



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 603 17 917 T2 2008.11.27**

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 552 428 B1

(51) Int Cl.⁸: **G06F 17/30 (2006.01)**

(21) Deutsches Aktenzeichen: **603 17 917.7**

(86) PCT-Aktenzeichen: **PCT/GB03/04201**

(96) Europäisches Aktenzeichen: **03 750 989.0**

(87) PCT-Veröffentlichungs-Nr.: **WO 2004/031882**

(86) PCT-Anmeldetag: **01.10.2003**

(87) Veröffentlichungstag

der PCT-Anmeldung: **15.04.2004**

(97) Erstveröffentlichung durch das EPA: **13.07.2005**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **05.12.2007**

(47) Veröffentlichungstag im Patentblatt: **27.11.2008**

(30) Unionspriorität:

2406713 04.10.2002 CA

(84) Benannte Vertragsstaaten:

AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LI, LU, MC, NL, PT, RO, SE, SI, SK, TR

(73) Patentinhaber:

International Business Machines Corp., Armonk, N.Y., US

(72) Erfinder:

NG, Joanna, Unionville, Ontario L3R 9C7, CA; KAPOOR, Rohit, Richmond Hill, Ontario L3B 4E1, CA; NIGUL, Leho, Toronto, Ontario M2R 3V4, CA

(74) Vertreter:

Duscher, R., Dipl.-Phys. Dr.rer.nat., Pat.-Anw., 70176 Stuttgart

(54) Bezeichnung: **VERFAHREN UND VORRICHTUNG ZUM WEITERLEITEN VON SITZUNGSGEDECKEN VON EINEM PORTAL-SERVER**

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 eingereicht, 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**Gebiet der Erfindung**

[0001] Diese Erfindung betrifft das Internet und insbesondere Verfahren und Vorrichtungen zum Erzeugen und Verwenden von Portalen und Portlets in Netzanwendungen, um bessere Einsatzmöglichkeiten für Websites zu schaffen.

Hintergrund der Erfindung

[0002] Das World Wide Web brachte eine Paradigmenverschiebung zu Datenübertragungen über das Internet mit sich, wobei grafische Informationen zu Benutzern befördert werden. Mit dem Aufkommen des Web gab es eine Forderung nach einer Verbesserung der Kommunikationsfähigkeit und der breiten Verbindungsähigkeit, wobei diese Forderung immer noch besteht.

[0003] Das Portal (früher als Netzportal bekannt) brachte eine wesentliche Paradigmenverschiebung im Internetraum mit sich. Eine Website, die eine große Anzahl von Ressourcen oder Diensten wie z. B. eMail, Foren, Suchmaschinen, Datenbanken oder andere Informationen anbietet, kann als ein Portal betrachtet werden. Die ersten Netzportale waren Online-Dienste. Zum ersten Mal konnten Benutzer, die im Internet surften, Webseiten sehen, die mit Informationen assembliert wurden und diese anboten, die von verschiedenen Sites im World Wide Web kamen, die Beschaffenheit der Verknüpfung war jedoch für den Benutzer transparent. Ein Benutzer, der einen typischen Webbrowser verwendet, sieht eine zusammenhängende Webseite, die angezeigt wird. Die Herkunft von verschiedenen Teilen der Seite von verschiedenen Internetsites, die nicht mit der Website, die gerade betrachtet wird, verbunden sind, ist nicht in einfacher Weise offensichtlich. Diese Teile werden als Portlets bezeichnet.

[0004] Portlets sind die sichtbaren aktiven Komponenten, die Endbenutzer in ihren Portalseiten sehen. Ähnlich wie ein Fenster in einem PC-Desktop "besitzt" jedes Portlet einen Abschnitt des Browser-Bildschirms oder des Bildschirms des persönlichen digitalen Geräts, wo es Ergebnisse anzeigt.

[0005] Aus Sicht des Benutzers ist ein Portlet ein Inhaltkanal oder eine Anwendung, bei dem bzw. der sich ein Benutzer anmeldet, den bzw. die er seiner persönlichen Portalseite hinzufügt und so konfiguriert, dass personalisierter Inhalt angezeigt wird.

[0006] Aus Sicht der Inhaltanbieter ist ein Portlet ein Mittel, um ihre Inhalte zur Verfügung zu stellen.

[0007] Aus Sicht eines Portaladministrators ist ein Portlet ein Inhaltbehälter, der bei dem Portal registriert sein kann, so dass sich Benutzer bei ihm anmelden können.

[0008] Vom Standpunkt eines Portals ist ein Portlet eine Komponente, die in eine seiner Seiten aufgenommen (rendered) wurde.

[0009] Vom technischen Standpunkt ist ein Portlet ein Stück eines Codes oder eine kleine Anwendung, die auf einem Portalserver läuft und Inhalt bereitstellt, der in Portalseiten eingebettet werden soll. In der einfachsten Form kann ein Portlet ein Java™-Servlet sein, das innerhalb eines Portals betrieben wird.

[0010] Jeder Teil (Portlet) einer vorgegebenen Seite (der typischerweise von unterschiedlichen Stellen im World Wide Web stammt) kann mit jedem anderen Teil (Portlet) der gleichen Seite zusammenwirken, um eine höhere Funktion für einen Benutzer zu erreichen, der auf der Seite surft oder auf diese zugreift. Dadurch wird ein Portal der einzige Zugangspunkt für eine Vielzahl von Benutzern, um über eine Vielzahl von Kanälen auf eine Vielzahl von Informationsquellen zuzugreifen.

[0011] Portale können in verschiedenen Geschäftsmodellen angewendet werden und zwar: Geschäfte mit Kunden, Geschäfte mit Firmen oder Geschäfte mit Unternehmen. Der Schlüssel zu einer schnellen Anpassung des Portalparadigmas hängt stark von seiner Fähigkeit ab, vorhandene Netzanwendungsdaten nahtlos in das Portalgerüst zu integrieren.

[0012] Es bestehen jedoch trotzdem verschiedene technische Hürden für eine derartige nahtlose Integration von Netzanwendungen in ein Portal.

[0013] Wenn Benutzer auf eine Portalseite zugreifen, wird die ursprüngliche http-Anforderung für jeden Benutzer zum Portalserver geschickt. Jedes der Portlets besitzt seine eigene unabhängige Sitzung, die als Portlet-Sitzung bezeichnet wird. Wenn ein Portlet Informationen aufnehmen muss, die von einer vorgegebenen Netzanwendung kommen, gibt es keinen Mechanismus, um diese Vielzahl von http-Anforderungen, die von verschiedenen Portlets an eine vorgegebene Netzanwendung erzeugt wurden, vom Standpunkt der Netzanwendung als eine zusammenhängende http-Sitzung zu führen. Darüber hinaus gibt es keinen Mechanismus, um Sitzungsinformationen zwischen der Vielzahl von Portlet-Sitzungen und der Netzanwendungssitzung weiterzuleiten. Sitzungsinformationen von den Portlets müssen zur Netzanwendung weitergeleitet werden, damit der Aufnahmevergäng von der Netzanwendung korrekt ausgeführt werden kann. Zu Beispielen von wichtigen Sitzungsinformationen, die zum Weiterleiten von Portlet-Sitzungen an die Netzanwendung erforderlich sind, gehören lokale Informationen, Benutzeragent- und Sitzungszeitüberschreitungs-Informationen.

[0014] Es gibt nach dem Stand der Technik Einschränkungen in Bezug auf die Art und Weise, wie die folgenden Portalartifakte mit vorhandenen Netzanwendungen zusammenarbeiten. Die Realisierung der Integration von Netzanwendungen in die Portalarchitektur ist nicht eindeutig definiert. Diese Entitäten enthalten:

- ursprüngliche http-Anforderung an ein Portal;
- eine Portlet-Sitzung innerhalb eines Portals;
- eine http-Anforderung von dem Portal an die entsprechende Netzanwendung.

[0015] Wenn verschiedene Benutzer auf eine Portalseite zugreifen, wird die ursprüngliche http-Anforderung für jeden Benutzer zum Portalserver geleitet (a). Die ursprüngliche http-Sitzung für jeden Benutzer ist außerdem im vollständigen "Besitz" des Portalservers. Jedes der Portlets besitzt seine eigene unabhängige Sitzung, die als Portlet-Sitzung bezeichnet wird. Wenn ein Portlet Informationen aufnehmen muss, die von einer vorgegebenen Netzanwendung kommen (b), gibt es typischerweise die folgenden technischen Hindernisse:

- i. Es gibt keinen Mechanismus für ein Portlet, um für die Back-End-Netzanwendung und von dieser http-Anforderungen und Antworten zu erzeugen.
- j. Es gibt keinen Mechanismus, um mehrere Anforderungen und Antworten für ein anrufendes Portlet (und die Portlet-Sitzung) zu verwalten, die mit mehreren Anforderungen und Antworten für eine Back-End-Netzanwendung (und die Sitzung der Netzanwendung) korrekt übereinstimmen. Jeder (sowohl das Portlet als auch die Netzanwendung) führt seine Benutzersitzung dementsprechend.

[0016] Das wird dann kompliziert, wenn mehrere Portlets die gleiche Netzanwendung aufrufen, wobei die Netzanwendung diese mehreren Portlet-Anforderungen innerhalb der gleichen Sitzung der Netzanwendung abwickelt.

- k. Es gibt keinen Mechanismus, um Sitzungsinformationen zwischen den mehreren Portlet-Sitzungen und der Sitzung der Netzanwendung weiterzuleiten.

[0017] Wenn eine eindeutig definierte Menge von Portlets innerhalb der gleichen Portlet-Anwendung mit der einen nachgeschalteten Netzanwendung (web application at the backend) zusammenwirkt, müssen alle beteiligten Portlets in der Lage sein, die korrekten Sitzungsinformationen abzurufen und zur nachgeschalteten Netzanwendung weiterzuleiten, so dass die von der Netzanwendung aufgenommenen Informationen mit den Einstellungen der Informationen des Portals der Portlets konsistent sind. Beispiele solcher Einstellungen sind lokale Informationen, Benutzeragenten dieses speziellen Zugriffs usw. Die Antworten, die von der Netzanwendung gesendet werden, müssen z. B. die gleichen lokalen Informationen verwenden wie das Portlet in dem Portalserver, der es anzeigt.

[0018] Es gibt keinen Mechanismus für eine Einmalanmeldung, bei dem die Berechtigungsnachweise des Portalbenutzers nicht durch die Back-End-Netzanwendung abgefragt werden. Dies ist eine wesentliche Forderung. Ihr Fehlen hat zur Folge, dass die Berechtigungsnachweise des Benutzers abgerufen werden, wenn sich der Benutzer von einem Teil einer Webseite zu einem anderen Teil der gleichen Webseite bewegt, da die Portlets unterschiedliche Entstehungsorte und Identifizierungsanforderungen besitzen.

[0019] Es gibt keinen Mechanismus für eine Synchronisation mehrerer Anforderungen oder Antworten zwischen Portlets einer vorgegebenen Portlet-Anwendung und der vorhandenen Back-End-Netzanwendung.

[0020] Nach dem Stand der Technik bestehen Einschränkungen, da nicht definiert ist, wie eine Vielzahl von Portlets (die den gleichen Kontext gemeinsam verwenden) innerhalb der gleichen Portlet-Anwendung untereinander sowie mit den verschiedenen integrierten Netzanwendungen dynamisch zusammenarbeiten können.

[0021] Ein Benutzerszenario, das eine Vielzahl von Portlets enthält, die zusammenarbeiten, indem sie den

gleichen "Kontext" gemeinsam verwenden, dient dazu, die Einschränkung konzeptionell zu erläutern:
Bei drei Portlets, die auf der gleichen Portal-Webseite angezeigt werden,

- zeigt ein Portlet die Kontozusammenfassung durch Anzeigen einer Liste von Konten;
- zeigt das zweite Portlet eine Liste von offenen Rechnungen eines bestimmten Kontos und
- zeigt das dritte Portlet eine Zusammenfassung der Auftragsgeschichte des bestimmten Kontos.

[0022] Das zweite und das dritte Portlet sind kontextmäßig mit dem ersten Portlet dynamisch verbunden, indem sie offene Rechnungen (zweites Portlet) und die Auftragsgeschichte (drittes Portlet) wiedergeben, und sie sind mit einem Konto synchronisiert, das aus der Kontoliste des ersten Portlet ausgewählt ist.

[0023] Einschränkungen, die nach dem Stand der Technik bestehen:

- i. Es gibt keinen Mechanismus, um eine Untergruppierung von Portlets innerhalb einer Portlet-Anwendung zu definieren, die gemeinschaftlich arbeiten würden.
- j. Es gibt keinen Mechanismus, um einen Kontext zu definieren (der dynamisch geändert werden kann), der von dieser Untergruppe von Portlets innerhalb einer vorgegebenen Portlet-Anwendung gemeinsam verwendet wird: ein Beispiel eines Kontexts ist hier das ausgewählte Konto im Portlet 1, wobei diese Kontoauswahl dynamisch geändert werden kann.
- k. Es gibt keinen Mechanismus, um die Änderung im Kontext dynamisch zu erfassen: ein Beispiel ist die Änderung der Auswahl von einem Konto zum anderen Konto aus der Kontoliste im Portlet 1 des oben genannten Beispiels.
- l. es gibt keinen Mechanismus, um eine im Voraus definierte Aktion (oder Antworten) für alle beteiligten Portlets in der Untergruppe von Portlets, die den gleichen Kontext gemeinsam verwenden: das Beispiel der Anzeige der Liste offener Rechnungen (Aktion im Portlet 2), wenn der Kontext geändert wird (von einer Kontoauswahl zu einer anderen im Portlet 1).
- m. Es gibt keinen Mechanismus, um diesen dynamischen Kontext zu den relevanten integrierten Netzanwendungen weiterzuleiten.

[0024] Nach dem Stand der Technik gibt es keinen Mechanismus, um eine Auffrischungssequenz für eine Gruppe von Portlets innerhalb einer Portlet-Anwendung zu definieren.

- i. Es gibt gegenwärtig keine Vorkehrung für einen Portalentwickler, die Auffrischreihenfolge einer vorgegebenen Gruppe von Portlets, die angezeigt werden, festzulegen.

[0025] In unserem oben genannten Szenario würde der Portalentwickler wünschen, dass das erste Portlet (Kontoliste) zuerst aufgefrischt wird, das zweite Portlet als zweites aufgefrischt wird usw., so dass das zweite und das dritte Portlet automatisch definierte Aktionen haben (wenn das Portlet angezeigt wird), die in einer korrekten Sequenz erfolgen.

[0026] Es fehlt ein eindeutig definierter Mechanismus in der Portalarchitektur, um die Verknüpfung von Portlets anhand von Geschäftsregeln und Benutzerprofilierungsinformationen, die die Rolle des Benutzers enthalten, zu unterstützen.

- i. Es gibt keinen Mechanismus, um eine Verknüpfung von Portalressourcen für jeden Benutzer anhand von Geschäftsregeln zu definieren.
Beispiel: Alle Portalbenutzer im Jugendalter sehen eine Gruppe von Portlets, und alle älteren Portalbenutzer sehen eine andere Gruppe von Portlets.
- j. Es gibt keinen Mechanismus für eine derartige regelbasierte und benutzerbasierte Verknüpfung von Portlets, die zur Laufzeit dynamisch ausgeführt wird.

[0027] Es gibt keine gemeinsame Verwendung von Geschäftsregeln auf Portalebene und von Benutzerprofilinformationen mit entsprechenden Back-End-Netzanwendungen.

[0028] Es gibt keine gemeinsame Verwendung von Geschäftsregeln oder Benutzersegmentierungsinformationen bei einer integrierten Netzanwendung, so dass diese Regeln und die Benutzersegmentierung innerhalb eines Portals und seiner integrierten Back-End-Netzanwendung konsistent sein können. Wenn es z. B. eine Regel gibt, die den Altersbereich eines Jugendlichen definiert, sollte eine derartige Regel sichtbar und für eine Konsistenz auf die integrierte Netzanwendung anwendbar sein.

[0029] Ein Artikel mit dem Titel "Developing Portlets", BEA WEBLOGIC PERSONALIZATION SERVER 3.1.1, [Online] 2000, XP002314066 BEA HOMEPAGE beschreibt die Erzeugung von Portlet-Anwendungen.

Terminologie

Portlets

[0030] Portlets sind die sichtbaren aktiven Komponenten, die die Endbenutzer innerhalb ihrer Webseiten des Portals sehen. Ähnlich wie ein Fenster in einem PC-Desktop "besitzt" jedes Portlet einen Abschnitt des Bildschirms des Browsers oder des PDA (persönliches digitales Gerät), in dem es portlet-spezifische Informationen anzeigt.

Portlet-Anwendung

[0031] Portlets können außerdem in einer Portlet-Anwendung gemeinsam gruppiert werden. Portlet-Anwendungen werden unter Verwendung von Webarchivdateien (WAR) verteilt und verwendet. Es gibt Portlet-spezifische Erweiterungen für den Verwendungsdeskriptor der Standard-Netzanwendung.

Portlet-Nachrichten

[0032] Portlet-Nachrichten werden für die Datenübertragung zwischen zwei Portlets unter Verwendung von Portlet-Aktionen und Portlet-Nachrichten verwendet. Das sendende Portlet erzeugt eine Portlet-Aktion, wobei es die Aktion in eine URL codiert.

