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

(10) **DE 698 37 461 T2** 2007.11.29

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 903 889 B1**
(21) Deutsches Aktenzeichen: **698 37 461.4**
(96) Europäisches Aktenzeichen: **98 306 134.2**
(96) Europäischer Anmeldetag: **31.07.1998**
(97) Erstveröffentlichung durch das EPA: **24.03.1999**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **04.04.2007**
(47) Veröffentlichungstag im Patentblatt: **29.11.2007**

(51) Int Cl.[8]: ***H04L 12/24*** (2006.01)
G06F 17/00 (2006.01)

(30) Unionspriorität:
932733 **17.09.1997** **US**

(73) Patentinhaber:
Hewlett-Packard Development Company, L.P., Houston, Tex., US

(74) Vertreter:
Schoppe, Zimmermann, Stöckeler & Zinkler, 82049 Pullach

(84) Benannte Vertragsstaaten:
DE, FR, GB

(72) Erfinder:
Singh, Jitendra K., San Jose, California 95120, US

(54) Bezeichnung: **Rechner Systemverwaltung über das Internet**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Eine gegenwärtig implementierte Computersystemverwaltung stützt sich stark auf derartige bekannte Verwaltungsplattformen, wie OpenView von Hewlett-Packard Co. („HP"), NetView von IBM, SunNet Manager von Sun Microsystems und andere. Diese Plattformen werden typischerweise mit Drittteilnehmerwerkzeugen verwendet, die die spezifischen Aufgaben durchführen, die erforderlich sind, um spezielle Geräte zu verwalten, einschließlich Intel-basierten PC-Tischcomputern und Serversystemen, Netzwerkgeräten, wie beispielsweise Netzknoten (Hubs), Brücken und Routern, und eine andere ähnliche Ausrüstung. Beispiele dieser Drittteilnehmerwerkzeuge umfassen NetServer Assistant von HP zum Verwalten der NetServer-Linie von Computern und Interconnect Manager, der zum Verwalten von Netzwerkgeräten verwendet wird, wie beispielsweise Routern. Als eine allgemeine Regel sind diese Werkzeuge komplex, teuer und ohne extensives Training schwierig zu verwenden.

[0002] Das Internet macht es möglich, Anwendungen zu erzeugen, die viele der Funktionen, die nun durch Verwaltungsplattformen und Drittteilnehmer-Nachrüstwerkzeuge durchgeführt werden, auf viele einfachere Weise durchführen. Diese Anwendungen sind durch Neulinge einfacher zu verwenden als bekannte Werkzeuge und senken die Gesamtkosten einer Systemverwaltung.

[0003] Das Dokument „Welcome to the Revolution" von Mary Jander, Data Communications, Bd. 25, Nr. 16, 21. November 1996, liefert eine Übersicht über netzbasierte Verwaltungssysteme.

[0004] Gemäß einem Aspekt der Erfindung ist ein Verfahren gemäß Anspruch 1 vorgesehen. Gemäß einem anderen Aspekt der vorliegenden Erfindung ist ein Verfahren gemäß Anspruch 2 vorgesehen.

[0005] Die Ausführungsbeispiele der vorliegenden Erfindung, die hierin beschrieben sind, benötigen bestimmte allgemeine Computersysteme und Komponenten, um zu funktionieren. Es muss einen Satz von Computersystemen oder Netzwerkvorrichtungen geben, die verwaltet werden müssen. Ein Satz von Client-Systemen wird verwendet, um die Sätze von Computersystemen und/oder Netzwerkvorrichtungen zu verwalten. In einigen Fällen sind das verwaltete System und das Client-System das gleiche System. Am Herstellerort ist ein System positioniert und wird zum Lagern und Analysieren von Daten von den verwalteten Systemen verwendet. Das System des Herstellers wird lediglich zum Implementieren derartiger Verwaltungsmerkmale, wie Analyse und Verifizierung von Systeminformationen, einer Übertragung von beratenden Informationen zurück zu Nutzern und einer Systemregistrierung benötigt.

[0006] Irgendeiner der bekannten Web-Browser, wie beispielsweise Netscape Navigator von Netscape Corp. oder Internet Explorer von Microsoft Corp., müssen auf allen Client-Systemen installiert sein, die als Verwaltungssysteme verwendet werden, und zumindest eines der verwalteten oder der Client-Systeme muss einen HTTP-Server (Web-Server) aufweisen, der auf demselben läuft. Schließlich wird eine Implementierung von einer der bekannten Technologien, die es möglich machen, Konfigurationsinformationen wiederzuerlangen und/oder zu ändern, auf den verwalteten und den Client-Systemen benötigt, einschließlich einer Implementierung von irgendeinem von Simple Network Management Protocols („SNMP"), DMTF/DMI, ISO/CMIP oder anderen proprietären Protokollen.

[0007] Mit diesen erforderlichen Komponenten, von denen alle bekannt sind, kann die Konfiguration des Systems über das Internet unter Verwendung eines HTML-Dokuments betrachtet oder verändert werden, um die verwalteten Systeme aufzulisten und anzuzeigen, zusammen mit Bildsymbolen, die den Zustand der verwalteten Systeme darstellen. Durch ein Verwenden von „aktiven Steuerungen" oder Java-Scripts kann der Zustand der verwalteten Systeme durch ein Verändern der Farbe zugeordneter Bildsymbole oder des angezeigten Texts dynamisch aktualisiert werden. Unter Verwendung eingebetteter Befehle oder Identifizierer innerhalb von Schablonendokumenten kann ein Programm erzeugt werden, um benötigte Systeminformationen automatisch zu gewinnen.

[0008] Bei einem anderen Ausführungsbeispiel wird ein HTML-CGI-Dokument erzeugt, das erwünschte Systeminformationen und eine Referenzverknüpfung zurück zu dem System an dem ausgewählten Ort des Herstellers enthält, wobei ermöglicht wird, dass das System des Herstellers diese Systeminformationen automatisch wiedererlangt. Die Systeminformationen werden dann gegenüber einer Liste von aktuell gültigen Systemkonfigurationen analysiert, um potentielle Probleme zu erfassen. Falls potentielle Probleme erfasst werden, werden die Informationen wiederum automatisch zurück zu dem verwalteten System gesendet.

[0009] Fig. 1 ist ein Blockdiagramm, das die Systemkomponenten und die Architektur der vorliegenden Erfindung zeigt;

[0010] Fig. 2 ist ein Flussdiagramm für ein erstes Ausführungsbeispiel der vorliegenden Erfindung;

[0011] Fig. 3 ist ein Flussdiagramm für ein zweites Ausführungsbeispiel der vorliegenden Erfindung;

[0012] Fig. 4 ist ein Flussdiagramm für ein weiteres Ausführungsbeispiel der vorliegenden Erfindung;

[0013] Fig. 5 ist ein Flussdiagramm für noch ein weiteres Ausführungsbeispiel der vorliegenden Erfindung; und

[0014] Fig. 6, Fig. 7, Fig. 8 und Fig. 9 sind Flussdiagramme für unterschiedliche Implementierungen eines anderen Ausführungsbeispiels der vorliegenden Erfindung.

[0015] Die verschiedenen Ausführungsbeispiele der vorliegenden Erfindung können innerhalb irgendeiner speziellen Realisierung der verallgemeinerten Systemarchitektur wirksam sein, die in Fig. 1 gezeigt ist. Diese verallgemeinerte Architektur weist einen Satz von Computersystemen **10**, einen Satz von Netzwerkvorrichtungen **15** oder irgendeine Kombination von Computersystemen **10** und Netzwerkvorrichtungen **15** auf, von denen alle verwaltet werden müssen. Ein Satz von Client-Systemen **20** wird verwendet, um die Computersysteme **10** und die Netzwerkvorrichtungen **15** zu verwalten. Bei zumindest einem Ausführungsbeispiel ist es möglich, dass das verwaltete Computersystem **10** und das Client-System **20**, das die Verwaltung vornimmt, das gleiche System aufweisen.

[0016] Bei mehreren Ausführungsbeispielen der vorliegenden Erfindung wird ein System **50** eines Herstellers zum Lagern und Analysieren von Daten von den und für die verwalteten Systeme verwendet. Das System **50** des Herstellers ist lediglich zum Implementieren derartiger Merkmale notwendig, wie Analyse und Verifizierung von Informationen verwalteter Systeme, einer Systemregistrierung und einer Übertragung von beratenden Informationen zurück zu den Kunden.

[0017] Ein Web-Browser, wie beispielsweise Navigator von Netscape Corp, oder Internet Explorer von Microsoft Corp., müssen auf allen Client-Systemen **20** installiert sein, die verwendet werden, um andere Systeme zu verwalten. Zumindest eines der verwalteten Systeme **10** oder der Client-Systeme **20** benötigt einen Internet-HTTP-Server, auch als ein Web-Server bekannt, der auf demselben läuft. Abschließend benötigen die verwalteten Systeme **10** und die Client-Systeme **20** eine Implementierung von zumindest einer von mehreren bekannten oder proprietären Technologien, die die Wiedererlangung und Änderung erwünschter Konfigurationsinformationen gestatten. Diese Implementierungen können irgendeines von SNMP, DMTF/DMI oder ISO/CMIP umfassen.

Konfigurationsverwaltung

[0018] Bei einem ersten Ausführungsbeispiel der vorliegenden Erfindung wird eine Konfigurationsverwaltung durch ein Verwenden verschiedener Web-basierter Elemente erzielt. Wie es in Fig. 2 gezeigt ist, wird ein HTML-Dokument **101**, das die verwalteten Systeme und Vorrichtungen auflistet, erzeugt (Fig. 2 Schritt **201**). Das Dokument **101** enthält eine Liste von Systemnamen, die mit Einheitsressourcenlokatoren („URL"s = Uniform Resource Locators) verknüpft sind, die eindeutig zu einem Scriptprogramm **151** einer Gemeinsam-Netzübergang-Schnittstelle („CGI" = Common Gateway Interface) oder einer Microsoft-Internetserver-Anwendungsprogrammierschnittstelle („MS ISAPI" = Microsoft Internet Server Application Programming Interface) zeigen. Wenn dasselbe durch Benutzerhandlungen aufgerufen oder ausgeführt wird (Fig. 2, Schritte **201** und **203**), wobei diese Handlungen ein Anklicken des Systemnamens umfassen, erzeugt das Scriptprogramm **151** ein HTML-Dokument **103**, das die Systeminformationen enthält und das dann angezeigt werden kann (Fig. 2, Schritt **209**). Das Dokument **101** kann durch ein Programm unter Verwendung eines Autoentdeckungsalgorithmus automatisch erzeugt werden, wie beispielsweise OpenView von HP, oder dasselbe kann manuell unter Verwendung eines HTML-Editors erzeugt werden, in welchem Fall die Nutzer die URLs für die verwalteten Systeme kennen müssen.

