgang mühsam, zeitraubend und mit Fehlern befrachtet, die von Bedienern eingeführt werden. Ferner wissen diese verschiedenen Steuermodule und Anzeigen nach ihrem Programmieren nichts von einander. Um eine Änderung an den bereits erzeugten Steuermodulen vorzunehmen, muss der Techniker oder die Bediener daher die gleiche Änderung an jedem der verschiedenen Steuermodule für die verschiedene replizierte Einrichtung manuell vornehmen, was wiederum zeitraubend und mühsam ist. Das gleiche Problem trifft auf die grafischen Ansichten zu, die für die verschiedenen Gruppen von replizierten Einrichtungen innerhalb des Betriebs erzeugt werden. Mit anderen Worten, sobald ein bestimmtes Steuermodul oder eine bestimmte grafische Ansicht einmal erzeugt worden ist (individuell oder durch Kopieren von einem Schablonenobjekt) und dann an eine bestimmten Gruppe von Einrichtungen innerhalb des Betriebs gebunden ist, existiert dieses Steuermodul oder diese grafische Ansicht als ein separates Subjekt oder ein separates Objekt im System ohne jegliches automatisches Wissen um die anderen Steuermodule oder grafischen Anzeigen, die ihm gleich oder ähnlich sind. Im Ergebnis müssen Änderungen, die auf jedes der Steuermodule und jede der grafischen Anzeigen eines bestimmten Typs anwendbar sind, individuell an diesen Modulen und Anzeigen vorgenommen werden. Dieses Problem tritt sogar noch deutlicher zu Tage, wenn grafische Ansichten für die gleiche Einrichtung, aber in verschiedenen Funktionszusammenhängen innerhalb der Anlage erzeugt werden, wie etwa für Steuer-, Betrachtungs-, Wartungsansicht- und Simulationsfunktionen. In diesem Fall werden die grafischen Ansichten separat, ohne gegenseitiges Wissen oder gegenseitige Bewusstheit erzeugt.

**[0012]** Auch wenn grafische Anzeigen verschiedenen Anwendungen zur Verfügung gestellt und zugeordnet worden sind, die für verschiedene, innerhalb einer Verarbeitungsanlage durchgeführte allgemeine Aktivitäten verwendet werden, wurden diese grafischen Anzeigen und zugeordneten Grafikanzeige-Editoren daher im Allgemeinen auf der Funktionsebene der Anwendung, für deren Unterstützung sie erzeugt wurden, hinzugefügt. Im Ergebnis ermöglich-

ten die Grafik-Editoren, so weit sie existierten, dem Anwender nur das Erzeugen von Grafiken, die eine konkrete, von einer konkreten Anwendung benötigte Funktionalität unterstützen. Frühere Verarbeitungsanlagen stellten keinen Grafikanzeige-Editor zur Verfügung, der von verschiedenen oder mehreren im Kontext der Anlagenkonfiguration durchgeführten Aktivitäten verwendet werden konnte oder deren Grafikbedürfnisse unterstützen konnte. Somit versetzte beispielsweise ein Grafikanzeige-Editor, der dazu verwendet wurde, Steuerkonfigurationsaktivitäten zu unterstützen oder zu ermöglichen, den Anwender nur in die Lage, Steuerprogramme zu erzeugen, unterstützte aber nicht die Bedürfnisse oder die Funktionalität von Bediener- oder Wartungsanzeigen. Auf ähnliche Weise unterstützten Grafikanzeige-Editoren, die dazu verwendet wurden, Bedieneransichten, Wartungsansichten oder dergleichen zu erzeugen, die einem Steuerungsbediener oder einem Wartungstechniker während des Betriebs einer Anlage zur Verfügung gestellt werden, keine Funktionalität im Zusammenhang mit Konfigurationsaktivitäten, Simulationsaktivitäten oder dergleichen. Als Ergebnis davon, dass die grafischen Anzeigebedürfnisse auf den einzelnen Funktionsebenen der Verarbeitungsanlage wie etwa den Funktionsebenen der Steuerungskonfiguration, der Wartungsunterstützung, der Steuerungsbediener-Unterstützung und der Simulationsunterstützung unterstützt werden, werden letztlich gleiche Komponenten innerhalb der Anlage von verschiedenen der von diesen unterschiedlichen Editoren erzeugten Anzeigen modelliert und abgebildet, was zu einer Duplizierung des Aufwandes für die grafische Anzeige durch verschiedene, unterschiedliches Personal in der Verarbeitungsanlage führt. Diese Duplizierung des Aufwands zeigt sich nicht nur an dem Aufwand, der erforderlich ist, um die verschiedenen grafischen Anzeigen zu erzeugen, welche das gleiche Prozesselement für verschiedene Verwendungen abbilden, sondern auch in dem Aufwand, der erforderlich ist, um die in verschiedenen Anzeigeanwendungen verwendeten grafischen Elemente mit den tatsächlichen Hardware- oder Software-Elementen innerhalb der Verarbeitungsanlage, denen sie zugeordnet sind, einzubinden.

**[0013]** Weil grafische Unterstützung für verschiedene Aktivitäten einer Verarbeitungsanlage erst im Nachhinein und als Teil der tatsächlich durchgeführten Aktivität zur Verfügung gestellt wurde, ist die grafische Unterstützung nicht so in die Betriebsumgebung integriert, dass gemeinsame Grafiken innerhalb der Anlage auf den verschiedenen unterschiedlichen Funktionsebenen der Anlage erzeugt und verwendet werden könnten. Diese Nicht-Integration von Grafik führt dazu, dass die tatsächlich für die verschiedenen Funktionen erzeugten Grafiken von Funktion zu Funktion oder von Anwendung zu Anwendung verschieden sind, was zu Verwirrung seitens der Anwen-

der führen kann, die zwar mit einem konkreten Typ einer grafischen Anzeige vertraut sein mögen, jedoch gelegentlich verschiedene Anzeigen im Zusammenhang mit verschiedenen Operationen oder Funktionen innerhalb der Anlage betrachten müssen. Ebenso, wie weiter oben angemerkt wurde, führt das Vorsehen einer Grafikanzeige-Unterstützung auf den verschiedenen unterschiedlichen Funktionsebenen der Anlage zu einer Duplizierung der grafischen Unterstützung sowohl beim Erzeugen von Anzeigen als auch beim angemessenen Verbinden der Elemente innerhalb der Anzeigen mit tatsächlichen Hardware- oder Software-Elementen innerhalb der Anlage.

**[0014]** Zusätzlich sind Fehlererfassung und andere Programmierungen nützlich, um Bedingungen, Fehler, Warnungen usw. im Zusammenhang mit Steuer Schleifen, die auf den verschiedenen Controllern ablaufen, und Probleme innerhalb der einzelnen Vorrichtungen zu entdecken. Eine solche Fehlererfassung wurde herkömmlicherweise auf den verschiedenen Funktionsebenen der Verarbeitungsanlage durchgeführt und auf grafischen Anzeigen angezeigt, die für diese verschiedenen funktionalen Aktivitäten erzeugt worden waren. Es war daher schwierig, das Prozesssteuerungssystem so zu programmieren, dass es Zustände oder Fehler auf Systemebene erfasst, die durch das Analysieren von Daten von verschiedenen möglichen Vorrichtungen an unterschiedlichen Orten innerhalb der Verarbeitungsanlage entdeckt werden müssen, und sogar noch schwieriger, diese Fehlertypen auf Bedieneranzeigen zu zeigen, die nicht zu dem Zweck erzeugt wurden, solche Informationen über Bedingungen auf Systemebene für Bediener oder Wartungspersonal anzuzeigen oder darzustellen. Ferner ist es schwierig, Objekte in Bedieneranzeigen mit diesen wechselnden Quellen von Informationen oder Daten für die verschiedenen Elemente in der Anzeige zu animieren.

#### Zusammenfassung der Beschreibung

**[0015]** Grafikanzeige-Unterstützung wird innerhalb einer Verarbeitungsanlagekonfigurations-, Überwachungs- und Simulationssystem zur Verfügung gestellt, um grafische Anzeigen zu ermöglichen, die auf eine solche Weise erzeugt werden, dass sie einander in der Laufzeitumgebung zugeordnet sind. Insbesondere kann ein einzelner Grafikanzeige-Editor verwendet werden, um verschiedene unter einander in Beziehung stehende grafische Anzeigen zu erzeugen, auf die beispielsweise von einander in der Laufzeitumgebung zugegriffen werden kann, um weitere Informationen über ein Prozess-Subjekt in einer der grafischen Anzeigen zur Verfügung zu stellen, durch benachbarte Sektionen einer Verarbeitungsanlage zu scrollen, oder verschiedene Anzeigen für verschiedene Funktionen innerhalb der Verarbeitungsanlage zur Verfügung zu stellen, wie etwa für eine Bediener-Betrachtungsfunktion, eine Simulations-

funktion und eine Wartungsfunktion. Weil der gleiche Grafik-Editor verwendet wird, um die grafischen Anzeigen zu erzeugen, können die resultierenden grafischen Anzeigen das gleiche Aussehen und Gefühl besitzen und können innerhalb der Anlage im Allgemeinen auf eine gleiche Weise an die Laufzeitumgebung gebunden werden, wodurch die Zeit verkürzt wird, die erforderlich ist, um in einer Verarbeitungsanlage verwendete grafische Prozessanzeigen zu konfigurieren und zu erzeugen.

#### Kurze Beschreibung der Zeichnung

**[0016]** [Fig. 1](#) ist ein Blockdiagramm eines verteilten Prozesssteuerungssystems, das sich innerhalb einer Verarbeitungsanlage befindet und Bediener-Workstations aufweist, welche Anzeigeroutinen und andere Anwendungen im Zusammenhang mit verschiedenen Funktionen innerhalb der Verarbeitungsanlage implementieren, sowie eine Workstation, die grafische Unterstützung auf Systemebene zur Verfügung stellt, welche verwendet werden kann, um grafische Anzeigeelemente und grafische Anzeigen für jeden der verschiedenen Funktionsbereiche der Anlage zu erzeugen;

**[0017]** [Fig. 2](#) ist ein logisches Blockdiagramm, das die Integration der grafischen Unterstützung auf Systemebene innerhalb des Steuer-, Betrachtungs- und Simulationssystems einer Verarbeitungsanlage veranschaulicht;

**[0018]** [Fig. 3](#) ist ein Logikdiagramm, das eine Konfigurationsumgebung veranschaulicht, in der grafische Elemente und Anzeigen erzeugt werden, sowie eine Laufzeitumgebung, in der grafische Elemente und Anzeigen ausgeführt werden können;

**[0019]** [Fig. 4](#) ist ein vereinfachter Anzeigebildschirm, der von einem Grafik-Editor hergestellt wird, um einen Anwender in die Lage zu versetzen, ein grafisches Element in Form eines Pumpenelementes zu erzeugen oder zu editieren;

**[0020]** [Fig. 5](#) ist ein weiterer Anzeigebildschirm, der von einem Grafik-Editor hergestellt wird, während er ein grafisches Element in Form eines Reaktors herstellt;

**[0021]** [Fig. 6](#) ist ein Blockdiagramm, das eine Weise veranschaulicht, auf die verschiedene Visualisierungen an ein grafisches Element gebunden oder diesem zugeordnet werden können;

**[0022]** [Fig. 7](#) stellt einen Ausschnitt einer Bildschirmanzeige dar, der eine erste Eigenschaftenansicht zeigt, die einem grafischen Element zugeordnet ist;

**[0023]** [Fig. 8](#) stellt einen Ausschnitt einer Bildschirmanzeige dar, der eine zweite Eigenschaftenansicht

im Zusammenhang mit einem grafischen Element zeigt;

[0024] [Fig. 9](#) stellt einen Aktionen/Animationen-Ausschnitt eines Anzeigebildschirms dar, der Aktionen im Zusammenhang mit Eigenschaften eines grafischen Elementes zeigt;

[0025] [Fig. 10](#) ist ein Blockdiagramm, das eine Weise veranschaulicht, auf die visuelle Trigger mit Eigenschaften und Visualisierungen eines grafischen Elementes integriert werden können;

[0026] [Fig. 11](#) stellt eine Dialogbox dar, die verwendet werden kann, um eine Transformierungsanimation für eine Visualisierung eines grafischen Elementes zur Verfügung zu stellen oder zu definieren;

[0027] [Fig. 12](#) stellt eine Dialogbox dar, die verwendet werden kann, um eine Eigenschaftenanimation einschließlich einer Farbanimation für eine Visualisierung eines grafischen Elementes zur Verfügung zu stellen oder zu definieren;

[0028] [Fig. 13](#) ist ein Diagramm, das die Weise, auf die grafische Elemente durch eine Auflösungstabelle an eine Laufzeitumgebung gebunden werden können, allgemein veranschaulicht;

[0029] [Fig. 14](#) ist ein Blockdiagramm, das ein an mehrere verschiedene Datenquellen gebundenes Anzeigeelement in einer Prozessumgebung darstellt;

[0030] [Fig. 15](#) ist ein von einem Grafik-Editor hergestellter Anzeigebildschirm, während er eine grafische Anzeige aus mehreren Anzeigeelementen und Verbindern herstellt;

[0031] [Fig. 16](#) ist ein Anzeigebildschirm zum Erzeugen einer grafischen Anzeige, der verschiedene Visualisierungen im Zusammenhang mit einem in der grafischen Anzeige angeordneten grafischen Element veranschaulicht;

[0032] [Fig. 17](#) ist ein Anzeigebildschirm im Zusammenhang mit einer grafischen Anzeige, die aus verschiedenen untereinander verbundenen grafischen Elementen und Verbindern besteht;

[0033] [Fig. 18](#) ist eine Gruppe von Grafikanzeige-Bildschirmen, auf die über die grafische Anzeige von [Fig. 17](#) zugegriffen werden kann, um eine Schalttafel und ein Faceplate für ein Element in der Anzeige von [Fig. 17](#) zu zeigen;

[0034] [Fig. 19](#) ist ein Anzeigebildschirm einer grafischen Anzeige, die einen aus verschiedenen Anzeigeelementen bestehenden Kalkofen veranschaulicht und Daten von verschiedenen anderen Anwendungen und Datenquellen innerhalb einer Verarbeitungs-

anlage zur Verfügung stellt;

[0035] [Fig. 20A–Fig. 20E](#) sind Grafikanzeige-Bildschirme, die verschiedene Ansichten eines Kalkofens mit einem gleichen Aussehen und Gefühl einschließlich einer Bedieneransicht, einer Ingenieursansicht, einer Geschäftsansicht, einer Simulationsansicht und einer Wartungsansicht veranschaulichen;

[0036] [Fig. 21A](#) und [Fig. 21B](#) sind Anzeigebildschirme im Zusammenhang mit einer zum Betreiben des in den [Fig. 20A–Fig. 20E](#) gezeigten Kalkofens verwendeten Steuerroutine;

[0037] [Fig. 22](#) ist ein Blockdiagramm, das Zusammenhänge zwischen verschiedenen unterschiedlichen grafischen Anzeigen auf verschiedenen Ebenen einschließlich physischen Ebenen und Funktionsebenen veranschaulicht;

[0038] [Fig. 23](#) stellt einen ersten Konfigurationsbildschirm dar, der eine grafische Konfigurationshierarchie im Zusammenhang mit einem Konfigurationssystem veranschaulicht;

[0039] [Fig. 24](#) stellt einen zweiten Konfigurationsbildschirm dar, der eine grafische Konfigurationshierarchie veranschaulicht und eine Weise zeigt, auf die grafische Elemente und grafische Anzeigen anderen Elementen in dem Konfigurationssystem zugeteilt und mit diesen integriert werden können; und

[0040] [Fig. 25](#) stellt eine Dialogbox dar, die einem Anwender zur Verfügung gestellt werden kann, um die Art und Weise, auf die eine oder mehr grafische Anzeigen innerhalb eines Verarbeitungsanlagen-Konfigurationssystem zugeteilt werden, zusammenfassend darzustellen.

#### Detaillierte Beschreibung

[0041] [Fig. 1](#) veranschaulicht eine beispielhafte Verarbeitungsanlage **10**, in der grafische Unterstützung auf Systemebene für verschiedene Funktionsbereiche der Anlage **10** zur Verfügung gestellt wird. Wie üblich weist die Verarbeitungsanlage **10** ein verteiltes Prozesssteuerungssystem mit einem oder mehr Controllern **12** auf, die jeweils mit einer oder mehr Feldvorrichtungen **14** und **16** über Eingabe-/Ausgabe (I/O)-Vorrichtungen oder Karten **18** verbunden sein können, wie beispielsweise Fieldbus-Schnittstellen, Profibus-Schnittstellen, HART-Schnittstellen, Standard **4–20** ma-Schnittstellen, usw. Die Controller **12** sind ferner mit einer oder mehr Host- oder Bediener-Workstations **20–23** über eine Datenbus **24** gekoppelt, bei der es sich beispielsweise um ein Ethernet-Link handeln kann. Eine Datenbank **28** kann mit dem die Datenbus **24** verbunden sein und dient als eine Datenhistorieneinrichtung



zum Sammeln und Speichern von Parameter-, Zustands- und anderen Daten im Zusammenhang mit den Controllern und Feldvorrichtungen innerhalb der Anlage **10** und/oder als eine Konfigurationsdatenbank, welche die aktuellen Konfigurationen des Prozesssteuerungssystems innerhalb der Anlage **10** speichert, die auf die Controllern **12** und Feldvorrichtungen **14** und **16** heruntergeladen und in diesen gespeichert werden. Die Datenbank **28** kann zusätzlich Grafikobjekte speichern, die auf die vorliegend beschriebene Weise erzeugt wurden, um grafische Unterstützung innerhalb der Verarbeitungsanlage **10** zur Verfügung zu stellen. Während sich die Controllern **12**, I/O-Karten **18** und Feldvorrichtungen **14** und **16** für gewöhnlich in der manchmal rauen Betriebsumgebung befinden und in dieser verteilt sind, befinden sich die Bediener-Workstations **20–23** und die Datenbank **28** üblicherweise in Steuerzentralen oder anderen, weniger rauen Umgebungen, auf die Controller oder Wartungspersonal leicht zugreifen können. In einigen Fällen können jedoch tragbare Vorrichtungen verwendet werden, um diese Funktionen zu implementieren, und diese tragbaren Vorrichtungen werden für gewöhnlich an verschiedene Orte in der Anlage getragen.

**[0042]** Bekanntermaßen wird von jedem der Controller **12**, bei denen es sich beispielhaft um den von Emerson Process Management vertriebenen DeltaV™-Controller handeln kann, eine Controller-Anwendung gespeichert und ausgeführt, die eine Steuerungsstrategie unter Verwendung einer beliebigen Anzahl von verschiedenen, unabhängig ausgeführten Steuermodulen oder Blöcken **29** implementiert. Jedes der Steuermodule **29** kann aus üblicherweise als Funktionsblöcke bezeichneten Elementen bestehen, wobei jeder Funktionsblock ein Teil oder eine Subroutine einer Gesamt-Steueroutine ist und in Verbindung mit anderen Funktionsblöcken (über Kommunikationen, die als Links bezeichnet werden) arbeitet, um Prozesssteuerschleifen innerhalb der Verarbeitungsanlage **10** zu implementieren. Wie allgemein bekannt ist, erfüllen Funktionsblöcke, die Objekte in einem objektorientierten Programmierprotokoll sein können, für gewöhnlich eine von einer Eingabefunktion, wie etwa der im Zusammenhang mit einem Sender, einem Sensor oder einer anderen Prozessparameter-Messvorrichtung, einer Steuerfunktion, wie etwa der im Zusammenhang mit einer Steueroutine, die eine PID-, Fuzzy Logic- oder eine andere Steuerung durchführt, oder einer Ausgabefunktion, welche den Betrieb einer Vorrichtung wie etwa eines Ventils steuert, um eine physische Funktion innerhalb der Verarbeitungsanlage **10** zu erfüllen. Natürlich gibt es hybride und andere Typen von komplexen Funktionsblöcken, wie etwa Model Predictive Controllern (MPCs), Optimierer usw. Während das Fieldbus-Protokoll und das DeltaV-Systemprotokoll Steuermodule und Funktionsblöcke verwenden, die in einem objektorientierten Programmierprotokoll

entworfen und implementiert sind, könnten die Steuermodule unter Verwendung einer jeglichen gewünschten Steuerprogrammierungstechnik einschließlich z.B. sequentieller Funktionsblock, Ladder Logic usw. entworfen werden und sind nicht auf einen Entwurf und eine Implementierung unter Verwendung des Funktionsblocks oder einer bestimmten anderen Programmierstechnik beschränkt.

**[0043]** In der in [Fig. 1](#) veranschaulichten Anlage **10** können die mit den Controllern **12** verbundenen Feldvorrichtungen **14** und **16** Standard 4–20 ma-Vorrichtungen, Smart-Feldvorrichtungen wie etwa HART, Profibus, oder FOUNDATION™ Fieldbus-Feldvorrichtungen sein, die einen Prozessor und einen Speicher aufweisen, oder können jeglicher andere gewünschte Vorrichtungstyp sein. Einige dieser Vorrichtungen, wie etwa Fieldbus-Feldvorrichtungen (Bezugszeichen **16** in [Fig. 1](#)), können Module oder Sub-Module wie etwa Funktionsblöcke im Zusammenhang mit der in den Controllern **12** implementierten Steuerungsstrategie speichern und ausführen. Die Funktionsblöcke **30**, die in [Fig. 1](#) als in zwei verschiedenen der Fieldbus-Feldvorrichtungen **16** angeordnet veranschaulicht sind, können in Verbindung mit der Ausführung der Steuermodule **29** in den Controllern **12** ausgeführt werden, um eine Prozesssteuerung zu implementieren, wie an sich bekannt ist. Natürlich können die Feldvorrichtungen **14** und **16** jeglicher Typ von Vorrichtungen sein, wie etwa Sensoren, Ventile, Sender, Stellglieder usw., und die I/O-Vorrichtungen **18** können jegliche Typen von I/O-Vorrichtungen sein, die mit jedem gewünschten Kommunikations- oder Controllerprotokoll wie etwa HART, Fieldbus, Profibus usw. konform sein können.

**[0044]** In der Verarbeitungsanlage **10** von [Fig. 1](#) können die Workstations **20–23** verschiedene Anwendungen aufweisen, die für verschiedene unterschiedliche Funktionen verwendet werden, welche von dem gleichen oder von wechselndem Personal innerhalb der Anlage **10** durchgeführt werden. Jede der Workstations **20–23** weist einen Speicher **31** auf, der verschiedene Anwendungen, Programme, Datenstrukturen usw. speichert, und einen Prozessor **32**, der verwendet werden kann, um jegliche der in dem Speicher **31** gespeicherten Anwendungen auszuführen. In dem in [Fig. 1](#) veranschaulichten Beispiel ist die Workstation **20** als eine Konfigurations-Workstation bezeichnet und weist eine oder mehr Konfigurationsanwendungen **33** auf, die beispielsweise Steuermodulerzeugungs-Anwendungen, Bediener-schnittstellen-Anwendungen und andere Datenstrukturen umfassen können, auf die von jedem befugten Konfigurierungstechniker zugegriffen werden kann, um Steuermodulen oder -module wie etwa die Steuermodule **29** und **30** zu erzeugen und auf die verschiedenen Controller **12** und Vorrichtungen **16** der Anlage **10** herunterzuladen. Die Workstation **21** ist in [Fig. 1](#) allgemein als eine Steuerungsbedieneran-



sicht-Workstation veranschaulicht und umfasst eine Anzahl von Anzeigeanwendungen **34**, die einem Steuerungsbediener während des Betriebs der Verarbeitungsanlage **10** verschiedene Anzeigen zur Verfügung stellen können, um es dem Bediener zu ermöglichen, die Geschehnisse innerhalb der Verarbeitungsanlage **10** oder in verschiedenen Sektionen der Anlage zu betrachten und zu steuern. Die Anwendungen **34** können Unterstützungsanwendungen **34a** wie etwa Steuer-, Diagnose-Anwendungen, Abstimmmanwendungen, Berichterzeugungs-Anwendungen oder jegliche andere Steuerungs-Unterstützungsanwendungen umfassen, die verwendet werden können, um einem Steuerungsbediener bei der Durchführung von Steuerfunktionen zu helfen. Auf ähnliche Weise ist die Workstation **22** als eine Wartungsansicht-Workstation veranschaulicht und umfasst eine Anzahl von Wartungsanwendungen **35**, die von verschiedenem Wartungspersonal verwendet werden können, um die Wartungserfordernisse der Anlage **10** zu betrachten, die Betriebs- oder Arbeitsbedingungen von verschiedenen Vorrichtungen **12**, **14**, **16** zu betrachten, usw. Natürlich können die Anwendungen **35** Unterstützungsanwendungen **35a** wie etwa Wartungsdiagnose-Anwendungen, Kalibrierungsanwendungen, Schwingungsanalyse-Anwendungen, Berichterzeugungs-Anwendungen oder jegliche andere Wartungsunterstützungs-Anwendungen umfassen, die verwendet werden können, um einer Wartungsperson bei der Durchführung von Wartungsfunktionen innerhalb der Anlage **10** zu helfen. Zusätzlich ist die Workstation **23** als eine Simulation-Workstation bezeichnet, die eine Anzahl von Simulationsanwendungen **36** umfasst, die verwendet werden können, um den Betrieb der Anlage **10** oder verschiedener Sektionen der Anlage **10** für eine beliebige Anzahl von Zwecken zu simulieren, einschließlich für Schulungszwecke, für Anlagen-Modellierungszwecke, um bei der Wartung und Steuerung der Anlage zu helfen, usw. Wie üblich weist jede der Workstations **20–23** einen Anzeigebildschirm **37** ebenso wie andere standardmäßige Peripherievorrichtungen, wie etwa eine Tastatur, eine Maus usw. auf.

**[0045]** Auch wenn die verschiedenen Konfigurations-, Steuerungs-, Wartungs- und Simulationsanwendungen **33–36** in [Fig. 1](#) so veranschaulicht sind, als befänden sie sich in verschiedenen Workstations, die für eine dieser Funktionen reserviert ist, sollte es natürlich verständlich sein, dass sich die verschiedenen Anwendungen **33–36** im Zusammenhang mit diesen oder anderen Anlagefunktionen je nach den Erfordernissen und des Aufbaus der Anlage **10** in einer gleichen oder in verschiedenen Workstations oder Computern innerhalb der Anlage **10** befinden und dort ausgeführt werden können. So können beispielsweise eine oder mehr Simulationsanwendungen **36** und Steueranwendungen **33** auf der gleichen Workstation **20–23** ausgeführt werden, während verschiedene einzelne Simulationsanwendungen **36**

oder verschiedene einzelne Steueranwendungen **33** in verschiedenen der Workstations **20–23** ausgeführt werden können.

**[0046]** In der Vergangenheit wurde die Entwicklung der verschiedenen Anwendungen, die in den verschiedenen Funktionsbereichen der Anlage **10** verwendet werden, auf ziemlich unabhängige Weise vorgenommen. Somit war die Entwicklung der Konfigurationsanwendungen **33** nicht mit den Simulationsanwendungen **36**, den Wartungsanwendungen **35** oder den Bedienersteueranwendungen **34** integriert. Tatsächlich konnte ein Anlage oftmals Anwendungen für die verschiedenen Funktionsbereiche umfassen, die von verschiedenen Firmen oder Softwarelieferanten entwickelt wurden, und die eigentlich dazu entwickelt wurden, um unabhängig von der anderen Software innerhalb der Anlage **10** zu laufen. Als ein Ergebnis dieser unabhängigen Entwicklung und dieses unabhängigen Betriebs der verschiedenen Anwendungen im Zusammenhang mit den verschiedenen Funktionsbereichen der Anlage **10** musste das Personal der Anlage im Allgemeinen eine Anlage separat auf jeder der Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsfunktionsebenen konfigurieren oder einstellen. Insbesondere musste das gleiche oder wechselndes Personal der Anlage im Allgemeinen verschiedene Programme verwenden, um neue Datenstrukturen und grafische Anzeigen auf jeder der Funktionsebenen zu erstellen. Somit umfasste oder verwendete bei Bezugnahme auf [Fig. 1](#) jede der verschiedenen Anwendungen **33–36**, die Konfigurations-, Steuerungs-, Wartungs- und Simulationsfunktionen durchführte, im Allgemeinen verschiedene Grafikanzeige-Editoren und Datenbankstrukturen, um dem Personal der Anlage bei der Durchführung dieser Konfigurations-, Bedienersteuerungs-, Wartungs- und Simulationsfunktionen zu helfen. Oftmals wurden diese verschiedenen Grafikanzeige-Editoren und Datenbanken verwendet, um verschiedene grafische Anzeigen zum Abbilden oder Modellieren der gleichen Sektionen oder Bereiche der Anlage **10** oder der gleichen Hardware innerhalb der Anlage **10** zu erzeugen, und um wechselndem Personal der Anlage dabei zu helfen, die Geschehnisse in der Verarbeitungsanlage im Hinblick auf Konfigurations-, Bedienersteuerungs-, Wartungs- oder Simulationsaktivitäten zu visualisieren und zu verstehen.

**[0047]** Da die Anwendungen **33–36** und die zugeordneten Anzeigen für jede der verschiedenen Funktionen innerhalb der Anlage **10** im Allgemeinen unabhängig von einander entwickelt und implementiert wurden, manchmal von verschiedenen Personen und sogar von verschiedenen Firmen, waren die in den verschiedenen Funktionsbereichen der Verarbeitungsanlage erzeugten oder verwendeten grafischen Anzeigen unter dem Gesichtspunkt des Zurverfügungstellens einer grafischen Visualisierung nicht auf

eine gleich bleibende oder leicht verständliche Weise integriert. Im Ergebnis führte die unabhängige Erzeugung und Ausführung von grafischen Anzeigen auf den verschiedenen unterschiedlichen Funktionsebenen der Anlage dazu, dass die grafischen Anzeigen von Funktion zu Funktion ein verschiedenes Aussehen hatten, so dass die grafischen Anzeigen durch die Funktionsbereiche kein gleichbleibendes Aussehen und Gefühl besaßen. Zusätzlich führte diese unabhängige Erzeugung zu der Duplizierung des Aufwands bei der Erzeugung von grafischen Anzeigen für die gleichen Sektionen oder Bereiche der Anlage, jedoch für verschiedene funktionale Verwendungen, und machte es erforderlich, dass die so erzeugten grafischen Anzeigen separat auf der Funktionsebenen der Anlage **10** eingebunden werden und Daten von den verschiedenen Vorrichtungen in der Anlage **10** wie etwa den Controllern **12** und den Feldvorrichtungen **14**, **16** empfangen. Dieser Umstand erforderte wiederum eine Duplizierung der Datenbankstrukturen, um die gleichen Hardware-Elemente für die verschiedenen Anzeigen zu verfolgen. Somit wurde beispielsweise in der Vergangenheit eine erste Anwendung (z.B. eine der Anwendungen **35**) dazu verwendet, ein Wartungsanzeige zu erzeugen, die eine Sektion der Anlage **10** für Wartungszwecke veranschaulicht, während eine zweite Anwendung (z.B. eine der Anwendungen **34**) verwendet wurde, um eine Steuerungsbediener-Anzeige zu erzeugen, welche die gleiche Sektion der Anlage **10** für Steuerungszwecke veranschaulicht. Die von diesen verschiedenen Anzeige-Editoren separat erzeugten Anzeigen konnten nach Aussehen und Gefühl ganz unterschiedlich sein, was es für einen Anwender schwierig machte, zwischen der Wartungsanzeige und der Bedieneranzeige hin und her zu schalten, ohne leicht verwirrt zu werden oder ohne eine Schulung auf jedem Anzeigetyp zu erfordern. Ebenso wurde der Aufwand bei der Erzeugung von sowohl Anzeigen unabhängig in den verschiedenen Anwendungen **34** und **35** dupliziert, und zusätzlicher Aufwand wurde getrieben, um Datenbankstrukturen zu erzeugen und die Wartungsanzeige und die Steuerungsbediener-Anzeige separat einzubinden oder mit den gleichen Hardware-Elementen in der Anlage **10** zu verbinden, um manchmal gleiche oder ähnliche Daten von diesen Hardware-Elementen zu erhalten.

**[0048]** Um diese Unzulänglichkeiten zu mildern und breiter verwendbare und verständliche Grafiken innerhalb der Anlage **10** zur Verfügung zu stellen, wird eine Grafikerunterstützungsschicht auf einer Systemebene der Verarbeitungsanlage **10** zur Verfügung gestellt, um die Erfordernisse im Hinblick auf die grafische Anzeige und Datenstruktur jedes der verschiedenen Funktionsbereiche der Anlage **10** zu unterstützen, einschließlich der Konfiguration, Bedieneransicht, Wartungsansicht, Simulationen und anderer Funktionsbereiche der Anlage **10**. Diese Systemebene der Unterstützung ist schematisch in [Fig. 2](#) abge-

bildet, die eine Anlagenbetriebsebene **40**, eine Anlagenfunktionsebene **42** und eine Systemebene **44** veranschaulicht. Wie aus [Fig. 2](#) ersichtlich ist, umfasst die Anlagenbetriebsebene **40** die Controller **12**, Feldvorrichtungen **14**, **16** usw., welche die Steuerroutinen oder Module **29** und **30** ausführen, sowie andere Software, die in der Anlage **10** betrieben wird, um Anlageoperationen während der Laufzeit der Anlage zu implementieren. Die Anlagenfunktionsebene **42** ist mit einem Konfigurationsfunktionsblock **46**, einem Steuerfunktionsblock **47**, einem Wartungsfunktionsblock **48** und einem Simulationsblock **49** abgebildet, obgleich ebenso gut andere oder verschiedene Funktionen wie etwa eine Ingenieurs- und eine Geschäftsfunktion zur Verfügung gestellt werden könnten. Der Konfigurationsfunktionsblock **46** implementiert die Konfigurationsroutinen **33**, die mit Komponenten innerhalb der Anlagenbetriebsebene **40** gekoppelt sind oder kommunizieren, um Steuerstrategien oder Steuermodule dafür zur Verfügung zu stellen. Der Steuerfunktionsblock **47** umfasst die Steuerungsansicht- und andere Anwendungen **34** und **34a**, die auch für gewöhnlich direkt mit den verschiedenen physischen und logischen Komponenten innerhalb der Anlagenbetriebsebene **40** gekoppelt sind oder kommunizieren, um von Bedienern veranlasste Änderungen in der Anlage **10** zu implementieren, den Bedienern Informationen über Steuerungsanzeigen **34** zur Verfügung zu stellen, Daten für die Steueranwendungen **34a** zu erhalten, usw. Der Wartungsfunktionsblock **48** umfasst die Wartungsroutinen und -anwendungen **35** und **35a**, die mit verschiedenen physischen und logischen Komponenten innerhalb der Anlagenbetriebsebene **40** gekoppelt sind oder kommunizieren, um Wartungsprozeduren zu implementieren, Wartungsdaten zu sammeln, einer Wartungsperson Wartungsdaten oder Informationen über Wartungsanzeigen **35** zur Verfügung zu stellen, diagnostische Anwendungen **35a** zu betreiben, usw. Ebenso umfasst der Simulationsfunktionsblock **49** Simulationsroutinen **36**, die Simulationen der Anlage **10** implementieren und kommunikationsmäßig mit Komponenten innerhalb der Anlagenbetriebsebene **40** gekoppelt sein können, um Daten bezüglich der Anlage **10** zu erhalten.

**[0049]** Wie in [Fig. 2](#) veranschaulicht ist, ist die Systemebene-Unterstützungsschicht **44** mit jedem der Funktionsblöcke **46–49** innerhalb der Anlagefunktionsschicht **42** eingebunden und unterstützt diese, um beispielsweise die Erzeugung und Führung von gemeinsamen Datenbank- und Anzeigestrukturen, wie etwa Software-Objekten, grafischen Elementen und grafischen Anzeigen zur Verwendung in den verschiedenen Funktionsbereichen **46–49** zu ermöglichen. Genauer gesagt umfasst die Systemebene-Unterstützungsschicht **44** Anwendungs-, Datenbank- und grafische Unterstützungselemente, welche es ermöglichen, dass die in jedem der Funktionsblöcke **46–49** durchgeführten Grafikaktivitäten mit-

einander integriert werden oder unter Verwendung gemeinsamer Datenbankstrukturen und grafischer Elemente, die auf der Systemunterstützungsschicht **44** erzeugt werden, entwickelt werden. Um diese Unterstützung auf Systemebene zur Verfügung zu stellen, kann die Systemunterstützungsschicht **44** einen Grafik-Editor **50** und eine Grafikobjekt-Datenbank **52** aufweisen. Der Grafik-Editor **50** kann dazu verwendet werden, grafische Elemente **54** und grafische Anzeigen **56** zu erzeugen, während die Grafikobjekt-Datenbank **52** die Elemente **54** und Anzeigen **56** in einem Speicher speichert, auf den von dem Editor **50** und von den verschiedenen Anwendungen in den Blöcken **46–49** zugegriffen werden kann. Die Datenbank **52** kann auch andere Objekte **58** und Datenstrukturen speichern, welche die grafischen Elemente **54** mit einzelnen Hardware- und Software-Elementen innerhalb der Anlagenbetriebsebene **40** verbinden. Zusätzlich kann die Datenbank **52** grafische Element- oder Anzeigeschablonen oder Grundformen ("primitives") speichern, die verwendet werden können, um weitere grafische Elemente oder Anzeigen zu erzeugen. Aus [Fig. 2](#) dürfte ersichtlich sein, dass die grafischen Anzeigeelemente **54**, Anzeigen **56** und andere Datenbankstrukturen **58** von jeglichen und allen der Funktionsblöcke **46–49** verwendet werden können, um Grafiken im Zusammenhang mit diesen Funktionsblöcken zu erzeugen und zu verwenden.

**[0050]** Allgemein gesprochen stellt der Systemebene-Unterstützungsblock **44** eine Vorgehensweise zum Integrieren der in der Verarbeitungsanlage **10** von [Fig. 1](#) verwendeten Grafiken in alle der Funktionsbereiche **46–49** zur Verfügung, um dadurch die Notwendigkeit zum wiederholten Erzeugen verschiedener grafischer Elemente für die gleiche Einrichtung der Anlage in verschiedenen funktionalen Zusammenhängen zu verringern oder zu beseitigen, und es für einen Anwender in jedem der Funktionsbereiche **46–49** einfach zu machen, mit Daten einzubinden, die der Einrichtung zugeordnet sind, welche in grafischen Ansichten dargestellt wird, die diesen Funktionsbereichen zugeordnet sind. Es dürfte verständlich sein, dass die Systemebene-Unterstützungsschicht **44** verwendet werden kann, um Grafik- und Datenbankunterstützung für mehrere Anwendungen in jedem der Funktionsbereiche **46–49**, für verschiedene Anwendungen in verschiedenen der Funktionsbereiche **46–49**, usw. zur Verfügung zu stellen.

**[0051]** Unter erneuter Bezugnahme auf [Fig. 1](#) kann der Systemebene-Unterstützungsblock **44** unter Verwendung einer zusätzlichen Workstation oder Anwenderschnittstelle **60** implementiert sein, die mit jeder der anderen Workstations **20–23** verbunden sein kann. Die Workstation **60** kann im Allgemeinen den Grafik-Editor **50** und die Datenbank **52** speichern und kann, falls gewünscht, die anderen Elemente **54**, **56** und **58** speichern. Zusätzlich kann die Workstation **60**

mit den Workstations **20–23** über den Datenbus **24**, über separate fest verdrahtete oder drahtlose Kommunikationsverbindungen (veranschaulicht durch gepunktete Linien in [Fig. 1](#)) oder auf jegliche andere gewünschte Weise kommunikationsmäßig verbunden sein. In der in [Fig. 1](#) veranschaulichten Konfiguration wird der Anzeige-Editor **50** von der Workstation **60** gespeichert und ausgeführt, um es einem Anwender zu ermöglichen, grafische Elemente zu erzeugen und solche Elemente in eine oder mehr grafische Anzeigen zu gruppieren, die vorliegend beide als Anzeigemodule bezeichnet werden. Diese Anzeigemodule können dann in der Datenbank **52** für den Zugriff und die Verwendung durch verschiedene der in [Fig. 2](#) veranschaulichten und auf den verschiedenen Workstations **20–23** implementierten Funktionsblöcke **46–49** gespeichert werden. Während zu Veranschaulichungszwecken die Funktionalität des Systemebene-Blocks **44** und der Funktionsebene-Blöcke **46–49** in [Fig. 1](#) als auf verschiedenen oder separaten Workstations **20–23** und **60** implementiert veranschaulicht ist, sollte es verständlich sein, dass jegliche oder alle der Anwendungen im Zusammenhang mit einem jeglichen dieser verschiedenen Blöcke auf der gleichen oder verschiedenen Workstations oder anderen Computern innerhalb oder in Zuordnung zu der Verarbeitungsanlage **10** implementiert sein könnte. Somit kann der Grafik-Editor **50** in einer jeglichen der anderen Workstations **20–23** oder in jeglichem anderen Computer, der der Anlage **10** zugeordnet ist, gespeichert sein und dort ausgeführt werden, und braucht kein Standalone- oder separater Computer zu sein.