[0033] Wenn die URL adressiert wird, z. B. durch einen Benutzer, der versucht, eine Aufgabe auszuführen, wird die Aktionsabreinrichtung (action listener) aufgerufen und sendet eine Portlet-Nachricht, um die erforderlichen Daten zu senden.

Portlet-Sitzung

[0034] Eine Portlet-Sitzung wird für jedes Portlet für jeden Benutzer, der sich anmeldet, erzeugt, um Sitzungsinformationen für jeden Benutzer und jedes Portlet zu führen.

Zusammenfassung der Erfindung

[0035] Die verschiedenen Ausführungsformen der vorliegenden Erfindung sind auf ein oder mehrere Nachteile, die nach dem Stand der Technik vorhanden sind, gerichtet.

[0036] Die Erfindung stellt ein Verfahren bereit zum Weiterleiten von Sitzungsinformationen von einem Portal zu zugehörigen Portlets zu Back-End-Netzanwendungen. Es ermöglicht das Weiterleiten von Sitzungsinformationen vom Portal zur Back-End-Netzanwendung, wodurch sich die Netzanwendung gemäß den durch das Portal bereitgestellten Sitzungsinformationen konsistent verhalten kann.

[0037] Eine Ausführungsform der vorliegenden Erfindung stellt eine Benutzer-Sitzungsinformationen-Speichereinrichtung (Abbildungstabelle) zum Speichern von Benutzer-Sitzungsinformationen bereit. Die zugehörigen Portlets haben Portlet-Parameterabbildungen, die Daten und Befehle aus den Benutzeranforderungen an die Portlets speichern. Diese Parameter aus den Portlet-Anforderungen werden an die http-Anforderungen von dem Portlet an die Back-End-Netzanwendung weitergeleitet.

[0038] Kritische Sitzungsinformationen wie lokale Informationen und Zeitüberschreitungsinformationen der Sitzung können nun zur Back-End-Netzanwendung weitergeleitet werden.

[0039] Bei einer Ausführungsform dieser Erfindung ist es nun möglich, dass eine Realisierungsmöglichkeit der Sitzungsweiterleitung vorhanden ist, um die gemeinsamen Sitzungsdaten zwischen einem Portalserver und seiner Back-End-Netzanwendung gemeinsam zu verwenden, wodurch die Back-End-Netzanwendung Informationen von dem Portalserver empfangen kann, damit diese von der Back-End-Netzanwendung genutzt werden können, wobei die Back-End-Netzanwendung synchron mit dem Portalserver reagiert.

[0040] In einer Ausführungsform der Erfindung wird eine Vorrichtung bereitgestellt, um einem Benutzer ein Netzportal für eine Netzanwendung anzuzeigen, wobei das Netzportal eine Vielzahl von zugehörigen Portlets, die Informationen untereinander gemeinsam verwenden, anzeigt, auf die durch den Benutzer zugegriffen werden kann, wobei die Vorrichtung Folgendes enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Portlet-Anwendung, die an dem Portalserver betrie-

ben wird, zum Verwalten einer Sammlung zugehöriger Portlets; wobei die Portlet-Anwendung Folgendes enthält: ein Mittel zum Starten von Portlets auf Anforderung eines Benutzers, um auf die Netzanwendung zuzugreifen; ein Mittel zum Verwalten eines Portlet-Anwendungs-Sitzungsobjekts für die Portlets; und eine Datenspeichereinrichtung des Portlet-Anwendungs-Sitzungsobjekts, die durch das Portlet-Anwendungs-Sitzungsobjekt gesteuert wird, um Parameter von Benutzeranforderungen zu speichern, um die Portlets dem Portlet-Anwendungs-Sitzungsobjekt zuzuordnen.

[0041] Die Vorrichtung der Erfindung kann in der Portlet-Anwendung einen Datenaustausch-Client der Portlet-Anwendung für einen Datenaustausch zwischen dem Portlet-Anwendungs-Sitzungsobjekt und der Netzanwendung enthalten, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zur Netzanwendung zu befördern. Die Portlet-Anwendung kann jedem Portlet, das einem Portlet-Anwendungs-Sitzungsobjekt zugeordnet ist, einen gemeinsamen Schlüssel zuweisen.

[0042] In einer weiteren Ausführungsform der Erfindung wird eine Vorrichtung bereitgestellt, um ein Netzportal für eine Netzanwendung einer Vielzahl von Benutzern anzuzeigen, wobei das Netzportal eine Vielzahl von Portlets anzeigt, Informationen gemeinsam verwendet, auf die durch die Benutzer zugegriffen werden kann, wobei die Vorrichtung Folgendes enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Portlet-Anwendung zum Betreiben an dem Portalserver für jeden aus der Vielzahl von Benutzern, um eine Sammlung zugehöriger Portlets für jeden aus der Vielzahl von Benutzern zu verwalten; wobei jede Portlet-Anwendung enthält: ein Mittel zum Starten von Portlets auf Anforderung eines aus der Vielzahl von Benutzern, um auf die Netzanwendung zuzugreifen; ein Mittel zum Verwalten eines Portlet-Anwendungs-Sitzungsobjekts für die Portlets; und eine Datenspeichereinrichtung des Portlet-Anwendungs-Sitzungsobjekts, die durch das Portlet-Anwendungs-Sitzungsobjekt gesteuert wird, um Parameter von Benutzeranforderungen zu speichern, um die Portlets dem Portlet-Anwendungs-Sitzungsobjekt zuzuordnen.

[0043] In einer weiteren Ausführungsform der Erfindung wird eine Vorrichtung bereitgestellt, um einem Benutzer ein Netzportal für die Vielzahl von Netzanwendungen anzuzeigen, wobei das Netzportal eine Vielzahl von zugehörigen Portlets anzeigt, die Informationen gemeinsam verwenden, auf die durch den Benutzer zugegriffen werden kann; wobei die Vorrichtung Folgendes enthält: einen Portlet-Server zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Vielzahl von Portlet-Anwendungen, die jeweils die Vielzahl von Netzanwendungen zum Betreiben am Portalserver betreffen, wobei jede Portlet-Anwendung in der Lage ist, eine Sammlung zugehöriger Portlets zu verwalten; wobei jede Portlet-Anwendung enthält: ein Mittel zum Starten von Portlets auf Anforderung eines Benutzers, um auf eine aus der Vielzahl von Netzanwendungen zuzugreifen; ein Mittel zum Verwalten eines Sitzungsobjekts der Portlet-Anwendung für die Portlets; und eine Datenspeichereinrichtung des Sitzungsobjekts der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird, um Parameter von Benutzeranforderungen zu speichern, um die Portlets der Portlet-Anwendung dem Sitzungsobjekt der Portlet-Anwendung der Sitzung der Portlet-Anwendung zuzuordnen.

[0044] Ein weiterer Aspekt der Vorrichtung der Erfindung enthält eine Benutzersitzungs-Informationstabelle, die so beschaffen ist, dass sie mehrere Netzanwendungen mit dem Sitzungsobjekt der Portlet-Anwendung verbindet.

[0045] In einer weiteren Ausführungsform der Erfindung wird eine Vorrichtung bereitgestellt, um einem Benutzer ein Netzportal für eine Netzanwendung anzuzeigen, wobei das Netzportal eine Vielzahl zugehöriger Portlets anzeigt, die untereinander Informationen gemeinsam verwenden, auf die durch den Benutzer zugegriffen werden kann; wobei die Vorrichtung Folgendes enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Portlet-Anwendung zum Betreiben am Portalserver, um eine Sammlung zugehöriger Portlets zu verwalten; wobei die Portlet-Anwendung enthält: ein Mittel zum Starten eines ersten Portlet auf Anforderung eines Benutzers, auf die Netzanwendung zuzugreifen; ein Mittel zum Erzeugen eines Sitzungsobjekts der Portlet-Anwendung für den Benutzer für das erste Portlet; ein Mittel, um Parameter von der Anforderung zu speichern; ein Mittel zum Erzeugen zusätzlicher Portlets, die mit dem ersten Portlet verbunden sind, auf weitere Anforderungen des Benutzers, auf die Netzanwendung zuzugreifen; eine Datenspeichereinrichtung des Sitzungsobjekts der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird, um die gespeicherten Parameter zu verwenden, die zusätzlichen Portlets dem Sitzungsobjekt der Portlet-Anwendung zuzuordnen; und ein Mittel zum Erzeugen eines Datenaustausch-Client der Portlet-Anwendung (`httpClient`) für einen Datenaustausch mit dem Sitzungsobjekt der Portlet-Anwendung und der Netzanwendung, um Benutzeranforderungen, die von dem ersten und den zusätzlichen Portlets empfangen werden, zur Netzanwendung zu befördern.

[0046] Die Vorrichtung kann in der Portlet-Anwendung einen Datenaustausch-Client der Portlet-Anwendung für einen Datenaustausch zwischen dem Sitzungsobjekt der Portlet-Anwendung und der Netzanwendung enthalten, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zur Netzanwendung zu befördern.

[0047] Die Portlet-Anwendung weist vorzugsweise einen gemeinsamen Schlüssel jedem Portlet zu, das einem Sitzungsobjekt der Portlet-Anwendung zugeordnet ist.

[0048] Eine Benutzersitzungs-Informationstabelle, die so beschaffen ist, dass sie mehrere Netzanwendungen mit dem Sitzungsobjekt der Portlet-Anwendung verbindet, kann vorteilhaft bereitgestellt werden.

[0049] In einer weiteren Ausführungsform der Erfindung wird eine Vorrichtung bereitgestellt, um einem Benutzer ein Netzportal für eine Netzanwendung anzuzeigen, wobei das Netzportal eine Vielzahl zugehöriger Portlets anzeigt, die Informationen gemeinsam verwenden, auf die durch den Benutzer zugegriffen werden kann; wobei die Vorrichtung enthält: einen Portalserver, der ein Netzportal betreibt, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Portlet-Anwendung, die am Portalserver betrieben wird, um eine Sammlung zugehöriger Portlets zu verwalten; wobei die Portlet-Anwendung enthält: ein Mittel zum Starten von Portlets auf Anforderungen eines Benutzers, auf die Netzanwendung zuzugreifen; ein Mittel zum Verwalten eines Sitzungsobjekts der Portlet-Anwendung für die Portlets; und eine Datenspeichereinrichtung für Sitzungsobjekte der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird, zum Speichern von Parametern von den Benutzeranforderungen, um die Portlets dem Sitzungsobjekt der Portlet-Anwendung zuzuordnen.

[0050] In einem weiteren Aspekt der Erfindung wird ein Verfahren zum gemeinsamen Verwenden von Informationen zwischen einer Vielzahl von zugehörigen Portlets in einem Netzportal bereitgestellt, wobei das Verfahren die folgenden Schritte beinhaltet: Gewähren des Zugriffs für jedes Portlet aus der Vielzahl von Portlets auf eine Portlet-Datenspeichereinrichtung; Zulassen, dass jedes Portlet aus der Vielzahl von zugehörigen Portlets Daten in die Portlet-Datenspeichereinrichtung schreibt und Daten von der Portlet-Datenspeichereinrichtung liest.

[0051] Das obige Verfahren kann vorteilhaft ein System bereitstellen, in dem die zugehörigen Portlets durch eine Portlet-Anwendung verwaltet wird, die so beschaffen ist, dass sie an einem Datenverarbeitungssystem betrieben wird; wobei die Portlet-Datenspeichereinrichtung einen Speicherbereich der Portlet-Anwendung umfasst, der durch ein Sitzungsobjekt der Portlet-Anwendung verwaltet wird, das das Lesen und Schreiben von Daten durch die zugehörigen Portlets in die Datenspeichereinrichtung steuert, wodurch ein Datenaustausch zwischen den zugehörigen Portlets in der Portlet-Anwendung ermöglicht wird.

[0052] In einem weiteren Aspekt der Erfindung wird eine Vorrichtung bereitgestellt, um Informationen zwischen mehreren zugehörigen Portlets in einem Netzportal gemeinsam zu verwenden, wobei die Vorrichtung Folgendes enthält: eine Portlet-Anwendung zum Verwalten der mehreren zugehörigen Portlets; eine Datenspeichereinrichtung der Portlet-Anwendung; ein Mittel zum Gewähren eines Lese/Schreibzugriffs auf die Datenspeichereinrichtung durch die mehreren zugehörigen Portlets, damit die Portlets untereinander Daten austauschen können.

[0053] In einem weiteren Aspekt der Erfindung wird ein Portlet-Server (Anwendungsserver) bereitgestellt, der an einem Portalserver betrieben werden kann, um mehrere zugehörige Portlets in einem Netzportal aufzunehmen, wobei der Portlet-Server Folgendes enthält: ein Mittel zum Verwalten der mehreren zugehörigen Portlets; ein Mittel zum Verwalten eines Sitzungsobjekt der Portlet-Anwendung; eine Datenspeichereinrichtung der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung verwaltet wird, um einen Lese/Schreibzugriff auf die Datenspeichereinrichtung für die mehreren zugehörigen Portlets zu gewähren, damit die zugehörigen Portlets untereinander Daten austauschen können.

[0054] In einem weiteren Aspekt der Erfindung wird ein Portlet-Server (Anwendungsserver) bereitgestellt, der an einem Portalserver betrieben werden kann, um mehrere zugehörige Portlets in einem Netzportal aufzunehmen, wobei der Portlet-Server enthält: ein Mittel zum Verwalten der mehreren zugehörigen Portlets; ein Mittel zum Erzeugen und Verwalten eines Sitzungsobjekt der Portlet-Anwendung; eine Datenspeichereinrichtung der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung erzeugt und verwaltet wird, um einen Lese/Schreibzugriff auf die Datenspeichereinrichtung für die mehreren zugehörigen Portlets zu gewähren, damit die zugehörigen Portlets untereinander Daten austauschen können.

[0055] Die Portlet-Anwendung weist vorteilhaft jedem Portlet, das einem Sitzungsobjekt der Portlet-Anwendung zugehörig ist, einen gemeinsamen Schlüssel zu.

[0056] In einem weiteren Aspekt der Erfindung wird eine Portlet-Anwendung bereitgestellt, die an einem Portalserver betrieben werden kann, um mehrere zugehörige Portlets in ein Netzportal aufzunehmen, auf das durch einen Benutzer zugegriffen werden kann, wobei die Portlet-Anwendung enthält: ein Portlet-Anwendungsmittel zum Verwalten der mehreren zugehörigen Portlets; ein Portlet-Anwendungsmittel zum Verwalten eines Sitzungsobjekts der Portlet-Anwendung für den Benutzer; ein Portlet-Anwendungsmittel zum Gewähren des Schlüssels für jedes zugehörige Portlet, um einen Zugriff auf das Objekt der Portlet-Anwendung zu steuern.

[0057] In einem weiteren Aspekt der Erfindung wird eine Portlet-Anwendung bereitgestellt, die an einem Portalserver betrieben werden kann, um mehrere zugehörige Portlets in einem Netzportal aufzunehmen, auf das durch einen Benutzer zugegriffen werden kann; wobei die Portlet-Anwendung Folgendes enthält: ein Portlet-Anwendungsmittel zum Verwalten der mehreren zugehörigen Portlets; ein Portlet-Anwendungsmittel zum Erzeugen und Verwalten eines Sitzungsobjekts der Portlet-Anwendung für den Benutzer; ein Portlet-Anwendungsmittel zum Erzeugen und Verwalten eines Schlüssels für den Benutzer für das Sitzungsobjekt der Portlet-Anwendung; ein Portlet-Anwendungsmittel zum Gewähren des Schlüssels für jedes zugehörige Portlet, um einen Zugriff auf das Objekt der Portlet-Anwendung zu steuern.

[0058] Eine Portlet-Anwendung ist vorteilhaft jedem Benutzer zugewiesen und jeweils ein Schlüssel ist jedem Benutzer für entsprechende Objekte der Portlet-Anwendung für alle Portlet-Anwendungen zugewiesen.

[0059] In einem weiteren Aspekt der Erfindung wird eine Vorrichtung bereitgestellt, um einem Benutzer ein Netzportal für eine Netzanwendung anzuzeigen, wobei die Vorrichtung enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung durch einen Benutzer bereitzustellen; eine Portlet-Anwendung zum Verwalten einer Sammlung zugehöriger Portlets zum Betreiben am Portalserver; ein Sitzungsobjekt der Portlet-Anwendung für den Benutzer für die zugehörigen Portlets; eine Datenspeichereinrichtung des Sitzungsobjekts der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird; einen Datenaustausch-Client der Portlet-Anwendung, der mit der Datenspeichereinrichtung der Portlet-Anwendung für einen Datenaustausch zwischen den zugehörigen Portlets und der Netzanwendung verbunden ist, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zur Netzanwendung zu befördern; wobei der Datenaustausch-Client einen Anforderungspuffer zum Speichern und Synchronisieren von Anforderungen von den zugehörigen Portlets zu speichern und zu synchronisieren, damit der Datenaustausch-Client synchron mit der Netzanwendung arbeiten kann.

[0060] Der Datenaustausch-Client der Portlet-Anwendung ist vorzugsweise so beschaffen, dass er Informationen, die Anforderungen enthalten, über ein Netzwerk zu einer Netzanwendung sendet, und Informationen, die Antworten auf die Anforderungen enthalten, von der Netzanwendung empfängt.