[0019] Bei dieser Implementierung kann das Scriptprogramm **151** entweder auf dem System/der Vorrichtung, das/die verwaltet wird oder auf einem anderen System positioniert sein, das einen HTTP-Server (Web-Server) aufweist, oder auf beiden Systemen. Dies ermöglicht, dass ein System ein Proxy für ein anderes System oder ein Backup-System ist. Dies ist hilfreich, wenn das spezielle interessierende System ausgefallen ist oder falls das System einen HTTP-Server nicht ausführen kann. Ein Verwenden von Proxys erzeugt eine Redundanz und gestattet ferner ein Verwenden der Verwaltungseinrichtungen, die hierin beschrieben sind, bei Vorrichtungen, die einen HTTP-Server nicht ausführen können oder wollen.

[0020] Ein HTML-Konfigurationsschablonendokument **102** wird unter Verwendung irgendeines bevorzugten HTML-Editors erzeugt (Fig. 2, Schritt **211**). Das Dokument **102** enthält derartige standardmäßige HTML-Elemente, wie Etiketten, Bildsymbole, Text, Bezüge auf Java-Scripts, aktive Objekte und andere Dokumente, falls nötig. An Stellen, an denen die tatsächlichen Parameterwerte angezeigt werden, ist ein Platzhalter eingebettet. Der Platzhalter umfasst ein identifizierendes Anfangsmetaschriftzeichen am Anfang und ein Endmetaschriftzeichen am Ende des Platzhalters, um den Platzhalter für das Scriptprogramm **151** identifizierbar zu machen. Der Körper des Platzhalters enthält Identifikationsinformationen, wie beispielsweise eine SNMP-Objekt-ID, die das Attribut, dessen Wert wiedererlangt und angezeigt werden soll, eindeutig identifiziert.

[0021] Das CGI- oder ISAPI-Scriptprogramm **151** wird durch den HTTP-Server infolge einer Endbenutzeranforderung nach Informationen aufgerufen, die der Nutzer durch ein „Klicken" auf das Bildsymbol oder Symbol einleitet, das mit dem Vorrichtungsnamen in der System-/Vorrichtungsliste des Dokuments **101** etikettiert ist (Fig. 2, Schritt **203**). Wenn dasselbe aufgerufen ist, erlangt das Scriptprogramm **151** Parameter wieder, die unter Verwendung von Standard-CGI/ISAPI-Schnittstellen zu demselben geleitet wurden (Fig. 2, Schritt **209**). Bei diesem Ausführungsbeispiel der vorliegenden Erfindung lauten die Parameter: a) Informationstyp (existierend/neu); b) IP-Adresse des interessierenden Systems; c) SNMP-Gemeinschaftsname; d) Systemkonfigurationsdateiname; und e) Schablonendateiname. Der Informationstypparameter gibt an, ob das Programm eine bestehende Konfigurationsdatei zurückgeben muss oder ob dasselbe eine neue erzeugen muss. Der IP-Adressparameter identifiziert das System, für das Informationen wiedererlangt werden sollen (das verwaltete System). Der SNMP-Gemeinschaftsname (ein SNMP-Artefakt, das zu einer Sicherheit verwendet wird) identifiziert die Gemeinschaft, zu der der SNMP-Agent gehört, der zum Wiedererlangen der angeforderten Informationen verwendet wird. Der Systemkonfigurationsdateiname ist der Name der Datei, zu der neu wiedererlangte Informationen geschrieben werden. Der Schablonendateiname ist der Name der Schablonendatei, die das Programm verwenden wird, um zu bestimmen, welche Informationen wiedererlangt werden sollen.

[0022] Nachdem das Scriptprogramm **151** diese Informationen wiedererlangt, analysiert dasselbe das Schablonendokument syntaktisch, erlangt sequentiell die eingebetteten Objektidentifizierer wieder, führt eine SNMP- oder eine andere Anforderung durch, um den Wert des angeforderten Attributs (Objekts) wiederzuerlangen, wobei der wiedererlangte Wert falls nötig in eine bedeutsame Form umgewandelt wird, und ersetzt den eingebetteten Platzhalter mit diesem Wert (Fig. 2, Schritt **209**). Sobald alle erforderlichen Werte erhalten wurden, schreibt das Scriptprogramm **151** die erzeugte Datei, das Dokument **103**, unter Verwendung des Systemkonfigurationsdateinamens, der von den weitergeleiteten Parametern wiedererlangt wurde, auf eine Platte (Fig. 2, Schritt **213**). Dann leitet dasselbe eine Referenz zurück zu dem HTTP-Server, wobei angegeben wird, dass der Server diese Datei zu dem Nutzer zurückgeben sollte, der die Anforderung einleitet (Fig. 2, Pfeil von Schritt **207** zu Schritt **201**).

Echtzeitsystemkonfigurationsverifizierung

[0023] Bei diesem Ausführungsbeispiel können Systemkonfigurationsinformationen in Echtzeit mit minimalem Kundenaufwand verifiziert werden. Diese Funktion ist unter bestehenden nicht Web-basierten Technologien schwierig zu implementieren und ist typischerweise durch Verkäufer nicht vorgesehen.

[0024] Der Prozess, um Systemkonfigurationen zu verifizieren, beginnt damit, dass der Benutzer das HTML-Dokument **101** lädt und das geeignete Bildsymbol (Icon) oder Etikett anklickt, das das interessierende System darstellt. Dies bewirkt, dass das CGI-Scriptprogramm **151** auf einem der verwalteten Systeme ausgeführt wird. Das Scriptprogramm **151** füllt dann die Felder in einem Schablonenformular **102** aus, wobei das Dokument **103** erzeugt wird (unten beschrieben). Der Prozess, durch den das Scriptprogramm **151** das Schablonenformular **102** ausfüllt, um das Dokument **103** zu erzeugen, ist ähnlich diesem, der bei dem vorhergehenden Ausführungsbeispiel beschrieben ist. Sobald das Dokument **102** durch das Programm **151** ausgefüllt ist, wird das Dokument **102** zu dem Web-Browser des Nutzers als das Dokument **103** zurückgegeben (Fig. 3, Schritt **251**).

[0025] Das Dokument **103** ist ein CGI-Formular, bei dem alle Felder etikettiert sind. Zusätzlich zu einem Umfassen eines Etiketts „Systemname" gibt es beispielsweise einen Feldwertparameter, der dem Wert des Systemnamens zugewiesen ist, z. B. „Mango".

[0026] Das CGI-Formulardokument **103** enthält einen „Absenden"-Button. Wenn ein Benutzer diesen Button anklickt (Fig. 3, Schritt **253**), werden die Inhalte (Name-Wert-Paare) des Dokuments **103** zu dem System übertragen, auf das in dem Formular-URL Bezug genommen wird. In diesem Fall ist das referenzierte System ein System **50** an dem Ort des Herstellers (Fig. 4, Schritt **255**). Auf ein Empfangen des Dokuments **103** hin, führt der HTTP-Server an dem System **50** ein Scriptprogramm **153** aus, das in dem URL definiert ist. Das Programm **153** analysiert das Dokument **103** syntaktisch und sichert die Parameterwerte, die aus demselben wiedererlangt werden, in einer Datenbank **200**.

[0027] Das Scriptprogramm **153** führt dann ein Programm **155** auf dem System **50** mit einem Zeiger zu den Daten aus, die gerade, in die Datenbank **200** eingegeben wurden (Fig. 3, Schritt **257**). Das Programm **155** erlangt diese Daten wieder und vergleicht spezifische Systemkonfigurationsinformationen, wie beispielsweise die Versionsnummern der Softwarekomponenten und eine unterstützte Hardware/Software, mit diesen einer Datenbank **202** einer standardmäßigen/unterstützten Systemkonfiguration, die unabhängig erzeugt wird (Fig. 3, Schritt **259**). Die Ergebnisse dieses Vergleichs lassen Unterschiede zwischen den Konfigurationen der verwalteten Systeme **10** und aktuell gültigen Konfigurationen erkennen. Beispiele dieser Unterschiede könnten Unterschiede bei den Versionen von Software/Firmware/Hardware-Komponenten oder dem Vorhandensein von Hard-

ware/Software-Komponenten sein, die bekanntlich potentielle Probleme aufweisen. Sobald diese Unterschiede bestimmt sind, bereitet das Programm **155** einen Unterschiedsbericht vor, der als ein HTML-Dokument, Dokument **107**, formatiert ist, und leitet denselben zurück zu dem Scriptprogramm **153** (Fig. 3, Schritt **259** zu Schritt **255**). Das Programm **153** gibt dann unter Verwendung eines Standard-HTTP-Protokolls das Dokument **107** zu dem Client-Browser **20** zurück, von dem die Anforderung stammte (Fig. 3, Schritt **255** zu Schritt **251**).

Mail/HTTP-basierte Systemkonfigurationsverifizierung und Kundenratschläge

[0028] Die vorhergehend beschriebenen Verfahren für eine Konfigurationsverwaltung und Echtzeit-Systemkonfigurationsvalidierung nehmen an, dass ein Benutzer einen Web-Browser aufruft und Systeminformationen anfordert. Bei dem folgenden Ausführungsbeispiel werden Konfigurationsinformationen ohne einen Benutzereingriff erzeugt und übertragen, und wobei die Beratung asynchron über das Internet oder andere E-Mail-Mechanismen zurück zu dem Nutzer übertragen wird.

[0029] Dieses Verfahren erfordert ein Programm **157**, das auf einem oder mehreren der verwalteten Systeme **10** läuft (Fig. 4, Schritt **301**). Eine modifizierte Version des Programms **153**, hier Programm **159** genannt, plus eine modifizierte Version des Programms **155**, hier Programm **161** genannt, laufen auf dem System **50** an dem Ort des Verkäufers. Zusätzlich zu diesen Programmen muss ein E-Mail-System auf dem System **50** laufen. Das E-Mail-System muss in der Lage sein, Elektronische-Post-Nachrichten zu anderen Systemen zu senden und von denselben zu empfangen, die mit dem Internet verbunden sind.

[0030] Das Programm **157** führt in vorbestimmten periodischen Intervallen auf den Systemen **10** aus. Dasselbe kann konfiguriert sein, um auf jedem der verwalteten Systeme **10**, auf einem der verwalteten Systeme **10** oder auf einer gewissen Anzahl von Systemen zwischen diesen Extremen zu laufen. In diesen Fällen, in denen das Programm **157** nicht auf allen der Systeme läuft, verhält sich dasselbe als ein Proxy-Agent und ist in der Lage, Informationen von den anderen Systemen wiederzuerlangen, für die dasselbe als ein Proxy konfiguriert ist.

