



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 08 589 T2** 2004.04.29

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 076 852 B1**

(51) Int Cl.⁷: **G06F 9/46**

(21) Deutsches Aktenzeichen: **699 08 589.6**

(86) PCT-Aktenzeichen: **PCT/GB99/01329**

(96) Europäisches Aktenzeichen: **99 919 404.6**

(87) PCT-Veröffentlichungs-Nr.: **WO 99/059062**

(86) PCT-Anmeldetag: **29.04.1999**

(87) Veröffentlichungstag
der PCT-Anmeldung: **18.11.1999**

(97) Erstveröffentlichung durch das EPA: **21.02.2001**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **04.06.2003**

(47) Veröffentlichungstag im Patentblatt: **29.04.2004**

(30) Unionspriorität:

9810279	13.05.1998	GB
9823714	29.10.1998	GB

(84) Benannte Vertragsstaaten:

BE, CH, DE, ES, FR, GB, IT, LI, NL

(73) Patentinhaber:

British Telecommunications p.l.c., London, GB

(72) Erfinder:

**GHANEA-HERCOCK, Alan, Robert, Ipswich IP3
8JJ, GB; COLLIS, Clements, Jaron, Nr. Ipswich,
Suffolk IP5 2TX, GB**

(74) Vertreter:

Beetz & Partner, 80538 München

(54) Bezeichnung: **Ferndatenverarbeitung**

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] Diese Erfindung bezieht sich auf die Ferndatenverarbeitung und insbesondere, aber nicht ausschließlich, auf die Ferndatenverarbeitung unter Verwendung sogenannter "Agenten".

[0002] Es ist bekannt, Software in Form "mobiler Agenten" zu schaffen. Derartige Agenten umfassen ein Programm, das seine Ausführung anhalten kann, seinen Code in eine serielle Form, die für die Übertragung über ein Übertragungsmedium, wie z. B. ein Telekommunikationsnetz, geeignet ist, umsetzen kann und bewirken kann, daß der Code zu einem weiteren Computer übertragen und anschließend dort ausgeführt wird. Folglich besitzen derartige Agenten das Aussehen einzelner mobiler Entitäten, die von Computer zu Computer geleitet werden können. Der Begriff "Agent" wird in den verschiedenen Kontexten in der Technik mit verschiedenen Bedeutungen verwendet. Wie der Begriff "mobiler Agent" hierin verwendet wird, bezieht er sich auf ein Programm, das sich beim Empfang eines Befehls selbst zu einem weiteren Computer bewegen und die Ausführung dort ohne einen Verlust der Kontinuität beginnen kann.

[0003] Derartige mobile Agenten sind in US 5603031 (White u. a.) beschrieben, das sich auf die von General Magic Inc., Sunnyvale, California, vermarktete Sprache "Teleskript" bezieht. Die Computer-Sprache "JAVA" besteht aus Code, um eine "virtuelle Maschine" auf irgendeinem Computer zu erzeugen, die über das Internet empfangenen Code ausführen kann. Folglich schafft JAVA einige der Elemente, die für die Datenverarbeitung mit mobilen Agenten notwendig sind. Neulich haben die Ankunft von JAVA 1.1 und seine weitverbreitete Verwendung in der Technik zur Entwicklung einer Anzahl von Sprachen, um Agentenprogramme zu schreiben, und von Support-Plattformen, um die Agenten zu unterstützen, in "Beta"-Form oder in Form einer Vorversion geführt, die von der Sprache JAVA und der Laufzeit-Umgebung ("virtuelle Maschine") Gebrauch machen.

[0004] Eines von in diesen ist "VOYAGER", das von OBJECTSPACE Inc., 14901 Quorum Drive, Dallas, Texas, TX 75240, USA, verfügbar ist. VOYAGER ist beschaffen, um Java-Code in eine Form umzusetzen, in der er mobile Agenten-Programme bereitstellen kann, die sich selbst zu einem weiteren spezifizierten Computer bewegen und dort die Operation wieder beginnen können, wobei es Verbesserungen enthält, wie z. B. die Fähigkeit, derartige Programme automatisch zu beenden. Voyager ist in einer Beta-2-Version von ObjectSpace Inc. verfügbar.

[0005] Andere enthalten die von IBM Inc. gelieferten Aglets-Systeme, Odyssey von General Magic Inc. und Concordia von Mitsubishi; und andere, einschließlich der Sprache Agent TCL.

[0006] Im allgemeinen bestehen derartige mobile Agentensysteme aus einzelnen Programmen für spezielle Aufgaben.

[0007] In "Agent design patterns: elements of agent application design", Aridor und Lange, präsentiert auf der Second International Conference on Mobile Agents (Agents '98) in Minneapolis/ St. Pauls, 10–13. Mai 1998, ist eine Erörterung der Konstruktion heterogener Agenten angegeben, die die Erfinder der vorliegenden Erfindung empfangen haben. Der Artikel beschreibt mobile Agenten mit Reiserouten, die eine Folge zu besuchender aufeinanderfolgender Computer spezifizieren. Die Agenten können sich in einer Master/Slave-Beziehung befinden. In diesem Fall bleibt ein "Master"-Agent (der statisch ist) an einem ersten Standort, während ein "Slave"-Agent einen weiteren Standort besucht und zum Master-Agenten zurück meldet. Der Slave-Agent kann unter Verwendung eines "Boten"-Agenten melden, um die Daten zum Master-Agenten zu befördern.

[0008] Darin ist außerdem die Möglichkeit der Bereitstellung einer "organisierten Gruppe" von Agenten kurz erwähnt, die sich alle zusammen bewegen.

[0009] Die vorliegende Erfindung schafft verschiedene Verbesserungen an vorhandenen Agenten-Vorschlägen, wobei sie unter Verwendung einer der obenerwähnten verfügbaren Agenten-Sprachen und Support-Plattformen implementiert sein kann.

[0010] In einem Aspekt schafft die vorliegende Erfindung ein Ferndatenverarbeitungsverfahren, das das Liefern mehrerer paralleler Verarbeitungsaufgaben-Programme von einem ersten Computer zu wenigstens einem zweiten Computer; das Liefern eines Koordinationsprogramms vom ersten Computer zum zweiten Computer; und das Koordinieren der Operation der Aufgaben-Programme durch das Koordinationsprogramm umfaßt.

[0011] Die Verwendung eines heterogenen Agenten-Teams auf den zweiten (entfernten) Computern vereinfacht die notwendigen Steuer- und Kommunikationsaufgaben, weil das Koordinationsprogramm am entfernten Standort die lokalen Operationen der Aufgabenprogramme dort managen kann und als ein Sammel- und Verteilungspunkt für die Kommunikation für die Aufgabenprogramme dort wirken kann, wobei folglich die Kommunikation mit dem ersten Computer (Ausgangs-Computer) verringert wird.

[0012] Andere Aspekte, Ausführungsformen und Merkmale der Erfindung werden aus der folgenden Beschreibung und den Ansprüchen offensichtlich.

[0013] Die Erfindung wird nun lediglich beispielhaft unter Bezugnahme auf die beigefügte Zeichnung veranschaulicht, worin:

[0014] **Fig. 1** ein Blockschaltplan ist, der die Elemente eines Netzes aus entfernten Computern zeigt;

[0015] **Fig. 2** ein Blockschaltplan ist, der die Elemente von einem der entfernten Computer nach **Fig. 1** zeigt;

[0016] **Fig. 3** ein Blockschaltplan ist, der die Programme zeigt, die im Speicher des Computers nach **Fig. 2** vorhanden sind;

[0017] **Fig. 4** ein Blockschaltplan ist, der die Pro-

gramme zeigt, die im Netz nach **Fig. 1** in einer ersten Ausführungsform der Erfindung vorhanden sind;

[0018] **Fig. 5** ein Ablaufplan ist, der den Gesamtbetrieb eines Agentensteuerprogramms zeigt, das einen Teil der **Fig. 4** bildet;

[0019] **Fig. 6a** ein Ablaufplan ist, der den anfänglichen Teil der Operation nach **Fig. 5** zeigt;

[0020] **Fig. 6b** ein Ablaufplan ist, der die Stufe der Plattformerzeugung nach **Fig. 5** zeigt;

[0021] **Fig. 6c** ein Ablaufplan ist, der die Stufe der Überwachung der entfernten Computer nach **Fig. 5** zeigt;

[0022] **Fig. 6d** ein Ablaufplan ist, der die Stufe der Überwachung der Agentenprogramme nach **Fig. 5** zeigt;

[0023] **Fig. 6e** ein Ablaufplan ist, der die Beendigungsstufen nach **Fig. 5** zeigt;

[0024] **Fig. 7** ein Ablaufplan ist, der die Operation eines Scout-Agenten nach **Fig. 4** zeigt;

[0025] **Fig. 8** (die die **Fig. 8a** bis **8e** umfaßt) ein Ablaufplan ist, der die Operation eines Koordinationsagenten nach **Fig. 4** zeigt;

[0026] **Fig. 9** ein Ablaufplan ist, der die Operation eines Aufgabenagenten zeigt, der einen Teil der **Fig. 4** bildet;

[0027] **Fig. 10** eine Darstellung ist, die eine gemäß der ersten Ausführungsform erzeugte erste Anzeige zeigt; und

[0028] **Fig. 11** eine Darstellung ist, die eine gemäß der ersten Ausführungsform erzeugte zweite Anzeige zeigt.

DIE ERSTE AUSFÜHRUNGSFORM

[0029] In **Fig. 1** umfaßt ein Netz aus Computern einen Anwender-Computer **10**, der über ein Telekommunikationsnetz **20** mit mehreren entfernten Computern **30a**, **30b**, **30c** verbunden ist. Der Anwender-Computer **10** kann z. B. einen Sun-Arbeitsplatzrechner umfassen, während die Ziel-Computer **30a**, **30b**, **30c** einen Netz-Server **30a**, einen Arbeitsplatzrechner **30b** und einen Laptop-Personal-Computer **30c** umfassen können.

[0030] Jeder der entfernten Computer **30** umfaßt eine Zentraleinheit (CPU), einen Speicher **32**, ein Plattenlaufwerk **33**, eine Bildanzeigeeinheit (VDU) **34**, eine Tastatur **35** und einen Kommunikationsanschluß **36**, der an den Kommunikationskanal **37** (wie z. B. eine ISDN-Verbindung) gekoppelt ist. Diese Elemente sind über eine (nicht gezeigte) herkömmliche Busstruktur miteinander verbunden.

[0031] In **Fig. 3** sind innerhalb des Speichers **32** jedes der entfernten Computer **30** mehrere Steuerprogramme für die Ausführung gespeichert. In dieser Ausführungsform umfassen diese ein Betriebssystem **321**, wie z. B. Windows **95** oder Windows NT, einen Kommunikationsprotokollstapel **322**, wie z. B. TCP/IP, eine virtuelle Java-Maschine **324** und eine mobile Agentenplattform **326**, wie z. B. VOYAGER, die von ObjectSpace verfügbar ist, die gemeinsam

eine Betriebsumgebung für Anwendungsprogramme umfassen, die in **Fig. 3** gemeinsam als **328** bezeichnet sind. Die Struktur des Anwender-Computers **10** entspricht im allgemeinen der des entfernten Computers **30**, die unter Bezugnahme auf die **Fig. 2** und **3** beschrieben ist, mit Ausnahme, soweit wie sie modifiziert ist, wie im folgenden beschrieben ist.

[0032] Im Gebrauch, wenn der serialisierte Code, der das mobile Agentenprogramm darstellt, durch den Kommunikationsanschluß **37** empfangen wird, veranlassen die mobile Agentenplattform **326** und die virtuelle Java-Maschine **324**, daß er im Anwendungsbereich **328** des Speichers **32** gespeichert wird. Die mobile Agentenplattform **326** setzt dann den Code in lokalen Java-Code um, der auf der virtuellen Java-Maschine **324** ausgeführt wird, wobei vom lokalen Betriebssystem **321** Gebrauch gemacht wird. Die allgemeinen Einzelheiten der Operation der mobilen Agentenplattform **326** und der virtuellen Java-Maschine **324** sind im Voyager-Benutzerhandbuch, Version 1.0, Beta 3.0, Juli 1997, verfügbar von OBJECTSPACE Inc., zu finden. Soweit arbeitet das obenbeschriebene System in der Weise eines herkömmlichen Datenverarbeitungssystems mit mobilen Agenten.

EIN ÜBERBLICK ÜBER DIE OPERATION DER ERSTEN AUSFÜHRUNGSFORM

[0033] Unter Bezugnahme auf **Fig. 4** wird die während der Laufzeitoperation der vorliegenden Ausführungsform vorhandene Software beschrieben.

[0034] Im Anwender-Computer **10** ist ein Agentensteuerprogramm **40** vorgesehen. Das Agentensteuerprogramm **40** kommuniziert über das Telekommunikationsnetz **20** mit einem Koordinationsagentenprogramm **42**, das in einem ersten Ziel-Computer **38** resident vorhanden ist. Im ersten Ziel-Computer **38** sind außerdem mehrere Aufgabenagentenprogramme **44a-44d** resident vorhanden, von denen jedes beschaffen ist, um mit dem Koordinationsagentenprogramm **42** zu kommunizieren.

[0035] Auf einem weiteren entfernten Computer **30b** ist ein Scout-Agentenprogramm **46** vorgesehen.

[0036] Im Betrieb arbeiten die Aufgabenagentenprogramme **44a-44d**, um einen Teil einer Aufgabe parallel zu verarbeiten. Typischerweise sind alle Aufgabenagentenprogramme **44** völlig gleich.

[0037] Das Koordinationsagentenprogramm **42** tritt mit dem Betriebssystem **321** des entfernten Computers **38**, auf dem es resident vorhanden ist, in Wechselwirkung. Es kommuniziert mit den Aufgabenagentenprogrammen **44a-44d** über lokale Nachrichtenweiterleitung und mit dem Steueragenten-Steuerprogramm **40** über durch das Telekommunikationsnetz **20** übertragene Nachrichten. Der Scout-Agent **46** kommuniziert mit dem Koordinationsagenten **42** und mit dem Agentensteuerprogramm **40** durch das Senden von Nachrichten durch das Netz **20**.