[0061] In einem weiteren Aspekt der Erfindung wird eine Vorrichtung bereitgestellt, um einem Benutzer ein Netzportal für eine Netzanwendung anzuzeigen, wobei die Vorrichtung enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung durch einen Benutzer bereitzustellen; eine Portlet-Anwendung zum Verwalten einer Sammlung zugehöriger Portlets zum Betreiben an dem Portalserver; ein Sitzungsobjekt der Portlet-Anwendung für den Benutzer für die zugehörigen Portlets; eine Datenspeichereinrichtung des Sitzungsobjekts der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird; einen Datenaustausch-Client der Portlet-Anwendung, der mit der Datenspeichereinrichtung der Portlet-Anwendung verbunden ist für einen Datenaustausch zwischen den zugehörigen Portlets und der Netzanwendung, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zur Netzanwendung zu befördern; wobei der Datenaustausch-Client einen Anforderungspuffer zum Speichern und zum Herstellen einer seriellen Reihenfolge von Anforderungen von den zugehörigen Portlets, damit der Datenaustausch-Client in serieller Reihenfolge für die Netzanwendung erzeugen kann.

[0062] Der Datenaustausch-Client der Portlet-Anwendung ist vorzugsweise so beschaffen, dass er Informationen, die Anforderungen enthalten, über ein Netzwerk zu einer Netzanwendung oder zu einem Netzanwendungsserver sendet und Informationen, die Antworten auf die Anforderungen enthalten, von der Netzanwendung empfängt.

[0063] Ein weiterer Aspekt der Erfindung dient einem Portalserver, der so beschaffen ist, dass er ein Netzportal betreibt, um einen Zugriff auf eine Netzanwendung bereitzustellen; eine Portlet-Anwendung aufweist, die

am Portalserver betrieben wird, um eine Sammlung zugehöriger Portlets zu verwalten; wobei die Portlet-Anwendung Folgendes enthält: ein Mittel zum Starten von Portlets auf Anforderung eines Benutzers, auf die Netzanwendung zuzugreifen; ein Mittel zum Verwalten eines Sitzungsobjekts der Portlet-Anwendung für die Portlets; und eine Datenspeichereinrichtung des Sitzungsobjekts der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird, zum Speichern von Parametern von Benutzeranforderungen zum Zuordnen der Portlets zu dem Sitzungsobjekt der Portlet-Anwendung, wobei die Vorrichtung Folgendes enthält: einen Datenaustausch-Client der Portlet-Anwendung (`httpClient`), der mit der Datenspeichereinrichtung der Portlet-Anwendung verbunden ist, für einen Datenaustausch zwischen den zugehörigen Portlets und der Netzanwendung, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zu den Netzanwendungen zu befördern; wobei der Datenaustausch-Client der Portlet-Anwendung eine Informationsspeichereinrichtung der Benutzersitzung (Abbildungstabelle) zum Speichern von Benutzersitzungsinformationen, die ausgewählte Informationen aus der Menge der folgenden Benutzersitzungsinformationen enthalten: Benutzerkennung, Benutzer-Berechtigungsnachweise, Sprachpräferenzen, Sitzungszeitüberschreitungsinformationen, Sitzungskennung usw. zum Zuordnen der Benutzersitzungsinformationen zu einer entsprechenden Sitzung der Netzanwendung.

[0064] Die Sitzungszeitüberschreitungsinformationen enthalten vorzugsweise Sitzungszeitüberschreitungsinformationen des Portal servers und der Netzanwendung.

[0065] In einem weiteren Aspekt der Erfindung wird eine Portlet-Anwendung zum Verwalten einer Sammlung zugehöriger Portlets in einem Portal zum Betreiben an einem Server bereitgestellt, der einen Zugriff auf eine Netzanwendung durch einen Benutzer bereitstellt; wobei die zugehörigen Portlets Parameterabbildungen der Portlet-Anforderungen aufweisen, die Daten und Befehle von Benutzeranforderungen an die Portlets speichern; ein Sitzungsobjekt der Portlet-Anwendung für den Benutzer für die zugehörigen Portlets; eine Datenspeichereinrichtung der Sitzung der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird; einen Datenaustausch-Client der Portlet-Anwendung (`httpClient`), der mit der Datenspeichereinrichtung der Portlet-Anwendung für einen Datenaustausch zwischen den zugehörigen Portlets und der Netzanwendung verbunden ist, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zur Netzanwendung zu befördern; wobei der Datenaustausch-Client einen Anforderungspuffer aufweist, um Anforderungen von Parameterabbildungen der Portlet-Anforderungen der zugehörigen Portlets zu speichern, damit der Datenaustausch-Client Daten und Befehle für die Netzanwendung bereitstellen kann.

[0066] In einem weiteren Aspekt der Erfindung wird ein Datenaustausch-Client der Portlet-Anwendung (`httpClient`) bereitgestellt, der mit der Datenspeichereinrichtung der Portlet-Anwendung für einen Datenaustausch zwischen den zugehörigen Portlets und der Netzanwendung verbunden ist, um Benutzeranforderungen, die von den zugehörigen Portlets empfangen werden, zur Netzanwendung zu befördern; wobei der Datenaustausch-Client der Portlet-Anwendung eine Speichereinrichtung für Benutzeranforderungsinformationen (Abbildungstabelle) zum Speichern von Benutzeranforderungsinformationen aufweist, die ausgewählte Informationen aus der Menge der folgenden Benutzeranforderungsinformationen enthalten: Benutzerkennung, Benutzer-Berechtigungsnachweise, Sprachpräferenzen, Sitzungszeitüberschreitungsinformationen, Sitzungskennung usw. zum Zuordnen der Benutzersitzungsinformationen zu einer entsprechenden Sitzung der Netzanwendung; wobei die Sitzungszeitüberschreitungsinformationen Sitzungszeitüberschreitungsinformationen des Portal servers und der Netzanwendung enthalten.

[0067] Das oben Genannte enthält vorzugsweise ein Synchronisationsmittel für den Datenaustausch-Client der Portlet-Anwendung für das Anpassen von Sitzungszeitüberschreitungsgliedern zwischen dem Portalserver und der Netzanwendung durch Erteilen einer erneuten Berechtigung an den Benutzer, wenn das Zeitüberschreitungsglied der Netzanwendung vor dem Portalserver abläuft.

[0068] In einem weiteren Aspekt der Erfindung wird ein Portlet-Anwendung bereitgestellt, die an einem Portalserver zum Aufnehmen mehrerer zugehöriger Portlets in einem Netzportal betrieben werden kann, auf das durch einen Benutzer zugegriffen werden kann, wobei der Portalserver ein Nachrichtenübertragungsmittel bereitstellt, damit die zugehörigen Portlets untereinander Nachrichten übertragen können, wobei die Portlet-Anwendung Folgendes enthält: ein Portlet-Anwendungsmittel zum Verwalten der Vielzahl zugehöriger Portlets; wobei jedes zugeordnete Portlet einen Portlet-Deskriptor aufweist, der Kontextnamen beschreibt; wobei die zugehörigen Portlets Zusammenarbeitsgruppen aus Portlets enthält, die entsprechende Kontextnamen haben, die Kontextwerte definieren; wobei jede Gruppe ein Master-Portlet und wenigstens ein Slave-Portlet enthält; wobei jede Gruppe aus Portlets Kontextnamen gemeinsam verwenden; ein Mittel in dem Portalserver, um Änderungen im Datenaustausch in den Kontextwerten eines Master-Portlet an Slave-Portlets des Master-Portlets rundzusenden; ein Mittel in dem Portalserver zum Ändern von Kontextwerten der Slave-Portlets, um Kontext-

werte des Master-Portlets als Rundsendedaten anzupassen.

[0069] In einem weiteren Aspekt der Erfindung wird eine Portlet-Anwendung bereitgestellt, die an einem Portalserver zum Aufnehmen mehrerer zugehöriger Portlets in ein Netzportal zu betreiben, auf das durch einen Benutzer zugegriffen werden kann, wobei der Portalserver eine Portlet-Auffrischungsmöglichkeit aufweist, wobei die Portlet-Anwendung Folgendes enthält: ein Portlet-Anwendungsmittel zum Verwalten der Vielzahl zugehöriger Portlets; wobei jedes zugehörige Portlet einen Portlet-Deskriptor aufweist; wobei jeder Portlet-Deskriptor eine Prioritätsbeschreibung der Auffrischung für das Portlet aufweist; wobei die zugehörigen Portlets Zusammenarbeitsgruppen aus Portlets enthalten; wobei jede Gruppe aus Portlets ein Master-Portlet und wenigstens ein Slave-Portlet enthält, ein Mittel in dem Portlet-Anwendungsmittel zum Auffrischen der Portlets in der Reihenfolge ihrer Auffrischungsprioritäten.

[0070] In einem weiteren Aspekt der Erfindung wird eine Portlet-Anwendung bereitgestellt, die an einem Portalserver zum Aufnehmen mehrerer zugehöriger Portlets in ein Netzportal betrieben werden kann, auf das durch einen Benutzer zugegriffen werden kann; wobei die Portlet-Anwendung Folgendes enthält: die zugehörigen Portlets, die Zusammenarbeitsgruppen aus Portlets enthalten; ein Portlet-Anwendungsmittel zum Verwalten der Vielzahl zugehöriger Portlets; wobei jedes zugehörige Portlet einen Portlet-Deskriptor aufweist; wobei jeder Portlet-Deskriptor eine Prioritätsbeschreibung der Auffrischung für das Portlet aufweist und eine Auffrischungsbeschreibungsriorität für die Gruppe aus Portlets, in der das Portlet Mitglied ist; wobei die Gruppe aus Portlets ein Master-Portlet und wenigstens ein Slave-Portlet enthält; ein Mittel in dem Portlet-Anwendungsmittel zum Auffrischen der Portlets in der Reihenfolge ihrer Prioritäten; ein Mittel in dem Portlet-Anwendungsmittel zum Auffrischen der Zusammenarbeitsgruppen aus Portlets in der Reihenfolge ihrer Gruppenauffrischungsprioritäten.

[0071] Die Master-Portlets haben höhere Prioritäten als Slave-Portlets.

[0072] Die Portlet-Anwendung frischt vorzugsweise die Gruppen in der Reihenfolge der Gruppenpriorität auf und frischt dann innerhalb jeder Gruppe in der Prioritätsreihenfolge auf.

[0073] In einem weiteren Aspekt der Erfindung wird eine Vorrichtung bereitgestellt, um dem Benutzer eine Netzseitensitzung für eine Netzanwendung anzuzeigen, wobei die Netzseitensitzung eine Vielzahl zugehöriger zusammenarbeitender Portlets anzeigt, die untereinander Informationen gemeinsam verwenden, auf durch den Benutzer zugegriffen werden kann, wobei die Vorrichtung Folgendes enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Portlet-Anwendung zum Verwalten einer Sammlung zugehöriger Portlets für einen Betrieb am Portalserver; ein Zugriffsmittel zum Zugreifen auf eine Regel-Datenbank; wobei die Regeln solche Regeln enthalten, die die Anzeige der Gruppe von Portlets, Seiten, Seitengruppen für Benutzer steuern; ein Auswahlmittel zum Auswählen einer Menge von Portlets, Seiten und Seitengruppen, die einem Benutzer anhand von Informationen, die durch den Benutzer bereitgestellt werden (Informationseigenschaften) angezeigt werden sollen.

[0074] In einer weiteren Variation der Erfindung enthält das Auswahlmittel eine ansteckbare Regelmaschine, eine Regel-Datenbank und eine Verknüpfungsmaschine der Portlet-Anwendung, die Regeln anwendet, um ausgewählte Portlets, Seiten und Seitengruppen auszuwählen und einem Benutzer anzuzeigen.

[0075] In einem weiteren Aspekt der Erfindung wird eine Vorrichtung zum Anzeigen einer Netzseitensitzung für eine Netzanwendung für einen Benutzer bereitgestellt, wobei die Netzseitensitzung eine Vielzahl zugehöriger zusammenarbeitender Portlets anzeigt, die Informationen gemeinsam verwenden, auf die der Benutzer zugreifen kann, wobei die Vorrichtung Folgendes enthält: einen Portalserver zum Betreiben eines Netzportals, um einen Zugriff auf die Netzanwendung bereitzustellen; eine Portlet-Anwendung zum Verwalten einer Sammlung zugehöriger Portlets für einen Betrieb am Portalserver; ein Rollenzugriffsmittel zum Zugreifen auf eine Rollen-Datenbank, wobei die Rollen-Datenbank Regeln enthält, die die Anzeige von Mengen von Portlets, Seiten, Seitengruppen an Benutzer anhand von Benutzerrollen steuert; ein Rollenauswahlmittel, um eine Menge von Portlets, Seiten und Seitengruppen, die einem Benutzer angezeigt werden sollen, anhand einer identifizierten Rolle des Benutzers auszuwählen.

[0076] In weiteren Aspekten der Erfindung wird ein Artikel bereitgestellt, der Folgendes enthält: ein computerlesbares signaltragendes Medium; ein Computerprogrammcodemittel, das auf dem Medium aufgezeichnet und so beschaffen ist, dass es die Verfahren der Ausführungsformen der oben beschriebenen Erfindung ausführt.

[0077] In weiteren Aspekten der Erfindung wird ein Artikel bereitgestellt, der Folgendes enthält: ein computerlesbares signaltragendes Medium; ein Computerprogrammcodemittel, das auf dem Medium aufgezeichnet ist und so beschaffen ist, dass es die Vorrichtung einer der Ausführungsformen der oben beschriebenen Erfindung realisiert.

[0078] Das Medium kann aus der Gruppe ausgewählt sein, die aus magnetischen, optischen, biologischen und gegebenenfalls atomaren Datenspeichermedien besteht.

[0079] Das Medium kann ein moduliertes Trägersignal sein.

[0080] Das Signal kann eine Übertragung über ein Netzwerk sein.

Kurzbeschreibung der Zeichnungen

[0081] Ausführungsformen der vorliegenden Erfindung werden beispielhaft unter Bezugnahme auf die beigefügten Zeichnungen beschrieben; es zeigen:

[0082] [Fig. 1](#) ein Modell der Verkettung von dynamischem Kontext;

[0083] [Fig. 2](#) eine Integration einer Netzanwendung mit einem Portal;

[0084] [Fig. 3](#) eine strukturelle Darstellung der Integration;

[0085] [Fig. 4](#) einen Ablaufplan der Integration;

[0086] [Fig. 5](#) eine strukturelle Darstellung für eine Portalintegration mit einer Netzanwendung;

[0087] [Fig. 6](#) einen Ablaufplan für eine Integration;

[0088] [Fig. 7](#) ein Beispiel einer Gruppe für dynamischen Kontext für Portlets;

[0089] [Fig. 8](#) eine Initialisierung einer Portlet-Anwendung für dynamischen Kontext, wie in den Definitionen festgelegt;

[0090] [Fig. 9](#) einen Laufzeitablauf einer Portlet-Gruppe für dynamischen Kontext;

[0091] [Fig. 10](#) eine rollenbasierte Strukturabbildung für eine dynamische Verknüpfung von Komponenten;

[0092] [Fig. 11](#) eine regelbasierte Ablaufabbildung für eine dynamische Verknüpfung von Komponenten;

[0093] [Fig. 12](#) einen rollenbasierten Ablaufplan für eine dynamische Verknüpfung;

[0094] [Fig. 13](#) die Behandlung von Portlet-Anforderungen an Netzanwendungen;

[0095] [Fig. 14](#) eine Synchronisationsmodell-Darstellung;

[0096] [Fig. 15](#) einen Ablaufplan für eine Sequenz-Aware-Portalverknüpfungsmaschine;

[0097] [Fig. 16](#) das Definieren einer dynamischen Gruppe, die mit "MaleTeen" bezeichnet wird, und das Zuweisen von Benutzern zu der Gruppe;

[0098] [Fig. 17](#) das Zuweisen einer regelbasierten Auswahlaktion einer Datenbank-Inhaltgruppe zu einer dynamischen Benutzergruppe; und

[0099] [Fig. 18](#) die Erzeugung einer neuen Aktion, die mit "maleTeenAction" bezeichnet wird.

Genaue Beschreibung bevorzugter Ausführungsformen der Erfindung

[0100] In diesem Abschnitt werden bevorzugte Ausführungsformen der Erfindung beschrieben.

A.1. Ausrüstung zur Integration von Portal- und Netzanwendungen

[0101] [Fig. 2](#) veranschaulicht eine bevorzugte Ausführungsform der Erfindung, bei der ihre Verwendung mit einem Netzportalserver veranschaulicht wird.

A.1.1 http-Client der Portlet-Anwendung

[0102] Das Portlet (das http-Anforderungen an die Back-End-Netzanwendung übermittelt) verwendet den http-Client **209** der Portlet-Anwendung, der verwendet wird, um eine http-Verbindung zu einer Back-End-Netzanwendung herzustellen, die auf einem Back-End-Anwendungsserver **210** läuft. Die Back-End-Netzanwendung benötigt einen http-Client **209** der Portlet-Anwendung, um eine Sitzungsunterstützung für mehrere Anforderungen und Antworten, eine Cookie-Handhabung und eine Logik für eine Einmalanmeldung (Single Sign on – SSO) bereitzustellen. Alle Portlets in der gleichen Portlet-Anwendung verwenden das gleiche http-Client-Objekt **209** der Portlet-Anwendung, um mit einer oder mehreren Back-End-Netzanwendungen zu verbinden. Es gibt für jede Portlet-Anwendung **204** einen http-Client **209** der Portlet-Anwendung.