[0031] Wenn das Programm **157** ausführt, erlangt dasselbe Daten von einem oder mehreren der verwalteten Systeme **10** unter Verwendung eines der standardmäßigen oder proprietären Protokolle wieder, wie beispielsweise SNMP oder DMI. Dasselbe erzeugt dann einen Satz von Dateien **109**, eine für jedes System, für die dasselbe konfiguriert ist, die detaillierte Systeminformationen enthält. Die Spezifika der Informationen sind durch das Schablonendokument **102** bestimmt. Zusätzlich zu den Systeminformationen erzeugt das Programm **157** ferner geeignete E-Mail-Kopfblöcke, die Mailing-Listen umfassen, so dass das Dokument **109** ein Format und Felder aufweist, die zu dem E-Mail-System kompatibel sind, so dass das Dokument **109** zu den konfigurierten Bestimmungsorten gesendet werden kann.

[0032] Das Programm **157** versendet die Dateien **109** zu dem E-Mail-Programm durch ein Platzieren derselben in dem ausgehenden Behälter oder alternativ ein direktes Kommunizieren mit dem E-Mail-Server unter Verwendung von Standard-APIs, wie beispielsweise MAPI.

[0033] Das E-Mail-System nimmt die Dateien, die durch das Programm **157** erzeugt werden, und liefert dieselben an den Empfänger, der bei dieser Implementierung das Programm **159** ist, das auf dem System **50** läuft. Das Programm **159** extrahiert die Systeminformationen und lädt dieselben in die Datenbank **200** herunter (Fig. 4, Schritt **303**).

[0034] Das Programm **161** führt in konfigurierten Intervallen auf dem System **50** aus. Dasselbe extrahiert Informationen aus der Datenbank **200** sequenziell, vergleicht diese Informationen mit Standardinformationen gültiger Konfigurationen in der Datenbank **202** und erzeugt einen Satz von Dateien, Dokument **111** genannt, eine für jedes System, für das eine Konfigurationsanalyse durchgeführt wird oder auf dem eine Konfigurationsveralterung erfasst wird (Fig. 4, Schritt **305**). Das Programm **161** fügt geeignete E-Mail-Systemkopfblöcke hinzu, die Bestimmungsadressen konfigurierter Empfänger umfassen, und schreibt diese zu dem ausgehenden Behälter des E-Mail-Systems oder verwendet die APIs des E-Mail-Systems, um dieselben an das E-Mail-System zu versenden (Fig. 4, Schritt **307**). Das E-Mail-System wiederum liefert dieselben an den Empfänger.

Systemregistrierung

[0035] Im Allgemeinen füllen die meisten Kunden keine Systemregistrierungsformulare aus. Vielleicht sieht der Kunde keinen Nutzen in einem Verbringen von Zeit damit, die Formulare auszufüllen. Dieses Ausführungsbeispiel der vorliegenden Erfindung eliminiert etwas des Aufwands, der nötig ist, um diese Formulare auszufüllen. Dasselbe macht ferner die Sammlung von wesentlich mehr Systeminformationen möglich, wobei unter Verwendung der vorhergehend beschriebenen Ausführungsbeispiele möglich gemacht wird, Beratungsinformationen automatisch zurück zu Kunden zu senden, wenn Komponenten veraltet werden oder wenn Probleme entdeckt werden.

[0036] Nachdem der Kunde das neu erworbene System empfangen hat, würde eine Installation durch

ein Ausführen eines Installationsprogramms erzielt. Der letzte Teil des Installationsprogramms ist modifiziert, so dass dasselbe ein Programm **163** ausführt (Fig. 5, Schritt **351**). Das Programm **163** lässt ein elektronisches Formulardokument **113** für den Kunden zum Ausfüllen erscheinen (Fig. 5, Schritt **353**). Der Kunde füllt grundlegende Informationen aus, wie beispielsweise den Namen und die E-Mail-Adresse des Kunden. Nachdem diese Informationen eingegeben sind, verwendet das Programm **163** das Schablonendokument **102**, um Systemkonfigurationsinformationen in einer Weise zu sammeln, die ähnlich dem Programm **151** ist, und erzeugt ein Dokument **117** durch ein Anhängen der grundlegenden Kundeninformationen an Konfigurationsinformationen unter Verwendung des CGI-Formular-„Name-Wert"-Formats (Fig. 5, Schritt **355**).

[0037] Das Dokument **117** ist ein HTML-Formular, das aus Name-Wert-Paaren für Kundeninformationen, die durch den Kunden eingegeben werden, und den Systemkonfigurationsinformationen besteht, die durch das Programm **163** eingegeben werden. Der Formular-URL zeigt zu dem Programm **153** auf dem System **50**.

[0038] Nach dem Erzeugen des Dokuments **117** unternimmt das Programm **163** eine oder beide der folgenden Handlungen: (1) überträgt das Dokument **117** als ein CGI-Scriptformular zu dem System **50** an dem Herstellerort unter Verwendung des HTTP-Protokolls (Fig. 5, Schritt **357**) und (2) fügt E-Mail-Kopfblöcke zu dem Dokument **117** hinzu und platziert dasselbe in dem ausgehenden Behälter des Elektronische-Post-Systems, das irgendwo in dem Netzwerk des Kunden positioniert ist.

[0039] Das Mail-System liefert das Dokument **117** zu Systemen auf einer vorbestimmten Postverteilungsliste. Die genaue unternommene Handlung hängt von Folgendem ab: (1) ob der Kunde eine Echtzeitprüfung seiner Systemkonfiguration wünscht und (2) ob der Kunde wünscht, dass Systeme an mehreren Orten diese Konfigurationsinformationen sichern. Nach dem Übertragen dieser Dokumente wartet das Programm **163** auf eine Antwort an dem gleichen TCP/IP-Port wie ein Web-Browser, typischerweise Port **80**.

[0040] Für den Web-Server auf dem System **50** sieht das Dokument **117**, das über HTTP übertragen wird, identisch zu einer CGI-Formularanfrage aus, die erzeugt worden wäre, falls das Dokument **117** in dem Zusammenhang eines Web-Browsers angezeigt worden wäre und ein Benutzer den Button „Konfiguration validieren" angeklickt hätte. Dies resultiert in den gleichen Handlungen, die unter dem Abschnitt „Echtzeitsystemkonfigurationsverifikation" beschrieben sind (Fig. 5, Schritt **369**). Das Programm **153** erlangt Informationen in dem Dokument **117** wieder, gibt dieselbe in die Datenbank **200** ein und führt das Programm **155** mit einem Zeiger zu den eben in die Datenbank **200** eingegebenen Daten aus (Fig. 5, Schritt **359**). Das Programm **155** erlangt die eben eingegebenen Daten wieder und vergleicht dieselben mit der Datenbank **202** eines standardmäßigen/unterstützten Systems (Fig. 5, Schritt **371**). Die Unterschiede zwischen der Konfiguration des Systems, das aktuell registriert wird, und der Standardkonfiguration werden dann als ein HTML-Dokument **107** formatiert und zurück zu dem Scriptprogramm **153** geleitet. Das Programm **153** gibt dann das Dokument **107** zu dem System zurück, das registriert wird. Das Dokument **107** wird dann durch das Programm **163** empfangen (Fig. 5, Schritt **357**), das auf eine Antwort an dem Port **80** wartete. Das Programm **163** schreibt dann das Dokument **107** zu einer Datei (Fig. 5, Schritt **367**) und führt den Web-Browser aus, wobei ein Befehlszeilenparameter zu demselben zeigt (Fig. 5, Schritt **369**). Dies bewirkt, dass das Dokument **107** durch den Web-Browser auf dem System angezeigt wird, das registriert wird und liefert eine unmittelbare Rückkopplung über irgendwelche potentiellen Konfigurationsprobleme.

Systemwarnungsüberwachung und Ausnahmenhandhabung

[0041] Eine Systemwarnungsbedingung wird erzeugt, wenn ein bestimmter Systemparameter vorbestimmte Grenzbedingungen überschreitet. Falls beispielsweise die Systemtemperatur zu hoch wird, wird eine Warnung ausgelöst. Herkömmliche Verfahren zum Handhaben von Warnungen verwenden Industriestandard- oder proprietäre Protokolle, wie beispielsweise SNMP oder DMTF/DMI, und senden Warnungsinformationspakete zu Empfangssystemverwaltungskonsolen, wie OpenView von HP. An diesen Konsolen verändern Bildsymbole, die die Systeme darstellen, von denen die Warnungen stammen, eine Farbe (beispielsweise von Grün zu Rot). Dies benachrichtigt den Administrator, dass etwas nicht stimmt.

[0042] Bei diesem Ausführungsbeispiel führt das Programm **165** auf verwalteten Systemen **10** aus, die auf Warnungen von den lokalen SNMP- oder DMI-Agenten warten (Fig. 6, Schritte **401** und **403**). Bei einem Empfang einer Warnung decodiert das Programm **165** die Warnungen unter Verwendung der Warnungs-ID, um in ein Warnungsübersetzungsdatendateidokument **119** zu indexieren. Basierend auf der Warnungs-ID gibt die Warnungsdatendatei eine Warnungsaufzeichnung zurück, die aus den folgenden Feldern besteht: (1) Warnungstyp, (2) Warnungsbeschreibung, (3) Systemname, (4) der Name einer Bildsymbol-Bitmap-Datei, die verwendet wird, um diese Warnung in dem Web-Browser anzuzeigen, und (5) eine URL-Referenz zu den Hilfdateien, einem Dokument **120**, das zusätzliche Informationen über

die Warnung liefert. Diese Aufzeichnung wird in eine Verwaltungsinformationsbasis-Tabelle („MIB"-Tabelle; MIB = Management Information Base) einer lokalen Programmunterbrechung (Trap), falls SNMP verwendet wird, oder ein anderes Dokument **121** eingegeben (Fig. 6, Schritt **405**).

[0043] Wie das Programm **165** empfängt das Programm **167** auch SNMP- oder DMI-Warnungen auf dem Verwaltungssystem **20**. Wenn eine Warnung empfangen wird, bestimmt das Programm **167** das System, von dem die Warnung kam, basierend auf den Adressinformationen in dem Warnungspaket (Fig. 6, Schritt **407**). Dasselbe startet dann den lokalen Web-Browser mit Befehlszeilenparametern, die zu dem CGI-Scriptprogramm **151** und dem Systemnamen zeigen, von dem die Warnung empfangen wurde. Das Scriptprogramm **151** rekonstruiert das Dokument **103** unter Verwendung geeigneter Bildsymbole, die in dem Dokument **121** oder der SNMP-Lokal-Programmunterbrechung-MIB spezifiziert sind (Fig. 6, Schritt **409**), verknüpft die Bildsymbole Hilfedateien, die den Bildsymbolen zugeordnet sind, und gibt das Dokument **103** zu dem Browser zurück, wie es unter dem früheren Ausführungsbeispiel einer Konfigurationsverwaltung beschrieben ist (Fig. 6, 413). Der Browser wiederum zeigt das Dokument **103** an. Wenn ein Benutzer das Bildsymbol, das einen System/Untersystem-Status darstellt, anklickt, zeigt der Browser das Hilfsdokument an, das mit dem Bildsymbol verknüpft wurde.

