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

(10) **DE 699 36 152 T2** 2008.01.24

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 192 535 B1**
(21) Deutsches Aktenzeichen: **699 36 152.4**
(86) PCT-Aktenzeichen: **PCT/US99/31135**
(96) Europäisches Aktenzeichen: **99 967 716.4**
(87) PCT-Veröffentlichungs-Nr.: **WO 2000/039674**
(86) PCT-Anmeldetag: **29.12.1999**
(87) Veröffentlichungstag der PCT-Anmeldung: **06.07.2000**
(97) Erstveröffentlichung durch das EPA: **03.04.2002**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **23.05.2007**
(47) Veröffentlichungstag im Patentblatt: **24.01.2008**

(51) Int Cl.[8]: ***G06F 9/44*** (2006.01)
H04L 12/24 (2006.01)

(30) Unionspriorität:
224487 **31.12.1998** **US**

(84) Benannte Vertragsstaaten:
AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE

(73) Patentinhaber:
Computer Associates Think, Inc., Islandia, N.Y., US

(74) Vertreter:
Rummler, F., Dipl.-Ing.Univ., Pat.-Anw., 80802 München

(72) Erfinder:
POHLMANN, William N., Newtown Square, PA 19073, US; MATSON, Kenneth D., Bellevue, WA 98008, US; CANTRELL, Paul, Sudbury, MA 01776, US

(54) Bezeichnung: **SYSTEM UND VERFAHREN ZUR DYNAMISCHEN KORRELATION VON EREIGNISSEN**

Beschreibung

GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft das Gebiet von Ereignisverwaltungssystemen und insbesondere die dynamische Korrelation von Ereignissen.

HINTERGRUNDINFORMATION

[0002] Die Informationstechnologie (IT) hat sich aus Nur-Großrechensystemen zu komplexen, stark verteilten Computersystemen entwickelt, die sich durch Netze über Arbeitsplätze und Abteilungen erstrecken. Diese verteilten Rechenumgebungen stellen Vorteile bereit, zu denen die Flexibilität bei der Auswahl jeder beliebigen Anzahl von Plattformen, Domänen, Werkzeugen und Netzkonfigurationen gehört.

[0003] US 5,787,409 beschreibt ein Überwachungssystem zum Überwachen eines verteilten Systems von einem Systemadministrator durch eine Benutzerschnittstelle. US 5,063,523 beschreibt ein Datenkommunikationsnetz-Verwaltungssystem, das einem Benutzer ermöglicht, Regeln zu erstellen, die ausgehend von Netzobjekten bezüglich eines Musters mit Attributen ankommender Ereignisse wie Alarmen übereinstimmen. WO/99/13427 beschreibt ein System und Verfahren zum Erkennen und Behandeln von Betrug.

[0004] Verteilte Rechenumgebungen können jedoch komplex sein. Ferner kann ein Mangel an Kompatibilität und Integration zwischen Softwarewerkzeugen und Plattformen bestehen. Zum Beispiel richten sich herkömmliche Point-Products (zum Beispiel das Produkt Platinum DBVision, das von Platinum technology, inc., Oakbrook Terrace, Illinois, hergestellt wird) im Allgemeinen an eine spezifische Funktion oder einen spezifischen Fachbereich wie eine Lösung zur Datenbanküberwachung und -verwaltung, Aufgabenplanung oder Verwaltung von Datenbankneuorganisationen. Jedes Point-Product stellt eine spezifische Fähigkeit bereit und jedes weist auch eine eigene Schnittstelle auf. Andererseits stellt die Benutzung einer Rahmentechnologie eine integrierte Lösung bereit, wenngleich die Werkzeugfunktionalität einen bedeutenden Verlust erleidet. Ferner sind zur Verwaltung der gegenwärtigen Unternehmensumgebung unter Verwendung entweder herkömmlicher Point-Products oder einer Rahmentechnologie viele Ressourcen und viel Geld von IT-Organisationen erforderlich.

[0005] Dementsprechend besteht ein Bedarf an einem integrierten System zum Bereitstellen von Werkzeugen, die eine kompatible Schnittstelle benutzen, ohne die Werkzeugfunktionalität bedeutend einzuschränken.

KURZDARSTELLUNG DER ERFINDUNG

[0006] Gemäß der vorliegenden Erfindung werden ein Ereignisverwaltungssystem und ein Verfahren zur Verwaltung eines Computersystems gemäß den Ansprüchen bereitgestellt.

[0007] Folglich wird ein verbessertes integriertes System zum Überwachen von Ereignissen bereitgestellt, die in Point-Products durch ein gemeinsames Ereignisverwaltungssystem eintreten.

[0008] Vorteilhaft hat die beschriebene und beanspruchte Anordnung die Fähigkeit zur dynamischen Korrelation von Ereignissen über Knoten eines Unternehmens.

[0009] Vorzugsweise stellt das System und Verfahren eine dynamische Ausführung einer geänderten Regel und/oder einer neu hinzugefügten Regel zur Ereigniskorrelation bereit.

[0010] In ihrer bevorzugten Ausführungsform stellt die vorliegende Erfindung ein Ereignisverwaltungssystem bereit, das einen Ereignisverwalter und einen Alarmregelspeicher und Ereigniskorrelator und eine Antwortmaschine aufweist. Der Ereignisverwalter empfängt und verteilt Ereignisinformation, die von einem Ereignisverwalter basierend auf Alarmregeln bereitgestellt wird, die in einem Alarmregelspeicher gespeichert sind. Die Antwortmaschine führt ein Antwortregelwerk basierend auf der Korrelation von Ereignissen von dem Ereigniskorrelator aus.

[0011] Ein bevorzugtes Merkmal stellt ein Verfahren zum dynamischen Ausführen einer modifizierten Regel oder einer neuen Regel zur Ereigniskorrelation bereit. Das bevorzugte Verfahren weist ferner das Vergleichen eines ersten Regelsatzes mit einem zweiten Regelsatz auf, um Modifikationen, Hinzufügungen oder Löschungen von Regeln zu identifizieren und alle entsprechenden Regeln, die modifiziert oder hinzugefügt worden

sind, dynamisch auszuführen.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0012] Fig. 1 zeigt ein Ausführungsbeispiel eines Unternehmens der vorliegenden Erfindung, das mehrere Knoten aufweist.

[0013] Fig. 2 zeigt ein Ausführungsbeispiel einer Folge integrierter Point-Products der vorliegenden Erfindung.

[0014] Fig. 3 zeigt ein Ausführungsbeispiel eines Blockdiagramms eines Ereignisverwaltungssystems der vorliegenden Erfindung.

[0015] Fig. 4 zeigt ein Ausführungsbeispiel von Ereignisinformationen, die zwischen mehreren Knoten des Ereignisverwaltungssystems der vorliegenden Erfindung übermittelt werden.

[0016] Fig. 5 zeigt ein Ausführungsbeispiel eines Flussdiagramms einer dynamischen Ausführung einer neu hinzugefügten oder geänderten Regel zur Ereigniskorrelation gemäß der vorliegenden Erfindung.

AUSFÜHRLICHE BESCHREIBUNG

[0017] Das Ereignisverwaltungssystem der vorliegenden Erfindung verwaltet ein Unternehmen (zum Beispiel ein Computernetz wie ein lokales Netz (LAN) oder ein Fernnetz (WAN)), korreliert Ereignisinformation, die in dem Unternehmen auftritt, und ergreift korrigierende Maßnahmen basierend auf vorbestimmten Antwortregelwerken. Das Ereignisverwaltungssystem empfängt zum Beispiel Ereignisnachrichten von kompatiblen Point-Products innerhalb des Unternehmens. Wie in Fig. 1 dargestellt, kann das Unternehmen **100** mehrere Knoten **110**, **120**, **130** aufweisen, die zum Beispiel durch ein Netz (nicht dargestellt) verbunden sein können. Ein Knoten ist zum Beispiel ein physikalisches Gehäuse wie ein Personalcomputer, Server usw., der ein Betriebssystem betreibt. In einer beispielhaften Form der vorliegenden Erfindung kann ein Knoten ein Personalcomputer mit einem darauf installierten kompatiblen Point-Product sein. In einem Ausführungsbeispiel der vorliegenden Erfindung verwaltet das Ereignisverwaltungssystem **140** Ereignisse auf den Knoten **110**, **120**, **130**, bei denen die Ereignisse erzeugt werden, wobei die Datenbewegung auf dem Netz minimiert wird und Aktionen wie Auswertung, Berichten und automatische Korrektur von Daten in der Nähe der Quelle gehalten werden.

[0018] In einer beispielhaften Form der vorliegenden Erfindung, wie in Fig. 2 dargestellt, kann das Ereignisverwaltungssystem **270** in einer Folge **200** integrierter Werkzeuge enthalten sein, einschließlich kompatibler Point-Products, die zum Beispiel zum Verwalten von Anwendungen, Datenbanken, Arbeitsplätzen, Netzen und Systemen benutzt werden. Die Werkzeuge **230**, **240**, **250**, **260** können einen Satz gemeinsamer Dienste **210** benutzen, um Daten gemeinsam zu benutzen, die mit Ereignissen in Beziehung stehen. Ein Direktor **220** ist zum Beispiel eine gemeinsam benutzte, grafische Benutzerschnittstelle (GUI), die in einem herkömmlichen Servercomputer für kompatible Point-Products **230**, **240**, **250**, **260** läuft. In einem Ausführungsbeispiel der vorliegenden Erfindung läuft der Direktor **220** unter Windows NT und Windows **95** und arbeitet als eine einzige Konsole, die es Benutzern ermöglicht, alle Ressourcen, einschließlich kompatibler Point-Products **230**, **240**, **250**, **260** in dem Unternehmen zu sehen und damit zu interagieren.

[0019] Die primäre verwaltbare Informationseinheit, die von dem Ereignisverwaltungssystem der vorliegenden Erfindung benutzt wird, ist eine strukturierte Nachricht, die ein Ereignis genannt wird. Ein Ereignis steht für Information über ein bestimmtes bedeutendes Vorkommnis von oder bezüglich einer Ressource in dem Unternehmen. Ereignisse werden von kompatiblen Point-Products und Ereignisverwaltern in dem Ereignisverwaltungssystem gesendet und empfangen. Ereignisse sind zum Beispiel die grundlegenden Bausteine der Echtzeit-Informationsverarbeitung, die zum Verwalten des Unternehmens benutzt wird. Ereignisse definieren das fundamentale Element von veröffentlichbarer, teilbarer Information und das Format, in dem die Ereignisse verwaltet und übertragen werden sollen. Zur Struktur eines Ereignisses gehören zum Beispiel vordefinierte Felder für die grundlegende Information, die für jede Ereignisaufzeichnung notwendig sind. Die vorbestimmten Felder fallen in zwei Kategorien: identifizierende Felder und nicht-identifizierende Felder. Identifizierende Felder sind Felder, die zusammen einen eindeutigen Schlüssel für das Ereignis bilden, der es von einem anderen Ereignis unterscheidet. Nicht-identifizierende Felder sind Felder, die zusätzliche Information über ein Ereignis hinzufügen, jedoch nicht an dem Schlüssel des Ereignisses teilnehmen. Das Ereignisverwaltungssystem der vorliegenden Erfindung ermöglicht, dass die Ereignisstruktur von einem Benutzer durch dynamisches Hinzufügen von Schlüsselwertpaaren und somit eindeutiges Definieren der jeweiligen Ereignisstruktur erweitert wird. Dem-

entsprechend befähigt die Hinzufügung von Schlüsselwertpaaren zu einer Ereignisstruktur ein Point-Product dazu, das eindeutig definierte Ereignis zu veröffentlichen, das anderenfalls nicht veröffentlicht worden wäre, da es von dem vorbestimmten Feldern des Ereignisses nicht eindeutig definiert worden wäre.