A.1.2 Portlet-Anwendungssitzung

[0103] Das Portlet-Anwendungssitzungsobjekt **208** ist ein vereinheitlichtes Datenspeicherungsobjekt, das von allen Portlets in einer Portlet-Anwendung gemeinsam verwendet werden kann. Dieses Objekt ist jeweils für einen Benutzer und eine Portlet-Anwendung vorhanden. Das Portlet-Anwendungssitzungsobjekt **208** stellt eine Infrastruktur dar, damit mehrere Portlets in einer Portlet-Anwendung unabhängige Benutzersitzungen haben (als Portlet-Sitzungen **204**, **205**, **206** bezeichnet), die jedoch die gleiche Portlet-Anwendungssitzung verwenden und mit der Netzanwendung an dem Back-End-Anwendungsserver **210** mit einer einzelnen Netzanwendungssitzung Daten austauschen.

A.1.3 Inhalt der Portlet-Anwendungssitzung

[0104] Der Inhalt der Portlet-Anwendungssitzung liefert Informationen, die für jeweils einen Benutzer und eine Portlet-Anwendung vorhanden sind. Das bedeutet, dass alle Portlets innerhalb der gleichen Portlet-Anwendung (**204**, **203**) nun eine Möglichkeit haben können, um allgemeine Informationen gemeinsam zu verwenden.

A.1.4 Sitzungsweiterleitungs-Mechanismus **320**

[0105] Der Sitzungsweiterleitungs-Mechanismus **320** ermöglicht das Weiterleiten von Informationen von der ursprünglichen http-Sitzung, die durch den Portalserver abgehalten wird, zur Back-End-http-Sitzung, die durch den http-Client der Portlet-Anwendung erzeugt wird. Dieser Mechanismus verwendet die folgende Infrastruktur:

Cookie-Tabelle **305** und Cookie-Nachschlagschlüssel

Die Cookie-Tabelle **305** (eine Benutzersitzungsinformationstabelle) ist die wesentliche Entität zum Abbilden der Portalserver-Cookies auf die Back-End-Netzanwendungssitzungs-Cookies. Die Abbildungsbeziehung zwischen den Cookies der http-Anforderung des Portalservers und dem Cookie des http-Client der Portlet-Anwendung für eine Netzanwendung ist eins zu eins. Ein http-Client der Portlet-Anwendung kann jedoch http-Anforderungen an unterschiedliche Netzanwendungen machen, wobei jede Netzanwendung unabhängige Sitzungen aufrechterhält. In dieser Hinsicht können die Abbildung zwischen dem Portalserver-Cookie und jene der Back-End-Netzanwendungen in der Form eins zu viele erfolgen (infolge von mehreren Back-End-Netzanwendungsservern).

[0106] [Fig. 13](#) zeigt diese Abbildung, bei der eine Anzahl von Elementen dargestellt ist:

RQ1: Cookie von der http-Anforderung eines Benutzeragenten (Browser) an den Portalserver

RQA: Cookie von der http-Anforderung des Portlet-http-Anwendungs-Client an die Netzanwendung A

RQB: Cookie von der http-Anforderung des Portlet-http-Anwendungs-Client an die Netzanwendung B

[0107] Der Portlet-Anwendungs-http-Client **209** verwendet diese Tabelle, um das passende Cookie für die Back-End-Netzanwendung, die auf dem Back-End-Netzanwendungsserver **210** läuft, nachzuschlagen.

[0108] Das Vorhandensein dieser Cookie-Abbildungstabelle **305** ermöglicht die automatische Beendigung einer Back-End-Netzanwendungssitzung, wenn die Portalserver-sitzung endet.

[0109] Der Portlet-Anwendungs-*http-Client 209* wird für jede Portlet-Anwendung erzeugt. Der Cookie-Nachschriftschlüssel wird in dem Sitzungsobjekt der Portalanwendung, auf das durch alle Portlets innerhalb der gleichen Portlet-Anwendung zugegriffen werden kann. Dieser Cookie-Nachschriftschlüssel ist zuständig für das Anpassen der *http*-Sitzung des Portalserver an die *http*-Sitzung der Back-End-Anwendung.

[0110] Die Verwendung des Cookie-Nachschriftschlüssels ermöglicht, dass alle Portlets in einer Portlet-Anwendung, die den gleichen *http-Client-Schlüssel* gemeinsam verwenden, die korrekte Menge von Back-End-Netzanwendungsinformationen für die gegenwärtig angemeldeten Benutzer abrufen und weiterleiten, so dass alle Portlets in der gleichen Portlet-Anwendung synchron arbeiten, um die verwendete Back-End-Anwendung zu aktualisieren. Die Wirkung besteht darin, dass der Endbenutzer eine vereinheitlichte Ansicht der Back-End-Netzanwendung über eine Vielzahl von Portlets sieht.

Abbildung von Portlet-Anforderungsparametern

[0111] Die **Abb. 308** von Portlet-Anforderungsparametern befindet sich in einem Speicherobjekt, das in der gemeinsam verwendeten Speichereinrichtung der Anwendungssitzungsdaten gespeichert ist, die für jedes Portlet und jede Portalserversitzung erzeugt wird. Sie wird verwendet, um alle Anforderungsparameter aus einer ankommenden Benutzeranforderung an ein bestimmtes Portlet zu speichern.

A.2 Dynamische Inhaltsynchronisation von Portlets

A.2.1 Dynamische Kontextdefinitionsmaske

[0112] [Fig. 5](#) veranschaulicht eine Portalintegration mit einer Back-End-Netzanwendung. Eine Bezugnahme auf [Fig. 5](#) ist für das Folgende nützlich:

Die dynamische Kontextdefinitionsmaske **503** definiert das Folgende für jede dynamische Kontextgruppe:

- den Kontext und seinen Typ (in unserem vorhergehenden Beispiel ist das die Kontokennung)
- das Master-Portlet, das den Wert des definierten Kontexts ändern kann
- das Slave-Portlet (die Slave-Portlets), das benachrichtigt wird, wenn der definierte Kontext geändert wird
- die registrierte Antwort (oder Aktion) des Slave-Portlet (der Slave-Portlets) bei Benachrichtigung über die Kontextänderung
- definiert optional die Auffrischungssequenz der Slave-Portlets (der Master wird innerhalb einer Gruppe stets zuerst aktualisiert)

[0113] Eine dynamische Kontextdefinitionsmaske **503** kann eine oder viele dynamische Kontextgruppen enthalten. Jede dynamische Kontextgruppe kann jedoch lediglich aufweisen

- ein Master-Portlet
- einen definierten Kontext
- ein oder mehrere Slave-Portlets

[0114] Hinweis: Ein Portlet kann an mehr als einer dynamischen Kontextgruppe bei unterschiedlichen Rollen in jeder Gruppe beteiligt sein.

A.2.2 Erzeugungswerkzeug einer Portlet-Gruppierung mit dynamischem Kontext

[0115] Dieses Werkzeug **501** liest in der dynamischen Kontextdefinitionsmaske **503** und erzeugt ein Master-Portlet und Slave-Portlets für dynamischen Kontext für alle dynamischen Kontextgruppen gemäß der Definition, die durch entsprechendes Aktualisieren der Portlet-Verwendungsdeskriptoren **502** festgelegt ist.

A.2.3 Dynamische Kontextgruppe

[0116] Eine dynamische Kontextgruppe ist eine Teilmenge von Portlets, die den gleichen Kontext gemeinsam verwenden und in einer dynamischen Kontextgruppe gruppiert sind. Ein Portlet kann zu mehr als einer dynamischen Kontextgruppe gehören.

[0117] Das Definitionsdocument **504** der dynamischen Kontextgruppe wird verwendet, um den dynamischen Kontext einer bestimmten dynamischen Kontextgruppe zu definieren.

Master-Portlet für dynamischen Kontext

[0118] Das Master-Portlet für dynamischen Kontext ist verantwortlich für das

- Erfassen der Zustandsänderung des Kontexts
- Benachrichtigen aller Slave-Portlets über die Zustandsänderung des Kontexts.

Slave-Portlet(s) für dynamischen Kontext

[0119] Die Slave-Portlets für dynamischen Kontext bewirken Folgendes:

- Abrufen einer Kontextänderung, die durch das Master-Portlet gemeldet wird
- Ausführen der registrierten Aktion, die auf die entsprechenden Back-End-Anwendung gerichtet ist, bei der Benachrichtigung über eine Kontextänderung

Modelle dynamischen Kontexts

[0120] Es gibt zwei Typen von Modellen dynamischen Kontexts, die verwendet werden können, um Portlets einander zuzuordnen:

A.2.4 Das Synchronisationsmodell

[0121] Bei dem Synchronisationsmodell, das in [Fig. 14](#) dargestellt ist, informiert das Master-Portlet **101** die Slaves **1701** bis **1703** über die Zustandsänderung des Kontexts des Master-Portlet für dynamischen Kontext. Alle Slaves führen Aktionen anhand einer im Voraus definierten Antwort aus, um sich mit der Zustandsänderung des Kontexts des Masters zu synchronisieren.

Darstellung des Synchronisationsmodells

A 2.5 Das Verkettungsmodell

[0122] Bei dem Verkettungsmodell, das in [Fig. 1](#) angegeben ist, führt die Änderung des Zustands im Master **A 101** zu der Antwortaktion des Slave **A 102**, wobei der Slave **A** außerdem das Master-Portlet **B** ist, das zur Änderung des Zustands im Kontext **B** führt, die die Kontextänderungsantwort des Slave **B 102** zur Folge hat, wobei der Slave **B** außerdem das Master-Portlet der dynamischen Kontextgruppe **C** ist, was die Antwortaktion des Slave **C** zur Folge hat.

A.2.6 Portlet-Transaktionsmanager

[0123] Bei einer Sequenz-Aware-Erweiterung der Portalverknüpfungsmaschine unter Bezugnahme auf [Fig. 15](#) ist der Portlet-Transaktionsmanager **1802** die Komponente, die für das Verwalten der Laufzeit-Auffrischungsfolgesteuerung der Portlets, einschließlich der Erzeugung von Portlet-Anforderungen, -Antworten und -Sitzungen zuständig ist.

1. Das erste Portlet, das bei jeder Portlet-Anwendung aufzufrischen ist, ist als das Portlet definiert, das für einen Benutzer von allen Portlets zuerst aufgefrischt wird. Es gibt keinen Mechanismus zum Definieren der Auffrischungsfolge von Portlets innerhalb einer Seite.

[0124] Deswegen wird eine Logik benötigt, die das Master-Portlet zur Laufzeit dynamisch identifizieren kann. In der vorliegenden Erfindung wird ein einfacher Notizzettel verwendet, auf dem jedes Portlet eine Markierung anbringt, wenn es aufgefrischt wird. Wenn zum ersten Mal eine Markierung auf diesem Notizzettel durch ein Portlet angebracht wird, weiß das Portlet, dass es das erste Portlet oder das Master-Portlet ist. Das nächste Portlet, das eine Markierung auf dem Notizzettel anbringt, sieht, dass bereits ein anderes Portlet eine Markierung angebracht hat, und weiß, dass es nicht das Master-Portlet ist. Wenn die Portalseite das nächste Mal aufgefrischt wird, wird das erste Portlet, das eine doppelte Markierung auf diesem Zettel anbringt, das Master-Portlet. Das Master-Portlet initialisiert dann diesen Zettel neu, indem es die Markierungen aller anderen Portlets sowie außerdem eine seine doppelten Markierungen für die nächste Anforderung entfernt. Dieser Algorithmus ermöglicht, das Master-Portlet immer dann dynamisch zu erkennen, wenn eine Anforderung vom Portalserver der Portlets eintrifft.

[0125] Nachdem das erste Portlet aufgefrischt wurde, übernimmt der Transaktionsmanager das Auffrischen der anderen Portlets in der Abfolge, die in der Abbildung der Master- und Slave-Portlets der Gruppe für dynamischen Kontext im Voraus definiert ist.

2. Sequenzsortierer: Das Sequenzsortierermodul **1804** wird verwendet, um die Portlets in ihrer Reihenfolge der Auffrischungsabfolge zu sortieren. Es verwendet den Portlet-Verwendungsdeskriptor, um die Auffrischungsreihenfolge aller Portlets zu kennzeichnen, und sortiert sie dann für die Anforderungsabgabemaschine.

3. Erweiterung der Sequenz-Aware-Anforderungsabgabemaschine: Diese Maschine **1805** wird verwendet, um Anforderungen an die Portlets auszugeben und überschreibt die Portalverknüpfungsmaschine. Ihre Aufgabe besteht darin, die geeigneten Portlet-Anforderungs- und -Antwortobjekte sowie die Portlet-Sitzung für alle Portlets in der Handels-Portal-Anwendung zu bilden. Sie wird dann von dem Transaktionsmanager verwendet, um die Portlets tatsächlich aufzufrischen.

4. Zwischenspeichereinheit des Transaktionsmanagers: Die Zwischenspeichereinheit **1806** des Transaktionsmanagers wird vom Transaktionsmanager **1802** verwendet, um die Antworten, die von den Portlets erzeugt werden, wenn sie durch die Anforderungsabgabemaschine aufgefrischt werden, zwischenspeichern. Das ist erforderlich, da dann, wenn die Portalverknüpfungsmaschine nun eine Portlet-Auffrischung anfordert, diese zwischengespeicherten Antworten durch den Transaktionsmanager zurückgesendet werden. Dadurch wird das Problem der doppelten Auffrischung für jede ankommende Portalanforderung vermieden.

A.3. Regelbasierte und rollenbasierte Verknüpfung

[0126] [Fig. 11](#) veranschaulicht eine regelbasierte Strukturabbildung einer dynamischen Verknüpfungskomponente einer bevorzugten Ausführungsform dieser Erfindung. Es folgt eine Beschreibung der Komponenten der veranschaulichten Ausführungsform und ihrer Funktionsweise.

Portalressourcen-Umsetzungsmodul

[0127] Das Portalressourcen-Umsetzungsmodul **1015** ist zuständig für das Umsetzen der Menge von Portalressourcen, die Portlets, Seiten und Seitengruppen enthalten, in eine Form, die syntaktisch analysiert und durch die externe Regelmaschine **1022** verarbeitet werden kann.

Regeldatenbank

[0128] Die Regeldatenbank **1001** hält vom Geschäftsmanager definierte Regeln für die Portalverknüpfungsmaschine **1006**.

Benutzerressourcen-Umsetzungsmodul

[0129] Das Benutzerressourcen-Umsetzungsmodul **1013** ist zuständig für das Umsetzen von Benutzerressourcen und den verschiedenen Benutzereigenschaften in eine Form, die syntaktisch analysiert und durch die externe Regelmaschine bearbeitet werden kann.

Einstechbare Regelmaschine

[0130] Die Regelmaschine **1022** ist (in dieser Ausführungsform der Erfindung) eine externe einsteckbare Regelmaschine wie etwa die Websphere™-Personalisierungsmaschine, die für das dynamische analytische Analysieren und die Ausführung von Regeln verwendet wird. Wenn die Maschine läuft, erzeugt sie die Menge von Portalressourcen, die der Benutzer sehen sollte, anhand der Geschäftsregeln, die durch den geschäftlichen Benutzer und die Benutzereigenschaften des gegenwärtigen Benutzers definiert sind.

Rollenbasierte Personalisierungsmaschine des Portals

[0131] Die rollenbasierte Personalisierungsmaschine **1008** des Portals ist ein rollenbasiertes Ressourcenauswahlmodul, das zum Extrahieren der Liste von Portalressourcen, auf die ein Benutzer zugreifen darf, und der Liste von Portalressourcen, auf die ein Benutzer nicht zugreifen darf, anhand der Organisationsmitgliedschaft des Benutzers verwendet wird.

[0132] Die rollenbasierte Maschine **1008** schaut zuerst auf die Organisation des Benutzers, indem auf die Rollendatenbank **1007** zugegriffen wird. Nachdem die Organisation des Benutzers festgestellt wurde, wird angenommen, dass seine Rolle die gleiche ist wie die Rolle dieser Organisation. Anschließend extrahiert die rollenbasierte Personalisierungsmaschine **1008** die Liste von Ressourcen, die für diese Organisation in der Weise definiert wurden, dass auf sie durch den Benutzer zugegriffen bzw. nicht zugegriffen werden kann. Nach-

dem diese Liste ermittelt wurde, wird sie durch dieses Modul für eine Weiterverarbeitung an das Übertragungs-undersystem für verknüpfte Ressourcen der Portalverknüpfungsmaschine weitergeleitet.

Rollen-Datenbank

[0133] In der Rollen-Datenbank **1007** sind die Organisationsdaten für den Portalserver gespeichert. Die Datenbank speichert Informationen über die Organisationsmitgliedschaft für verschiedene Benutzer sowie die Liste von Portalressourcen, auf die Mitglieder einer Organisation anhand ihrer Rollen zugreifen bzw. nicht zugreifen können.