[0038] Um eine sich wiederholende Aufgabe, die ei-

nen hohen Datenverarbeitungsgrad erfordert, auszuführen, führen folglich die Aufgabenagenten **44a–44d** jeder einen Abschnitt der Rechenaufgabe parallel aus. Ein Koordinationsagent **42** steuert die Menge der Datenverarbeitungsbetriebsmittel, die durch die Aufgabenagenten auf dem entfernten Computer **30a** verwendet werden. Er überträgt außerdem die durch die Aufgabenagenten berechneten Ergebnisse zum Agentensteuerprogramm **40**. Schließlich entscheidet er, ob und wann die Bewegung vom entfernten Computer **38** zu einem weiteren entfernten Computer **30b** erfolgt, um die Ausführung dort wieder zu beginnen.

[0039] Das Agentensteuerprogramm **40** ist im Anwender-Computer **10** permanent funktionsfähig, wobei es sich nicht zu den entfernten Computern bewegt. Im Betrieb überwacht es durch den Empfang von Nachrichten von den Scout-Agentenprogrammen **46** und, falls notwendig, durch das Aussenden neuer Scout-Agentenprogramme **46** zu neu lokalisierten Ziel-Computern **30** kontinuierlich die Verwendung und die Verfügbarkeit der Datenverarbeitungsbetriebsmittel auf den entfernten Computern **30a–30d**.

[0040] Wenn ein Anwender wünscht, daß eine neue Aufgabe ausgeführt wird, erlaubt das Agentensteuerprogramm **40** dem Anwender, die auszuführende Aufgabe zu definieren, wobei es einen geeigneten Code für die Aufgabenagenten **44** erzeugt.

[0041] Es bestimmt dann, welcher der entfernten Computer **30** für das Ausführen der Aufgabe geeignet ist, wobei es sie in einer Folge oder einer Reiseroute ordnet. Es erzeugt dann den Code für das Koordinationsagentenprogramm **42** und überträgt diesen zusammen mit dem Code, um die Erzeugung des Aufgabenagentenprogramms **44** zu ermöglichen, zum ersten entfernten Computer **30a** auf der Reiseroute.

[0042] Wenn das Koordinationsagentenprogramm **42** am ersten derartigen entfernten Computer **30a** ankommt, erzeugt es mehrere Aufgabenagentenprogramme **44** und steuert ihren Betrieb in Abhängigkeit von den auf dem entfernten Computer **30a** verfügbaren Datenverarbeitungsbetriebsmitteln. Wenn die Betriebsmittel nicht mehr geeignet sind, weil z. B. ein Anwender des entfernten Computers **30a** dort eine andere Aufgabe begonnen hat, bestimmt der Koordinationsagent **42** einen weiteren entfernten Computer **30b**, zu dem die Bewegung erfolgen soll (entweder aus der Reiseroute oder aus den Meldungen der Scout-Agenten **46**), wobei er sich selbst und die Aufgabenagenten **44** über das Telekommunikationsnetz **20** zum entfernten Computer **30b** bewegt.

DIE OPERATION DES AGENTENSTEUERPROGRAMMS 40

[0043] Die Operation des Agentensteuerprogramms **40** wird nun unter Bezugnahme auf die **Fig. 5** und **6** beschrieben. In einem Schritt **110** startet der Anwender-Computer **10** das Agentensteuerpro-

gramm **40**, wobei die Initialisierung ausgeführt wird, wie im folgenden in **Fig. 6a** ausführlicher beschrieben ist.

[0044] In einem Schritt **110** werden neue geeignete Hosts für die Ferndatenverarbeitung in der Weise beschrieben, die in **Fig. 6b** beschrieben ist. In einem Schritt **130** wird eine Datenbank der verfügbaren entfernten Computer erzeugt und aufrechterhalten, wie im folgenden in **Fig. 6c** ausführlicher beschrieben ist, während in einem Schritt **140** eine Reiseroute der Host-Computer für das oder jedes Agenten-Team berechnet wird, das erzeugt wird.

[0045] Im Schritt **160** werden ein oder mehrere Teams mobiler Agenten erzeugt und zu einem ersten Computer auf jeder Reiseroute geschickt. Im Schritt **170** überwacht das Agentensteuerprogramm **40** die Erzeugung, Beendigung und Bewegung der Teams, wie im folgenden in **Fig. 6d** ausführlicher beschrieben ist, während im Schritt **190** das Agentensteuerprogramm **40** beendet wird, nachdem die gewünschte Aufgabe abgeschlossen worden ist.

[0046] In **Fig. 6a** bestimmt anfangs in einem Schritt **101** das Agentensteuerprogramm **40** die IP-Adresse des Anwender-Computers **10**. In einem Schritt **103** erzeugt das Agentensteuerprogramm **40** eine graphische Anwenderschnittstelle (GUI), die eine Anzeige enthält, wie in **Fig. 10** zeigt ist.

[0047] In einem Schritt **105** erzeugt das Agentensteuerprogramm **40** eine Instanz eines Scout-Agenten **46**, eines Koordinationsagenten **42** und eines Aufgabenagenten **44**, während in einem Schritt **107** das Agentensteuerprogramm **40** eine Konfigurationsdatenbank erzeugt und aktualisiert, die eine Liste aller vorhandenen Agenten und ihre Orte (d. h. die IP-Adressen der Computer, auf denen sie ausgeführt werden) umfaßt.

[0048] In einem Schritt **109** definiert das Agentensteuerprogramm **40** die auszuführende Aufgabe, indem es dem Anwender erlaubt, mit der GUI in Wechselwirkung zu treten, entweder um eine vorhandene vorgegebene Aufgabe auszuwählen oder um durch das Liefern des Codes für die Ausführung eine neue Aufgabe zu erzeugen. Beispiele geeigneter Aufgaben sind Aufgaben, die mehrmals ausgeführt werden müssen, wie z. B. umfassende Suchen nach Lösungen für Gleichungen oder das Knacken von Codes. Ein Beispiel, das unter Verwendung dieser Ausführungsform erfolgreich ausgeführt worden ist, ist eine Auswertung eines genetischen Algorithmus.

[0049] In **Fig. 6b** wird der Schritt des Erzeugens neuer Hosts wie folgt ausgeführt. In einem Schritt **111** werden die entfernten Computer, die nicht bereits eine Agentenplattform tragen, über das Telekommunikationsnetz **20** lokalisiert, wobei in einem Schritt **112** mit derartigen Computern ein Dialog für die Zugriffsberechtigungsprüfung ausgeführt wird. Wenn der Dialog für die Zugriffsberechtigungsprüfung erfolgreich ist (Schritt **113**), dann wird im Schritt **114** eine Telnet-Sitzung mit dem entfernten Computer **30** aufgebaut, während in einem Schritt **115** die Agen-

tenplattform (z. B. ein VOYAGER-Server) zum entfernten Computer **30** heruntergeladen wird. In einem Schritt **116** wird die Agentenplattform auf dem entfernten Computer **30** gestartet, wobei dann im Schritt **117** die Telnet-Sitzung beendet wird.

[0050] Unter Bezugnahme auf **Fig. 6c** werden nun die beim Erzeugen und Aktualisieren der Datenbank der entfernten Hosts **30** ausgeführten Schritte ausführlicher beschrieben. Im Schritt **131** werden mehrere Scout-Agenten **46** erzeugt, wobei jeder zu einem entsprechenden der entfernten Computer **30a–30d** geschickt wird. Beim Senden jedes Agenten sendet das Agentensteuerprogramm **40** jedem Agenten eine Nachricht, die in anweist, sich zu der IP-Adresse des Computers, den er überwachen soll, zu bewegen, wobei es seine Konfigurationsdatenbank aktualisiert, um anzuzeigen, daß der Scout-Agent zu den entsprechenden Computern gegangen ist.

[0051] In einem Schritt **132** empfängt das Agentensteuerprogramm ein Signal von einem der Scout-Agenten **46**, wobei es in einem Schritt **133** eine Datenbank der entfernten Computer **30** aktualisiert (oder erzeugt, falls kein Signal vorausgehend empfangen worden ist), die für jeden entfernten Computer eine Anzeige des Prozessortyps; des verfügbaren Speichers; des Computertyps (z. B. Laptop oder Desktop); der IP-Adresse; der Größe der Festplatte und des verfügbaren Raums; des Typs des Betriebssystems, der JAVA-Versionsnummer; des Niveaus der aktuellen CPU-Nutzung; und aller anderen vom Scout-Agenten **46** verfügbaren Daten umfaßt. Sie enthält außerdem den geographischen Ort (wo er bekannt ist). Dieser kann aus der IP-Adresse oder aus den Daten, die sich am entfernten Computer befinden, wie z. B. der durch die Systemuhr gehaltenen Tageszeit (die die Breite anzeigt) oder der Telefonnummer (die das Land und/oder das Gebiet anzeigt) bestimmt werden.

[0052] In einem Schritt **134** wird die Anzeige nach **Fig. 10** aktualisiert, um in Kurzschriftform die verfügbare Kapazität in einer visuellen Darstellung jedes entfernten Computers **30** graphisch zu zeigen.

[0053] In **Fig. 10** umfaßt diese Anzeige einen Bedienungsfeldbereich **1050**, der aus einer Anzahl von Bereichen für die Auswahl mit einer "Maus" oder einer anderen Cursor-Steuervorrichtung durch einen Anwender besteht, wobei er in dieser Ausführungsform "Schaltflächen" **1051**, um ein Telnet-Programm zu starten, um eine neue Agenten-Server-Plattform zu installieren; **1052** und **1053**, um jeweils einen Kartenanzeigebereich **1070** zu zeigen oder zu verbergen, wie im folgenden erörtert ist; **1054**, um einen ausgewählten Agenten zu beenden; **1055**, um das Lesen des Status eines ausgewählten Agenten anzuweisen; und **1056**, um das Lesen des Status eines ausgewählten Servers anzuweisen, enthält. In jedem Fall wird der Agent oder der Server durch den Anwender graphisch ausgewählt, wie im folgenden erörtert ist. Ein Statusbereich **1058** zeigt für jedes Team den Fortschritt der Aufgabe (d. h. als ein Prozentsatz

der Zeit bis zum Abschluß).

[0054] Der Kartenanzeigebereich **1070**, auf den oben angespielt worden ist, umfaßt eine Karte (hier einen Etagenplan) der Datenverarbeitungsbetriebsmittel im geographischen Bereich des Ausgangs-Servers (in diesem Fall der umgebenden Räume). Er besteht aus den Piktogrammen **1030a–1030h**, von denen jedes einen der entfernten Computer **30a–30c** repräsentiert, und einem Piktogramm **1010**, das den Anwender-Computer **10** repräsentiert. Es ist außerdem ein Piktogramm **1040** vorhanden, das das Vorhandensein des Agentensteuerprogramms **40** auf dem Anwender-Computer **10** anzeigt, dies wird durch die Nähe auf dem Bildschirm repräsentiert.

[0055] Ähnlich ist auf dem Anzeigetext für das Piktogramm **1030** für den Host-Computer, auf dem das Programm gegenwärtig resident vorhanden ist, ein Piktogramm **1042**, **1044** oder **1046** gezeigt, das jeweils ein Agentenkoordinationsprogramm **42**, einen Aufgabenagenten **44** oder einen Scout-Agenten **46** repräsentiert, obwohl dies in **Fig. 10** nicht gezeigt ist. [0056] In dieser Ausführungsform sind die Übertragungen zwischen den entfernten Computern **30** und dem Anwender-Computer **10** graphisch durch verbundene Linien dargestellt, wobei deren Breite die Geschwindigkeit der Informationsübertragung bezeichnet.

[0057] Innerhalb der Anzeige nach **Fig. 10** ist außerdem ein Datenbereich **1080**, um die von den Rechen-Teams empfangenen Daten anzuzeigen, und ein Anzeigebereich **1090** für die Agenten-Support-Plattform, der anzeigt, welche Computer Support-Plattformen installiert haben, vorgesehen.

[0058] Der Anwender kann unter Verwendung einer Cursorsteuerungs-Eingabevorrichtung einen abzufragenden Agenten oder Computer auswählen. Die Anzeige nach **Fig. 10** ist unter Verwendung einer herkömmlichen Anwenderschnittstellentechnologie, wie z. B. dem "Windows"-Betriebssystem, implementiert. [0059] Vorzugsweise ist in dieser Ausführungsform das Agentensteuerprogramm **40** beschaffen, um ein periodisches TCP/IP-"PING"-Signal zu erzeugen, um die Zeitverzögerung zu jedem entfernten Computer zu erfassen. Die Zeitverzögerungsdaten sind in diesem Fall für jeden Computer in der Datenbank der entfernten Computer gespeichert, wobei sie beim Auswählen der Reiseroute verwendet werden können, um die Gesamtverzögerung niedrig zu halten.

[0060] In diesem Fall enthält die GUI außerdem einen Anzeigerahmen, der ein "Bullauge" oder eine Zielansicht zeigt, wobei sich der Anwender-Computer **10** zentral innerhalb der Ansicht und die entfernten Computer **30** bei Peilungen, die ihrer geographischen Peilung vom Anwender-Computer **10** entsprechen, und bei Entfernungen, die den gemessenen Zeitverzögerungen von ihnen entsprechen, befinden.

[0061] Die Anzeige umfaßt ein Piktogramm **2010**, das den Anwender-Computer **10** repräsentiert, und die Piktogramme **2030a–2030c**, die jedes einen der

entfernten Computer **30a–30c** repräsentieren.

[0062] Im Schritt **135** bestimmt das Agentensteuerprogramm **40**, ob der Scout-Agent **46** bewegt werden sollte (z. B. wenn der entfernte Computer **30** außer Betrieb geht), wobei es irgendwelche notwendigen Bewegungsbefehle verursacht. Diese Überwachungs- und Aktualisierungsschritte **132–135** werden kontinuierlich durch die Operation des Agentensteuerprogramms **40** oder in periodischen Intervallen ausgeführt. In dieser Ausführungsform sind die Scout-Agenten **46** beschaffen, um sich selbst auf dem entfernten Computer **30** zu beenden, auf dem sie ausgeführt werden, falls sie nicht in einer vorgegebenen Periode eine Nachricht von Agentensteuerprogramm **40** empfangen.

[0063] Der Schritt **140** des Erzeugens einer Reiseroute wird nun ausführlicher erörtert. Nachdem das Agentensteuerprogramm eine Datenbank der verfügbaren Host-Computer konstruiert hat, fährt es fort, zu bestimmen, wie viele Agenten-Teams erzeugt werden sollten. Als ein Beispiel kann angenommen werden, daß die entfernten Computer **30** eine Anzahl relativ leistungsfähiger Netz-Server-Computer, eine Anzahl (weniger leistungsfähiger) Konstruktions-Arbeitsplatzrechner, eine Anzahl (weniger leistungsfähiger) Personal-Computer und eine Anzahl von Laptop-Computern umfassen. In jedem derartigen entfernten Computer wird bereits eine Anzahl von Programmen ausgeführt, so daß die für die auszuführende Aufgabe verfügbaren Betriebsmittel durch sich ändernde Mengen eingeschränkt sind.