[0020] Ein Ausführungsbeispiel einer Ereignisstruktur gemäß einer Ausführungsform der vorliegenden Erfindung wird nachstehend zum Beispiel in der C-Sprache definiert.

```
typedef struct_PT_EVENT
{
int             Version;  /* Event structure version    */
PT_CHAR_T       *Node;    /* Node of event              */
PT_CHAR_T       *Class;   /* Event class                */
PT_CHAR_T       *Instance;     /*     Originating     product
instance  */
PT_CHAR_T       *Name;    /* Event name                 */
PT_CHAR_T       *Time;    /* Event time (yyyymmdd hh24miss)
     */
PT_CHAR_T       *CondTime;     /* Condition time (yyyymmdd
hh24miss) */
PT_CHAR_T       *AgentNode;    /* Node where detecting agent
is running     */
PT_CHAR_T       *EvmgrNode;    /*  Node  where  responsible
event mgr runs */
PT_EVENT_TYPE   Type;     /*  Event  type  (EV_DISCRETE...)
     */
PT_RESP_TYPE    Response; /* Response type (EVRE_SILENCE...)
     */
PT_CHAR_T       *RespPolicy;   /* Response Policy  */
PTJOHARJT *Descr;   /* Description */
PT_CHAR_T       *DescrId;      /*  Message  string  key  for
MsgPut    */
PT_CHAR_T       *DescrFields; /* Field list for substitution
in message     */
```

```
double    Value;    /* Value (for condition/alarm events)
     */
int  Level;    /* Alarm level (for type= EV_ALARM_SET) */
PT_CHAR_T      *IntKeys; /* Application specific internal
keys */
int  NumKey;   /* Number of key/value pairs following */
PT_CHAR_T **Keys;   /* Array of other key (attribute) names
     */
PT_CHAR_T **Values; /* Array of other key (attribute)
values    */
BOOL Archived; /* Boolean flag, TRUE if event archived
     */
PT_CHAR_T *Id; /* Unique event id  */
PT_CHAR_T *CondId;  /* Condition id     */
int  Repeat Count;  /* Count of duplicate events of this
type used for storm suppression    */
int  HopCount; /* Here comes Peter Cotton Tail... */
PT_CHAR_T *GMTOffset;    /* GMT Offset  */
int  ActionTaken;  /* Boolean flag, Response action in
progress  */
BOOL Silenced; /* Boolean flag, Alarm silenced    */
PT_CHAR_T *ProductName;  /* Product name of submitting
product   */
PT_CHAR_T *InstanceType; /* Type of instance in Instance
     */
void with event     *localParam;  /* Hook to allow local
associated data     */
PT_CHAR_T *AuthString;   /* Placeholder for authorization
string, some form of event content signature */
PT_CHAR_T *TTId;    /* Trouble ticket id     */
PT_CHAR_T *TTStatus;     /* Trouble ticket status */
}_PT-EVENT;
```

[0021] Die identifizierenden Felder des beispielhaften Ereignisses sind Knoten, Name, Produkt, Instanz, Typ, Bedingungszeit, wenn der Typ nicht diskret ist, Ereigniszeit, wenn der Typ diskret ist, wobei alle Schlüsselwertpaare die Inhalte der Schlüsselfeldanordnung und Wertefeldanordnung (zum Beispiel mit dem Feld NumKeys, einschließlich einer Zahl, die die Anzahl von Schlüsselwertpaaren in ihren jeweiligen Feldanordnungen angibt) aufweisen. Alle anderen vordefinierten Felder sind nicht-identifizierende Felder.

[0022] Die Ereignisse können in mehrere Typen kategorisiert werden, einschließlich zum Beispiel diskrete Ereignisse, Bedingungen und Alarme. Diskrete Ereignisses sind Ereignisse, die anzeigen, dass etwas zu einem bestimmten Zeitpunkt eingetreten ist, und sind vollkommen unabhängig. Der Eintritt für ein diskretes Ereignis hat keinen Zustand und wird nicht aktualisiert. Ein fehlgeschlagener Anmeldeversuch kann zum Beispiel die Erzeugung eines diskreten Ereignisses verursachen. Bedingungen sind Ereignisse, die den Zustand von et-

was anzeigen, das über einen Zeitraum andauert und Attribute aufweisen kann, die aktualisiert werden. Die Ereignisse, die von einem Produkt angegeben werden, sind Eigentum dieses Produkts. Im Allgemeinen kann nur das jeweilige Point-Product die Bedingungsereignisse, die bei dem Point-Product erzeugt werden, aktualisieren oder löschen. Die Inhalte von diskreten Ereignissen und Bedingungsereignissen repräsentieren echte Information über das Unternehmen, die nicht ohne eine Änderung des Vorkommnisses, das das Ereignis verursacht, geändert werden kann. Ein Alarm ist zum Beispiel eine Interpretation anderer Ereignisse, die auf einem Regelwerk basieren, das von einem Benutzer konfiguriert werden kann. Demgemäss könnte der Benutzer einen Alarm jederzeit löschen, ohne die Bedingung aufzuheben, die ihn verursacht hat. In ähnlicher Weise kann der Alarm über das Löschen eines Ereignisses, auf dem der Alarm basiert, hinaus andauern.

[0023] Wie in Fig. 3 dargestellt, weist ein Ausführungsbeispiel eines Ereignisverwaltungssystems **300** der vorliegenden Erfindung einen Ereignisverwalter **310**, ein Ereignisarchiv **320**, einen Ereigniskorrelator **330**, einen Alarmregelspeicher **340** und eine Antwortmaschine **350** auf. In einer beispielhaften Form der vorliegenden Erfindung sind ein Ereignisverwalter **310**, ein Ereignisarchiv **320**, ein Ereigniskorrelator **330** und eine Antwortmaschine **350** auf allen Knoten des Unternehmens enthalten und der Alarmregelspeicher **340** ist auf einem zentralen Knoten enthalten, der ermöglicht, dass Ereignisse lokal gespeichert und verwaltet werden.

[0024] In einer beispielhaften Form der vorliegenden Erfindung kann ein Ereignisverwaltungssystem zum Beispiel Ereignisnachrichten von Point-Products empfangen, zum Beispiel in einem Unternehmen. Die Ereignisse werden auf einem Knoten des Unternehmens verwaltet, bei dem die Ereignisse von einem Ereignisverwalter **310** empfangen werden, der sich auf dem entsprechenden Knoten befindet. Der Ereignisverwalter **310** kann zum Beispiel alle Ereignisse empfangen, die Zustände von vorher gesendeten Ereignissen verwalten, eine Teilnahmeliste führen und Ereignisse an die geeigneten Teilnehmer leiten. In einer beispielhaften Form der vorliegenden Erfindung können die Ereignisse und ihr Zustand und die Teilnahmeliste lokal gespeichert werden.

[0025] Wie in Fig. 4 dargestellt, empfangen der Ereignisverwalter **402** von Knoten a **401** und der Ereignisverwalter **411** von Knoten b **410** auch Ereignisinformation von dem Ereigniskorrelator **413** von Knoten b **410**. Der Ereignisverwalter **411** von Knoten b **410** stellt auch dem Ereigniskorrelator **413** auf Knoten b Ereignisse bereit. Der Ereignisverwalter **411** empfängt auch Ereignisinformation von dem Point-Product **415**, bei dem Ereignisse tatsächlich eintreten. Der Ereignisverwalter **402**, **411** verwaltet zum Beispiel die Ereignisse und ihren zugehörigen Zustand und eine Liste von Teilnahmen. Jeder Ereignisverwalter kann einen Datenspeicher im lokalen Speicher, zum Beispiel eine Wandtafel aufweisen, auf der zustandsorientierte Ereignisse gespeichert sind. Die Wandtafel kann zur Wiederherstellung der Information über Erzeugungen (Prozessaufruf des Ereignisverwalters) dauerhaft in einem dateigestützten Speicher aufbewahrt werden. Die Clients, die an Ereignissen teilnehmen, sind für die erneute Erstellung der jeweiligen Teilnahmen über neue Aufrufe des Ereignisverwalters verantwortlich. Dementsprechend können die Teilnahmen im Speicher verwaltet werden. Das lokale Ereignisarchiv wird für alle Ereignisse verwaltet, die von dem Ereignisverwalter empfangen werden. Das Ereignisverwaltungssystem der vorliegenden Erfindung kann auch Ereignisse von einer Vielzahl von Knoten korrelieren. In einer beispielhaften Form der vorliegenden Erfindung stellt das Ereignisverwaltungssystem Ansichten von Ereignissen bereit, die mit einzelnen Verwaltungsstationen oder in Ansichten/Kategorien konsolidiert sind, die Knotengrenzen überschreiten.

[0026] Das Ereignisverwaltungssystem der vorliegenden Erfindung ist strukturiert, um durch zum Beispiel eine Ereignisteilnahme ein herausragendes Interesse an Gruppen von Ereignissen durch Kriterien abzufragen oder auszudrücken, die kein Knoten sind. Die Ereignisteilnahme ermöglicht die Spezifizierung von Kriterien bezüglich der Inhalte der Felder des Ereignisses. Die Bestimmung der betreffenden Ereignisse auf einen bestimmten Prozess kann analogisiert werden, zum Beispiel auf das Schreiben einer Datenbankabfrage, die Aufzeichnungen anfordert, die mit Kriterien über verschiedene Felder der Aufzeichnung übereinstimmen. Die Anfrage der vorliegenden Erfindung unterscheidet sich von einer normalen Datenbankabfrage insofern, als sie nicht nur eine Anfrage für bereits existierende Daten ist, sondern eine fortlaufende Anfrage für eine Sammlung von Ereignissen, die noch eintreten oder existieren müssen.

[0027] Wenn eine Teilnahme für ein Ereignis wie zum Beispiel ein Ereignis, das bei einem Point-Product **415** eintritt, ausgeführt wird, wird eine Teilnahmeanfrage an den Ereignisverwalter **411** auf Knoten b **410** gesendet. Der Ereignisverwalter **411** empfängt die Anfrage und fügt diese Anfrage zu seiner Liste ausstehender Anfragen hinzu, die zum Beispiel im Speicher gespeichert werden können. Der Ereignisverwalter **411** überprüft ausstehende Ereignisse, die vorher zum Beispiel in einer Wandtafel gespeichert wurden, um zu sehen, ob sie mit den Anfragekriterien übereinstimmen. Jedes übereinstimmende Ereignis wird an den Anfrager, zum Beispiel den Teilnehmer des Ereignisses weitergeleitet, zum Beispiel veröffentlicht. Sämtliche neuen Ereignisse, die empfangen werden und mit den Teilnahmekriterien übereinstimmen, werden auch weitergeleitet. Dies kann solan-

ge fortgesetzt werden, bis die Teilnahme annulliert wird.