Umsetzungsteilsystem für verknüpfte Ressourcen der Portalverknüpfungsmaschine

[0134] Dieses Modul **1004** ist zuständig für das Erzeugen der Master-Liste von Portalressourcen, die der gegenwärtige Benutzer sehen darf (dazu gehören Portlets, Seiten und Seitengruppen) anhand der Ausgabe der regel- und rollenbasierten Personalisierungsmaschinen. Dieses Modul ist außerdem ein Adapter für die eigentliche Portalverknüpfungsmaschine. Ihre Aufgabe besteht darin, nicht nur diese Master-Liste zu erzeugen, sondern sie außerdem in eine Form zu umzusetzen, auf die durch die eigentliche Portalverknüpfungsmaschine zugegriffen werden kann, um die endgültige Website für den Endbenutzer zu erzeugen.

Teil B: Funktionsbeschreibung

3.1 Beschreibung, wie Portal- und Netzanwendungen integriert werden können

B.1.1 Vollständige Integrationsstruktur und Ablaufpläne

[0135] Die [Fig. 2](#), [Fig. 3](#) und [Fig. 4](#) zeigen eine Integration von Netzanwendung mit einem Portal; eine strukturelle Darstellung der Integration; bzw. einen Ablaufplan der Integration.

3.1.2 Genaue Beschreibung

[0136] Wenn in [Fig. 2](#) eine Back-End-Netzanwendung mit dem Portalserver integriert ist, empfängt die Back-End-Netzanwendung **221** Anforderungen von dem Portalserver **201** über Portlets. Die Back-End-Netzanwendung **221** sendet Antworten an das Portlet zurück, das die Anforderung bildet.

[0137] Die Antwort von der Netzanwendung **221** wird über Portlets des Portalserver **201** einem Benutzer zugänglich gemacht, der auf das Portlet zugreift.

[0138] Bei der Realisierung des http-Client **209** der Portalanwendung werden mehrere Anforderungen und Antworten für die Back-End-Netzanwendung durch die Back-End-Netzanwendung als zusammenhängende Sitzungen wahrgenommen. Der http-Client **209** der Portalanwendung wird verwendet, um http-Datenübertragungsverbindungen zur Back-End-Netzanwendung **221** herzustellen. Die Back-End-Netzanwendung fordert den http-Client **209** der Portalanwendung auf, eine Sitzungsunterstützung, eine Cookie-Handhabung und Einmalanmelde-Funktionen (Single Sign On – SSO) bereitzustellen. Wenn der http-Client **209** der Portalanwendung vorhanden ist, können die Portlets mit der Netzanwendung wirksam Daten austauschen. Alle Portlets in einer Portlet-Anwendung (wie etwa die Portlet-Anwendung **205**) müssen einen Zugriff auf ein Sitzungsobjekt **211** der Portlet-Anwendung der Back-End-Netzanwendung **221** haben, das bedeutet, dass der http-Client **r** der Portalanwendung von allen Portlets innerhalb der gleichen Portlet-Anwendung gemeinsam verwendet werden muss.

[0139] Um eine derartige gemeinsame Verwendung zu ermöglichen, wurde festgestellt, dass ein vereinheitlichtes Sitzungsobjekt benötigt wird, das durch alle Portlets in einer Portlet-Anwendung gemeinsam verwendet wird. Um ein derartiges Objekt bereitzustellen, stellt die Erfindung ein Sitzungsobjekt **208** der Portlet-Anwendung bereit. Das Sitzungsobjekt **208** der Portlet-Anwendung ist ein Objekt, das durch die kommerzielle Portlet-Anwendung erzeugt wird. Auf das Sitzungsobjekt **208** der Portlet-Anwendung kann von allen Portlets in einer Portlet-Anwendung zugegriffen werden (wie etwa die Portlets **204, 205, 206** in der Portlet-Anwendung 1, **207**). Ohne das Sitzungsobjekt **208** der Portlet-Anwendung haben mehrere Portlets in einer Portlet-Anwendung unabhängige Benutzersitzungen und sind nicht in der Lage, sitzungsbezogene Informationen gemeinsam zu verwenden.

[0140] Der http-Client **209** der Portalanwendung ist in der Portlet-Anwendungssitzung **208** gespeichert, wo-

durch er unter den Portlets innerhalb der gleichen Portlet-Anwendung gemeinsam verwendet werden kann. Ohne dieses Sitzungsobjekt der Portlet-Anwendung könnten die Portlets nicht mit einer einzelnen Netzanwendungssitzung bei der Datenbankverwaltung Daten austauschen.

[0141] Alle Daten, die in dem Sitzungsobjekt **208** der Portlet-Anwendung gespeichert sind, repräsentieren Sitzungsinhalt der Portlet-Anwendung und sind jeweils einmal pro Benutzer und pro Portlet-Anwendung vorhanden.

[0142] Da der http-Client **209** der Portalanwendung alle Sitzungsinformationen für die Back-End-Netzanwendung **221** gespeichert hat, wird er als eine Basis für den Sitzungsweiterleitungsmechanismus **320** verwendet, der in [Fig. 3](#) dargestellt ist.

[0143] Eine Sitzungsweiterleitung ermöglicht, dass Sitzungsinformationen weitergeleitet werden, die für den gesamten Portalserver **201** spezifisch sind (wie etwa Sprachinformationen, Benutzeragent-Informationen usw.), zu den Sitzungsinformationen der Back-End-Netzanwendung **221** weitergeleitet werden. Das bedeutet, dass die Back-End-Netzanwendung **221** in der Lage ist, die Datendarstellung zu liefern, die mit allen Anforderungen konform ist, die in der ursprünglichen Anforderung enthalten sind, die durch einen Benutzer an den Portalserver gesendet wurde.

[0144] Wenn z. B. der Benutzer unter Verwendung einer WAP-fähigen Mobileinheit (WAP, wireless application protocol, Protokoll für Drahtlosanwendungen) auf das Portlet zugreift, wobei die voreingestellte lokale Sprache auf "Französisch" eingestellt ist, hat die ursprünglichen http-Anforderung an den Portalserver **201** einen ITS-Sprachparameter, der auf "Französisch" eingestellt ist, und das Benutzeragent-Feld des http-Kopfabschnitts ist auf "WAP" eingestellt. Der Sitzungsweiterleitungsmechanismus **320** leitet diese Informationen an die Netzanwendung **221** weiter, und die Netzanwendung sendet eine Antwort auf Französisch zurück, die für eine Anzeige auf der Mobileinheit des Benutzers auf Französisch geeignet ist. Wenn die Sitzungsweiterleitung fehlen würde, würde die Netzanwendung die Informationen in der bevorzugt eingestellten Sprache (z. B. Englisch), die für die bevorzugte Einheit geeignet ist (z. B. ein Internet-Browser), zurücksenden. In diesem Fall wäre der Benutzer nicht in der Lage, die abgerufenen Daten zu sehen, da sie mit der Mobileinheit des Benutzers inkompatibel wären.

[0145] Es erfolgt eine Bezugnahme auf Elemente in der strukturellen Darstellung von [Fig. 3](#), während Prozessstufen von [Fig. 4](#) durch nummerierte Schritte angegeben sind.

[0146] Im Schritt **401** tritt der Benutzer mit Portlets an einem Netzportal in Wechselwirkung, z. B. unter Verwendung einer Computermaus, um eine Verbindung oder ein Objekt anzuklicken, die bzw. das in einem Portlet auf dem Netzbrowser des Benutzers angezeigt sind. Jedes Portlet hat seine eigene Portlet-Sitzung **310** (Portlet-Sitzung ist ein Teil des Standes der Technik). Als Teil der Benutzerwechselwirkung erfolgt eine Fernanforderung **306** an die Back-End-Netzanwendung **307**.

2. Damit im Schritt **403** alle Parameter in der Portlet-Sitzung korrekt zur Back-End-Netzanwendung geleitet werden, wird die Parameterliste jeder Portlet-Anforderung in der Parameterabbildung (Nr. 8) **308** der Portlet-Anforderung gespeichert. Diese Parameter werden zur entfernten Back-End-Anforderung geleitet.
3. Im Schritt **404** verwendet das Handels-Portlet einen http-Client-Schlüssel **301**, um festzustellen, ob bereits ein Sitzungsobjekt **208** der Portlet-Anwendung und ein http-Client **303** der Portlet-Anwendung vorhanden sind, indem auf die Datenspeichereinrichtung Nr. 4, **302** der Portlet-Anwendung zugegriffen wird. Wenn die Elemente nicht gefunden werden, werden neue Elemente für alle Portlets innerhalb der gleichen Portlet-Anwendung erzeugt. (Wenn die vorhandenen Elemente dagegen im Schritt **407** gefunden werden, werden sie verwendet.)
4. Im Schritt **406** wird jeder Benutzer-Berechtigungsnachweis aus der ursprünglichen Portlet-Sitzung in der Cookie-Tabelle **305** gespeichert.
5. Im Schritt **408** werden die Benutzer-Berechtigungsnachweise aus der Cookie-Tabelle **305** und die Parameter, die zuvor in der Parameterabbildung **308** der Portlet-Anforderung gespeichert wurden, verwendet, um eine neue http-Anforderung an die Back-End-Netzanwendung zu bilden.
6. Im Schritt **409** erfolgt der Anruf an die entfernte Netzanwendung **307**.
7. Im Schritt **410** sendet die entfernte Netzanwendung **307** eine Antwort auf den Anruf für das Portlet zur Anzeige zurück.

B.2 Dynamische Kontextsynchronisation von Portlets

B.2.1 Beschreibung der Entwicklungszeit

[0147] Bei Bezugnahme auf [Fig. 5](#), die eine strukturelle Darstellung der Portalintegration mit einer Back-End-Netzanwendung darstellt, kann erkannt werden, dass ein Portalentwickler das Werkzeug **501** zur Portlet-Gruppierung bei dynamischem Kontext verwenden kann, um jeweils eine neue dynamische Gruppendefinitionsinstanz **504** zu erzeugen. Diese Instanz ist eine Gruppierung von zugehörigen Portlets und definiert, welche Portlets Slaves sind und welches von diesen Portlets der Master ist. Die erforderlichen Elemente der dynamischen Gruppdefinition sind in der Gruppdefinitionsmaske **503** für dynamischen Kontext festgelegt.

[0148] Der Benutzer verwendet das gleiche Werkzeug **501** zum Aktualisieren einer vorhandenen Gruppendefinition für dynamischen Kontext.

[0149] Nachdem der Benutzer die neueste Gruppdefinition für dynamischen Kontext bereitgestellt hat, aktualisiert das Werkzeug zur Portlet-Gruppierung bei dynamischem Kontext die geeigneten Verwendungsdeskriptoren **502** der Portlet-Anwendung, damit sie die in der Gruppe definierten Beziehungen widerspiegeln.

[0150] In [Fig. 6](#) können in einem Ablaufplan, der die Portalintegration darstellt, die Schritte des oben genannten Prozesses deutlicher veranschaulicht werden:

Wenn ein Benutzer eine Gruppe mit dynamischem Kontext erzeugen (**608**) oder aktualisieren (**609**) möchte, kann er das Gruppierungswerkzeug **501** verwenden ([Fig. 5](#)).

[0151] Im Schritt **601** fordert das Gruppierungswerkzeug für dynamischen Kontext eine Benutzereingabe auf Grundlage der Festlegung in der Gruppdefinitionsmaske **503** für dynamischen Kontext oder liest beim Aktualisieren des Gruppierungswerkzeugs für dynamischen Kontext in einer vorhandenen Gruppeninstanz für dynamischen Kontext, wobei sie mit der Definitionsmaske **503** gültig gemacht wird.

[0152] Im Schritt **603** legt der Benutzer die geforderten Informationen fest, um eine Gruppe für dynamischen Kontext zu definieren oder zu aktualisieren.

[0153] Im Schritt **605** wird die Gruppeninstanz **504** für dynamischen Kontext erzeugt.

[0154] Im Schritt **606** wird der Verwendungsdeskriptor aller betroffenen Portlets aktualisiert.

Gruppierung für dynamischen Kontext

[0155] [Fig. 7](#) veranschaulicht dynamischen Kontext für Portlets. Die dynamische Gruppe **701** besteht aus dem Master-Portlet **704**, dem Slave-Portlet **705** und dem Slave-Portlet **707**.

[0156] Die Gruppe **703** besteht aus dem Master-Portlet **705**, dem Slave-Portlet **706** und dem Slave-Portlet **707**.

[0157] Die dynamische Gruppe **702** besteht aus dem Master-Portlet **704** und dem Slave-Portlet **708**.

[0158] Wenn die Daten, die durch Portlets in einer Portlet-Anwendung repräsentiert werden, auf der Ebene der Back-End-Anwendung synchronisiert sind, bieten die Portlets eine koordinierte Ansicht der Daten, indem diese Daten einfach von der Netzanwendung abgerufen werden. Jedoch nicht alle Portlet-Wechselwirkungen haben Änderungen an der Back-End-Netzanwendung zur Folge. Der dynamische Kontext dient zur "sofortigen" Synchronisation. Sie ist am wirkungsvollsten, wenn eine Änderung im Kontext eine andere Anfrage erfordert. Das Auswählen eines anderen Kontos aus der Kontoliste erfordert z. B. das Anzeigen von Rechnungsinformationen, die mit dem ausgewählten Konto aufgefrischt werden müssen.

[0159] In Systemen nach dem Stand der Technik waren Portlets normalerweise voneinander unabhängig. Diese Erfindung stellt ein Verfahren und eine Vorrichtung bereit, um die gegenseitigen Beziehungen von Portlets abzubilden und ihre gegenseitige Abhängigkeit zum Zeitpunkt der Verwendung und Konfiguration der Portlet-Anwendung darzulegen. Die Portlets selbst müssen nicht geändert werden.

[0160] Die Abhängigkeitsbeziehungen zwischen Portlets können in einer Beziehungsmaske **503** für dynamischen Kontext definiert werden, in der die Master- und Slave-Beziehungen definiert sind.

[0161] Die Beziehungsmaske **503** für dynamischen Kontext wird vorzugsweise als eine XML-Datendarstellung codiert, die Folgendes definiert:

- die Teilmenge von Portlets, die eine Gruppe für dynamischen Kontext bildet
- das Master-Portlet einer Gruppe für dynamischen Kontext
- das Slave-Portlet (die Slave-Portlets) dieser Gruppe für dynamischen Kontext
- die Slave-Aktion, die das Slave-Portlet (die Slave-Portlets) bei Änderungen des Kontextzustands ausführen müssen
- den Kontext, den alle Bestandteile dieser Gruppe für dynamischen Kontext gemeinsam verwenden

[0162] Es folgt ein Beispiel einer Definitionsinstanz einer Gruppe für dynamischen Kontext:

```

<DynamicContextGroup>
  <DynamicContextGroupName>OrderRelatedPortletGroup
  </DynamicContextGroupName>
  <DynamicContextMasterPortlet>
    OrderItems
  </DynamicContextMasterPortlet>
  <DynamicContext>itemName
  </DynamicContext>
  <DynamicContextSlavePortlet>
    <DynamicContextSlavePortletName>UPSTracking
    </DynamicContextSlavePortletName>
    <SlavePortletAction>
      http://inventoryserver.com/inStock/
    </SlavePortletAction>
  </DynamicContextSlavePortlet>
</DynamicContextGroup>

<DynamicContextGroup>
  <DynamicContextGroupName>StockInventoryPortletGroup
  </DynamicContextGroupName>
  <DynamicContextMasterPortlet>
    InStockInventory
  </DynamicContextMasterPortlet>
  <DynamicContext>itemSKUnumber
  </DynamicContext>
  <DynamicContextSlavePortlet>
    <DynamicContextSlavePortletName>OrderedItems
    </DynamicContextSlavePortletName>
    <SlavePortletAction>
      http://myserver.com/lastOrdered/
    </SlavePortletAction>
  </DynamicContextSlavePortlet>
</DynamicContextGroup>

```

[0163] Anmerkung zu Definitionsinstanzen einer Gruppe für dynamischen Kontext: eine Definition einer Gruppe für dynamischen Kontext ist eine Instanz. Mehrere Definitionen einer Gruppe für dynamischen Kontext können in einer Datei vereinigt werden, um mehrere Instanzen zu definieren. Das oben Stehende definiert zwei Portlet-Mengen in einer Portlet-Anwendung, die aus 3 Portlets besteht.

[0164] In der ersten Gruppe für dynamischen Kontext ist der von den Portlets gemeinsam verwendete dynamische Kontext `itemName`, wobei das Portlet mit der Bezeichnung `OrderedItems` als Master-Portlet für dynamischen Kontext dient und die Portlets `UPSTracking` und `InStockInventory` als Slave-Portlets des dynamischen Kontexts dienen.

[0165] In der zweiten Gruppe für dynamischen Kontext ist der von den Portlets gemeinsam verwendete dynamische Kontext `itemSkuNumber`, wobei das Portlet mit der Bezeichnung `InStockInventory` als Master-Portlet für dynamischen Kontext und das Portlet `OrderedItems` als Slave-Portlet für dynamischen Kontext dient.