[0064] Das Agentensteuerprogramm **40** bestimmt einen Nützlichkeitsfaktor für jeden entfernten Computer als das Produkt eines Index der Leistung des Computers (der z. B. eine gewichtete Summe seiner Prozessorgeschwindigkeit und des verfügbaren Speichers umfaßt) und einem Bruchteil seiner Kapazität, der verfügbar ist.

[0065] Um einen störenden Eingriff in die entfernten Computer zu vermeiden, selbst wenn ein Scout-Agent **46** meldet, daß fast nichts der Kapazität eines entfernten Computers **30** verwendet wird, legt das Agentensteuerprogramm **40** eine maximale Kapazität für die Nutzung (z. B. von 50% der Betriebsmittel des entfernten Computers) fest.

[0066] Der auf diese Weise für jeden entfernten Computer berechnete Index repräsentiert folglich die Erwünschtheit des Computers für die Verwendung bei der Lösung der Aufgabe.

[0067] Als nächstes bestimmt das Agentensteuerprogramm **40**, wie viele Agenten-Teams herzustellen sind. Für ein Parallel-Rechenproblem wird im allgemeinen irgendein Leistungsvorteil gewonnen, indem eine Anzahl von Instanzen des Programms konfiguriert wird, die das Problem auf einem gegebenen einzelnen Prozessor lösen soll, wobei unterhalb oder oberhalb dieser Anzahl die Leistung verringert ist. Falls folglich die auszuführende Aufgabe z. B. eine umfassende Suche nach Lösungen für eine Gleichung mit 100 Startpunkten ist, erzeugt das Agenten-

steuerprogramm eine ausreichende Anzahl von Agenten-Teams, um **100** Aufgabenagenten **44** zu schaffen, wobei sich N Aufgabenagenten in jedem Team befinden, dabei zeigt N die optimale Anzahl der Aufgabenagenten pro verschiedenen entfernten Computer an, wobei die Anzahl der entfernten Computer (und folglich der Agenten-Teams) $100/N = M$ beträgt.

[0068] Nächstes werden aus der Datenbank der entfernten Computer die M Computer mit dem höchsten Nützlichkeitsindex (d. h. die leistungsfähigsten Computer mit der meisten verfügbaren Kapazität) als die entsprechenden Startpunkte für die Reiseroute jedes Agenten-Teams ausgewählt.

[0069] Wenn eine große Anzahl geeigneter Computer mit etwa äquivalenten Kapazitäten verfügbar ist, dann werden die Positionsinformationen (wo sie verfügbar sind) für jeden Computer verwendet, um die Startpunkte für jede Reiseroute auszuwählen, die räumlich getrennt sind und/oder sich relativ nah beim Anwender-Computer **10** befinden.

[0070] Die räumliche Trennung unterstützt die Verhinderung von Konflikten zwischen verschiedenen Agenten-Teams und die Verteilung des Einflusses der Ferndatenverarbeitungsaufgabe, anstatt sie in einem geographischen Bereich (und folglich wahrscheinlich einen Unternehmensbereich) zu konzentrieren.

[0071] Das Agentensteuerprogramm **40** untersucht dann die Datenbank der entfernten Computer, um (in der gleichen Weise) die nächsten M brauchbarsten Computer zu bestimmen, und wählt aus diesen (oder wenn mehr als M Computer mit starker ähnlicher Kapazität verfügbar sind, dann aus dieser größeren Gruppe) für jedes Agenten-Team einen nächsten Punkt der Reiseroute unter Verwendung der Ortsinformationen (wo sie verfügbar sind) aus, um einen Computer auszuwählen, der sich relativ nah zum ersten in der betreffenden Reiseroute befindet.

[0072] Dieser Prozeß wird wiederholt, bis eine vorgegebene Anzahl P von Computern für jede Reiseroute ausgewählt worden ist.

[0073] Ferner wird unter Verwendung der Daten von den Scout-Agenten **46** jede Reiseroute periodisch verfeinert, wobei, falls notwendig, Änderungen an der Reiseroute jedem Koordinationsagenten **42** übermittelt werden. Wenn Computer verfügbar werden, die leistungsfähiger als diejenigen sind, auf denen die Teams gegenwärtig resident vorhanden sind, übermittelt das Agentensteuerprogramm **40** einem oder mehreren Teams, sich zu den neu verfügbaren Computern zu bewegen.

[0074] Die Reiseroute umfaßt in dieser Ausführungsform eine einfache verkettete Liste von IP-Adressen, zu denen sich das Agenten-Team nacheinander bewegen soll.

[0075] Unter Bezugnahme auf **Fig. 6d** wird nun der Agentenüberwachungsschritt nach Schritt **170** nach **Fig. 5** ausführlicher erörtert.

[0076] In einem Schritt **171** empfängt das Agenten-

steuerprogramm von einem Koordinationsagenten **42** eine Nachricht, die eines oder mehrere Elemente der Ergebnisdaten von einem oder mehreren Aufgabenagenten **44** enthält. Beim Empfang einer derartigen Nachricht aktualisiert im Schritt **175** das Agentensteuerprogramm **40** das aktuelle Ergebnis der Aufgabe und zeigt die Ergebnisse auf der GUI an.

[0077] Wenn z. B. die Aufgabe ist, eine Suche nach einer Lösung für eine Gleichung auszuführen, wird das aktuell beste Ergebnis von jedem der Aufgabenagenten **44** auf der GUI angezeigt.

[0078] Falls die Aufgabe eine Endbedingung enthält (z. B. die Erzeugung von Klartext aus einer Entschlüsselungsaufgabe oder das Erreichen eines vorgegebenen Konvergenzniveaus in der Aufgabe für die Lösung einer Gleichung), bestimmt das Steuerprogramm **40**, ob die Bedingung erfüllt worden ist, wobei es, falls die Bedingung erfüllt ist, die Agenten beendet (Schritt **190**, der im folgenden erörtert ist).

[0079] Das Agentensteuerprogramm **40** erfaßt im Schritt **172** irgendwelche Nachrichten von den Koordinationsagenten **42** oder den Scout-Agenten **46**, die anzeigen, daß sie oder irgendwelche Aufgabenagenten **44** sich zu einem anderen Ort bewegt haben, wobei es den Datensatz der Agenten-Konfigurationsdatenbank für den betreffenden Agenten aktualisiert (Schritt **176**), um die IP-Adresse des Computers **30** anzuzeigen, auf dem der Agent resident vorhanden ist.

[0080] Im Schritt **173** erfaßt das Agentensteuerprogramm **40** von den Koordinationsagenten **42** irgendeine Nachricht, die die Erzeugung oder Zerstörung eines Aufgabenagenten oder eines Koordinationsagenten anzeigt, wobei es dies im Schritt **176** in der Konfigurationsdatenbank aufzeichnet.

[0081] Im Schritt **174** erfaßt das Agentensteuerprogramm **40** irgendeine Rückkehr eines Agenten zum Anwender-Computer **10**, wobei es im Schritt **176** den Datensatz der Konfigurationsdatenbank für den Agenten aktualisiert, um den neuen Ort des Agenten aufzuzeichnen. Es fragt dann den Agenten ab, um alle Ergebnisdaten, die er trägt, zu lesen, wobei es im Schritt **175** die Aufgabe aktualisiert, wie oben erörtert ist.

[0082] Im Schritt **190** in Fig. 6e gibt das Steuerprogramm **40**, wenn das gewünschte Ergebnis erreicht worden ist oder aus irgendeinem anderen Grund (z. B. die Nichtverfügbarkeit der Datenverarbeitungsbetriebsmittel) die Ergebnisse der Aufgabe im Schritt **192** aus (z. B. in eine auf einer Festplatte gespeicherte Datei), und sendet dann im Schritt **194** ein Signal an jeden Koordinationsagenten **42**, damit er sich selbst und seine Aufgabenagenten beendet, und an jeden Scout-Agenten **46**, damit er sich selbst beendet. Im Schritt **196** werden die Agentenplattformen **326** aus denjenigen entfernten Computern **30** entfernt, auf denen sie installiert waren, indem eine Telnet-Sitzung erzeugt wird (wie in Fig. 6b beschrieben ist). Schließlich beendet das Agentenprogramm **40** die Ausführung.

[0083] Es ist deshalb in groben Zügen zu sehen, daß das Agentensteuerprogramm die folgenden Funktionen ausführt:

1. Anfängliches Erzeugen neuer mobiler Agentenplattformen **326** und Löschen dieser am Ende der Sitzung. (Diese Operationen könnten außerdem während einer Aufgabe ausgeführt werden, wenn das Agentensteuerprogramm **40** einen neuen Computer erfaßt oder ein vorhandener Computer unbenutzbar wird.)
2. Senden der Scout-Agentenprogramme **46** an jeden entfernten Computer **30** und Überwachen der von jedem empfangenen Daten, um ein Modell der verteilten Datenverarbeitungsumgebung aufrechtzuerhalten. Diese Aktivität wird während der Aufgabe kontinuierlich ausgeführt, wobei sie einen Ausführungs-Teilprozeß des Agentensteuerprogramms **40** bildet.
3. Überwachen und Steuern der Positionen und Bewegungen der Agenten. Dies bildet einen weiteren Ausführungs-Teilprozeß des Agentensteuerprogramms **40**.
4. Empfangen und bis zu einem gewissen Grad Verwenden der durch die Aufgabenagenten weitergeleiteten Ergebnisse. Dies bildet einen dritten Teilprozeß der Ausführung des Agentensteuerprogramms.
5. Analysieren der verfügbaren Betriebsmittel, um eine anfängliche Reiseroute der entfernten Computer **30** für jedes Agenten-Team zu konstruieren. Dieser Analyseprozeß bildet deshalb einen weiteren Teilprozeß der Ausführung des Agentensteuerprogramms.
6. Aufrechterhalten und dynamisches Aktualisieren der Anzeige, die die geographischen Positionen der entfernten Computer **30**, der darauf resident vorhandenen Agenten-Teams und vorzugsweise die Fähigkeiten jedes Computers **30** zeigt, zusammen mit einer Anzeige der Ergebnisse der Aufgabe bis jetzt. Diese Anzeige erfolgt in der Form einer graphischen Anwenderschnittstelle, über die ein Anwender die Elemente der Operation des Prozesses steuern kann (z. B. ein Team bewegen, ein Team erzeugen oder löschen).

[0084] Die Ergebnisdaten und die Überwachungsdaten von den Agenten können zum Agentensteuerprogramm **40** entweder durch Nachrichten, die über das Telekommunikationsnetz **20** zurück übermittelt werden, oder durch die Rückkehr des betroffenen Agenten, gefolgt von seiner lokalen Abfrage, zurückgeschickt werden, wie oben beschrieben ist.

[0085] Das erstere Verfahren ist vorteilhafter, wenn die Ergebnisse oder die Überwachungsdaten relativ häufig erzeugt werden, weil es schneller ist, lediglich die Ergebnisdaten zu übermitteln, anstatt den Code für den Agenten zu bewegen. Das letztere ist vorzuziehen, wenn die Überwachungsdaten relativ selten erforderlich sind, weil es das Scout-Agentenprogramm **46** aus dem Speicher des entfernten Compu-

ters **30** entfernt und folglich die Aufdringlichkeit des Ferndatenverarbeitungsprozesses im entfernten Computer **30** selbst verringert. In dieser Ausführungsform ist aus diesem Grund das letztere Verfahren im allgemeinen bevorzugt.

DIE OPERATION DES SCOUT-AGENTENPROGRAMMS 46

[0086] In **Fig. 7** liest im Schritt **202** bei der Ankunft an einem entfernten Computer **30** das Scout-Agentenprogramm den Status des Computers. Demzufolge führt das Scout-Agentenprogramm **46** die folgenden Prüfungen aus:

1. Der Typ und die Version des Betriebssystems (dies wird durch das Ausführen eines Aufrufs für eine Standardroutine innerhalb der virtuellen JAVA-Maschine **324** ausgeführt).
2. Die JAVA-Versionsnummer (sie wird durch das Ausführen eines Aufrufs für die virtuelle JAVA-Maschine **324** erhalten).
3. Der Gesamtspeicher **32** und freie Speicher. Diese werden gleichermaßen durch das Ausführen von Aufrufen für die JAVA-Plattform erhalten.
4. Das Abtasten der Tastaturaktivität. Dies wird ausgelöst, wann immer das Betriebssystem ein durch eine Anwendermanipulation der Tastatur **45** (oder einer anderen Eingabevorrichtung, wie z. B. der Maus) verursachtes Ereignis erfaßt. Andere Eingangsströme (z. B. LAN-Anschlüsse) können außerdem überwacht werden.
5. Die CPU-Aktivität. Diese wird durch den folgenden Prozeß überwacht:
 - a) Start eines Teilprozesses mit maximaler Priorität der Ausführung für eine kurze Zeit und Aufzeichnen eines ganzzahligen Zählerstandes der Anzahl der Ausführungen.
 - b) Start eines Teilprozesses mit minimaler Priorität und Aufnehmen einer weiteren Anzeige des Zählerstandes.
 - c) Mitteilung der Auswirkung des zweiten Teilprozesses auf den ersten. Je beschäftigter die CPU **31** ist, desto größer ist die Wirkung.

[0087] Der Scout-Agent **46** bestimmt, ob das Betriebssystem **321** das eines tragbaren Computers oder eines Desktop-Computers ist. Falls ein tragbarer Computer erfaßt wird, werden vorzugsweise die Lebensdauer der Batterie, der Status der Festplatte **33** und der Status der Anzeige **34** erfaßt und überwacht.

[0088] Wenn eine Statusänderung aufgezeichnet wird (Schritt **204**), wird sie im Schritt **206** zum Agentensteuerprogramm **40** zurück übermittelt.

[0089] Alternativ könnte sich das Scout-Programm **46** zurück zum Anwender-Computer **10** bewegen, wie oben erörtert ist.

[0090] Innerhalb eines weiteren Ausführungs-Teilprozesses empfängt der Scout-Agent **46** Nachrichten vom Agentensteuerprogramm **40** (Schritt **208**). Wenn

die Nachricht einen Befehl für den Scout-Agenten bildet (Schritt **210**), sich zu beenden, dann führt er dies im Schritt **212** aus. Wenn sie einen Befehl bildet (Schritt **210**), sich zu einem neuen Computer zu bewegen, dann führt dies der Scout-Agent **46** im Schritt **214** aus. Sowohl das Bewegungsverfahren als auch das Beendungsverfahren ist innerhalb des in dieser Ausführungsform verwendeten Voyager-Agentensystems vorgesehen.