**[0052]** Wie vorausgehend erörtert wurde, implementiert die Systemebene-Schicht **44** von [Fig. 2](#) Systemebene-Anzeige und Datenbankobjekte, die in einer Vielfalt der Funktionsumgebungen verwendet werden können, wie auch, um eine Anzeigefähigkeit auf einer höheren Ebene zur Verfügung zu stellen. Allgemein gesprochen können die auf der Systemebene **44** von [Fig. 2](#) erzeugten Anzeigeobjekte als grafische Elemente und grafische Anzeigen kategorisiert werden. Grafische Elemente sind im Allgemeinen Anzeigeobjekte mit Grundformen bzw. auf einer niedrigen Ebene, die einem bestimmten physischen Subjekt innerhalb der Anlage wie etwa einer Hardware-Vorrichtung wie einem Ventil, einem Sensor, einer Pumpe, einem Controller usw. zugeordnet sind. Grafische Anzeigen bestehen im Allgemeinen aus einer Gruppe von untereinander verbundenen grafischen Elementen und werden verwendet, um komplexere Hardwaregruppen innerhalb einer Anlage wie etwa eine Einheit, einen Bereich usw. darzustellen und zu modellieren, und umfassen Zwischenverbindungen zwischen verschiedenen Hardware-Einheiten. Zusätzlich können grafische Anzeigen Graphen, Diagramme und andere Daten umfassen, die von der Anlage, von anderen Anwendungen wie etwa Diagnose- und Geschäftsanwendungen, die auf den

Workstations **20–23** und **60** ablaufen, usw geliefert werden.

**[0053]** **Fig. 3** veranschaulicht allgemein die Entwicklung und Verwendung von grafischen Elementen und grafischen Anzeigen in zwei Umgebungen, in denen diese Elemente und Anzeigen vorliegen können, insbesondere eine Konfigurationsumgebung **70** und eine Laufzeitumgebung **72**. Allgemein gesprochen werden Anzeigeobjekte in Form von grafischen Elementen **74** (abgebildet als separate Elementobjekte **74a**, **74b** usw.) und grafische Anzeigen **76** (abgebildet als separate Anzeigeobjekte **76a**, **76b** usw.) in der Konfigurationsumgebung **70** beispielsweise unter Verwendung des Anzeige-Editors **50** erzeugt. Nach ihrer Erzeugung können die Objekte **74** und **76** in der Datenbank **52** gespeichert werden. Die Objekte **74** und **76** können als Klassenobjekte erzeugt werden, die vorliegend als Anzeigemodul-Klassenobjekte bezeichnet werden und ein generisches Objekt definieren, das nicht an bestimmte Hardware- oder Logikelemente innerhalb der Verarbeitungsanlage **10** gebunden oder mit diesen eingebunden ist. Klassenobjekte können jedoch verwendet werden, um Laufzeit-Grafikobjekte zu erzeugen, die die gleichen grundlegenden Eigenschaften wie die Klassenobjekte besitzen, die aber an eine bestimmte Hardware innerhalb der Verarbeitungsanlage **10** oder damit eingebunden sind. Allgemein gesprochen bleiben Klassenobjekte jedoch an die daraus erstellten Kinderobjekte gebunden, so dass Änderungen an den Klassenobjekten automatisch an die Kinderobjekte weitergegeben werden können, selbst wenn diese Kinderobjekte innerhalb einer Laufzeitumgebung erstellt werden.

**[0054]** Wie in **Fig. 3** veranschaulicht ist, umfasst jedes der Grafikelementobjekte **74** eine Anzahl von Komponenten, die das grafische Element in vielen verschiedenen Zusammenhängen nützlich machen. Insbesondere umfasst jedes grafische Element **74** eine oder mehr Visualisierungen **77**, eine jegliche Anzahl von Parametern oder Eigenschaften **78**, eine jegliche Anzahl von Aktionen oder Animationen **79**, die unter Verwendung von Scripts oder Triggern implementiert werden können, und Bindungen **80**. Allgemein gesprochen definiert jede Visualisierung **77** die visuellen Eigenschaften oder Elemente, die tatsächlich auf einem Anzeigebildschirm angezeigt werden sollen, wenn das grafische Element **74** in der Laufzeitumgebung **72** implementiert wird. Für gewöhnlich definieren Visualisierungen eine Darstellung einer physischen oder logischen Vorrichtung oder Gruppe von Vorrichtungen, obgleich eine Visualisierung andere Subjekte darstellen könnte. Visualisierungen **77** können in der Laufzeitumgebung **72** unter Verwendung einer jeglichen gewünschten Beschreibung oder Programmierparadigmas implementiert werden, das die Einzelheiten der grafischen Abbildung einer Subjekt angibt. Bei einer Ausführungsform können

die Visualisierungen **77** unter Verwendung von PGXML- oder Avalon-Steuerungen implementiert werden, bei denen es sich um allgemein bekannte Steuerungen handelt, die von Microsoft® zur Verfügung gestellt werden und die, weil sie objektbasiert sind, eine Vorgehensweise zur Verfügung stellen, um die Visualisierungen in Anzeigen vom Standard-Windows®-Type leicht implementierbar und zwischen Anzeigenumgebungen tragbar zu machen. Dieses Merkmal soll weiter unten ausführlicher erörtert werden und ist in der ebenfalls anhängigen Anwendung mit der Bezeichnung "Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface" (Anwaltsaktenzeichen Nr. 06005/41127) erläutert, auf die hiermit vollinhaltlich Bezug genommen wird.

**[0055]** Allgemein gesprochen definieren die Parameter und Eigenschaften **78** Variable oder andere Eigenschaften wie etwa statische oder veränderliche intrinsische Eigenschaften, die der durch die Visualisierung abgebildeten Subjekt zugeordnet sind, und diese Eigenschaften können durch den Erzeuger des Elementes **74** definiert werden. Ebenso definieren die Aktionen und Animationen **79** Routinen oder Programme (die als Scripts implementiert sein können, welche Transformationen an Eigenschaften durchführen, welche Umstände einer Prozess-Subjekt auf der Grundlage von Eigenschaftenwerten erfassen, usw.), Animationsroutinen, die jegliche Routinen umfassen können, welche die auf bzw. unter Verwendung der Visualisierung **77** durchzuführende grafische Visualisierung oder Verhaltensweisen ändern, wenn die Visualisierung **77** auf einem Anzeigebildschirm abgebildet ist, oder Routinen, die es einem Anwender ermöglichen, die Visualisierung **77** zu verwenden oder mit ihr zu interagieren, um eine Änderung in dem Prozess hervorzurufen, wie etwa ein Änderung einer Eingabe in den Prozess. Diese Aktionen und Animationen verleihen der Visualisierung **77** interessantere, leichter verständliche oder nützlichere Grafikeigenschaften und ermöglichen es dem Anwender, mit der Visualisierung **77** zu interagieren. In einem Fall können diese Aktionen oder Animationen die Form von Änderungen der Farbe, Größe (z.B. Höhe und Breite, Liniengröße, Fonts usw.) verschiedener Komponenten der Visualisierung, Einfärbungen, und Animationen wie etwa Änderungen der Farbe, Drehung, Änderungen der Größe und Skalierung, Kippen usw. annehmen. Diese Aktionen und Animationen verleihen dem grafischen Element **74** sowohl Grafikeigenschaften als auch Anwenderinteraktions-Eigenschaften. Die Bindungen **80**, bei denen es sich um statische oder feste Bindungen oder Bindungen handeln kann, die Aliase verwenden, definieren die Art und Weise, wie die Parameter oder Eigenschaften **78** an Daten, Etiketen oder andere Subjekte in der Laufzeitumgebung **72** gebunden werden sollen, wenn das grafische Element **74** als Teil einer Anzeige in der Laufzeitumgebung **72** implementiert ist.



Im Grunde erstellen die Bindungen **80** für jedes grafische Element **74** die Art und Weise, auf die das grafische Element **74** an eine oder mehr Subjekte oder Datenelemente gebunden ist, die anderweitig in der Betriebsumgebung definiert sind, und definieren dadurch eine Schnittstelle zwischen der tatsächlichen Laufzeitumgebung **72** und dem grafischen Element **74**.

**[0056]** Wie in [Fig. 3](#) veranschaulicht ist, umfasst jedes der grafischen Anzeigeobjekte **76** zahlreiche Komponenten, wie etwa einen Bezug auf oder eine Kopie von ein(em) oder mehr grafische(n) Elemente(n) **81**, Verbindungselemente(n) **82**, Aktionen und Animationen **83**, Eigenschaften **84** und Bindungen **85**. Allgemein gesprochen kann eine grafische Anzeige **76** eine Anzeige sein, welche die Interaktion von verschiedenen grafischen Elementen **81** darstellt, welche visuell mit Verbindungselementen **82** verbunden sein können, die Rohre, Leitungen, Förderbänder usw. repräsentieren. Solche Verbindungsobjekte sind in der US-Veröffentlichung Nr. 2004/0153804 beschrieben. Eine gepunktete Linie in [Fig. 3](#) veranschaulicht einen Bezug auf eines der grafischen Elemente **74** durch das grafische Anzeigeobjekt **76a**. Es dürfte verständlich sein, dass die grafische Anzeige **76**, die einen Bezug auf ein grafisches Element **74** darstellt, alle Eigenschaften, Parameter, Aktionen und Animationen usw. dieses grafischen Elementes **74** aufweist. Ähnlich wie die grafischen Elemente **74** kann jede grafische Anzeige **76** eine oder mehr zusätzliche Aktionen oder Animationen im Zusammenhang damit umfassen, die beispielsweise Animationen auf der Anzeige, Anwenderschnittstelle-Interaktionen, Datenmanipulationen usw. durchführen. Ebenso kann jede grafische Anzeige **76** eine jegliche Anzahl von Eigenschaften aufweisen, die der Anzeige zugeordnet ist, und für gewöhnlich definieren diese Eigenschaften Eigenschaften von Einheiten, Bereichen, oder anderen in der Anzeige abgebildeten Gruppen von Elementen. Natürlich definieren die Bindungen **85** die Art und Weise, auf die die grafische Anzeige **76** an ein oder mehr Subjekte oder Datenelemente gebunden ist, die anderweitig in der Betriebsumgebung definiert sind, und definieren somit eine Schnittstelle zwischen der tatsächlichen Laufzeitumgebung **72** und der grafischen Anzeige **76**.

**[0057]** Nach ihrer Erzeugung können die grafischen Elemente **74** und die grafischen Anzeigen **76** an die Laufzeitumgebung **72** gebunden und in ihr ausgeführt werden, beispielsweise auf einer jeglichen der Workstations **20–23** von [Fig. 1](#). Insbesondere nachdem ein grafisches Element **74** oder eine grafische Anzeige **76** als ein Klassenobjekt erzeugt und in der Datenbank **52** gespeichert wird, kann dieses Element bzw. diese Anzeige als ein tatsächliches Laufzeitobjekt erstellt werden und kann in der Laufzeitumgebung **72** ausgeführt werden. Wie durch den Block **86** veranschaulicht ist, füllt der Erstellungsvorgang die in

den Objekten **74** und **76** definierten Bindungen aus, was unter Verwendung von einer oder mehr Auflösungstabellen bewerkstelligt werden kann, die mit angemessenen variablen Namen, Etiketten, Aliasen usw. in der Verarbeitungsanlage oder in dem Prozesssteuerungssystem geladen werden können, um eine spezifische Verbindung zwischen tatsächlichen Subjekten in der Verarbeitungsanlage und den Grafikobjekten, die auf einer Anzeigevorrichtung in der Anlage **10** ablaufen, zur Verfügung zu stellen. Als Teil des Bindungsvorgangs verbinden sich die Objekte **74** und **76** mit Datenquellen in der Verarbeitungsanlage gemäß der Definition durch die Auflösungstabelle und erhalten dadurch Zugang zu der Anlage, so dass sie logisch und kommunikationsmäßig mit der Verarbeitungsanlage **10** verbunden sind.

**[0058]** Wie durch die Blöcke **87** veranschaulicht ist, kann ein Anzeigeelement **74** oder eine grafische Anzeige **76** in oder als Teil einer Anzahl von verschiedenen Funktionen in der Laufzeitumgebung **72** ausgeführt werden, einschließlich einer Konfigurationsanzeige, einer Steuerungsbediener-Anzeige, einer Wartungsanzeige und einer Simulationsanzeige, um nur einige zu nennen. Zusätzlich können die Anzeigeobjekte **74** und **76** bei der Durchführung von Systemebenenfunktionen verwendet werden, z.B. solchen, die Daten aus verschiedenen der in [Fig. 2](#) abgebildeten Funktionsebenen verwenden, einschließlich z.B. prädiktiver Steuerungsfunktionen oder prädiktiver Wartungsfunktionen, Systemebene-Fehlererfassung, Diagnose usw. Tatsächlich können die Anzeigen **76** nach ihrer Erzeugung in der Konfigurationsumgebung **70** und Speicherung in der Datenbank **52** für eine Anzahl von verschiedenen Aktivitäten verwendet werden. Ferner können die Anzeigeobjekte **74** und **76** auf jeglicher gewünschten Anzeige- oder Computer-Hardware ausgeführt werden, wie etwa einer Workstation **90**, einem Laptop-Computer **91**, einer tragbaren Vorrichtung **92** wie einem persönlichen Datenassistenten (PDA), einer Telefonvorrichtung usw., oder jeglichen anderen speziellen Anzeige **93**, wie etwa einer großformatigen Bildschirmanzeige mit mehreren Monitoren, usw. Falls gewünscht, kann eine einzelne grafische Anzeige **76** geschichtet sein, so dass sie eine oder mehr Ansichten wie etwa eine Konfigurationsansicht, eine Bedieneransicht, eine Wartungsansicht und eine Simulationsansicht umfasst. Als Alternative können separate grafische Anzeigen **76** so konfiguriert werden, dass sie diese separaten Ansichten unter Verwendung der gleichen oder ähnlicher grafischer Elemente **81** zur Verfügung stellen, um ein beständiges Aussehen und Gefühl über Anzeigen hinweg zur Verfügung zu stellen, die für diese verschiedene Funktionen erzeugt wurden.

**[0059]** Wie durch den Block **95** veranschaulicht ist, der an die Laufzeitumgebung **72** angeschlossen ist, kann ein grafisches Element **74** oder eine grafische Anzeige **76** kopiert oder erstellt und auf die Laufzeit-

maschine geladen werden. Allgemein gesprochen ist es wünschenswert, wenn das Anzeigeobjekt **74** oder **76** nur dann an die Laufzeitumgebung **72** gebunden wird, wenn es auf einer Laufzeitmaschine aufgerufen oder tatsächlich ausgeführt wird, was vorliegend als Laufzeitbindung bezeichnet wird. Mit anderen Worten, die Auflösungstabelle für jedes der erstellten Objekte wird nur dann ausgefüllt oder an die Laufzeitumgebung gebunden, wenn das Anzeigeobjekt tatsächlich abläuft oder in einem Laufzeitcomputer ausgeführt wird. Diese Prozedur stellt sicher, dass das Anzeigeobjekt einschließlich seiner Visualisierungen, Steuerungen, Scripts usw. nur ausführt und dadurch Verarbeitungsleistung verwendet, wenn die Visualisierung(en) des Objektes tatsächlich auf einem Anzeigebildschirm wiedergegeben wird/werden. Somit wird das Anzeigeobjekt bevorzugt nur an die Laufzeitumgebung **72** gebunden, wenn das Objekt tatsächlich auf einem Laufzeitcomputer abläuft, was bedeutet, dass die Anzeigeobjekte **74** und **76** intermittierend mit der Laufzeitumgebung **72** auf eine Weise verbunden sein können, welche durch die Aktivitäten der Anwender definiert ist, welche die von diesen Objekten erzeugten Visualisierungen betrachten. Insbesondere können diese Objekte an eine Laufzeitumgebung **72** zu den Zeiten gebunden sein, an denen ihre Betrachtung erforderlich ist, und können ungebunden oder freigesetzt werden, wenn sie nicht von einem Anwender betrachtet werden, wie etwa wenn ein Anwender einen Bildschirm, in dem diese Objekte eine Visualisierung zur Verfügung stellen, minimiert oder schließt.

**[0060]** Die Anzeigeobjekte **74** und **76** sind somit Objekte, die in einer Standalone-Umgebung, d.h. der Konfigurationsumgebung **70**, erzeugt werden können, die aber an andere Objekte oder Datenstrukturen, die in der Verarbeitungsanlagenumgebung oder jeder in der Umgebung der Verarbeitungsanlage ablaufenden Anwendung ablaufen, gebunden oder mit ihnen eingebunden sein, einschließlich z.B. Objekte, Datenstrukturen, Anwendungen, usw., die in jeglicher Steuerung, Simulation, Wartung, oder Konfigurationsumgebung definiert sind. Ferner können die Anzeigeobjekte **74** und **76** nach ihrer Erzeugung an physische oder logische Prozess-Subjekte gebunden werden, und zwar entweder direkt, über direkte Bezüge, Variable oder Etiketten, die in einer Auflösungstabelle definiert sind, oder indirekt durch die Verwendung von Alias-Namen, Variablen und Parametern, die entweder aufgelöst werden können, wenn die Anzeigeobjekt **74** oder **76** heruntergeladen oder in der Laufzeitumgebung **72** erstellt werden, oder in einigen Fällen, wenn das Anzeigeobjekt **74** oder **76** tatsächlich in der Laufzeitumgebung **72** läuft.

**[0061]** Der Anzeige-Editor **50** von **Fig. 3** kann es ermöglichen, dass die Anzeigeobjekte **74** und **76** auf verschiedenen Detailstufen erzeugt werden, um die Leichtigkeit der Verwendung und die Vielseitigkeit der

Anzeigeobjekte **74** und **76** zu verbessern. Beispielsweise können erst grafische Elemente **74** erzeugt werden, um die Eigenschaften und Operationen primitiverer physischer und logischer Subjekte zu definieren, und daraufhin können grafische Anzeigen **76** durch Verbinden von einem oder mehr grafischen Elementen **74** erzeugt werden, um höhere Ebene oder komplexere Anzeigen zu erzeugen, die kompliziertere physische oder logische Subjekte oder Gruppen von physischen oder logischen Subjekte darstellen. Natürlich können sowohl grafische Elemente **74** als auch grafische Anzeigen **76** in verschiedenen unterschiedlichen Kategorien gespeichert und auf sie zugegriffen werden, um die Erzeugung von Anzeigeobjekten einer höheren Ebene für den Anwender einfacher zu machen.

**[0062]** **Fig. 4** veranschaulicht eine beispielhafte Bildschirmanzeige **100**, die von dem Anzeige-Editor **50** erzeugt werden kann. Die Bildschirm **100**, der bei dem Vorgang der Erzeugung eines grafischen Elementes für eine Pumpe abgebildet ist, umfasst eine Haupteditier-Sektion **102**, eine Elementhierarchie-Sektion **104**, eine Eigenschaftendefinitions-Sektion **106** und eine Visualisierungs-Sektion **108**. Die Haupteditier-Sektion **102** stellt einen Arbeitsraum für den Anwender oder Designer zum Definieren oder Erzeugen einer Visualisierung für das grafische Element und somit zum Definieren der visuellen Eigenschaften des grafischen Elementes zur Verfügung, in diesem Fall einer Pumpe, die durch eine Visualisierung **109** veranschaulicht wird. Allgemein gesprochen stellt die Elementhierarchie-Sektion **104** unter Verwendung einer hierarchischen Ansicht oder einer Baumstruktur die Komponenten im Zusammenhang mit der Visualisierung **109** in der Haupteditier-Sektion **102** zur Verfügung. Bei dem Beispiel von **Fig. 4** zeigt die Hierarchiesektion **104**, dass die in der Haupteditier-Sektion **102** definierte Visualisierung **109** Grundformen oder Unter-elemente eines Kreises (als Circle1 bezeichnet) und zwei Rechtecke (als Rect1 bzw. Rect2 bezeichnet) umfasst. Während dies in **Fig. 4** nicht gezeigt ist, kann die Hierarchiesektion **104** Angaben von Animationen, Aktionen und anderen Anzeigemerkmale wie etwa Scripts, visuellen Triggern usw. umfassen, die für die Visualisierung **109** definiert wurden.

**[0063]** Die Eigenschaftendefinitions-Sektion **106** veranschaulicht alle Eigenschaften, einschließlich der intrinsischen Eigenschaften, die gegenwärtig für das grafische Element im Zusammenhang mit der in der Editiersektion **102** gezeigten Visualisierung **109** definiert wurden. Der beispielhafte Bildschirm **100** von **Fig. 4** veranschaulicht zwei Eigenschaften einschließlich einer IsOn-Eigenschaft, die definiert, ob die der Visualisierung **109** zugeordnete Pumpe an- oder abgeschaltet ist, und eine Drehzahleigenschaft, welche die Drehzahl der der Visualisierung **109** zugeordneten Pumpe definiert. Der Anwender oder Desi-



gner könnte zu dem grafischen Element andere Eigenschaften und Parameter hinzufügen, indem er die Namen, Typen und Bindungen anderer Variablen, Eigenschaften usw. in der Eigenschaftendefinitions-Sektion **106** definiert, um dadurch andere Aspekte des grafischen Elementes zu definieren. Die zwei in der Sektion **106** von [Fig. 4](#) gezeigten Eigenschaften sind als Boole'sche und Gleitkommavariablen aufgeführt. Es könnten jedoch andere Typen von Variablen statt dessen oder ebenso verwendet werden. Somit könnten beispielsweise die in der Sektion **106** definierten Eigenschaften Arrays, Tabellen, Aufzählungslisten oder jegliche andere Typen von Variablen oder Datenstrukturen sein.

**[0064]** Falls gewünscht, können dem in dem Bildschirm **100** definierten grafischen Element mehrere Visualisierungen zugeordnet sein. Die verschiedenen dieser Visualisierungen können in der Visualisierungs-Sektion **108** abgebildet werden und können individuell für die Anordnung in der Haupteditier-Sektion **102** ausgewählt werden. Beispielsweise sind zwei Visualisierungen **110A** und **110B** in der Visualisierungs-Sektion **108** von [Fig. 4](#) gezeigt, jedoch könnte jede andere Anzahl und jeder andere Typ von Visualisierungen für das zu erzeugende grafische Element definiert werden. Während des Vorgangs zur Erzeugung des grafischen Elementes kann jede Visualisierung in der Haupteditier-Sektion **102** angeordnet werden, wie etwa durch Rechtsklicken oder Doppelklick auf eine Angabe der Visualisierung in der Sektion **108**, Ziehen und Fallenlassen der Angabe der Visualisierung auf die Editiersektion **102**, usw. Nach ihrer Anordnung in der Haupteditier-Sektion **102** kann eine Visualisierung editiert werden, um ihre Anzeigeeigenschaften zu definieren oder neu zu definieren. Allgemein gesprochen wird eine der Visualisierungen als die Vorgabevisualisierung für das zu erzeugende grafische Element eingestellt oder definiert. Diese Vorgabeeinstellung kann auf eine bestimmte Weise angegeben werden, wie etwa dadurch, dass die Vorgabevisualisierung hervorgehoben wird, mit einer gepunkteten Linie in der Visualisierungs-Sektion **108** umgeben wird, wie für die Visualisierung **110A** in [Fig. 4](#) veranschaulicht ist, oder auf jegliche andere Weise.

**[0065]** Allgemein gesprochen können die von dem Element-Editor **50** erzeugten Visualisierungen aus einem oder mehr Kombinationen verschiedener Formen bestehen, die zusammen auf eine definierte Weise angeordnet oder gruppiert werden. Diese Visualisierungen können daher als Formkombinationen bezeichnet werden. Beispielsweise können Formkombinationen Kreise, Linien, Punkte, Polygone, Quadrate, Rechtecke, Dreiecke oder andere einfache grafische Formen umfassen. Bei einer Definition auf eine solche Weise können separate Aktionen oder Animationen auf jede verschiedenen Formen, die ein Formkombination bilden, angewendet oder dieser

zugeordnet werden. Natürlich können Formkombinationen sorgfältigere künstlerische Wiedergaben von Elementen umfassen. Um eine Formkombination zu definieren oder aufzubauen, kann ein Anwender oder Designer jegliche Anzahl von grundlegenden grafischen Elementen zu der Haupteditier-Sektion **102** hinzufügen und diese Grundformen auf jegliche beliebige Weise gruppieren. Nach ihrer Erzeugung definieren Formkombinationen die Visualisierung für ein tatsächliches Objekt, das in der Laufzeit als ein XAML-Objekt implementiert werden könnte und das auf einem Bildschirm angezeigt wird oder als eine Visualisierung angezeigt wird, wenn das grafische Element in einer Laufzeitumgebung verwendet wird. Die Formen oder Grundformen, welche die Formkombination bilden, können in einer Kombinationshierarchie in der Hierarchiesektion **104** veranschaulicht sein.

**[0066]** Die Erzeugung von mehreren Visualisierungen für ein einzelnes grafisches Element ermöglicht die Verwendung verschiedener Visualisierungen in verschiedenen Zusammenhängen oder für verschiedene Verwendungen, so dass für die verschiedenen Zwecke in der Anlage erzeugte Anzeigen aus dem gleichen grafischen Element erzeugt (d.h. aus dem gleichen grafischen Klassenobjekt erzeugt) werden können, während verschiedene Visualisierungen eines gemeinsamen Elementes angezeigt werden. Beispielsweise können verschiedene der Visualisierungen **110A**, **110B** usw. in verschiedenen Funktionszusammenhängen verwendet werden, so dass eine erste Visualisierung **110A** verwendet werden kann, wenn das grafische Element als Teil einer Steuerungsbediener-Anzeige verwendet wird, eine zweite Visualisierung **110B** kann verwendet werden, wenn das grafische Element als Teil einer Wartungsanzeige verwendet wird, und eine dritte Visualisierung (nicht gezeigt) kann verwendet werden, wenn das Pumpenelement in einer Simulationsanzeige verwendet wird. Andererseits können verschiedene Visualisierungen für Anzeigezwecke auf verschiedenen Typen von Anzeigevorrichtungen verwendet werden. Beispielsweise kann eine erste Visualisierung eines grafischen Elementes zur Verwendung auf einem typischen Computer oder einer typischen Workstation geeignet sein, während eine andere Visualisierung für eine vorteilhafte Verwendung auf einer tragbaren Vorrichtung wie einem PDA oder einer Telefonvorrichtung erzeugt werden, während wieder eine andere Visualisierung für großformatige Bildschirmanzeigen oder einen Mehrbildschirm-Monitor erzeugt werden kann. Es dürfte verständlich sein, dass verschiedene Visualisierungen für ein grafisches Element auf der Grundlage des Formats der Anzeige, auf der die Visualisierung angezeigt werden soll, zugeschnitten werden können, so dass eine erste Visualisierung erzeugt werden kann, die für einen größeren Bildschirm wie etwa einen typischen Computerbildschirm geeignet ist, und zweite Visualisie-

nung erzeugt werden kann, die eher auf einem beträchtlich kleineren Bildschirm wie etwa einem Anzeigebildschirm einer tragbaren Vorrichtung wie eines PDA oder einer Funktelefonvorrichtung geeignet ist. Somit können verschiedene Visualisierungen für das gleiche grafische Element in verschiedenen Laufzeitvorrichtungen verwendet werden.

**[0067]** Andererseits können verschiedene Visualisierungen zu dem grafischen Element an verschiedenen Zeitpunkten im Entwicklungszyklus des grafischen Elementes hinzugefügt werden. Beispielsweise kann ein Konfigurierungstechniker mit wenig Geschick im Zeichnen oder der Herstellung ansprechender Visualisierungen beim Erzeugen des grafischen Elementes zuerst eine erste, primitive Visualisierung herstellen, die vorläufig und grundlegender Art ist (wie eine Zeichnung vom Typ eines Strichmännchens), und diese Visualisierung zu diesem Zeitpunkt als Teil des grafischen Elementes abspeichern. An einem späteren Datum oder Zeitpunkt kann ein geschickter Zeichner oder Künstler eine detailliertere und künstlerisch ansprechendere Visualisierung erzeugen und diese zweite Visualisierung zu dem grafischen Element als eine alternative Visualisierung hinzufügen. Falls gewünscht, kann die zweite Visualisierung zu diesem Zeitpunkt als die Vorgabevisualisierung eingestellt werden, und diese zweite Visualisierung kann durch alle Laufzeit-Grafikelemente, die aus dem Master- oder Klassen-Grafikelement erstellt wurden, weiter gegeben werden, so dass diese zweite Visualisierung in den Laufzeitanzeigen erscheint oder zur Verwendung in ihnen verfügbar ist.

**[0068]** Bei einem anderen Beispiel können verschiedene Visualisierungen für das gleiche grafische Element zur Verfügung gestellt werden, um verschiedene Themen, Grafikstandards, Normen oder Stile von Anzeigen zu unterstützen. Bekanntermaßen verwenden verschiedene Industrien für gewöhnlich verschiedene grafische Normen oder Grafikstandards für die Darstellung von Pumpen, Ventilen, Sensoren und anderen Verarbeitungsanlagensubjekten. So verwendet die Öl- und Gasindustrie andere Grafikstandards als die pharmazeutische Industrie. Unter Verwendung des Element-Editors **50** ist es möglich, verschiedene Visualisierungen für jedes grafische Element zur Verfügung zu stellen, um mehrere Grafikstandards oder Normen, wie etwa die Öl- und Gasnorm und die pharmazeutische Norm zu unterstützen. Auf diese Weise kann das gleiche grafische Element in grafischen Anzeigen verwendet werden, die erzeugt werden, um verschiedene Standards oder Normen oder Themen von Anzeigen zu unterstützen, z.B. Anzeigen, die in verschiedenen Industrien akzeptabel oder nützlich sind. Ferner können verschiedene Grafikstile wie etwa künstlerische Stile in verschiedenen Visualisierungen verwendet werden, um die Erzeugung von Anzeigen verschiedener artistischer Typen durch einen Anwender zu ermöglichen.

Natürlich sind dies nur einige Beispiele für die Vorteile des Vorsehens von mehreren Visualisierungen für das gleiche grafische Element, und es gibt auch andere Verwendungen für diese mehreren Visualisierungen.

**[0069]** Falls gewünscht, kann das grafische Element ein oder mehr grafische Verhaltensweisen besitzen, die damit definiert oder ihm zugeordnet sind. Insbesondere kann ein Designer oder Erzeuger Animationen wie etwa Drehungen, lineare Verschiebungen, Hintergrundänderungen, Farbänderungen, Größenänderungen, Farbgradient-Animationen, Undurchsichtigkeitsanimationen, fontcharacteristische Animationen, Videos und Videomerkmale, wie etwa Start/Stop-Merkmale, zweidimensionale oder dreidimensionale Änderungen usw. für jede Visualisierung des grafischen Elementes definieren, wenn die Visualisierung auf einem Bildschirm angezeigt wird. Um diese dynamische Verhaltensweise hinzuzufügen, kann der Anwender ein grafisches Element auswählen und wählen, der Visualisierung eine Animation (auch als Animationsroutine bezeichnet) hinzuzufügen. An diesem Punkt kann der Anwender Konfigurationsinformationen eingeben, die sich auf die ausgewählte Animation beziehen, oder er kann die gewünschte Verhaltensweise unter Verwendung von Scripts definieren, die z.B. unter Verwendung von Dialogboxen, Ausdruckeditoren usw. eingegeben werden können. Nach ihrer Definition können diese Scripts in der Hierarchie der Hierarchiesektion **104** erscheinen. Im Grunde ist ein Script ein Programm oder eine Routine, die als Teil des Anzeigeelementes abläuft oder ausgeführt wird, wenn die Visualisierung für das Anzeigeelement während einer Laufzeitaktivität betrachtet oder auf einem Bildschirm präsentiert wird. Obgleich dies nicht nötig ist, können diese Verhaltensweisen oder Scripts an eines oder mehr der für das grafische Element definierten Eigenschaften oder Parameter gebunden sein und auf sie einwirken. Beispielsweise kann ein Script einem grafischen Element zugeordnet sein, um eine Farbe in der Visualisierung des grafischen Elementes auf der Grundlage des Wertes von einer der Eigenschaften zu ändern, die in der Eigenschaftensektion **106** des grafischen Elementes definiert sind. Beispielsweise kann sich die Farbe der Pumpenvisualisierung **109** von gelb zu grün ändern, wenn die IsOn-Eigenschaft (die an die Laufzeitumgebung gebunden ist) True [Zutreffend] ist, d.h. wenn die Pumpe angeschaltet ist. Als ein weiteres Beispiel kann ein Script für das Pumpenelement definiert werden, das den Wert der Drehzahleigenschaft der Pumpe mit einem Sollwert vergleicht und, wenn der Wert der Drehzahleigenschaft über einem bestimmten Niveau liegt, eine grafische Animation einer bestimmten Art in der Visualisierung **109** veranlasst. Diese Animation kann beispielsweise das Rotfärben der Pumpe, Drehen der Pumpe, Darstellen eines drehenden Motors in der Pumpe, Pulsieren oder Vibrierenlassen der Pumpenvisualisierung usw.

umfassen.

**[0070]** Die Scripts, die für eine Visualisierung definiert sind, um beispielsweise eine Animation oder andere Verhaltensweisen für die Visualisierung zur Verfügung stellen, können so entworfen werden, dass sie auf einzelne der Formen oder Grundformen in einer Formkombination, welche die Visualisierung darstellen, oder auf mehrere der Formen in den Formkombinationen einwirken. Da die Scripts auf die intrinsischen Eigenschaften, die an tatsächliche physische Elemente in der Laufzeitumgebung gebunden sind, einwirken oder sie verwenden, ermöglichen es diese Scripts, dass sich die Visualisierung auf der Grundlage von tatsächlichen Operationen der Anlage ändert, oder auch auf der Grundlage von Daten aus anderen Bereichen der Verarbeitungsanlage, welche die Eigenschaften einer tatsächlichen, dem grafischen Element zugeordneten Subjekt wiedergeben.

**[0071]** Es dürfte verständlich sein, dass verschiedene Typen von grafischen Elementen verschiedene Graphiken aufweisen können, die zur Verwendung mit ihnen geeignet sein können. Somit sind die Verhaltensweisen, die für ein grafisches Element zur Verfügung gestellt werden können, nicht durch die vorliegend gegebenen Beispiele beschränkt. Beispielsweise könnten grafische Elemente für drehende Einrichtung Scripts umfassen, die Vibrationsgraphiken, Bewegung, Farbänderungen usw. vorsehen, während grafische Elemente für Einrichtungen wie etwa Sensoren Scripts aufweisen können, die eine Überschreitungs- oder Unterschreitungsbedingung, einen Kalibrierungsbedarfzustand usw. darstellen. Natürlich können jegliche Eigenschaften für grafische Elemente verwendet oder definiert werden, und diese Eigenschaften beruhen im Allgemeinen auf dem Typ der darzustellenden Subjekt. Es dürfte auch verständlich sein, dass verschiedene Scripts für jede der verschiedenen Visualisierungen zur Verfügung gestellt werden können, um verschiedene Verhaltensweisen für die verschiedenen Visualisierungen vorzusehen. Andererseits können einige Scripts verwendet werden, um eine Verhaltensweise für jede der Visualisierungen im Zusammenhang mit einem grafischen Element zur Verfügung zu stellen.

**[0072]** Zusätzlich können Scripts oder andere Routinen einem grafischen Element zugeordnet sein, um bestimmte Bedingungen bezüglich der zugeordneten physischen Subjekt auf der Grundlage von einem oder mehr der Parameter für das grafische Element zu erfassen. Diese Bedingungen können die Erfassung von Zuständen des Subjektes, einschließlich eines physischen Zustandes, eines Zustandes wie eines Kommunikationszustandes, eines Vorrichtungs Zustands, ein Wertezustands usw. im Zusammenhang mit der Subjekt umfassen. Alle diese erfassten Bedingungen oder Zustände können in der Grafik unter Verwendung einer Animation oder einer anderen Ak-

tion oder Verhaltensweise, die an den von dem Script erzeugten Zustand oder Wert gebunden ist, reflektiert sein. Beispielsweise kann ein Script für ein grafisches Pumpenelement zur Verfügung gestellt werden, das einen Überhitzungszustand der zugeordneten Pumpe auf der Grundlage eines oder mehrerer Parameter für dieses Element erfasst, d.h. eines oder mehrerer der in der Eigenschaftensektion **106** von **Fig. 4** definierten oder dargelegten Parameter. Als ein weiteres Beispiel kann ein Script für ein grafisches Element zur Verfügung gestellt werden, um eine exzessive Schwingung oder einen jeglichen anderen Zustand der Pumpe zu erfassen. Falls ein Zustand wie etwa Überhitzung oder exzessive Schwingung erfasst wird, kann eine Animation oder eine andere dem grafischen Element zugeordnete Aktion auf den Zustand einwirken, um eine grafische Angabe des Zustandes in oder auf der Visualisierung des grafischen Elementes zur Verfügung zu stellen.

**[0073]** **Fig. 5** veranschaulicht eine andere beispielhafte Bildschirmanzeige **112**, die von dem Anzeige-Editor **50** hergestellt werden kann. Ähnlich wie bei dem Bildschirm **100** von **Fig. 4** umfasst die Bildschirmanzeige **112** eine Haupteditier-Sektion **114**, eine Palettenansicht **116**, ein Hierarchieansicht **118** und eine Eigenschaftenansicht **120**. Eine erste Visualisierung eines Reaktorelementes **122** ist in der Haupteditier-Sektion **114** abgebildet. Wie in der Hierarchieansicht **118** gezeigt ist, ist die Bezeichnung des Elementes Reactor1, und dieses Element umfasst drei Visualisierungen mit der Bezeichnung Visual1 (die Vorgabevisualisierung), Visual1 und Visual3. Wie unter der Visual1-Überschrift in der Hierarchieansicht **118** angegeben ist, besteht die erste Visualisierung aus einem Canvas-Hintergrund mit einem Rechteckelement und einem Ellipsenelement. Die gegenwärtig definierten Eigenschaften, in diesem Fall ein Name, eine Höhe und eine Breite der Visualisierung, sind in der Eigenschaftenansicht **120** aufgeführt. Wenn eine Visualisierung in der Hierarchieansicht **118** ausgewählt wird, wird jede der Kindergrundformen oder Elemente, die der Visualisierung zugeordnet sind, in der Editieransicht **114** gezeigt, und die Eigenschaften des gegenwärtig ausgewählten Elementes sind in der Eigenschaftenansicht **120** angezeigt.

**[0074]** In dem Bildschirm **112** umfasst die Palettenansicht **116** eine Anzahl von grundlegenden Elementen, die verwendet werden können, um eine Visualisierung zu erzeugen. Beispielsweise umfasst die Palettenansicht **116** eine Gruppe von grundlegenden UI (Anwenderschnittstellen)-Elementen wie etwa Knöpfen, Textboxen, Schiebern, Drehknöpfen usw., ein Gruppe von grundlegenden Tafeln, und eine Gruppe von grundlegenden Formen. Die definierten Tafeln können eine Canvas-Tafel umfassen, die einen Bereich definiert, in dem der Anwender ausdrücklich Elemente durch Koordinaten relativ zu dem Canvas-Bereich anordnen kann, eine Dock-Tafel, die ei-

nen Bereich definiert, in dem der Anwender Elemente entweder horizontal oder vertikal aufeinander bezogen anordnen kann, und eine Flow-Tafel, die verwendet werden kann, um ihre Inhalte innerhalb eines Flow-Tafelbereichs aufzubrechen, zu verschlingen und auszurichten, wobei eine Fließrichtung angegeben ist. Fließrichtungen können beispielsweise eine beliebige Kombination von oben, rechts, links und unten sein, wie etwa von links nach rechts und von oben nach unten, oder von rechts nach links, oben nach unten, usw. Ferner können die grundlegenden Formen in der Palettenansicht **116** ISA (Instrument Society of America)-Symbole, Sendersymbole, Ventilsymbole, PI&D-Diagrammsymbole oder andere Steuersymbole usw. oder jegliche andere gewünschte Formen aufweisen, die alle zum Aufbauen eines grafischen Elementes verwendet werden können.

[0075] Natürlich können andere grundlegende Elemente, wie etwa grundlegende Steuerelemente, Vorrichtungen usw. in der Palettenansicht **116** als Grundformen zur Verfügung gestellt werden, um zum Erzeugen von Visualisierungen für ein zu definierendes grafisches Element verwendet zu werden. Die Palettenansicht **116** kann auch eine Liste von anwenderdefinierten Kategorien oder Elementen zur Verfügung stellen, die erzeugt werden können, um es Anwendern zu ermöglichen, jegliche andere nützliche Formen aus der Palettenansicht **116** in die Editieransicht **114** zu ziehen. Falls gewünscht, kann der Zugriff auf solche anwenderdefinierte Kategorien auf die Anwender beschränkt sein, die sie erzeugen, und diese Kategorien und Elemente können in einer Anwenderpräferenz-Datei gespeichert werden, die einem bestimmten Anwender zugeordnet ist. Eingebaute Kategorien und Palettengegenstände können jedoch in der Datenbank gespeichert und global für alle Anwender verfügbar gemacht werden. Auf jeden Fall kann die Palettenansicht **116** verwendet werden, um eine Anzeige von oder einen Zugriff auf Bibliotheken mit Grundformenelementen zur Verfügung zu stellen, die verwendet werden können, um ein grafisches Element zusammenzustellen, und diese Bibliotheken können nach Wunsch verriegelt, als Versionen abgelegt, auf bestimmte Anwender beschränkt werden, usw.

[0076] Wie weiter oben erwähnt wurde, können jeglichen der Visualisierungen für ein grafisches Element Animationen und/oder Aktionen zugeordnet sein, und solche Animationen oder Aktionen können in einer Aktion/Animation-Ansicht **123** des Bildschirms **112** gezeigt werden. Wenn Visualisierungen Animationen oder Aktionen umfassen, können diese Animationen oder Aktionen auch in der Hierarchie mit besonderen Symbolen wie etwa Sternen usw. angegeben sein. Wenn irgend welche Aktionen oder Animationen, die für eine Visualisierung oder ein Unter-element einer Visualisierung definiert sind, in der Hierarchieansicht **118** ausgewählt werden, werden sie in

der Aktion/Animation-Ansicht **123** gezeigt. Aktionen oder Animationen können durch das Definieren solcher Aktionen oder Animationen in der Ansicht **123** oder durch das Hinzufügen solcher Aktionen oder Animationen zu der Hierarchieansicht **118** erzeugt und zugeteilt werden. Wenn ein Anwender eine Aktion oder Animation zu erzeugen oder zu editieren wünscht, kann der Editor **50** eine Dialog- oder Editierbox zur Verfügung stellen, um es zu ermöglichen, dass dieses Merkmal vollständig spezifiziert oder definiert wird. Natürlich können Aktionen oder Animationen unter Verwendung von Scripts, visuellen Triggern oder anderen Programmen definiert werden.