[0028] In einer beispielhaften Form der vorliegenden Erfindung wird einer Teilnahme bei ihrer Erstellung eine eindeutige ID zugewiesen. Die eindeutige ID und eine Datengruppe, aus der diese Anforderung kam, definieren die Teilnahme eindeutig. Eine Teilnahme wird durch Rufen einer API mit einer zurückgesendeten Anfragehandhabung von der ursprünglichen Teilnahme annulliert. Hieraus folgt das Senden einer Annullierungsnachricht an den Ereignisverwalter mit der jeweiligen Anfrage-ID. Der Ereignisverwalter kann dann die Annullierungsanfrage mit der ursprünglichen Teilnahme in Übereinstimmung bringen und sie aus einer Verarbeitungswarteschlange des Ereignisverwalters entfernen.

[0029] Alle Ereignisse, die auf einem Knoten veröffentlicht werden, werden von dem Ereignisverwalter des Knotens empfangen. Der Ereignisverwalter empfängt und verwaltet auch alle Anfragen, die durch Prozesse von seinem Knoten und anderen Knoten gesendet wurden. Bei Empfang eines Ereignisses weist der Ereignisverwalter auch eine Ereignis-ID zu. Der Ereignisverwalter bestimmt, ob das Ereignis eine Bedingung ist, und falls ja, überprüft der Ereignisverwalter zum Beispiel eine Wandtafel, um zu bestimmen, ob das Ereignis mit einer existierenden Bedingung übereinstimmt. Falls ja, wird dem Bedingungsereignis eine Bedingungs-ID der existierenden Bedingung zugewiesen und als eine Aktualisierung auf die existierende Bedingung angewendet. Wenn ferner die Archivierung aktiviert wird, wird das Ereignis archiviert. In einem Ausführungsbeispiel der vorliegenden Erfindung kann das Archivieren zum Beispiel das Speichern des Ereignisses in einer Datenbank wie einem Einfachdateiarchiv einschließen. Eine getrennte Datei kann für jeden Kalendertag benutzt werden. Ereignisse können in das Archiv für den Tag des Zeitstempels in dem Ereignis als bezüglich des Zeilenvorschubs begrenzte Aufzeichnungen in der Reihenfolge geschrieben werden, in der sie empfangen werden. Der Ereignisverwalter überprüft auch alle ausstehenden Teilnahmeanfragen. Wenn im Hinblick auf ein Ereignis, das keine Bedingungsaktualisierung ist, das empfangene Ereignis mit der Anfrage übereinstimmt, leitet der Ereignisverwalter das Ereignis an den Anforderer weiter. Im Falle einer Aktualisierung einer Bedingung wird das aktualisierte Ereignis mit der Anfrage in Übereinstimmung gebracht. Dementsprechend wird das aktualisierte Ereignis von dem Ereignisverwalter an den Anforderer weitergeleitet.

[0030] Ereignisfilter beschreiben identifizierende Kriterien für die betreffenden Ereignisse und ermöglichen eine Spezifikation verschiedener Vergleichsformen, die für jedes der Felder eines Ereignisses spezifiziert werden können, das Schlüsselwertpaare der erweiterten Ereignisstruktur aufweist. Ein Ereignisfilter ist zum Beispiel ein boolescher Ausdruck, der aus untergeordneten Ausdrücken gebildet ist, die ein Feld eines Ereignisses mit einem Wert vergleichen, der von einem Benutzer spezifiziert wird. Ereignisfilter sind zum Beispiel dem "WHERE"-Satz in der strukturierten Abfragesprache (Structured Quert Language = SQL) ähnlich. Der grundlegende untergeordnete Ausdruck eines Ereignisfilters ist zum Beispiel ein Satz, der ein Feld eines Ereignisses mit einem Wert, zum Beispiel node=ptisun20 vergleicht. Der untergeordnete Ausdruck node=ptisun20 bedeutet, dass das Knotenfeld des Ereignisses mit der Kette „ptisun20" exakt übereinstimmen muss. Irgendeines der Felder der Ereignisstruktur kann wie das Knotenfeld in dem Beispiel benutzt werden, mit Ausnahme der Schlüsselfeldanordnung und Wertefeldanordnung, die eine spezielle Syntax erfordern. Wenn zum Beispiel eines der hinzugefügten Schlüsselwertpaare Folgendes wäre:
key value
FileSystem /usr,
wäre ein exakt übereinstimmender Filter für dieses Schlüsselwertpaar
keyfield.FileSystem="/usr". Ferner könnte die Existenz eines Schlüssels mit jedem beliebigen Wert geprüft werden, indem geprüft wird, dass er nicht mit einem Null-Wert übereinstimmt, zum Beispiel keyfield.FileSystem !="□. Die Ereignisfilter können an einem beliebigen Platz gespeichert werden, an dem eine Textkette gespeichert werden könnte, zum Beispiel in einer Konfigurationsdatei wie einer Einfachtextdatei, in einer Datenbank, im Speicher, als C-Quellencode (zum Beispiel in einem Programm hartcodiert) usw.

[0031] Das Filtern ist auch bezüglich der Werte von Schlüsselwertpaaren des Ereignisses verfügbar. Wie in dem vorherigen Beispiel kann es einen Schlüssel „FileSystem" mit einem zugehörigen Wert geben, der der Name eines spezifischen Dateisystems ist. Die gewünschten Ereignisse können nur diejenigen für ein bestimmtes Dateisystem, zum Beispiel /usr sein. Der Filtermechanismus für entsprechende Werte eines Schlüssels spezifiziert den Schlüssel und prüft den zugehörigen Wert. Zur Unterscheidung von Schlüsseln aus anderen Feldern des Ereignisses wird eine spezielle Syntax benutzt, die einen anderen Namensraum für die Schlüssel aus den vordefinierten Feldern des Ereignisses ermöglicht. Die Syntax lautet "keyfield.[name]". Eine beispielhafte Prüfung für den Wert /usr eines Schlüsselwertpaares wäre zum Beispiel keyfield.FileSystem= /usr. In einer beispielhaften Form der vorliegenden Erfindung kann der Ereignisfilter Vergleichsoperatoren wie = und eine vollständige Übereinstimmung des regulären Ausdrucks aufweisen, die mit dem Operator „like" angegeben wird. Ein Filter könnte zum Beispiel node=ptisun05 sein. Eine Filterübereinstimmung aller Knotenwer-

te, die einem ptisun[#]-Muster folgen, wäre node like "ptisun[0–9]+". Im Folgenden wird eine beispielhafte Liste von Ereignisfilter-Vergleichsoperatoren bereitgestellt: >= (größer als oder gleich), <= (kleiner als oder gleich), > (größer als), < (kleiner als), = (gleich), like (stimmt mit einem regulären Ausdruck überein), likeci (fallunabhängige Kettenübereinstimmung) und != (nicht gleich).

[0032] In einer beispielhaften Form der vorliegenden Erfindung können die folgenden Token, Produktionsregeln und eine Ereignisfilterdefinition, die mittels noch eines anderen Kompilierers compiler (yacc) implementiert wird, benutzt werden.

```
%token FILTER
%token <symp> NAME
%token <symp> FUNC_PART
%token <keywp> BOOLEAN
%token <keywp> COMPOP
%token <keywp> SEPARATOR
%token<keywp> LEFTPAREN
%token <keywp> RIGHTPAREN
/* get rid of expression grammar shift/reduce conflict */
%left BOOLEAN
%type <evalp> filter
%type <evalp> statement
%%
statement:          FILTER filter NO_MORE_TOKENS
filter:   NAME COMPOP NAME
     |    FUNC_PART SEPARATOR NAME COMPOP NAME
     |    NAME COMPOP FUNCPART SEPARATOR NAME
     |    filter BOOLEAN filter
|    LEFTPAREN filter RIGHTPAREN
```

[0033] In einer beispielhaften Form der vorliegenden Erfindung kann der Ereignisverwalter **411** als ein Daemon (zum Beispiel ein Agentenprogramm, das kontinuierlich auf einem UNIX-Server betrieben wird und den Client-Systemen auf dem Netz Ressourcen bereitstellt) umgesetzt sein. Bei Empfang eines Ereignisses bestimmt der Ereignisverwalter **411** die Disposition des Ereignisses, einschließlich dessen, ob er das Ereignis bereits empfangen hat und ob sich der Ereigniszustand geändert hat. Der Ereignisverwalter **411** schreibt das Ereignis auch in ein lokales Ereignisarchiv **412** und leitet das Ereignis an alle Clients, die an dem Ereignisinhalt teilnehmen. Zum Beispiel kann der Ereignisverwalter 411 dem Ereigniskorrelator **413** und dem Ereignisverwalter auf einem Knoten **402** Ereignisinformation bereitstellen. Das Ereignisarchiv **412** kann einen Ereignisarchiv-Dienstprozessor aufweisen. Der Dienstprozessor des Ereignisarchivs **412** liest Ereignisse aus dem Ereignisarchiv. Teilnehmer können jede beliebige Instanz eines Ereigniskorrelators **413**, die eine Alarmregel aufweist, die an dem Ereignis teilnimmt, und zum Beispiel einen Direktor aufweisen, der ein Ressourcenobjekt oder ein Produkt enthält, das an Aktualisierungsereignissen über die Inhalte in einem Zentralspeicher wie einem Datenaustauschdienst (DEX) teilnimmt.

[0034] Der Ereigniskorrelator **413** kann zum Beispiel einen Ereigniskorrelator-Dienstprozessor aufweisen. Der Ereigniskorrelator **413** verwendet ein Benutzerregelwerk, das in einer Korrelationsregel spezifiziert ist. Ein Alarm ist ein Ereignistyp, der Teilnehmer über eine bedeutende Bedingung oder ein bedeutendes Vorkommnis informiert. Die Bestimmung eines Alarms kann die Gegenwart eines einzigen Ereignisses, die Gegenwart eines bestimmten existierenden Zustands beziehungsweise von Zuständen, wenn ein anderes Ereignis eintritt, oder das erneute Eintreten eines bestimmten Ereignisses innerhalb eines festen Zeitfensters aufweisen. Ferner kann ein Alarm eine Kombination des erneuten Eintretens eines bestimmten Ereignisses innerhalb eines festen Zeitfensters sein, wenn ein bestimmter Zustand oder bestimmte Zustände vorliegen.

[0035] Die Ereignisse, die bestimmen, ob ein Alarm eintritt, können auf Ereignissen auf dem gleichen Knoten wie der Ereigniskorrelator **413** beruhen oder aus einem oder mehreren anderen Knoten **401**, **410** kommen. Ein Alarm kann auch mit einem automatischen Antwortregelwerk verbunden sein, das einer Antwortmaschine **414** ermöglicht, jegliche automatische Mitteilung oder Korrektur des Alarms zu steuern. Der Ereigniskorrelator **413** kann zum Beispiel einen Alarm, eine modifizierte Version eines empfangenen Ereignisses oder ein vollkommen neues Ereignis, das mit einer anderen Alarmregel korreliert werden kann, erzeugen.