[0091] In dieser Ausführungsform besitzt jeder Scout-Agent **46** eine eingeschränkte vorgegebene Lebensdauer, wobei er außerdem den Schritt **212** aufruft, um sich selbst zu beenden, falls er innerhalb dieser vorgegebenen Zeitperiode keine Nachricht vom Agentensteuerprogramm **40** empfangen hat.

[0092] Schließlich ist der Scout-Agent vorzugsweise so beschaffen, um sein Vorhandensein auf dem entfernten Computer **30** durch das Anzeigen eines Bildes auf der VDU **34** des entfernten Computers anzuzeigen, (z. B. eines Piktogramms, das sein Vorhandensein anzeigt).

DIE OPERATION DES KOORDINATIONSAGENTEN 42

[0093] Unter Bezugnahme auf **Fig. 8** wird nun die Operation des Koordinationsagenten **42** beschrieben.

[0094] Bei der Erzeugung ist der Koordinationsagent **42** beschaffen (Schritt **502**), um seine gespeicherte Kopie seiner Reiseroute zu lesen und den nächsten entfernten Computer auf der Reiseroute zu bestimmen. Bei der Erzeugung wird dies natürlich der erste auf der Reiseroute sein.

[0095] In einem Schritt **504** bewegt der Koordinationsagent **42** sein Team, wie im folgenden unter Bezugnahme auf **Fig. 8c** beschrieben ist.

[0096] Im Schritt **506** liest der Koordinationsagent **42** den Status des Computers, auf dem er angekommen ist. In dieser Ausführungsform sind die ausgeführten Statusüberprüfungen die gleichen wie diejenigen, die durch den Scout-Agenten **46** ausgeführt werden, wobei sie durch den gleichen Code implementiert sind.

[0097] Im Schritt **508** bestimmt der Koordinationsagent **42** basierend auf den im vorhergehenden Schritt abgeleiteten Daten über den freien Speicher und die CPU-Nutzung, wieviel der verfügbaren Kapazität verwendet werden sollte. Im allgemeinen läßt der Koordinationsagent **42** etwas freien Speicher und etwas CPU-Zeit, falls der Anwender des entfernten Computers **30** gleichzeitig irgendeine weitere Aufgabe ausführen will. Es können z. B. (angenommen) bis zu 30% des Speichers durch den Koordinationsagenten **42** als frei zugewiesen werden.

[0098] Im Schritt **510** bestimmt der Koordinationsagent **42**, ob die verbleibende Kapazität über diejenige, die freizulassen ist, einen zusätzlichen Aufgabenagenten **44** bei der Ausführung unterstützen wird, oder ob der ein Aufgabenagent **44** beendet werden

muß, um die notwendige Menge des freien Raumes zu erreichen. Im Schritt **512** wird, wenn es ausreichend zusätzlichen Raum gibt, um einen zusätzlichen Aufgabenagenten aufzunehmen, ein zusätzlicher Aufgabenagent **44** erzeugt; wenn es notwendig ist, die Anzahl der Aufgabenagenten zu verringern, dann wird einer beendet, wie im folgenden ausführlicher beschrieben ist. In beiden Fällen wird die Erzeugung oder Zerstörung eines Agenten im Schritt **514** dem Agentensteuerprogramm **40** übermittelt. Der Prozeß wird dann im Schritt **506** wiederholt, bis so viele Aufgabenagenten, wie erforderlich sind, oder wie auf dem entfernten Computer **30** unterstützt werden können, erzeugt worden sind.

[0099] Dann erzeugt im Schritt **516** der Koordinationsagent **42** eine Anzeige auf der VDU **34** des entfernten Computers **30**, die sein Vorhandensein und das Vorhandensein der Aufgabenagenten **44** anzeigt.

[0100] Während der Operation des Koordinationsagenten **42** überwacht er den Computer **30** (Schritt **520**). Im Fall einer Änderung (Schritt **522**) wird das Ergebnis zum Agentensteuerprogramm **40** gemeldet. Wenn die Änderung signifikant ist (Schritt **524**), wie z. B. daß der Anwender die Tastatur **35** manipuliert, oder daß eine Aufgabe mit hoher Priorität auf der CPU **31** gestartet wird, dann leitet der Koordinationsagent **42** eine Bewegung ein, indem er zurück zum Schritt **502** weitergibt.

[0101] Die Überwachung der Aktivität des Computers **30** bildet einen Teilprozeß der Ausführung des Koordinationsagenten **42**.

[0102] Als ein weiterer Teilprozeß der Ausführung erfaßt der Koordinationsagent **42** Nachrichten von den Aufgabenagenten, die Ergebnisse weitergeben (Schritt **526**), wobei er diese dem Agentensteuerprogramm **40** übermittelt (Schritt **528**). In dieser Ausführungsform wird irgendeine Kollation der Ergebnisse durch den Koordinationsagenten **42** ausgeführt, anstatt die Ergebnisse unmittelbar weiterzuleiten. Um die Anzahl der separaten Signale und folglich den Zusatzaufwand der Adressierung zu verringern, werden Gruppen der Ergebnisdaten für die Weiterleitung in einer einzelnen Nachricht kombiniert.

[0103] Im Schritt **530** überprüft der Koordinationsagent **42** auf Nachrichten vom Agentensteuerprogramm **40**, die entsprechend ihren Inhalten interpretiert werden (Schritt **532**). Wenn die Nachricht dazu dient, die Reiseroute zu aktualisieren, dann werden im Schritt **552** in Fig. 8e die neuen IP-Adressen der betroffenen Computer innerhalb der Reiseroute ersetzt.

[0104] Falls das Signal lautet, das Agenten-Team zu bewegen, kehrt das Koordinationsprogramm **42** zum Schritt **502** zurück. In Fig. 8c wird nach dem Bestimmen in der nächsten Adresse in der Reiseroute im Schritt **502** der Schritt **504** ausgeführt, der für jeden Aufgabenagenten das Senden von Befehlen an diesen Aufgabenagenten, damit er sich selbst zu der Adresse bewegt (Schritt **534**), und, wenn nicht der

letzte Agent verarbeitet worden ist, das Übermitteln der Bewegung zurück zum Agentensteuerprogramm **40** (Schritt **538**) und das Auswählen des nächsten Agenten (Schritt **540**) umfaßt. Nach der Verarbeitung des letzten Aufgabenagenten **44** bewegt sich der Koordinationsagent **42** selbst (Schritt **504**).

[0105] Wenn das Agenten-Team zu beenden ist, wird in Fig. 8d gleichermaßen jedem Aufgabenagenten **44** eine Nachricht geschickt (Schritt **542**), damit er sich selbst beendet, wobei die Beendigung dem Agentensteuerprogramm **40** gemeldet wird (Schritt **546**), wobei diesem folgend der Koordinationsagent **42** sich selbst beendet (Schritt **550**).

[0106] Folglich schafft der Koordinationsagent **42** auf jedem entfernten Computer **30** eine zentrale Steuereinrichtung um den Zustand der Maschine zu überwachen, eine Bewegung einzuleiten, wo es notwendig ist; und die Kommunikation zwischen dem Agentensteuerprogramm **40** und den Aufgabenagenten **44** zu koordinieren.

[0107] Durch das Schaffen einer einzelnen Steuerungsfunktion für die mehreren Aufgabenagenten **44** wird die Komplexität von jedem verringert, wobei das Volumen der Nachrichtenübermittlung über das Telekommunikationsnetz **20** verringert wird.

[0108] Indem vorgesehen wird, daß die Entscheidung darüber, wann die Bewegung zu einem neuen entfernten Computer erfolgen soll, basierend auf der lokalen Überwachung lokal getroffen wird, wird eine schnellere Reaktion auf den Zustand des Computers erhalten. Wenn folglich ein Anwender des entfernten Computers **30** die Maschine verwenden will, ist es nicht notwendig, auf die Ergebnisse eines Nachrichtenübermittlungsprozesses mit dem sich entfernt befindenden Agentensteuerprogramm **40** zu warten, bevor der entfernte Computer **30** von den mobilen Agentenprogrammen befreit werden kann.

[0109] Jeder Netzkoordinationsagent **42** ist im Prinzip innerhalb seiner Struktur standardisiert, weil er die Bewegungs- und Kommunikationsfunktionen des Teams steuert, aber in dieser Ausführungsform an der Ausführung der Aufgabe selbst nicht teilnimmt.

[0110] Um zu verhindern, daß im Fall irgendeines Ausfalls des Betriebs des Koordinationsagenten **42** die entfernten Computer **30** mit unerwünschten Aufgabenagenten **44** überladen werden, ist dem Aufgabenagenten **44** in dieser Ausführungsform lediglich eine eingeschränkte Lebensdauer gegeben, wobei er sich selbst beenden wird, wenn er nicht innerhalb dieser Lebensdauer eine Nachricht vom Koordinationsagenten **42** empfängt. Der Koordinationsagent **42** ist beschaffen, um derartige Nachrichten periodisch in Intervallen zu senden, die kleiner als die betroffene Lebensdauer sind. Wenn der Koordinationsagent **42** das Team zu einem weiteren entfernten Computer **30** bewegt oder sich selbst beendet hat, werden sich folglich die Aufgabenagenten **44** beenden.

DIE OPERATION DES AUFGABENAGENTEN 44

[0111] Jeder Aufgabenagent **44** besteht aus zwei Teilen; einem generischen Teil und einem aufgabenspezifischen Teil. Der generische Teil besteht aus einem mobilen Voyager-Agenten, der beschaffen ist, wie oben angegeben ist, um sich nach einer vorgegebenen Zeit selbst zu beenden, wenn er nicht die Signale vom Koordinationsagenten **42** empfängt; um die Signale vom Koordinationsagenten **42** zu erwarten und sich nach ihnen zu richten, um sich zu bewegen oder selbst zu beenden; und um die Ergebnisse zum Koordinationsagenten **42** weiterzuleiten.

[0112] Der aufgabenspezifische Teil ist beschaffen, um eine wiederholte Rechenaufgabe auszuführen, wie z. B. das Lösen einer Gleichung, die Entschlüsselung oder das Suchen von Zeichenketten, und um periodisch oder bei der Erfüllung einer Aufgabenbedingung ein Ergebnis zu erzeugen, das für die Weiterleitung zum Koordinationsagenten **42** geliefert wird.

[0113] Demzufolge umfaßt in **Fig. 9** der Schritt **402** eine Ausführung der Aufgabe; der Schritt **404** umfaßt das Übermitteln ihrer Ergebnisse zum Koordinationsagenten **42**; der Schritt **406** umfaßt das Empfangen eines Signals vom Koordinationsagenten **42** und (Schritt **408**) in Abhängigkeit von seinen Inhalten das Beenden (Schritt **410**) oder das Bewegen (Schritt **412**), wie es geeignet ist. Diese Schritte sind, wie oben angegeben ist, in zwei wiederholt ausgeführten Teilprozessen angeordnet.

[0114] Der Koordinationsagent **42** enthält ein Verfahren für das Erzeugen eines neuen Aufgabenagenten aus einer gespeicherten Aufgabenagentenklasse, dessen generischer Teil als ein Teil des Netzkoordinationsagenten gespeichert ist und dessen aufgabenspezifischer Teil anfangs durch das Agentensteuerprogramm **40** geliefert wird.

[0115] Wo es nicht ausführlich beschrieben ist, ist es selbstverständlich, daß die Aspekte des Voyager-Systems die Funktionalität bereitstellen, um die Erfindung zu implementieren. Das Beenden wird z. B. in dieser Ausführungsform unter Verwendung der Voyager-Funktion `dieNow()` ausgeführt, während die Bewegung durch die Voyager-Funktion `MoveTo("Ortsname")` ausgeführt wird.

MÖGLICHE ANWENDUNGEN

[0116] Das folgende ist eine nichterschöpfende Liste möglicher Anwendungen der Erfindung:

1. die parallele Verarbeitung (z. B. genetische Algorithmen, Lösungssuchen),
2. die Kryptographie, das verteilte Knacken von Schlüsseln,
3. die Sprachverarbeitung,
4. das Suchen entfernter Computer,
5. die verteilte Echtzeitsteuerung (z. B. mobiler Roboter),
6. Graphik; 3D-Bildaufbreitung,

7. die Verwendung mobiler Computer-Betriebsmittel.

MOBILE ENTFERNT E COMPUTER

[0117] Vorzugsweise ist in dieser Ausführungsform der Koordinationsagent **42** beschaffen, um den Kommunikationsanschluß **37** zu überwachen, wobei er beschaffen ist, alle von den Aufgabenagenten **44** empfangenen Ergebnisse zu puffern, wenn die Verbindung des Anschlusses mit einem Telekommunikationsnetz endet.

[0118] Bei der nächsten Gelegenheit, zu der das Telekommunikationsnetz **20** verfügbar wird, übermittelt der Koordinationsagent **42** die Ergebnisse und alle Überwachungsdaten, die erhalten worden sind, seit das Netz zuletzt nicht verfügbar geworden war, dem Agentensteuerprogramm **20**. Ähnlich werden im Agentensteuerprogramm **20** alle seit der letzten vom Agenten-Team empfangenen Nachricht zum Agenten-Team gesendeten Nachrichten gepuffert, wobei sie beim nächsten Empfang einer Nachricht von ihm erneut durch das Telekommunikationsnetz gesendet werden.

[0119] Folglich kann diese Ausführungsform der Erfindung ein Agenten-Team auf einen mobilen Computer schicken und von den Betriebsmitteln des mobilen Computers Gebrauch machen, während es nicht mit dem Telekommunikationsnetz **20** kommuniziert, wobei es die Ergebnisse sammelt, wenn es das nächste Mal verfügbar wird.

ANDERE AUSFÜHRUNGSFORMEN UND MODIFIKATIONEN

[0120] Im vorangehenden ist die Operation des Scout-Agenten beim Melden des Niveaus der Verwendung des entfernten Computers beschrieben. Es ist außerdem vorstellbar, daß der Scout-Agent den Zustand des entfernten Computers bewerten würde, in dem er lokale Ereignisse überwacht, um zu überprüfen, ob die Tastatur gedrückt wird, und indem er aufzeichnet, ob die Menge des freien Speichers statisch, ansteigend oder abfallend ist. Er kann die Änderungen im Ausmaß der Verwendung im der entfernten Maschine anstatt des tatsächlichen Niveaus der Verwendung melden.