[0044] Bei einer alternativen Implementierung dieses Ausführungsbeispiels (siehe Fig. 7) wird das Dokument **102** durch ein Verändern der Kopfblockinformationen modifiziert, um ein Dokument **123** zu erzeugen. Wenn das Programm **151** eine Anforderung von dem HTTP-Server empfängt, verwendet dasselbe das modifizierte Dokument **123**, um ein Dokument **125** zu erzeugen, das mit dem Dokument **103** identisch ist, außer den Kopfblockinformationen (Fig. 7, Schritt **451**). Die Kopfblockinformationen geben dem Web-Browser an, dass das Dokument **125** periodisch aktualisiert werden muss. Jedes Mal, wenn der Browser eine Aktualisierung anfordert (Fig. 7, Schritt **453**), wird das Programm **151** durch den HTTP-Server ausgeführt (Fig. 7, Schritt **455**), und dasselbe erzeugt das Dokument **125** mit dem neuesten Systemstatus und den neuesten Konfigurationsinformationen neu und gibt dasselbe zu dem Web-Browser zurück. Dies stellt sicher, dass alle Warnungen, die seit der letzten Aktualisierung erzeugt wurden, in dem Systemstatusabschnitt des neu erzeugten Dokuments **125** widergespiegelt sind.

[0045] Bei einer dritten alternativen Implementierung dieses Ausführungsbeispiels (siehe Fig. 8) führen das Programm **151**, eine modifizierte Version des Programms **165**, hier ein Programm **169** genannt, und der HTTP-Webserver alle auf verwalteten Systemen **10** aus. Das Programm **169** ist dem Programm **165** ähnlich, mit der Ausnahme, dass dasselbe das Programm **151** auf ein Empfangen einer Warnung ausführt, zusätzlich zu den normalen Funktionen desselben. Bei dieser Implementierung ist das Dokument **102** durch ein Dokument **127** ersetzt. Das Dokument **127** ist mit dem Dokument **102** identisch, außer dass die Kopfblockinformationen modifiziert sind, um anzugeben, dass dasselbe in „Mehr-Teil-Mit-Ersetzung"-Dokument („multi-part-with-replace") ist, ein bekannter Dokumententyp. Wenn Web-Browser derartige Dokumente empfangen, stellen dieselben asynchron eine Verbindung mit dem Web-Browser her. Wenn ein Nutzer eine Anforderung bei dieser Implementierung einleitet, verwendet das Programm **151** das Dokument **127**, um ein Dokument **129** zu erzeugen (Fig. 8, Schritt **501**), was das Dokument **103** mit einem modifizierten Kopfblock ist. Wenn die Kopfblockinformationen aus der Schablone direkt zu dem Ausgangsdokument kopiert werden, ist das Dokument **129** ebenfalls ein „Mehr-Teil-Mit-Ersetzung"-Dokument. Wenn eine Warnung durch das Programm **169** empfangen wird (Fig. 8, Schritte **503** und **505**), startet dasselbe das Programm **151** mit einer geeigneten Befehlszeile, die angibt, dass dasselbe das Dokument **129** wieder erzeugen und dasselbe zu dem Web-Browser wieder übertragen sollte (Fig. 8, Schritt **507**). Die hier verwendete Technik, um Dokumente bei Warnungsbedingungen asynchron zu dem Web-Browser zu senden, wird häufig als „Server-Push" bezeichnet, d. h. Dokumente werden von dem HTTP-Server geschoben, ohne durch den Web-Browser explizit angefordert zu werden.

[0046] Bei einer vierten Implementierung dieses speziellen Ausführungsbeispiels (siehe Fig. 9) ist das Dokument **102** modifiziert, um Bezüge auf „aktive Objekte" zu umfassen, wie beispielsweise ein Java-Rpplet oder ein ActiveX-Programm, um ein Dokument **131** zu erzeugen. Bei dieser Implementierung verwendet das Programm **151** das Dokument **131** als eine Eingabe und erzeugt ein Dokument **133**, das dem Dokument **103** ähnlich ist, aber die Bezüge auf die aktiven Objekte in dem Dokument **131** umfasst. Eine Eigenschaft derartiger aktiver Objekte besteht darin, dass ein Anzeigen derselben bewirkt, dass der Browser den „Bytecode" ausführt, der den aktiven Objekten entspricht. Die ausführenden eingebetteten Programme (die aktiven Objekte) lesen wiederum das Systemstatusdateidokument **121** in programmierten Intervallen und verändern die angezeigten Bildsymbole, um dem aktuellen Systemzustand zu entsprechen, auf die gleiche Weise, wie es für die anderen Implementierungen dieses Ausführungsbeispiel beschrieben ist.

Patentansprüche

1. Ein Verfahren zur Echtzeitkonfigurationsvalidierung in einer Rechenumgebung, die zumindest

ein verwaltetes System (**10**; **15**) und zumindest ein System (**20**) mit einer Konfigurationsdatenbank aufweist, die das verwaltete System verwaltet, wobei sowohl das System mit einer Konfigurationsdatenbank als auch verwaltete Systeme zum Kommunizieren miteinander über das Internet in der Lage sind, wobei das Verfahren folgende Schritte aufweist:
Erzeugen einer Mehrzahl von Systemkonfigurations- und Statusdokumenten (**103**), eines für jedes verwaltete System, unter Verwendung eines Programms, um die Dokumente aus einem Schablonenformular (**102**) zu erzeugen;
Auswählen (**253**) eines der Systemkonfigurations- und Statusdokumente;
Erzeugen einer Datenbank einer standardmäßig unterstützten Systemkonfiguration an zumindest einem ersten System mit einer Konfigurationsdatenbank (**50**);
Übertragen (**255**) des ausgewählten Systemkonfigurations- und Statusdokuments zu dem zumindest ersten System mit einer Konfigurationsdatenbank (**50**);
Vergleichen (**259**) des ausgewählten Systemkonfigurations- und Statusdokuments mit einer standardmäßig unterstützten Systemkonfiguration; und
Zurückgeben irgendwelcher Unterschiede zwischen dem ausgewählten Systemkonfigurations- und Statusdokument und der standardmäßig unterstützten Konfiguration.

2. Ein Verfahren zum Verifizieren einer Systemkonfiguration und Liefern von Systemratschlägen in einer Computerumgebung, die aus zumindest einem verwalteten System (**10**; **15**) und zumindest einem System (**20**) mit einer Konfigurationsdatenbank gebildet ist, die das verwaltete System verwaltet, wobei sowohl das System mit einer Konfigurationsdatenbank als auch das verwaltete System zum Kommunizieren miteinander über das Internet in der Lage sind, wobei das Verfahren folgende Schritte aufweist:
Ausführen (**301**) eines Programms (**157**) auf jedem der verwalteten Systeme in einem vorbestimmten Intervall und Erzeugen eines ersten Dokuments, das aus vorausgewählten Systeminformationen für jedes verwaltete System gebildet ist;
Senden des ersten Dokuments von jedem verwalteten System durch elektronische Post zu einem System mit einer Konfigurationsdatenbank (**50**);
Erzeugen einer Datenbank von Informationen aus den ersten Dokumenten auf dem System mit einer Konfigurationsdatenbank (**50**); und
Ausführen (**305**) eines zweiten Skriptprogramms (**161**) auf dem System mit einer Konfigurationsdatenbank (**50**) in einem vorbestimmten Intervall, das die Datenbank von Informationen aus den ersten Dokumenten mit einer zweiten Datenbank von gültigen Konfigurationsinformationen auf dem System mit einer Konfigurationsdatenbank vergleicht;
Erzeugen einer Mehrzahl von zweiten Dokumenten (**111**), die jegliche Unterschiede zwischen der Datenbank von Informationen aus den ersten Dokumenten und den gültigen Konfigurationsinformationen angeben; und
Zurückgeben (**307**) des zweiten Dokuments, das den Unterschieden zwischen dem ersten Dokument, das für dieses verwaltete System erzeugt wurde, und der zweiten Datenbank entspricht, durch elektronische Post zu jedem verwalteten System.

3. Ein Verfahren gemäß Anspruch 1 oder 2, das ferner folgende Schritte aufweist:
Ausführen (**351**) eines ersten Programms (**163**) automatisch auf eine anfängliche Verwendung eines verwalteten Systems hin, wobei das Programm einem ersten Benutzer des verwalteten Systems ein erstes Formular (**113**) präsentiert (**353**), um Benutzerregistrierungsinformationen zu erhalten;
Anhängen (**355**) von Systemkonfigurationsinformationen aus dem verwalteten System an das erste Formular, wobei die Systemkonfigurationsinformationen durch ein zweites Programm erhalten werden, das auf dem verwalteten System läuft;
Übertragen des ersten Formulars (**117**) zu einem System mit einer Konfigurationsdatenbank (**50**);
Erzeugen einer ersten Datenbank in dem System mit einer Konfigurationsdatenbank mit ersten Formularen, die von allen verwalteten Systemen erhalten werden, die durch das System mit einer Konfigurationsdatenbank verwaltet werden.

4. Das Verfahren gemäß Anspruch 3, bei dem die Schritte von Anspruch 1 oder 2 durchgeführt werden, nachdem jedes verwaltete System automatisch registriert wurde.

5. Ein Verfahren gemäß einem der vorhergehenden Ansprüche, das ferner folgende Schritte aufweist:
Erfassen (**403**, **503**) von Ausnahmen bei jedem der verwalteten Systeme mittels eines ersten Programms (**165**, **169**);
Erzeugen (**403**, **501**) eines ersten Dokuments bei dem verwalteten System mit Informationen hinsichtlich der erfassten Ausnahmen;
Übertragen des ersten Dokuments zu einem System mit einer Konfigurationsdatenbank über das Internet; und
Anzeigen des ersten Dokuments einer Bedienperson des Systems mit einer Konfigurationsdatenbank, um die Bedienperson vor Ausnahmen bei jedem der verwalteten Systeme zu warnen.