[0077] Während der Verwendung des Bildschirms **112** kann ein Anwender oder Designer eine Visualisierung eines Elementes durch Ziehen und Fallenlassen oder anderweitiges Auswählen verschiedener Gegenstände in der Palettenansicht **116** erzeugen und durch Anordnen dieser Gegenstände in der Editieransicht **114**, um die gewünschte Visualisierung zu erzeugen. Eine oder mehr Werkzeugeisten **124** können verwendet werden, um jegliche Standardeditierfunktionen zur Verfügung zu stellen wie etwa das Hinzufügen neuer Visualisierungen oder Animationen, Löschen, Entfernen, Editieren von Animationen, Anordnen von Grundformen oder Elementen von z.B. hinten nach vorne, Vorsehen von Verbindungselementen zum Zeigen oder Freigeben von Verbindungen, die an der zu erzeugenden Visualisierung vorgenommen werden sollen, Gruppieren von verschiedenen Grundformen so, dass sie ihre Position relativ zu einander beibehalten, Hinzufügen von statischen Elementen wie Linien und Text, usw.

[0078] Wie in [Fig. 6](#) veranschaulicht ist, kann ein grafisches Element **130** mehrere ihm zugeordnete Visualisierungen **132** oder visuelle Darstellungen haben, wobei angemerkt wird, dass die verschiedenen Visualisierungen **132** zu verschiedenen Zeitpunkten oder in verschiedenen Anzeigen verwendet werden können, in denen das grafische Element **130** verwendet wird. Wie weiter oben bereits erläutert wurde, kann jede der Visualisierungen **132** aus einer jeglichen Anzahl von Grundformen **134** sowie Animationen und Aktionen **136** bestehen. Ferner kann das grafische Element **130** eine jegliche Anzahl von Eigenschaften oder Parametern **138** besitzen, welche in die Aktionen und Animationen **136** eingebunden oder von ihnen verwendet werden können, um Änderungen an den Visualisierungen **132** vorzunehmen und visuelle Trigger zu betätigen, die den Visualisierungen **132** zugeordnet sind. Ferner können die Visualisierungen **132** oder individuelle Grundformen, aus denen eine Visualisierung **132** besteht, Aktionen aufweisen, die für vordefinierte Ereignisse definiert sind, beispielsweise Mouse-over-Ereignisse, Mausclickereignisse, usw. Diese Aktionen (die auch als Routinen bezeichnet werden) ermöglichen das Einstellen oder Definieren von Ereignishandlern, um die Verhaltens-



weise des grafischen Elementes **130** noch weiter zu personalisieren und einem Anwender eine Interaktion mit der Visualisierung **132** zu ermöglichen, um beispielsweise eine Änderung in der Laufzeitumgebung zu veranlassen. Insbesondere kann der Anwender durch das Eingeben von Werten oder anderen Zahlen oder Informationen, Verschieben eines Elementes auf der Visualisierung, wie etwa einer Schiebeleiste, oder durch Vornehmen einer anderen Aktion mit der Visualisierung **132** interagieren, um beispielsweise eine Eigenschaft in der Visualisierung zu ändern. Diese Visualisierungseigenschaft kann über ein Script oder direkt mit einer Prozesseingabe wie etwa einer Prozesslaufzeit-Variable eingebunden werden, um eine Änderung dieser Variablen zu veranlassen. Insbesondere kann die Visualisierungseigenschaft mit einer für das grafische Element definierten Eigenschaft verbunden sein, die wiederum an eine Prozesseingabe gebunden sein kann. Auf diese Weise kann ein Anwender mit der Visualisierung über eine Aktion oder Animationsroutine interagieren, um eine Änderung zu veranlassen oder eine Einage an die Prozessumgebung oder eine andere Laufzeitumgebung zur Verfügung zu stellen, wie etwa eine Simulationsumgebung.

**[0079]** Falls gewünscht, kann eine Visualisierung **132** mit dem grafischen Element **130** verbunden werden, indem dies entweder von einem Anwender spezifiziert wird oder über ein Programm spezifiziert wird. Insbesondere kann das grafische Element **130** eine Aufzählungseigenschaft aufweisen, die es ermöglicht, dass die Visualisierung über ein Programm geändert wird, indem beispielsweise der Hersteller des Elementes durch das grafische Element angegeben wird, oder andere Parametern im Zusammenhang mit dem grafischen Element **130**, wie etwa den Zustand der Vorrichtung, die dem grafischen Element zugeordnet ist.

**[0080]** Wie weiter oben erwähnt wurde, liefert oder zeigt die Eigenschaftenansicht **120** von [Fig. 5](#) die Eigenschaften und Ereignisse, die für den ausgewählten Gegenstand in der hierarchischen Ansicht **118** und somit für den in der Haupteditieransicht **114** abgebildeten Gegenstand definiert sind. Ein Anwender kann zwischen Eigenschaften, intrinsischen Eigenschaften und Ereignissen in der Eigenschaftenansicht **120** hin und her schalten, z.B. durch die Verwendung von Werkzeugleistenknöpfen. [Fig. 7](#) zeigt eine Eigenschaftenansicht **120A**, in der die intrinsischen Eigenschaften gezeigt sind, in diesem Fall mit einer IsOn-Eigenschaft und einer Drehzahleigenschaft. In diesem Fall zeigt die Eigenschaftenansicht **120A** den Datentyp dieser Variablen und jegliche Vorgabeeinstellungen für diese Variablen. Wie weiter oben erwähnt wurde, kann der Anwender an dieser Eigenschaftenliste Hinzufügungen, Löschungen oder Editierungen vornehmen, um Eigenschaften mit jeglichem gewünschten Datentyp zu definieren, ein-

schließlich Aufzählungen, Tabellenstrukturen usw. Falls gewünscht, kann die Eigenschaftenansicht **120** auch jegliche Laufzeitbindungen zeigen, die für diese Eigenschaften definiert sind, und diese Laufzeitbindungen können feste Variable oder Etiketten sein, oder können Etiketten unter Verwendung von Aliases sein, die in der Laufzeit oder durch Herunterladen des grafischen Elementes auf ein Laufzeitmaschine ausgefüllt werden.

**[0081]** Die Werte der Eigenschaften, die für ein grafisches Element definiert sind, können als Triggerbedingungen für Animationen, Aktionen usw. verwendet werden, und diese Verhaltensweisen können durch ein oder mehr Scripts definiert sein, die dem grafischen Element zugeordnet sind. Falls beispielsweise ein Motor eine On-Eigenschaft aufweist, die zutreffend ist, kann eine Animation des in Betrieb befindlichen Motors in der grafischen Anzeige ausgelöst werden. Ferner kann eine Grundformseigenschaft, beispielsweise eine Rechteck-Ausfülleigenschaft einer Grundform mit der Form eines Rechtecks an eine Grafikelement-Eigenschaft gebunden werden, so dass sich Änderungen in der Grafikelement-Eigenschaft auf die Grundformseigenschaft auswirken. Ebenso können Änderungen an einer Visualisierung an eine Eigenschaft gebunden sein, so dass Änderungen an der Visualisierung Änderungen an dem Eigenschaftenwert verursachen können.

**[0082]** Falls gewünscht, kann der Editor **50** einen Anwender in die Lage versetzen, eine oder mehr Transformationsfunktionen zu spezifizieren, um eine wünschenswertere Bindung zwischen Grafikelement-Eigenschaften und Animationen oder Aktionen zur Verfügung zu stellen. Beispielsweise kann ein Anwender den Wunsch haben, eine Grafikelementeigenschaft TankLevel (die einen Fluidspiegel in einem Tank definiert) an eine Grundformseigenschaft des Ausfüllens eines Rechtecks zu binden, um dadurch den Tankpegel mit einer Füllungsfarbe für eine Grundform, die als Teil der Visualisierung definiert ist, grafisch zu veranschaulichen. In diesem Fall kann der Anwender jedoch eine Transformationsfunktion definieren, welche die Eigenschaft (TankLevel) in eine aufgelistete Gruppe oder Bedingung transformiert, so dass die Rechteckfüllfarbe auf Grün gesetzt wird, wenn der Tankpegel zwischen einer ersten und einer zweiten Ebene liegt, die Rechteckfüllfarbe auf Gelb gesetzt wird, wenn der Tankpegel zwischen der zweiten und einer dritten Ebene liegt, und die Rechteckfüllfarbe auf Rot gesetzt wird, wenn der Tankpegel über dem dritten Pegel liegt. Diese Transformationsfunktion kann als ein Script oder als jegliches andere Programm definiert sein, das mit dem grafischen Element ausgeführt wird und dazu verwendet werden kann, eine gewünschte Eigenschaftentransformation zu verursachen, wie etwa das Ändern eines Eigenschaftenwertes zu einer Länge, einem Fontnamen, einem lokalisierten String, einer Zeitdauer, einer Dre-

hung, einer Farbabstufung, einer Undurchsichtigkeit, einem Pinselmuster, usw. Ebenso können Transformationsroutinen Eingänge von einem Anwender über die Visualisierung zu jeglichen gewünschten Eigenschaftswerten transformieren.

**[0083]** Als ein anderes Beispiel können Grafikelement-Eigenschaften oder Parameter unter Verwendung von Transformationsfunktionen an externe Referenzen gebunden sein. Hierbei kann es der Editor **50** einem Anwender ermöglichen, eine oder mehr Transformationsfunktionen zu spezifizieren, die automatisch angewendet werden, wenn das grafische Element in einer Laufzeitumgebung verwendet wird, um Quellwerte (Laufzeitvariable) in Sollwerte (grafische Variable) zu konvertieren, oder umgekehrt. Beispielsweise könnte eine Geschwindigkeits-Sollvariable, die an eine Datenquelle gebunden sein könnte, welche eine Geschwindigkeit bzw. Drehzahl als einen numerischen Wert zur Verfügung stellt, in eine Variable vom String-Typ konvertiert werden, wie etwa einen String, der entweder "langsam", "moderat" und "schnell" auf der Grundlage des Wertes der Quellvariablen angibt. Solche Transformationsfunktionen können auch verwendet werden, um Einheiten zu konvertieren oder andere Transformationen zur Verfügung zu stellen. Auf jeden Fall können diese Transformationsfunktionen als Scripts oder andere Programme implementiert werden und können für jegliche Zwecke verwendet werden, um Aktionen oder Animationen zur Verfügung zu stellen oder Anzeigeeigenschaften für ein grafisches Element anderweitig zu beeinflussen.

**[0084]** Natürlich können jegliche gewünschte Eigenschaften für verschiedene Typen von grafischen Elementen definiert werden, die für verschiedene physische Elemente in einer Prozesssteuerung erzeugt wurden. Beispielsweise kann ein Pumpenelement einen Pumpenzustand (z.B. Ein oder Aus), eine Eingangsdruck-Eigenschaft, eine Ausgangsdruck-Eigenschaft, eine Einlauf-Eigenschaft und eine Auslauf-Eigenschaft aufweisen. Ebenso kann ein mit einem Betätigungsglied verwendetes Regelventil beispielsweise eine Nameneigenschaft, eine Eingangsdichte-Eigenschaft, eine Ausgangsdichte-Eigenschaft, eine Einlauf-Eigenschaft, eine Auslauf-Eigenschaft, eine Eingangsdruck-Eigenschaft, eine Ausgangsdruck-Eigenschaft, eine Eingangstemperatur-Eigenschaft, eine Ausgangstemperatur-Eigenschaft, eine Ventilstellungseigenschaft, eine Ventil-geöffnet- und Ventil-geschlossen-Eigenschaft (die definieren können, ob das Ventil vollständig geöffnet oder geschlossen ist), einen Sollwert, einen Prozesswert, einen Ventiltyp wie etwa linear, schnellöffnend, gleichprozentig usw. aufweisen. Diese Liste soll natürlich nicht als umfassend angesehen werden. Darüber hinaus können jegliche dieser Eigenschaften mit Animationen oder Aktionen für grafische Elemente verbunden werden.

**[0085]** **Fig. 8** stellt eine zweite Eigenschaftenansicht **120B** dar, die verwendet werden kann, um die Animationen und Bindungen im Zusammenhang mit Grafikelement-Eigenschaften für ein ausgewähltes Element (in diesem Fall das Reactor1-Element im Allgemeinen) in der hierarchischen Ansicht **118** (**Fig. 5**) zu betrachten. Die Eigenschaftenansicht **120B** von **Fig. 8** veranschaulicht, in welchem Zusammenhang IsOn-Eigenschaft mit verschiedenen Komponenten der Visualisierung für das grafische Element steht. Insbesondere wie durch den Tabelleneintrag **140** angegeben ist, gibt es eine Animation, die mit dem Hintergrund der Visualisierung im Zusammenhang steht und auf der Grundlage des Wertes der IsOn-Eigenschaft arbeitet. Ein Tabelleneintrag **142** zeigt die Bindungen zwischen der IsOn-Eigenschaft und kann dazu verwendet werden, auf Animationen, Scripts und Transformationen zuzugreifen, welche die IsOn-Eigenschaft mit dem Hintergrundelement oder der Grundform in der Visualisierung in Beziehung setzen. In diesem Fall kann der Anwender durch Wählen der Knöpfe in den Tabelleneintragboxen **140** und **142** zusätzliche Informationen erhalten. Beispielsweise kann der Anwender durch Wählen des Knopfes in dem Tabelleneintrag **140** auf eine Animation zugreifen, die veranlasst, dass sich die Farbe des Hintergrundes der Visualisierung auf der Grundlage des Wertes für den IsOn-Parameter ändert. Das Klicken auf einen solchen Animationsknopf kann auch den Editor **50** veranlassen, eine Eigenschaftsanimation-Dialogbox zu öffnen, die ein Verwalten und Konfigurieren der Animationen ermöglicht. Ferner kann ein Binden durch Klicken auf einen Bindungsknopf in der Box **142** durchgeführt und verwaltet werden, das den Editor **50** veranlasst, eine Dialogbox zu öffnen, um das Erzeugen, Betrachten und Editieren von Bindungen zu ermöglichen. Natürlich könnten ähnliche Bildschirme für andere Eigenschaften des grafischen Elementes zur Verfügung gestellt werden, und die Liste von Eigenschaften, die in dem Eigenschaften-Bildschirm **120B** gezeigt sind, hängt von dem gegenwärtig ausgewählten Gegenstand in der Hierarchieansicht **118** ab.

**[0086]** Unter erneuter Bezugnahme auf **Fig. 5**, wenn eine Visualisierung in der Hierarchieansicht **118** ausgewählt wird, zeigt die Eigenschaftenansicht **120** die Visualisierungseigenschaften an wie etwa einen Namen, eine I.D., die eine eindeutige Identifikation für die Visualisierung zur Verfügung stellt, eine Beschreibung, und ob diese Visualisierung als die Vorgabevisualisierung für das grafische Element gestellt ist. Die Hierarchieansicht **118** oder die Eigenschaftenansicht **120** können auch anzeigen, ob die Visualisierung verriegelt ist, d.h. ob sie modifiziert werden kann.

**[0087]** Wenn zusätzlich ein Verbindungselement auf der Hierarchiesektion **118** ausgewählt ist, zeigt die Eigenschaftenansicht **120** die Eigenschaften die-



ses Verbindungselementes, die eine Angabe des Verbindertyps (z.B. eine Fluidleitung, eine elektrische Leitung, eine Gasleitung, eine kreisförmige oder quadratische Rohrleitung, ein Förderband usw.), ob die Verbindung in die oder aus der Vorrichtung führt (d.h. die Strömungsrichtung von Material in Bezug auf das durch das grafische Element dargestellte Subjekt), die minimale und maximale Anzahl von erforderlichen oder zugelassenen Verbindungen, die Breite und Höhe der Verbindung; die Position der Verbindung in Bezug auf das grafische Element, wie etwa oben links, usw. umfassen kann.

**[0088]** Wenn eine Grundform in der Hierarchieansicht **118** ausgewählt wird, zeigt die Eigenschaftenansicht **120** die Eigenschaften dieser Grundform an, die beispielsweise eine I.D., eine Kategoriebeschreibung, ob die Grundform wählbar und/oder sichtbar ist, eine Größe und Länge oder eine andere Definition der Form oder Art der Grundform, eine Hintergrundfarbe und den Ausfüllraum für die Grundform usw. umfassen kann. Natürlich veranlasst jeder andere ausgewählte Gegenstand in der Hierarchieansicht **118**, dass die Eigenschaften dieses Gegenstands in der Eigenschaftenansicht **120** angezeigt werden, und der Typ und die Art der Eigenschaften hängt von der Art des ausgewählten Gegenstandes ab.

**[0089]** Falls gewünscht, kann ein Eventsheet zur Verfügung gestellt werden, um die Ereignisse zu zeigen, die jedem oder allen der ausgewählten Hierarchiegegenstände in der Hierarchieansicht **118** zugeordnet sind. Ein solches Eventsheet könnte Ereignisse umfassen, die eintreten, wenn ein Anwender eine Aktion in Bezug auf ein grafisches Element durchführt, wie etwa ein "on-click"-Ereignis und ein "on-mouse-over"-Ereignis, die definieren, was passiert, wenn ein Anwender eine Visualisierung für das grafische Element oder einen Teil davon anklickt, oder was passiert, wenn der Anwender den Mauszeiger über der Visualisierung des grafischen Elementes oder einem Teil davon positioniert. Als Teil dieses Eventsheet kann einem Anwender Zugriff auf einen Script-Editor zur Verfügung gestellt werden, um ein Script zu definieren, das bei Auftreten des Ereignisses (das vorliegend als ein Routine definiert ist) ausgeführt wird, oder um auf dieses Script zuzugreifen.

**[0090]** Zusätzlich können, falls dies gewünscht wird, grafischen Elementen personalisierte Ereignisse zugeordnet sein. Personalisierte Ereignisse sind im Allgemeinen Ereignisse, die als ein Resultat eines äusseren oder externen Ereignisses definiert sind, oder sind Ereignisse, die eine Kommunikation mit einer externen Anwendung oder Datenquelle als Resultat einer von dem Anwender des grafischen Elementes unternommenen Aktion erfordern. Im Wesentlichen ist ein grafisches Ereignis eine Nachricht, die von dem grafischen Element gesendet wird, um das Auf-

treten einer Aktion im Hinblick auf die Visualisierung des grafischen Elementes anzuzeigen. Die enthaltende oder Laufzeitanwendung, welche das grafische Element implementiert, kann diese Ereignisse beispielsweise unter Verwendung einer C#-Syntax oder durch Registrieren der Ereignisbenachrichtigung auf jegliche bekannte Weise erfassen. Bei einem Beispiel kann die enthaltende Anwendung eine Pumpenüberhitzungs-Ereignis registrieren und eine My-Handler-Funktion zur Verfügung stellen, die will ein Script oder eine andere Routine fährt, um es einem Anwender zu ermöglichen, das Ereignis zu behandeln, wenn es ausgelöst wird. Diese personalisierten Ereignisse sind insbesondere nützlich, wenn das grafische Element als Teil eines Steuerungsbediener-Bildschirms implementiert ist. In diesem Fall kann das grafische Element selbst eine Ereignisbedingung bestimmen, wie etwa durch Vergleichen einer Pumpenhitze-Eigenschaft (die an eine externe Messung gebunden ist) mit einem Sollwert oder einer anderen Grenze, und kann eine Ereignisantwort durch Melden des Ereignisses an den Anwender und Implementieren einer Anwendung, eines Scripts, Dialogs usw. auslösen, um es dem Anwender zu ermöglichen, das Ereignis, in diesem Fall eine überhitzende Pumpe, zu behandeln oder sich damit zu befassen. Um diesen Ereignisbehandlungsprozess freizugeben, ermöglicht es der Element-Editor **50** einem Anwender, Ereignisargumente zu definieren, d.h. was als ein Ereignis anerkannt wird und was aus dem Auftreten eines Ereignisses resultiert.

**[0091]** Falls gewünscht, können grafische Elemente natürlich Standardereignisse zur Verfügung stellen, die aus einer Avalon-Steuerklasse vererbt sind. Bekanntermaßen sind die Avalon-Objekte oder -Steuerungen von Microsoft definierte Grafiken für die Avalon-Anwenderschnittstellenarchitektur für das Microsoft Longhorn-Betriebssystem, welche die Erstellung von Vektorgrafiken an einer Anwenderschnittstelle unterstützen. Diese Standardereignisse können beispielsweise das Einbinden der Betrachtung einer Faceplate-Anzeige oder einer Vorrichtungsdetail-Anzeige mit einem Klickereignis oder einem Doppelklick (Maus)-Ereignis umfassen. Beispielsweise kann ein Klick- oder Doppelklickereignis, bei dem es sich um ein Ereignis handelt, das einen Mausklick oder Doppelklick innerhalb der Grenzen der Visualisierung des grafischen Elementes meldet, die Anzeige einer Faceplate- oder Detailanzeige für die von der Visualisierung dargestellte Vorrichtung oder Subjekt auslösen und stellt dem Anwender vollständigere oder gründlichere Informationen über den Zustand, den Entwurf, die Herstellung usw. des dargestellten Subjektes zur Verfügung. Natürlich können andere Ereignisse, einschließlich von Standardereignissen, für ein Key Down/Up-Ereignis definiert sein, bei dem es sich um ein Ereignis handelt, das meldet, dass eine Taste gedrückt wurde, Mausereignisse einschließlich von Mouse Enter, Mouse Hover, Mouse Move, Mouse

Wheel und dergleichen Aktionen, oder jeglichen anderen, von einem Anwender ausgelösten Ereignissen.

**[0092]** Somit können Grafikelementereignisse, falls gewünscht, innerhalb des Scripts eines Ereignishandlers ausgelöst werden, wie etwa in dem Ereignishandler für ein Grundformereignis, oder innerhalb einer Transformationsfunktion, bei Änderungen einer Eigenschaft des grafischen Elementes. Ferner können Grundformen eines grafischen Elementes Ereignisse aufzeigen, die durch Interaktionen des Anwenders mit ihnen ausgelöst werden können, beispielsweise über die Maus und Tastatur. Diese Ereignisse stellen dem Anwender Optionen zur Verfügung, mit dem grafischen Element auf der Formen- und Grundformebene zu interagieren. Entwickler können die Ereignisse intern innerhalb des grafischen Elementes z.B. durch Spezifizieren von Ereignishandlern unter Verwendung von C#-Verfahren behandeln.

**[0093]** Unter erneuter Bezugnahme auf [Fig. 5](#) kann die Aktionen/Animation-Ansicht **123** eine Liste von Transformierungsanimationen und Eigenschaftanimationen, die für die gegenwärtig in der Editieransicht **114** angezeigte Visualisierung definiert sind, zur Verfügung stellen oder zeigen. Ein Anwender kann beispielsweise auf eine Zeile in der Ansicht **123** doppelklicken, um den Editor **50** zu veranlassen, eine Dialogbox zu präsentieren, bei der es sich um eine Transformierungsanimation-Dialogbox oder eine Eigenschaftanimation-Dialogbox handeln könnte, die es dem Anwender ermöglicht, die ausgewählte Animation zu editieren. Für gewöhnlich werden nur die Animationen der gegenwärtig ausgewählten Visualisierung gezeigt, aber das Auswählen einer Show-All-Box **144** könnte Animationen und Aktionen für alle Visualisierungen zeigen. Angenommen, dass Animationen mit sinnvollen Namen oder Beschreibungen, beispielsweise "Hebel animieren", "Motor drehen" usw. vorgesehen sind, stellt die Aktion/Animation-Ansicht **123** ein einfaches Betrachten und Zugriff auf Animationsverhaltensweisen zur Verfügung, die für eine Visualisierung definiert sind. [Fig. 9](#) veranschaulicht eine beispielhafte Aktionen/Animationen-Ansicht **123A**, welche die für die IsOn-Eigenschaft definierten Aktionen veranschaulicht. In diesem Fall, wenn die IsOn-Eigenschaft "True" ist, implementiert die als Rectangle1 bezeichnete Grundform der als Visual1 bezeichneten Visualisierung eine Einfüllaktion mit dem Füllwert "Rot". Ebenso wenn die IsOn-Eigenschaft "True" ist, implementiert die als Ellipse1 bezeichnete Grundform der als Visual1 bezeichneten Visualisierung eine sichtbare Aktion mit dem Wert von falsch (z.B. wird unsichtbar). Natürlich können Aktionen und Animationen für separate Grundformen einer Visualisierung separat definiert sein, aber diese Aktionen und Animationen können gleichzeitig auf ein gleiches Ereignis einwirken, auslösen, oder Eigenschaften ändern, um eine

komplexere, aber ansprechende Animation zu präsentieren. Ferner können Aktionen und Animationen das Durchführen einzelner Operationen umfassen, wie etwa Vergrößern, Einfärben usw., oder Wiederholoperationen, die sich kontinuierlich ereignen, bis sie abgestellt werden.

**[0094]** [Fig. 10](#) veranschaulicht eine Weise, auf die ein visueller Trigger **148** an einer Visualisierung **150** eines grafischen Elementes **152** auf der Grundlage der Eigenschaften wie etwa intrinsischen Eigenschaften des grafischen Elementes **152** implementiert werden kann. Insbesondere werden eine oder mehr der intrinsischen Eigenschaften des grafischen Elementes **152** von einem oder mehr Eigenschaftentriggern **154** (von denen drei gezeigt sind in [Fig. 10](#)) überwacht, die für den visuellen Trigger **148** definiert sind. Jeder Eigenschaftentrigger, der unter Verwendung eines Scripts implementiert werden kann, kann bestimmte Grafikelement-Eigenschaftenwerte überwachen, wie durch die gepunktete Linie **155** gezeigt ist. Danach kann jeder Eigenschaftentrigger einen oder mehr Grundformereigenschaftenwerte der Visualisierung **150** einstellen, wie durch die gepunkteten Linien **156** veranschaulicht ist, wenn der überwachte Grafikelement-Eigenschaftenwert erfüllt ist oder eine spezifizierte Bedingung erfüllt ist. So kann beispielsweise einer der Eigenschaftentrigger **154** eine oder mehr Grafikelement-Eigenschaften überwachen, um zu bestimmen, wann ein Wert von einer oder mehr dieser Eigenschaften in einen bestimmten Bereich fällt. Wenn diese Bedingung erfüllt ist, kann der eine der Eigenschaftentrigger **154** veranlassen, dass eine Animation oder ein anderes visuelles Script beispielsweise an einer Grundform oder einem anderen Element der Visualisierung **150** ausgeführt wird, um den visuellen Trigger **148** zur Verfügung zu stellen. Natürlich können mehrere der Eigenschaftentrigger **154** zusammen operieren, um mehrere gleichzeitige Änderungen oder Animationen als Teil des visuellen Triggers **148** zur Verfügung zu stellen, oder verschiedene der Eigenschaftentrigger **154** können unabhängig von einander z.B. auf verschiedene intrinsische Eigenschaften oder auf der Grundlage verschiedener Werte der gleichen intrinsischen Eigenschaft einwirken, um verschiedene Operationen des visuellen Triggers **148** zu verschiedenen Zeitpunkten oder im Ansprechen auf verschiedene Prozessbedingungen zur Verfügung zu stellen. Auf diese Weise können Farbänderungen, Animationen usw. in der Visualisierung **150** auf der Grundlage der Werte der intrinsischen Eigenschaften des grafischen Elementes **152** zur Verfügung gestellt werden.

**[0095]** Falls gewünscht, kann der Editor **50** eine visuelle Trigger-Tafel zur Verfügung stellen oder anzeigen, die alle Eigenschaftentrigger für eine gegenwärtig ausgewählte Visualisierung auflistet. Solch eine Tafel ist in [Fig. 9](#) veranschaulicht, in der die Element-eigenschaft-Spalte den Grafikelement-Eigenschafts-

namen auflistet, die erste Spalte mit Werten den Grafikelement-Eigenschaftenwert angibt, der beobachtet wird, die Sollspalte eine Visualisierung oder eine Grundformidentifizierung zur Verfügung stellt, die geändert wird, die Pfadspalte die Grundform-eigenschaft ist, die geändert wird, und die zweite Spalte mit Werten die Grundform-eigenschaft ist, die gelten soll, wenn der beobachtete Grafikelement-Eigenschaftenwert erfüllt ist. Natürlich können Animationen und andere Visualisierungsänderungen unter Verwendung dieser Struktur zur Verfügung gestellt werden.

**[0096]** [Fig. 11](#) veranschaulicht eine Transformierungsanimation-Dialogbox **160**, die verwendet werden kann, um einem Anwender zu helfen oder es ihm zu ermöglichen, eine Transformierungsanimation zu spezifizieren. Wie veranschaulicht ist, umfasst die Transformierungsanimation-Dialogbox **160** eine Verschiebungs-Sektion **162**, die es dem Anwender ermöglicht, eine Richtung und eine Strecke als Pixel zu spezifizieren, die eine Bewegung in einer Animation definieren, eine Drehungs-Sektion **164**, die es einem Anwender ermöglicht, eine Drehrichtung und einen Winkel für die Animation zu spezifizieren, und eine Skalierungs-Sektion **166**, die es dem Anwender ermöglicht, einen Skalierungsgrad für eine Animation sowohl in der horizontalen als auch in der vertikalen Richtung zu spezifizieren, und zu spezifizieren, ob das Darstellungsverhältnis verriegelt werden soll. Die Dialogbox **160** umfasst auch eine Kipp-Sektion **168**, die es dem Anwender ermöglicht, ein Kippen, das während einer Animation angewendet werden soll, sowohl in der horizontalen Richtung als auch in der vertikalen Richtung zu spezifizieren. Eine Einstellungs-Sektion **170** ermöglicht es einem Anwender, zu definieren, ob die Animation kontinuierlich sein soll oder nicht, und jede der anderen Verschiebungs-, Drehungs-, Skalierungs- und Kipp-Aktionen freizugeben. Ferner kann eine Vorschau **172** eine Vorschau der Animation veranschaulichen.

**[0097]** Bekanntermaßen ist eine Animation ein Objekt, dessen Wert sich über eine Zeitperiode ändert. Eine Eigenschaftsanimation kann erhalten werden, indem einer Animation eine Grundform-eigenschaft zugeordnet wird. Eigenschaftsanimationen können so geringfügig sein wie das Veranlassen, dass sich die Farbe des Textes ändert, oder das Veranlassen, dass ein Element wie eine Linie blinkt. Andererseits können Eigenschaftsanimationen komplexer sein, wie etwa das Animieren der Punkte einer Mehrfachlinie usw. Natürlich sind dies nur einige wenige Beispiele für Animationen, und andere Animationen können das Hinzufügen von oder Ändern von Farben, das Ändern der Größe (wie etwa der Breite, Länge oder point Größe) eines Grundform, das Verschieben, Drehen, Kippen, Skalieren von Grundformen usw. umfassen. Ferner könnten andere Animationen auf jeder Ebene der Visualisierung zur Verfügung gestellt werden, wie etwa auf der Grundformebene oder

als Teil der gesamten Visualisierung. Selbstverständlich können mehrere Animationen zur Verfügung gestellt oder auf jede bestimmte Visualisierung oder auf jeden bestimmten bestimmten Grundform einer Visualisierung angewendet werden, falls dies gewünscht wird. Ferner können mehrere Animationen im Zusammenhang mit einer bestimmten Visualisierung oder einer Grundform einer Visualisierung gleichzeitig oder zu verschiedenen Zeiten auf der Grundlage von Änderungen von Eigenschaftenwerten oder auf der Grundlage von durch einen Anwender eingeleiteten Triggerereignissen wie etwa Mausereignissen operieren.

**[0098]** [Fig. 12](#) veranschaulicht eine Eigenschaftsanimation-Dialogbox **180**, die von dem Editor **50** hergestellt werden kann, um es einem Anwender zu ermöglichen, eine Eigenschaft der Animation zu definieren oder zu ändern, um dadurch die Animation zu definieren. Die Dialogbox **180** umfasst eine Grenzdefinition-Sektion **182** und eine Zeitliniendefinition-Sektion **184**. Die Grenzdefinition-Sektion **182** stellt die Grenzen der Animation zur Verfügung oder definiert sie. Insbesondere definiert eine "from"-Eigenschaft die Startwerte der Animation, und eine "to"-Eigenschaft definiert die Endwerte der Animation. Die Zeitliniendefinition-Sektion **184** definiert eine Zeitdauer für die Animation, eine Anfangszeit und eine Endzeit für die Animation. Die Zeitdauer-Eigenschaft definiert die Länge der Zeit, welche die Animation für ihre vollständige Durchführung braucht, die Anfangszeit-Eigenschaft definiert einen Zeitversatz in Bezug auf die Startzeit der Animation, und die Endzeit-Eigenschaft definiert die Endzeit der Animation in Bezug auf die Startzeit. Eine Geschwindigkeitsdefinition-Sektion **186** ermöglicht es einem Anwender, eine Geschwindigkeit, eine Beschleunigung und eine Verlangsamung beispielsweise unter Verwendung von Schiebeteilen zu spezifizieren. Es dürfte ersichtlich sein, dass die Geschwindigkeit-Eigenschaft die Geschwindigkeit der Animation definiert, die Beschleunigung-Eigenschaft die Animation mit zunehmender Zeit schneller macht, während die Verlangsamung-Eigenschaft die Animation mit zunehmender Zeit langsamer macht. Eine Wiederholungsdefinition-Sektion **188** ermöglicht es einem Anwender, die Art und Weise zu definieren, auf die die Animation sich z.B. für eine spezifizierte Anzahl von Malen, für eine spezifizierte Zähl- oder Zeitdauer oder kontinuierlich wiederholt, bis sie abgestellt wird.

**[0099]** Eine weitere Animation-Dialogbox **190**, auf die durch die Dialogbox **180** zugegriffen werden kann, kann verwendet werden, um verschiedene Farben zu spezifizieren, die als Grenzen in der Animation verwendet werden, wenn die Grenzdefinition-Sektion **182** verwendet wird. Ähnliche Dialogboxen können verwendet werden, um es dem Anwender zu ermöglichen, andere nicht-numerische Werte als Eigenschaften einer Animation zu wählen. Obgleich

dies nicht gezeigt ist, können andere Animationseigenschaften ebenfalls unter Verwendung dieser oder anderer Dialogboxen ausgewählt oder spezifiziert werden. Beispielsweise kann eine automatische Umkehr-eigenschaft verwendet werden, um einen Boole'schen Wert zu definieren, der angibt, ob die Animation umgekehrt abläuft, nachdem sie ihre Vorwärtsrichtung durchlaufen hat, und eine "by" Eigenschaft kann verwendet werden, um den Gesamtbetrag zu definieren, um den die Animation ihren Startwert ändert. Eine relative Geschwindigkeit-Eigenschaft kann verwendet werden, um eine relative Geschwindigkeit zu definieren, mit der die Zeit für die Animation im Vergleich mit Elternanimationen verläuft, d.h. Animationen im Zusammenhang mit Elternelementen der Visualisierung. Ein Wert 1 kann beispielsweise angeben, dass die Animation mit der gleichen Geschwindigkeit wie die Elternanimation abläuft, ein Wert 2 kann anzeigen, dass die Animation zweimal so schnell wie ihre Elternanimation abläuft, und so weiter.

**[0100]** Auch wenn dies nicht gezeigt ist, können Dialogboxen auch dazu verwendet werden, die Fonteigenschaften einer Animation wie etwa den Stil, die Größe, den Font und die Farbe von Text in der Animation oder Änderungen daran einzustellen. Zusätzlich kann eine Stricheigenschaft-Dialogbox verwendet werden, um den Kanten- oder Linienstil, die Dicke und die Farbe von Linien einzustellen, und diese Eigenschaften zu ändern.

**[0101]** Grafische Dialogboxen können auch zum Editieren von anderen Merkmalen eines grafischen Elementes verwendet werden. Beispielsweise können grafische Dialogboxen dazu verwendet werden, um neue Prozessgrafiken zu erzeugen, die das Hinzufügen einer Grafikelement-Eigenschaft zu einem grafischen Element und das Binden einer Grafikelement-Eigenschaft an eine Laufzeitumgebung umfassen. Eine solche Bindungs-Dialogbox kann einen Browser zur Verfügung stellen, der einen Anwender in die Lage versetzt, durch verschiedene Etiketten oder Variablen zu browsen, die in dem Steuersystem oder einer anderen Laufzeitumgebung definiert sind, um gewünschte Etiketten, Variable, Namen usw. für das Durchführen einer Bindung aufzufinden. Ebenso können andere Dialogboxen verwendet werden, um Visualisierungen hinzuzufügen, visuelle Trigger hinzuzufügen, durch grafische Elemente oder Anzeigen zu browsen, und Ereignisse hinzuzufügen.

**[0102]** Somit dürfte verständlich sein, dass der Grafik-Editor **50** Unterstützung für das Definieren intrinsischen Eigenschaften von grafischen Elementen, Unterstützung für Visualisierungen grafischer Elemente einschließlich der Erzeugung und Manipulation von grafischen Grundformen oder Formen, des Hinzufügens von dynamischen Verhaltensweisen, konkret von Transformationsanimationen (Drehung, Ver-

schiebung, Skalierung und Kippen) und Eigenschaftsanimationen (Längenanimation, Farbanimation usw.) und das Definieren von Bedingungen, welche die Ausführung dynamischer Verhaltensweisen auslösen, zur Verfügung stellt. Ferner stellt der Grafik-Editor **50** Unterstützung für das Speichern und das Wiederauffinden grafischer Elemente in der Datenbank zur Verfügung, einschließlich Unterstützung für die Serialisierung grafischer Elemente in einen xml-Blob, und Unterstützung für das Kategorisieren grafischer Elemente in einer Grafikelemente-Bibliothek. Ferner stellt das System das Speichern von grafischen Elementen in anwenderdefinierten Kategorien oder Gruppen, das Verriegeln oder das Vorsehen einer Sicherung für solche grafische Elemente, das Speichern der grafischen Elemente in anwenderdefinierten oder anderen nützlichen Kategorien, Versionserstellung für grafische Elemente usw. zur Verfügung.

**[0103]** Auf jeden Fall wird ein grafisches Element nach seiner Erzeugung in einer Grafikelement-Datenbank auf eine Weise gespeichert, die nicht an Prozessvariable oder andere Daten in der Laufzeitumgebung der Verarbeitungsanlage gebunden ist. Dieses gespeicherte grafische Element wird nicht unbedingt in Anzeigen verwendet, ist aber für eine solche Verwendung verfügbar, da dieses Element nun auf eine Laufzeitumgebung heruntergeladen und an bestimmte tatsächliche oder simulierte physische Elemente in der Verarbeitungsanlage oder in dem Prozesssteuerungssystem gebunden werden kann. Wenn es gebunden ist, sind die intrinsischen Eigenschaften des grafischen Elementes, wie etwa die Prozessvariable, der Sollwert, die aktuelle Geschwindigkeit usw., an Datenreferenzen in der Laufzeitumgebung gebunden und damit eingebunden.

**[0104]** [Fig. 13](#) veranschaulicht eine Weise, auf die grafische Elemente mit tatsächlichen physischen Komponenten oder Elementen innerhalb einer Verarbeitungsanlage oder einem in einer Verarbeitungsanlage verwendetes Prozesssteuerungssystem eingebunden oder daran gebunden sein können. Insbesondere umfasst jedes grafische Element **192** ein XAML **194**, das im Wesentlichen die Visualisierungen für ein Element definiert, und zugeordnete Scripts **196**, Animationen **198**, Trigger **200**, Aktionen **202** und Ereignisse **204**. Das grafische Element **192** umfasst auch eine Referenztabelle **206**, die alle Referenzen mit Zusammenhang mit dem XAML **194** auflistet oder enthält, die in dem XAML **194** verfügbar oder offen und exponiert sind. Die Referenztabelle **206** besteht im Wesentlichen aus, oder umfasst die Eigenschaften und Parameter des grafischen Elementes, sowie jegliche andere Variablen oder Referenzen, die von den Scripts **196**, Animationen **198**, Triggern **200**, Aktionen **202** und Ereignisse **204** verwendet werden. Die Variablen oder Subjekte in der Referenztabelle **206** können sich auf Variable, Tabellen, Tabellenein-



träge in anderen Programmen, oder jegliche andere Typen oder Arten von Daten beziehen oder an solche gebunden sein, die anderweitig in dem Prozesssteuerungssystem definiert sind.

**[0105]** Wie in [Fig. 13](#) veranschaulicht ist, wird eine Auflösungstabelle **208** verwendet, um die Referenzen oder Variablen in der Referenztabelle **206** mit der tatsächlichen Prozesssteuerungs-Umgebung oder einer anderen Laufzeitumgebung einzubinden. Allgemein gesprochen wird die Auflösungstabelle **208**, die Referenzen entweder direkt oder über Aliasing definieren kann, für das grafische Element **192** zur Verfügung gestellt oder erzeugt, wenn das grafische Element **192** tatsächlich konfiguriert wird, um in einer bestimmten Anzeige in einer Laufzeitumgebung verwendet zu werden und auf eine Laufzeitmaschine heruntergeladen wird. Vor oder während der Operation des grafischen Elementes **192** in einer Anzeige löst die Auflösungstabelle **208** die Aliase und andere Parameter auf und bindet diese aufgelösten Datenverbindung mit der Referenztabelle **206** ein, um eine Bindung zwischen den Variablen in der Referenztabelle **206** und den tatsächlichen Datenquellen in dem Prozesssteuerungssystem oder einer anderen Laufzeitumgebung zur Verfügung zu stellen.