[0036] Ereignisse können durch eine Alarmregel korreliert werden. Die Grundlage einer Alarmregel ist die Bestimmung der Ereignisse, die analysiert werden sollen. Alarmregeln können erzeugt werden, um zu definieren, welches einzelne Ereignis oder welcher Satz von Ereignissen für ein bedeutendes Vorkommnis steht, auf das sie reagieren. Die Alarmregel kann eine Beschreibung und Logik aufweisen, die die Ereignisse beschreibt, die ein Vorkommnis und eine Definition der Antwort des Ereignisverwaltungssystems auf das Vorkommnis repräsentieren. Das Ergebnis der Korrelation ist die Erzeugung eines oder mehrerer Ereignisse. Eine Alarmregel kann zum Beispiel durch den Direktor **404** definiert werden.

[0037] Eine Alarmregel weist zum Beispiel einen eindeutigen Namen auf, der ermöglicht, dass auf die Regel in anderen Ausdrücken mit Hilfe des Namens Bezug genommen werden kann. Eine Alarmregel kann auch eine Liste von Knoten (zum Beispiel eine Verteilungsliste) aufweisen, auf der die Regel gestartet werden soll. Die Verteilungsliste für eine Regel kann zum Beispiel durch eine Liste der Knoten spezifiziert werden, die die Regel ausführen sollen, oder eine Abfrage, die die Knoten spezifiziert, die die Regel ausführen sollen. Die Abfrage kann zum Beispiel "alle" lauten, was jedem Knoten entspricht, der einen Ereigniskorrelator startet. Ferner kann eine Abfrage in einem Zentralspeicher wie dem DEX **405** laufen, um bestimmte Kriterien der Knoten in dem Unternehmen zu bestimmen, wie verfügbare Ereignisse. Eine Alarmregel weist auch eine Definition von Korrelationsfiltern auf.

[0038] Korrelationsfilter ähneln den Ereignisfiltern, die bei der Ereignisteilnahme benutzt werden, wie in den folgenden ebenfalls abhängigen Patentanmeldungen beschrieben ist: Bevollmächtigtenaktenzeichen 22074661/25529, eingereicht am 31. Dezember 1998 mit der Bezeichnung METHOD AND APPARATUS FOR A USER EXTENSIBLE EVENT STRUCTURE und Bevollmächtigtenaktenzeichen 22074661/25546, eingereicht am 31. Dezember 1998 mit der Bezeichnung METHOD AND APPARATUS FOR THE DYNAMIC FILTERING AND ROUTING OF EVENTS; und Platinum Provision Common Services Reference Guide, Version 1.08 (Oktober 1998, Platinum technology, inc.). Den Korrelationsfiltern kann jedoch ein Name zugewiesen werden und sie können Qualifikatoren aufweisen. Die Qualifikatoren können zum Beispiel benutzt werden, um eine Vielzahl von Ereignissen in einer einzigen Regel zu korrelieren. Die Qualifikatoren können auch anzeigen, dass der Filter nicht auf das gegenwärtig Eingabeereignis anwendbar ist. Die Alarmregel kann auch Logikprüffilter und Aktualisierungs/Sendeereignisse aufweisen, die auf den Ergebnissen der Prüfung basieren, zum Beispiel eine if/then/else-Logik, die eine Prüfung von booleschen Kombinationen der Filter und das Ausführen von Aktionen ermöglicht, die bewirken, dass Ereignisse erzeugt werden.

[0039] In einer beispielhaften Form der vorliegenden Erfindung benutzt die Alarmregel das folgende Format:

```
if( correlation filter expression)
{    <correlation event expression 1>;
     <correlation event expression 2>;
     ...
     <correlation event expression n>;}
```

[0040] In dem oben erwähnten beispielhaften Format folgt der Korrelationsfilterausdruck der "If"-Aussage und ist in Klammern eingeschlossen. Dies ist ein logischer Ausdruck, der einen oder mehrere Korrelationsfilternamen enthält, die durch boolesche Operatoren wie AND und OR verbunden sind. Ferner können zusätzliche Klammern enthalten sein, um die Auswertungsreihenfolge zu steuern. In dem oben erwähnten beispielhaften Format ist der Korrelationsereignisausdruck in einem einzigen Satz von geschwungenen Klammern eingeschlossen. Die geschwungene aufgehende und zugehende Klammer kann sich in getrennten Zeilen befinden oder jede Ereignisdefinition kann sich in einer getrennten Zeile befinden, die mit einem Semikolon endet. Der Korrelationsfilterausdruck definiert, welche Filter erfüllt sein müssen, damit die Korrelation ausgeführt werden kann. Wie ein Filter erfüllt wird (zum Beispiel als wahr bewertet wird), hängt von der Qualifizierung ab, die dem Filter zugewiesen wird. Für einen Standardfilter muss das gegenwärtige Eingabeereignis übereinstimmen. Für einen Zählfilter muss das gegenwärtige Ereignis einen Satz von Ereignissen vervollständigen, der in einem Zeitfenster für den zu erfüllenden Filter empfangen wird. Ferner ist ein conditionExists-Filter erfüllt, wenn in der

Vergangenheit ein gegebenes Ereignis empfangen wurde, das anzeigt, dass eine bestimmte Bedingung noch immer existiert. Der Korrelationsfilterausdruck kann Kriterien hinsichtlich eines vergangenen und gegenwärtigen Ereignisses durch jede beliebige Kombination über einen booleschen Operator dieser einzelnen Filter spezifizieren, einschließlich der Negation davon (zum Beispiel die Bedingung ist gegenwärtig nicht vorhanden, die Zählung ist nicht vollendet worden oder das gegenwärtige Ereignis stimmt nicht überein).

[0041] Wenn zum Beispiel f1 ein standardgemäßer, unqualifizierter Filter ist, der für das gegenwärtige Ereignis gilt, zeigt der Ausdruck „if (f1)" an, dass das gegenwärtige Ereignis mit diesem Filter übereinstimmen muss, damit die Korrelation ausgeführt wird. Wenn dann ein zweiter standardgemäßer Filter f2 vorhanden wäre, könnten beide Filter mit dem Ausdruck „if (f1 or f2)" geprüft werden. Wenn das Eingabeereignis in diesem Ausdruck entweder mit f1 oder f2 übereinstimmt, wird die Korrelation ausgeführt. Der Korrelationsfilterausdruck kann auch verschachtelte Bedingungen wie einen logischen „if/else if"-Ausdruck in einem Format aufweisen, das folgendem ähnlich ist:

```
If(F1)
{ <correlation event 1>;      }
Else If (F2)
{ <correlation event 2>;      }
```

[0042] Eine Alarmregel der vorliegenden Erfindung kann mit oder ohne Logik benutzt werden. In Abwesenheit einer Steuerlogik wird eine Aktionsaussage in der Regel ausgeführt, immer wenn ein Ereignis empfangen wird. Ein Ereignis wird empfangen, immer wenn ein Ereignis vorhanden ist, das mit jedem beliebigen der angegebenen Filter übereinstimmt. Die Regel kann in verschiedene Gruppen von Aktionen segmentiert werden, die auf einer Prüfung dahingehend beruhen, ob einer oder mehrere der angegebenen Filter in der Regel wahr ist, zum Beispiel durch Bilden eines booleschen Ausdrucks unter Verwendung der Namen der Filter und Prüfen seiner Wahrheit zu jedem beliebigen gegebenen Zeitpunkt. Wenn zum Beispiel drei Filter mit dem Namen A, B und C vorhanden sind, könnte eine Prüfung für (A und (B oder C)) ausgeführt werden. Dies würde bedeuten, dass der Ausdruck wahr ist, wenn Filter A wahr ist und Filter B oder C wahr ist. Diese Regel kann zum Beispiel wie folgt ausgedrückt werden.

```
If (A and (B or C))
SendModifedAlarm (–110);
else
SendModifiedAlarm (–1 0);
```

[0043] In diesem Beispiel werden basierend auf der Prüfung, welche Filter wahr sind, unterschiedliche Alarmstufen festgelegt.

[0044] In einer beispielhaften Form der vorliegenden Erfindung gibt es vier Aktionsaussagen SendEvent, SendModifiedInputEvent, SendModifiedAlarm und SendModifiedCondition, die in der Syntax der Ereigniskorrelationsregel unterstützt werden, die Ereignisse sendet. Die Aktionsaussage SendEvent beginnt mit einem neu erschaffenen Ereignis und muss jedes vorgeschriebene Feld für das Ereignis aufweisen, das als ein Argument geliefert wird, mit Ausnahme von Feldern, denen vorgegebene Werte wie Zeit, Knoten und Agentenknoten zugewiesen werden können. Jeder beliebige Ereignistyp kann mit dieser Aktion gesendet werden. Die Aktionsaussage SendModifiedInputEvent benutzt das Eingabeergebnis als eine Vorlage für das Ausgabeereignis, wobei nur Felder ersetzt werden, die als Argumente bezüglich SendModifiedInputEvent spezifiziert werden. Die Ereigniszeit wird auf die gegenwärtige Zeit aktualisiert, und wenn dies ein Problem für eine Bedingung oder einen Alarm (Bedingungen und Alarme erforderten, dass die Ereigniszeit > Bedingungszeit ist) verursacht, wird die Zeit zum Beispiel auf die Zeit des Eingabeereignisses (die bereits größer als die condition_time sein musste) plus einer Sekunde eingestellt. Die Aktionsaussage SendModifiedAlarm wird auf SendModifiedInputEvent gebildet. Der einzige Unterschied besteht darin, dass sie nur einen Alarm sendet, zum Beispiel eine Einstellung des Ereignistyps auf alarm_set erzwingt. Sie steuert auch die Erschaffung eines Alarms aus einem diskreten Ereignis, das keine Bedingungszeit aufweist, wobei die Bedingungszeit auf die gegenwärtige Zeit eingestellt wird. Die Aktionsaussage SendModifiedCondition ist die gleiche wie SendModifiedAlarm, außer dass der Typ auf condition_set eingestellt werden muss.

[0045] Die Qualifikatoren für einen Korrelationsfilter können zum Beispiel ConditionExists, Count und CountDiscrete aufweisen. Der Qualifikator ConditionExists zeigt an, dass der Filter auf Ereignisse angewendet werden sollte, die bereits zu dem Zeitpunkt empfangen worden sind, zu dem das gegenwärtige Ereignis empfangen wurde. Wenn ein vorheriges zustandsorientiertes Ereignis existiert, das mit dem spezifizierten Filter übereinstimmt, ist der Filterausdruck wahr. Anderenfalls ist er falsch. Der Qualifikator Count wird benutzt, um die

Anzahl von Malen zu verfolgen, die ein übereinstimmendes Ereignis innerhalb eines spezifizierten Zeitraums eingetreten ist. Die Spezifikation lautet Zählung (Anzahl von Malen, Zeitintervall, Filter). Beim Initialisieren wird der Satz gelöscht, so dass die Zählung bei Empfang des nächsten übereinstimmenden Ereignisses null sein wird. Das Zeitfenster kann scrollen, wobei die ältesten übereinstimmenden Ereignisse von der ausstehenden Liste entfernt werden, da sie sich von dem zuletzt hinzugefügten Ereignis durch das angegebene Zeitintervall unterscheiden. Der Qualifikator CountDiscrete zählt diskrete Ereignisse. Er ist die Entsprechung von Count, wobei sein Filter zum Beispiel zu filter filter AND type=discrete modifiziert wird.