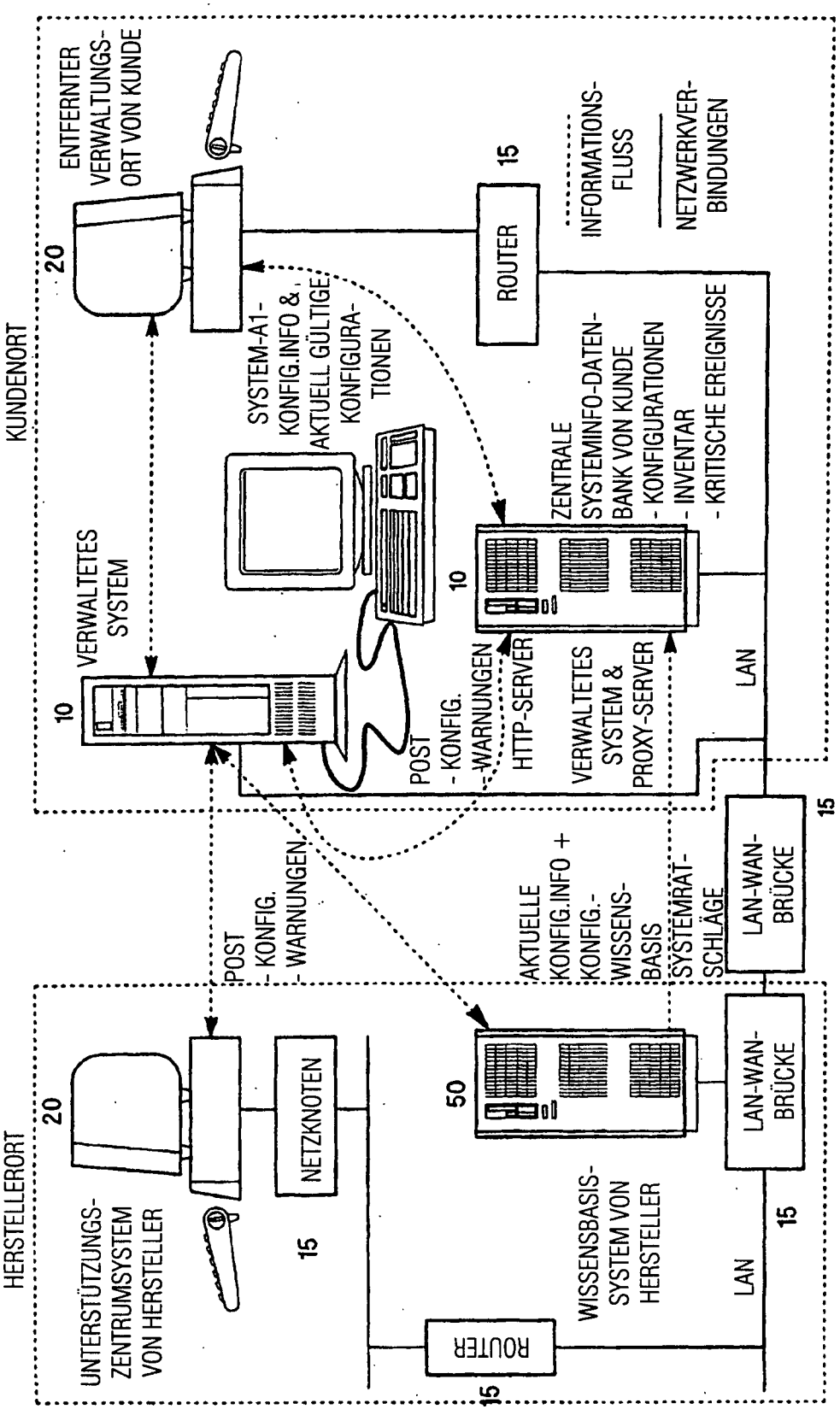
6. Ein Computerprogramm, das eine Computerprogrammcodeeinrichtung aufweist, die, wenn dieselbe an einem Computer ausgeführt wird, den Computer zum Durchführen aller Schritte von irgendeinem der vorhergehenden Ansprüche anpasst.

7. Ein Computerprogramm gemäß Anspruch 6, das auf einem computerlesbaren Medium ausgeführt

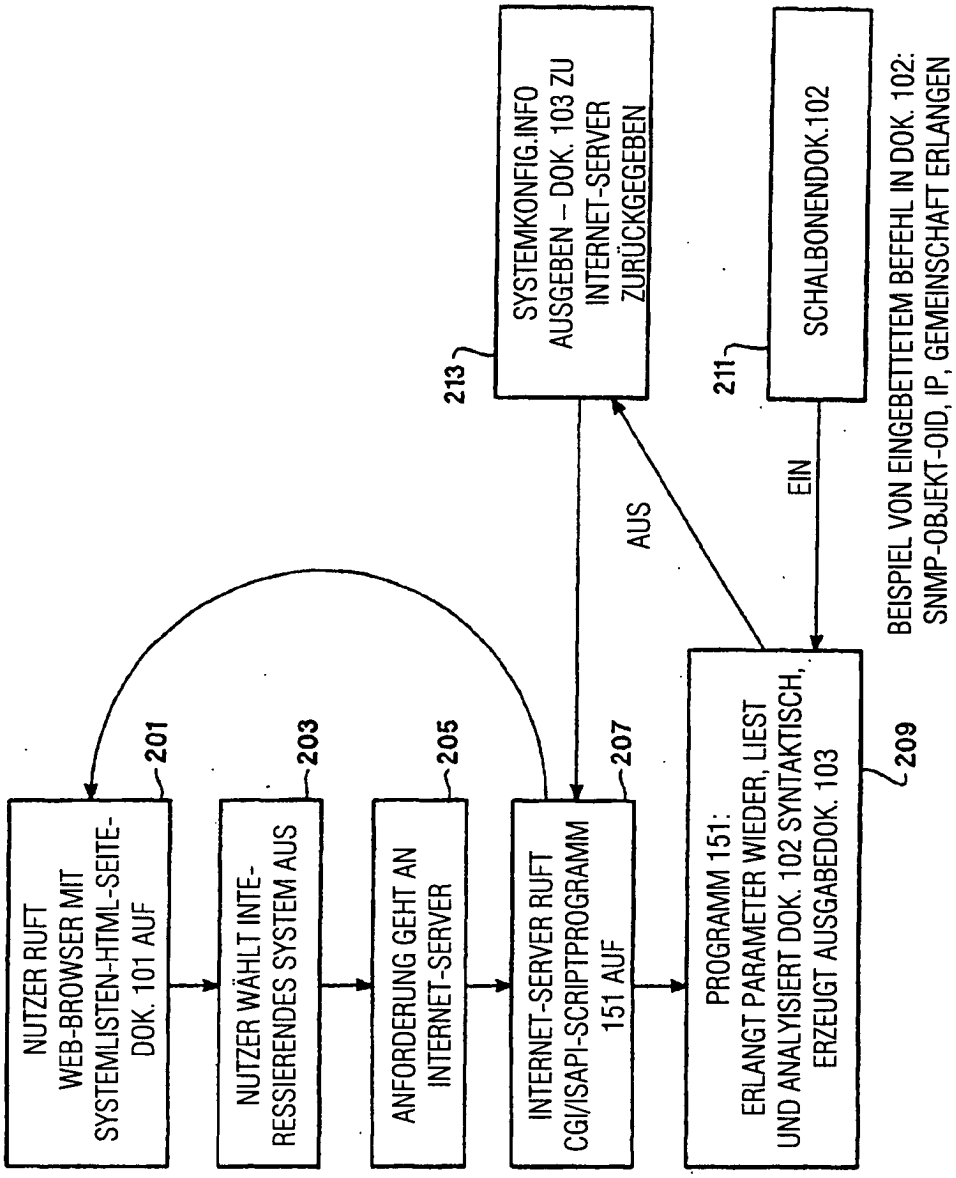
ist.