**[0106]** Um eine unnötige Verwendung von Verarbeitungsleistung in dem Prozesssteuerungssystem während der Laufzeit zu vermeiden, müssen grafische Elemente und die grafischen Anzeigen, in denen sie verwendet werden, nicht an die Laufzeitumgebung gebunden bleiben, wenn sie nicht tatsächlich auf einem Anzeigebildschirm angezeigt oder verwendet werden. Statt dessen kann die Auflösungstabelle **208** nur dann an die Referenztabelle **206** des grafischen Elementes **192** gebunden werden, wenn das grafische Element **192** auf einem Bildschirm in der Laufzeitumgebung läuft oder angezeigt wird.

**[0107]** Da jedes grafische Element durch die Verwendung einer Referenztabelle **206** und einer Auflösungstabelle **208** an das Prozesssteuerungssystem oder die Laufzeitumgebung gebunden ist, und da eine Bindung stattfindet, nachdem das grafische Element auf einer Systemebene erzeugt und in eine Laufzeitumgebung oder Maschine kopiert und geladen wurde, kann das grafische Element **192** separat an verschiedene Datenquellen für verschiedene Verwendungen zu verschiedenen Zeiten gebunden werden. Ferner kann das grafische Element **192** an Daten gebunden werden, die durch eine beliebige Anzahl von verschiedenen Quellen einschließlich Datenquellen in den verschiedenen Funktionsbereichen **42** von [Fig. 2](#) erzeugt oder verfügbar gemacht werden, wie etwa Datenquellen im Zusammenhang mit Steueraktivitäten, Wartungsaktivitäten, Modellierungsaktivitäten, Simulationsaktivitäten, Konfigurationsaktivitäten usw.

**[0108]** Wie beispielsweise in [Fig. 14](#) veranschaulicht ist, kann ein Anzeigeelement **192** an Daten gebunden sein, die in einer großen Vielfalt von Datenquellentypen erzeugt oder von ihnen verfügbar gemacht werden, und kann verwendet werden, um diese Daten anzuzeigen oder zu verarbeiten. Diese Datenquellen können Steuerumgebungs-Datenquellen **210** wie Controllerprogramme wie etwa das allgemein bekannte DeltaV-Steuersystem, OPC-Datenquellen **212**, die Verbindungen zu anderen Systemen über die allgemein bekannte OPC-Verbindungsschnittstelle zur Verfügung stellen, Wartungsdatenquellen **214** wie das allgemein bekannte AMS-System, höhere Ebene- oder Geschäftssysteme **216** wie das allgemein bekannte Ovation-System, und sogar Datenquellen, die Konkurrenzsysteme **218** verwenden, wie Konkurrenz-Steueranwendungen, umfassen. Auf diese Weise kann das grafische Element **192** verwendet werden, um Daten oder und Visualisierungen von physischen Elementen auf einer beliebigen Ebene des Systems unter Verwendung von Daten von einer jeglichen Anwendung in dem System anzuzeigen, selbst wenn die Daten von einer Vielfalt von verschiedenen Typen von Anwendungen kommen oder stammen, einschließlich Anwendungen im Zusammenhang mit Konkurrenzsystemen, von denen nie vorgesehen war, dass sie zusammen operieren. Da die grafischen Elemente und die aus grafischen Elementen erzeugten grafischen Anzeigen somit auf einer Systemebene erzeugt werden, können sie verwendet werden, um Visualisierungen der Geschehnisse innerhalb der Anlage für jeglichen Zweck zur Verfügung zu stellen, selbst wenn verschiedene Typen von Software auf verschiedene Hardware- und Software-Komponenten innerhalb der Anlage zugreifen und sie betreiben.

**[0109]** Wegen der modularen Natur der grafischen Elemente ist es möglich, automatisch oder halbautomatisch grafische Elemente aus Vorrichtungsbeschreibungen (DDs) zu erzeugen, die in der allgemein bekannten und gut unterstützten Vorrichtungsbeschreibungssprache (DDL) geschrieben sind. Insbesondere stellen die Hersteller einer Vorrichtung für gewöhnlich eine DD für jede von ihnen hergestellte Vorrichtung zur Verfügung, die in der DDL die Parameter im Zusammenhang mit der Vorrichtung definieren, wie mit der Vorrichtung kommuniziert werden kann, Grenzen für die Vorrichtung usw. Im Ergebnis kann eine Grafikelemente-Erzeugungsanwendung die DD für eine Vorrichtung in der DDL lesen, um den Vorrichtungstyp und die wichtigen Parameter, Grenzen usw. im Zusammenhang mit dieser Vorrichtung zu bestimmen, und können dann diese Parameter als die intrinsischen Eigenschaften oder Parameter des grafischen Elementes für die Vorrichtung definieren. Das Programm kann auch eine grundlegende Formenkombination als eine Visualisierung für die Vorrichtung wählen oder definieren, und kann ein oder mehr generische Scripts wählen, die verwendet wer-

den sollen, um grundlegende Aktionen und Animationen für die Vorrichtung zur Verfügung zu stellen, entweder auf der Grundlage von Informationen aus der DD oder auf der Grundlage von Schablonen, die für Vorrichtungen des Vorrichtungstyps gespeichert sind, der von der DD für die Vorrichtung definiert wird. Falls gewünscht, kann das Programm während dieser Prozesse einen Anwender auffordern, Informationen über die Vorrichtung zur Verfügung zu stellen oder eine Auswahl zu treffen, welche Animationen, Visualisierungen, Grundformen usw. für das grafische Element verwendet werden sollen.

**[0110]** Für ein vollständigeres oder besser definiertes grafisches Element kann das Programm verschiedene generische Grafikschemen für verschiedene Typen von Vorrichtungen wie etwa für Sensoren, Ventile, Motoren, Tanks usw. speichern. Das Programm kann dann das zu verwendende grafische Schablonelement auf der Grundlage des Vorrichtungstyps gemäß der Definition durch die DD für die Vorrichtung bestimmen. Falls gewünscht, kann die Schablone verschiedene Wahlmöglichkeiten oder Optionen, die in dem grafischen Element verwendet werden sollen, zur Verfügung stellen oder verfügbar halten, und diese Wahlmöglichkeiten können auf der Grundlage von Informationen in der DD für die Vorrichtung oder auf der Grundlage einer Anwendereingabe bestimmt werden. Somit kann die Schablone beispielsweise verschiedene intrinsische Parameter im Zusammenhang mit verschiedenen Untertypen von Vorrichtungen wie etwa Sensoren zur Verfügung stellen, und das Programm kann die intrinsischen Parametern für die Definition des grafischen Elementes auf der Grundlage der Informationen in der DD bestimmen.

**[0111]** Verschiedene elementare Scripts können als Teil der Schablone zur Verfügung gestellt werden und können verwendet werden, um Verhaltensweisen für die Visualisierung zur Verfügung zu stellen. Wiederum können die Scripts, die in dem grafischen Element verwendet werden sollen, automatisch auf der Grundlage der Informationen in der DD wie etwa dem Vorrichtungstyp usw., oder, falls gewünscht, auf der Grundlage von Fragen an den Anwender ausgewählt werden. Zusätzlich können die verschiedenen, in dem Script verwendeten Grenzen oder Variablen aus Informationen in der DD bestimmt werden. Somit kann beispielsweise ein Script, wie etwa eines, das eine drehende Grafik zur Verfügung stellt, zur Verwendung ausgewählt oder in dem grafischen Element zur Verfügung gestellt werden, falls die DD angibt, dass die Vorrichtung eine drehende Einrichtung ist, und bestimmte Aspekte der Drehungsgrafik, wie etwa die Geschwindigkeit bzw. Drehzahl, wann sich die Farbe der Grafik ändert, usw., kann auf Grenzen basieren, die gemäß der Definition in der DD der Vorrichtung zugeordnet sind. Diese Grenzen können beispielsweise eine normale oder Nennbetriebsge-

schwindigkeit, die definierten zu hohe oder zu niedrige Geschwindigkeitsbedingungen oder Geschwindigkeitsgrenzen usw. sein. Als ein anderes Beispiel, falls die Vorrichtung ein Sensor ist, können die hohen und niedrigen Werte des Sensors verwendet werden, um Grafiken im Zusammenhang mit dem aktuellen Anzeigewert des Sensor, Grafiken zur Darstellung, ob der Sensor defekt ist, usw. zur Verfügung zu stellen.

**[0112]** Auf diese Weise kann ein grundlegendes grafisches Element automatisch aus einer DD für eine Vorrichtung erzeugt werden, wenn die Vorrichtung an eine Verarbeitungsanlage angeschlossen oder in ihr erkannt ist, indem bestimmte grundlegende Scripts, grafische Animationen, Visualisierungen und intrinsische Parameter für das grafische Element auf der Grundlage des Vorrichtungstyps und bekannter Charakteristiken der Vorrichtung gemäß der Definition in der DD für die Vorrichtung definiert werden. Diese automatische Erzeugung eines grafischen Elementes stellt einem Anwender eine gewisse Fähigkeit zum automatischen Integrieren einer neu hinzugefügten Vorrichtung in Grafiken ohne viel bzw. sogar ohne grafisches Editieren für die Vorrichtung zur Verfügung, so dass Vorrichtungen automatisch, zumindest auf einer elementaren Ebene in grafischen Anzeigen unterstützt werden können, wenn die DD für diese Vorrichtungen in das System geladen wird. Mit anderen Worten, ein Anwender kann automatisch ein grafisches Element aus einer DD für eine Vorrichtung erzeugen, indem er dem System eine DD für eine Vorrichtung zur Verfügung stellt und dann das Programm zum Erzeugen eines grafischen Elementes für die Vorrichtung aus der DD ausführt. Danach kann dieses grafische Element in einer oder mehr grafischen Anzeigen verwendet werden, um eine Visualisierung für die Vorrichtung zu modellieren oder zur Verfügung zu stellen, die dem Anwender die Fähigkeit zur Verfügung stellt, die Vorrichtung zu modellieren oder die Vorrichtung in grafischen Anzeigen zu veranschaulichen, ohne manuell ein grafisches Element für die Vorrichtung erzeugen zu müssen.

**[0113]** Wie weiter oben erwähnt wurde, kann der Grafik-Editor **50** nach dem Erzeugen einer Anzahl von grafischen Elementen **74** von [Fig. 3](#) verwendet werden, um eine oder mehr grafische Anzeigen wie etwa die Anzeigen **76** von [Fig. 3](#) zu erzeugen. Tatsächlich kann der Anzeige-Editor **50**, falls gewünscht, einem Anwender oder Käufer mit einer Anzahl von vorkonfigurierten grafischen Elementen zur Verfügung gestellt werden, denen verschiedene Visualisierungen wie etwa Visualisierungen für verschiedene Industrien, verschiedene funktionelle Verwendungen usw. zugeordnet sein können. Der Grafik-Editor **50** ermöglicht es dann dem Anwender, personalisierte Elemente zu erzeugen, sowie grafische Anzeigen **76** zu erzeugen oder aufzubauen.



**[0114]** Allgemein gesprochen wird der Anwender zum Erzeugen einer Anzeige **76** aus einer Bibliothek von grafischen Elementen **74** und anderen visuellen Elementen eine Auswahl treffen und diese zusammen anordnen, um eine Anzeige aufzubauen. Nachdem eine grafische Anzeige vollständig ist, kann die resultierende Datenstruktur oder das resultierende Objekt in einer Datenbank wie etwa einer Konfigurations-Datenbank als ein Anzeige-Klassenobjekt gespeichert werden, dessen verschiedene Elemente, intrinsischen Eigenschaften und Visualisierungen sämtlich als ein einziges Definitionssubjekt definiert sind. Dieses Anzeige-Klassenobjekt ist jedoch nicht an Prozessvariable gebunden und wird nicht unbedingt in Laufzeitanzeigen verwendet. Falls gewünscht, kann dieses Klassenobjekt in der Datenbank als ein XML-Blob gespeichert werden, dessen Visualisierungen, Scripts usw. sämtlich als ein XML Subjekt gespeichert und verkettet sind. Danach können individuelle grafische Anzeigen aus diesem Klassenobjekt erzeugt werden, und diese individuellen grafischen Anzeigen können auf Bediener-Workstations oder andere Laufzeitumgebungen zugeteilt und heruntergeladen werden.

**[0115]** Wenn eine Anzeige heruntergeladen wird, werden die Grafikelementdefinitionen darin in Avalon-Steuerungen konvertiert, in Assemblies kompiliert, und auf der Zielmaschine, d.h. der Laufzeitmaschine angeordnet. Die Anzeige, die selbst ein Avalon-Subjekt ist, referenziert die kompilierten Steuerungs-Assemblies, und zusätzlich kann eine personalisierte Datenquelle erzeugt werden, die als ein Datenadapter dient, welcher die Avalon-Steuerungen mit den Backend-Datenquellen wie den Steuerungs- oder Wartungs-Laufzeitanwendungen verbindet. Auf diese Weise können die grafischen Anzeigen (und, falls gewünscht, individuelle grafische Elemente) in einer Sprache erzeugt und editiert, in einer anderen Sprache oder Form (z.B. als ein XML-Blob) gespeichert, und in einer dritten Sprache oder Form ausgeführt werden (z.B. als Avalon-Steuerungen, die an Scripts gebunden sind, welche in jeglicher von verschiedenen ausführbaren Sprachen geschrieben sind).

**[0116]** Unter Bezugnahme auf [Fig. 15](#) kann ein Bildschirm **220** von dem Grafik-Editor **50** hergestellt werden, um es einem Anwender zu ermöglichen, eine oder mehr grafische Anzeigen zu erzeugen. Allgemein gesprochen ist eine grafische Anzeige aus untereinander verbundenen Elementen zusammengesetzt, die physische Vorrichtungen in einer Anlage darstellen, und kann zusätzliche Informationen im Zusammenhang mit diesen Vorrichtungen umfassen oder anzeigen. Jedes der grafischen Elemente in einer grafischen Anzeige weist eine Gruppe von intrinsischen Eigenschaften auf, die gleich Prozessvariablen, Konstanten, oder anderen externen Werten sind, und wie weiter oben erläutert wurde, kann jedes gra-

fische Element mehrere visuelle Darstellungen besitzen, die eine dynamische Verhaltensweise, z.B. Farbänderungen oder Animation, aufweisen können. Zusätzlich kann die grafische Anzeige statische Elemente umfassen, wie etwa Text, Boxen usw., Anwenderdynamos, die es dem Anwender ermöglichen, mit der grafischen Anzeige auf eine bestimmte Weise zu interagieren, variable Boxen, die dem Anwender einen Prozess oder andere Informationen anzeigen können, usw.

**[0117]** Ähnlich dem Editier-Bildschirm **112** von [Fig. 5](#) für ein grafisches Element weist der beispielhafte Grafikanzeige-Editor Bildschirm **220** von [Fig. 15](#) einen Haupteditier-Canvas **224** auf, in dem eine grafische Anzeige aufgebaut wird, und eine Paletten-Sektion **226**, in der Schablonen-Grafikelemente, Grundformen oder andere Bibliothekselemente angezeigt werden können, und aus der diese verschiedenen Elemente ausgewählt und auf den Haupteditier-Canvas **224** gezogen und fallengelassen werden können. Der Haupteditier-Canvas **224** stellt eine Diagrammansicht zur Verfügung, in welcher der Anwender interagiert, um Grafiken zu erzeugen und Grafiken, entweder Grundformen oder Kombinationen, zu editieren, und stellt Vektorgrafikeditier- und -betrachtungsmerkmale zur Verfügung. Der Editor **50** ermöglicht es einem Anwender, grafische Objekte wie etwa grafische Elemente irgendwo in einem grenzenlosen Raum anzuordnen, der sich innerhalb oder ausserhalb des gegenwärtig im Editier-Canvas **224** angezeigten Anzeigerahmens befinden kann. Somit kann der Haupteditier-Canvas **224** nur einen Teil der gegenwärtig erzeugten Anzeige abbilden, und ein Anwender kann sich in der Ansicht umherbewegen, um andere Teile einer Anzeige zu betrachten, ein- und auszoomen, um die Vergrößerungsstufe der Ansicht zu ändern, und die Ansicht drehen, um jederzeit die bequemste Orientierung für die Bearbeitung der Anzeige zu erhalten. Es dürfte jedoch verständlich sein, dass jedes Zeichnen und Manipulieren von Grafikobjekten im Haupteditier-Canvas **224** abgeschlossen wird. Um beim Zeichnen zu helfen, kann die X-Y-Position der Ansicht aus den Linealen **227** abgelesen werden, die Vergrößerungsstufe kann in einer Dropdown-Liste auf einer Werkzeugleiste **228** angezeigt werden und kann durch das Wählen von Einträgen auf einem Zoom-Popup geändert werden, usw.

**[0118]** Der Bildschirm **220** umfasst auch eine Hierarchiesektion **230**, die eine hierarchische Anzeige oder Auflistung der Elemente in dem Haupteditier-Canvas **224** abbildet, eine Eigenschaften-Sektion **232**, welche die Eigenschaften oder Parameter im Zusammenhang mit der im Canvas **224** erzeugten Anzeige oder mit hervorgehobenen Elementen im Canvas **224** auflistet, und eine Bindungs-Sektion **234**, welche die Art und Weise veranschaulicht oder auflistet, auf die verschiedene Parameter oder Merkmale an Elemente in dem Prozesssteuerungssystem

oder in jeder anderen Laufzeitumgebung gebunden sind. Zusätzlich weist der Bildschirm **220** eine Werkzeugleiste **236** auf, welche die verschiedenen Ansichten oder Schichten im Zusammenhang mit der im Haupteditier-Canvas **224** erzeugten grafischen Anzeige auflistet. Genauer gesagt kann jede bestimmte Anzeige verschiedene Schichten oder Ansichten umfassen, die von verschiedenen Menschen in verschiedenen Zusammenhängen verwendet werden können, wie etwa eine Bedieneranzeige, eine Wartungsanzeige, eine Simulationsanzeige, eine Modellanzeige, eine Ingenieursanzeige, eine Geschäftsanzeige usw. Bei dem beispielhaften Bildschirm **220** von **Fig. 15** weist die Werkzeugleiste **236** drei Ansichten oder Ebenen auf, die als eine Operationsansicht (gegenwärtig im Haupteditier-Canvas **102** gezeigt) veranschaulicht, welche eine typische Steuerungsbedieneransicht zur Verfügung stellt, eine Eng/Maintenance-Ansicht, welche eine typische Wartungs- oder Ingenieursansicht zur Verfügung stellt, und eine Schulungsansicht, welche eine Simulationsansicht zur Verfügung stellt, die beim Simulieren der Verarbeitungsanlage oder eines in der zu erzeugenden Anzeige abgebildeten Teils der Verarbeitungsanlage verwendet wird. Weil in dem Haupteditier-Canvas **224** von **Fig. 15** keine Elemente ausgewählt sind, zeigt die Eigenschaften-Sektion **232** den aktuellen Namen der zu erzeugenden Anzeige zusammen mit Merkmalen der Anzeige, wie etwa der Breite und Höhe, einer von dem Erzeuger zur Verfügung gestellten Beschreibung, einem Namen, einer Hintergrundbeschreibung und jeglichen anderen Informationen an, die ein Erzeuger möglicherweise für die Anzeige speichern möchte. Auf ähnliche Weise, weil kein tatsächlichen Elemente in dem Haupteditier-Canvas **224** von **Fig. 15** angeordnet sind, zeigt die Hierarchiesektion **230** nur den Namen der Anzeige, und es werden keine Bindungen in der Bindungen-Sektion **234** gezeigt.

**[0119]** Während der Erzeugung einer Anzeige kann ein Anwender ein Element wie etwa ein grafisches Element auf dem Haupteditier-Canvas **224** anordnen, indem er auf dieses Element beispielsweise über eine von einer Gruppe von Elementkategorien zugreift, die in der Palettensektion **226** definiert ist. Bei dem Beispiel von **Fig. 15** veranschaulicht die Palettensektion **226** verschiedene Kategorien von Elementen, die in dem Haupteditier-Canvas **224** angeordnet und miteinander verbunden werden können, um eine vollständige Anzeige einschließlich von Betätigungselementen, Berechnungs- und Steuerelementen, Verarbeitungselementen, Eigenschaften und Messungen, Formen, Anwenderschnittstellen-Steuerungen und anwenderdefinierten Elementen zu erzeugen. Natürlich können in der Palettensektion **226** jegliche andere Arten und Kategorien oder Unterkategorien von vordefinierten Elementen zur Verfügung gestellt oder auf diese zugegriffen werden. Bei diesem Beispiel können Betätigungsglieder Ven-

tile und andere Betätigungselemente umfassen, während Berechnungs- und Steuerelemente jegliche steuerungsbezogene Elemente wie etwa Angaben von Controllern, Steuerschleifen wie etwa PID-Steuerschleifen, oder andere Typen von Steuerschleifen, Funktionsblöcken, Steuermodulen usw. umfassen können. Verarbeitungselemente gemäß der Darstellung in **Fig. 15** können Tanks, Reaktoren, Mischer oder andere Elemente umfassen, die Materialien auf irgend eine Weise verarbeiten, sowie jegliche andere Typen von Vorrichtungen, Einheiten usw. Eigenschaften und Messungen können Boxen oder Anzeigeelemente umfassen, die entworfen werden, um Eigenschaften oder Messungen oder andere Daten in der Laufzeitumgebung zu zeigen, wie etwa Prozessvariable, Warnungen usw. Formen können Grundformen oder andere vordefinierte Formen sein, während UI-Steuerungen verschiedene Anwenderschnittstellen-Steuerelemente wie Knöpfe, Schieber, Drehknöpfe, Toolboxes usw. umfassen, die der Anwender auf einem Anzeigebildschirm manipulieren kann, um eine Eingabe in die Anzeige zu bewirken. Natürlich können anwenderdefinierte Elemente jegliche vordefinierten Elemente wie etwa jegliche anderen grafischen Elemente oder aus grafischen Elementen hergestellte Elemente einer höheren Ebene umfassen. In einem Fall können anwenderdefinierte Elemente Verarbeitungseinheiten, Verarbeitungsanlagenbereiche oder andere Prozess-Subjekte einer höheren Ebene umfassen. Wie weiter oben erwähnt wurde, kann ein Zugriff auf anwenderdefinierte Elemente auf der Grundlage der Identität des Anwenders, der den Editor **50** verwendet, beschränkt werden, oder der Zugriff kann allgemein verfügbar sein. Es dürfte natürlich verständlich sein, dass jegliche anderen Elemente, Formen usw. in der Palettensektion **226** unter geeigneten Überschriften angeordnet werden können, um diese Elemente organisiert und durch einen Anwender leicht zugreifbar zu halten.

**[0120]** Wenn ein Anwender ein Element wie etwa einen Mischtank im Haupteditier-Canvas **224** anordnet, kann dieses Element im Canvas **224** unter Verwendung der Vorgabevisualisierung für dieses Element erstellt werden. In diesem Fall zeigt die grafische Hierarchiesektion **230** das Element und stellt eine Hierarchie von Unterelementen für dieses Element, wie etwa Visualisierungen, Animationen, Grundformen, usw. im Zusammenhang mit diesem grafischen Element auf eine Weise ähnlich der in **Fig. 5** Abgebildeten zur Verfügung. Wenn ferner ein grafisches Element in dem Canvas **224** gezeigt ist, das Visualisierungen oder Animationen umfasst, die an die Laufzeitumgebung gebunden werden können, veranschaulicht die Bindungsanschluss-Sektion **234** die gegenwärtig definierten Bindungen.

**[0121]** Falls gewünscht, kann der Anwender die Visualisierung eines grafischen Elementes in dem Haupteditier-Canvas **224** oder in der Palettensektion

**226** durch Auswählen einer verschiedenen Visualisierung auf jegliche gewünschte Weise auswählen oder ändern. Der Anwender kann diese Visualisierungs-Auswahlfunktion unter Verwendung einer Dropdown-Liste oder -Box durchführen, auf die z.B. durch Rechtsklicken mit dem Mauszeiger über dem Element oder durch Auswählen einer verschiedenen Visualisierung in der Hierarchiesektion **230**, oder auf jegliche andere gewünschte Weise zugegriffen werden kann. [Fig. 16](#) veranschaulicht einen Teil des Haupteditier-Canvas **224** und der Palettensektion **226** des Bildschirms **220**, in dem ein vertikales Tankelement **240** gezeigt ist, das aus der Palettensektion **226** im Haupteditier-Canvas **224** angeordnet wird. Das vertikale Tankelement **240** ist eine Kopie oder Erstellung einer Schablone oder eines vertikalen Klassen-Tankelementes **241**, das in der Palettensektion **226** abgebildet ist. Es können jedoch zusätzliche Visualisierungen für den vertikalen Tank **240**, einschließlich der in ein Seitenleiste **242** gezeigten Visualisierungen 1–8, auf die durch einen Rechtsklick auf die Maus zugegriffen wird, wenn sich der Mauszeiger über der Visualisierung **240** befindet, als die für das vertikale Tankelement **240** zu verwendende Visualisierung ausgewählt werden, wenn sie sich in der Anzeige befinden, die im Editier-Canvas **224** erzeugt wird. Wie in der Palettensektion **226** veranschaulicht ist, kann ein ähnlicher Typ einer Seitenleistenanzeige **243** durch Rechtsklicken auf das vertikale Schablonen-Tankelement **241** in der Palettensektion **226** erhalten werden, um die möglichen Visualisierungen des Schablonen-Tankelementes **241** zu betrachten oder zu erhalten. Das Auswählen einer verschiedenen Visualisierung in der Seitenleiste **242** verändert die Visualisierung des Tankelementes **240**, das im Canvas **224** verwendet wird, während das Auswählen einer verschiedenen Visualisierung in der Seitenleiste **243** die Vorgabeeinstellung oder Visualisierung für das Schablonen-Tankelement **241** verändert, das in der Palettensektion **226** gespeichert ist.

**[0122]** Zusätzlich zum Auswählen von grafischen Elementen und anderen Elementen aus der Palettensektion **226** zum Erzeugen einer Anzeige kann ein Anwender auch Gegenstände aus der in [Fig. 15](#) abgebildeten Werkzeugleiste **228** verwenden oder auswählen, um grundlegende Zeichenwerkzeuge wie etwa Linien, Formen wie Quadrate, Rechtecke, Kreise, Fünfecke usw., Text usw. zu erhalten, und kann diese einfachen Zeichenwerkzeuge oder Elemente verwenden, um Linien zu zeichnen oder Text in der Anzeige hinzuzufügen. Zusätzlich kann der Anwender Verbindungselemente zu der Anzeige im Canvas **224** unter Verwendung der in die Werkzeugleiste **228** gezeigten Werkzeugleiste-Verbindungselemente **245** hinzufügen. Wenn das Verbindungselement **245** ausgewählt wird, kann es dem Anwender eine Liste von Verbindern zur Verwendung in der Anzeige unter Verwendung beispielsweise eines Dropdown-Menüs, einer Dialogbox usw. zur Verfügung stellen. Mögliche

Verbindungselemente umfassen Rohre, Förderbänder, elektrische Leitungen, Fluidstromleitungen oder andere Typen von Verbindern, die tatsächlich ein Hardware-Element wie etwa einen Tank, einen Mischer, eine Pumpe usw. mit einem anderen Hardware-Element wie etwa einem Ventil, einem Sensor usw. verbinden. Solche Verbindungselemente und das Vorsehen von Verbindungselementen zwischen verschiedenen Darstellungen von physischen Vorrichtungen wie etwa Ventiltanks, Pumpen usw. sind ausführlicher in der US-Veröffentlichung Nr.2004/0153804 erläutert, auf die hiermit ausdrücklich Bezug genommen wird. Die Verbindungselemente können es einem Anwender ermöglichen, verschiedene Elemente in der zu erzeugenden Anzeige miteinander zu verbinden und mit den Verbindungspunkten abgleichen, die für die verschiedenen, weiter oben erläuterten Elemente für grafische Elemente definiert sind. Beispielsweise können Leitungsverbindern verwendet werden, um verschiedene Elemente anzubringen, die Leitungsverbindern aufweisen, während ein Rohrleitungsverbindern verwendet werden kann, um Elemente anzubringen, die mit Rohrleitungsverbindern definiert wurden. Falls gewünscht, kann der Editor **50** Verbindungselemente implementieren, die es einem Anwender nur ermöglichen, verschiedene grafische Elemente wie etwa ein Ventil und einen Tank mit dem geeigneten Type eines Verbinders zu verbinden, der durch die Verbindungspunkte des Ventils und des Tanks definiert ist. Natürlich kann die Erscheinung der Verbindungselemente auf der Grundlage des Verbindungstyps verschieden sein, um ein besseres Aussehen und Gefühl für die zu erzeugende Anzeige zur Verfügung zu stellen, die mit Verbindungselementen erzeugt wird.

**[0123]** Natürlich kann die Werkzeugleiste **228** andere Vektor-Zeichenwerkzeuge und Standardbefehle oder -funktionen wie etwa typische Dateioptionen (Neu, Öffnen, Speichern, Speichern unter, Schließen, Beenden usw.), Editieroptionen (wie etwa Rückgängigmachen, Wiederherstellen, Ausschneiden, Kopieren, Einfügen, Löschen), Suchoptionen usw. enthalten. Zusätzlich kann die Werkzeugleiste **228** Menügegenstände enthalten, die grafikbezogene Befehle wie etwa eine Add Property-Funktion zur Verfügung stellen, welche dazu verwendet werden können, zu einem grafischen Element intrinsische Eigenschaften hinzuzufügen, eine Add Event-Funktion, welche dazu verwendet werden kann, ein Grafikergebnis hinzuzufügen, eine Add Visualization-Funktion, welche es einem Anwender ermöglichen kann, eine Visualisierung hinzuzufügen, eine Add Visual Trigger-Funktion, welche eine Dialogbox öffnen kann, um es dem Anwender zu ermöglichen, Informationen bezüglich des zu erzeugenden Triggers einzugeben, usw. Ferner kann ein Menü Format-Funktionen (wie Einstellen eines bezogenen Textes, Linien- und Fülleigenschaften für Text, Linien und zweidimensionale und dreidimensionale Bilder)

und Formfunktionen ermöglichen, welche das Gruppieren oder Vereinzeln von Objekten, Auswählen der Reihenfolge von Objekten von vorne nach hinten, Ausrichten von Objekten sowohl vertikal als auch horizontal usw. umfassen können. Das Werkzeugleistenmenü **228** kann auch Funktionen zur Verfügung stellen, die es einem Anwender ermöglichen, Animationen auf Objekte anzuwenden, und solche Animationen können auf eine Grundform eines Elementes oder auf ein weiteres Element eines Objektes angewendet werden, wie etwa durch Animieren der Breite, Farbe, Form usw. eines Elementes, um Kippen, Drehungen oder Skalieren der Form zur Verfügung zu stellen, Elemente zu drehen oder zu kippen, usw. Natürlich kann ein Anwender Zugriff auf die Animationen und Aktionen im Zusammenhang mit jedem Element in der Anzeige erhalten und kann diese Animationen und Aktionen unter Verwendung geeigneter Dialogboxen oder anderer Programmierwerkzeuge verändern.

**[0124]** Es dürfte verständlich sein, dass ein Anwender oder Designer den Anzeige-Editor **50** verwenden kann und insbesondere mit dem Bildschirm **220** von [Fig. 15](#) koppeln kann, um schnell und einfach grafische Anzeigen zu erzeugen, die aus standardmäßigen zweidimensionalen oder sogar dreidimensionalen grafischen Elementen bestehen, welche auf dynamische Weise Messungen, Betätigungsglieder und Prozesseinrichtungen zeigen können. Es werden auch statische Elemente und Anwenderdynamos im Zusammenhang mit Steuerung und Berechnungen unterstützt und können in den Anzeigen zur Verfügung gestellt werden. Zusätzlich kann der Anwender Schichten definieren, um die Schnittstellenerfordernisse verschiedener Anwender der grafischen Anzeigen abzudecken, indem er beispielsweise Bediener-, Wartungs- und Simulationsanzeigen unter Verwendung der gleichen Elemente mit einer geringfügig verschiedenen Ansicht oder Umgebungsinformation erzeugt.

**[0125]** [Fig. 17](#) veranschaulicht einen Anzeigebildschirm **300** einer beispielhaften grafischen Anzeige in Form einer Steuerungsbediener-Anzeige, die unter Verwendung des Grafik-Editors **50** erzeugt werden kann. Der Anzeigebildschirm **300** umfasst ein Reaktorelement **310**, das mit einem Ventilelement **312** und mit einem oder mehr Pumpenelementen **314** über Verbindungselemente **316** verbunden ist, bei denen es sich in diesem Fall um Fluidstromleitungen handelt. Fluidstrom- oder Stromelemente **318** liefern Referenzen zu den Materialströmen, welche in die durch den Bildschirm **300** abgebildete Sektion der Anlage eintreten bzw. aus ihr austreten. Es dürfte verständlich sein, dass die Grafiken im Anzeigebildschirm **300** durch Auswählen und Verbinden verschiedener grafischer Elemente erzeugt werden, und somit kann der Anzeigebildschirm **300** hochauflösende Grafiken, Animationen, Aktionen, visuelle Trigger usw. enthal-

ten, die auf die oben in Bezug auf individuelle grafische Elemente beschriebene Weise entwickelt wurden. Natürlich können zusätzliche Grafiken, einschließlich Animationen, Aktionen, visuelle Trigger, Fluidstromelemente usw. auf der Grafikanzeige-Ebene hinzugefügt werden. Auf diese Weise können Animations- und Anwenderaktionen in die grafische Anzeige eingebracht werden. Beispielsweise können unter Verwendung dieser Fähigkeit statische grafische Komponenten für Prozesseinrichtungen modifiziert werden, um den Status der Einrichtung anzugeben, z.B. ob ein Motor eingeschaltet oder ausgelöst ist, den Betriebszustand oder die Stellung eines Ventils usw. Zusätzlich kann eine Animation verwendet werden, um dynamische Daten im Zusammenhang mit der Einrichtung darzustellen, wie etwa, um den Füllpegel eines Tanks zu zeigen (z.B. unter Verwendung einer Ausfülltechnik) oder den Status eines Rührers durch Anzeigenänderungen (Animationen) zu zeigen, welche eine Bewegung angeben. Ebenso können Datenanzeigeelemente **319** in dem Bildschirm **300** angeordnet werden, um Prozessdaten oder andere Daten zu zeigen, die ausserhalb der Anzeige **300** entwickelt wurden, sich aber auf den Betrieb der Komponenten in der Anzeige **300** beziehen.

**[0126]** Falls gewünscht, kann die grafische Anzeige **300** auch Dynamos oder Anwenderschnittstellenelemente **320A** und **320B** umfassen, die es einem Anwender ermöglichen, mit der Anzeige **300** auf eine Weise zu koppeln, die es dem Anwender ermöglicht, zusätzliche Informationen zu betrachten oder Aktionen in Bezug auf die Laufzeitumgebung über die Interaktion mit dem Anzeigebildschirm **300** zu unternehmen. In einigen Fällen können diese Interaktionen mit visuellen Triggern oder Aktionen implementiert werden, wie weiter oben in Bezug auf grafische Elemente beschrieben wurde. Beispielsweise kann der Knopf **320A** dem Anwender eine weitere Ansicht von Informationen über den Kühlturmaufbau zur Verfügung stellen, während der Knopf **320B** eine Faceplate-Anzeige für den Reaktor **310** zur Verfügung stellen kann. Während somit die Anzeige **300** eine Gruppe von Elementen umfasst, welche die Einrichtungen, aus denen ein Prozess oder ein Teil eines Prozesses besteht, unter Verwendung dreidimensionaler Komponenten veranschaulicht, die wie in der Vergangenheit auf der Grundlage auf Grafiken basieren können, die aus einer Anzahl von verschiedenen Quellen einschließlich z.B. In-Tools, Auto-Cad, Windows-Metafiles wie etwa Visio, Vector Drawings, JPEG- und Bitmap-Bildformaten importiert wurden, kann der Bildschirm **300** nun Animationen beinhalten, um beispielsweise Drehungen, Größeneinstellung, Skalieren, Kippen, Farbänderungen usw. auf der Elementebene interessanter und realistischere Animationen, und dadurch leichter verständliche Anzeigen zur Verfügung zu stellen.

**[0127]** Ferner können grundlegende Anwender-



schnittstellen-Komponenten wie etwa Drehknöpfe, Wählscheiben, Schiebeleisten und Knöpfe in der Anzeige **300** gezeigt und dynamisch mit Informationen oder Steuerungen im Steuersystem oder in einer anderen Laufzeitumgebung verbunden werden. Datenbetrachtungselemente oder Dynamos können auch Schlüsselparameter im Zusammenhang mit Messungen in Steuerfunktionen, Warnungen, Ereignissen usw. zur Verfügung stellen oder veranschaulichen. Beispielsweise kann ein Dynamo verwendet werden, um einen weiteren Bildschirm mit Informationen in Bezug auf einen Steuerschleifenparameter und Maschineneinheiten anzuzeigen, und kann neben dem Dynamo gezeigt werden, um den von dem Dynamo angezeigten Werten einen Zusammenhang zu verleihen. Der Status von Prozesswarnungen im Zusammenhang mit einer Steuerschleife kann in dem Dynamo durch eine Farbänderung, beispielsweise eine Änderung der Hintergrundfarbe des Steuerparameterwertes reflektiert sein. Um eine Überfüllung in der Anzeige zu beseitigen, kann die Tatsache, dass eine Schleife sich nicht in einem entworfen normalen Betriebsmodus befindet, auch durch eine Farbänderung angegeben sein. Natürlich können solche Dynamos so erstellt werden, dass sie mit jeglichem Standard konform sind.

**[0128]** Wenn bei einem Beispiel ein Bediener auf eine Anwender-interaktive Komponente oder einen Dynamo zugreift, kann ein Script oder ein anderes Programm einen weiteren Bildschirm oder eine weitere Anzeige hochziehen, wie etwa eine Faceplate-Anzeige oder eine Schalttafelanzeige, wofür Beispiele in [Fig. 18](#) veranschaulicht sind. Beispielsweise wenn auf einen Anwenderschnittstellenknopf wie etwa den in dem Bildschirm **300** gezeigten Knopf **320B** zugegriffen wird, kann dem Bediener eine Faceplate für den Reaktor **310** präsentiert werden, und der Bediener kann dann diese Faceplate verwenden, um Details bezüglich des Reaktors **310** zu modifizieren oder zu betrachten. Bei dem Beispiel von [Fig. 18](#) sind die Faceplate-Informationen **350** einer Steuerschleife (mit der Bezeichnung FIG2\_28/TC2-1) für den Reaktor **310** zugeordnet, auf die der Anwender über den Knopf **320A** in dem Bildschirm **300** zugreifen kann. Unter Verwendung des Teils **350A** der Faceplate **350** kann der Anwender den Modus (von Kaskade zu beispielsweise manuell oder automatisch) unter Verwendung der Knöpfe **352** ändern (deren Operationen durch Aktionsroutinen definiert sind), kann die aktuellen Werte der Betriebsparameter auf den Schieberanzeigen **354** betrachten, kann einen der Steuerschleife zugeordneten Sollwert unter Verwendung eines Pfeils **355** ändern, usw. Zusätzlich können dem Anwender Informationen über die Grenzen und Feineinstellparameter der Steuerschleife in der Anzeige-Sektion **350B** zur Verfügung gestellt werden, und er kann Simulationsfähigkeiten in der Sektion **356** freigeben. Falls gewünscht, kann die Sektion **350B** durch Auswählen eines der Knöpfe **358**

in der Sektion **350A** aus der Ansicht der Sektion **350A** erhalten werden. Ebenso kann der Anwender über die Steuerschleife auf weitere Informationen wie etwa Trenddaten, Diagnosedaten usw. zugreifen oder kann Steuer- und Diagnoseprogramme wie etwa auf Schleifenfeineinstellprogramme über die anderen Knöpfe **358** zugreifen und diese ausführen. Somit kann auf jegliche anderen Aktivitäten, Bildschirme, und Aktionen über die Anzeige **300** im Ansprechen auf Anwenderaktionen, die auf dem Bildschirm **300** ergriffen oder zugelassen werden, zugegriffen werden.

**[0129]** Falls das Steuersystem die Verwendung von Aliasen bei der Definition von ähnlichen Einrichtungen unterstützt, können die dynamischen Anzeigekomponenten auch so entworfen werden, dass sie dynamisches Referenzieren auf der Grundlage der auf dem Anzeigebildschirm **300** ausgewählten Einrichtung unterstützen. In solchen Fällen können vorkonfigurierte Aliase und Attribute an Stelle eines Objektetiketts oder von Grafikattributen verwendet werden, die normalerweise als Teil eines Anzeigobjektes definiert sind. Diese Aliasing-Fähigkeit unterstützt einen Grad an Flexibilität und Wiederverwendbarkeit, da ähnliche Anzeigobjekte mit verschiedenen I/O-Punkten verbunden sind und verschiedene Grafikattribute, Erscheinungsformen und Sicherheit darstellen können. Eine solche Fähigkeit kann die Notwendigkeit beseitigen, ähnliche Anzeigobjekte oder ähnliche Anzeigen für verschiedene replizierte Einrichtungen in einer Anlage neu aufzubauen. Auf diese Weise kann die gleiche grafische Anzeige verwendet werden, um verschiedene Hardware-Einheiten zu betrachten, welche die gleiche Zusammensetzung und Verwendung in einer Anlage besitzen.