[0046] Ereignisse können aus jedem beliebigen Knoten abgerufen werden, bei dem die Alarmregel abläuft. In einer beispielhaften Form der vorliegenden Erfindung kann die Bestimmung, wo eine Regel ablaufen soll, von der Beschaffenheit der Regel abhängen. Wenn die Regel Ereignisse spezifiziert, die zum Beispiel nur auf einem lokalen Knoten eintreten, dann muss die Regel auf allen Knoten laufen, bei denen solch eine Korrelation gewünscht wird. Wenn die Regel zum Beispiel eine bestimmte Korrelationsform für ein Ereignis über die CPU-Benutzung einer Maschine ausführen soll und der Filter keinen Knoten bestimmt hat, dann empfängt der Ereigniskorrelator, der die Regel startet, nur Ereignisse über die CPU-Benutzung auf dem Knoten, bei dem die Teilnahme stattfindet. Dementsprechend können der Filter und die Regel generisch und auf jede Maschine anwendbar sein. Folglich sollte die Verteilungsliste für die Regel jeden Knoten aufweisen, auf dem die jeweilige Korrelation stattfindet.

[0047] Eine generische Regel ermöglicht auch, dass eine einzige Definition auf den gleichen Problemtyp auf jedem Host in dem Unternehmen anwendbar ist, wodurch die Anzahl von zu verwaltenden und konfigurierenden Regeln verringert wird. Eine generische Regel ermöglicht die Analyse und Korrektur des Problems, das dort zu lokalisieren ist, wo es auftritt, zum Beispiel auf jedem Knoten.

[0048] Die Alarmregel **406** kann einen Alarmregel-Dienstprozessor aufweisen, der Alarmregeln und Antwortregelwerke verwaltet, die in einer zentralen Speicherstelle wie dem DEX **405** gespeichert sind. In einer beispielhaften Form der vorliegenden Erfindung ist der DEX **405** eine zentrale Speicherstelle für wieder verwendbare Information, einschließlich Datenkonfiguration, Produktinformation, Ereignisinformation, Sicherheitsinformation und Regeln. Der DEX **405** kann ein verteilter, gemeinsamer Datenspeicher mit einer Vielzahl von Plattformen und Datenformaten, mit mehreren Datenspeichern, relationalen Daten, Datenstellen und Dateien sein.

[0049] Die Antwortmaschine **414** führt ein Antwortregelwerk aus. Die Antwortmaschine **414** weist mehrere Prozesse auf. Das Antwortregelwerk ist zum Beispiel ein logischer Ausdruck, der eine Liste von Aktionen enthält, die durch logische Aussagen verbunden sind, und der durch einen Auslösealarm aufgerufen wird, der von dem Ereigniskorrelator **413** erzeugt wird. Eine Vielzahl von Aktionen kann in dem Antwortregelwerk definiert und ausgeführt werden. In einer beispielhaften Form der vorliegenden Erfindung kann die Vielzahl von Aktionen in einer aufgelisteten Sequenz oder mit hinzugefügter Logik vorliegen, um jede Aktion nach Empfang eines Antwortcodes einer anderen Aktion oder eines spezifischen Feldes, das in dem Alarm gesendet wird, abhängig zu machen. Das Antwortregelwerk kann durch Definieren eines Satzes von Aktionen und Erstellen eines Ausdrucks erzeugt werden, der eine oder mehrere dieser Aktionen in einer logischen Sequenz angibt. Die Aktionen können global sein und als eine Vielzahl von Antwortregelwerken bezeichnet werden. In einem Ausführungsbeispiel der vorliegenden Erfindung fügt die Antwortmaschine **414** bei Vollendung jedes Schritts auch Information über den Erfolg jeder Aktion zu dem Auslösealarm hinzu. Dies kann zum Beispiel durch Senden eines Aktualisierungsalarmereignisses erreicht werden, das das Ereignis mit der sequentiellen Zahl des gerade vollendeten Schritts, der Art des Schritts (zum Beispiel E-Mail/Seite) und in dem Ereignisarchiv mit dem Namen des Aktionsschritts aktualisiert. Auf diese Information kann durch einen Direktor zugegriffen werden.

[0050] In einer beispielhaften Form der vorliegenden Erfindung kann die Antwortmaschine **414** zum Beispiel einen ersten Prozess, einen zweiten Prozess und einen dritten Prozess aufweisen. Der erste Prozess nimmt an Ereignissen teil, die von einem Ereigniskorrelator **413** gesendet werden, der auf dem gleichen Knoten **410** wie die Antwortmaschine **414** läuft. Nach Empfang eines Alarms, der ein zugehöriges Antwortregelwerk aufweist, ruft der erste Prozess den zweiten Prozess auf, um das Regelwerk auszuführen. Der erste Prozess kann je nach den Einstellungen in Konfigurationsdateien, die mit den entsprechenden Prozessen verbunden sind, eine Vielzahl von Instanzen des zweiten Prozesses aufrufen. Der zweite Prozess führt eine Aktion aus, die das Senden von Ereignissen erfordert, und ruft den dritten Prozess auf, um irgendwelche anderen Aktionen auszuführen. Der dritte Prozess der Antwortmaschine **414** kann auch eine Antwortregelwerkaktion ausführen, die einen Ruf an das Betriebssystem erfordert, wie das Senden einer E-Mail-Nachricht, das Aufrufen eines Scripts, das Senden einer Seite, das Schreiben einer Nachricht an eine Datei oder das Weiterleiten des Ereignisses als eine Programmunterbrechung des Simple Network Management Protocol (SNMP).

[0051] In einer beispielhaften Form der vorliegenden Erfindung können die folgenden Token, Produktionsregeln und Korrelationssprache, die mittels yacc angewendet wird, benutzt werden.

```
%union {
     struct FilterEvalNode *evalp;
     struct FilterSymTab *symp;
     struct keyword_table *keywp;
     struct ARG_LIST *argp;}

%nonassoc LOWER_THAN_ELSE
%nonassoc <symp> ELSE
%token NO_MORE_TOKENS
%token <symp> DECLARED_FILTER
%token <symp> ACTION_FUNC
%token ALARM
%token <symp> NAME
%token <keywp> IF
%token <keywp> CONDITION
%token<keywp> CONDITIONEXISTS
%token <keywp> BOOLEAN
%token <keywp> UNARYOP
%token <keywp> LEFTPAREN
```

```
%token <keywp> RIGHTPAREN
%token <keywp> LEFTCURLYBRACE
%token <keywp> RIGHTCURLYBRACE
%token <keywp> ENDSTATEMENT
%token <keywp> ASSIGN
%token <keywp> COUNTDISCRETE
%token <keywp> COUNT
%token <keywp> SECONDS
%token <keywp> MINUTES
%token <keywp> HOURS
%token <keywp> DAYS
%token <keywp> COMMA
%left BOOLEAN
%left UNARYOP
%type <evalp> Actions
%type <evalp> Action
%type<argp> ArgList
%type <argp> Arg
%type <evalp> AlarmPrograms
%type <evalp> AlarmProgram
%type <evalp> filter
%type <evalp> statement
%type <evalp> declaredFilterExpr
%type <symp> timeInterval
%type <symp> nameAsInteger
%type <evalp> countType
%type <keywp> maybeNOT

statement:     ALARM AlarmPrograms NO_MORE_TOKENS
AlarmPrograms:      AlarmPrograms AlarmProgram
          |    AlarmProgram
          |    UNARYOP AlarmProgram
AlarmProgram:       NAME ASSIGN filter
          |    NAME ASSIGN maybeNOT CONDITION filter
          |    NAME    ASSIGN    maybeNOT    CONDITIONEXISTS
     LEFTPAREN filter
```

```
          |    NAME ASSIGN maybeNOT countType LEFTPAREN
     NAME COMMA
          timeInterval COMMA filter RIGHTPAREN
          |    Action
Actions:           Actions Action
Action:       ACTION_FUNC LEFTPAREN ArgList RIGHTPAREN
ENDSTATEMENT
          |    IF LEFTPAREN declaredFilterExpr RIGHTPAREN
     Action %precLOWER_THAN_ELSE
          |    IF LEFTPAREN declaredFilterExpr RIGHTPAREN
     Action ELSE
               Action
          |    LEFTCURLYBRACE Actions RIGHTCURLYBRACE
          |    ENDSTATEMENT
ArgList:           ArgList Arg
          |    ArgList COMMA Arg
          |    Arg
Arg:          NAME
countType:    COUNT
          |    COUNTDISCRETE
timeInterval: nameAsInteger
          |    nameAsInteger SECONDS
          |    nameAsInteger MINUTES
          |    nameAsInteger HOURS •    ,
          |    nameAsInteger DAYS
nameAsInteger:     NAME
declaredFilterExpr: DECLARED_FILTER
          |    declaredFilterExpr                    BOOLEAN
     declaredFilterExpr
          |    LEFTPAREN declaredFilterExpr RIGHTPAREN
          |    UNARYOP declaredFilterExpr
maybeNOT:     /* empty */
          |    UNARYOP
```

[0052] Das Ereignisverwaltungssystem der vorliegenden Erfindung stellt eine hierarchische Konsolidierung auf einer Teilnahmebasis bereit. Zum Beispiel hat ein Direktor **403** basierend auf der Teilnahme an den verschiedenen Knoten, die Ereignisinformation bereitstellen, eine konsolidierte Ansicht von Ereignissen in dem Unternehmen. Die Ereignisinformation kann als eine einzige, vereinigte Ansicht präsentiert werden, kann jedoch auf einer Vielzahl von Teilnahmen bezüglich verschiedener Ereignisquellen basieren. Die Konfiguration von Korrelationsregeln ermöglicht, dass ein Satz von Korrelationsregeln eine Korrelations- oder Konsolidierungshierarchie implementiert, indem der gleiche Teilnahmemechanismus innerhalb der Regeln benutzt wird, den der Direktor **403** benutzen kann, um Ereignisse aus einer Vielzahl von Quellen zu konsolidieren. Wenn sich eine Anwendung, die auf Knoten A abläuft, zum Beispiel auf Datenbanken auf Knoten B und Knoten C stützt, kann eine einzige Regel, die diese Anwendung überwacht, Ereignisse über die Anwendung auf Knoten

A, die Fähigkeit zur Bewahrung der Verbindbarkeit mit Knoten B und C und den Status der Datenbanken auf Knoten B und C konsolidieren. Dies kann durch Konfigurieren der Filter in der Regel, um diese Quellen zu spezifizieren, erreicht werden.

[0053] Ferner kann die Konsolidierung aller Fehler von einem Knoten zu einem anderen Knoten mit geeigneten konfigurierten Korrelationsregeln in ähnlicher Weise ausgeführt werden. Zum Beispiel kann Knoten A durch die Benutzung einer Regel Probleme von Knoten B und C konsolidieren. Die Leistung des Ereigniskorrelators mit Bezug auf lokal erzeugte Ereignisse kann zum Beispiel eine Konsolidierung der Ereignisse des Knotens in Alarme sein. Gemäß einer Ausführungsform der vorliegenden Erfindung kann der Ereigniskorrelator auf Knoten A an den Alarmen teilnehmen, die auf Knoten B und C erzeugt werden. Dies bringt alle angegebenen Probleme effektiv zu Knoten A. Die angegebenen Probleme können dann als Alarme auf Knoten A basierend auf den Inhalten der Regel neu angegeben werden oder könnten in der Regel benutzt werden, um bestimmte Korrelationstypen bezüglich der Probleme zu bestimmen, die auf Knoten B, C und A berichtet werden. Das erneute Angeben von Problemen würde eine redundante lokale Kopie oder eine Teilnahme ermöglichen, die nur einen jeweiligen Knoten spezifiziert, um einem die Ansicht der Probleme zu ermöglichen, wobei alle drei Knoten umfasst werden.