Es folgen 9 Blatt Zeichnungen

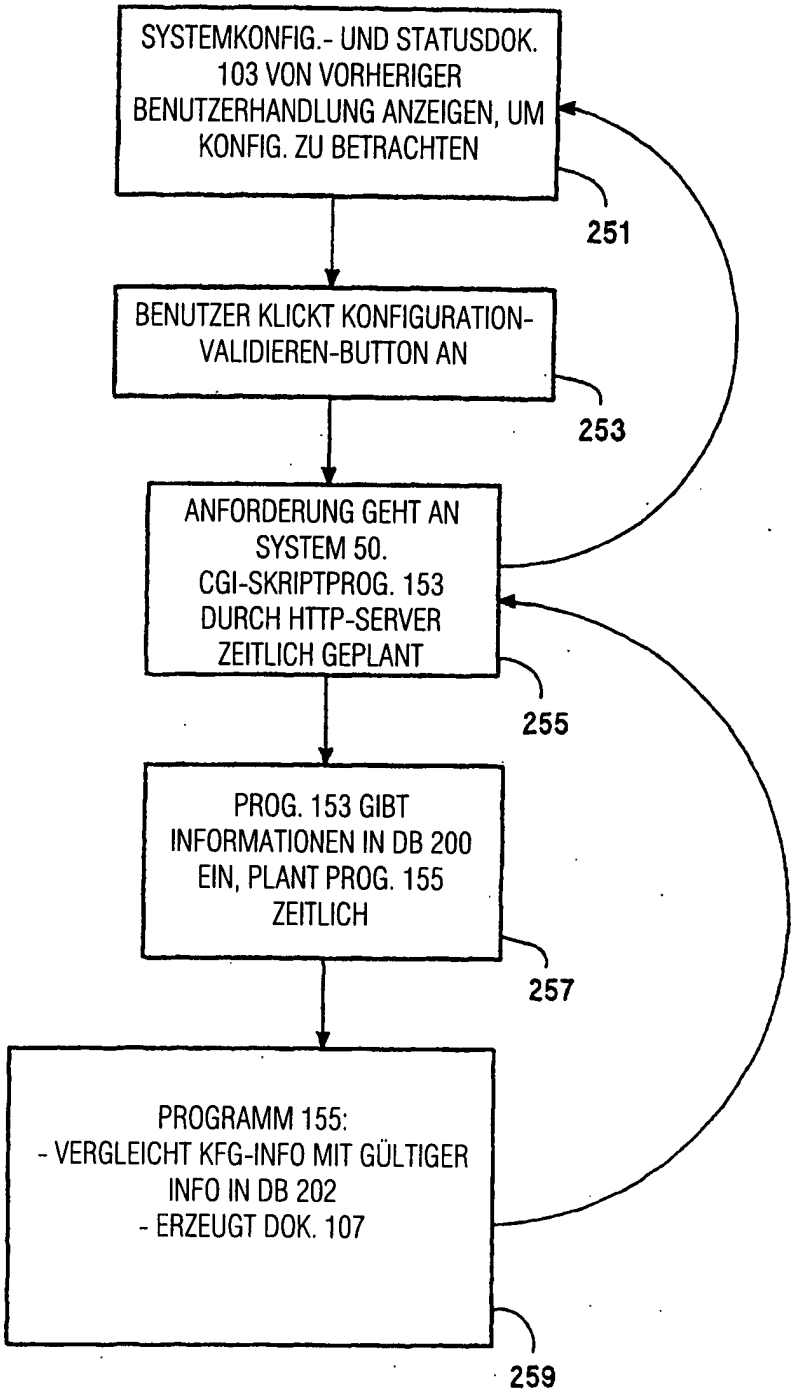
Anhängende Zeichnungen



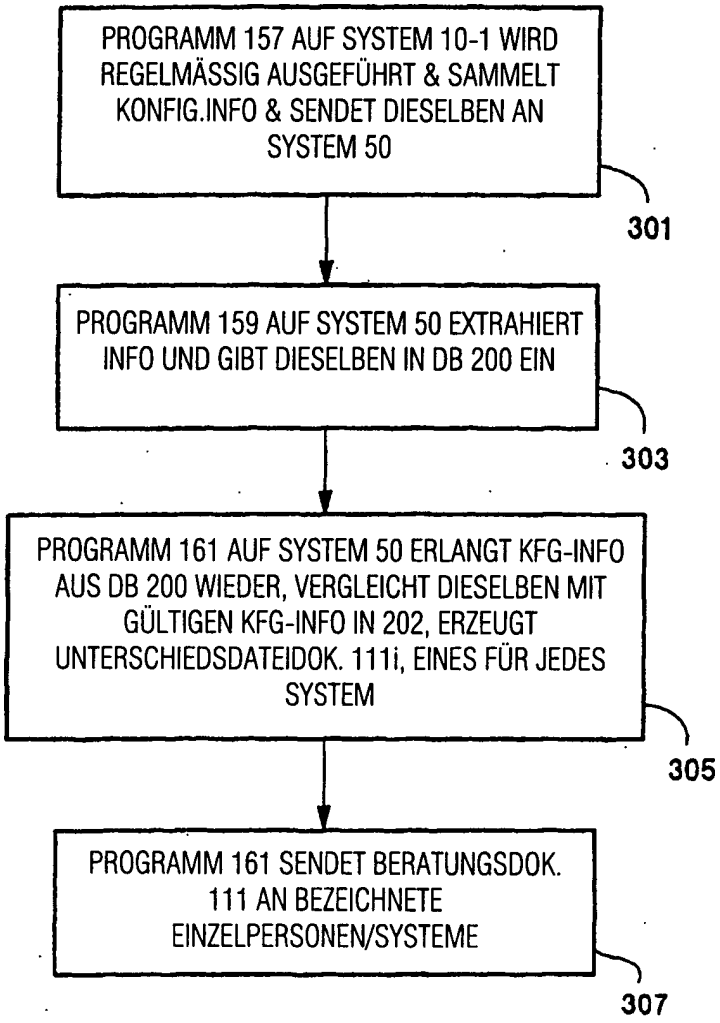
FIGUR 1


NUTZER RUFT WEB-BROWSER MIT SYSTEMLISTEN-HTML-SEITE-DOK. 101 AUF
201
NUTZER WÄHLT INTE-RESSIERENDES SYSTEM AUS
203
ANFORDERUNG GEHT AN INTERNET-SERVER
205
INTERNET-SERVER RUFT CGI/ISAPI-SCRIPTPROGRAMM 151 AUF
207
PROGRAMM 151: ERLANGT PARAMETER WIEDER, LIEST UND ANALYSIERT DOK. 102 SYNTAKTISCH, ERZEUGT AUSGABEDOK. 103
209
BEISPIEL VON PARAMETERN, DIE DURCH PROGRAMM 151 WIEDERERLANGT WERDEN: IP, GEMEINSCHAFTSNAME, SCHABLONENNAME, AUSGABENAME
SYSTEMKONFIG.INFO AUSGEBEN – DOK. 103 ZU INTERNET-SERVER ZURÜCKGEGEBEN
213
AUS
SCHALBONENDOK.102
211
EIN
BEISPIEL VON EINGEBETTETEM BEFEHL IN DOK. 102: SNMP-OBJEKT-OID, IP, GEMEINSCHAFT ERLANGEN

FIGUR 2

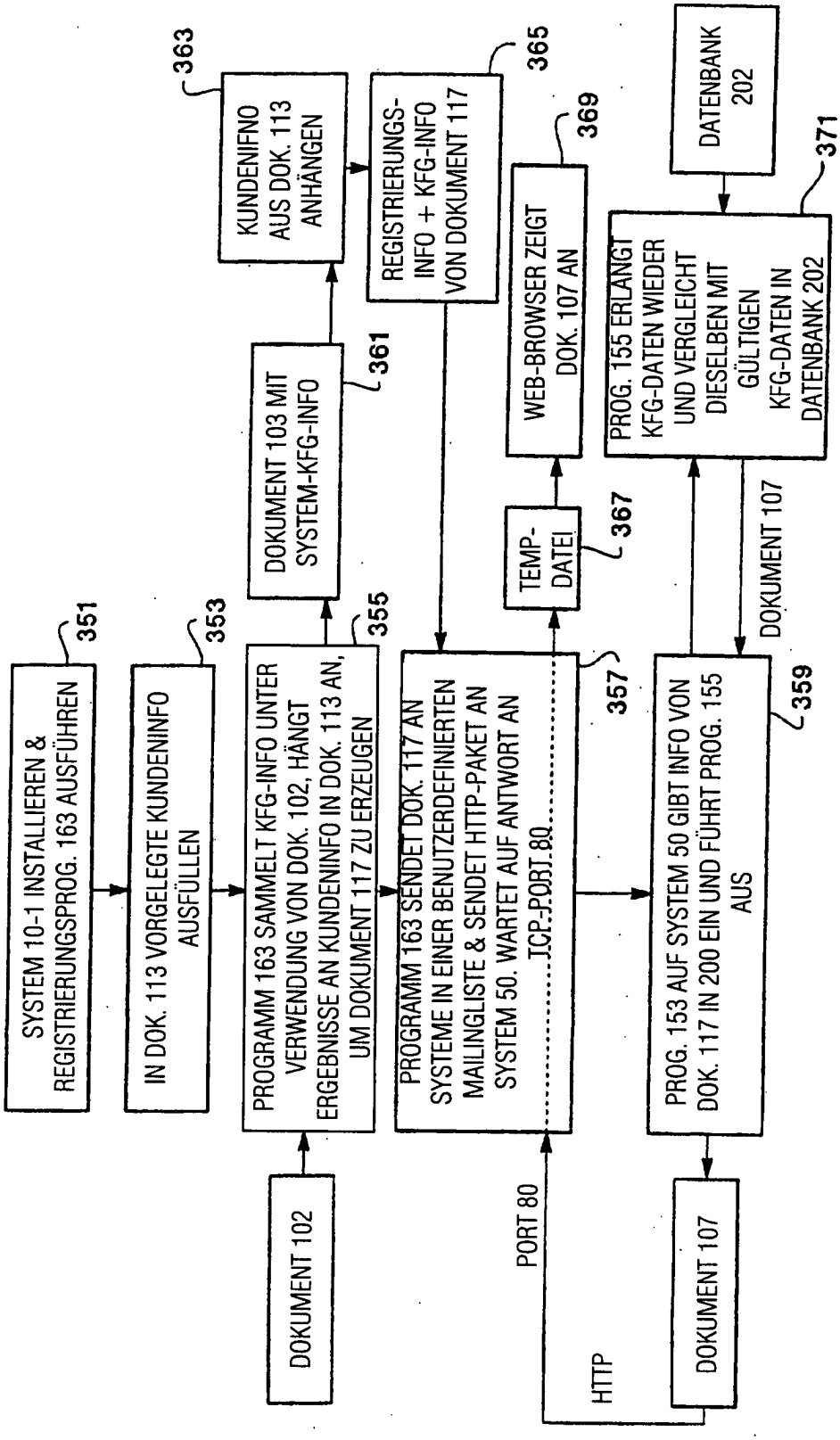

SYSTEMKONFIG.- UND STATUSDOK. 103 VON VORHERIGER BENUTZERHANDLUNG ANZEIGEN, UM KONFIG. ZU BETRACHTEN
251
BENUTZER KLICKT KONFIGURATION-VALIDIEREN-BUTTON AN
253
ANFORDERUNG GEHT AN SYSTEM 50. CGI-SKRIPTPROG. 153 DURCH HTTP-SERVER ZEITLICH GEPLANT
255
PROG. 153 GIBT INFORMATIONEN IN DB 200 EIN, PLANT PROG. 155 ZEITLICH
257
PROGRAMM 155:
- VERGLEICHT KFG-INFO MIT GÜLTIGER INFO IN DB 202
- ERZEUGT DOK. 107
259