**[0130]** Natürlich kann der Anzeigebildschirm **300** von [Fig. 17](#) so entworfen sein, dass er Werkzeugleisten unterstützt, einschließlich sowohl horizontaler (unter der Anzeige) und vertikaler (rechts von der Anzeige) Werkzeugleisten zusätzlich zu oder an Statt der Werkzeugleiste, die oben im Bildschirm **300** veranschaulicht ist. Falls gewünscht, können Vorgabe-Werkzeugleisten zur Verfügung gestellt werden, um die Zeit- und Datenanzeige, das Betrachten einer Warnliste mit dem Direktzugriff auf die Warnanzeige, der erforderlich ist, um die Warnung zu bestätigen oder den Alarm abzustellen, Navigation zu einer zusammenfassenden Alarmanzeige oder -menü, einem Hauptmenü, oder anderen Standardmenüs oder Anzeigen, Systemstatusanzeigen usw. zu unterstützen.

**[0131]** [Fig. 19](#) veranschaulicht einen Anzeigebildschirm **400** im Zusammenhang mit einer weiter fortgeschrittenen grafischen Anzeige, d.h. einer grafischen Anzeige mit mehr Elementen, Verbindungen, Anwenderschnittstellenaktionen, Dynamos und anderen Datenreferenzen. Insbesondere veranschaulicht der Bildschirm **400** den Betrieb einer Kalkofen-

einheit, bei der Luft und andere Brennstoffverbrennungsprodukte durch Pumpen **412**, Ventile **404** und zugeordnete Verbindungselemente zu einem Einlass **406** of eines Ofenkesseltrockners **408** gepumpt oder zugeführt werden. Ebenso wird Prozessmaterial in einem Tank **409** von einer Schichtfördervorrichtung **410** zugeführt, welche den Kalk durch den Kesseltrockner **408** transportiert. Natürlich können alle diese Elemente und die anderen in dem Bildschirm **400** veranschaulichten Vorrichtungen grafische Elemente sein, die individuell in dem Bildschirm **400** erzeugt und angeordnet werden können, wie weiter oben beschrieben wurde. Der Einlass **406** des Kesseltrockners **408** kann ein animiertes grafisches Element umfassen, das Feuer oder eine Flamme **415** als eine Animation zeigt, wenn der Trockner **408** in Betrieb ist, um einem Betrachter des Bildschirms **400** den Betrieb der Ofeneinheit klar anzuzeigen. Zusätzlich können die Temperaturen an verschiedenen Stellen in dem Trockner **408** mit Dynamos oder Temperaturanzeigeboxen **416** angegeben werden, und falls gewünscht, können Temperaturen an diesen verschiedenen Stellen über die Farbe der Feueranimation, des Trockners **408** oder auf jede andere Weise angegeben werden. Ebenso können andere Prozessparameter wie etwa Haubendruck, Ofengeschwindigkeit, Gesamtluft, Methanol, Einlass, Primär- und Sekundärluftzufuhr, Öl- und Gaszufuhr usw. mit Parameterboxen veranschaulicht werden, wie in [Fig. 19](#) gezeigt ist, wobei die Werte der Variablen in den Boxen an spezifische Referenzen in die Prozesssteuerungssystem gebunden und von diesen erhalten werden. Natürlich sind auch andere physische Komponenten der Kalkofeneinheit in dem Bildschirm **400** veranschaulicht.

**[0132]** Es können jedoch auch verschiedene andere Informationen auf dem Bildschirm **400** zur Verfügung gestellt werden, die aus anderen Datenquellen in der Laufzeitumgebung erhalten werden können, wie etwa anderen Anwendungen, die Daten aus dem Steuersystem oder aus den Vorrichtungen in dem Steuersystem verarbeiten. Solche Anwendungen könnten beispielsweise Steueranwendungen, Wartungsanwendungen, Diagnoseanwendungen, Geschäftsanwendungen usw. umfassen. Beispielsweise wird ein Trenddiagramm **420**, welches die Temperatur (in Grad) des Systems über der Zeit zeigt, in der oberen Mitte des Anzeigebildschirms **400** angezeigt und ist an eine Datenhistorieneinrichtung oder eine Trendanwendung gebunden und trägt diese Daten automatisch ab. Ebenso ist auf der oberen linken Seite des Bildschirms **400** ein Diagramm **422** angezeigt, das eine Zusammenfassung des Betriebs des Ofens **412** zur Verfügung stellt, welche von einer Geschäftsanwendung zur Verfügung gestellt werden kann, die verschiedene Geschäftsaspekte wie etwa Gewinn, Energieverbrauch, Produktionsrate usw. des Ofens verfolgt. Ferner kann auf andere Informationen zugegriffen werden, die für den Anwender nützlich sein

können, wie etwa historische Trends, Hilfeinformationen, Ofeninformationen, Steuerungsbeschränkungen, Ofenenergie und andere Informationen, indem die Anwenderschnittstellenknöpfe **424** im oberen Teil des Anzeigebildschirms **400** gewählt werden. Ferner können eine Warnungsflagge **430** oder andere Flaggen im unteren Teil des Bildschirms **400** zur Verfügung gestellt und an eine Warnanwendung gebunden werden. Natürlich können alle diese Elemente und Merkmale des Anzeigebildschirms **400** unter Verwendung der weiter oben beschriebenen Merkmale des Anzeige-Editors **50** zur Verfügung gestellt und in die Anzeige programmiert werden. Zusätzlich können diese und andere Merkmale in einer grafischen Anzeige auf jegliche gewünschte Weise kombiniert werden, um jeden gewünschten Anzeigetyp zu erzeugen. Ferner könnten jegliche gewünschten Animationen und Grafikaktionen in dem Bildschirm **400** zur Verfügung gestellt werden, um einem Anwender nützlichere oder besser verständliche visuelle Informationen zur Verfügung zu stellen, und um es dem Anwender ermöglichen, andere relevante Informationen noch unmittelbarer auf dem Bildschirm **400** zu betrachten, usw.

**[0133]** Somit können gemäß der Veranschaulichung in [Fig. 19](#) Informationen aus verschiedenen unterschiedlichen Datenquellen, einschließlich Prozesselementen, auf die von dem Controller-Subsystem zugegriffen wird, Anwendungen in Form von Wartung, Steuerung, Diagnose, Feineinstellung, und Geschäftsanwendungen, Historiendaten oder Trenddaten aus Datenbanken wie etwa einer Datenhistorieneinrichtung, oder jeglichen anderen Datenquellen in dem Anzeigebildschirm **400** angezeigt werden. Ferner kann auf andere Informationen oder Anzeigen direkt aus dem Anzeigebildschirm **400** über Anwenderlinks zu solchen Anzeigen zugegriffen werden. Auf diese Weise stellt die von dem Bildschirm **400** veranschaulichte grafische Anzeige einem Anwender mehr Informationen auf eine Weise zur Verfügung, die noch nützlicher sein kann.

**[0134]** Da der Editor **50** zum Erzeugen von beliebigen grafischen Anzeigen unter Verwendung einer gemeinsamen Gruppe von grafischen Elementen, Verbindungselementen usw. verwendet werden kann, kann der Anzeige-Editor **50** verwendet werden, um einfach ähnliche oder darauf bezogene Anzeigen zu erzeugen. Solche Anzeigen können sich auf die gleiche Gruppe von Anlagen-Hardware beziehen und zeigen, aber verschiedene Informationen für diese Hardware für verschiedene Zwecke veranschaulichen, wie etwa für Steuerungsbedienerzwecke, Simulationszwecke, Wartungszwecke usw. In diesem Fall kann eine gemeinsame oder grundlegende Anzeige erstellt werden, welche die Hardware-Elemente im Zusammenhang mit einer Anlage oder einer Sektion einer Anlage zeigt, und verschiedene Anzeigen können erzeugt werden, welche diese grundle-



gende Anzeige verwenden, aber verschiedene Informationen für verschiedene Anwender oder Anwendertypen zur Verfügung stellen.

**[0135]** Als ein Beispiel zeigen die [Fig. 20A–Fig. 20E](#) beispielhafte Anzeigen, einschließlich einer Bedieneransicht, einer Ingenieursansicht, einer Manageransicht, einer Simulationsansicht und einer Wartungsansicht, die für eine Kalkofeneinheit erzeugt wurden und die gleichen grundlegenden Hardware-Elemente in der gleichen Konfiguration und unter Verwendung der gleichen Visualisierungen zeigen, denen jedoch verschiedene Informationen für verschiedene funktionale Zwecke hinzugefügt wurden. Somit hat jede der Anzeigen der [Fig. 20A–Fig. 20E](#) das gleiche Aussehen und Gefühl, da sie mit dem gleichen Anzeige-Editor unter Verwendung der gleichen grundlegenden grafischen Elemente erstellt wurden, und daher ist es einfach, zwischen ihnen zu navigieren und dabei ein Verständnis der präsentierten Informationen und ihres Zusammenhangs mit der Hardware innerhalb der Anlage zu behalten.

**[0136]** Insbesondere veranschaulicht [Fig. 20A](#) eine Bedieneransicht **500** einer Ofeneinheit mit einer Kalk- und Schlammzufuhrquelle **502** (bei der es sich um ein Strömungselement handeln kann), die einem kalten Ende **506** eines Ofentrockners **508** Rohmaterial durch eine Pumpe **504** zuführt. Eine Fördereinrichtung **510** führt den verarbeiteten Kalk aus einem heißen Ende **512** des Ofentrockners **508** ab und legt den Kalk in einem Tank **514** ab. Eine Produktleitung bzw. ein Strömungselement **516** gibt die Menge des Produktes (Kalk) an, die den Tank **514** verlässt. Ebenso wird Brennstoff von einem Brennstoffquellenstrom-Indikator **520** durch ein Ventil **522** zur Verfügung gestellt, dessen Farbe den Betrieb des Ventils in das heiße Ende **512** des Ofentrockners **508** grafisch veranschaulichen kann. Ein Gebläse **524** pumpt Luft durch ein Röhrenverbindungselement **526** an das heiße Ende **512** des Ofentrockners **508**, wo sie mit dem Brennstoff gemischt wird. Animationen wie etwa Feuer oder Flamme **528** können an dem heißen Ende des Ofentrockners **508** gezeigt werden, um den Betriebsstatus des Ofentrockners **508** zu veranschaulichen. Ebenso zieht ein Ansauggebläse **530** Luft von dem kalten Ende **506** des Ofentrockners **508** durch ein zusätzliches Röhrensystem **532** ab und schickt diese Luft an einen Stack, wie durch ein Stack-Strömungselement **534** angegeben ist. Ferner sind verschiedene Variablen- oder Parameterboxen veranschaulicht, welche die Werte von verschiedenen Prozessparametern wie etwa Temperaturen in verschiedenen Teilen des Systems, der Geschwindigkeit des Kessels des Ofentrockners **508**, Luft- und Brennstoffströmen, usw. zeigen. Wie zu ersehen ist, sind diese Hardware und diese variablen Boxen durch die gesamten Bildschirme der [Fig. 20A–Fig. 20E](#) die Gleichen und stellen die

grundlegenden Elemente der Anzeigen zur Verfügung, die dazu führen, dass diese Anzeigen ein gleiches Aussehen und Gefühl besitzen.

**[0137]** Jeder der Bildschirme der [Fig. 20A–Fig. 20E](#) umfasst jedoch zusätzliche Informationen, Anwenderschnittstellenknöpfe und Aktionen, die auf verschiedene Funktionen innerhalb der Anlage zugeschnitten sind. Beispielsweise ist der Bildschirm **500** von [Fig. 20A](#) eine Bedieneransicht und veranschaulicht restliches Karbonat, das von einem Labortest gemessen, geschätzt und in einer Box **540** zur Verfügung gestellt wird, die Gesamtenergie und die spezifische Energie des Ofens in einer Box **542**, sowie ein Trenddiagramm **544**, das die vergangenen und zukünftigen Trends der Temperaturen, die von einer modellhaften prädiktiven Steuerroutine zur Verfügung gestellt werden und von entscheidender Wichtigkeit für die Erzielung eines optimalen Betriebs sind. Zusätzlich stellt der Bildschirm **500** verschiedene Anwenderschnittstellenknöpfe oder Dynamos **548** zur Verfügung, die es einem Anwender ermöglichen, zusätzliche Informationen über einen zugeordneten Gegenstand wie etwa Steuerschleifeninformationen, weitere Betriebsinformationen eines Elementes usw. zu betrachten.

**[0138]** Der Bildschirm **550** von [Fig. 20B](#) veranschaulicht eine Ingenieursansicht, die derjenigen von [Fig. 20A](#) sehr ähnlich ist, die aber durch gepunktete Linien **552** die Messorte zeigt, die für die Steuerung und die Signalpfade wichtig sind, um zu zeigen, wie das grundlegende Steuersystem und das modellhafte prädiktive Steuersystem zusammenwirken, um den Betrieb zu optimieren. Der Bildschirm **560** von [Fig. 20C](#) stellt eine Management-Ansicht zur Verfügung, die zwar den grundlegenden Betrieb der Ofeneinheit zeigt, aber keine Anwenderschnittstellenknöpfe zur Verfügung stellt, die den Erhalt von zusätzlichen Informationen über individuelle Komponenten oder Steuerschleifen des Systems ermöglichen. Ferner umfasst die Management-Ansicht **560** ein Diagramm **562**, welches die Gesamtenergie, spezifische Energie und restliches Karbonat (nicht-konvertiertes Einsatzmaterial) aus Labormessungen über den letzten Tag und Monat zeigt. Diese Informationen können beispielsweise aus einer Geschäftsanwendung zur Verfügung gestellt werden, die in einem Geschäftscomputer abläuft.

**[0139]** Ferner stellt [Fig. 20D](#) eine Simulationsansicht **570** zur Verfügung, die es einem Simulator ermöglicht, Parameter in einem simulierten System zu ändern und die simulierten Ergebnisse zu betrachten. Die Ansicht **570** kann beispielsweise dazu verwendet werden, Bediener zu schulen, verschiedene zukünftige Betriebsarten zu testen, usw. Wie in [Fig. 20D](#) gezeigt ist, kann der Simulator Parameter über eine oder mehr Dialogboxen **570** ändern, die dem Bediener der Simulation durch einen oder mehr

Anwenderschnittstellenknöpfe auf dem Bildschirm **570** zur Verfügung gestellt werden. Die Prozess-Simulationsansicht **570** kann in einer Offline-Situation für eine Schulung verwendet werden, oder in einer Online-Situation, um zusätzliche Informationen zur Verfügung zu stellen, die bei der Erfassung zukünftiger Probleme nützlich sein können. Die Prozess-Simulation kann aus den Grafiken erzeugt werden, da die Typen der Einrichtung und ihre Verbindungen auf Grund der Verwendung von Smart Objects oder grafischen Elementen mit Simulationsfähigkeit bei der Erstellung der Bedienergrafiken bekannt sind. Diese Smart Objects sind in der US-Veröffentlichung Nr. 2004/0153804 ausführlich beschrieben.

**[0140]** Ebenso veranschaulicht [Fig. 20E](#) eine Wartungsansicht **580**, die Informationen über oder in den Status der Einrichtungen in der Ofeneinheit zur Verfügung stellt. In der Ansicht **580** kann ein Defekt einer Einrichtung unter Verwendung von Angaben **582** und **584** über die Verfassung der Einrichtung gezeigt werden. In diesem Fall sind die Angaben **582** und **584** (von denen nicht alle in [Fig. 20E](#) beschriftet sind) Halbkreise, wobei eine Einfärbung die aktuell überwachte Verfassung der Einrichtung zeigt oder angibt. In der Ansicht **580** zeigen die Angaben **584** Einrichtungen mit einer weniger als optimalen Verfassung an. Natürlich können diese Angaben **582** und **584** aus Wartungs- oder Diagnoseanwendungen zur Verfügung gestellt. Ferner dürfte verständlich sein, dass ein Wartungstechniker den Bildschirm **580** verwenden kann, um weiter zu vertiefen und die empfohlenen Prozeduren zum Beseitigen des Problems zu finden, wenn ein Problem in dem Prozess erfasst wurde. Als ein Beispiel muss ein zugesetzter Schlammfilter ausgewaschen werden, was es erforderlich macht, die Zuführung abzustellen und den Ofen in einen Leerlaufmodus zu versetzen, was aus der Bedieneransicht **500** von [Fig. 20A](#) durchgeführt werden kann. Die Operationen können auch die Qualität der Zuführung auf übermäßigen Splitt überprüfen, was durch ein Problem des Prozesses stromaufwärts verursacht sein könnte. Wie durch dieses einfache Beispiel gezeigt ist, können somit verschiedene Interaktionen zwischen den verschiedenen Funktionen, wie etwa das Erfassen und Beheben eines Problems, einfach und leicht durch das Umschalten zwischen verschiedenen Ansichten erfolgen, welche ein gleiches Aussehen und Gefühl besitzen und daher leicht navigierbar sind. Natürlich können die Informationen in den verschiedenen Bildschirmen der [Fig. 20A–Fig. 20E](#) auf der Anzeige geschichtet und nur auf der Grundlage des Anwenders des Bildschirms gezeigt werden, falls dies gewünscht wird. Ferner können sich die für jedes Element oder Smart Object gezeigten Informationen mit dem Anwender oder der Identität des Anwenders ändern.

**[0141]** Ferner können zusätzliche Anzeigen zur Verfügung gestellt und den Anzeigen der

[Fig. 20A–Fig. 20E](#) zugeordnet werden. Diese Anzeigen können beispielsweise Steuerungskonfigurations-Anzeigen wie etwa die der [Fig. 21A](#) und [Fig. 21B](#) umfassen. Eine Steuerungsanzeige **585** von [Fig. 21A](#) veranschaulicht die Art und Weise, auf die verschiedene Steuersignale zu und von den verschiedenen Hardware-Elementen der in den [Fig. 20A–Fig. 20E](#) zu einem Controller gesendet werden, sowie die Etiketten oder variablen Namen, die diesen Steuerelementen zugeordnet sind. Eine Steuerungsanzeige **590** von [Fig. 21B](#) veranschaulicht ein Steuermodul, das die Steuerroutinen zeigt, welche eine Steuerung der Ofeneinheit der [Fig. 20A–Fig. 20E](#) implementieren. Obgleich dies nicht konkret gezeigt ist, kann die Steuerungsanzeige **585** von [Fig. 21A](#) unter Verwendung der gleichen Elemente der Bildschirme **20A–20E** erstellt werden (wobei diesen Elementen die gleichen oder verschiedene Visualisierungen zugeordnet sind) oder kann verschiedene Elemente verwenden. Da die Anzeigen **585** und **590** der [Fig. 21A](#) und [Fig. 21B](#) unter Verwendung des Anzeige-Editors **50** erstellt werden können, können diese Anzeigen auch mit jeder der oben erörterten Grafik- und Animationsfähigkeiten zur Verfügung gestellt werden. Ferner kann auf die Steuerungsanzeigen **585** und **590** beispielsweise über die Bedieneransicht **500** von [Fig. 20A](#) über einen der Anwenderschnittstellenknöpfe **548** zugegriffen werden, die darin zur Verfügung gestellt werden, um es einem Bediener zu ermöglichen, auf einfache Weise aus der Bedieneransicht **500** Zugriff auf die aktuell eingestellte Steuerung zu erhalten.

**[0142]** Wie sich aus der oben gegebenen Erörterung ergibt, können verschiedene in Beziehung stehende Anzeigen von dem Editor **50** erzeugt werden, und diese Anzeigen können auf verschiedene Weisen geschichtet werden. Insbesondere können die Anzeigen gemäß der oben gegebenen Beschreibung geschichtet werden, um in Beziehung stehende oder ähnliche Bediener-, Geschäfts-, Simulations-, Wartungs- und Ingenieursansichten zur Verfügung zu stellen, und von einer dieser Ansichten kann einfach auf eine andere zugegriffen werden. Ferner können verschiedene Anzeigen auf eine solche Weise geschichtet oder untereinander verbunden werden, dass eine bestimmte hierarchische, logische oder physische Struktur der Anlage reflektiert wird. Somit können beispielsweise Anzeigen für verschiedene physische oder geografische Bereiche einer Anlage erstellt werden. So kann etwa eine einzelne Anzeige erzeugt werden, die die Hauptbereiche der Anlage zeigt, und zusätzliche Anzeigen können erzeugt werden, welche die grundlegende Struktur zeigen, z.B. auf einer Einheit-um-Einheit-Basis in jedem Bereich der Anlage, während wieder andere Anzeigen für jede Einheit erzeugt werden können. Auf diese Weise kann ein Anwender durch die Anzeigen vertiefen, um immer mehr Details im Zusammenhang mit immer kleineren Sektionen der Anlage zu erhalten. Als ein

anderes Beispiel können Anzeigen für verschiedene Sektionen oder Teile der Anlage über Anwenderschnittstellenknöpfe verkettet oder aneinander angebracht werden, so dass der Bediener einfach in einer Vorwärts- und Rückwärtsrichtung durch die verschiedenen Anzeigen scrollen kann, um die verschiedenen, aber physisch miteinander verbundenen Sektionen der Anlage oder Komponenten der Anlage auf eine logische Weise zu betrachten.

**[0143]** Diese Arten der Schichtung von Anzeigen sind noch ausführlicher durch das Diagramm von [Fig. 22](#) veranschaulicht. Insbesondere kann eine allgemeine oder Gesamtanlagen-Übersichtsanzeige **600** eine Anzeige zur Verfügung stellen, die die grundlegende Struktur oder die Struktur der gesamten Anlage auf einer höheren Ebene veranschaulicht, obgleich sie nicht viel Detail über jede bestimmte Sektion oder jeden bestimmten Teil der Anlage zur Verfügung stellen kann. Aus der Anzeige **600** kann ein Anwender wählen (unter Verwendung der Schnittstellenknöpfe) oder zu jeder bestimmten Sektion oder zu jedem bestimmten Bereich der Anlage vertiefen, um eine oder mehr Anzeigen **602** für den Bereich A oder eine oder mehr Anzeigen **604** für den Bereich B der Anlage hochzuziehen. Wie in [Fig. 22](#) angegeben ist, können dem Bereich A n separate Anzeigen zugeordnet sein, die alle logisch in Reihe mit einander verbunden sind, um den Prozessfluss durch diesen Bereich der Anlage oder eine andere logische Struktur im Zusammenhang mit dem Bereich A der Anlage zu reflektieren. Auf die n Anzeigen für Bereich A mit den Bezeichnungen **602a**, **602b**, ... **602n** kann unter Verwendung einer Aktion vom Typ Page Forward oder Page Back zugegriffen werden, in der der Anwender von einer Anzeige zu der nächsten scrollen kann. Auf diese Weise kann ein Anwender leicht auf eine Weise durch die Anzeigen von einer Sektion des Bereichs A zu einer anderen Sektion des Bereichs A scrollen, die für den Bediener sinnvoll ist. Wie ferner in [Fig. 21](#) veranschaulicht ist, kann der Bediener vordefinierte Informationen oder andere Anzeigeelemente aus separaten Anzeigen der Anzeigen **602a–602n** erhalten. Somit kann der Bediener in der Lage sein, wenn er die Anzeige **602a** des Bereichs A betrachtet, eine weitere Anzeige **610** zu erhalten, die einen vordefinierten Trend von Schlüsselparametern in der Anzeige **602a** darstellen. Ebenso kann der Bediener bei der Betrachtung der Anzeige **602b** in der Lage sein, auf eine Liste oder ein Dokument **612** mit Hochfahr- und Abstellprozedur-Informationen zuzugreifen. Natürlich könnte der Anwender zwischen den Anzeigen **602a**, **602b** usw. in einer Vorwärts- und Rückwärtsrichtung scrollen.

**[0144]** Ebenso kann der Anwender bei einer Betrachtung des Bereichs B, der mit m Anzeigen **604a–604m** veranschaulicht ist, zwischen Anzeigen auf der gleichen Detailebene unter Verwendung der Vorwärts- und Rückwärts (Nächste und Letz-

te)-Knöpfe in den Anzeigen so scrollen, dass es für einen Bediener oder einen anderen Anwender möglich ist, auf Anzeigen zuzugreifen, die Informationen von stromaufwärts oder stromabwärts von der angezeigten Sektion des Prozesses enthalten. Zusätzlich können Dynamos oder andere Anwenderschnittstellenknöpfe zur Verfügung gestellt werden, um einen Zugriff auf eine andere Anzeige zu ermöglichen, um andere Informationen im Zusammenhang mit Elementen in der aktuellen Anzeige zu erhalten. Durch die Verwendung dieser Tools ist es möglich, eine Anzeigehierarchie zu erzeugen, aus der eine Übersichtsanzeige für einen Zugriff auf die Schlüsselanzeige in jedem Prozessbereich verwendet werden kann.

**[0145]** Wie in den [Fig. 20A–Fig. 20E](#) veranschaulicht ist, können Anzeigen zusätzlich so geschichtet sein, dass zahlreiche Anzeigen für die gleiche Sektion oder den gleichen Teil einer Anlage vorhanden sind, die aber für verschiedene funktionale Zwecke, wie etwa für Bedieneraktionen, Wartungsaktionen, Simulationsaktionen, Geschäftsaktionen, Ingenieursaktionen usw. verwendet werden. Diese separaten funktionalen Anzeigen sind in [Fig. 22](#) als unter den Anzeigen **602a**, **602b** usw. geschichtet veranschaulicht, und aus einer von ihnen kann auf eine andere zugegriffen werden, falls dies gewünscht wird. Wenn ein Bediener oder Anwender die Anzeige **602a** betrachtet, kann der Anwender daher zwischen funktionalen Anzeigen für diese Sektion der Anlage umschalten oder auf andere von ihnen wie etwa die Wartungsansicht, die Simulationsansicht, die Geschäftsansicht usw. zugreifen, falls ein solcher Zugriff gestattet ist. Natürlich stellt die Schichtung der Wartungsansicht, der Bedieneransicht, der Simulationsansicht usw. auf diese Weise unter Verwendung der gleichen grundlegenden Anzeigeelemente ein einfaches Umschalten zwischen diesen Ansichten und ein besseres Verständnis der Geschehnisse innerhalb der Anlage in Bezug auf die verschiedenen Funktionen in der Anlage zur Verfügung.

**[0146]** Um diese andere Funktionalität zu implementieren, können grafische Anzeigen erzeugt werden, die eine Angabe der vorgesehenen Rolle (oder Funktion) für die grafische Anzeige umfassen. Solche Rollen können beispielsweise eine Faceplate-Anzeige, eine Detailansicht, eine primäre Steuerungsanzeige, eine schematische Anzeige, eine Wartungsanzeige, eine Geschäftsanzeige, eine Simulationsanzeige oder jegliche andere anwenderdefinierte Rolle umfassen. Die Rolle ist Teil der grafischen Anzeige und kann verwendet werden, wenn die grafische Anzeige einem Steuermodul oder einer Hardware-Vorrichtung zugeteilt wird, um die Verwendung und einen Zugriff auf diese Anzeige während der Laufzeit zu definieren. Die Angabe der Rolle, die der grafischen Anzeige zugeteilt ist, kann dazu verwendet werden, um anzugeben, wo und welchen Sektionen des Konfigurationssystems oder der Anlage eine

bestimmte Anzeige zugeteilt werden kann. Zusätzlich können diese Rolleninformationen verwendet werden um zu bestimmen, auf welche Anzeigen ein bestimmter Anwender zugreifen kann, basierend auf der Arbeit, die von dem Anwender verrichtet wird. Beispielsweise können Steuerungsbediener nicht in der Lage sein, grafische Anzeigen zu betrachten oder darauf zuzugreifen, wenn sie als Simulations- oder Geschäftsanzeigen definiert sind.

**[0147]** Die oben beschriebene Grafikanzeige-fähigkeit kann auch verwendet werden, um spezielle Anzeigen zu erzeugen, die es ermöglichen, den Status von kritischen Einrichtungen leicht zu überwachen. Einige Beispiele für diese Anzeigetypen umfassen eine First-Out-Angabe bei einem Abschalten des Prozesses, Schwingungsüberwachung, Brennermanagement, Russgebläsebetrieb, und Sicherheitssystem-Status. Natürlich können die zugeordneten Anzeigen so strukturiert werden, dass sie die Informationen zusammenfassen, und in Fällen, in denen eine bewegliche Einrichtung eine Animation beinhaltet, z.B. ein Russgebläse, können Animationen effektiv verwendet werden, um es einem Bediener zu ermöglichen, schnell auf den Betrieb des Systems zuzugreifen oder ihn zu verstehen. Zusätzlich kann die Rechenfähigkeit der meisten Steuersysteme verwendet werden, um eine Online-Berechnung von Betriebskosten, Wirkungsgrad usw. zu implementieren, und diese Art von Informationen kann leicht in die grafische Anzeige des Bedieners eingebracht werden, so dass der Bediener diese Informationen verwenden kann, um die Prozessoperation zu verbessern. Es kann auch eine Vielfalt von Vorgehensweisen verwendet werden, um Subsysteminformationen in das Steuersystem einzubringen, so dass Standardgrafiken und Dynamos für die Erzeugung von Bedieneranzeigen verwendet werden können, um einen Zugriff auf Subsysteminformationen aus einer Anzeige einer höheren Ebene zu ermöglichen. In einigen Fällen kann ein 3D-Abtragen von Matrixwerten durchgeführt werden, um Informationen (z.B. Sheet-Gauging-Informationen) zu zeigen.

**[0148]** Zusätzlich ist anzumerken, dass die grafischen Elemente und Anzeigen vorteilhaft unter Verwendung von Vektorgrafiken implementiert werden können, wie etwa denen, die von Microsoft Avalon-Steuerungen verfügbar sind, um Flexibilität und Schnelligkeit zur Verfügung zu stellen. Die Verwendung von Vektorgrafiken stellt die Fähigkeit der Verwendung von skalierbaren Grafiken zur Verfügung, um es zu ermöglichen, dass die zu erzeugende Anzeige skaliert und in ihrer Größe an eine bestimmte Anzeigemaschine angepasst wird, um Animationen zur Verfügung zu stellen, die Skalieren verwenden.

**[0149]** Allgemein gesprochen, und wie in der ebenfalls anhängigen Anwendung mit dem Titel ... ausführlicher beschrieben ist, wird die Erzeugung jeder

grafischen Prozessanzeige und jedes darin enthaltenen grafischen Anzeigeelementes in einer jeweiligen Textbeschreibung aufgezeichnet, die in einem ersten deklarativen Format abgefasst ist. Scriptbefehle in jeder Beschreibung stellen trotz der Komplexität der zu erstellenden Grafiken einen effizienten, nicht speicherintensiven Mechanismus zum Definieren der Anzeigen zur Verfügung. Das deklarative Format und somit die Scriptbefehle können auf der Grundlage einer jeden von einer Anzahl von verschiedenen Markup-Sprachen basieren. Genauer gesagt kann auf eine Markup-Sprache auf der Grundlage von XML zurückgegriffen werden, um eine Erstellungsdefinition (auch als XML-Blob bezeichnet) für jede Anzeige und Anzeigeelement zu erstellen, und dieser XML-Blob kann verwendet werden, um diese Anzeigen und Elemente in einer Konfigurationsbibliothek oder Datenbank zu speichern, bevor sie in eine Prozess-Laufzeitumgebung heruntergeladen werden. Um fortgeschrittene Grafiken wie etwa Animationen zu unterstützen, kann die Markup-Sprache auch solche Grafiken gemäß einer Vektorgrafikmethode definieren.

**[0150]** Wie oben angegeben und noch spezifischer in der ebenfalls anhängigen Anwendung mit dem Titel "Markup Language-Based, Dynamic Process Graphics in a Process Plant User Interface" beschrieben ist, auf die ebenfalls Bezug genommen wurde, wird die dynamische Beschaffenheit der Prozessgrafiken so entworfen, dass sie die aktuellen Werte von Parametern der Verarbeitungsanlage als eine Änderung der Online- oder Simulationsbedingungen reflektieren. Hierfür können die Prozessgrafiken mit Quellen für die Daten verlinkt werden, welche solche Änderungen reflektieren. Jede auf XML basierende Beschreibung kann daher eine oder mehr Datenquellenreferenzen enthalten, die im Allgemeinen einen Datenquellort für jeden dynamischen Grafikparameter (z.B. die veränderliche Färbung des Inneren eines Tanks) identifizieren, der gemäß den Daten modifiziert werden soll. Der Datenquellort kann auch für eine spätere Spezifizierung während des Konfigurierens über die Editoren offen gelassen werden, so dass das Script ein Alias oder einen Platzhalter identifiziert, um auf die Datenquelle oder auf später zu spezifizierende Pfadinformationen zu verweisen. Da die Datenquelleninformationen und andere Charakteristiken der grafischen Prozessanzeigen (z.B. Verhaltensweisen wie etwa Ereignisbehandlung) über die XML-basierte Beschreibung spezifiziert sind, kann die XML-basierte Sprache als PGXML oder Prozessgrafiken-XML bezeichnet werden.

**[0151]** Bei Beendigung des Konfigurierens und der Entwurfsarbeit zum Definieren einer grafischen Prozessanzeige und ihrer Bestandteile kann der Konfigurierungstechniker oder ein anderer Anwender sich dafür entscheiden, die PGXML-Beschreibung als Vorbereitung für das Herunterladen der Prozessgra-



fiken auf die Bediener-Workstations oder andere Anwender-Anzeigevorrichtungen zu verarbeiten. Allgemein gesprochen wird jede PGXML-Beschreibung der grafischen Anzeigen und Anzeigeelemente verarbeitet, um Folgendes zu erzeugen: (i) Script in einem Vektorgrafikformat, das mit einer zu verwendenden Grafikerstellungsmaschine kompatibel ist, und (ii) Code mit Anweisungen, welche die Datenquelle-Referenzen und jede andere nicht-grafische Funktionalität (z.B. Verhaltensweisen) der Anzeige spezifizieren. Das Vektorgrafikformat des Scripts kann eine deklarative oder XML-basierte Sprache sein. Bei Ausführungsformen, die eine Microsoft Avalon-Anwenderschnittstellenarchitektur anwenden, kann das Vektorgrafik-Script in Microsoft XAML angegeben sein. Andere Ausführungsformen können das Open Source-Format SVG (Scalable Vector Graphics) verwenden. Der Code kann in C# oder jeder anderen geeigneten Programmiersprache angegeben sein.

**[0152]** Bei einigen Ausführungsformen werden das Vektorgrafik-Script und zugeordneter Code daraufhin kombiniert und in Dateien kompiliert, die ausführbare Befehle für die Bediener-Workstation oder eine andere Anwender-Anzeigevorrichtung angeben. Hierfür kann eine jeweilige Dynamic Link-Bibliothek (DLL)-Datei für jede grafische Prozessanzeige und jedes grafische Anzeigeelement erzeugt werden. Auf jeden Fall kann eine solche Kompilierung des Vektorgrafik-Scripts und des zugeordneten Codes vor dem Herunterladen durchgeführt werden, um die Anforderungen für die Netzwerkdatenübertragung zu minimieren.

**[0153]** Nach ihrer Erzeugung können grafische Elemente und grafische Anzeigen in einer Konfigurationsdatenbank als generische oder Schablonenobjekte gespeichert werden und können als klassenbasierte oder nicht-klassenbasierte Objekte oder Elemente gespeichert werden, bevor sie in einer Laufzeitumgebung verwendet werden. Allgemein gesprochen können die vorliegend erörterten grafischen Elemente und grafischen Anzeigemodule in Form von Anzeigemodulen sein, wie in der US-Veröffentlichung Nr. 2004/0199925 beschrieben ist, auf deren Offenbarungsgehalt hiermit Bezug genommen wird. Ein Klassenmodul (Objekt) ist ein Modul, das nicht an eine bestimmte Hardware oder Vorrichtung in einer Verarbeitungsanlage oder einem Prozesssteuerungssystem gebunden oder darin eingebunden ist, sondern ein Objekt ist, von dem andere, an die Verarbeitungsanlage oder das Prozesssteuerungssystem gebundene Objekte erstellt werden können. Allgemein gesprochen kann zum Konfigurieren einer Verarbeitungsanlage und insbesondere einer Laufzeitumgebung einer Verarbeitungsanlage eine Konfigurationsmaschine wie etwa eine der Konfigurationsanwendungen **33** verwendet werden, die in der Konfigurations-Workstation **20** von [Fig. 1](#) gezeigt sind, um die Grafikobjekte (einschließlich grafischer Ele-

mente und grafischer Anzeigen) anderen logischen und physischen Subjekte in der Verarbeitungsanlage, einschließlich logischen Subjekte wie Steuermodulen, Einrichtungsmodulen (z.B. Einheitsmodulen), Prozessmodulen usw. oder physischen Subjekte wie etwa Bediener-Workstations oder anderen Anzeigevorrichtungen zuzuordnen. In einigen Fällen können die Grafikobjekte Klassenobjekte sein und können anderen Klassenobjekten wie etwa Einrichtungsmodul-Klassenobjekten oder Steuermodul-Klassenobjekten zugeordnet sein, oder die Grafikobjekte können individuelle Objekte sein und erstellten Objekten zugeordnet sein, wie etwa Objekten, die an Einrichtungen in der Verarbeitungsanlage gebunden sind und beispielsweise auf einen Controller, eine Workstation oder eine andere Einrichtung in der Verarbeitungsanlage heruntergeladen wurden.

**[0154]** Somit können die oben genannten grafischen Elemente und grafischen Anzeigen innerhalb eines Verarbeitungsanlagen-Konfigurationssystems konfiguriert und auf Hardware innerhalb einer Verarbeitungsanlage-Laufzeitumgebung heruntergeladen werden, ähnlich der Art und Weise, auf die andere Klassenobjekte in einer Bibliothek gespeichert und in verschiedenen Sektionen oder Untersektionen des Prozesssteuerungssystems konfiguriert und daraufhin in diese heruntergeladen werden, wie in der US-Veröffentlichung No 2004/0199925 beschrieben ist. Insbesondere verwendet die Konfigurationsmaschine, die an Daten in einer Konfigurationsdatenbank wie etwa der Datenbank **28** von [Fig. 1](#) operiert, eine Konfigurationsdatenbankhierarchie wie etwa die in den [Fig. 23](#) und [Fig. 24](#) Veranschaulichte, um es einem Anwender zu ermöglichen, zu die Konfiguration der Verarbeitungsanlage zu betrachten und abzuändern. Der Konfigurationshierarchie-Bildschirm **700** von [Fig. 23](#) zeigt, dass die Konfigurationsdatenbank für das Steuersystem **702** (als DeltaV-Steuersystem bezeichnet), eine Bibliothek-Sektion **704**, eine Systemkonfiguration-Sektion **706** und eine Suchergebnisse-Sektion **708** umfasst.

**[0155]** Die Bibliothek-Sektion **704** umfasst Klassenobjekte und andere Objektdefinitionen, die nicht der Laufzeitumgebung zugeteilt oder heruntergeladen wurden, sondern statt dessen als Schablonen oder generische und ungebundene Objekte gespeichert sind. Wie in [Fig. 23](#) angegeben ist, umfasst die Bibliothek-Sektion **704** eine Steuermodule-Sektion **710**, eine Vorrichtungen-Sektion **712**, eine Prozessmodul-Sektion **714** und eine Bedienerchnittstelle-Sektion **716** sowie eine Batch-Sektion und eine Sicherheitsinstrumentensysteme-Sektion (nicht mit einer Nummer bezeichnet).

**[0156]** Während die Steuermodul-Sektion **710** nicht erweitert ist, umfasst diese Sektion der Konfigurationsdatenbank im Allgemeinen die verschiedenen Steuermodule wie etwa Steuermodulschablonen und

Steuermodul-Klassenobjekte, die für die Verarbeitungsanlage erzeugt wurden, um in verschiedenen Controllern und anderen Steuervorrichtungen in der Verarbeitungsanlage verwendet zu werden. Ferner umfasst die Vorrichtungen-Sektion **712** Angaben über Vorrichtungsobjekte, die Angaben über tatsächliche Vorrichtungen und Typen von Vorrichtungen enthalten könnten, die in der Verarbeitungsanlage verwendet werden, wie etwa Ventile, Sensoren, Controller, Workstations usw., sowie Angaben über Vorrichtungs-Klassenobjekte, Einrichtungs-Klassenobjekte usw., welche logische Elemente definieren, die verschiedenen Vorrichtungen oder verschiedener Hardware in der Verarbeitungsanlage zugeordnet sind. Ferner umfasst die Prozessmodul-Sektion **714** die verschiedenen, für die Verarbeitungsanlage erzeugten Prozessmodule, einschließlich Prozessmodulschablonen und Prozessblock-Definitionen, die für diese Prozessmodule definiert sind. Solche Prozessmodule sind ausführlich in der US-Veröffentlichung Nr. 2004/0199925 beschrieben, so dass vorliegend auf eine weitere Beschreibung verzichtet wird.

**[0157]** Wie jedoch in [Fig. 23](#) angegeben ist, speichert die Bedienerschnittstelle-Sektion **716** der Konfigurationshierarchie **700** Informationen, die den oben beschriebenen Grafikobjekten zugeordnet sind. Insbesondere umfasst eine Grafik-Sektion **720** verschiedene Kombinationsprozess-Grafikdefinitionen, Grafik-Klassenobjekte und Grafikschemata, welche die verschiedenen grafischen Elemente und grafischen Anzeigen definieren, die für das System erzeugt worden sind. Es wird angemerkt, dass die Kombinationsprozess-Grafikdefinitionssektion im Allgemeinen die grafischen Elemente umfasst, die für das System erzeugt wurden, die Prozessgrafikklassen-Sektion die grafischen Anzeigen umfasst, die auf die oben beschriebene Weise als Klassenobjekte erzeugt wurden, während die Prozessgrafikschablonen nicht-klassenbasierte grafische Anzeigen und andere Schablonenobjekte umfassen können. Die in dieser Sektion der Konfigurationsdatenbank gespeicherten Grafikobjekte sind im Allgemeinen ungebundene Objekte, die verwendet werden können, um andere Objekte zu erzeugen, und die während eines Konfigurationsprozesses verwendet werden können, um die Weise zu definieren, auf die grafische Elemente und grafische Anzeigen an die verschiedene Hardware und Software in der Verarbeitungsanlage gebunden wird. Ferner definiert eine Layout-Sektion **722** unter der Bedienerschnittstelle-Sektion **716** verschiedene Layouts für die Bedienerschnittstellen, während eine Rollen-Sektion **724** verschiedene Rollen definiert, die auf Bedienerschnittstellen implementiert werden können, sowie Rollen, die von verschiedenen der grafischen Anzeigen und Elemente erfüllt werden.