[0121] Das Agentensteuerprogramm kann beschaffen sein, um die historische Aktivität der verschiedenen entfernten Computer (und, wo verschiedene Anwender des entfernten Computers bekannt sind, das Aktivitätsmuster jedes Anwenders) zu überwachen, und um dies zu verwenden, um eine Reiseroute für jedes Agentenprogramm zu erzeugen, um zu sichern, daß die Agenten nicht zu Computern geschickt werden, die gegenwärtig verfügbar sind, die aber bald beschäftigt sein werden.

[0122] Durch Vorgabe kann, wo die Verwendungsinformationen nicht verfügbar sind, die Kenntnis der Zeitzone, in der sich der Computer befindet, verwen-

det werden, um z. B. vorherzusagen, daß es wahrscheinlicher ist, daß der Computer während der Nachtstunden in dieser Zeitzone verfügbar ist.

[0123] Wenn sich ein Agenten-Team aus Programmen auf einem entfernten Computer befindet und die Aktivität an diesem Computer beginnt, kann das Agentenkoordinationsprogramm oder der Koordinationsagent entscheiden, die Operation der Agentenprogramme vorübergehend anzuhalten, aber sie auf dem entfernten Computer zu lassen, anstatt das Team der Programme zu einem weiteren Computer zu bewegen. Dies ist geeignet, wo z. B. die Überwachung der historischen Aktivität dieses entfernten Computers nahelegt, die Verwendung nur für kurze Perioden auftreten wird, so daß der Computer in Kürze für das Agenten-Team nutzbar wird.

[0124] Obwohl die Aufgabenagentenprogramme als unabhängig auf dem entfernten Computer wirkend beschrieben worden sind, kann es in den Ausführungsformen bevorzugt sein, diesen Programmen zu erlauben, auf die Betriebsmittel des entfernten Computers nur durch den Koordinationsagenten zuzugreifen. Folglich besitzt der Koordinationsagent Zugriff auf die Betriebssystem-Routinen, um Platten zu lesen und zu schreiben oder um Speicher zuzuweisen, wobei diese Routinen durch die Aufgabenagentenprogramme aufgerufen werden.

[0125] In derartigen Ausführungsformen ist der Koordinationsagent beschaffen, um Sicherheitsüberprüfungen auszuführen, z. B. um zu sichern, das einzelne Aufgabenagenten nur die Dateien modifizieren können, die sie selbst erzeugt haben. Die Sicherheit dieser Art kann durch jeden Aufgabenagenten eingerichtet werden, der ein Anforderungssignal an den Koordinationsagenten unter Verwendung eines Austausches von Schlüsseln überträgt, so daß "Virus"-Programme nicht auf einen entfernten Computer eingeführt werden können, der programmiert ist, um ein Team aus Aufgabenagenten zu akzeptieren.

[0126] Außerdem kann jeder Aufgabenagent die Aufgabe identifizieren, die er ausführt, wenn er Zugriff auf die Systembetriebsmittel anfordert. Dies versetzt den Koordinationsagenten in die Lage, zu bestimmen, ob es erlaubt ist, daß Aufgaben dieser Art (z. B. für einen gegebenen identifizierten Anwender) auf einer gegebenen entfernten Maschine ausgeführt werden, auf der die Aufgabenagenten resident vorhanden sind.

[0127] Angesichts des vorangehenden werden für einen Fachmann viele andere Alternativen, Varianten oder Modifikationen offensichtlich sein. Es können z. B. andere Agentenplattformen als die obenbeschriebene Voyager-Plattform verwendet werden. Genauso erfordern die Aspekte der Erfindung nicht die "Agenten"-Technologie, dieser Begriff ist jedoch für ihre Implementierung definiert.

[0128] Obwohl das Agentensteuerprogramm **20** als ein statisches Programm auf dem Anwender-Computer **10** beschrieben worden ist, ist es möglich, es in die Lage zu versetzen, sich zu einem weiteren Com-

puter zu bewegen, wenn das gewünscht wird. Obwohl die Scout-Agentenprogramme **46** als mobile Agenten beschrieben sind, würde es möglich sein, eine statische Überwachung innerhalb der Agentenplattform vorzusehen, obwohl auf Kosten einer permanenteren Belegung der Betriebsmittel in dem entfernten Computer.

[0129] Während die vorangehende Ausführungsform mehrere ähnliche Aufgabenagentenprogramme für die parallele Ausführung beschreibt, ist es selbstverständlich, daß mehrere verschiedene Typen vorgesehen sein könnten. Für eine Fernsteuerungsanwendung könnten z. B. verschiedene Teile eines komplexen Mechanismus, wie z. B. eines Raumfahrzeugs, jeder durch ein anderes Parallelüberwachungsprogramm gesteuert werden, die alle durch ein Koordinationsprogramm koordiniert werden.

[0130] Demzufolge ist es selbstverständlich, daß die Erfindung nicht auf die vorangehenden Ausführungsformen eingeschränkt ist, sondern daß sie sich auf alle anderen Modifikationen oder Variationen erstreckt, die für den ausgebildeten Leser offensichtlich sind. Es wird für jeden und alle neuartigen Gegenstände oder jede und alle Kombinationen daraus, die hierin beschrieben sind, Schutz gesucht.

Patentansprüche

1. Ferndatenverarbeitungssystem, das umfaßt: einen ersten Computer (**10**); und wenigstens einen zweiten Computer (**30**), der daran über eine Kommunikationsverbindung (**20**) angeschlossen ist; wobei der erste Computer (**10**) so programmiert ist, daß er an den zweiten Computer (**30**) über die Verbindung (**20**) Daten überträgt, die ein Rechen-Team für die Ausführung einer Rechenaufgabe definieren; wobei das Team ein erstes Programm (**42**) und wenigstens ein von dem ersten Programm verschiedenes zweites Programm (**44**) umfaßt, die auf demselben Computer parallel ausgeführt werden; und wobei der zweite Computer (**30**) so programmiert ist, daß er die Daten empfängt und das erste und das wenigstens eine zweite Programm (**42**, **44**) parallel ausführt, **dadurch gekennzeichnet**, daß das oder jedes zweite Programm (**44**) einen Code für die Ausführung wenigstens eines Teils der Aufgabe und für die Kommunikation mit dem ersten Programm (**42**) umfaßt; und das erste Programm (**42**) ein Koordinationsprogramm ist, das einen Code für die Kommunikation mit dem ersten Computer (**10**) und für die Kommunikation mit jedem zweiten Programm (**44**) und für dessen Koordination umfaßt.

2. System nach Anspruch 1, bei dem das erste Programm einen Code zum Übertragen des ersten Programms an einen weiteren solchen Computer in Reaktion auf ein vorgegebenes Kriterium umfaßt.

3. System nach Anspruch 2, bei dem das erste

Programm so beschaffen ist, daß es einen der mehreren Computer bestimmt, zu dem es sich bewegt.

4. System nach Anspruch 3, bei dem das erste Programm so beschaffen ist, daß es eine Folge speichert, die eine Präferenzreihenfolge der Computer, zu denen es sich bewegt, definiert.

5. System nach einem vorhergehenden Anspruch, bei dem das erste Programm einen Überwachungscode zum Überwachen des Status des zweiten Computers umfaßt.

6. System nach Anspruch 5 wenn abhängig von Anspruch 2, bei dem das vorgegebene Kriterium eine Verringerung der Rechenkapazität des zweiten Computers ist.

7. System nach Anspruch 5, bei dem das erste Programm so beschaffen ist, daß es das oder jedes zweite Programm in Abhängigkeit von der Überwachung steuert.

8. System nach Anspruch 7, bei dem das erste Programm so beschaffen ist, daß es die Anzahl der zweiten Programme in Abhängigkeit von der Überwachung steuert.

9. System nach einem vorhergehenden Anspruch, bei dem das zweite Programm einen Code zum Übertragen des ersten Programms an einen weiteren solchen Computer in Reaktion auf einen Bewegungsbefehl vom ersten Programm umfaßt, wobei das erste Programm so beschaffen ist, daß es einen solchen Bewegungsbefehl überträgt.

10. System nach Anspruch 9, wenn abhängig von Anspruch 2, bei dem das erste Programm so beschaffen ist, daß es einen solchen Bewegungsbefehl in Reaktion auf die Überwachung überträgt.

11. System nach einem vorhergehenden Anspruch, bei dem das erste Programm so beschaffen ist, daß es das oder jedes zweite Programm von dem zweiten Computer entfernen und seine Ausführung beenden kann.

12. System nach Anspruch 11, bei dem die zweiten Programme jeweils einen Code umfassen, um den zweiten Computer dazu zu veranlassen, sie zu entfernen und zu beenden, und so beschaffen sind, daß sie dies bei Abwesenheit eines Signals vom ersten Programm unter vorgegebenen Bedingungen tun.

13. System nach einem vorhergehenden Anspruch, bei dem der erste Computer so programmiert ist, daß er auf mehrere zweite Computer zugreift; für jeden hiervon bestimmt, ob er das Computer-Team unterstützt, und in dem Fall, in dem ein zweiter Com-

puter das Team nicht unterstützt, an diesen ein Support-Programm überträgt und dessen Ausführung darauf veranlaßt, um den zweiten Computer für die Unterstützung der Teams geeignet zu machen.

14. System nach einem vorhergehenden Anspruch, bei dem der erste Computer so programmiert ist, daß er an mehrere der zweiten Computer über die Verbindung Daten überträgt, die ein Überwachungsprogramm definieren, das einen Überwachungscode zum Überwachen eines entsprechenden der zweiten Computer und einen Code für die Kommunikation mit dem ersten Computer enthält; und der erste Computer so beschaffen ist, daß er Statusdaten von dem oder jedem Überwachungsprogramm empfängt und die Operation des oder jedes Teams in Abhängigkeit hiervon steuert.

15. System nach Anspruch 14, wenn abhängig von Anspruch 3 oder Anspruch 4, bei dem der erste Computer so beschaffen ist, daß er in Abhängigkeit von den Überwachungsdaten Computerauswahldaten an das erste Programm übermittelt.

16. System nach Anspruch 3 oder Anspruch 14, bei dem der Überwachungscode der Überwachung des Speichers des zweiten Computers dient.

17. System nach Anspruch 3 oder Anspruch 14, bei dem der Überwachungscode der Überwachung der Nutzung des Prozessors des zweiten Computers dient.

18. System nach Anspruch 3 oder Anspruch 14, bei dem der Überwachungscode der Überwachung der Speicherkapazität des zweiten Computers dient.

19. System nach Anspruch 3 oder Anspruch 14, bei dem der Überwachungscode der Überwachung der Nutzung einer Eingabevorrichtung des zweiten Computers dient.

20. System nach Anspruch 3 oder Anspruch 14, bei dem der Überwachungscode der Überwachung einer Batterie des zweiten Computers dient.

21. Ferndatenverarbeitungssystem nach einem der Ansprüche 1 bis 20, bei dem der erste Computer ferner so beschaffen ist, daß er feststellt, ob der zweite Computer die Ausführung eines solchen mobilen Programms unterstützen kann und, falls nicht, so beschaffen ist, daß er an den zweiten Computer über die Verbindung Daten überträgt, die eine Supportumgebung für das mobile Programm definieren, die den zweiten Computer in die Lage versetzt, die Daten des mobilen Programms zu empfangen und das mobile Programm auszuführen, und deren Ausführung auf dem zweiten Computer veranlaßt.

22. System nach Anspruch 21, bei dem der erste

Computer so beschaffen ist, daß er den zweiten Computer dazu veranlassen kann, die Support-Umgebung für das mobile Programm zu entfernen.

23. Ferndatenverarbeitungssystem nach einem der Ansprüche 1 bis 22, bei dem der erste Computer ferner so beschaffen ist, daß er eine graphische Anzeige erzeugt, die den oder jeden zweiten Computer und das oder jedes mobile Programm, die darauf resident vorhanden sind, zeigt.