[0166] Jedes Master-Portlet für dynamischen Kontext beobachtet eine Benutzer-`http`-Anforderung und sucht nach dem dynamischen Kontext. Wenn der dynamische Kontext in der Anforderung gefunden wird, sendet das Portlet für dynamischen Kontext einen dynamischen Kontext (der das Parameterpaar `Name` und `Wert` in der `http`-Anforderung ist) zu den Slave-Portlets.

[0167] Wenn z. B. das Portlet `OrderedItems` eine `http`-Anforderung mit dem Attribut `itemName`, das auf "PentiumIV" gesetzt ist, empfängt, sendet es den dynamischen Kontext zu den Portlets `UPSTracking` und `InStockInventory`, wodurch sie benachrichtigt werden, dass der Kontext `itemName` mit dem Wert "PentiumIV" nun in den dynamischen Kontext eingesetzt wurde.

[0168] Jedes Slave-Portlet für dynamischen Kontext hört die Benachrichtigung des Masters an alle Slave-Portlets der gleichen Gruppe für dynamischen Kontext. Beim Empfang der Benachrichtigung des Masters wird die entsprechende Slave-Aktion aufgerufen, indem der dynamische Kontext der URL der Aktion angefügt wird, die in der Definitionsinstanz der Gruppe für dynamischen Kontext unter dem Attribut "SlavePortletAction" definiert ist.

[0169] Wenn z. B. das Portlet `inStockInventory` den dynamischen Kontext vom Portlet `OrderItems` mit dynamischem Kontext mit dem Typ "itemName" und dem Wert "PentiumIV" empfängt, ruft es die Daten von der URL `http://inventoryserver.com/inStock/itemName=PentiumIV` ab.

[0170] Es folgt eine Codierung für ein Beispiel der Definitionsmaske der Gruppe für dynamischen Kontext:

```

<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:cep=


"http://www.ibm.com/WebsphereCommerceEnabledPortal/DynamicContext
GroupDefinitionSchema">

  <annotation>

    <documentation xml:lang="en">
      Schema for Websphere Commerce Enabled Portal Dynamic Context
      Group Definition
      Copyright 2002 IBM Corporation
    </documentation>
  </annotation>

  <!--Dynamic Context Group Instance -->
  <xsd:element name="DynamicContextGroup"
    type="DynamicContextGroupDefinitionTemplate",
    minOccurs="1"/>

  <!--Dynamic Context Group Definition Template Schema --
  <xsd:complexType name="DynamicContextGroupDefinitionTemplate">
    <xsd:sequence>
      <xsd:element name="DynamicContextGroupName"
        type="xsd:string"/>
      <xsd:element name="DynamicContextMasterPortlet"
        type="PortletName"/>
      <!-- only one dynamic context per dynamic context group -->
      <xsd:element name="DynamicContext" type="ContextParameter"
        maxOccurs="1"/>
      <xsd:element name="DynamicContextSlavePortlet"
        type="SlavePortlet"
          minOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

<xsd:complexType name="SlavePortlet">
  <xsd:sequence>
    <xsd:element name="DynamicContextSlavePortlet"
type="PortletName"/>
    <xsd:element name="SlavePortletAction" type="xsd:string"/>
    <xsd:element name="SlavePortletRefreshPriority"
type="xsd:decimal",
minOccurs="0"/> .
    <!-- master's context is in the slave action url if slave param
map is absent -->
    <xsd:element name="SlaveParamMapToContext"
type="ContextParameter"
minOccurs="0"/>

  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="PortletName">
<xsd:string>
</xsd:simpleType>

<!-- name of the parameter in master's request url -->
<xsd:simpleType name="ContextParameter">
<xsd:string>
</xsd:simpleType>

</xsd:schema>

```

B.2.2 Laufzeit

[0171] Dieser Abschnitt wird durch Bezugnahme auf [Fig. 8](#) (Initialisierung einer Portlet-Anwendung für dynamischen Kontext gemäß Festlegung in der Definitionsinstanz) und [Fig. 9](#) (Laufzeitablauf der Portlet-Gruppe für dynamischen Kontext) am besten verstanden.

[0172] Es gibt zwei Hauptkomponenten, um Laufzeit-Aspekte des dynamischen Kontexts zu behandeln:

- 1) DynamicContextActionListener (**904**) (Abhöreinrichtung der Portlet-Aktion) – sie lauscht nach Änderungen des dynamischen Kontexts im Master-Portlet. Dem Master-Portlet in jeder Portlet-Gruppe für dynamischen Kontext ist DynamicContextActionListener angefügt.
- 2) DynamicContextMessageListener (**906**) (Abhöreinrichtung der Portlet-Nachricht) ist die Abhöreinrichtung für Nachrichten, die nach der Benachrichtigung von dem Master der Gruppe lauscht, in der spezieller dynamischer Kontext definiert ist. Jedem Slave-Portlet in der Portlet-Gruppe für dynamischen Kontext ist ein DynamicContextMessageListener angefügt.

Schrittweise Beschreibung des Laufzeitablaufs

[0173] Zum Zeitpunkt der Portlet-Initialisierung ([Fig. 8: 801](#)) fügen alle Master-Portlets den definierten dynamischen Kontext anhand des Portlet-Deskriptors (**802, 805**) der Aktionsabhöreinrichtung (**806**) des Master-Portlets an. Für alle Slave-Portlets werden der Typ des dynamischen Kontexts, die URL der Aktion, die Parameterabbildung und die Auffrischungssequenz vom Portlet-Deskriptor (**802, 809**) abgerufen und der Aktionsabhöreinrichtung (**810**) der Portlet-Nachricht des Slaves angefügt.

- 1) Die Benutzer-Wechselwirkung mit dem Master-Portlet der Portlet-Gruppe für dynamischen Kontext hat die Änderung des dynamischen Kontexts zur Folge (901).
- 2) Die Komponente DynamicContextActionListener des Master-Portlets erfasst die Aktion des Benutzers (902).
- 3) Die Komponente DynamicContextActionListener setzt das Name/Wert-Paar entsprechend dem dynamischen Kontext in dem Anforderungsobjekt des Master-Portlets (904).
- 4) Das master-Portlet erhält den Wert des dynamischen Kontexts und benachrichtigt alle Slave-Portlets innerhalb der gleichen dynamischen Portlet-Gruppe darüber (905).
- 5) Die Komponente DynamicContextMessageListener, die dem Slave-Portlet für den Master zugehörig ist, empfängt die Benachrichtigung (den Wert des dynamischen Kontexts) (906).
- 6) Die Komponente DynamicContextMessageListener setzt den Wert von DynamicContext im Portlet-Anforderungsobjekt des Slave-Portlets. (907)
- 7) Das Slave-Portlet erhält den Wert des dynamischen Kontexts (1008)
- 8) Das Slave-Portlet modifiziert eine Aktion, die für das Slave-Portlet definiert ist, wenn die Abbildung zwischen Kontext und einem Parameter festgelegt wurde (1009).
- 9) Wenn die Abbildung nicht festgelegt wurde, wird das Name/Wert-Paar des dynamischen Kontexts der Portlet-Aktion des Slaves angefügt.
- 10) Das Slave-Portlet führt die Aktion aus, die in der Instanzdefinition der Gruppe für dynamischen Kontext definiert ist (1011, 1012).

B.3 Regelbasierte/rollenbasierte dynamische Verknüpfung

[0174] Im diesem Abschnitt erfolgt eine Bezugnahme auf mehrere Figuren, dazu gehören: [Fig. 10](#) (strukturelle Abbildung einer rollenbasierten dynamischen Verknüpfung von Komponenten),

[0175] [Fig. 11](#) (strukturelle Abbildung einer regelbasierten dynamischen Verknüpfung von Komponenten) und [Fig. 12](#) (Ablaufplan einer regelbasierten dynamischen Verknüpfung).

[0176] Die rollen- und regelbasierten dynamischen Verknüpfungskomponenten für den Portlet-Server beruhen auf den Regel- und Rollen-Datenbanken und dem Konzept von Inhaltgruppen für jede Rolle und jede Regel.

[0177] Die Inhaltgruppen für die Regeln werden in der Komponente **1001** der Regel-Datenbank, die in [Fig. 10](#) gezeigt ist, geführt. In ähnlicher Weise sind die Rolleninhaltgruppen in der Komponente **1007** der Rollen-Datenbank, die in [Fig. 10](#) gezeigt ist, definiert. Jede Inhaltgruppe enthält eine Menge von Portalserver-Ressourcen, auf die ein Benutzer, der so bewertet wurde, dass er in den Bereich dieser speziellen Rolle oder Regel fällt, Zugriff haben sollte.

[0178] Eine weitere Hauptkomponente in diesem Schema ist die einsteckbare Regelmaschine **1022**. Die Aufgabe dieser Maschine besteht darin, die umgesetzten Benutzereigenschaften zu lesen und zur Laufzeit über die Menge von Benutzern dynamisch zu entscheiden, die sich anhand dieser Benutzereigenschaften für eine Mitgliedschaft in einer bestimmten, im Voraus definierten Benutzergruppe qualifizieren. Außerdem bildet diese Maschine die Menge dieser dynamischen Benutzergruppen auf die Menge der Inhaltgruppen ab, die in der Rollen- und Regel-Datenbank definiert wurden. Die einsteckbare Regelmaschine besitzt vorzugsweise eine grafische Benutzeroberfläche (GUI), um diese Aufgaben zu verwalten. Das Bildschirmfoto, das in [Fig. 16](#) dargestellt ist, veranschaulicht, wie die WebSphere-Personalisierungsservermaschine verwendet wird, um diese Aufgaben zu verwalten.

[0179] [Fig. 16](#) veranschaulicht z. B., wie eine dynamische Gruppe mit der Bezeichnung "MaleTeen" definiert wird und alle männlichen Benutzer im Alter zwischen 16 und 19 Jahren dieser Gruppe zuweist.

[0180] Wie in [Fig. 17](#) gezeigt ist, die darstellt, dass alle Benutzer, die anhand ihrer Eigenschaften dynamisch als männliche Jugendliche (Teenager) bewertet werden, nun den für sie ausgeführten Befehl "maleteenaction" aufweisen, der die Maschine **1022** zur regel- und rollenbasierten dynamischen Portalverknüpfung anweist, Inhaltressourcen für die Gruppe männlicher Jugendlicher aus der Rollen-DB **1007** auszuwählen.

[0181] Zum Entwicklungszeitpunkt ist es die Aufgabe eines Geschäftsmanagers, eine Menge von Portalressourcen wie etwa Seiten, Portlets usw. einer speziellen Inhaltgruppe in der Rollen- und Regel-Datenbank zuzuweisen. Das erfolgt gegenwärtig durch die Verwendung von SQL-Skripts, die die Regel- und Rollen-Datenbank direkt laden.

B.3.1 Beschreibung der Freigabe der Laufzeit einer regelbasierten, rollenbasierten dynamischen Verknüpfung

[0182] Zur Laufzeit ist der erste Befehl, der für einen Portalbenutzer ausgeführt wird, der Befehl Wrapper für die regelbasierte Maschine. Dieser Befehl ist eigentlich ein Proxy, der die Bewertung von Benutzereigenschaften durch die eigentliche einsteckbare Regelmaschine startet.

[0183] Im nächsten Schritt liest die Regelmaschine in den Benutzereigenschaften aus seinem gespeicherten Profil, indem das Benutzerressourcen-Übertragungsmodul verwendet wird, um sie in eine für sie verständliche Form umzusetzen.

[0184] [Fig. 18](#) veranschaulicht die Erzeugung einer neuen Aktion mit der Bezeichnung "MaleTeenAction", die alle Portalressourcen auswählt, die in der Regel-Datenbank in der Inhaltgruppe mit der Bezeichnung "male-teengrp" definiert sind.

[0185] [Fig. 17](#) veranschaulicht die Erzeugung eines dynamischen Verknüpfungsmodulbefehls, der das Verknüpfungsmodul anweist, die Inhalte von "maleteengrp" für alle Benutzer auszuwählen, die in den Bereich der zuvor erzeugten Regel zum Klassifizieren von "MaleTeens" anhand von dynamischen Benutzereigenschaften fallen.

[0186] [Fig. 17](#) veranschaulicht, wie eine Geschäftsregel (z. B. eine Geschäftsregel zum Definieren, was eine Gruppe männlicher Jugendlicher ausmacht) wirkt (z. B. maleTeenAction), um festzulegen, welcher Inhalt, der für einen Benutzer mit bestimmten Benutzereigenschaften verknüpft wird, in diese Klassifikation fällt.

[0187] Nach dem Lesen in den Benutzereigenschaften bewertet die einsteckbare Regelmaschine die Mitgliedschaft der dynamischen Gruppe dieses Benutzers anhand der Regeln, die für die verschiedenen dynamischen Gruppen, wie in [Fig. 18](#) gezeigt ist, definiert sind.

[0188] Nachdem die Menge dynamischer Gruppen für diesen Benutzer bestätigt wurde, wählt die Regelmaschine den geeigneten Portalinhalt für diesen Benutzer, indem die Inhaltauswahlaktionen ausgeführt werden, die für diese dynamische Gruppe definiert sind, wie in [Fig. 18](#) gezeigt ist. Diese Aktionen geben bei der Ausführung die Menge der Portalressourcen aus den Inhaltgruppen zurück, die für sie in der Regel-Datenbank definiert sind.

[0189] Der nächste Ausführungsschritt ist die Bewertung der Rollen, die diesem Benutzer durch die Rollenmaschine zugewiesen sind. Die Rollenmaschine verwendet die Organisationsmitgliedschaft (die den Benutzerprofileigenschaften entnommen wurden), um der Rollendatenbank die Menge von Inhaltressourcen für diese Benutzerrolle zu entnehmen. Diese Ressourcen werden dann der bereits vorhandenen Liste von Regeln anhand von Portalressourcen, die in der vorhergehenden Menge erzeugt wurden, angefügt.

[0190] Diese Liste wird dann an die Maschine der dynamischen Portalverknüpfung zur Ausführung weitergeleitet. Die Maschine der dynamischen Portalverknüpfung wählt dann die Portalressourcen, die durch diese Liste gekennzeichnet sind, um die Vorzugsportalansicht für diesen gegenwärtigen Benutzer einzurichten.

Zusammenfassung

1. Allgemeine Realisierung der Integration der Back-End-Netzanwendung

[0191] Mit dem http-Client der Portlet-Anwendung und der Sitzung der Portlet-Anwendung ist es nun möglich, dass ein allgemeines Modell der Integration der Back-End-Netzanwendung vorhanden ist. Dieses kann verwendet werden, es möglich zu machen, dass mehrere Portlets innerhalb der gleichen Portlet-Anwendung mit dieser Back-End-Netzanwendung Daten austauschen.

[0192] Diese Realisierungsform der Erfindung ermöglicht Folgendes:

- i. Eine ursprüngliche Portlet-Integration, ohne dass separate Browser gestartet werden und ohne dass mehrere Eingabeaufforderungen für eine Benutzererkennung und ein Kennwort erforderlich sind, um auf diese Netzanwendung zuzugreifen.
- ii. Das Erzeugen mehrerer Anforderungen und das Empfangen von Antworten an die Back-End-Anwendung bzw. von dieser mit einer Sitzungsverwaltung.

2. Einfaches allgemeines System, das zu einem einfachen Arbeitsverfahren führt

[0193] Die vorliegende Erfindung stellt ein einfaches und schnelles Verfahren bereit, um Portlet-Anwendungen mit einer vorhandenen Netzanwendung, die an einem Back-End-Server betrieben werden, zu integrieren, wobei lediglich die Spezifikation der URL der betreffenden Back-End-Netzanwendung in dem Verwendungsdeskriptor der Portlet-Anwendung erforderlich ist. Damit ist es nun möglich, ein Arbeitsverfahren aufzubauen, um die allgemeinen Aufgaben der Integration zu berücksichtigen.

3. Portlets innerhalb der Portlet-Anwendung verwenden allgemeine Sitzungen und Sitzungsdaten gemeinsam

[0194] Die Realisierung eines Sitzungsobjekts der Portlet-Anwendung ermöglicht, dass Portlets der gleichen Portlet-Anwendung allgemeine Daten, die innerhalb einer Portlet-Anwendung eindeutig sind, untereinander gemeinsam verwenden, während sie gleichzeitig von denen der ursprünglichen http-Sitzung des Portalservers verschieden sind.

4. Portalsitzung und Back-End-Sitzung verwenden allgemeine Sitzungsdaten gemeinsam

[0195] Die Realisierung der Sitzungsweiterleitung ermöglicht die gemeinsame Verwendung von allgemeinen Sitzungsdaten zwischen einem Portalserver und seiner Back-End-Netzanwendung. Das ermöglicht, dass die Back-End-Netzanwendung Informationen vom Portalserver empfängt, wodurch ermöglicht wird, dass eine Geschäftslogik der Netzanwendung diese Informationen ausnutzt, die vom Portalserver weitergegeben werden.

[0196] Wenn z. B. der gegenwärtige Portlet-Zustand darin besteht, die maximierte Ansicht des Portlets anzuzeigen, kann die Back-End-Netzanwendung diese Information empfangen und diese vorteilhaft verwenden, indem sie detaillierte Geschäftsinformationen zurücksendet im Unterschied zur normalen Ansicht eines Portlets, bei der die Back-End-Netzanwendung lediglich eine zusammenfassende Version der Informationen senden würde.