**[0158]** Somit kann die Konfigurationsdatenbank gemäß der Veranschaulichung durch die Konfigurati-

onshierarchie **700** eine Sektion umfassen, die Grafikobjekte einschließlich grafischer Elemente und grafischer Anzeigen als generische Schablonen, als Klassenobjekte oder als spezifische oder individuelle Elemente oder Anzeigen speichert, die zwar nicht an spezifische Prozesssubjekte gebunden sind, aber verwendet werden können, um Elemente oder Anzeigen zu definieren, die dann an spezifische Prozesssubjekte gebunden werden. Allgemein gesprochen kann ein Konfigurierungstechniker oder ein anderer Anwender die Hierarchie **700** verwenden, um eine Verarbeitungsanlage so zu konfigurieren, dass sie eine grafische Anzeigefähigkeit auf eine solche Weise besitzt, die im Nachfolgenden ausführlicher erörtert wird.

**[0159]** [Fig. 24](#) veranschaulicht einen Hierarchiebildschirm **730**, in dem die Systemkonfiguration-Sektion **706** erweitert wurde, um zu veranschaulichen, wie Prozessgrafikelemente und grafische Anzeigen verschiedenen logischen und physischen Subjekten der Verarbeitungsanlage als Teil einer Systemkonfiguration zugeordnet werden können. Insbesondere definiert eine Prozessgrafik-Sektion **734** unter der Systemkonfiguration-Sektion **706** verschiedene grafische Anzeigen, die im Allgemeinen definiert wurden, um auf die Verarbeitungsanlage zur Verwendung in einer Laufzeitumgebung heruntergeladen zu werden. Allgemein gesprochen werden die grafischen Anzeigen und Elemente in der Prozessgrafik-Sektion **734**, die beispielsweise Faceplate-Anzeigen, Bedienerschnittstellen, Detailanzeigen, Wartungsanzeigen, Simulationsanzeigen usw. umfassen können, auf jede Anzeigeeinheit (z.B. jede Workstation oder andere Anzeigevorrichtung) in der Verarbeitungsanlage heruntergeladen, da die Prozessgrafik-Sektion **734** auf die gesamte Systemkonfiguration angewendet wird. Grafische Elemente und Anzeigen können jedoch statt dessen bestimmten Sektionen der Systemkonfiguration einschließlich einer Steuerstrategien-Sektion **736** und einer Physikalische Netz-Sektion **740** zugeordnet werden, welche die Anzeigevorrichtungen einschränken können, auf denen diese grafischen Elemente und Anzeigen verfügbar sind.

**[0160]** Allgemein gesprochen definiert die Steuerstrategien-Sektion **736** die verschiedenen Steuerstrategien, die verschiedenen physischen und logischen Sektionen der Verarbeitungsanlage zugeteilt sind, oder Strategien, die in Bezug auf diese verschiedenen physischen und logischen Sektionen implementiert werden. Verschiedene der grafischen Anzeigen (bei denen es sich wiederum um jegliche Anzeigetypen wie etwa Faceplate-Anzeigen, Steuerungsanzeigen, Wartungsanzeigen usw. handeln kann) können spezifischen Unterteilungen oder Unterkategorien der Steuerstrategien-Sektion **736** einschließlich z.B. Bereichen (wie etwa den Bereichen mit der Bezeichnung Area\_A **742** und Area\_B **744**), Steuermodulen, die Bereichen wie etwa einer Steuermodul-Sektion

**746** zugeordnet sind, und Prozessmodule, die Bereichen wie etwa einem Prozessmodul **748** zugeteilt sind, zugeteilt werden. Somit ist gemäß der Veranschaulichung in [Fig. 24](#) eine Anzeige-Sektion **750** der Area\_A-Sektion **742** zugeordnet, die alle Anzeigen definiert, welche für und in Area\_A der Anlage verwendet werden. Obgleich dies nicht näher aufgliedert ist, könnten die Anzeigen unter die Ordner **750** verschiedene Anzeigen umfassen, die verschiedenen Hardwaregruppen in Area\_A zugeordnet sind, wie etwa die verschiedenen Anzeigen **602a–602n** von [Fig. 22](#), sowie verschiedene Unteranzeigen, auf die aus diesen Anzeigen zugegriffen werden kann, wie etwa Anzeigen **610** und **612** von [Fig. 22](#). Die für die Anzeigen definierten Rollen können angeben, wie auf diese Anzeigen aus anderen Anzeigen zugegriffen werden kann, oder von einem Bediener unter Verwendung ein Anzeigevorrichtung. Wie somit in Bezug auf [Fig. 22](#) beschrieben ist, kann ein Anwender durch die Anzeigen in der Sektion **750** scrollen, um verschiedene Teile des Area\_A der Anlage zu betrachten oder um größeres Detail in Bezug auf jeden bestimmten Bereich zu betrachten, einschließlich detaillierterer Anzeigen von bestimmten Teilen einer Anzeige auf einer höheren Ebene, die nicht in so viel Detail in der Anzeige der höhere Ebene gezeigt ist.

**[0161]** Weiter noch können Anzeigen spezifischen Steuermodulen in der Area\_A-Sektion **742** zugeteilt werden, wie durch das Anzeigen-Icon **752** unter der Steuermodul-Sektion **746** angegeben ist. In diesem Fall sind die Anzeigen **752** einer Steuermodulschleife (als LOOP bezeichnet) zugeordnet sein und können spezifische Steuerungsbediener-Anzeigen sein, die dieser bestimmten Steuerschleife zugeordnet sind. Wiederum kann eine für eine Anzeige definierte Rolle deren Rolle in einer Bedienerstation als Teil einer Steuerungsanzeige angeben. Wie ferner in Bezug auf die Area\_B-Sektion **744** veranschaulicht ist, können Anzeigen **754** allgemein dem Bereich zugeordnet sein, während Anzeigen **756** bestimmten Prozessmodulen zugeordnet sein können, die einem Bereich zugeteilt sind. In diesem Fall ist ein Anzeige-Icon **756** einem als PMOD2 bezeichneten Prozessmodul zugeordnet, bei dem es sich um ein Prozessmodul handelt, das der Einrichtung in Area\_B zugeordnet ist. Das PMOD2 kann beispielsweise ein Einheit-Modul oder ein Einrichtung-Modul oder jeder andere Typ von Prozessmodul sein.

**[0162]** Wie somit allgemein unter dem Steuerungsstrategie-Sektionen **736** angegeben ist, können Anzeigen bestimmten Steuerstrategien oder Steuerdefinitionen zugeordnet sein, da diese Steuerdefinitionen im Konfigurationssystem erzeugt werden. Wie in dem Beispiel von [Fig. 24](#) gezeigt wurde, können Anzeigen Steuerdefinitionen zugeordnet sein, die auf einer Bereich-um-Bereich-Basis definiert werden, wie etwa für die Bereich-Sektionen **742** und **744**, sowie auf einer Steuermodul-um-Steuermodul-Basis, wie etwa

für die Steuermodule **750**, und auf einer Prozessmodul-um-Prozessmodul-Basis, wie durch die Anzeige-Sektion **756** angegeben ist. Falls dies gewünscht wird, können Einrichtungsgegenständen zusätzlich Rollen zugeordnet werden, die es ermöglichen, dass auf die Anzeigen mit diesen zugeordneten Rollen dynamisch aus Einrichtungselementen zugegriffen wird. Solche Einrichtungsrollen und Anzeigen können in der Prozessgrafiken-Sektion **734** gespeichert sein.

**[0163]** Zusätzlich können Anzeigen auch tatsächliche Hardware-Elemente in der Verarbeitungsanlage zugeteilt oder zugeordnet sein, einschließlich Bediener-Workstations oder anderen Anzeigevorrichtungen. Wie unter der Physikalische Netz-Sektion **740** angegeben ist, können spezifische Anzeigen spezifischen Bediener-Workstations oder anderen Laufzeitmaschinen mit Anzeigen zugeteilt sein. Bei dem besonderen Beispiel von [Fig. 24](#) umfasst die Physikalische Netz-Sektion **740** ein Steuernetz **760**, dem Hardware wie etwa ein Controller mit der Bezeichnung CTLR1 zugeordnet ist, eine ProPlus-Station, die eine Konfigurationsstation sein kann, und eine Bediener-Workstation mit der Bezeichnung OperatorStation **762**. Die OperatorStation **762**, die eine bestimmte Workstation oder Anwenderschnittstelle innerhalb der Anlage definiert oder dieser zugeordnet ist, umfasst Warnungen und Ereignisfunktionen, Bedieneraktivitäten, eine ihr zugeordnete, kontinuierliche Historienfunktion und Simulationsaktivitäten, obgleich andere Aktivitäten oder Funktionen wie etwa Wartungsaktivitäten, Geschäftsaktivitäten usw. jeglicher bestimmten Anwenderschnittstelle zugeordnet und darauf durchgeführt werden könnten. Auf diese Weise können bestimmten Anwenderschnittstellen Rollen in der Verarbeitungsanlage zugeteilt sein, und die Anzeigen für diese Rollen können in der Anzeigevorrichtung gespeichert oder auf sie heruntergeladen werden.

**[0164]** Wie in [Fig. 24](#) angegeben ist, sind die Anzeigen **764** und **766** der Bedienerfunktion oder Bedieneraktivitäten sowie der Simulationsfunktion oder Simulationsaktivitäten zugeteilt, die von der Workstation **762** durchgeführt werden. Während dies in [Fig. 24](#) nicht gezeigt ist, können Anzeigen anderen Anzeigevorrichtungen zugeteilt sein, z.B. denen, die anderen Knoten in der Verarbeitungsanlage zugeordnet sind, um auf diesen Anzeigevorrichtungen implementiert zu werden. Ferner kann die Zuordnung von Anzeigen zu spezifischen Funktionen in den Bediener-Workstations oder anderen Laufzeit-Anzeigemaschinen die Rollen oder Funktionen bestimmen, welche diese Anzeigen in der Laufzeitumgebung durchführen können, wie etwa eine Bedienerfunktion, eine Simulationsfunktion, eine Wartungsfunktion, usw. Natürlich können diese verschiedenen Funktionen auf der gleichen Anzeigevorrichtung oder auf verschiedenen Anzeigevorrichtungen auf eine Weise

durchgeführt werden, die von dem Konfigurationssystem eingestellt wird. Auch wenn die Konfigurationshierarchie **730** Anzeigen veranschaulicht, die einer Bediener-Workstation zugeteilt sind, können Anzeigen auch anderen Typen von Anzeigevorrichtungen einschließlich Handcomputer-Anzeigevorrichtungen wie PDAs, Telefonvorrichtungen, Geschäfts-Workstations oder allen anderen gewünschten Typen von Anzeigevorrichtungen zugeteilt sein.

**[0165]** Die Konfigurationsmaschine ermöglicht es einem Anwender wie etwa einem Konfigurierungstechniker, die Konfigurationshierarchie zum Definieren und Verwalten der Art und Weise zu verwenden, auf die die grafischen Prozessanzeigen an die Verarbeitungsanlage gebunden und auf sie heruntergeladen werden. Allgemein gesprochen kann ein Anwender wie etwa ein Konfigurierungstechniker Anzeigebjekte in dem Konfigurationssystem auf die gleiche Weise erzeugen, löschen, umbenennen, zuteilen und herunterladen, wie es der Anwender in Bezug auf andere Objekte kann. Genauer gesagt ermöglicht es die Konfigurationsmaschine, dass grafische Anzeigen in die Konfiguration der Verarbeitungsanlage integriert werden, indem sie definiert, wo (welchen Vorrichtungen) diese Anzeigen für die Ausführung zugeteilt werden sollen, und die Prozess Subjekte (logische und physische) definiert, an welche diese Anzeigen während der Laufzeit gebunden sein sollen. Falls gewünscht, kann ein Konfigurationshierarchie-Bildschirm wie etwa der von [Fig. 23](#) und [Fig. 24](#) verwendet werden, um anzugeben, wo heruntergeladen werden muss, und um die Versionssteuerung an den Anzeigen zu implementieren.

**[0166]** Konfigurieren kann auf mehreren Ebenen in dem Konfigurationssystem stattfinden. Zuerst einmal kann ein Konfigurierungstechniker zuerst ein Anzeige-Klassenobjekt einem Modul-Klassenobjekt in der Bibliothek-Sektion **704** zuordnen, wodurch ein bestimmtes Anzeige-Klassenobjekt an ein bestimmtes Modul-Klassenobjekt wie etwa eine Einrichtungsmodul-Klassenobjekt, ein Bereichsmodul-Klassenobjekt usw. gebunden wird. Danach, wenn das Modul-Klassenobjekt erstellt und einem bestimmten Teil der Verarbeitungsanlage zugeteilt (oder anders konfiguriert) ist, werden die Anzeigebjekte (z.B. grafische Anzeigen), die dem erstellten Modulobjekt zugeordnet sind, an die gleiche Hardware zusammen mit dem Modulobjekt gebunden, und somit ist keine zusätzlicher Aufwand erforderlich, um eine grafische Anzeige der geeigneten Hardware innerhalb der Anlage zuzuteilen, oder dieses grafische Anzeigebjekt auf die geeignete Hardware herunterzuladen. Statt dessen folgt das grafische Anzeigebjekt in diesem Fall der Konfiguration des Modulobjektes, an das es gebunden ist. Wenn ein Anzeige-Klassenobjekt ferner gemäß der oben stehenden Beschreibung einem Modul-Klassenobjekt zugeordnet ist, können an dem Modul-Klassenobjekt vorgenommene Änderungen

automatisch dazu führen, dass diese Änderungen auf diese gleichen Elemente in den zugeordneten Anzeige-Klassenobjekten weiter gegeben werden, und diese Änderungen können dann wieder auf tatsächliche erstellt Versionen von grafischen Anzeigen weiter gegeben werden. Bei einem Beispiel, wenn eine grafische Anzeige einen Mischer (Einrichtung-Klassenobjekt) umfasst, der ein als Loop bezeichnetes Steuermodul enthält, und dieses Steuermodul ausgewählt und neu benannt wird, kann das Konfigurationssystem automatisch die Referenzen auf den Loop in der grafischen Anzeige auf den neuen Namen ändern, um dadurch sicher zu stellen, dass die Bindungen der grafischen Anzeigen auf den neuen Namen aktualisiert werden. Es kann jedoch erforderlich sein, dass die tatsächlich neu geänderten Anzeigen noch auf die Laufzeitmaschine heruntergeladen werden müssen, damit die Änderungen im Laufzeitsystem eintreten. Falls gewünscht, können Angaben, dass ein Download erforderlich ist, neben den in der Konfigurationshierarchie gespeicherten Anzeigen angeordnet werden, wie etwa durch Anordnen eines blauen Dreiecks neben den betreffenden Grafikanzeigen-Icons.

**[0167]** Andererseits können grafische Anzeigen separat oder direkt verschiedenen Komponenten in der Verarbeitungsanlage oder in dem Konfigurationssystem zugeteilt werden. Wie insbesondere in [Fig. 24](#) veranschaulicht ist, können individuelle grafische Anzeigen in der Prozessgrafik-Sektion **734** unter der Systemkonfiguration-Sektion **734** angeordnet werden. Diese Anzeigen können in dieser Sektion angeordnet werden, indem sie beispielsweise in der Bibliothek-Sektion **702** ausgewählt und dann mit Drag-and-Drop auf die Sektion **734** verschoben werden. Natürlich können andere Arten der Verschiebung von grafischen Anzeigen auf bestimmte Sektionen der Hierarchie **730** ebenso verwendet werden. Jedenfalls könnten die grafischen Anzeigen in der Sektion **734** beispielsweise Faceplate-Anzeigen oder andere klassenbasierte oder nicht-klassenbasierte Anzeigen umfassen und sind im Allgemeinen auf alle Workstations innerhalb der Anlage oder die von der Systemkonfiguration-Sektion **706** abgedeckte Sektion der Anlage herunterladbar.

**[0168]** Ein Anwender kann jedoch Anzeigen aus der Prozessgrafik-Sektion **734** und in spezifische Untersektionen verschieben, um die Identität der Workstations (oder anderer Anzeigevorrichtungen) in dem Prozesssteuerungssystem zu steuern, die Zugriff auf diese Anzeigen haben, und dadurch die Identität der Vorrichtungen zu steuern, auf denen diese Anzeigen verwendet werden können. Um eine spezifische Anzeige einer logischen oder physischen Anlagensektion zuzuteilen, kann der Anwender beispielsweise eine grafische Anzeige auswählen, die grafische Anzeige auf die spezifische Sektion der Konfigurationshierarchie **730** ziehen und fallen lassen, um dadurch



die Zuordnung zwischen dieser grafischen Anzeige und den logischen oder physischen Subjekten, denen diese Anzeige zugeordnet werden soll, zu definieren. Die Auflösetabelle für eine Anzeige kann auf der Grundlage der Sektion der Konfiguration, in der eine Anzeige fallen gelassen oder zugeordnet wird, automatisch zu dem Zeitpunkt ausgefüllt werden, an dem dieser Konfigurationsschritt stattfindet.

**[0169]** Beispielsweise können grafische Anzeigen von der Bibliothek-Sektion **702** in die Prozessgrafik-Sektion **706** gezogen werden, um die Anzeige der Laufzeitumgebung zuzuordnen. Grafische Anzeigen in z.B. der Prozessgrafik-Sektion **734** können auch in andere Sektionen verschoben werden, wie etwa in die Area\_A-Anzeigen-Sektion **750** (was diese Anzeige für die Area\_A entität und -Funktionalität spezifisch, aber allgemein verfügbar auf jeder Area\_A-Anzeigevorrichtung macht), die Area\_B-Anzeigen-Sektion **754** usw. Ebenso können grafische Anzeigen Untersektionen von logischen Subjekten zugeordnet werden, wie etwa dem als LOOP bezeichneten Steuermodul zugeordnet werden (wie durch den Anzeigendner **752** angegeben ist), was diese Anzeigen für dieses logische Subjekt spezifisch und nur auf Maschinen verfügbar macht, die diese Steuerschleifenfunktion durchführen oder als dieser zugeordnet definiert sind. Zusätzlich können Anzeigen spezifischen Anzeigevorrichtungen oder funktionalen Untersektionen von Anzeigevorrichtungen zugeteilt werden, wie etwa der durch die Bedienerfunktion in der OperatorStation **762** oder die Simulationsfunktion **770** in der OperatorStation **762** definierten. Während dies nicht spezifisch im Detail gezeigt ist, kann der Anwender auch grafische Anzeigen einem Bereich, einer Zelle, einer Einheit, einem Einrichtungsmodul, sowie verschiedenen Steuersektionen zuordnen, um die Art und Weise zu definieren, auf die die grafischen Anzeigen der Verarbeitungsanlage zugeordnet sind und in ihr verwendet werden.

**[0170]** Wenn eine grafische Anzeige einem logischen Subjekt wie etwa einer Steuerungsstrategie wie der durch eine Bereichsbezeichnung wie Area\_A oder Area\_B Definierten zugeteilt wird, wird die grafische Anzeige den physischen Schnittstellen (z.B. Workstations) zugeteilt, denen diese logischen Subjekte zugeteilt sind. Auf ähnliche Weise, wenn ein Anzeige einer Einheit, einer Zelle oder einem Einrichtungsmodul zugeteilt wird, wird die Anzeige der Workstation oder anderen Schnittstellenvorrichtungen zugeteilt, denen diese Einheit, Zelle oder Einrichtung zugeteilt ist.

**[0171]** Wenn ein Modul mehrere grafische Anzeigen umfasst, kann das Modul die Rollen für diese Anzeigen definieren oder Rollen definieren, die jede Anzeige erfüllt, die durch eine Rollenangaben definiert ist, welche den grafischen Anzeigen bei der Erzeugung der Anzeigen zugeteilt wird. Ein Zugriff auf diese gra-

fischen Anzeigen in jeglicher Anzeigevorrichtung kann dann der für die Anzeige definierten Rolle zugeordnet oder auf diese beschränkt werden. Wenn ferner eine grafische Anzeige einem Bereich oder einem Modul zugeteilt wird, kann ein Bezug auf diese grafische Anzeige als eine grafische Anzeige im Anzeigenordner des Bereichs oder Moduls, zu dem sie gehört, erscheinen. Sobald eine grafische Anzeige einem Bereich oder Modul zugeteilt wurde, folgen die Anzeigereferenzen, die zu diesem Bereich oder Modul gehören, und somit diese Anzeige, dem Bereich oder Modul während jeder Zuordnung oder Verschiebung des Bereichs oder Moduls. Wenn andererseits ein Bereich oder Modul einer Workstation zugeteilt wird, wird die Anzeige, die zu diesem Bereich oder Modul gehört, automatisch dem Bediener-Subsystem oder der Bedienerfunktion dieser Workstation gemäß der Definition der Rolle dieser Anzeige zugeordnet. Natürlich können individuelle grafische Anzeigen direkt dem Bediener-Subsystem einer Anzeigevorrichtung zugeteilt werden, indem sie direkt in oder unter der bestimmten Schnittstellenvorrichtung oder dem Subsystem der Schnittstellenvorrichtung in der Physikalische Netz-Sektion **740** der Konfigurationshierarchie angeordnet werden.

**[0172]** Somit ist ersichtlich, dass grafische Anzeigen individuellen Knoten, Steuerbereichen, Steuermodulen, Prozessmodulen oder anderen logischen Steuerungssubjekten, sowie individuellen Anzeigevorrichtungen und Subsystemen von diesen zugeteilt werden können, indem sie durch Drag-and-Drop in jedes dieser Subjekte in dem Konfigurationsbildschirm **730** verschoben werden. Falls gewünscht, kann die gleiche grafische Anzeige natürlich mehreren Stationen zugeteilt werden, und der Status jeder Anzeige (heruntergeladen oder nur zugeteilt) kann durch einen Icon angegeben werden, der für jeden Anzeigenamen in den verschiedenen folders in der Hierarchie **730** zur Verfügung gestellt wird. Falls gewünscht, können andere Icons in der Hierarchie **730** verwendet werden, um anzugeben, wenn grafische Anzeigen im Speicher verriegelt sind, oder ob sie im virtuellen Speicher auf der Platte geführt werden können.

**[0173]** Allgemein gesprochen bestehen grafische Anzeigen aus einem Anzeigehalt (z.B. einem XAML-Script, das die Art und Weise definiert, auf welche die Anzeige visuell auf einem Anzeigebildschirm erscheint), lokalen Tabellen und Referenzen, bei denen es sich um Referenzen auf andere Anzeigen sowie um Referenzen auf Laufzeitparameter wie etwa Steuerparameter usw., Referenzen auf Anzeigesteuern, Laufzeit-Aliases (von denen einige sich an die Laufzeit binden können), Lokaltabellen-Referenzen und Globaltabellen-Referenzen handeln kann. Allgemein gesprochen behandelt die Konfigurationsmaschine den grafischen Teil des grafischen Elementes (den XAML) separat von den Refe-

renzen, was es einfacher macht, grafische Anzeigen zu behandeln und zu verwenden. Insbesondere kann das Konfigurationssystem eine grafische Anzeige während des Konfigurationsprozess durch Ausfüllen der Referenzen binden, die keine dynamischen Alias verwenden, auf der Grundlage der in der Konfigurationshierarchie durchgeführten Zuordnungen. Wenn somit ein Anwender beispielsweise eine grafische Anzeige aus der Prozessgrafik-Sektion **734** oder aus der Bibliothek-Sektion **704** der Konfigurationshierarchie **730** in einen bestimmten Bereich oder in ein bestimmtes Steuermodul zieht, wird eine Bindung in der Grafik automatisch durchgeführt, um die Referenzen in der grafischen Anzeige an die spezifischen Elemente zu binden, die diesem Bereich oder Modul zugeordnet sind. Auf diese Weise können heruntergeladene Bindungen automatisch auf der Grundlage der Konfiguration durchgeführt werden, die von dem Anwender vorgenommen wird, wodurch die Menge der Arbeit, die der Anwender zum Spezifizieren dieser Bindungen durchführen muss, verringert wird.

**[0174]** Falls ferner unaufgelösten Bindungen beim Konfigurationsschritt vorhanden sind, kann die Konfigurationmaschine den Anwender auffordern, Informationen zum Auflösen der Bindungen zur Verfügung zu stellen, wie etwa die Bindungen oder die Referenzen zu ändern. Falls gewünscht, können ferner Bulkeditier-Tools wie etwa ein Spreadsheet-Programm wie Microsoft Excel<sup>®</sup> verwendet werden, um einem Konfigurierungstechniker beim Ausfüllen oder Definieren von Auflösungs- oder Bindungsparametern während des Konfigurationsprozesses zu helfen. Auf jeden Fall füllen diese Konfigurationsschritte die in Bezug auf [Fig. 13](#) erörterte Auflösungstabelle **208** aus, was es ermöglicht, dass die grafischen Anzeigen in einer Laufzeitumgebung verwendet werden. Somit kann der Anwender Bindungen und Downloads von grafischen Anzeigen (and grafischen Elementen) im Allgemeinen auf eine gleiche Weise durchführen, wie es der Anwender in Bezug auf die Prozessmodule durchführt, die in der US-Veröffentlichung Nr. 2004/0199925 beschrieben sind.

**[0175]** Natürlich kann der Anwender oder Konfigurierungstechniker Zuordnungen von grafischen Anzeigen durch Klicken auf die grafische Anzeige und ihr Entfernen aus der Hierarchie **730** aufheben, oder anderweitig durch den Einsatz einer Dialogbox, um zu veranlassen, dass eine Zuteilung dieser grafischen Anzeige aufgehoben wird. Ferner kann ein Anwender die Art und Weise betrachten, auf die eine bestimmte grafische Anzeige oder Gruppe von grafischen Anzeigen zugeteilt ist, und solche Informationen können beispielsweise von einer Dialogbox **800** zur Verfügung gestellt werden, die in [Fig. 25](#) gezeigt ist. Die Dialogbox **800** von [Fig. 25](#) gibt die Art und Weise an, wie jede von zwei Anzeigen, Display1 und Display2, in der Verarbeitungsanlage konfiguriert ist.

Insbesondere die Dialogbox **800** gibt an, wo jede der Anzeigen verwendet wird (z.B. wird Display1 in einem als Area\_A bezeichneten Anlagebereich und in dem als Mod1 bezeichneten Steuermodul verwendet), die physische Station, der sie zugeteilt ist (z.B. ist Display1 einer als Oper1 bezeichneten Bedienerstation zugeteilt), und die Rolle, welche die Anzeige erfüllt (z.B. ist Display1 bei jeder ihrer Verwendungen eine primäre Anzeige).

**[0176]** Es dürfte verständlich sein, dass die Anzeigerollen auch auf jede gewünschte Weise geändert werden können, und dass die Rollen verwendet werden können, um einen Teil des Konfigurationssystems anzugeben, dem die grafische Anzeige zugeteilt werden kann, oder wer die grafische Anzeige verwenden kann, d.h. für welche Funktion er sie verwenden kann. Graphische Anzeigen können verwendet werden, um mehrere Rollen zu erfüllen, falls dies gewünscht wird. Wenn ein Anwender einen Bereich oder ein Modul irgend einer Art einer Workstation oder einer anderen Anzeigevorrichtung zuteilt, werden natürlich gemäß der oben stehenden Bemerkungen Anzeigen, die diesem Bereich oder Modul zugeordnet sind, auch der Workstation zugeteilt, um sicher zu stellen, dass die grafische Unterstützung für diesen Bereich oder ein anderes Modul auf der Workstation zur Verfügung gestellt wird. Wenn eine grafische Anzeige einer Workstation oder einer anderen Anzeigevorrichtung zugeteilt wird, kann die grafische Anzeige auch einem Subsystem der Workstation auf der Grundlage der Rolle dieser grafischen Anzeige zugeteilt werden, wie etwa, ob die grafische Anzeige eine Simulationsanzeige, eine Bedieneranzeige usw. ist. Zusätzlich kann der Anwender grafische Anzeigen auf bestimmte Subsysteme einer Workstation oder auf eine andere Anzeigevorrichtung ziehen und fallen lassen, und falls die Rolle nicht ordnungsgemäß zugeteilt wurde, kann das Konfigurationssystem eine Dialogbox präsentieren, um dem Anwender anzuzeigen, dass eine Falschzuordnung stattfindet, oder um es dem Anwender zu ermöglichen, die Rollen der grafischen Anzeige zu ändern.

**[0177]** Nach dem Durchführen von geeigneten oder gewünschten Zuordnungen unter Verwendung der Konfigurationmaschine und der Konfigurationshierarchie **730** der [Fig. 23](#) und [Fig. 24](#) kann der Anwender/Bediener einen tatsächlichen Download der grafischen Anzeigen durchführen, was dazu führt, dass die grafischen Anzeigen tatsächlich auf die geeigneten Bediener-Workstations heruntergeladen werden, wie in der Konfigurationshierarchie **730** für die in dem Konfigurationssystem definierten Zwecke und Rollen definiert ist. Während dieses Herunterladens werden die Bindungen in den Anzeigen auf der Grundlage der Zuordnungen zu den Bereichen, Schleifen, Steuermodulen, Prozessmodulen usw. sowie zu den Bediener-Workstations oder anderen Anzeigevorrichtungen aufgelöst.

**[0178]** Falls gewünscht, kann es die Suchergebnisse-Sektion **708** der Konfigurationshierarchie (in [Fig. 23](#) gezeigt) einem Anwender ermöglichen, Suchen nach jeglichen individuellen grafischen Anzeigen oder grafischen Elemente durchzuführen, die in grafischen Anzeigen in der gesamten Verarbeitungsanlage oder in dem gesamten Konfigurationssystem verwendet werden. Insbesondere kann die Konfigurationsmaschine eine Suchmaschine umfassen, die bei ihrem Einsatz durch den Anwender Suchen nach spezifischen grafischen Anzeigen oder grafischen Elementen durchführen kann, und die Ergebnisse solcher Suchen in der Suchergebnisse-Sektion **708** speichern kann, um dem Anwender eine vollständige Auflistung der Verwendungen eines jeglichen bestimmten grafischen Elementes oder einer jeglichen bestimmten grafischen Anzeige zur Verfügung zu stellen. Solche Suchen sind nützlich, um es Anwendern, die Änderungen vornehmen, zu ermöglichen, einzusehen, welche grafischen Anzeigen geändert werden müssen, oder welche Downloads stattfinden müssten, um diese Änderungen zu implementieren.

**[0179]** Bei ihrer Implementierung kann jede der vorliegend beschriebenen Softwares in jeglichem computerlesbaren Speicher wie etwa auf einer Magnetplatte, einer Laserplatte oder einem anderen Speichermedium, in einem RAM oder ROM eines Computers oder Prozessors usw. gespeichert werden. Ebenso können diese Softwares an einen Anwender, eine Verarbeitungsanlage oder eine Bediener-Workstation unter Verwendung jeglicher bekannten oder gewünschten Liefermethode geliefert werden, beispielsweise auf einer computerlesbaren Platte oder Diskette oder einem anderen transportablen Computerspeichermechanismus, oder über einen Kommunikationskanal wie etwa eine Telefonleitung, das Internet, das World Wide Web, jedes andere Lokalbereichsnetz oder Weitbereichsnetz usw. (wobei eine solche Lieferung als gleichwertig oder austauschbar mit dem Zurverfügungstellen einer solchen Software über ein transportables Speichermedium angesehen wird). Ferner kann diese Software direkt, ohne Modulation oder Verschlüsselung oder unter Verwendung einer jeglichen geeigneten Modulations-Trägerwelle und/oder Verschlüsselungsmethod moduliert und/oder verschlüsselt zur Verfügung gestellt werden, bevor sie über einen Kommunikationskanal übertragen wird.

**[0180]** Auch wenn die vorliegende Erfindung unter Bezugnahme auf konkrete Beispiele beschrieben wurde, die nur veranschaulichend und nicht als Einschränkung für die Erfindung gedacht sind, dürfte es für den Durchschnittsfachmann auf diesem technischen Gebiet offensichtlich sein, dass Änderungen, Hinzufügungen oder Weglassungen an den offengelegten Ausführungsformen vorgenommen werden können, ohne von dem Grundgedanken und dem Schutzzumfang der Erfindung abzuweichen.

**[0181]** Es wird eine Grafikanzeigeunterstützung innerhalb eines Konfigurations-, Überwachungs- und Simulationssystems einer Verarbeitungsanlage zur Verfügung gestellt, um die Erzeugung von grafischen Anzeigen auf eine Art und Weise zu ermöglichen, bei der sie einander in der Laufzeitumgebung zugeordnet sind. Insbesondere kann ein einzelner Grafikanzeige-Editor verwendet werden, um verschiedene unter einander in Beziehung stehende grafische Anzeigen zu erzeugen, auf die beispielsweise aus einer Laufzeitumgebung auf einander zugegriffen werden kann, um weitere Informationen über ein Prozess-Subjekt in einer der grafischen Anzeigen zur Verfügung zu stellen, um durch benachbarte Sektionen einer Verarbeitungsanlage zu scrollen, oder um verschiedene Anzeigen für verschiedene Funktionen innerhalb der Verarbeitungsanlage wie etwa für eine Bediener-Betrachtungsfunktion, eine Simulationsfunktion und eine Wartungsfunktion zur Verfügung zu stellen. Da der gleiche Grafik-Editor verwendet wird, um die grafischen Anzeigen zu erzeugen, können die resultierenden grafischen Anzeigen ein gleiches Aussehen und Gefühl haben und können innerhalb der Anlage auf eine im Allgemeinen gleiche Weise an die Laufzeitumgebung gebunden sein, wodurch die Zeit verkürzt wird, die erforderlich ist, um in einer Verarbeitungsanlage verwendete grafische Prozessanzeigen zu konfigurieren und zu erzeugen.

### Patentansprüche

1. Grafikanzeige-Editor zur Verwendung bei der Erzeugung einer Anwenderschnittstelle-Anzeige, welche den Betrieb von einem oder mehr Subjekten in der Verarbeitungsanlage darstellt, wobei der Grafikanzeige-Editor aufweist:

- eine Bibliothek von Grafikobjekten, wobei jedes Grafikobjekt eine visuelle Darstellung eines physischen oder eines logischen Subjektes in der Verarbeitungsanlage umfasst;
- eine grafikbasierte Editor-Canvas-Routine, die es einem Anwender ermöglicht, eine ausführbare grafische Anzeige durch Anordnen von einer oder mehr visuellen Darstellungen der Grafikobjekte aus der Bibliothek von Grafikobjekten auf einem Editier-Canvas zu definieren, um eine Weise zu definieren, auf die die eine oder mehr visuellen Darstellungen der Grafikobjekte einem Anwender auf einer Anzeigevorrichtung während der Ausführung der grafischen Anzeige angezeigt werden;
- eine Eigenschaftendefinitions-Canvasroutine, die dazu ausgelegt ist, es einem Anwender zu ermöglichen, ein Eigenschaft zu definieren, die mindestens einem der Vielzahl von Grafikobjekten zugeordnet ist;
- eine Bindungsdefinitions-Routine, die dazu ausgelegt ist, es einem Anwender zu ermöglichen, eine Bindung zwischen der Eigenschaft und einer Laufzeitumgebung in der Verarbeitungsanlage zu spezifi-

zieren; und eine Aktionsdefinitions-Routine, die dazu ausgelegt ist, es einem Anwender zu ermöglichen, mit der grafischen Anzeige zu interagieren, um eine Funktion unter Verwendung der grafischen Anzeige auszuführen.

2. Grafikanzeige-Editor nach Anspruch 1, wobei es die Aktionsdefinitions-Routine einem Anwender ermöglicht, eine Verbindung mit einer weiteren grafischen Anzeige herzustellen.

3. Grafikanzeige-Editor nach Anspruch 2, wobei die weitere grafische Anzeige Informationen über ein Subjekt in der grafischen Anzeige zur Verfügung stellt.

4. Grafikanzeige-Editor nach Anspruch 3, wobei die weitere grafische Anzeige eine Faceplate-Anzeige ist.

5. Grafikanzeige-Editor nach Anspruch 2, wobei die grafische Anzeige eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die weitere grafische Anzeige eine zweite und davon verschiedene Sektion der Verarbeitungsanlage veranschaulicht.

6. Grafikanzeige-Editor nach Anspruch 2, wobei die grafische Anzeige eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die weitere grafische Anzeige eine Untersektion der ersten Sektion der Verarbeitungsanlage veranschaulicht.

7. Grafikanzeige-Editor nach Anspruch 6, wobei die grafische Anzeige die erste Sektion der Verarbeitungsanlage auf einer ersten Detailebene veranschaulicht, und die weitere grafische Anzeige die Untersektion der ersten Sektion der Verarbeitungsanlage auf einer Ebene mit größerem Detail als die erste Detailebene veranschaulicht.

8. Grafikanzeige-Editor nach Anspruch 2, wobei die grafische Anzeige eine erste Funktionsansicht einer Sektion der Verarbeitungsanlage zur Verfügung stellt, und die weitere Anzeige eine zweite Funktionsansicht der Sektion der Verarbeitungsanlage zur Verfügung stellt.

9. Grafikanzeige-Editor nach Anspruch 8, wobei die erste Funktionsansicht eine Steuerungsbediener-Ansicht ist, und die zweite Funktionsansicht eine von einer Wartungsansicht oder einer Geschäftsansicht oder einer Simulationsansicht oder einer Ingenieursansicht ist.

10. Grafikanzeige-Editor nach Anspruch 9, wobei die Steuerungsbediener-Ansicht Prozesswerte im Zusammenhang mit einem oder mehr physischen Subjekten während des Betriebs der Verarbeitungsanlage veranschaulicht.

11. Grafikanzeige-Editor nach Anspruch 8, wobei die erste Funktionsansicht eine Wartungsansicht ist, und die zweite Funktionsansicht eine von einer Steuerungsansicht oder einer Geschäftsansicht oder einer Simulationsansicht oder einer Ingenieursansicht ist.

12. Grafikanzeige-Editor nach Anspruch 11, wobei die Wartungsansicht eine Angabe einer Verfassung oder eines Status eines physischen Subjektes in der Verarbeitungsanlage veranschaulicht.

13. Grafikanzeige-Editor nach Anspruch 1, wobei es die Aktionsdefinitions-Routine einem Anwender ermöglicht, eine Verbindung mit einer Steuerroutine-Ansicht herzustellen, die eine in der Verarbeitungsanlage implementierte Steuerroutine veranschaulicht.

14. Grafikanzeige-Editor nach Anspruch 1, welcher ferner eine Definitionsroutine umfasst, die dazu ausgelegt ist, es einem Anwender zu ermöglichen, eine Routine zu definieren, welche die visuelle Darstellung eines der grafischen Objekte auf der Grundlage der Eigenschaft während der Ausführung der grafischen Anzeige durchführt.

15. Grafikanzeige-Editor nach Anspruch 14, wobei es die Definitionsroutine einem Anwender ermöglicht, die Routine als eine Animationsroutine zu definieren, welche die visuelle Darstellung des einen der Grafikobjekte animiert.

16. Gruppe von grafischen Anzeigen, die auf einer oder mehr Anzeigevorrichtungen ausführbar sind, um den Betrieb von einem oder mehr Subjekten innerhalb einer Verarbeitungsanlage visuell darzustellen, wobei die Gruppe von grafischen Anzeigen aufweist:

ein erste grafische Anzeige und ein zweite grafische Anzeige, wobei die erste grafische Anzeige aufweist: einen Anzeigebereich;

ein oder mehr visuell untereinander verbundene Grafikobjekte, wobei jedes der Grafikobjekte eine visuelle Darstellung eines in der Verarbeitungsanlage befindlichen und in dem Anzeigebereich abgebildeten physischen oder logischen Subjektes umfasst; eine Eigenschaftendefinition, welche ein Eigenschaft definiert, die mindestens einem der Vielzahl von Grafikobjekten zugeordnet ist;

ein Bindungsdefinition, welche eine Bindung zwischen der Eigenschaft und einer Laufzeitumgebung in der Verarbeitungsanlage spezifiziert; und eine visuelle Verbindung in dem Anzeigebereich, welche einen Anwender in die Lage versetzt, mit der ersten grafischen Anzeige zu interagieren, um eine Verbindung mit der zweiten grafischen Anzeige herzustellen.

17. Gruppe von grafischen Anzeigen nach An-



spruch 16, wobei die zweite grafische Anzeige Informationen über ein Subjekt in der ersten grafischen Anzeige zur Verfügung stellt.

18. Gruppe von grafischen Anzeigen nach Anspruch 17, wobei die zweite grafische Anzeige eine Faceplate-Anzeige ist.

19. Gruppe von grafischen Anzeigen nach Anspruch 16, wobei die erste grafische Anzeige eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die zweite grafische Anzeige eine zweite und davon verschiedene Sektion der Verarbeitungsanlage veranschaulicht.

20. Gruppe von grafischen Anzeigen nach Anspruch 16, wobei die erste grafische Anzeige eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die zweite grafische Anzeige eine Untersektion der ersten Sektion der Verarbeitungsanlage veranschaulicht.