24. System nach Anspruch 23, bei dem die graphische Anzeige die geographischen Positionen der zweiten Computer angibt.

25. System nach Anspruch 23, bei dem die graphische Anzeige die Rechenkapazitäten der zweiten Computer angibt.

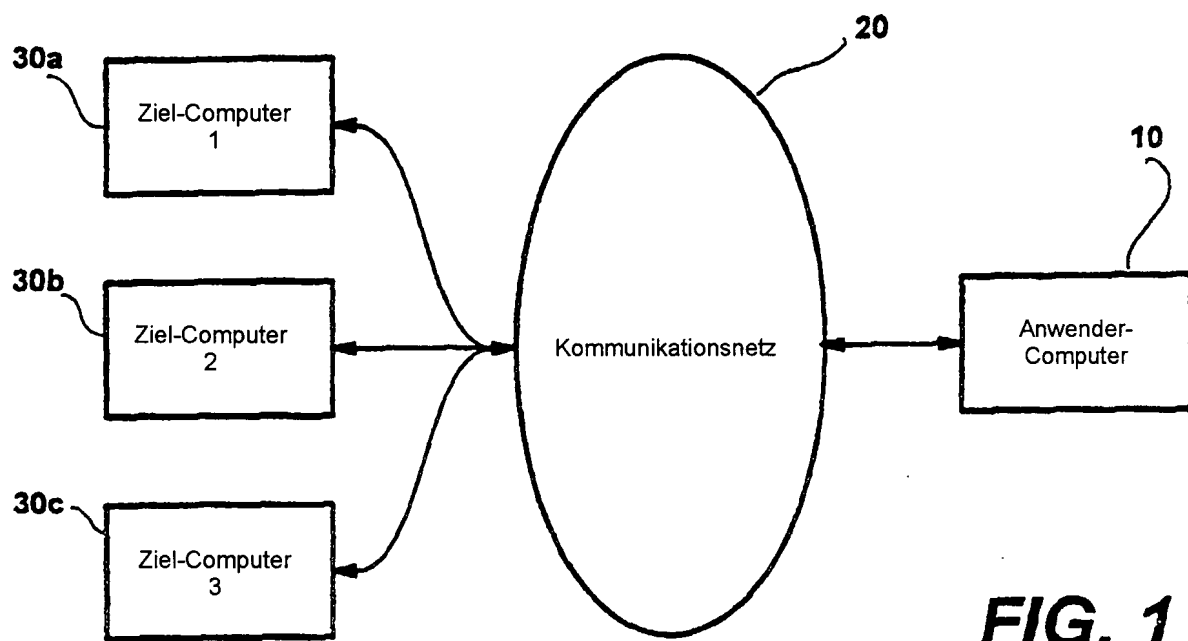
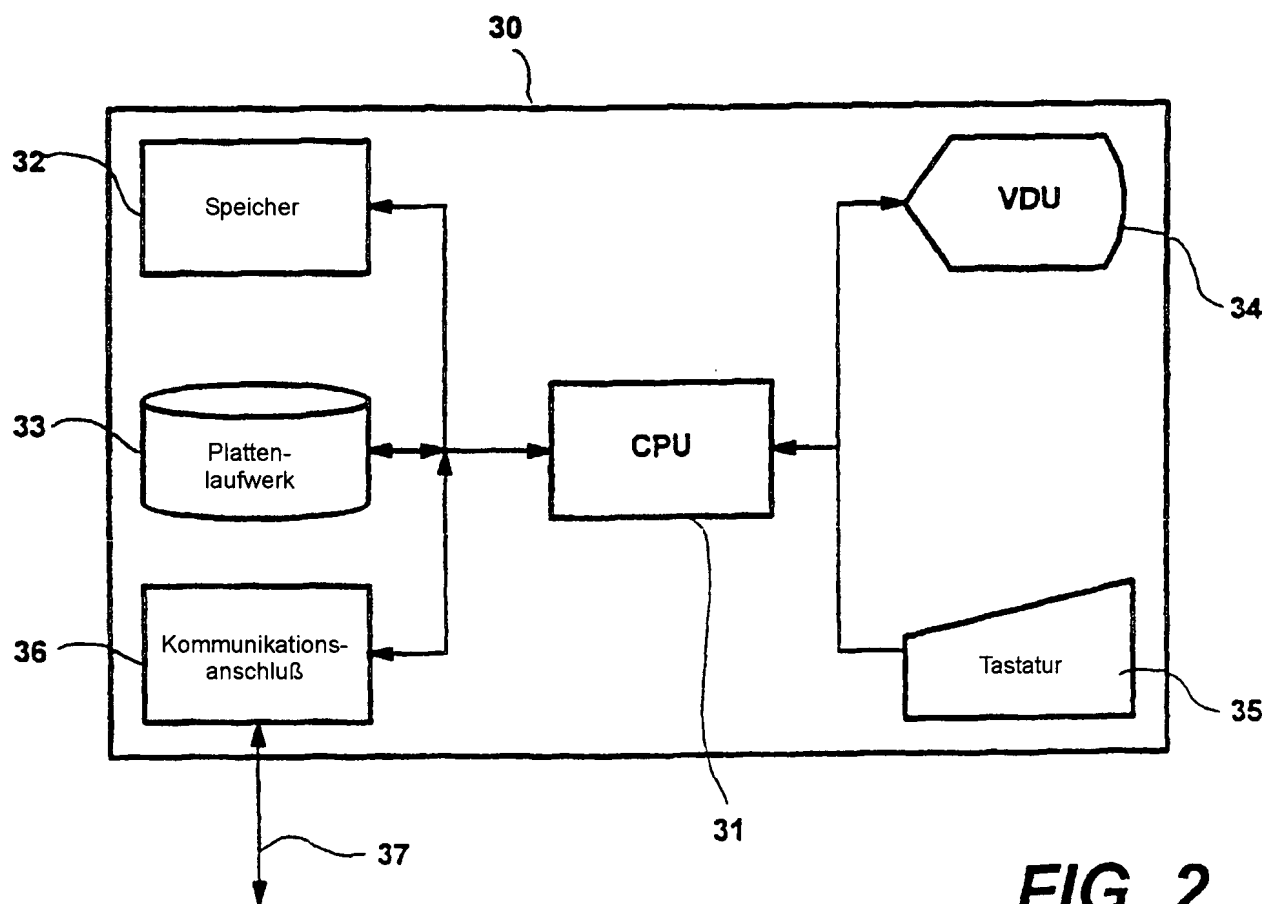
26. System nach Anspruch 23, bei dem die graphische Anzeige die Verzögerung der Kommunikation von dem ersten zu den zweiten Computern angibt.

27. System nach Anspruch 23, bei dem der erste Computer so beschaffen ist, daß er eine Anzeige auf dem oder jedem zweiten Computer, die das Vorhandensein des mobilen Programms darauf angibt, bewirkt.

28. Erster Computer, der so beschaffen ist, daß er in dem System nach einem vorhergehenden Anspruch arbeitet.

29. Ferndatenverarbeitungsverfahren, das das Liefern mehrerer paralleler Verarbeitungsaufgaben-Programme von einem ersten Computer (10) zu wenigstens einem zweiten Computer (30) umfaßt, gekennzeichnet durch ein Koordinierungsprogramm (42), das von dem ersten Computer (10) zu dem zweiten Computer (30) geliefert wird und die Operation für die Aufgaben-Programme durch die Ausführung des Koordinierungsprogramms (42) parallel zu der Ausführung der mehreren parallel verarbeiteten Aufgaben-Programme koordiniert, wobei das Koordinierungsprogramm mit den mehreren parallel verarbeiteten Aufgaben-Programmen und mit dem ersten Computer kommuniziert.