FIGUR 3

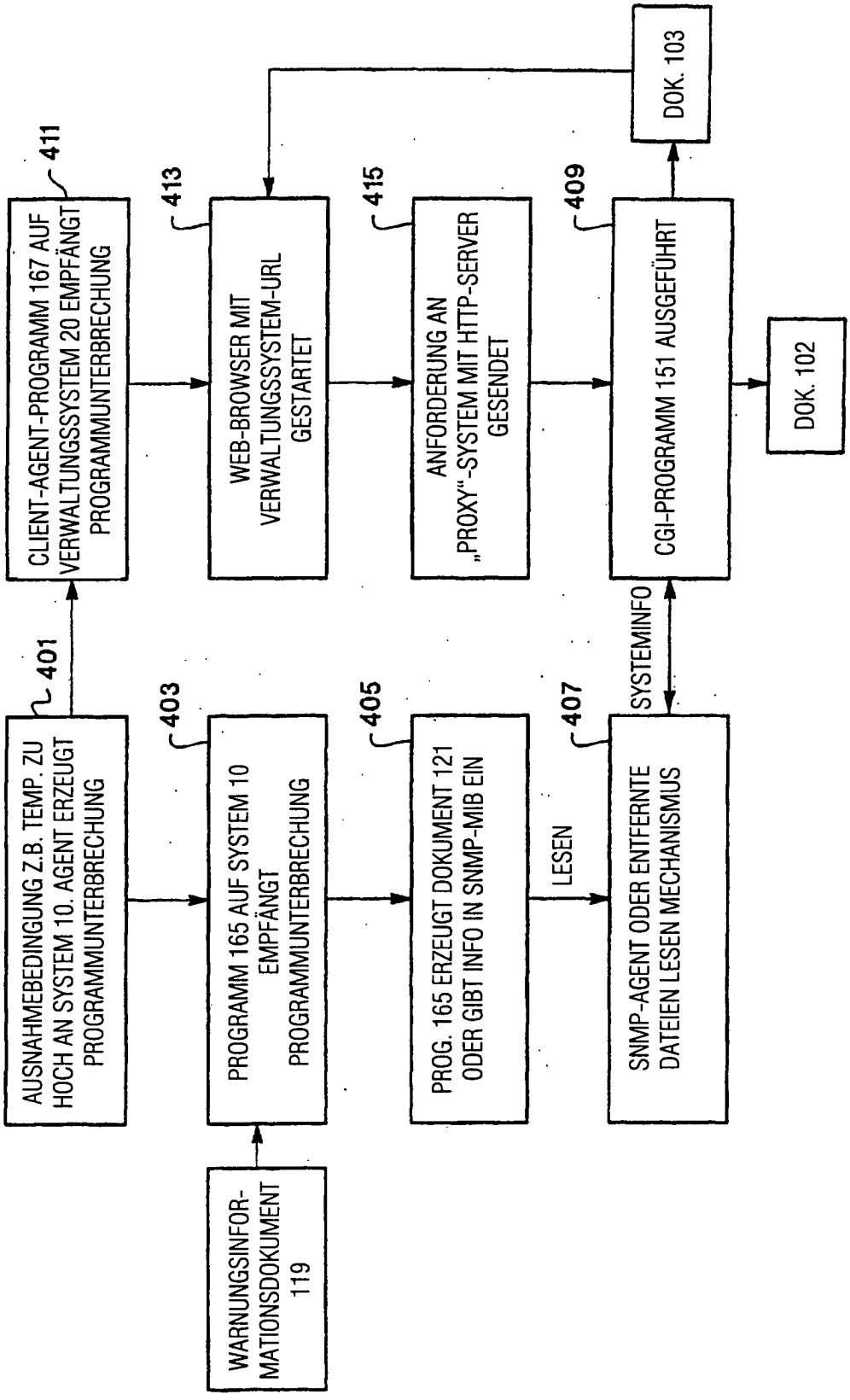

PROGRAMM 157 AUF SYSTEM 10-1 WIRD REGELMÄSSIG AUSGEFÜHRT & SAMMELT KONFIG.INFO & SENDET DIESELBEN AN SYSTEM 50
301
PROGRAMM 159 AUF SYSTEM 50 EXTRAHIERT INFO UND GIBT DIESELBEN IN DB 200 EIN
303
PROGRAMM 161 AUF SYSTEM 50 ERLANGT KFG-INFO AUS DB 200 WIEDER, VERGLEICHT DIESELBEN MIT GÜLTIGEN KFG-INFO IN 202, ERZEUGT UNTERSCHIEDSDATEIDOK. 111i, EINES FÜR JEDES SYSTEM
305
PROGRAMM 161 SENDET BERATUNGSDOK. 111 AN BEZEICHNETE EINZELPERSONEN/SYSTEME
307

FIGUR 4

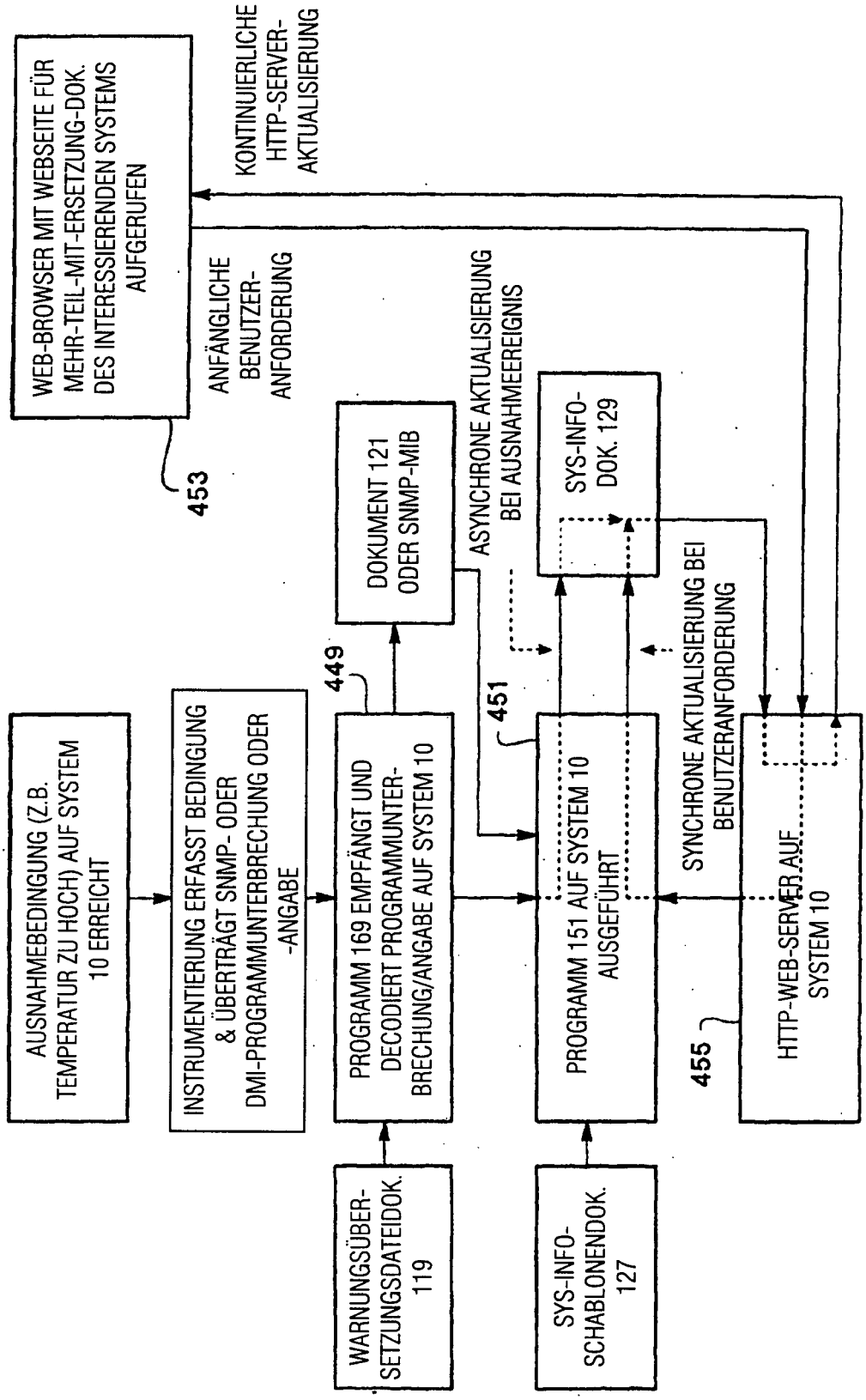
FIGUR 5


SYSTEM 10-1 INSTALLIEREN & REGISTRIERUNGSPROG. 163 AUSFÜHREN
351
IN DOK. 113 VORGELEGTE KUNDENINFO AUSFÜLLEN
353
PROGRAMM 163 SAMMELT KFG-INFO UNTER VERWENDUNG VON DOK. 102, HÄNGT ERGEBNISSE AN KUNDENINFO IN DOK. 113 AN, UM DOKUMENT 117 ZU ERZEUGEN
355
DOKUMENT 102
DOKUMENT 103 MIT SYSTEM-KFG-INFO
361
KUNDENINFO AUS DOK. 113 ANHÄNGEN
363
REGISTRIERUNGS-INFO + KFG-INFO VON DOKUMENT 117
365
PROGRAMM 163 SENDET DOK. 117 AN SYSTEME IN EINER BENUTZERDEFINIERTEN MAILINGLISTE & SENDET HTTP-PAKET AN SYSTEM 50. WARTET AUF ANTWORT AN ICP-PORT 80
357
TEMP-DATEI
367
WEB-BROWSER ZEIGT DOK. 107 AN
369
PROG. 153 AUF SYSTEM 50 GIBT INFO VON DOK. 117 IN 200 EIN UND FÜHRT PROG. 155 AUS
359
PROG. 155 ERLANGT KFG-DATEN WIEDER UND VERGLEICHT DIESELBEN MIT GÜLTIGEN KFG-DATEN IN DATENBANK 202
371
DATENBANK 202
DOKUMENT 107
PORT 80
HTTP
DOKUMENT 107