21. Gruppe von grafischen Anzeigen nach Anspruch 20, wobei die erste grafische Anzeige die erste Sektion der Verarbeitungsanlage auf einer ersten Detailebene veranschaulicht, und die zweite grafische Anzeige die Untersektion der ersten Sektion der Verarbeitungsanlage auf einer Ebene mit größerem Detail als die erste Detailebene veranschaulicht.

22. Gruppe von grafischen Anzeigen nach Anspruch 16, wobei die erste grafische Anzeige eine erste Funktionsansicht einer Sektion der Verarbeitungsanlage zur Verfügung stellt, und die zweite grafische Anzeige eine zweite Funktionsansicht der Sektion der Verarbeitungsanlage zur Verfügung stellt.

23. Gruppe von grafischen Anzeigen nach Anspruch 22, wobei die erste Funktionsansicht eine Steuerungsbediener-Ansicht ist, und die zweite Funktionsansicht eine von einer Wartungsansicht oder einer Geschäftsansicht oder einer Simulationsansicht oder einer Ingenieursansicht ist.

24. Gruppe von grafischen Anzeigen nach Anspruch 22, wobei die erste Funktionsansicht eine Wartungsansicht ist, und die zweite Funktionsansicht eine von einer Steuerungsansicht oder einer Geschäftsansicht oder einer Simulationsansicht oder einer Ingenieursansicht ist.

25. Gruppe von grafischen Anzeigen nach Anspruch 24, wobei die Wartungsansicht eine Angabe einer Verfassung oder eines Status eines physischen Subjektes in der Verarbeitungsanlage veranschaulicht.

26. Gruppe von grafischen Anzeigen nach Anspruch 16, wobei die erste grafische Anzeige eine Bedieneransicht ist, welche den Betrieb von einem

oder mehr physischen Subjekten in der Verarbeitungsanlage veranschaulicht, und die zweite grafische Anzeige eine Steuermodulanzeige ist, welche eine Steuerroutine für das eine oder die mehr physischen Subjekte in der Verarbeitungsanlage veranschaulicht.

27. Prozesssteuerungs- und Konfigurationssystem zur Verwendung in einer Verarbeitungsanlage, welches aufweist:

eine Vielzahl von physischen und logischen Prozesssubjekten, die zusammen operieren, um einen Prozess durchzuführen;

einen Prozess-Controller, der kommunikationsmäßig mit der Vielzahl von physischen und logischen Prozesssubjekten verbunden ist;

eine oder mehr Steuerroutinen, die auf dem Prozess-Controller implementiert werden, um den Betrieb der Vielzahl von physischen und logischen Prozesssubjekten zu steuern;

eine Anzeigevorrichtung, welche einen Prozessor und einen Anzeigebildschirm aufweist; und

eine Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen, wobei jede der grafischen Anzeigen auf der Anzeigevorrichtung ausführbar ist, um den Betrieb von einem oder mehr der Prozesssubjekte in der Verarbeitungsanlage auf dem Anzeigebildschirm visuell darzustellen, wobei auf eine der Gruppen von unter einander in Beziehung stehenden grafischen Anzeigen aus einer anderen der Gruppe von grafischen Anzeigen sequentiell zugegriffen werden kann.

28. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 27, welches ferner einen Grafikanzeige-Editor aufweist, der dazu ausgelegt ist, jede der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen zu erzeugen.

29. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 27, wobei eine der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen einen Anzeigebereich, ein oder mehr visuell untereinander verbundene Grafikobjekte, wobei die Grafikobjekte eine visuelle Darstellung eines physischen oder eines logischen Subjektes in der Verarbeitungsanlage umfassen, eine Eigenschaftendefinition, welche eine Eigenschaft definiert, die mindestens einem der Vielzahl von Grafikobjekten zugeordnet ist, und eine Bindungsdefinition, welche eine Bindung zwischen der Eigenschaft und einer Laufzeitumgebung in der Verarbeitungsanlage spezifiziert, umfasst.

30. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 29, wobei die eine der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine visuelle Verbindung in dem Anzeigebereich umfasst, die einen Anwender in die Lage versetzt, mit der einen der Gruppe von unter

einander in Beziehung stehenden grafischen Anzeigen zu interagieren, um eine Verbindung mit der anderen der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen herzustellen.

31. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 30, wobei die andere der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen Informationen über ein Subjekt in der einen der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen zur Verfügung stellt.

32. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 27, wobei die eine der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die andere der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine zweite und davon verschiedene Sektion der Verarbeitungsanlage veranschaulicht.

33. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 27, wobei die eine der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die andere der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine Untersektion der ersten Sektion der Verarbeitungsanlage veranschaulicht.

34. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 27, wobei die eine der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine erste Funktionsansicht einer Sektion der Verarbeitungsanlage veranschaulicht, und die andere der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen eine zweite Funktionsansicht der Sektion der Verarbeitungsanlage veranschaulicht.

35. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 34, wobei die erste Funktionsansicht eine Steuerungsbediener-Ansicht ist, und die zweite Funktionsansicht eine von einer Wartungsansicht oder einer Geschäftsansicht oder einer Simulationsansicht oder einer Ingenieursansicht ist.

36. Prozesssteuerungs- und Konfigurationssystem zur Verwendung in einer Verarbeitungsanlage, welches aufweist:  
eine Vielzahl von physischen und logischen Prozesssubjekten, die zusammen operieren, um einen Prozess durchzuführen;  
einen Prozess-Controller, der kommunikationsmäßig mit der Vielzahl von physischen und logischen Prozesssubjekten verbunden ist;  
eine oder mehr Steuerungen, die auf dem Prozess-Controller implementiert werden, um den Be-

trieb der Vielzahl von physischen und logischen Prozesssubjekten zu steuern;  
eine Anzeigevorrichtung mit einem Prozessor und einem Anzeigebildschirm; und  
einen Grafikanzeige-Editor, der einen Anwender in die Lage versetzt, eine Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen zu erzeugen, wobei jede der Gruppe von unter einander in Beziehung stehenden grafischen Anzeigen auf der Anzeigevorrichtung ausführbar ist, um den Betrieb von einem oder mehr der Prozesssubjekte in der Verarbeitungsanlage auf dem Anzeigebildschirm visuell darzustellen.

37. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 36, wobei der Grafikanzeige-Editor dazu ausgelegt ist, eine Beziehung zwischen den grafischen Anzeigen herzustellen, indem er es einem Anwender ermöglicht, eine Weise zu spezifizieren, auf welche eine der grafischen Anzeigen zu einer anderen der grafischen Anzeigen in einer Beziehung steht.

38. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 37, wobei die andere der grafischen Anzeigen Informationen über ein Subjekt in einer anderen der grafischen Anzeigen zur Verfügung stellt.

39. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 37, wobei die eine der grafischen Anzeigen eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die andere der grafischen Anzeigen eine zweite und davon verschiedene Sektion der Verarbeitungsanlage benachbart zu der ersten Sektion der Verarbeitungsanlage veranschaulicht.

40. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 37, wobei die eine der grafischen Anzeigen eine erste Sektion der Verarbeitungsanlage veranschaulicht, und die andere der grafischen Anzeigen eine Untersektion der ersten Sektion der Verarbeitungsanlage veranschaulicht.

41. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 36, wobei der Grafikanzeige-Editor dazu ausgelegt ist, eine Beziehung zwischen den grafischen Anzeigen herzustellen, indem er es einem Anwender ermöglicht, eine erste der grafischen Anzeigen als eine erste Funktionsanzeige zu erzeugen und eine zweite der grafischen Anzeigen als eine zweite Funktionsanzeige zu erzeugen.

42. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 41, wobei die erste Funktionsanzeige eine Steuerungsbediener-Anzeige ist, und die zweite Funktionsanzeige eine von einer Wartungsanzeige oder einer Geschäftsanzeige oder einer Simulationsanzeige oder einer Ingenieursanzeige ist.

43. Prozesssteuerungs- und Konfigurationssystem nach Anspruch 41, wobei die erste Funktionsanzeige eine Wartungsanzeige ist, und die zweite Funktionsanzeige eine von einer Steuerungsanzeige oder einer Geschäftsanzeige oder einer Simulationsanzeige oder einer Ingenieursanzeige ist.

Es folgen 27 Blatt Zeichnungen

Anhängende Zeichnungen

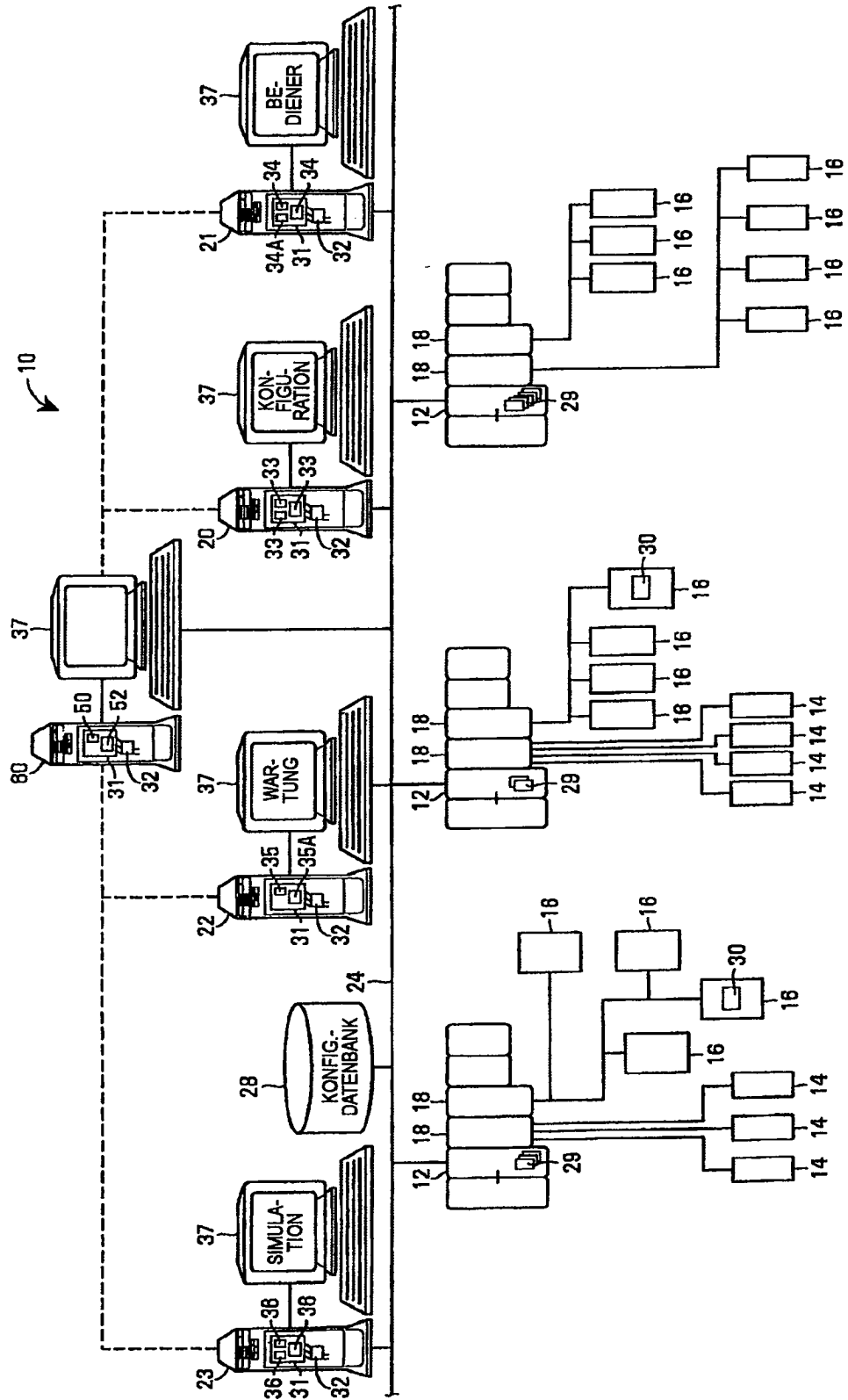
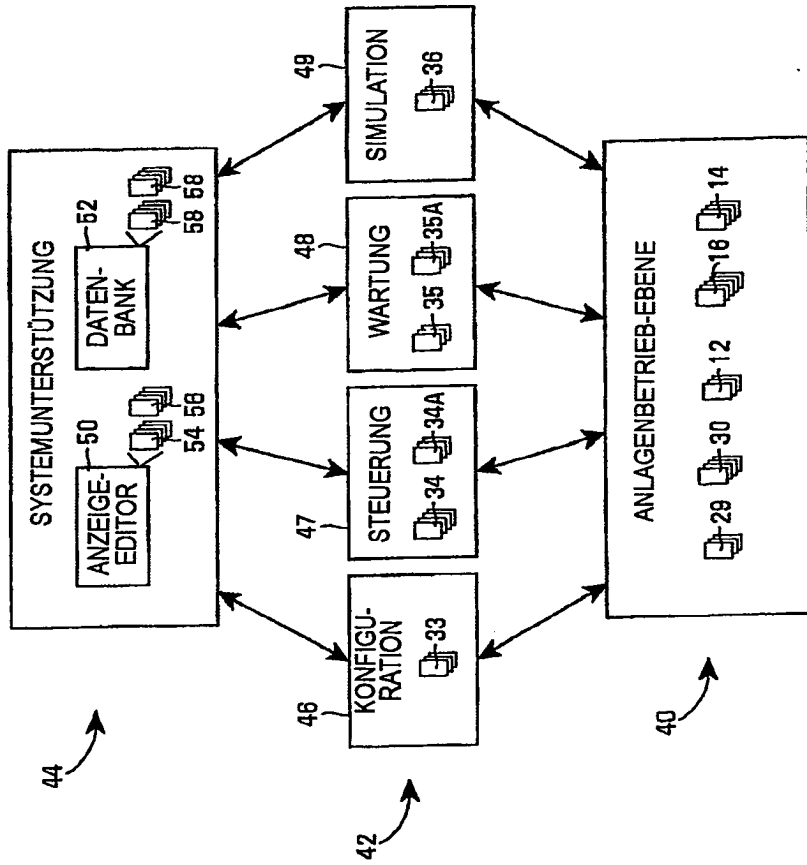


FIG. 1





**FIG. 2**

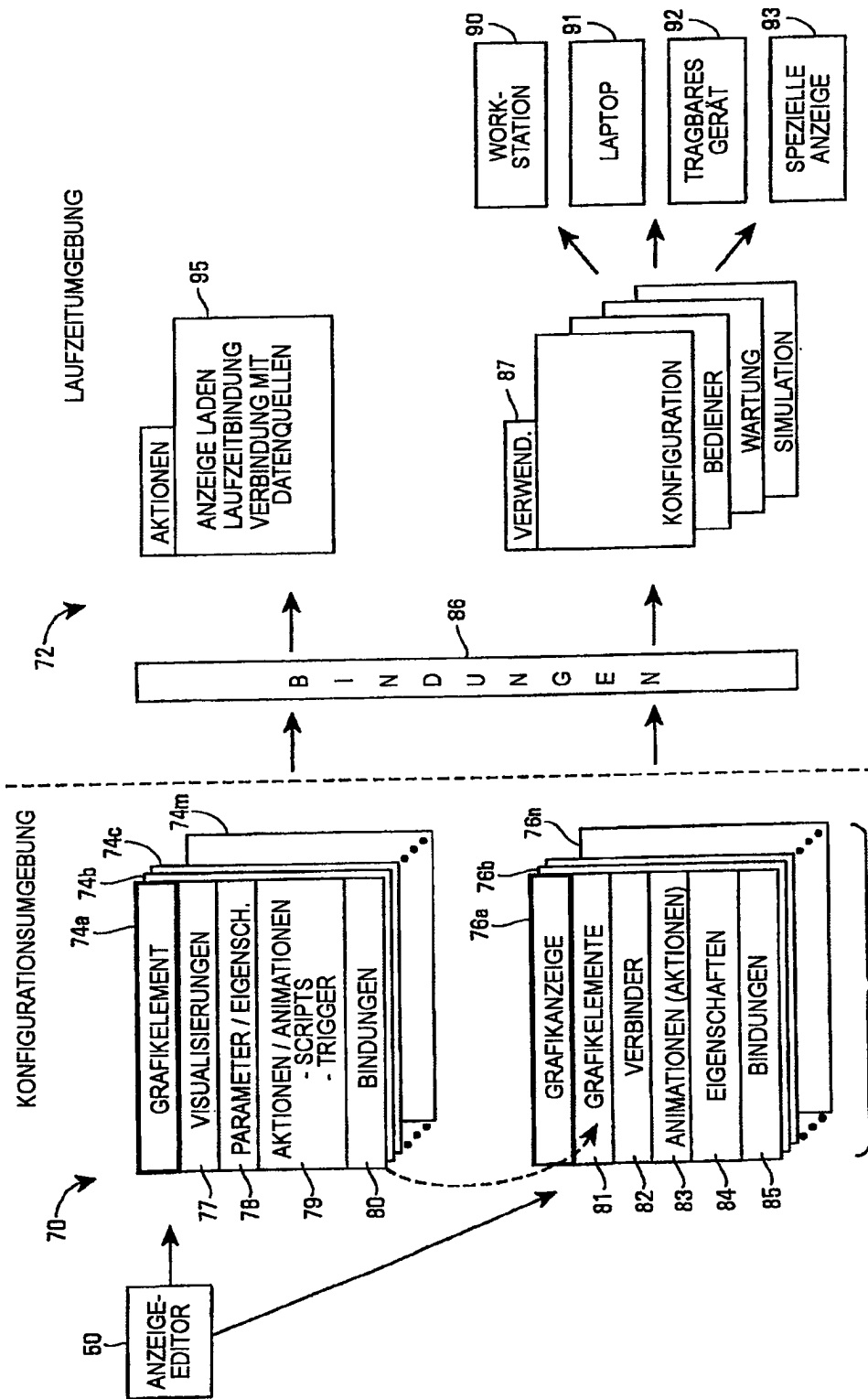
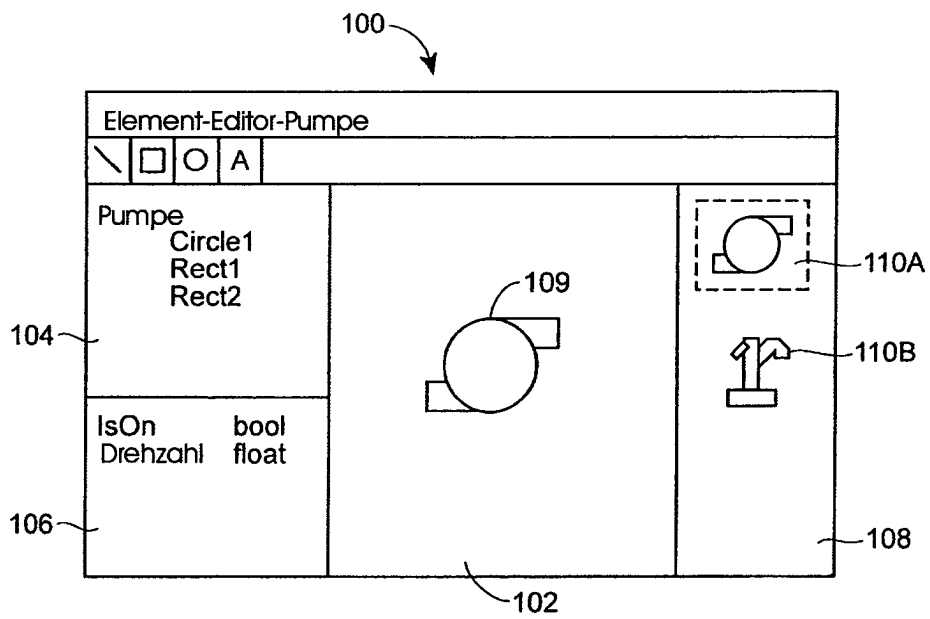


FIG. 3



**FIG. 4**

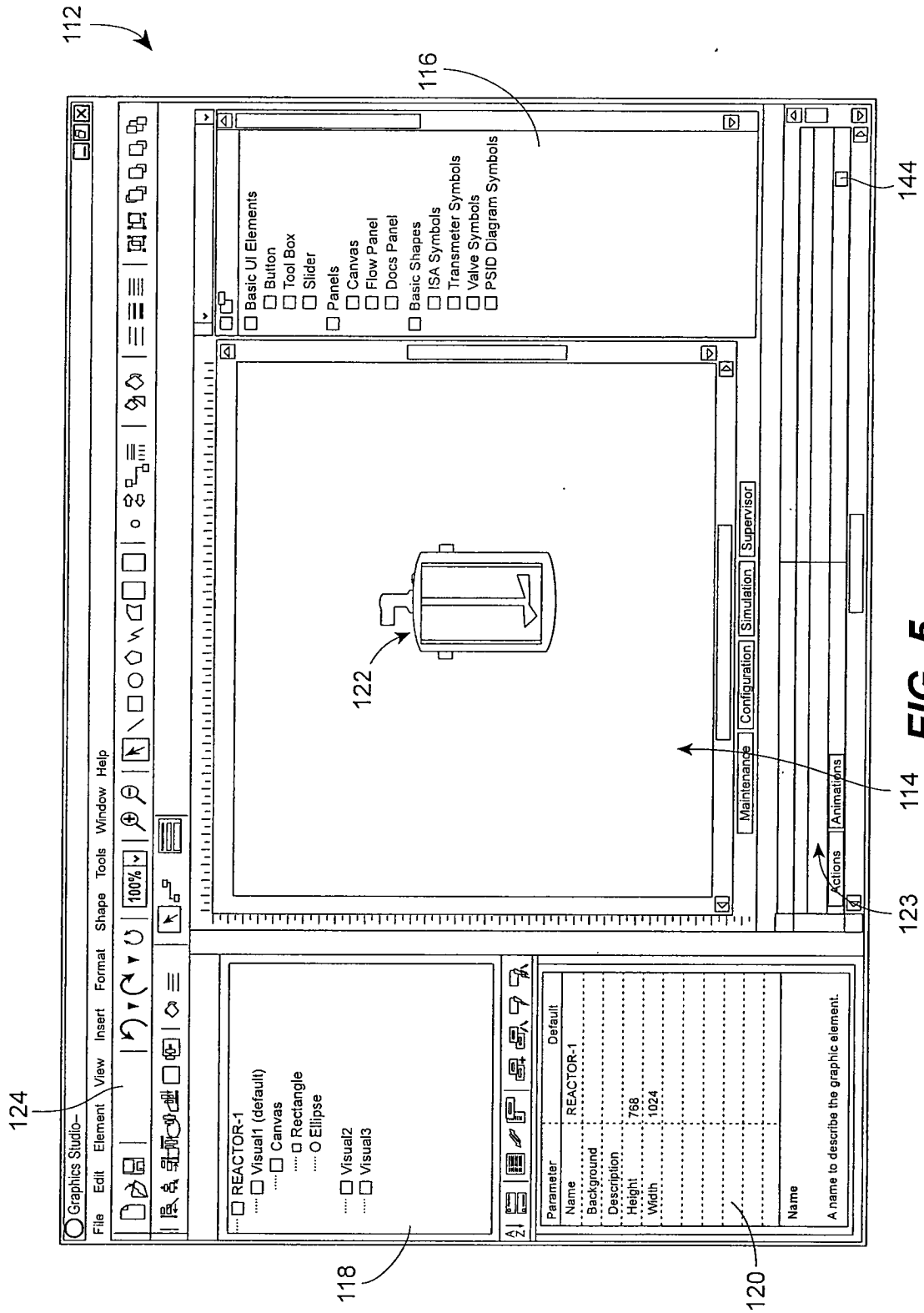
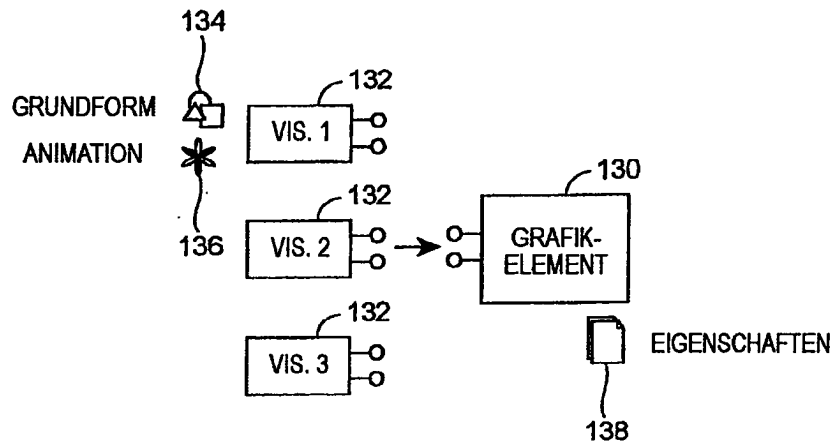


FIG. 5





**FIG. 6**

120A

Eigenschaften		
Name	Datentyp	Vorgabe
IsOn	Boole	True
Drehzahl	String	Schnell
<b>IsOn</b> Gibt an, ob die Pumpe angeschaltet ist		

FIG. 7

120B

Eigenschaften			
Name	Wert	Animation	Bindung
Name	1		
Hintergrund		1	<input type="checkbox"/> Spe... <input type="checkbox"/>
Beschreib.			
Höhe	768		
Breite	1024		
<b>IsOn</b> Gibt an, ob die Pumpe angeschaltet ist			

140

142

FIG. 8

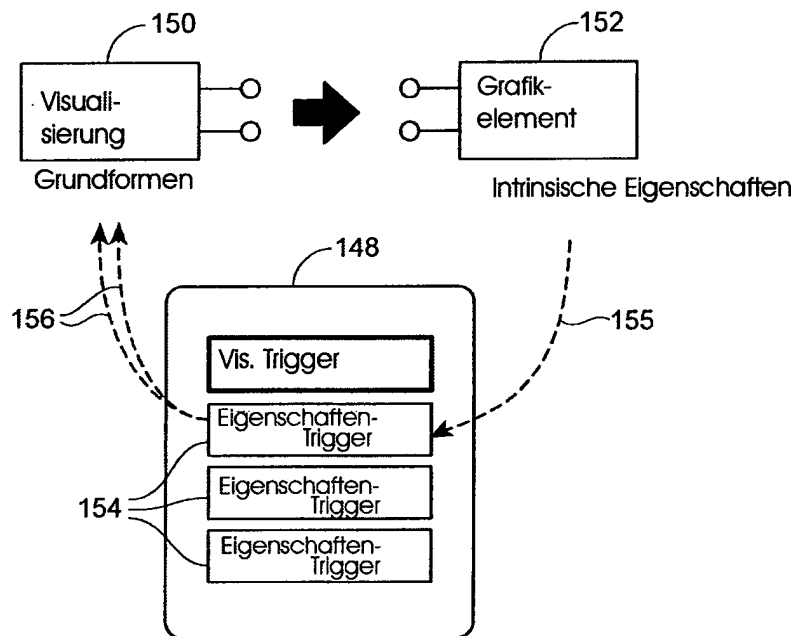
123A

Element Property	Value	Target	Path	Value
IsOn	True	Visual1/Rectangle1	Fill	Red
IsOn	True	Visual1/Ellipse1	Visible	False

144

Actions Animations Show All

**FIG. 9**



**FIG. 10**

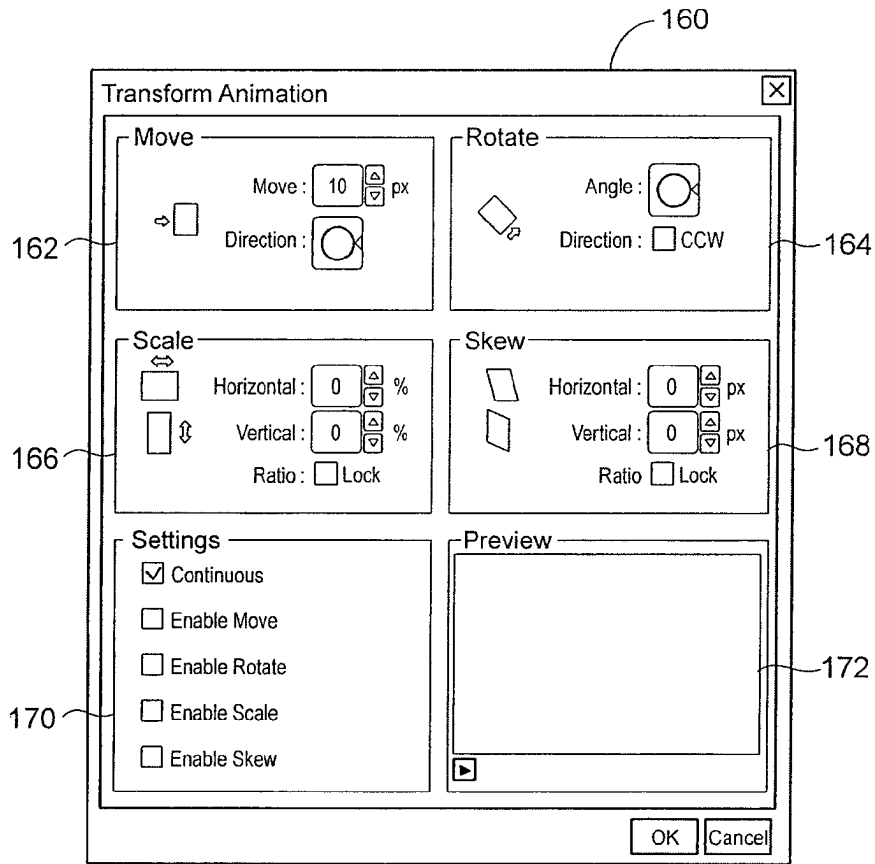


FIG. 11



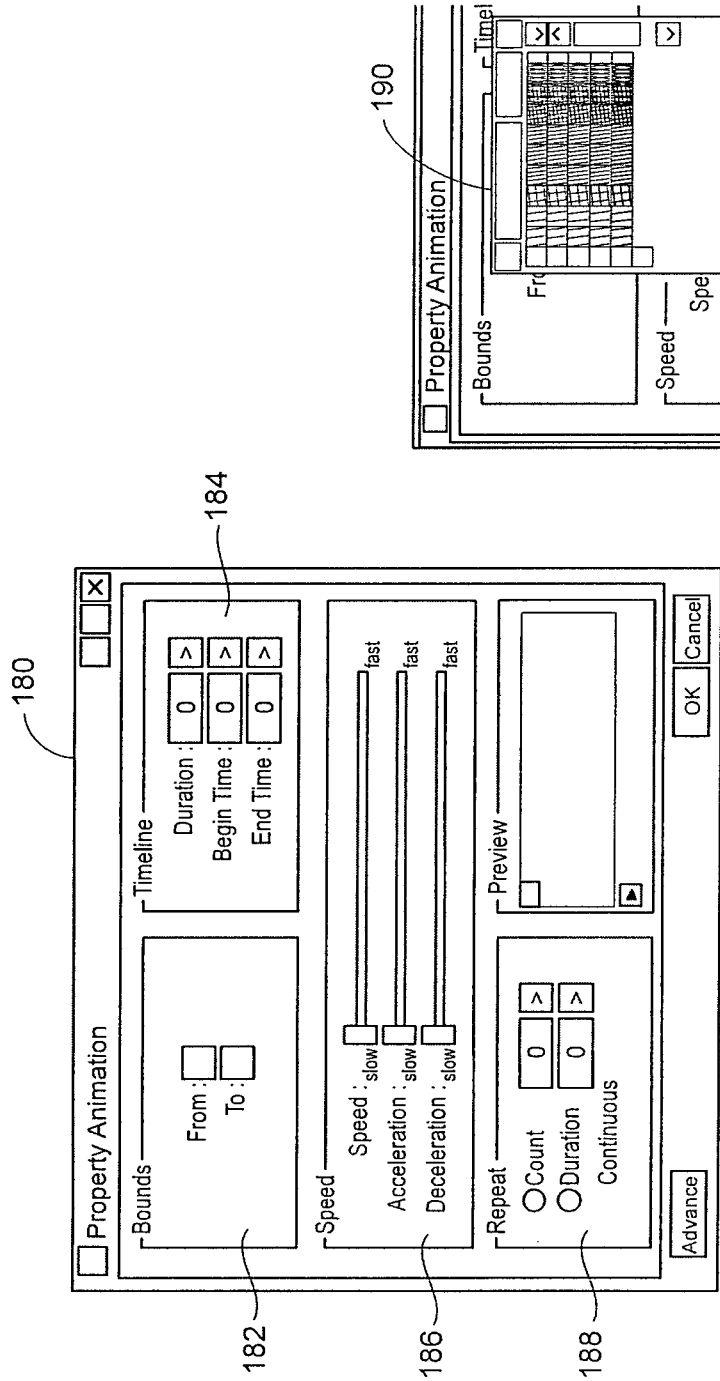
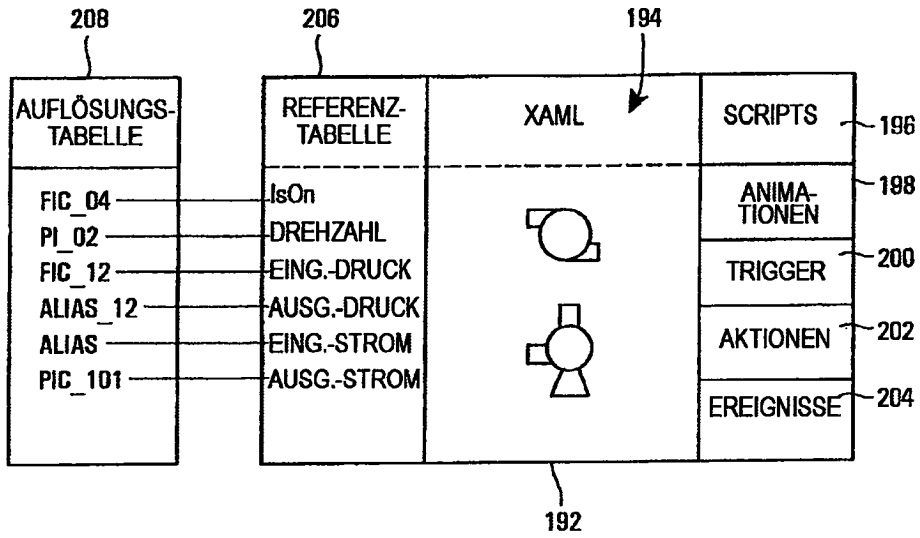
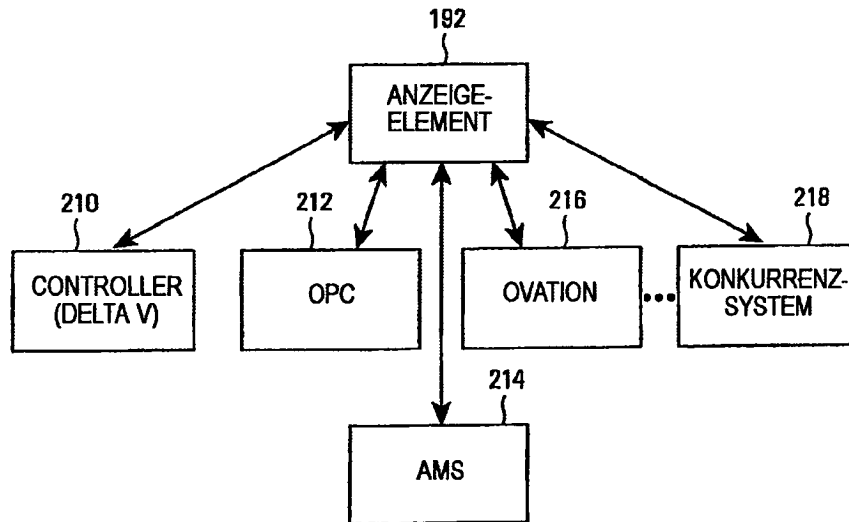


FIG. 12



**FIG. 13**



**FIG. 14**

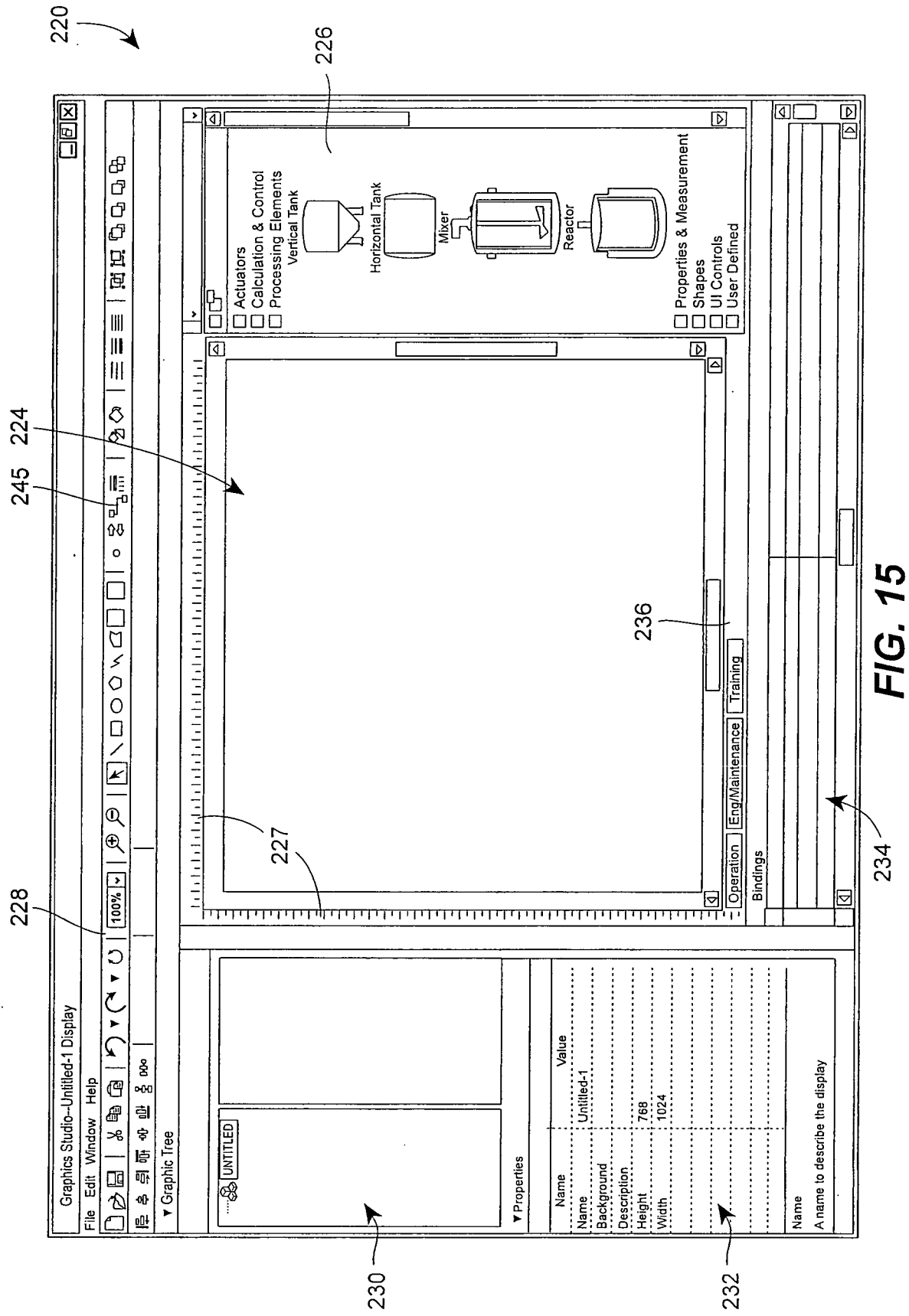


FIG. 15

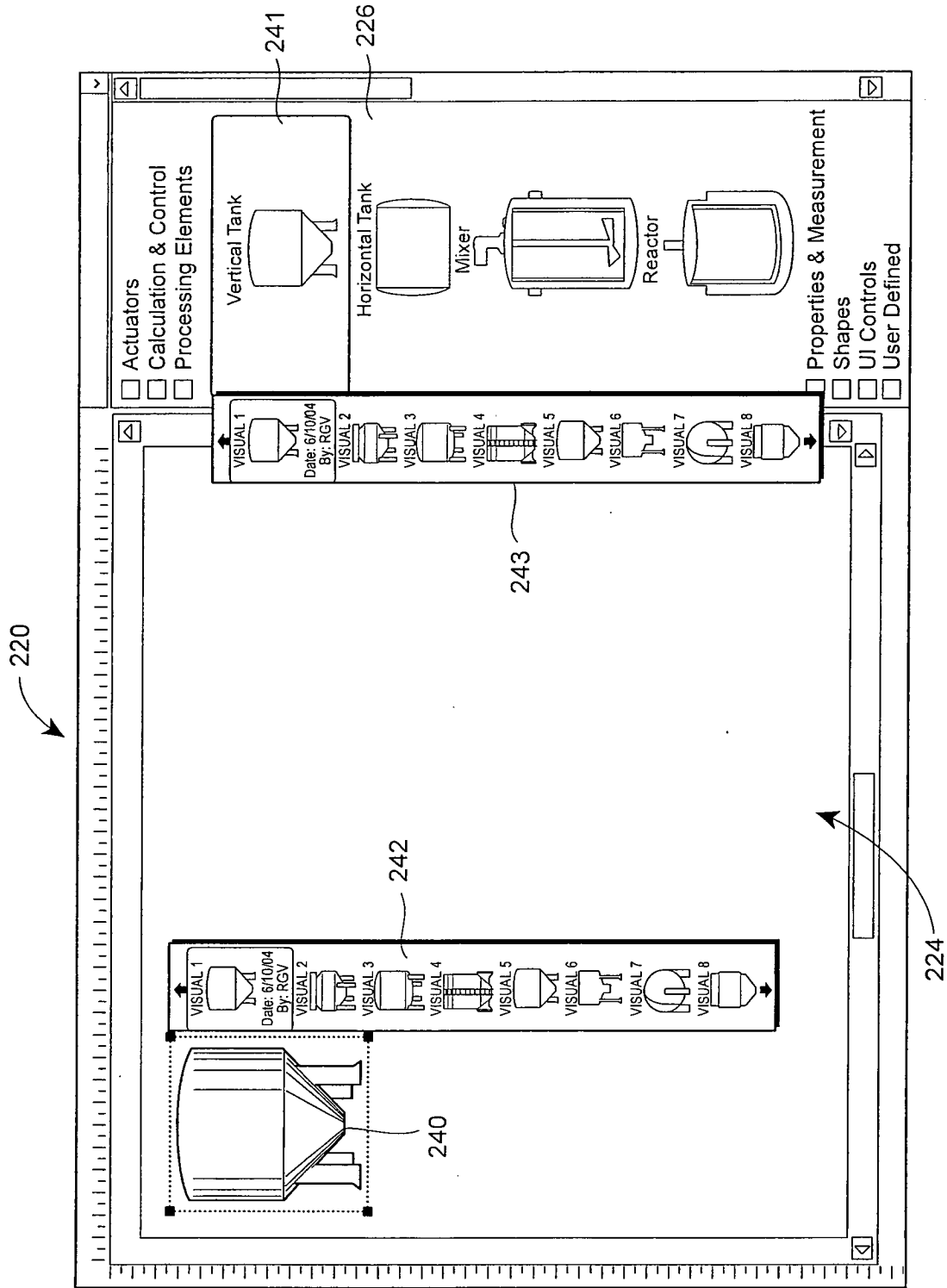
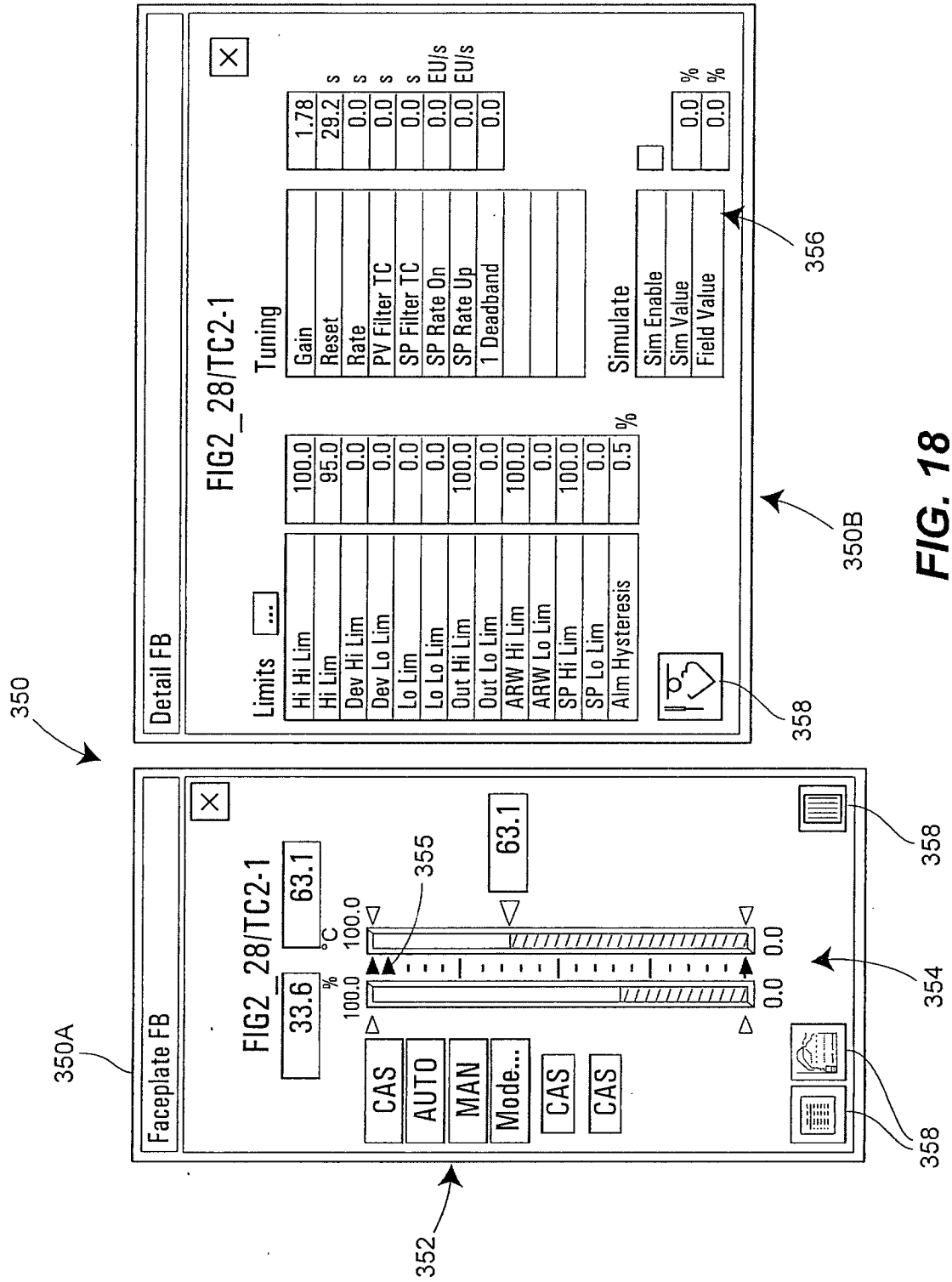