Es folgen 11 Blatt Zeichnungen

**FIG. 1****FIG. 2**

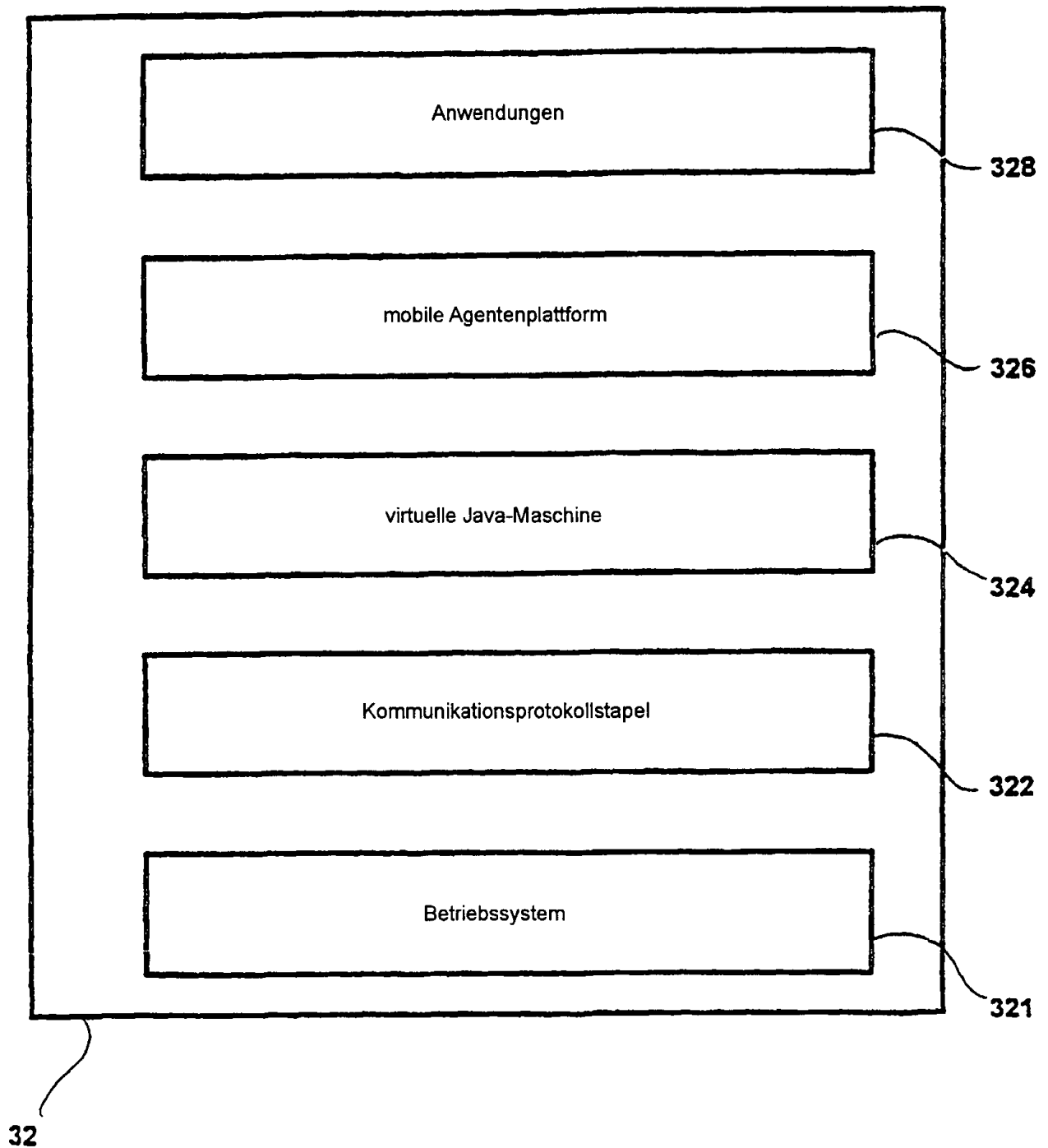


FIG. 3

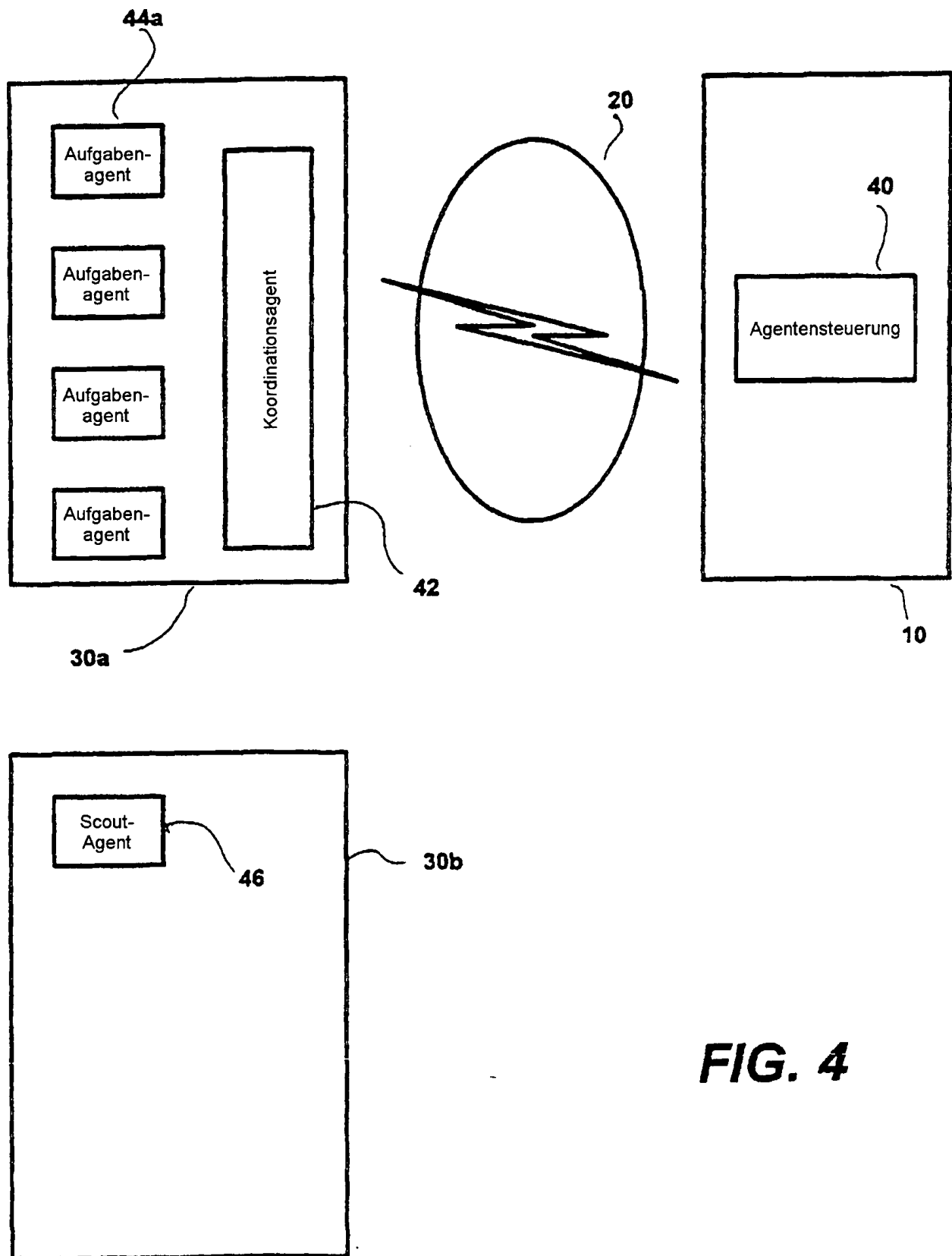
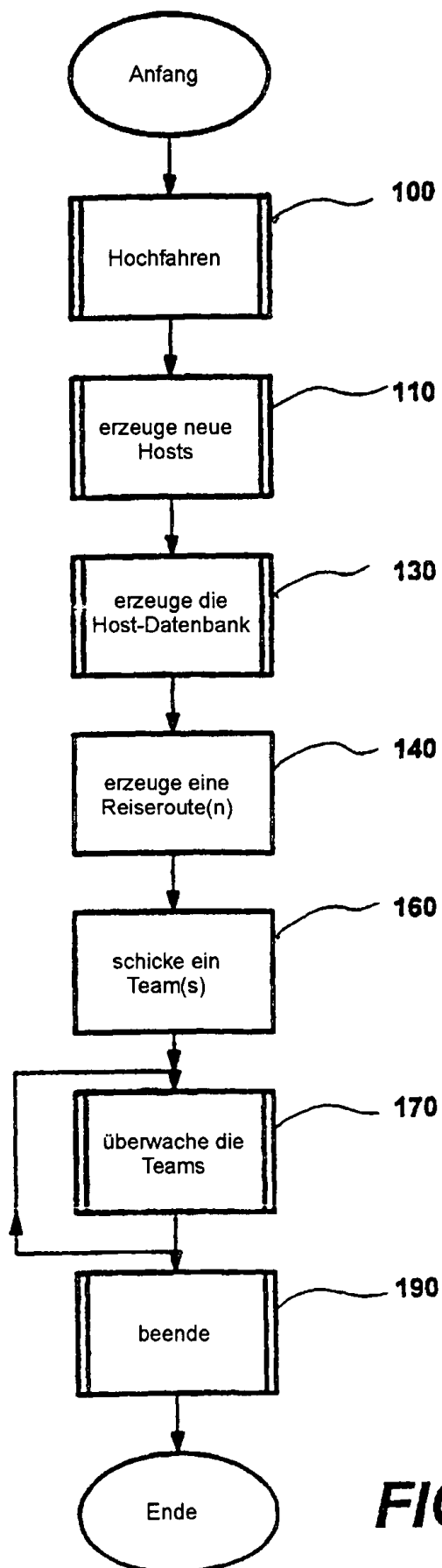
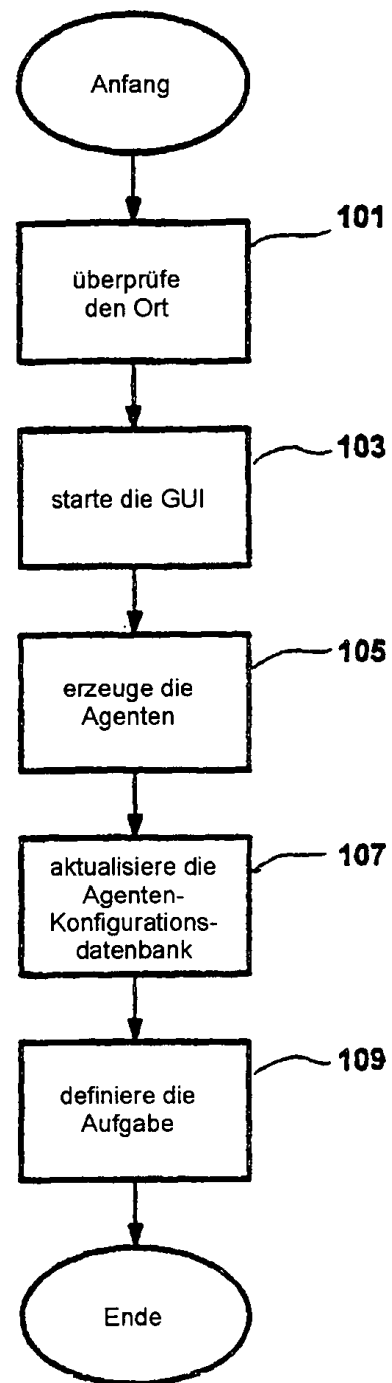
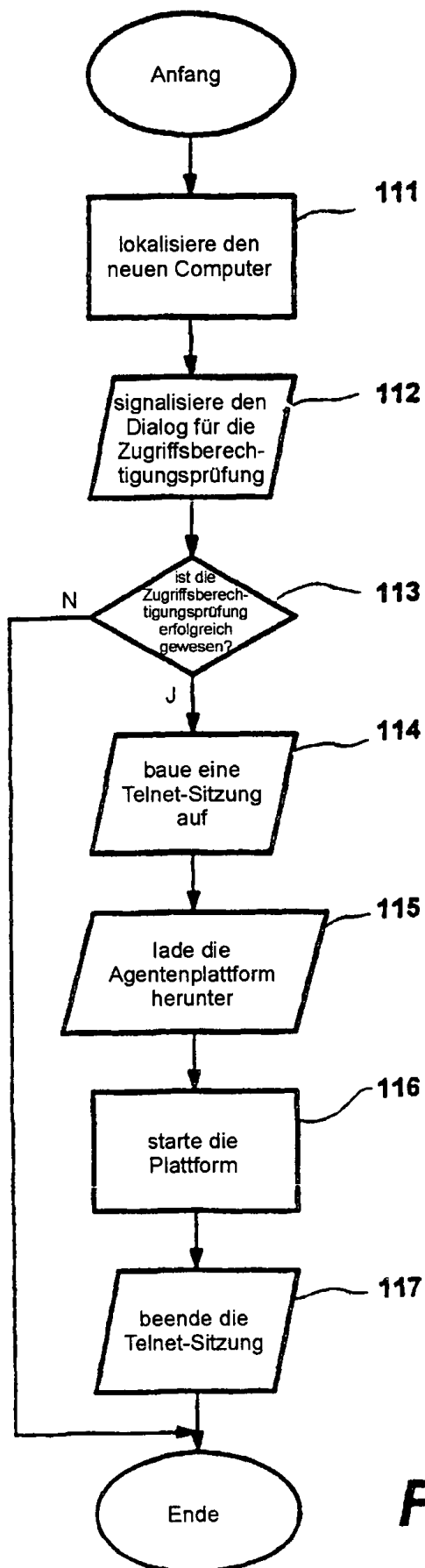
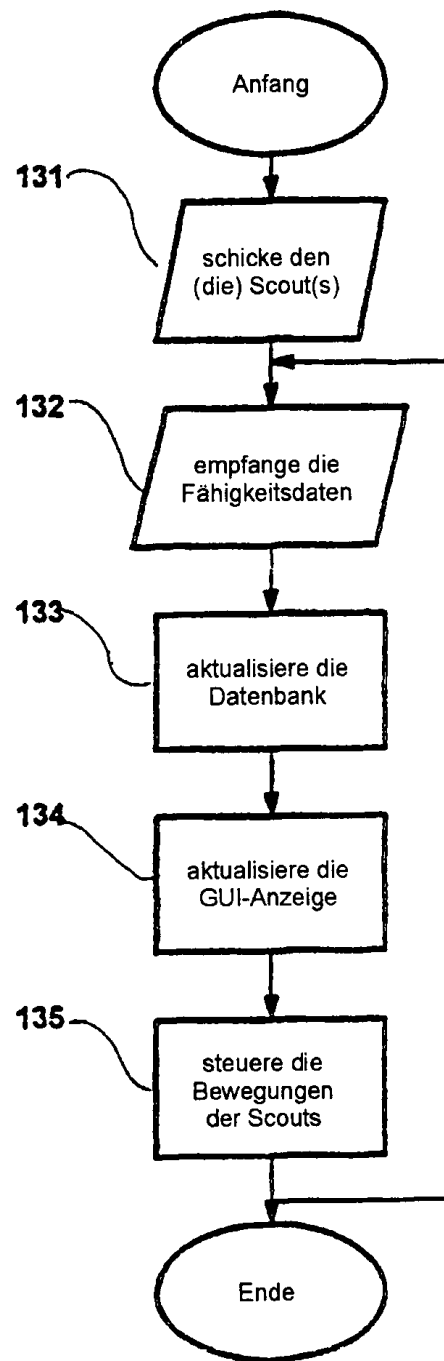
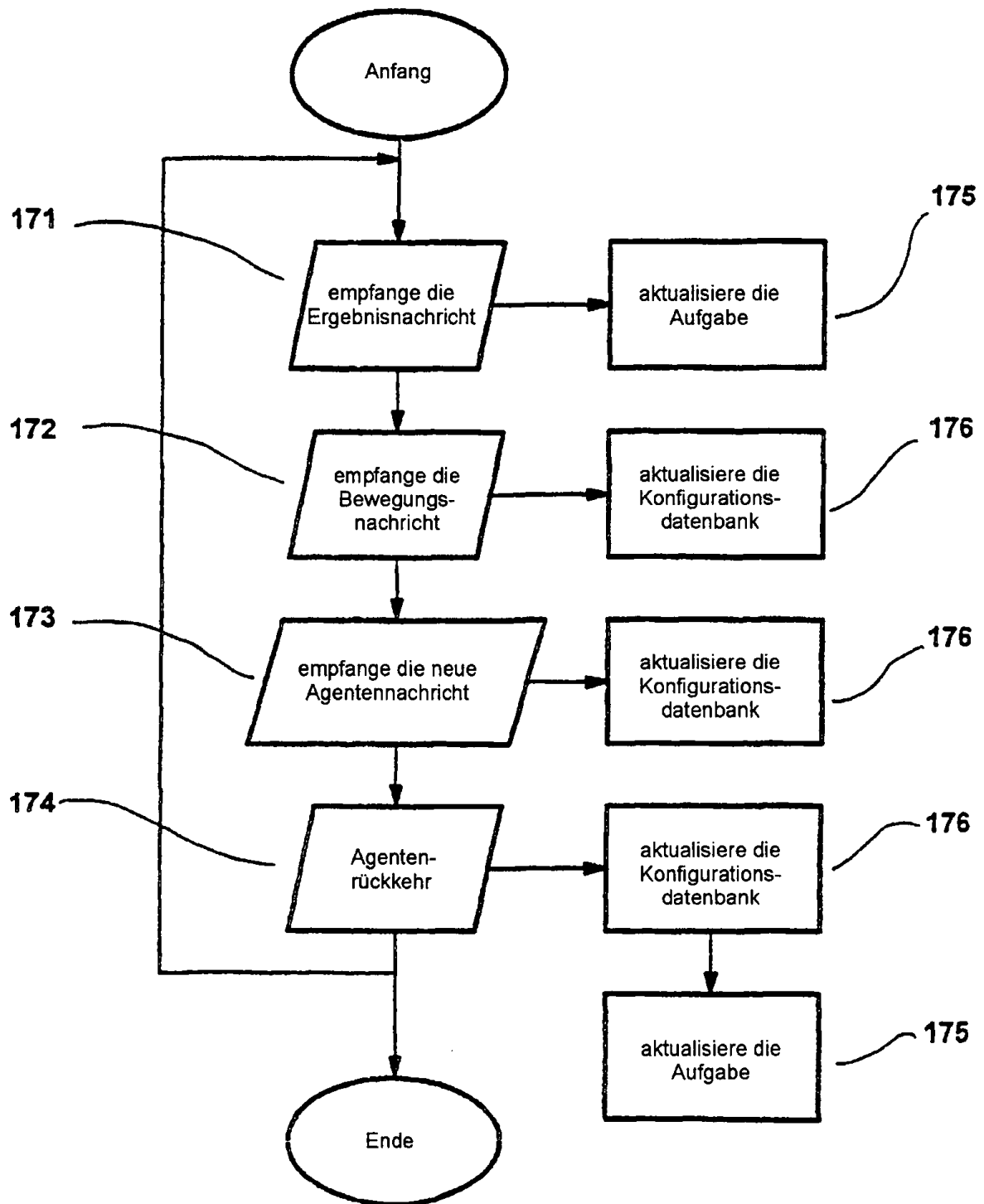
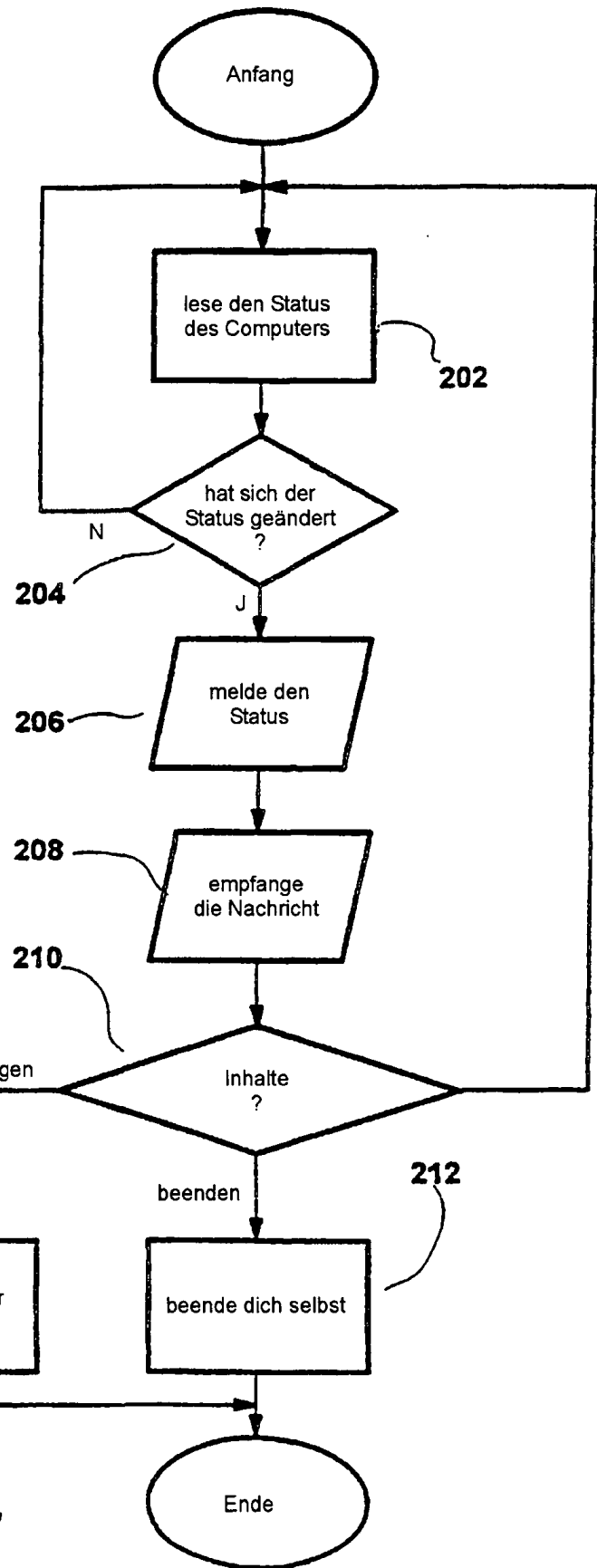
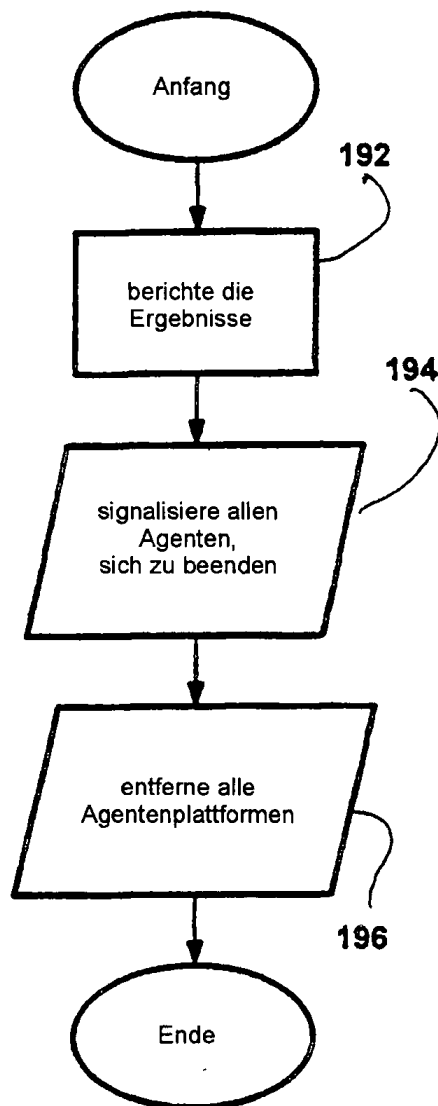