[0054] Dementsprechend erfordert die hierarchische Konsolidierung, die von dem Ereignisverwaltungssystem der vorliegenden Erfindung bereitgestellt wird, keine zusätzliche Konfiguration außerhalb der Regel selbst. Es besteht kein Bedarf an der Beibehaltung der Beziehungen außerhalb der Beziehungen zwischen Regeln. Die Hierarchie kann auf einer Regelbasis effektiv variieren. Ein Knoten kann ein Konsolidierungspunkt für bestimmte Probleme sein, während ein anderer Knoten der Konsolidierungsknoten für einen anderen Satz sein kann, wobei diese beiden Knoten zur gegenseitigen Konsolidierung beitragen.

[0055] Das Ereignisverwaltungssystem der vorliegenden Erfindung kann dynamisch, zum Beispiel während der Ereigniskorrelator abläuft, bestimmen, ob eine Regel geändert, hinzugefügt oder gelöscht wurde. Die benutzerdefinierten Alarmregeln werden zum Beispiel in einem Alarmregelspeicher (ARS) **406** in einem zentralen Speicher in dem DEX **405** verwaltet. Das Bearbeiten oder Hinzufügen von Regeln kann über einen Direktor **403** ausgeführt werden. Fig. 5 stellt ein beispielhaftes Flussdiagramm zur dynamischen Ausführung von hinzugefügten oder modifizierten Regeln dar. Wie in Fig. 5 dargestellt, fragt der Direktor **403** in Schritt **510** den ARS **406** nach Regeln ab und schreibt jegliche Veränderungen zurück in den ARS **406**. Der ARS **406** veröffentlicht ein Ereignis, immer wenn er die Regeln in dem ARS **406** aktualisiert. Der Ereigniskorrelator **413** auf jedem der Knoten nimmt an diesem Ereignis teil. Bei Empfang dieses Ereignisses kann jeder der Ereigniskorrelator **413** mit dem ARS **406** in Kontakt treten, um zu bestimmen, ob die Regeln, die sie ausführen, aktualisiert worden sind. Zum Beispiel kann das Ereignis, das von dem ARS **406** veröffentlicht wird, anzeigen, dass sich der Regelspeicher verändert hat.

[0056] In Schritt **530** tritt der entsprechende Ereigniskorrelator **413** mit dem ARS **406** in Kontakt und fordert seinen Regelsatz an. In Schritt **540** vergleicht der Ereigniskorrelator **413** dann seinen gegenwärtigen Satz von Regeln, den er ausführt, mit den Regeln, die von dem ARS **406** empfangen werden. In Schritt **550** bedürfen unveränderte Regeln keiner Maßnahme, neue Regeln müssen zu dem Regelsatz des entsprechenden Ereigniskorrelators hinzugefügt werden und Regeln, die aus dem ARS **406** gelöscht worden sind, müssen aus dem Regelsatz des jeweiligen Ereigniskorrelators entfernt werden. In Schritt **560** werden veränderte oder zu dem Regelsatz des jeweiligen Ereigniskorrelators hinzugefügte Regeln ausgeführt. Ferner kann der Regelsatz über eine Vielzahl von Korrelatoren verteilt sein. Dementsprechend kann eine Korrelation, die von einem bestimmten Knoten ausgeführt wird, an einen anderen Knoten umgeleitet werden, indem die Regelverteilungsliste geändert wird.

[0057] Eine Regel kann zu dem Ereignisverwaltungssystem gemäß einer Ausführungsform der vorliegenden Erfindung dynamisch hinzugefügt werden, indem eine Regel zu einem laufenden Ereigniskorrelator hinzugefügt wird, da jede Regel ihren eigenen Ereignisstrom verwaltet. Das Hinzufügen oder Entfernen einer Teilnahme einer Regel hat keinen Einfluss auf irgendeine andere Regel. Eine neue Regel kann dazu führen, dass der Ereigniskorrelator Ereignisse empfängt, die er vorher nicht empfangen hat. Diese Ereignisse ändern nichts daran, wie irgendeine andere Regel bewertet wird, da diese Regeln die neuen Ereignisse nicht sehen. In ähnlicher Weise führt das Löschen einer Regel dazu, dass der Ereigniskorrelator weniger Ereignisse als vorher empfängt. Die Tatsache, dass diese Ereignisse nicht empfangen werden, ändert jedoch nicht die Bewertung irgendeiner anderen Regel, die von dem Ereigniskorrelator ausgeführt wird. Da jede Regel einen unabhängigen Satz von Ereignissen spezifiziert, um ihren eigenen Ereignisstrom zu empfangen, sind Veränderungen an einer Regel isoliert und haben keinen Einfluss auf irgendeine andere Regel.

[0058] Die Regel müsste syntaktisch analysiert und die geeigneten Teilnahmen gebildet werden, so dass die Regel geeignete Ereignisse empfängt. Eine Regel kann aus dem Ereignisverwaltungssystem auch dynamisch gelöscht werden, indem die ausstehenden Teilnahmen für die Regel annulliert werden, die Speicherstrukturen, die beim Nachverfolgen des Zustands der Regel und ihres Bewertungsbaums benutzt werden, entfernt werden und Alarme, die die Regel erzeugte, entfernt werden. Eine Liste benutzerdefinierter Alarmregeln wird von dem Ereignisverwaltungssystem in einem Alarmregelspeicher verwaltet. Eine Liste von Alarmen, die laut Regel den Namen des Ereignisses aufweist, wird von dem Ereignisverwaltungssystem zum Beispiel in einer Wandtafel verwaltet.

[0059] Das Ereignisverwaltungssystem der vorliegenden Erfindung ermöglicht die Bestimmung eines Eingabeereignisses und einer Alarmregel, die zu dem jeweiligen Alarm führte. In einem Ausführungsbeispiel der vorliegenden Erfindung geht das Ereignisverwaltungssystem den jeweiligen Alarmen nach, die mit den jeweiligen Regeln verbunden sind. Die Beziehung eines Alarms zu seiner Ursache kann aus einer Analyse der Regel bestimmt werden. Zum Beispiel muss zwischen den Eingabeereignissen und dem Alarm eine n-zu-1-Abbildung ausgeführt werden, wobei n die Anzahl von Eingabeereignissen ist. Im einfachsten Fall stimmt ein einziges Eingabeereignis (n = 1) mit der Erzeugung eines Alarms überein. Demgemäß wird das Eingabeereignis, das den Alarm verursachte, ohne weiteres bestimmt, da nur ein Eingabeereignis mit dem Alarm verbunden ist.

[0060] Wenn Bedingungsfilter oder Zählfilter in Gebrauch sind, wird die Abbildung jedoch zu n-zu-1. Die Abbildung kann zum Beispiel als ein Informationstupel dargestellt werden, das mehrere Eingabeereignis-IDs mit der Ausgabealarm-ID in Beziehung setzt. Die Bestimmung dessen, welche Ereignisse berücksichtigt wurden, kann zum Beispiel das gegenwärtige Eingabeereignis zum Zeitpunkt des Erzeugens der Regel aufweisen. Die Bestimmung dessen, welche Ereignisse berücksichtigt wurden, kann auch Ereignisse aufweisen, die von jedem beliebigen der zustandsorientierten Ereignisse (zum Beispiel auf Condition Exists oder Count basierende Filter, die ein oder mehrere Ereignisse spezifizieren können, die bereits eingetreten sind) gespeichert wurden, die sich auch in dem „if"-Steuerblock befanden, der zu der Regelaktion führte, um einen Alarm zu erzeugen. In einem Ausführungsbeispiel der vorliegenden Erfindung weisen all diese Ereignisse eine einzige eindeutige ID auf, außer dem erzeugten Alarm. Es kann jedoch sein, dass der Alarm keine eindeutige ID aufweist, bis das Ereignis von einem Ereignisverwalter verarbeitet worden ist. Folglich kann in Abwesenheit einer eindeutigen Kennung für den Ausgabealarm ein voller mehrteiliger Schlüssel des Alarms benutzt werden, um zu ermöglichen, dass die Aufzeichnung der n-zu-1-Beziehung ausgeschrieben wird. Dementsprechend können das Eingabeereignis und die Alarmregel, die zu dem jeweiligen Alarm führte, bestimmt werden.

[0061] Als Alternative kann in Abwesenheit einer eindeutigen Kennung für den Ausgabealarm das Aufschieben der Ausgabe der Aufzeichnung der n-zu-1-Beziehung, bis das Alarmereignis von der Regel zurück empfangen wird, ermöglichen, dass die entsprechende eindeutige ID bekannt ist. Die Aufzeichnung der n-zu-1-Beziehung kann dann als ein Ergebnis der Bereitstellung mit der entsprechenden eindeutigen ID ausgeschrieben werden. Dementsprechend können das Eingabeereignis und die Alarmregel, die zu dem entsprechenden Alarm führte, bestimmt werden.

[0062] Als Alternative kann eine eindeutige, einem Sender zugewiesene ID zum Zeitpunkt der Erzeugung eines Ereignisses wie eines Alarms benutzt werden. Die einem Sender zugewiesene ID kann anstatt einer einem Ereignisverwalter zugewiesenen eindeutigen ID benutzt werden, um zu ermöglichen, dass die Aufzeichnung der n-zu-1-Beziehung ausgeschrieben wird. Dementsprechend können das Eingabeereignis und die Alarmregel, die zu dem jeweiligen Alarm führte, bestimmt werden.

[0063] In jedem Ereignis kann die Ausgabe, zum Beispiel die Aufzeichnung der n-zu-1-Beziehung, als eine Tabelle gespeichert und als Teil des Ereignisarchivs verwaltet werden. Dementsprechend könnte der Ereignisarchiv-Dienstprozessor verstärkt werden, um eine Abfrage nach verwandten Ereignissen bereitzustellen, die das Ereignisarchiv benutzen würden, um Information zu erhalten, die von der Aufzeichnung der n-zu-1-Beziehung bereitgestellt wird.

[0064] Die oben beschriebenen Ausführungsformen sind erläuternde Beispiele der vorliegenden Erfindung und sie soll nicht derart ausgelegt werden, dass die vorliegende Erfindung auf diese bestimmten Ausführungsformen eingeschränkt ist.