5. Zusammenhängende Sitzung der Back-End-Netzanwendung

[0197] Im Unterschied zum Portalserver mit der Sitzung der Portlet-Anwendung, dem Sitzungsobjekt der Portlet-Anwendung, dem Portlet-http-Client und dem Mechanismus der Sitzungsweiterleitung kann eine Back-End-Netzanwendung nun ihre eigene Sitzung, die von der des Portalservers verschieden ist, aufrechterhalten, jedoch trotzdem die gleichen Cookies wie die des Portalservers gemeinsam verwenden. Die Back-End-Netzanwendung kann nun unabhängig und korrekt betrieben werden, wobei sie Portlet-Anforderungen von verschiedenen Portlets in einem Portal als ein virtueller Client wahrnimmt, wodurch eine zusammenhängende Sitzung mit der Back-End-Netzanwendung ermöglicht wird.

6. Einmalanmeldung (Single Sign On) über Portalserver und Back-End-Netzanwendung

[0198] Die Ausführungsform der Sitzungsweiterleitung stellt die Einmalanmeldung-Funktion bereit, so dass ein Benutzer, nachdem er bei einem Portalserver angemeldet ist, nicht gezwungen ist, Benutzer-Berechtigungs-nachweise erneut zu übermitteln, um sich bei der betreffenden Back-End-Netzanwendung anzumelden. Das wird mittels einer Cookie-Tabelle mit einer eindeutigen Abbildung zwischen der http-Sitzung auf das Portal und der http-Sitzung vom Portlet-http-Client auf die Back-End-Netzanwendung ermöglicht.

7. Verhalten der Back-End-Netzanwendung, das mit dem des Portalservers synchronisiert ist

[0199] Die Ausführungsform der Sitzungsweiterleitung ermöglicht eine nahtlose Integration, indem das Verhalten der Back-End-Netzanwendung synchronisiert wird, indem die Sitzungsinformationen von der Portalsitzung an die Sitzung der Back-End-Netzanwendung weitergeleitet wird.

[0200] Das Folgende sind einige Beispiele:

[0201] Die Spracheinstellung und lokale Einstellung in einem Portalserver kann nun an seine Back-End-Netz-anwendung geleitet werden, so dass die Back-End-Anwendung nun eine Antwortnachricht anhand der lokalen und Spracheinstellungen des Portalservers bilden kann.

[0202] Ein weiteres Beispiel besteht darin, dass Sitzungsablaufinformation von dem Portalserver nun zur Sitzung der Back-End-Netzanwendung geleitet werden können, so dass bei der Sitzung der Back-End-Netzan-

wendung nun zur gleichen Zeit wie bei der Portalsitzung eine Zeitüberschreitung erfolgt. Die Back-End-Netz-anwendung kann nun auf den Portalzustand und die Ereignisse reagieren, die vom Portalserver weitergeleitet werden.

8. Synchronisierter Inhalt innerhalb der gleichen Portalseite

[0203] Eine Gruppierung von Portlets für dynamischen Inhalt ermöglicht eine Zusammenarbeit zwischen Portlets innerhalb der gleichen Gruppe für dynamischen Kontext, um einen Geschäftsprozess sowie eine Integration und Synchronisation von Informationen zu erreichen.

[0204] Jedes Portlet kann an mehreren Gruppen für dynamischen Kontext teilnehmen. Das schafft ein sehr offenes und einfaches Programmierungsmodell für Portaladministratoren, um Portlets in Gruppen von Portlets für dynamischen Kontext zu gruppieren.

[0205] Die einfache Struktur der Definition von dynamischem Kontext ermöglicht, dass eine einfache Verarbeitung für eine automatische Erzeugung von Master- und Slave-Portlets für dynamischen Kontext für jede Gruppierung gebildet werden kann.

[0206] Eine Realisierung der Definition von dynamischem Kontext, eine Gruppe für dynamischen Kontext, eine Realisierung von Master-Portlet und Slave-Portlet (einschließlich die Slave-Aufgaben, Slave-Kontextabbildung) helfen bei der Bereitstellung von Vorteilen der Erfindung.

9. Fähigkeit zum Definieren einer Auffrischungssequenz von Portlets

[0207] Der Transaktionsmanager stellt die Möglichkeit bereit, eine Auffrischungssequenz von Portlets beim ersten Mal zu definieren. Die Fähigkeit zum Definieren einer Auffrischungssequenz von Portlets ermöglicht eine geeignete Realisierung einer sequenziellen Geschäftslogik unter Verwendung der Portal-/Portlet-Architektur. Der Transaktionsmanager, die Ressourcensorientierung und die Zwischenspeicherung von Antworten helfen bei der Bereitstellung von Vorteilen der Erfindung.

10. Regelbasierte und rollenbasierte Verknüpfung

[0208] Eine feinabgestimmte Portal-Personalisierung kann gegenwärtig nur durch eine dynamische Verknüpfung erreicht werden. Diese unterscheidet sich deutlich von der Realisierung von regulären Netzanwendungen nach dem Stand der Technik, bei denen es kein formales Konzept von Portlets, Seiten oder Seitengruppen gibt, das gemäß der vorliegenden Erfindung angewendet wird. Eine feinabgestimmte Personalisierung wird immer wichtiger, wenn der Portalmarkt erfolgreich ist und Benutzeranforderungen für eine feinabgestimmte Kampagnenzielrichtung usw. aufkommen.

[0209] Die Ausführungsformen der Erfindung stellen mehrere Vorteile bereit, die im Folgenden aufgeführt sind:

1. Der Grad der Personalisierung, der durch diesen Lösungsansatz erreicht werden kann, ist viel feiner abgestuft als bei Portlet-Administrationseinrichtungen, die gegenwärtig durch Portalserver bereitgestellt werden. Die Portlet-Administrationseinrichtungen, die gegenwärtig zur Verfügung stehen, stellen naturgemäß eine manuelle Konfiguration dar. Nachdem sie konfiguriert wurde, ist sie statisch und ändert sich während der Laufzeit nicht. Die vorliegende Erfindung stellt dynamische Möglichkeiten bereit, um Portalressourcen anhand von Regeln aufzunehmen.
2. Da das Portalverknüpfungsmodul eine dynamische Entität ist, können durch eine direkte Verbindung mit Regel- und Rollenmaschinen Möglichkeiten der dynamischen Echtzeit-Verknüpfung ohne menschlichen Eingriff erreicht werden.
3. Durch eine Personalisierung von grob gegliederten Portalressourcen wie etwa Seiten und Seitengruppen lässt sich ein dynamisches Layout erzielen.
4. Viel wirkungsvollere Kampagnen, Verträge usw. können eingerichtet werden. Das ist sowohl für einen e-Commerce-Verkauf als auch für B2B-Organisationen von besonderer Wichtigkeit.
5. Die Erfindung ermöglicht einen viel höheren Grad der Personalisierung als eine reguläre Personalisierung von Inhalt. Es können z. B. tatsächlich ganze Abschnitte einer Netzseite anhand von Regeln gesperrt werden. Das kann durch eine reguläre Personalisierung nicht erfolgen. Des Weiteren wird keine dynamische Verknüpfung auf die Domäne der regulären Personalisierung angewendet, die inhaltsbezogen und nicht ressourcenbezogen ist.

Patentansprüche

1. Portalserver, der wenigstens eine Portlet-Anwendung enthält, um eine Sammlung zugehöriger Portlets zu verwalten, um Benutzersitzungsinformationen weiterzuleiten, wobei ein Portlet eine sichtbare aktive Komponente einer Portalseite ist, wobei die Portlet-Anwendung enthält:
 - ein Mittel zum Starten eines ersten Portlet (**205**) in Reaktion auf eine Benutzeranforderung nach einem Zugriff auf eine Netzanwendung;
 - ein Mittel zum Erzeugen eines Sitzungsobjekts (**208**) einer Portlet-Anwendung für den Benutzer für das erste Portlet;
 - ein Mittel (**308**) zum Speichern von Parametern von der Anforderung;
 - Mittel zum Erzeugen zusätzlicher, dem ersten Portlet zugehöriger Portlets bei weiteren Anforderungen des Benutzers, um auf die Netzanwendung zuzugreifen;
 - Sitzungsobjekt-Datenspeichereinheit (**302**) einer Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird, um die gespeicherten Parameter für die Zuordnung der zusätzlichen Portlets zu dem Sitzungsobjekt der Portlet-Anwendung zu verwenden; und
 - ein Mittel zum Erzeugen eines Datenaustausch-Client (**209**) der Portlet-Anwendung für einen Datenaustausch mit dem Sitzungsobjekt der Portlet-Anwendung und der Netzanwendung, um Benutzeranforderungen, die von dem ersten und den zusätzlichen Portlets empfangen werden, zur Netzanwendung zu befördern;
 - wobei die Benutzersitzungsinformationen Benutzersitzungsinformationen zum Abbilden der Benutzersitzungsinformationen auf eine entsprechende Sitzung der Netzanwendung enthalten.
2. System nach Anspruch 1, bei dem der Datenaustausch-Client der Portlet-Anwendung die Benutzersitzungsinformationen speichert.
3. System nach Anspruch 1, bei dem Portlet-Anforderungsparameterabbildungen zum Speichern von Daten und Anweisungen von Benutzeranforderungen an die Portlets verwendet werden.
4. System nach Anspruch 1, bei dem die Benutzersitzungsinformationen aus der Gruppe ausgewählt sind, die Folgendes umfasst: Benutzerkennung, Benutzer-Berechtigungsnachweise, Sprachpräferenzen, Sitzungszeitüberschreitungsinformationen, Sitzungskennung zum Zuordnen der Benutzersitzungsinformationen zu einer entsprechenden Sitzung der Netzanwendung.
5. System nach Anspruch 1, bei dem der Datenaustausch-Client der Portlet-Anwendung einen Anforderungspuffer aufweist, um Anforderungen von den zugehörigen Portlets zu speichern, damit der Datenaustausch-Client Daten und Anweisungen für die Netzanwendung bereitstellen kann.
6. System nach Anspruch 5, bei dem der Datenaustausch-Client einen Anforderungspuffer aufweist, um Anforderungen von den Portlet-Anforderungsparameterabbildungen der zugehörigen Portlets zu speichern, damit der Datenaustausch-Client Daten und Anweisungen für die Netzanwendung bereitstellen kann.
7. Verfahren für ein Portalserversystem zum Verwalten einer Sammlung zugehöriger Portlets in Reaktion auf Benutzeranforderungen zum Zugreifen auf eine Netzanwendung, um Benutzersitzungsinformationen zwischen den Portlets und der Netzanwendung weiterzuleiten, wobei ein Portlet eine sichtbare aktive Komponente einer Portalseite ist, wobei das Verfahren die folgenden Schritte umfasst:
 - Starten eines ersten Portlet (**205**) in Reaktion auf eine Benutzeranforderung zum zugreifen auf eine Netzanwendung;
 - Erzeugen eines Sitzungsobjekts (**209**) einer Portlet-Anwendung für den Benutzer für das erste Portlet;
 - Speichern von Parametern von der Anforderung;
 - Erzeugen von zusätzlichen, dem ersten Portlet zugehörigen Portlets bei weiteren Anforderungen des Benutzers, um auf die Netzanwendung zuzugreifen;
 - Verwenden der Sitzungsobjekt-Datenspeichereinheit (**302**) der Portlet-Anwendung, die durch das Sitzungsobjekt der Portlet-Anwendung gesteuert wird, um die gespeicherten Parameter zum Zuordnen der zusätzlichen Portlets zu dem Sitzungsobjekt der Portlet-Anwendung zu verwenden; und
 - Erzeugen eines Datenaustausch-Client (**209**) der Portlet-Anwendung für einen Datenaustausch mit dem Sitzungsobjekt der Portlet-Anwendung und der Netzanwendung, um die Benutzeranforderungen, die von dem ersten und den zusätzlichen Portlets empfangen werden, zu der Netzanwendung zu befördern;
 - wobei die Benutzersitzungsinformationen Benutzersitzungsinformationen zum Zuordnen der Benutzersitzungsinformationen zu einer entsprechenden Sitzung der Netzanwendung enthalten.
8. Gegenstand, der Folgendes umfasst:

ein signaltragendes computerlesbares Medium; und
ein Computerprogrammcodemittel zum Ausführen der Schritte vom Anspruch 7, wenn diese auf einem Computer ausgeführt werden.