FIG. 4

**FIG. 5****FIG. 6a**

**FIG. 6b****FIG. 6c**

**FIG. 6d**



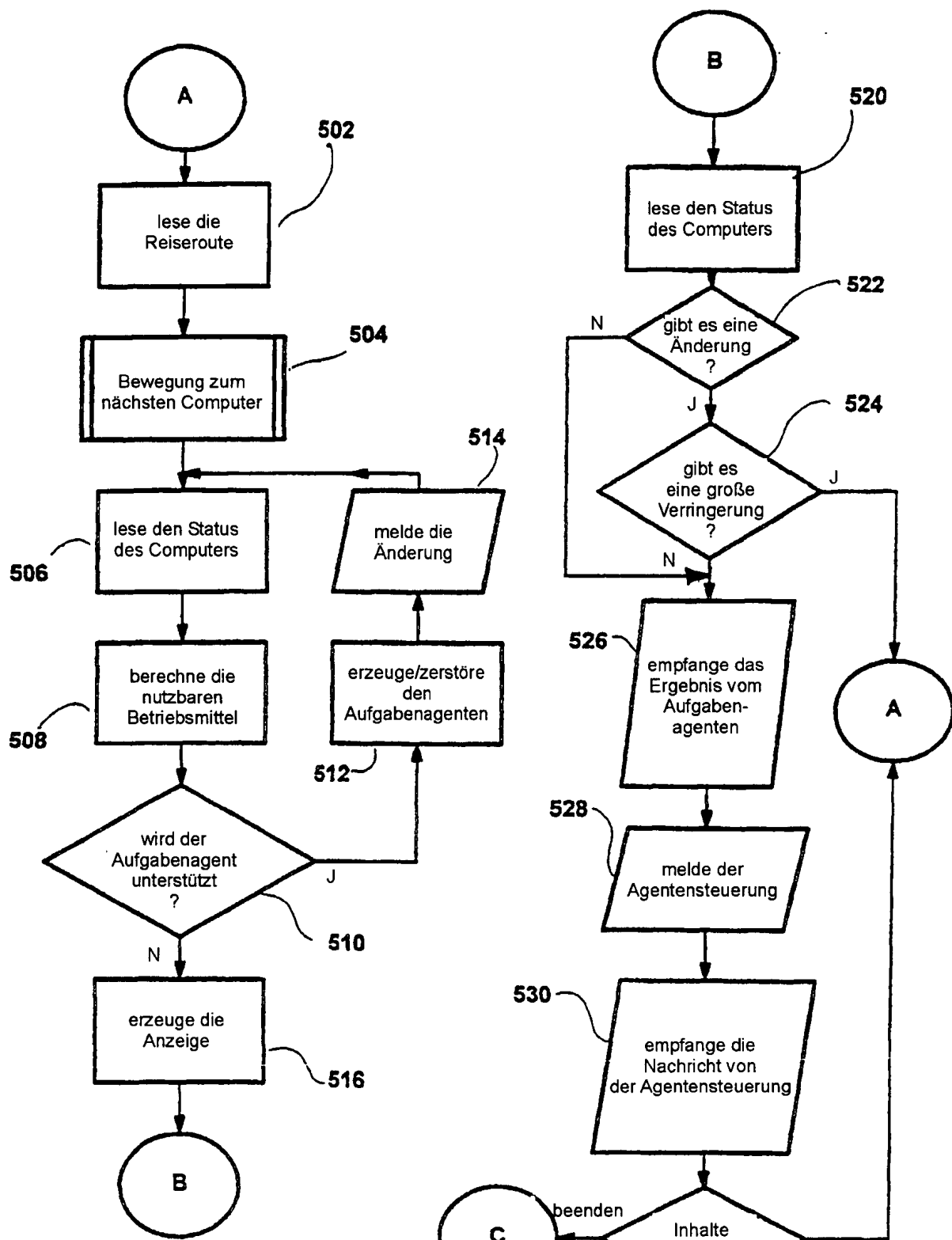
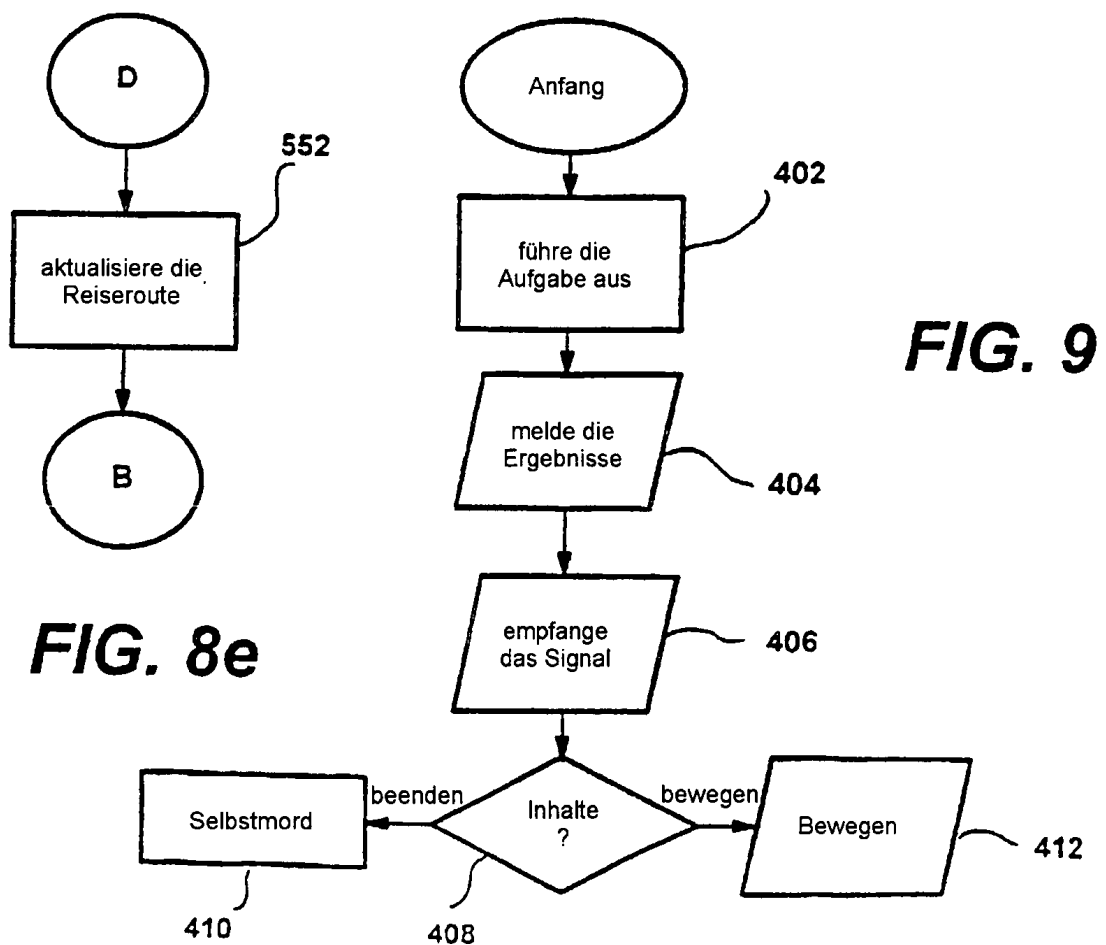
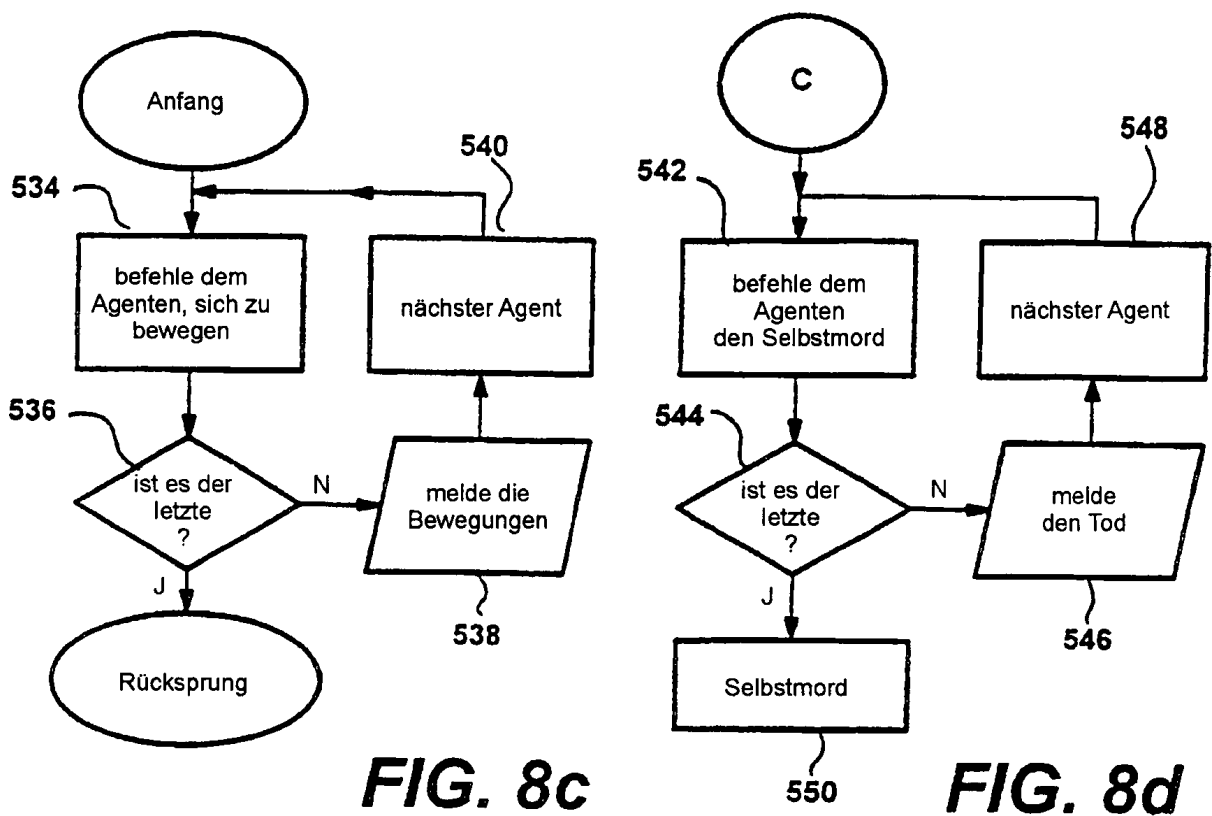


FIG. 8a

FIG. 8b



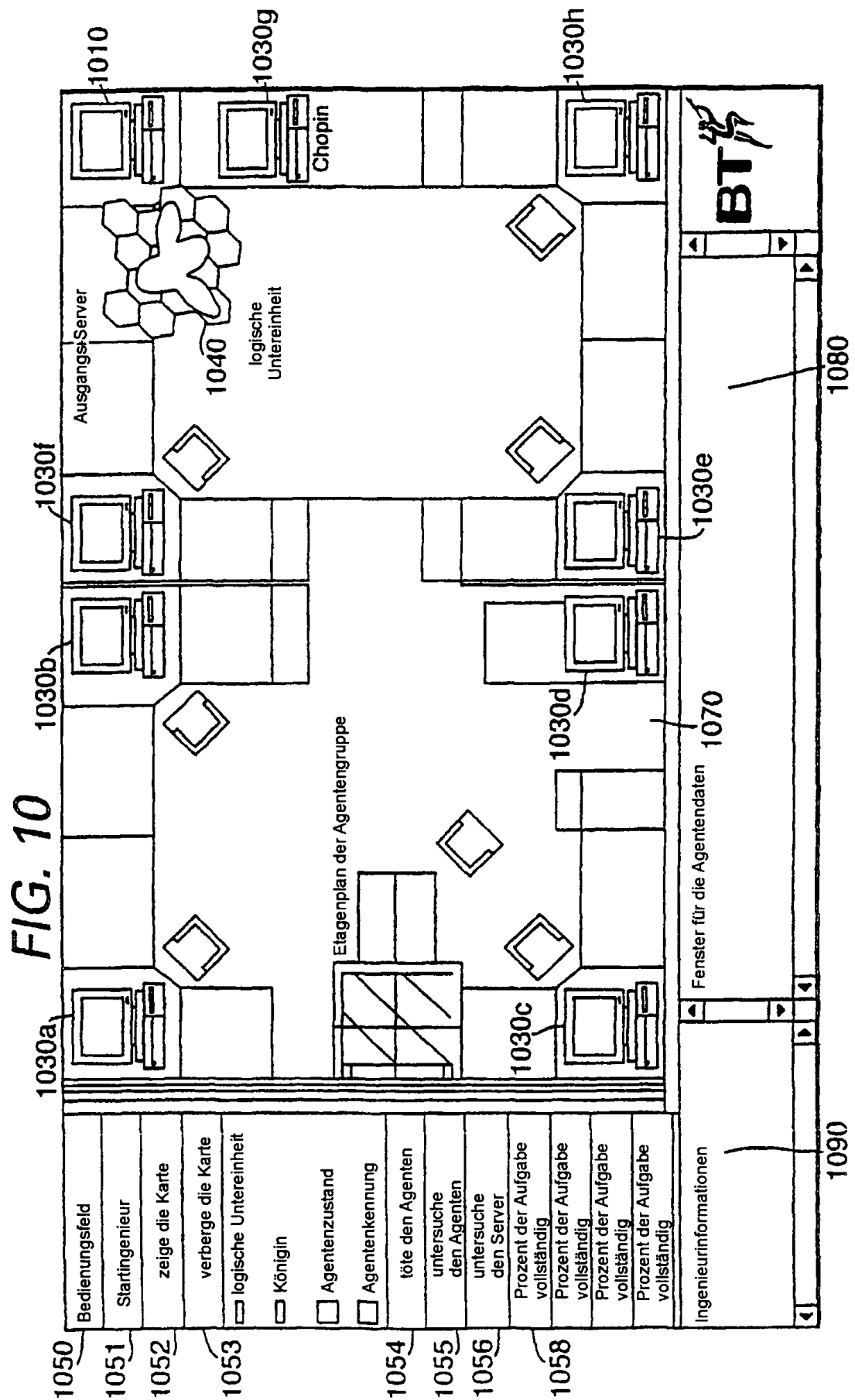


FIG. 11