Patentansprüche

1. Knoten (**410**) für ein Ereignisverwaltungssystem (**300**), umfassend:
einen Ereignisverwalter (**411**), umfassend Mittel zum Empfangen und Bereitstellen von Ereignisdaten;

dadurch gekennzeichnet, dass der Knoten (**410**) ferner Folgendes umfasst:
einen Ereigniskorrelator (**413**), der mit dem Ereignisverwalter (**411**) verbunden und gestaltet ist, um mit einem Alarmregelspeicher (**406**) außerhalb des Knotens zu kommunizieren, wobei der Ereigniskorrelator einen ersten Satz von Alarmregeln und Mittel zum Korrelieren der Ereignisdaten umfasst, die von dem Ereignisverwalter basierend auf den Alarmregeln bereitgestellt werden, wobei der Ereigniskorrelator ferner Folgendes umfasst:
Mittel zum Anfordern eines zweiten Satzes von Alarmregeln aus dem Alarmregelspeicher (**406**) in Antwort auf den Empfang eines Regelereignisses, das von dem Alarmregelspeicher (**406**) veröffentlicht wird;
Mittel zum Ändern einer beliebigen der jeweiligen Regeln aus dem ersten Satz von Alarmregeln, die in dem zweiten Satz von Alarmregeln geändert worden sind, zum Hinzufügen einer Regel zu dem ersten Satz von Alarmregeln, die zu dem zweiten Satz von Alarmregeln hinzugefügt worden ist, und zum Entfernen einer Regel aus dem ersten Satz von Alarmregeln, die aus dem zweiten Satz von Alarmregeln entfernt worden ist; und
Mittel zum Ausführen der Regeln des ersten Satzes von Alarmregeln, die geändert oder hinzugefügt worden sind; und
eine Antwortmaschine (**414**), die mit dem Ereignisverwalter (**411**) verbunden ist, wobei die Antwortmaschine Mittel zum Ausführen eines Antwortregelwerks basierend auf der Korrelation von Ereignissen durch den Ereigniskorrelator (**413**) umfasst.

2. Ereignisverwaltungssystem nach Anspruch 1, ferner umfassend ein Ereignisarchiv (**412**), das mit dem Ereignisverwalter (**411**) verbunden ist, wobei das Ereignisarchiv eine Ereignisarchivdatei und einen Ereignisarchiv-Dienstprozessor aufweist, wobei der Ereignisverwalter gestaltet ist, um Ereignisdaten in der Ereignisarchivdatei zu speichern, und der Ereignisarchiv-Dienstprozessor Mittel zum Verarbeiten der Ereignisdaten in der Ereignisarchivdatei umfasst.

3. Ereignisverwaltungssystem, umfassend einen ersten Knoten (**410**) nach Anspruch 1 oder Anspruch 2 in Kombination mit
einem zweiten Knoten (**401**), wobei der zweite Knoten Folgendes umfasst:
einen Ereignisverwalter (**402**), der Mittel zum Empfangen einer oder mehrerer Teilnahmeanfragen für ein oder mehrere Ereignisdaten von einem oder mehreren Teilnehmern (**413**) und Mittel zum Empfangen und Bereitstellen von Ereignisdaten dem einen oder den mehreren Teilnehmern (**413**) umfasst; und
wobei der Alarmregelspeicher (**406**) Folgendes umfasst:
Mittel zum Speichern von Alarmregeln, Mittel zum Veröffentlichen eines Regelereignisses in Antwort auf den Empfang einer modifizierten Alarmregel oder einer neuen Alarmregel; und
Mittel zum Bereitstellen eines Satzes von Alarmregeln einem Teilnehmer (**413**) in Antwort auf eine Anfrage für den Satz von Alarmregeln.

4. System nach Anspruch 3, ferner umfassend Mittel (**403**) zum Bearbeiten oder Hinzufügen von Alarmregeln zu dem Alarmregelspeicher (**406**).

5. System (**401**) nach Anspruch 4, wobei das Mittel (**403**) zum Bearbeiten oder Hinzufügen von Alarmregeln Folgendes umfasst:
Mittel zum Abfragen des Alarmregelspeichers (**406**) nach Alarmregeln; und
Mittel zum Schreiben von Veränderungen in den Alarmregelspeicher (**406**);

6. System nach Anspruch 3, 4 oder 5, wobei der Alarmregelspeicher (**406**) Mittel zum Veröffentlichen eines Regelereignisses in Antwort auf die Entfernung einer Alarmregel aus dem Alarmregelspeicher umfasst.

7. Ereignisverwaltungssystem nach einem der Ansprüche 3 bis 6, wobei das System mehrere Knoten (**410**) jeweils nach Anspruch 1 umfasst, wobei der Ereigniskorrelator (**413**) eines der mehreren Knoten (**410**) gestaltet ist, um mindestens jeweils zwei Ereignisse zu korrelieren, die von jedem von mindestens zwei der mehreren Ereignisverwalter (**411**) der mehreren Knoten (**410**) bereitgestellt werden.

8. Verfahren zur Verwaltung eines Computersystems, dadurch gekennzeichnet, dass das Verfahren von einem Knoten (**410**) ausgeführt wird, der einen ersten Satz von Alarmregeln umfasst, und ferner dadurch gekennzeichnet, dass das Verfahren Folgendes umfasst:
Korrelieren von Ereignisdaten, die von einem Ereignisverwalter (**402**) bereitgestellt werden, basierend auf den Alarmregeln;
Ausführen eines Antwortregelwerks basierend auf der Korrelation von Ereignissen von dem Ereigniskorrelator (**413**);
Empfangen eines Regelereignisses, das von einem Alarmregelspeicher (**406**) außerhalb des Knotens veröffentlicht wird;

Anfordern eines zweiten Satzes von Alarmregeln aus dem Alarmregelspeicher (**406**) in Antwort auf den Empfang des Regelereignisses;
Ändern einer beliebigen der jeweiligen Regeln aus dem ersten Satz von Alarmregeln, die in dem zweiten Satz von Alarmregeln geändert worden sind, Hinzufügen einer Regel zu dem ersten Satz von Alarmregeln, die zu dem zweiten Satz von Alarmregeln hinzugefügt worden ist, und Entfernen einer Regel aus dem ersten Satz von Alarmregeln, die aus dem zweiten Satz von Alarmregeln entfernt worden ist; und
Ausführen der Regeln des ersten Satzes von Alarmregeln, die geändert oder hinzugefügt worden sind.

9. Verfahren nach Anspruch 8, ferner umfassend:
Empfangen von Ereignisdaten;
Korrelieren der Ereignisdaten basierend auf den Alarmregeln; und Ausführen eines Antwortregelwerks basierend auf der Korrelation von Ereignissen.

10. Verfahren nach Anspruch 9, ferner umfassend das Starten einer oder mehrerer vorbestimmter Regeln, die mit den Ereignisdaten in Verbindung stehen.

11. Verfahren nach einem der Ansprüche 8 bis 10, ferner umfassend: dynamisches Bestimmen, ob eine Regel gelöscht wurde.

12. Verfahren nach einem der Ansprüche 8 bis 11, ferner umfassend: Speichern von Ereignisdaten in einer Ereignisarchivdatei; und Verarbeiten von Ereignisdaten in der Ereignisarchivdatei.

13. Verfahren nach einem der Ansprüche 8 bis 12, ferner umfassend, an einem zweiten Knoten:
Empfangen von Teilnahmeanfragen für Ereignisdaten von einem oder mehreren Teilnehmern (**413**);
Empfangen von Ereignisdaten und Bereitstellen der Ereignisdaten einem oder mehreren Teilnehmern (**413**);
Empfangen mindestens einer von einer modifizierten Alarmregel und einer neuen Alarmregel;
Veröffentlichen eines Regelereignisses in antwort auf den Empfang mindestens einer von einer modifizierten Alarmregel und einer neuen Alarmregel; und
Bereitstellen eines Satzes von Alarmregeln einem Teilnehmer (**413**) in Antwort auf eine Anfrage für den Satz von Alarmregeln.

14. Verfahren nach Anspruch 13, ferner umfassend einen Schritt des Bearbeitens oder Hinzufügens von Alarmregeln zu einem Alarmregelspeicher (**406**).

15. Verfahren nach Anspruch 14, wobei der Schritt des Bearbeitens oder Hinzufügens von Alarmregeln Folgendes umfasst:
Abfragen eines Alarmregelspeichers (**406**) nach Alarmregeln; und
Schreiben von Veränderungen in den Alarmregelspeicher (**406**).

16. Verfahren nach einem der Ansprüche 13 bis 15, ferner umfassend das Veröffentlichen eines Regelereignisses in Antwort auf die Entfernung einer Alarmregel.

17. Verfahren nach einem der Ansprüche 13 bis 16, ferner umfassend das Empfangen einer oder mehrerer Teilnahmen aus mehreren verteilten Computerknoten.

18. Verfahren nach Anspruch 17, wobei das Empfangen einer oder mehrerer Teilnahmen das Empfangen einer oder mehrerer Teilnahmen an einem oder mehreren Ereignissen aufweist, wobei die Ereignisse ein oder mehrere neue Ereignisse aufweisen, die durch Erzeugen eines oder mehrerer Schlüsselwertpaare dynamisch hinzugefügt werden, um das eine oder die mehreren neuen Ereignisse zu definieren.

19. Verfahren nach Anspruch 17, ferner aufweisend: Führen einer Liste der einen oder mehreren Teilnahmen und entsprechenden Teilnehmer.

20. Verfahren nach Anspruch 19, ferner aufweisend: Entfernen einer ausgewählten Teilnahme aus der Liste der einen oder mehreren Teilnahmen, wenn die ausgewählte Teilnahme annulliert wird.

21. Verfahren nach Anspruch 17, wobei die eine oder die mehreren Teilnahmen eine oder mehrere Teilnahmen an dem Eintritt eines Ereignisses aufweisen, das ein oder mehrere vorbestimmte Kriterien erfüllt.

22. Computerlesbares Medium, umfassend Befehle, die bei Ausführung von einem geeigneten Computer

bewirken, dass der Computer ein Verfahren nach einem der Ansprüche 8 bis 12 ausführt.