FIG. 16







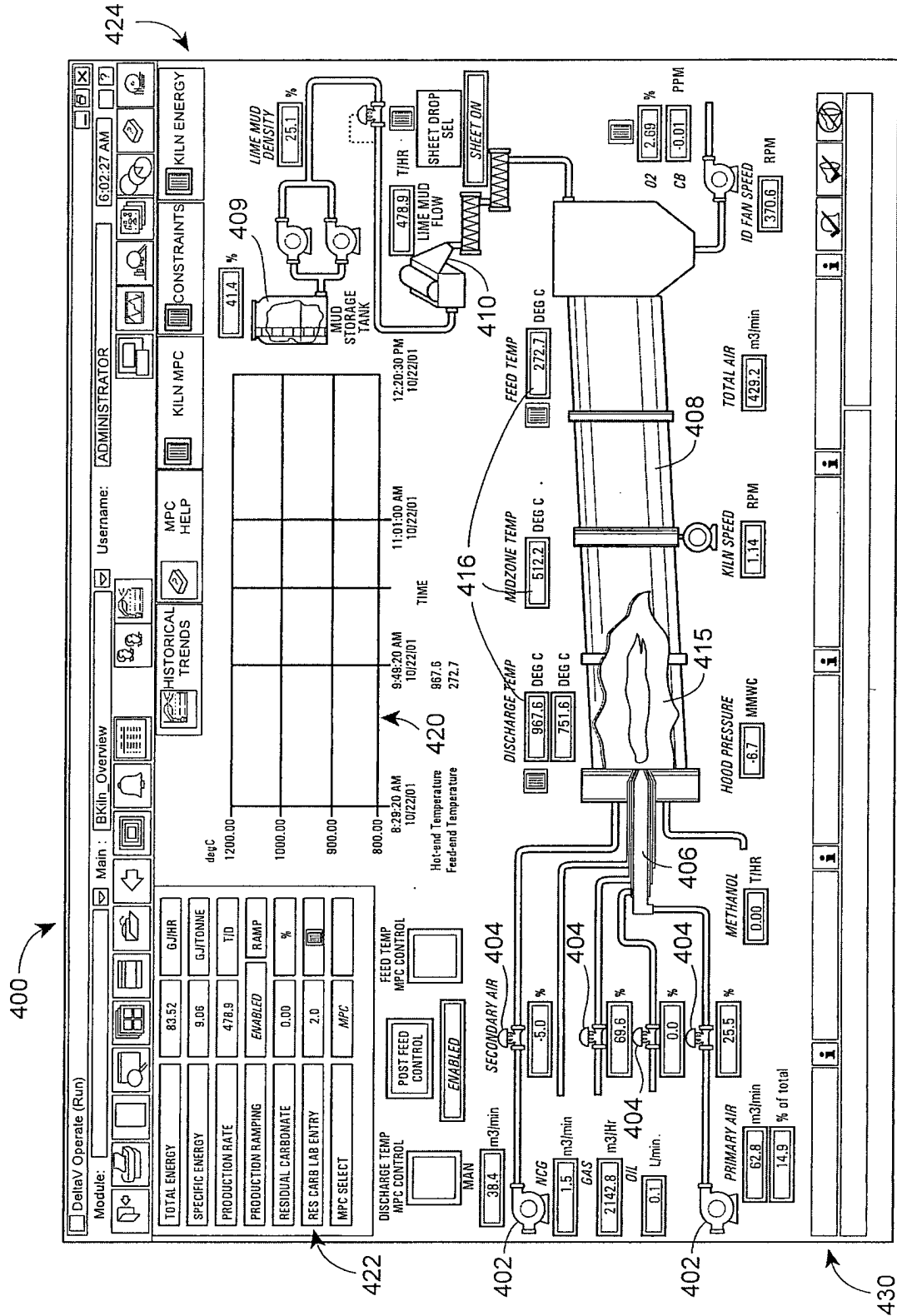


FIG. 19

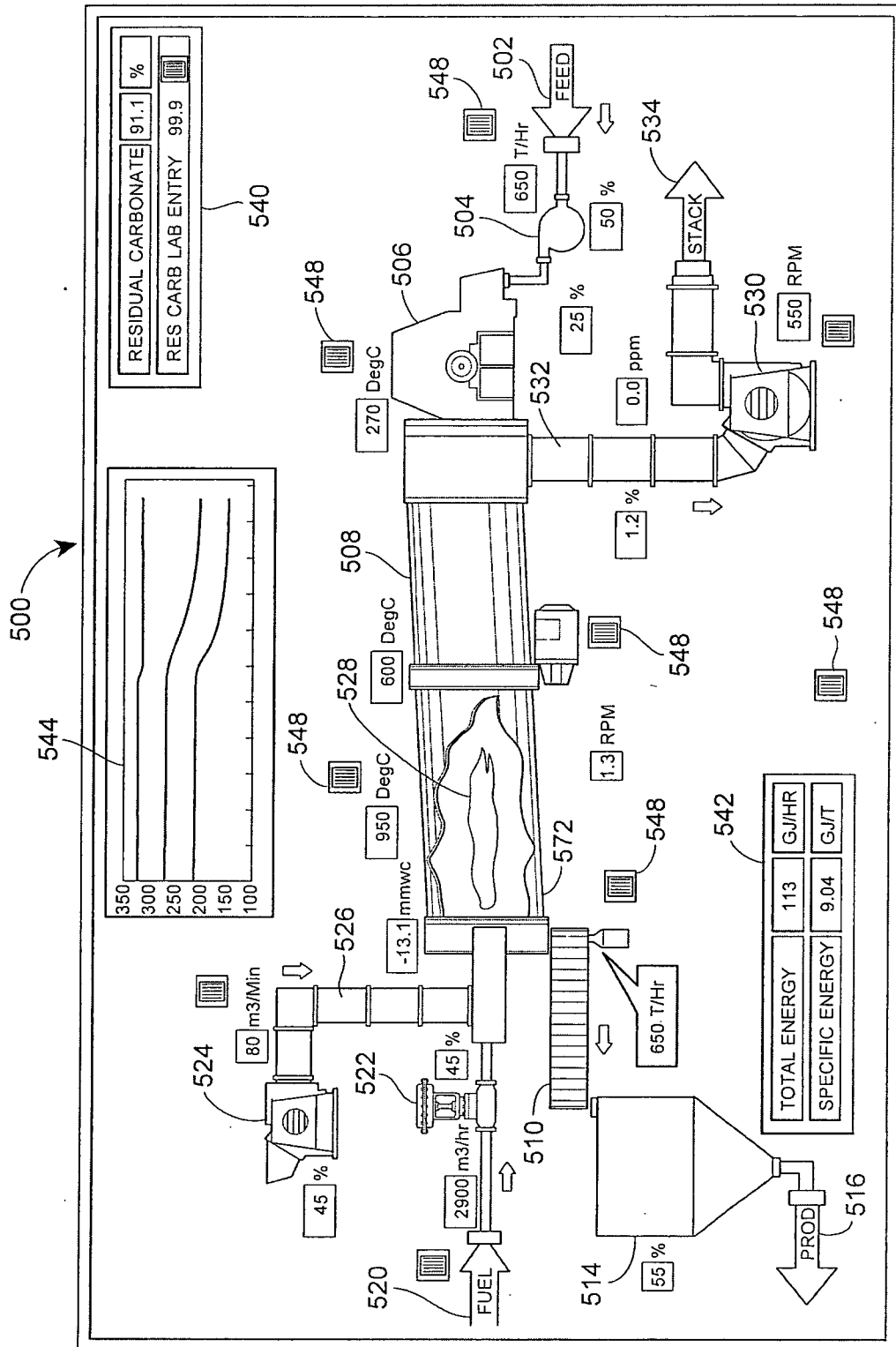


FIG. 20A

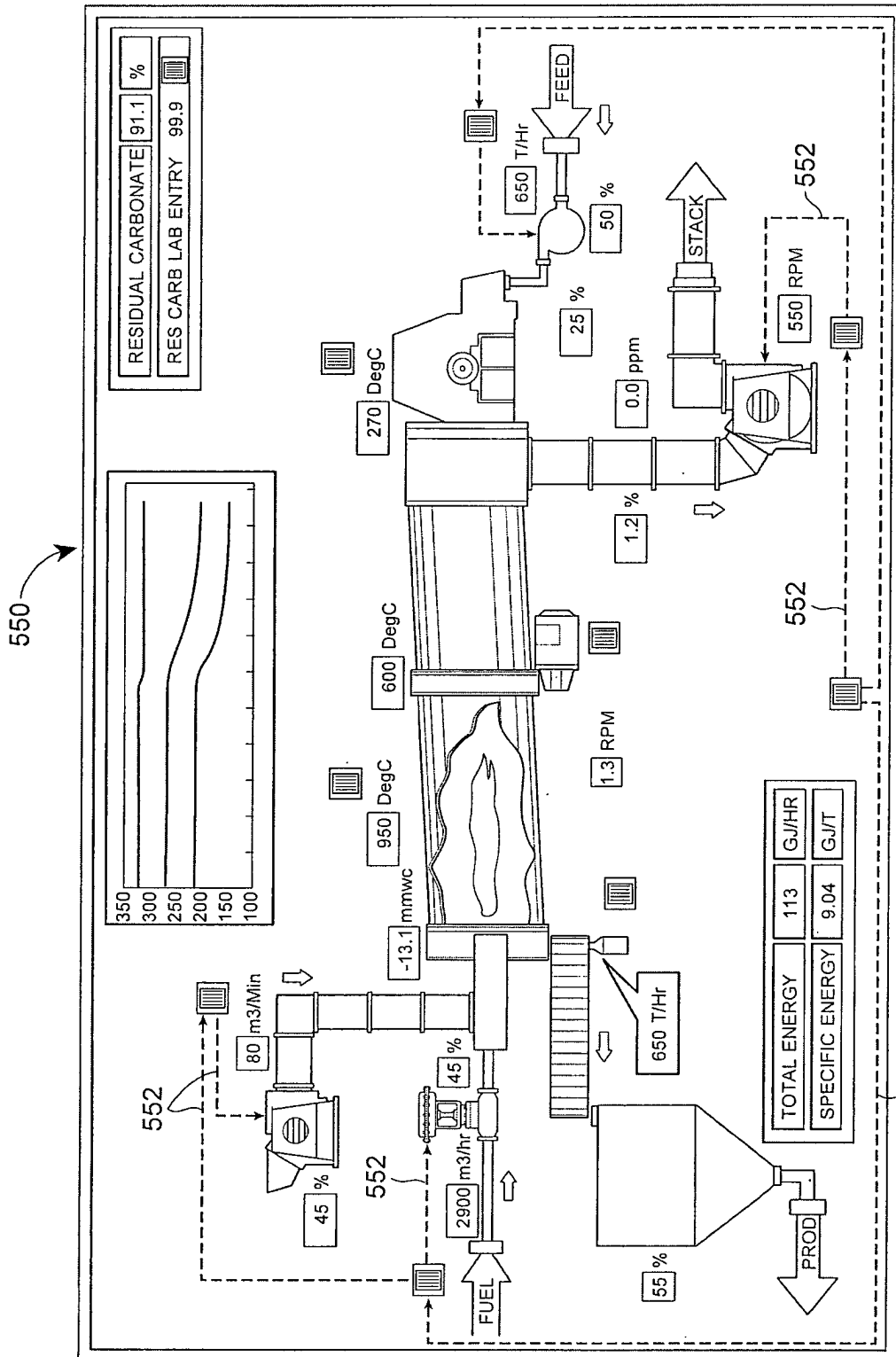


FIG. 20B





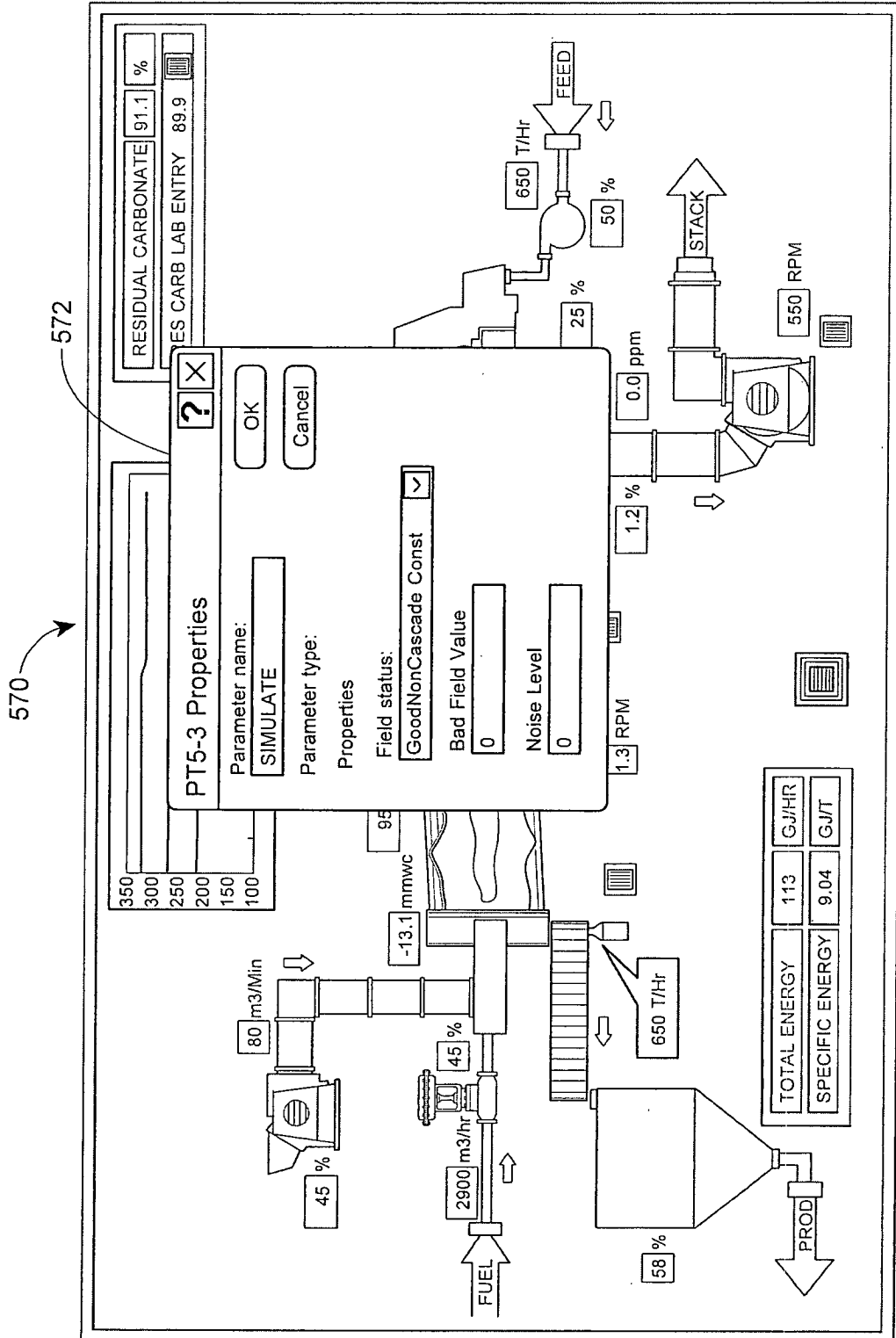


FIG. 20D

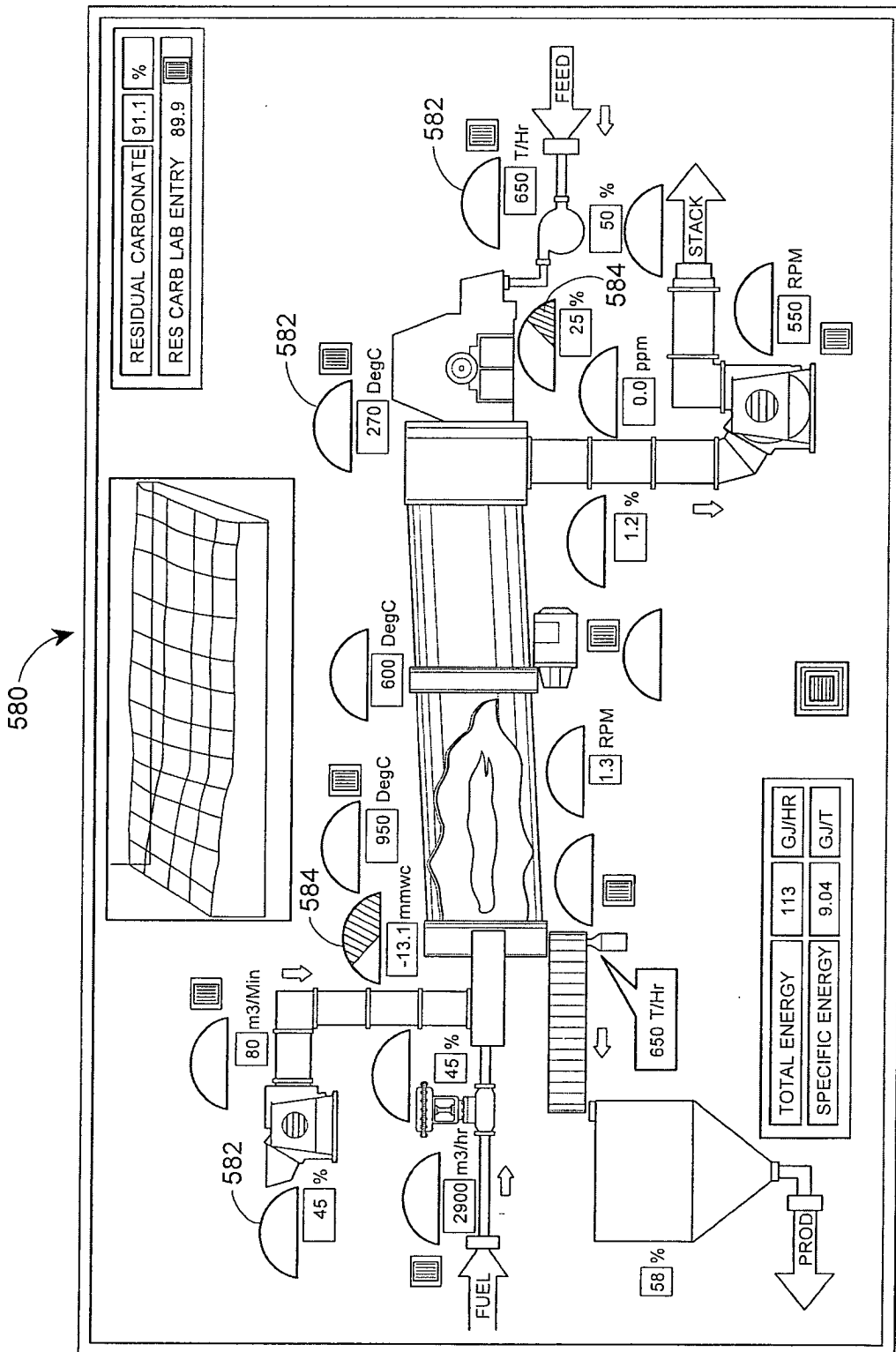
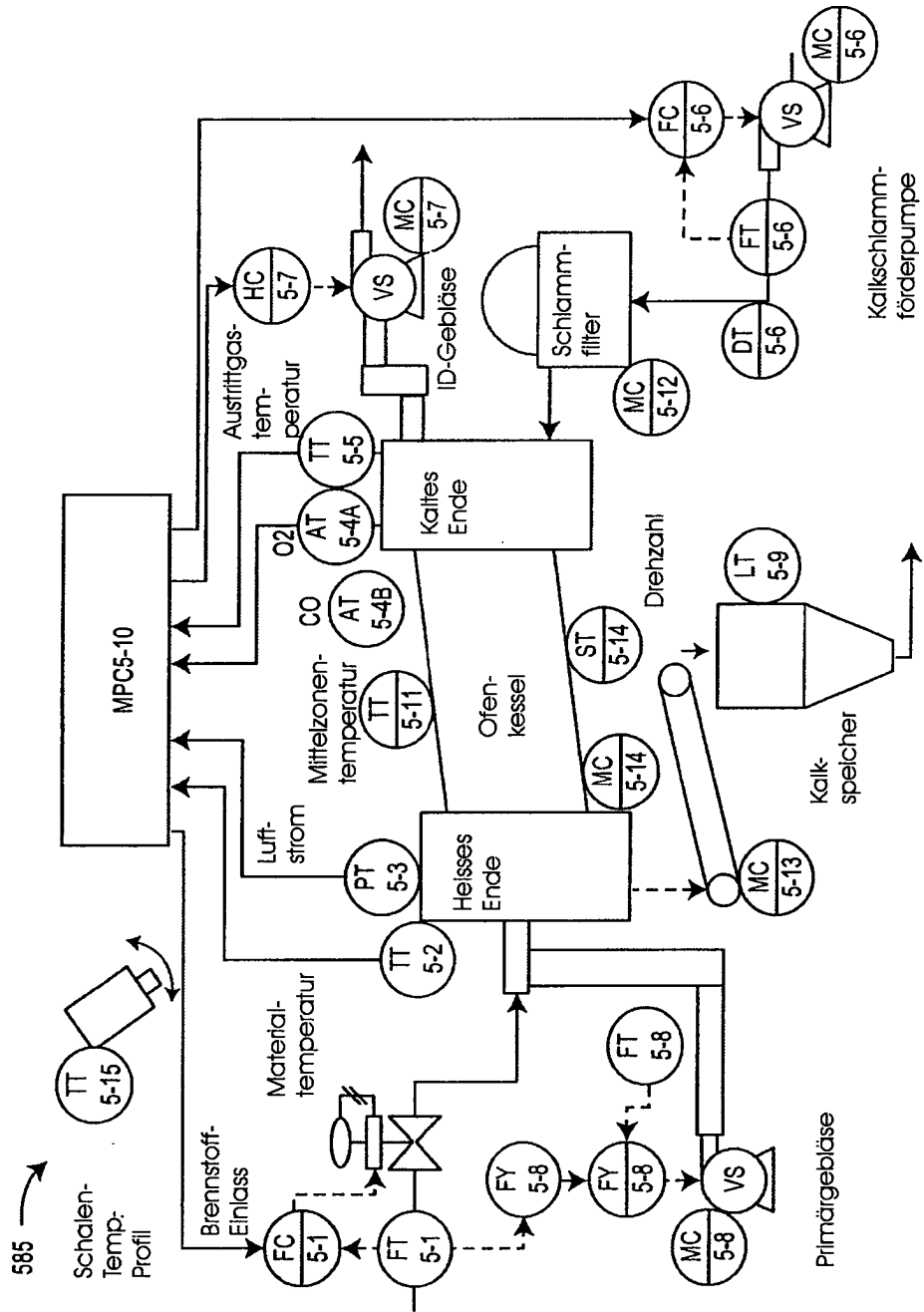
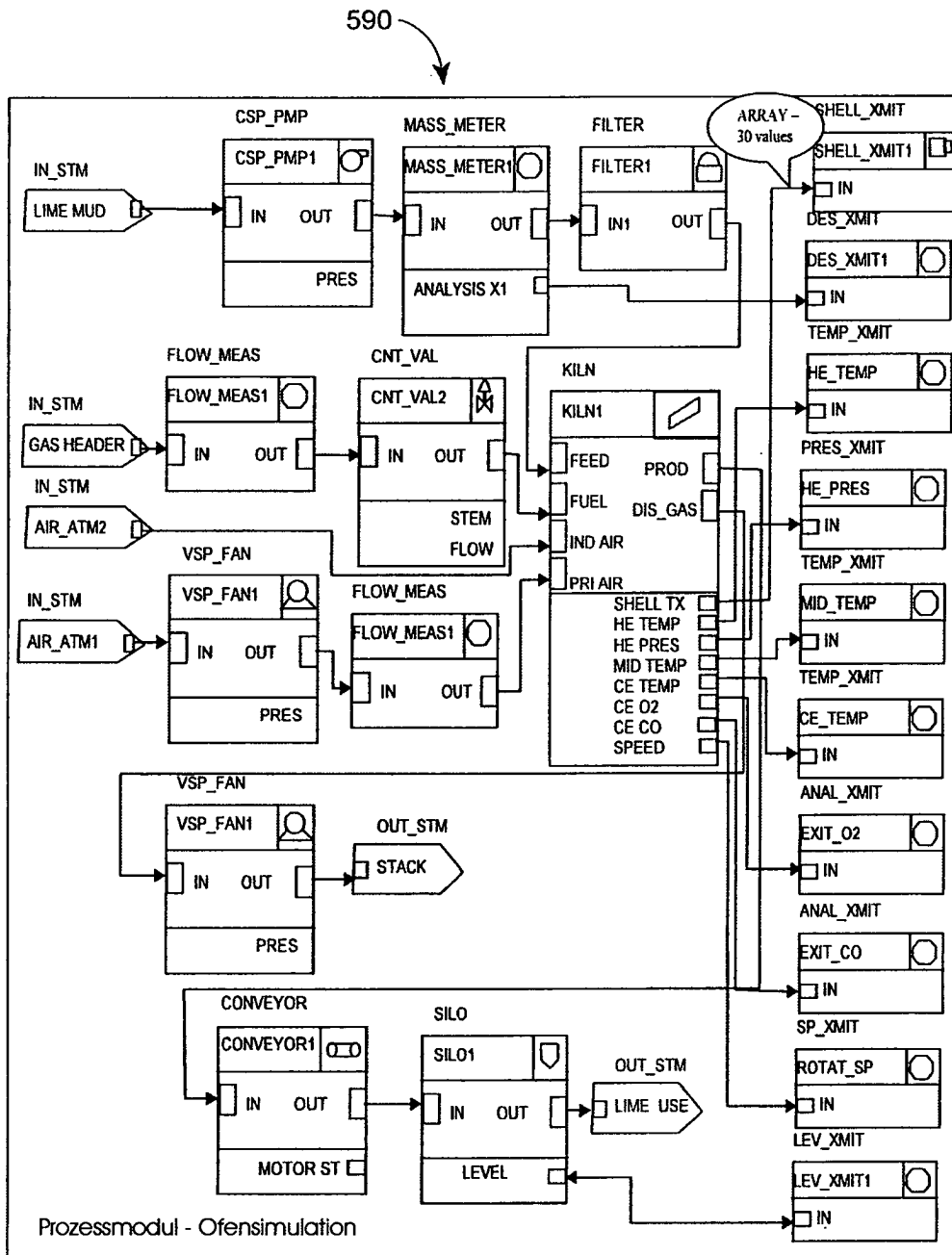


FIG. 20E





**FIG. 21B**

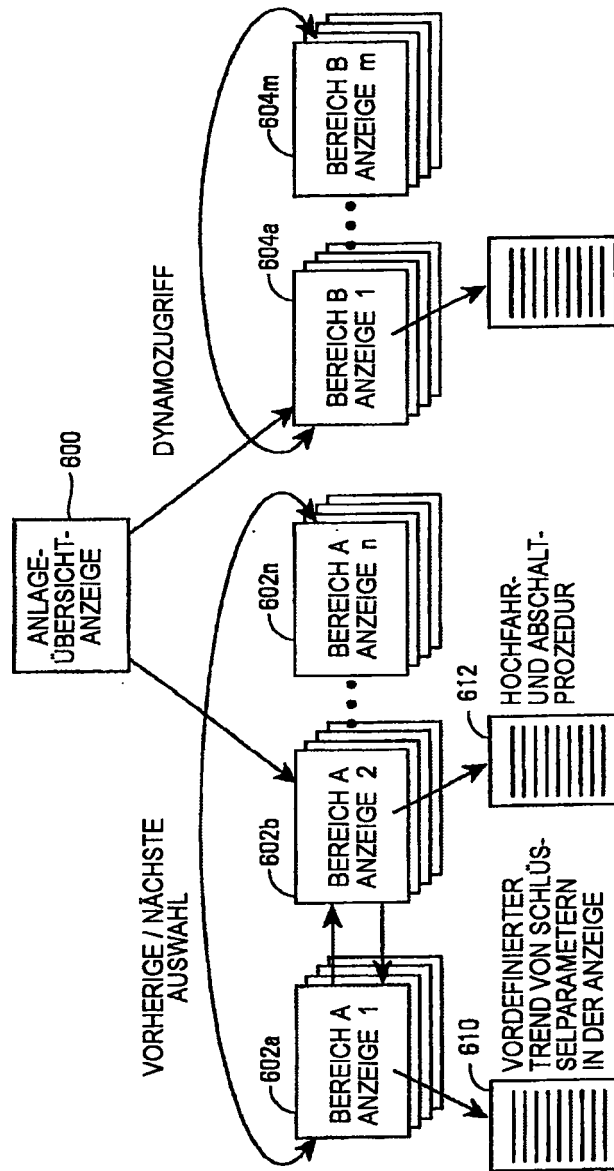


FIG. 22

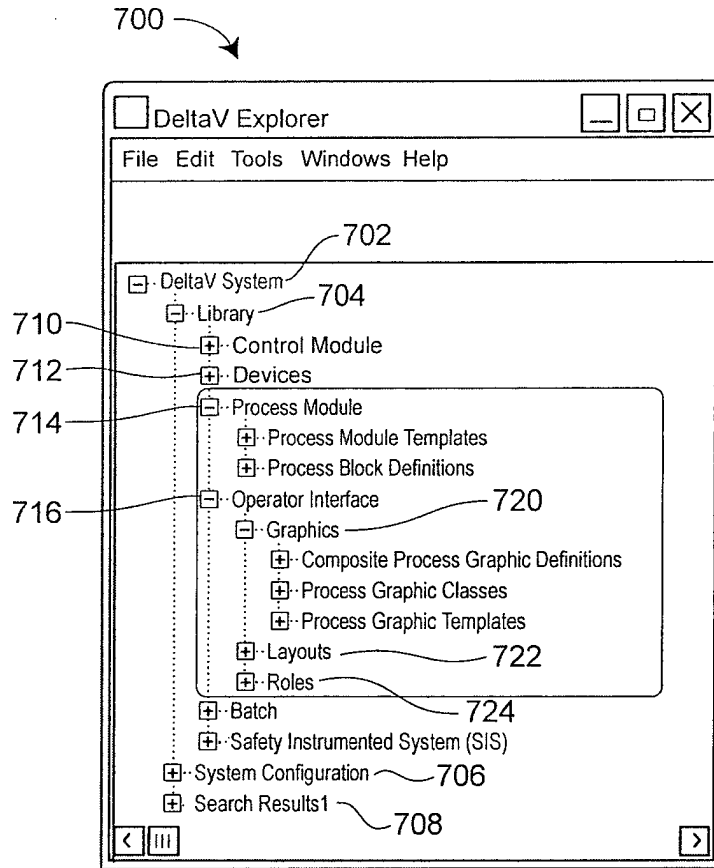


FIG. 23

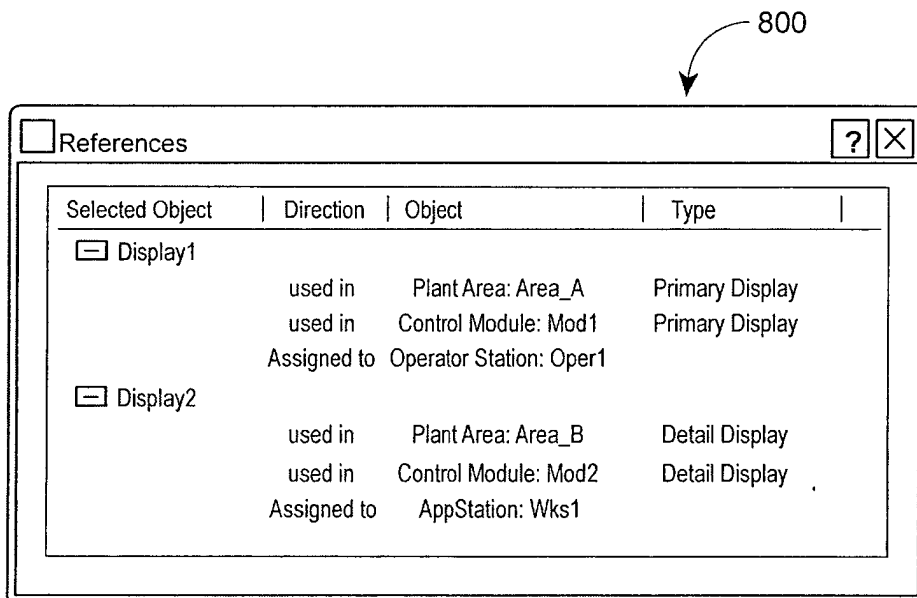


FIG. 25



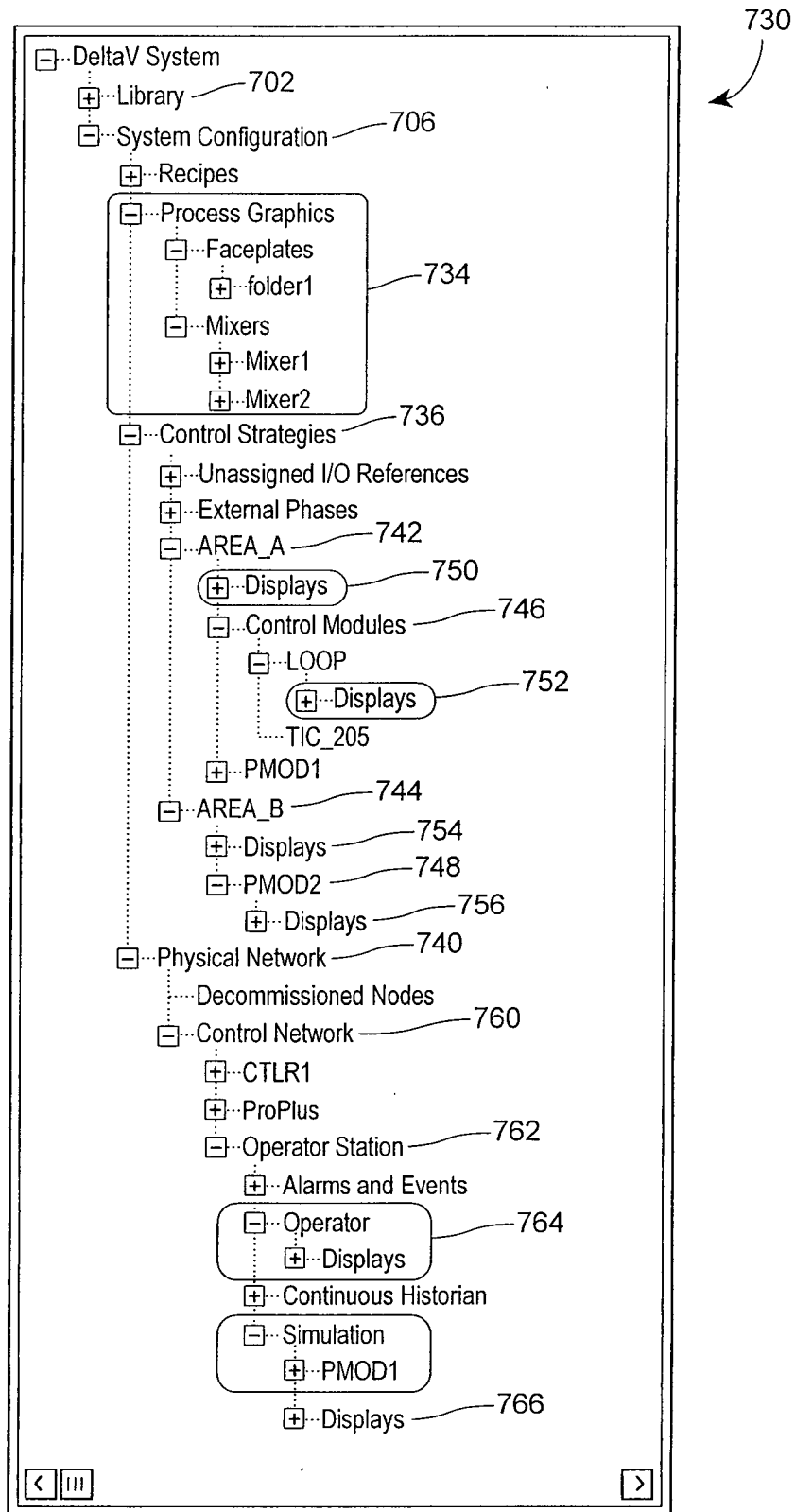


FIG. 24