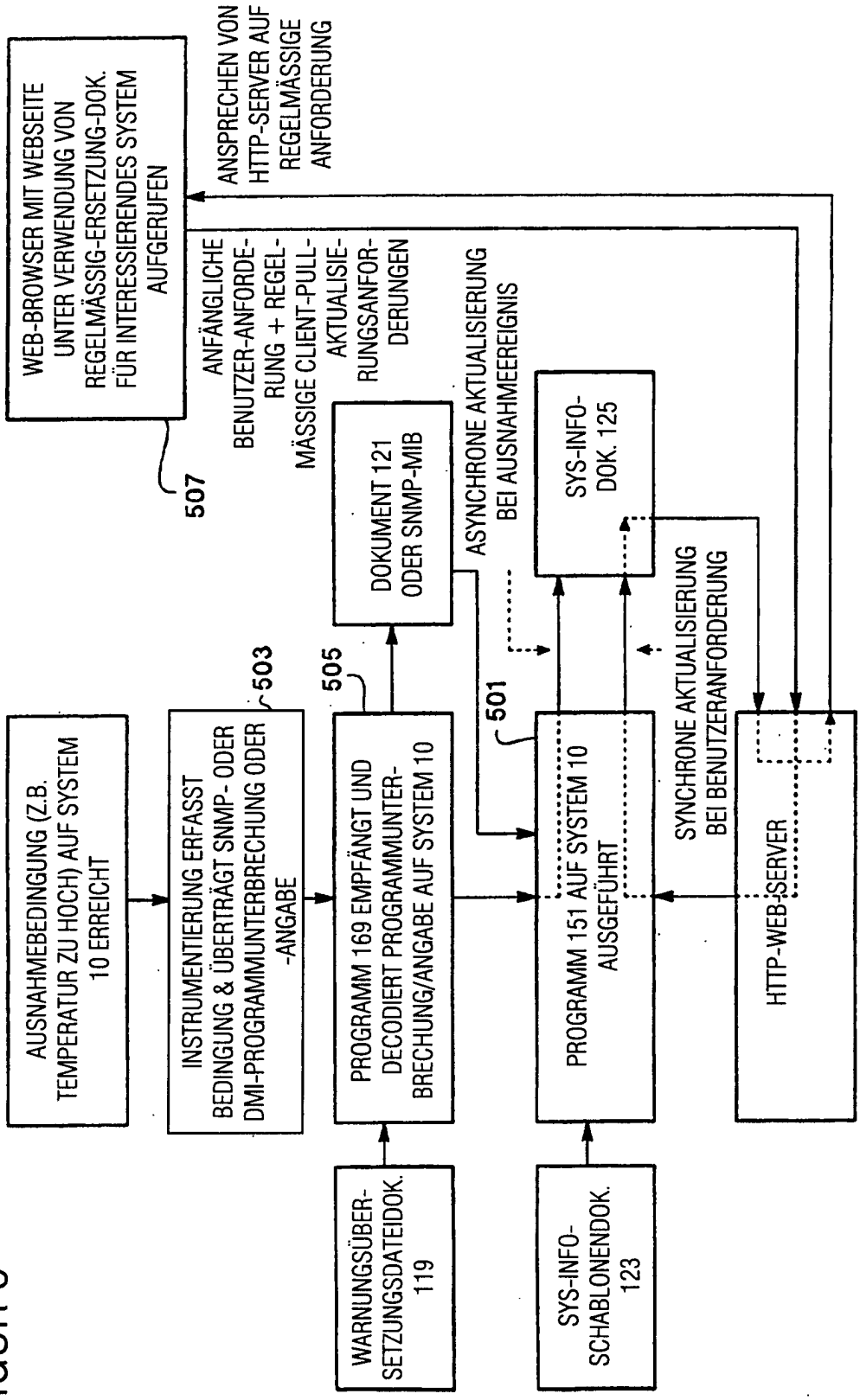
FIGUR 6


401
AUSNAHMEBEDINGUNG Z.B. TEMP. ZU HOCH AN SYSTEM 10. AGENT ERZEUGT PROGRAMMUNTERBRECHUNG
403
PROGRAMM 165 AUF SYSTEM 10 EMPFÄNGT PROGRAMMUNTERBRECHUNG
405
PROG. 165 ERZEUGT DOKUMENT 121 ODER GIBT INFO IN SNMP-MIB EIN
LESEN
407
SNMP-AGENT ODER ENTFERNTE DATEIEN LESEN MECHANISMUS
SYSTEMINFO
WARNUNGSINFOR-MATIONSDOKUMENT 119
411
CLIENT-AGENT-PROGRAMM 167 AUF VERWALTUNGSSYSTEM 20 EMPFÄNGT PROGRAMMUNTERBRECHUNG
413
WEB-BROWSER MIT VERWALTUNGSSYSTEM-URL GESTARTET
415
ANFORDERUNG AN „PROXY"-SYSTEM MIT HTTP-SERVER GESENDET
409
CGI-PROGRAMM 151 AUSGEFÜHRT
DOK. 103
DOK. 102

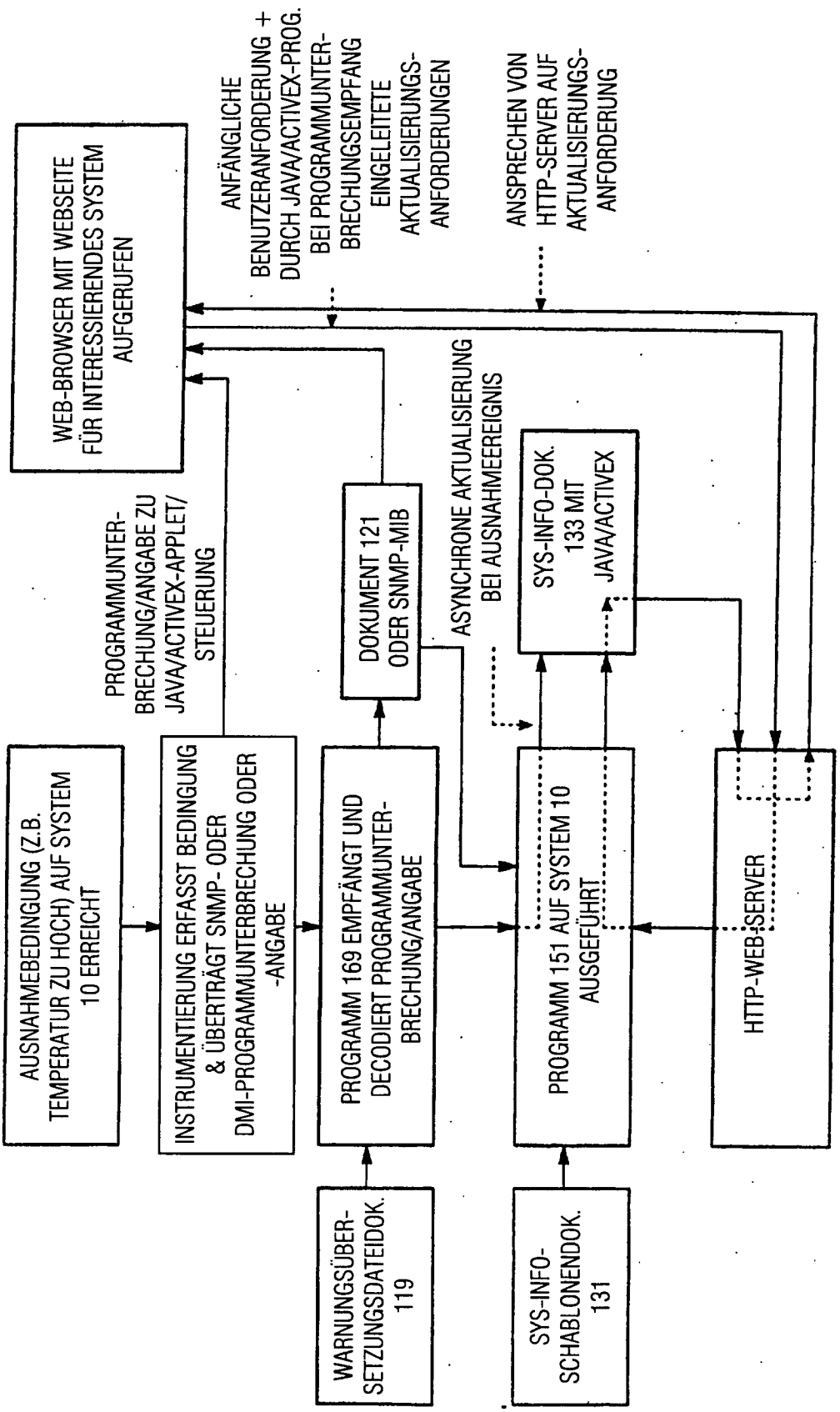
FIGUR 7


WEB-BROWSER MIT WEBSEITE FÜR MEHR-TEIL-MIT-ERSETZUNG-DOK. DES INTERESSIERENDEN SYSTEMS AUFGERUFEN
KONTINUIERLICHE HTTP-SERVER-AKTUALISIERUNG
ANFÄNGLICHE BENUTZER-ANFORDERUNG
453
DOKUMENT 121 ODER SNMP-MIB
ASYNCHRONE AKTUALISIERUNG BEI AUSNAHMEEREIGNIS
SYS-INFO-DOK. 129
449
451
AUSNAHMEBEDINGUNG (Z.B. TEMPERATUR ZU HOCH) AUF SYSTEM 10 ERREICHT
INSTRUMENTIERUNG ERFASST BEDINGUNG & ÜBERTRÄGT SNMP- ODER DMI-PROGRAMMUNTERBRECHUNG ODER -ANGABE
PROGRAMM 169 EMPFÄNGT UND DECODIERT PROGRAMMUNTER-BRECHUNG/ANGABE AUF SYSTEM 10
PROGRAMM 151 AUF SYSTEM 10 AUSGEFÜHRT
SYNCHRONE AKTUALISIERUNG BEI BENUTZERANFORDERUNG
HTTP-WEB-SERVER AUF SYSTEM 10
455
WARNUNGSÜBER-SETZUNGSDATEIDOK. 119
SYS-INFO-SCHABLONENDOK. 127

FIGUR 8


WEB-BROWSER MIT WEBSEITE UNTER VERWENDUNG VON REGELMÄSSIG-ERSETZUNG-DOK. FÜR INTERESSIERENDES SYSTEM AUFGERUFEN
507
ANSPRECHEN VON HTTP-SERVER AUF REGELMÄSSIGE ANFORDERUNG
ANFÄNGLICHE BENUTZER-ANFORDERUNG + REGELMÄSSIGE CLIENT-PULL-AKTUALISIERUNGSANFORDERUNGEN
DOKUMENT 121 ODER SNMP-MIB
ASYNCHRONE AKTUALISIERUNG BEI AUSNAHMEEREIGNIS
SYS-INFO-DOK. 125
AUSNAHMEBEDINGUNG (Z.B. TEMPERATUR ZU HOCH) AUF SYSTEM 10 ERREICHT
INSTRUMENTIERUNG ERFASST BEDINGUNG & ÜBERTRÄGT SNMP- ODER DMI-PROGRAMMUNTERBRECHUNG ODER -ANGABE
503
505
501
PROGRAMM 169 EMPFÄNGT UND DECODIERT PROGRAMMUNTERBRECHUNG/ANGABE AUF SYSTEM 10
PROGRAMM 151 AUF SYSTEM 10 AUSGEFÜHRT
SYNCHRONE AKTUALISIERUNG BEI BENUTZERANFORDERUNG
HTTP-WEB-SERVER
WARNUNGSÜBERSETZUNGSDATEIDOK. 119
SYS-INFO-SCHABLONENDOK. 123

FIGUR 9


WEB-BROWSER MIT WEBSEITE FÜR INTERESSIERENDES SYSTEM AUFGERUFEN
ANFÄNGLICHE BENUTZERANFORDERUNG + DURCH JAVA/ACTIVEX-PROG. BEI PROGRAMMUNTER-BRECHUNGSEMPFANG EINGELEITETE AKTUALISIERUNGS-ANFORDERUNGEN
ANSPRECHEN VON HTTP-SERVER AUF AKTUALISIERUNGS-ANFORDERUNG
PROGRAMMUNTER-BRECHUNG/ANGABE ZU JAVA/ACTIVEX-APPLET/ STEUERUNG
DOKUMENT 121 ODER SNMP-MIB
ASYNCHRONE AKTUALISIERUNG BEI AUSNAHMEEREIGNIS
SYS-INFO-DOK. 133 MIT JAVA/ACTIVEX
AUSNAHMEBEDINGUNG (Z.B. TEMPERATUR ZU HOCH) AUF SYSTEM 10 ERREICHT
INSTRUMENTIERUNG ERFASST BEDINGUNG & ÜBERTRÄGT SNMP- ODER DMI-PROGRAMMUNTERBRECHUNG ODER -ANGABE
PROGRAMM 169 EMPFÄNGT UND DECODIERT PROGRAMMUNTER-BRECHUNG/ANGABE
PROGRAMM 151 AUF SYSTEM 10 AUSGEFÜHRT
HTTP-WEB-SERVER
WARNUNGSÜBER-SETZUNGSDATEIDOK. 119
SYS-INFO-SCHABLONENDOK. 131