Es folgen 5 Blatt Zeichnungen

Anhängende Zeichnungen

FIG. 1

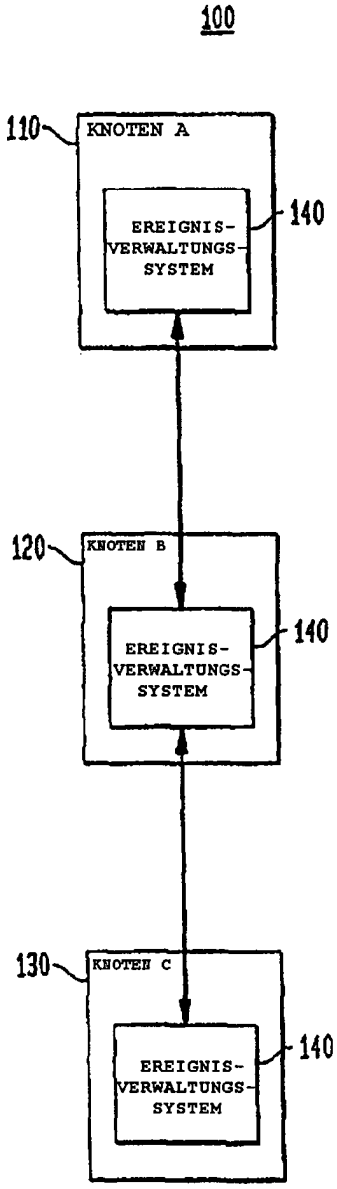


FIG. 2

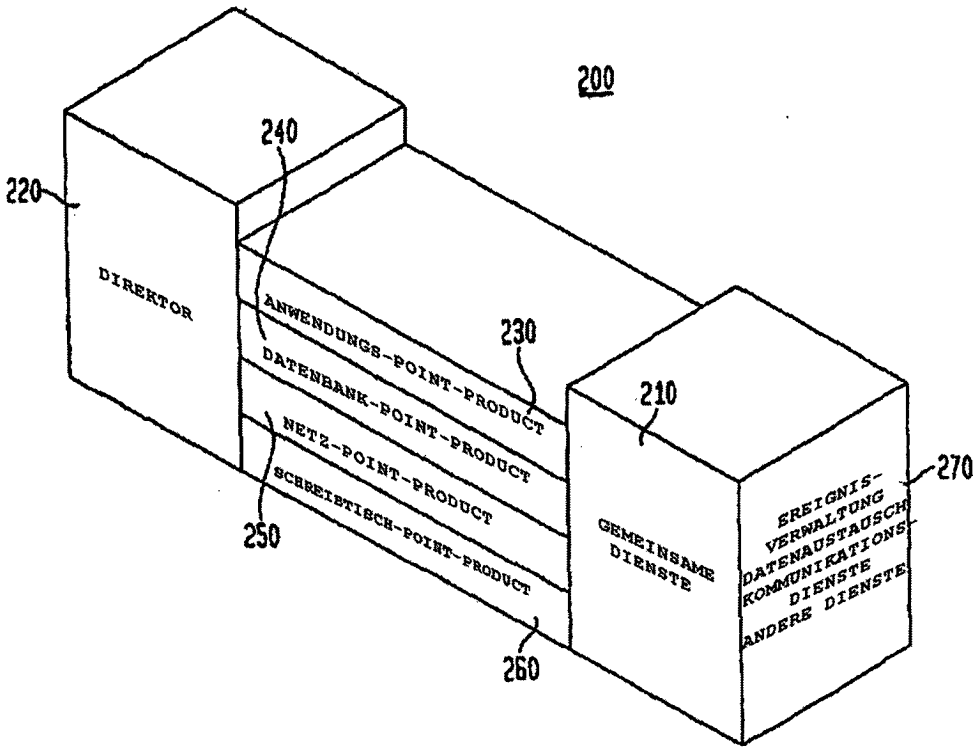

200
240
220
DIREKTOR
ANWENDUNGS-POINT-PRODUCT
DATENBANK-POINT-PRODUCT
NETZ-POINT-PRODUCT
SCHREIBTISCH-POINT-PRODUCT
230
210
250
260
GEMEINSAME DIENSTE
270
EREIGNIS-VERWALTUNG
DATENAUSTAUSCH
KOMMUNIKATIONS-DIENSTE
ANDERE DIENSTE

FIG. 3

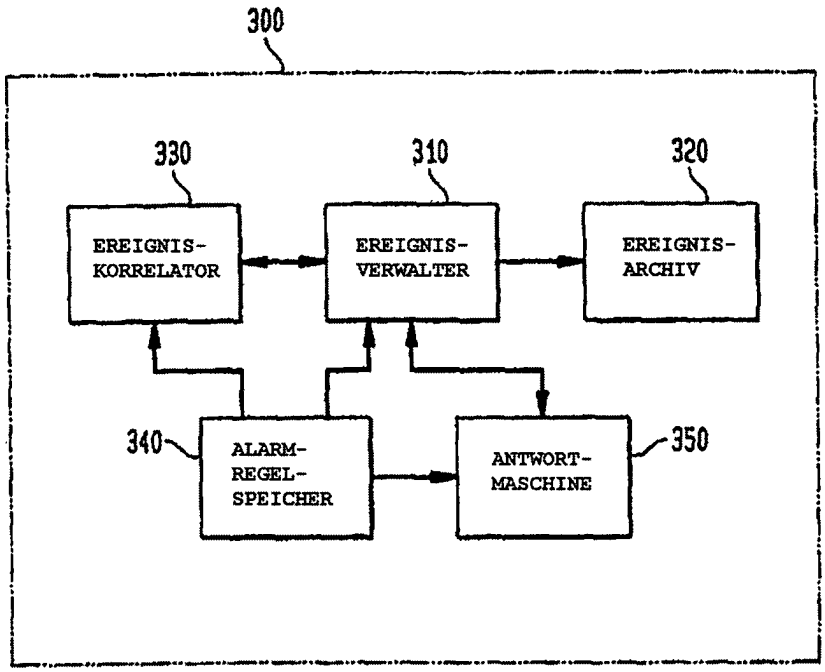

300
330
310
320
EREIGNIS-
KORRELATOR
EREIGNIS-
VERWALTER
EREIGNIS-
ARCHIV
340
ALARM-
REGEL-
SPEICHER
ANTWORT-
MASCHINE
350

FIG. 4

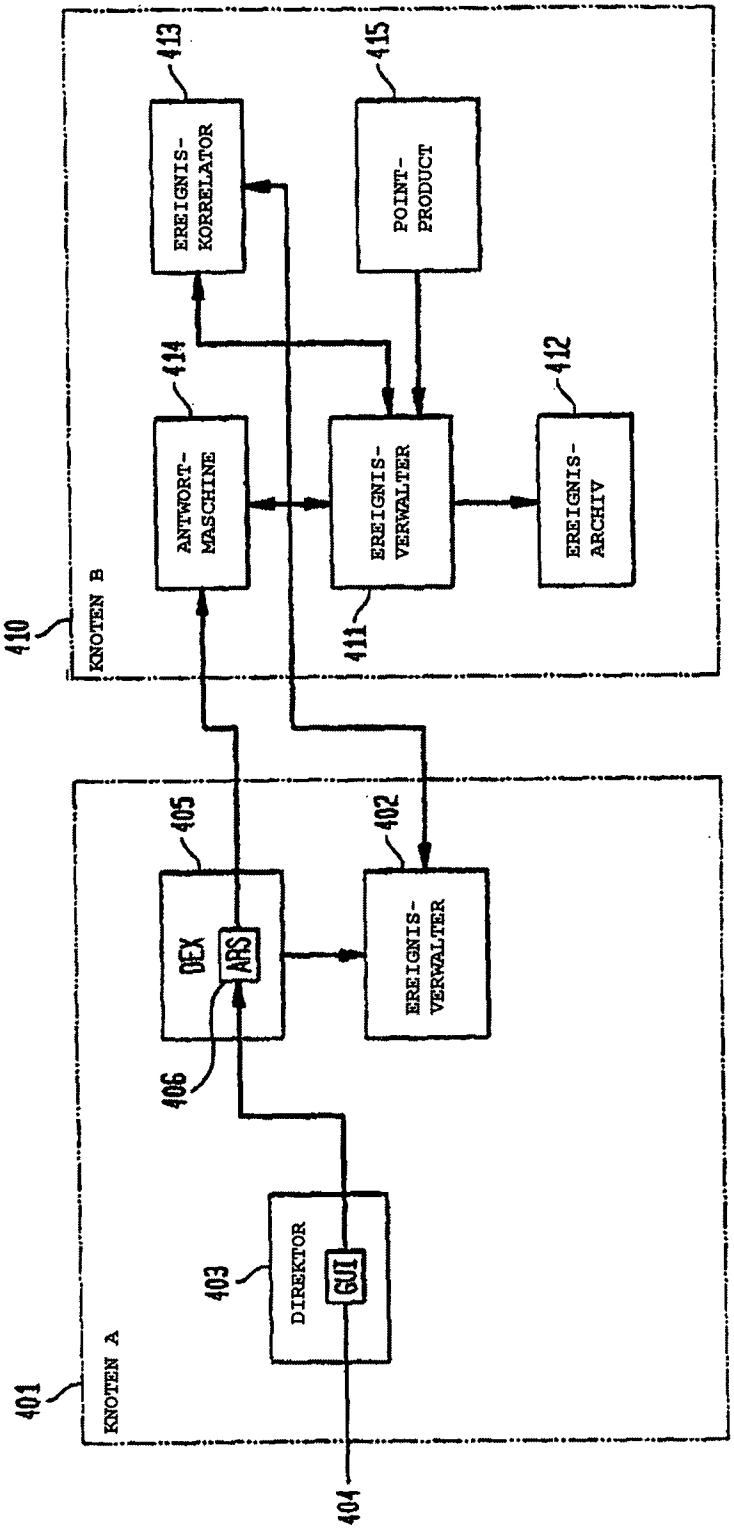
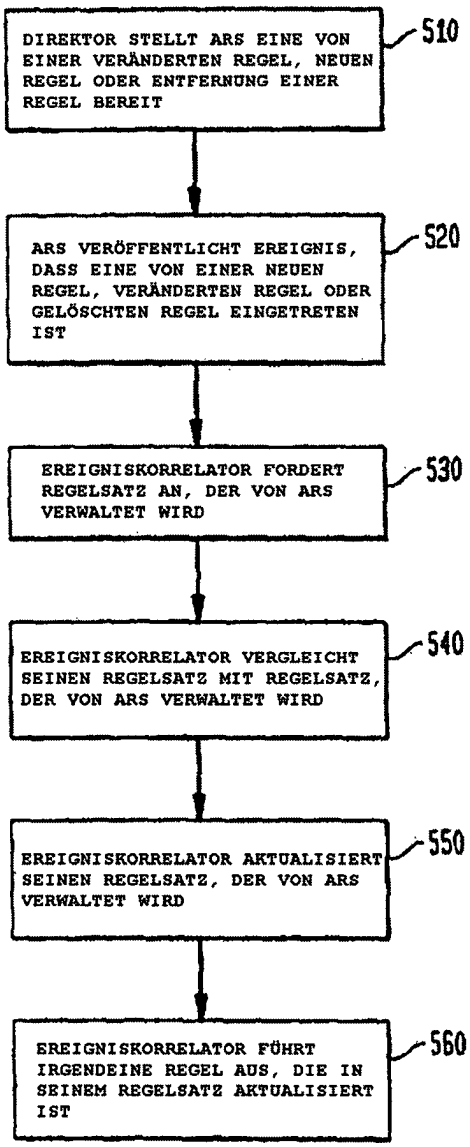

KNOTEN A
401
DIREKTOR
GUI
403
404
DEX
ARS
405
406
EREIGNIS-
VERWALTER
402
KNOTEN B
410
EREIGNIS-
KORRELATOR
413
POINT-
PRODUCT
415
ANTWORT-
MASCHINE
414
EREIGNIS-
VERWALTER
411
EREIGNIS-
ARCHIV
412

FIG. 5


DIREKTOR STELLT ARS EINE VON EINER VERÄNDERTEN REGEL, NEUEN REGEL ODER ENTFERNUNG EINER REGEL BEREIT
510
ARS VERÖFFENTLICHT EREIGNIS, DASS EINE VON EINER NEUEN REGEL, VERÄNDERTEN REGEL ODER GELÖSCHTEN REGEL EINGETRETEN IST
520
EREIGNISKORRELATOR FORDERT REGELSATZ AN, DER VON ARS VERWALTET WIRD
530
EREIGNISKORRELATOR VERGLEICHT SEINEN REGELSATZ MIT REGELSATZ, DER VON ARS VERWALTET WIRD
540
EREIGNISKORRELATOR AKTUALISIERT SEINEN REGELSATZ, DER VON ARS VERWALTET WIRD
550
EREIGNISKORRELATOR FÜHRT IRGENDEINE REGEL AUS, DIE IN SEINEM REGELSATZ AKTUALISIERT IST
560