Es folgen 21 Blatt Zeichnungen

FIG. 1
Modell der Verkettung von dynamischem Kontext

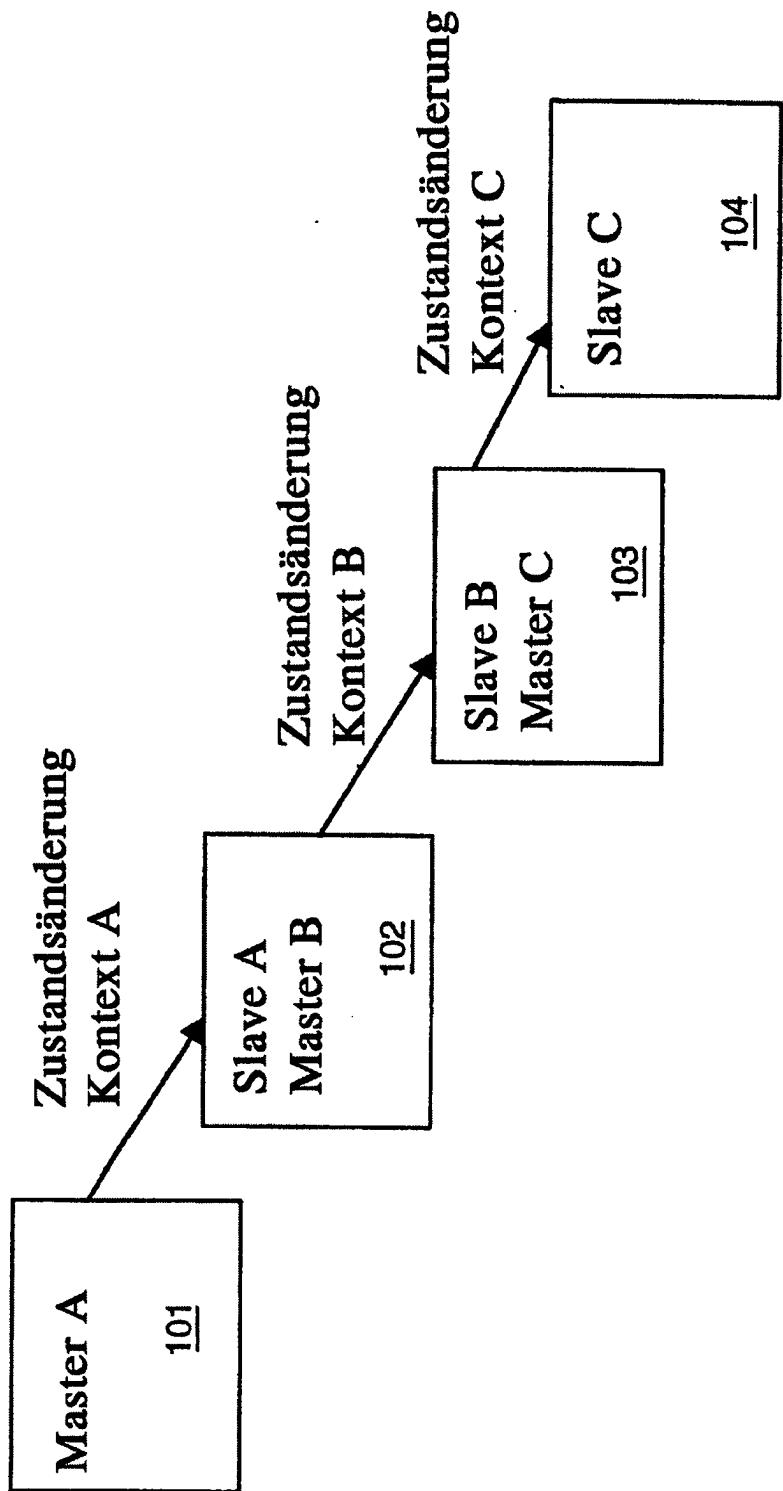


FIG. 2
Integration einer Netzanwendung mit Portal

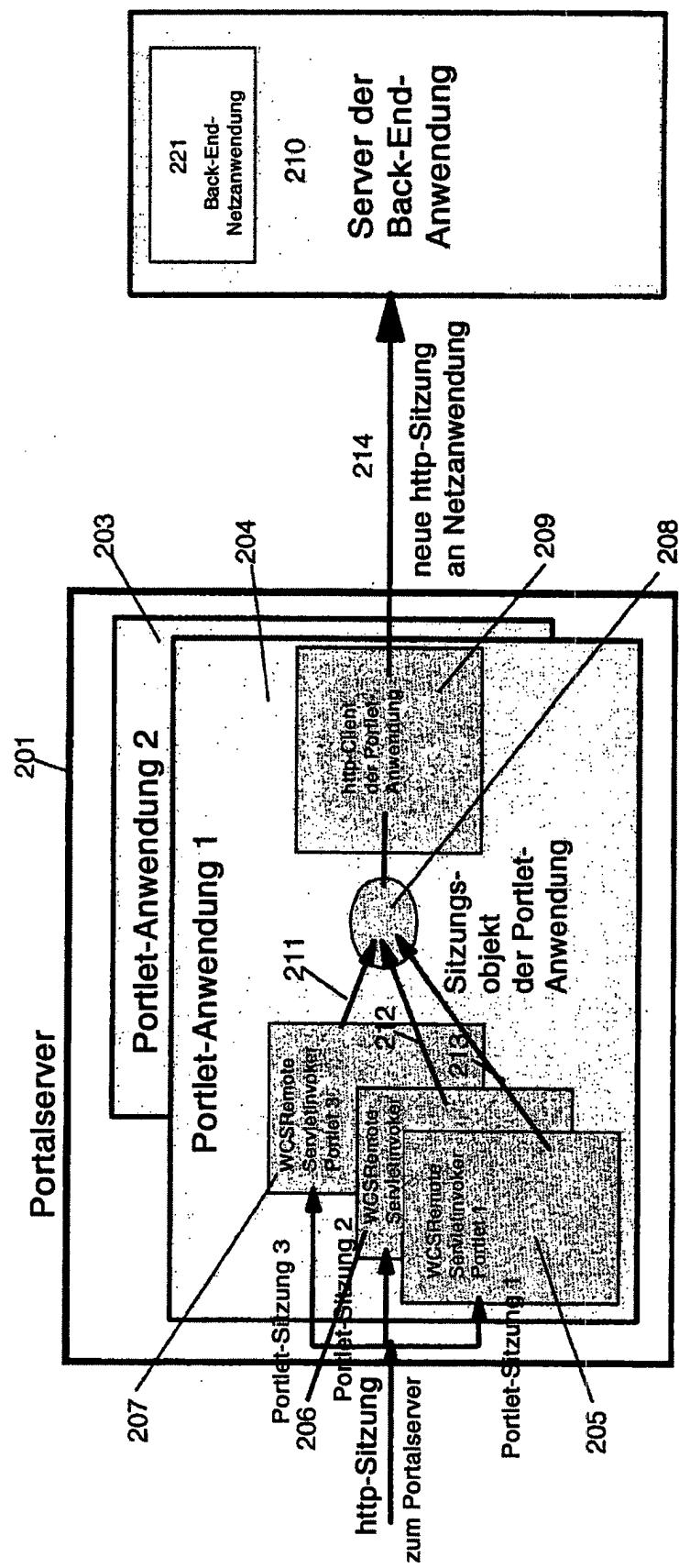


FIG. 3 Strukturelle Darstellung der Integration

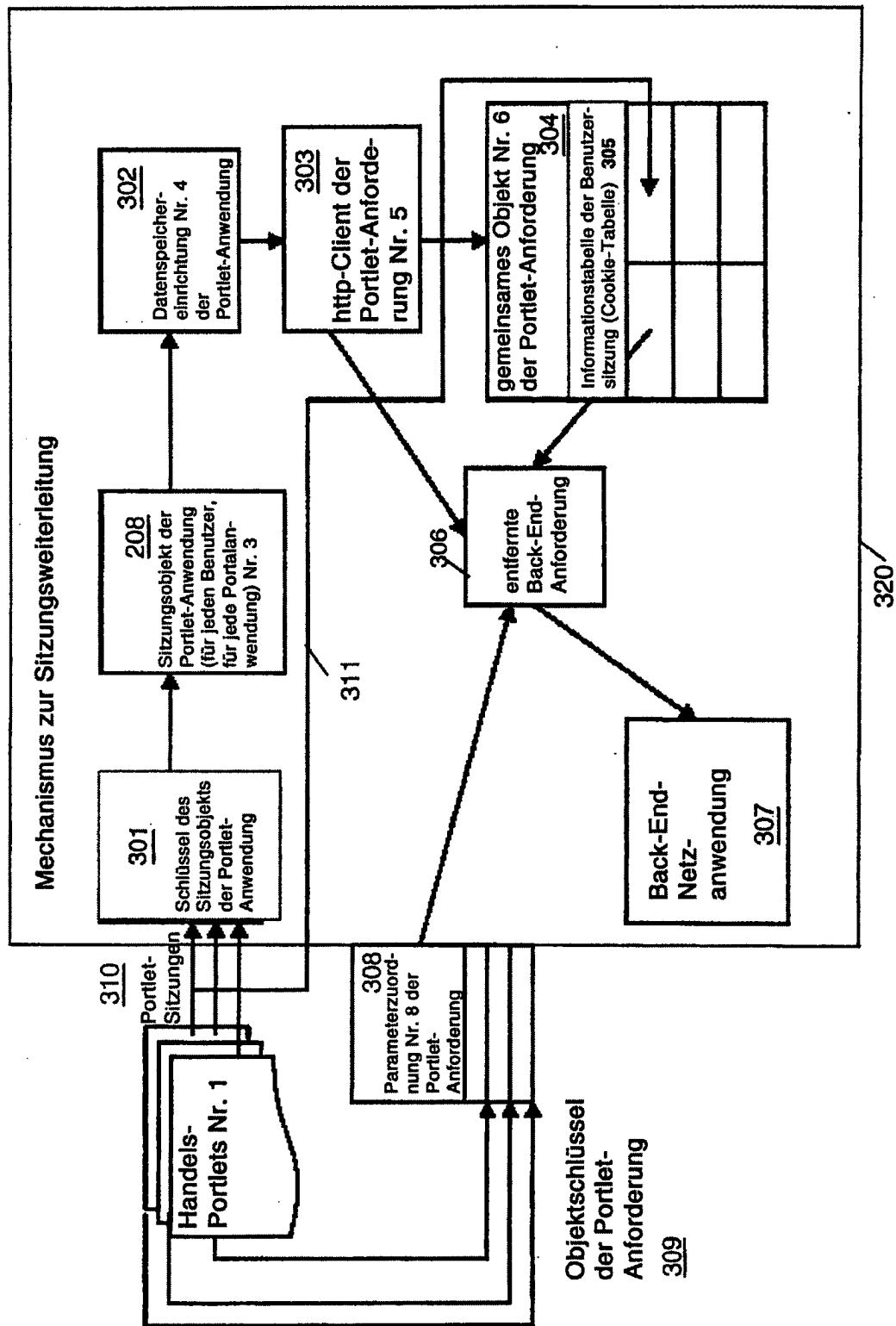


FIG. 4
Ablaufplan der Integration

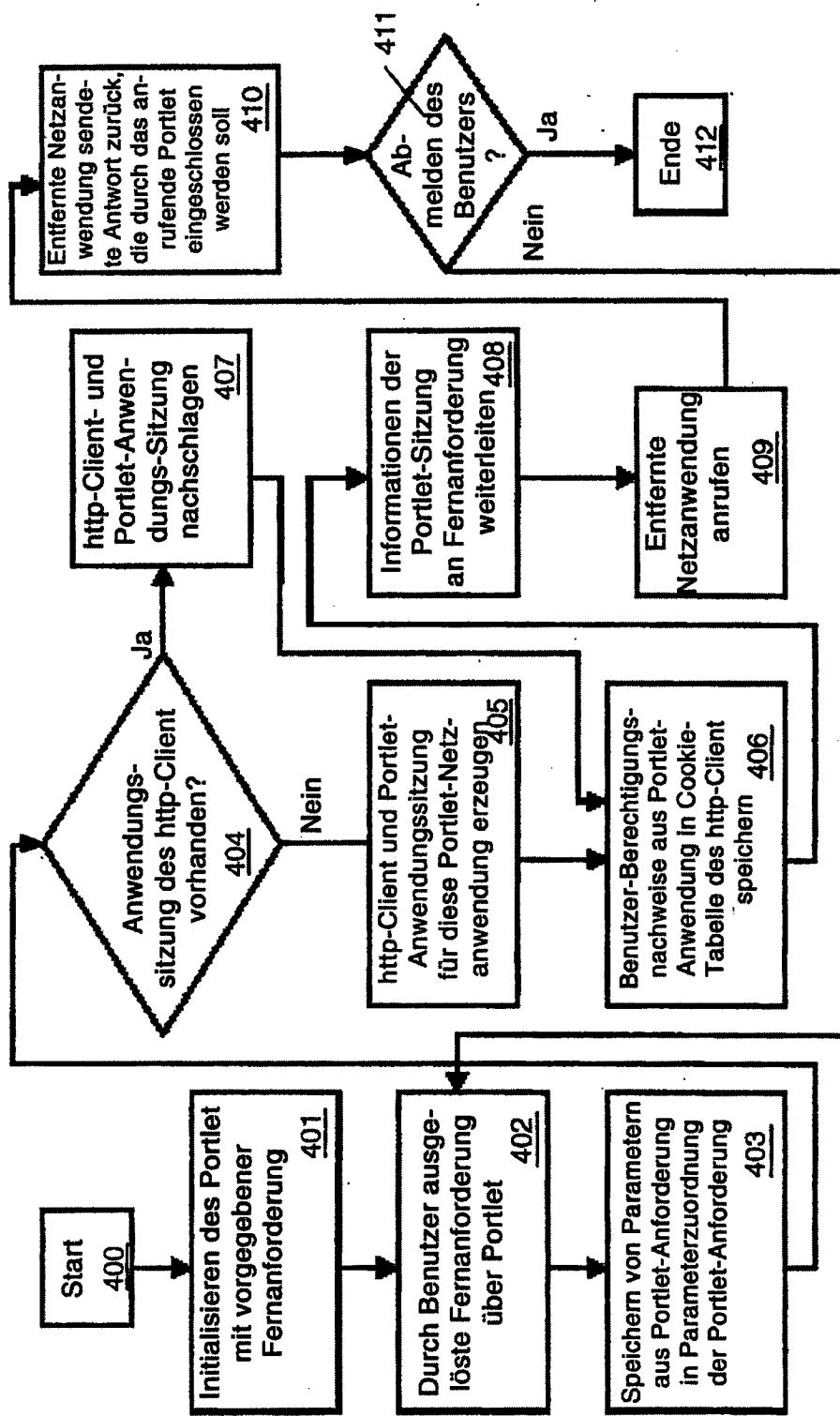


FIG. 5
Strukturelle Darstellung zur Portalintegration mit Back-End-Anwendung

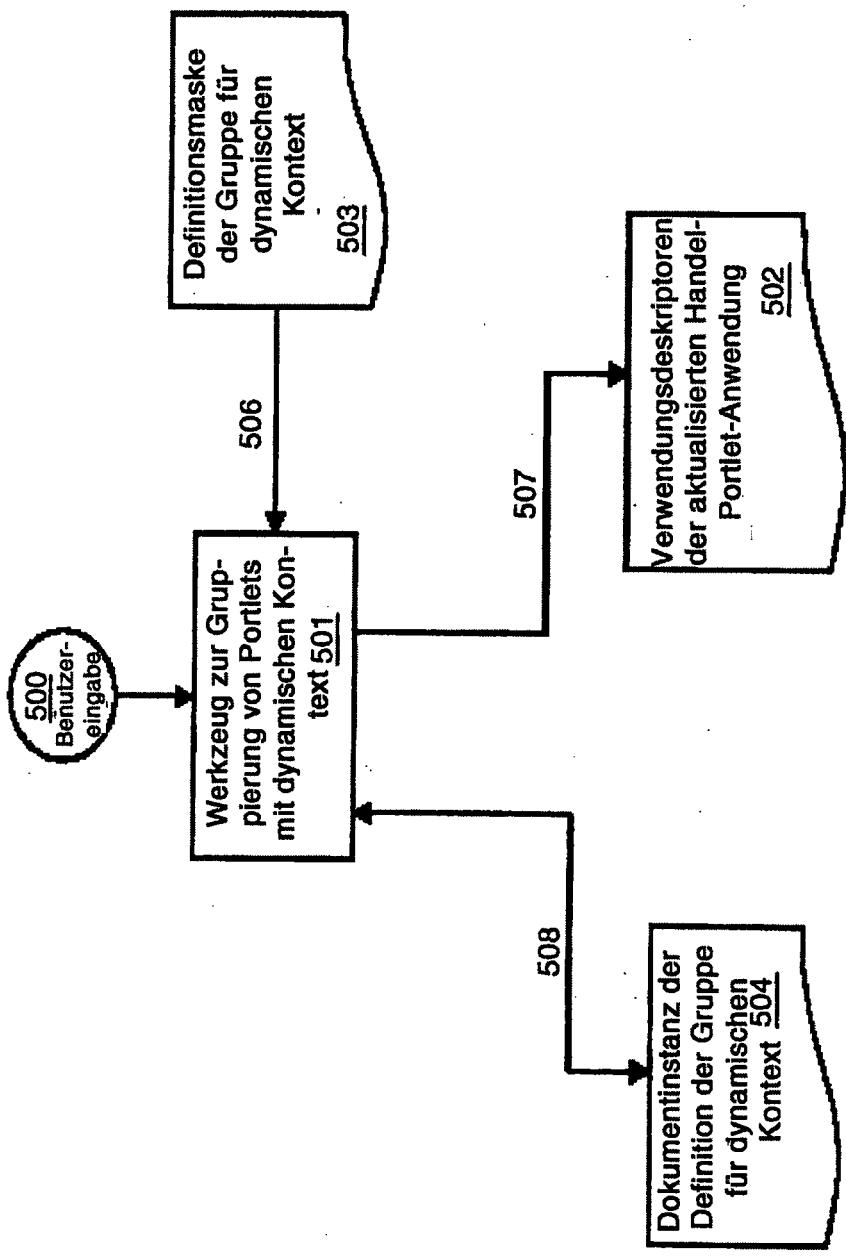


FIG. 6
Ablaufplan zur Integration

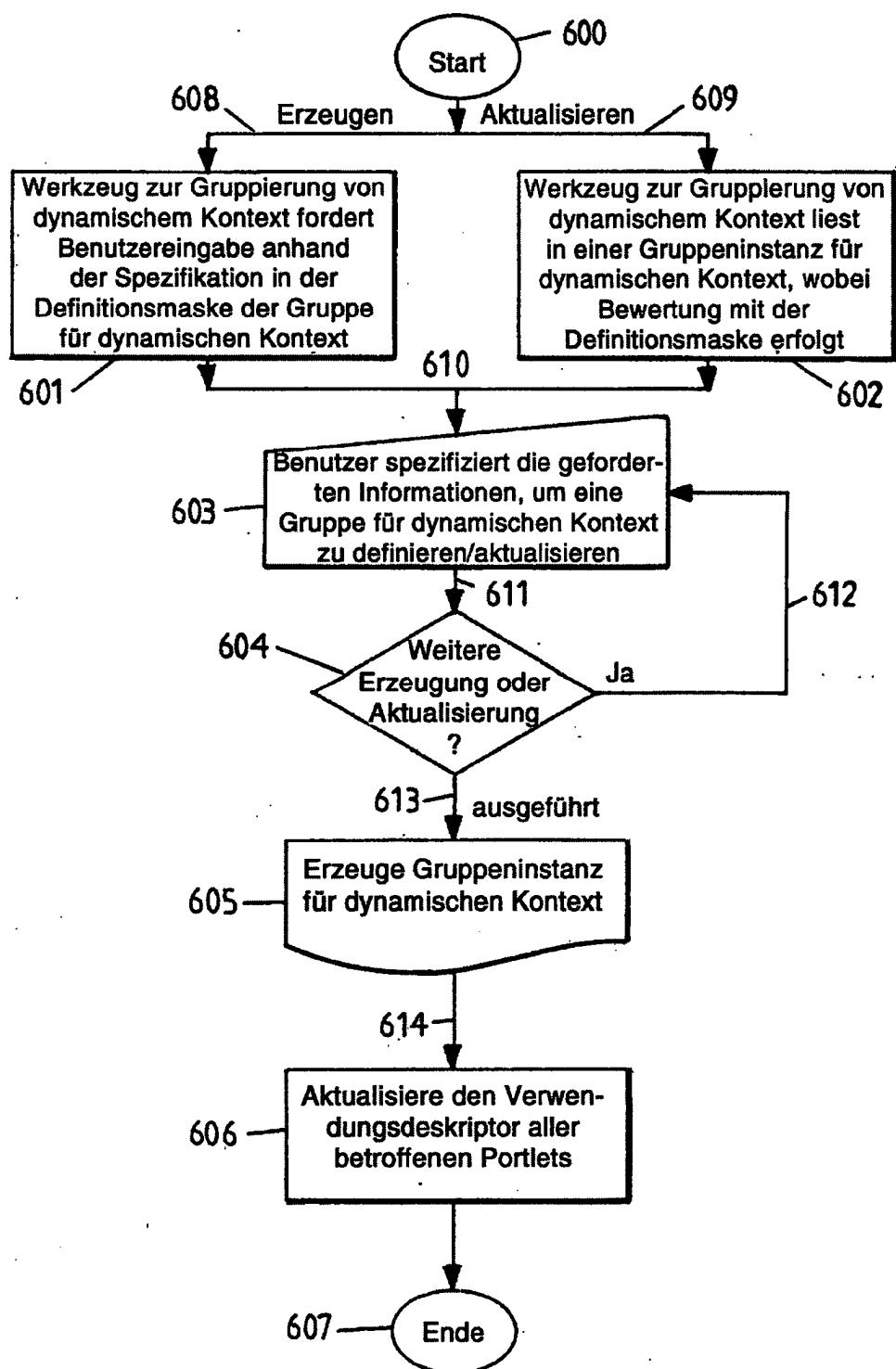
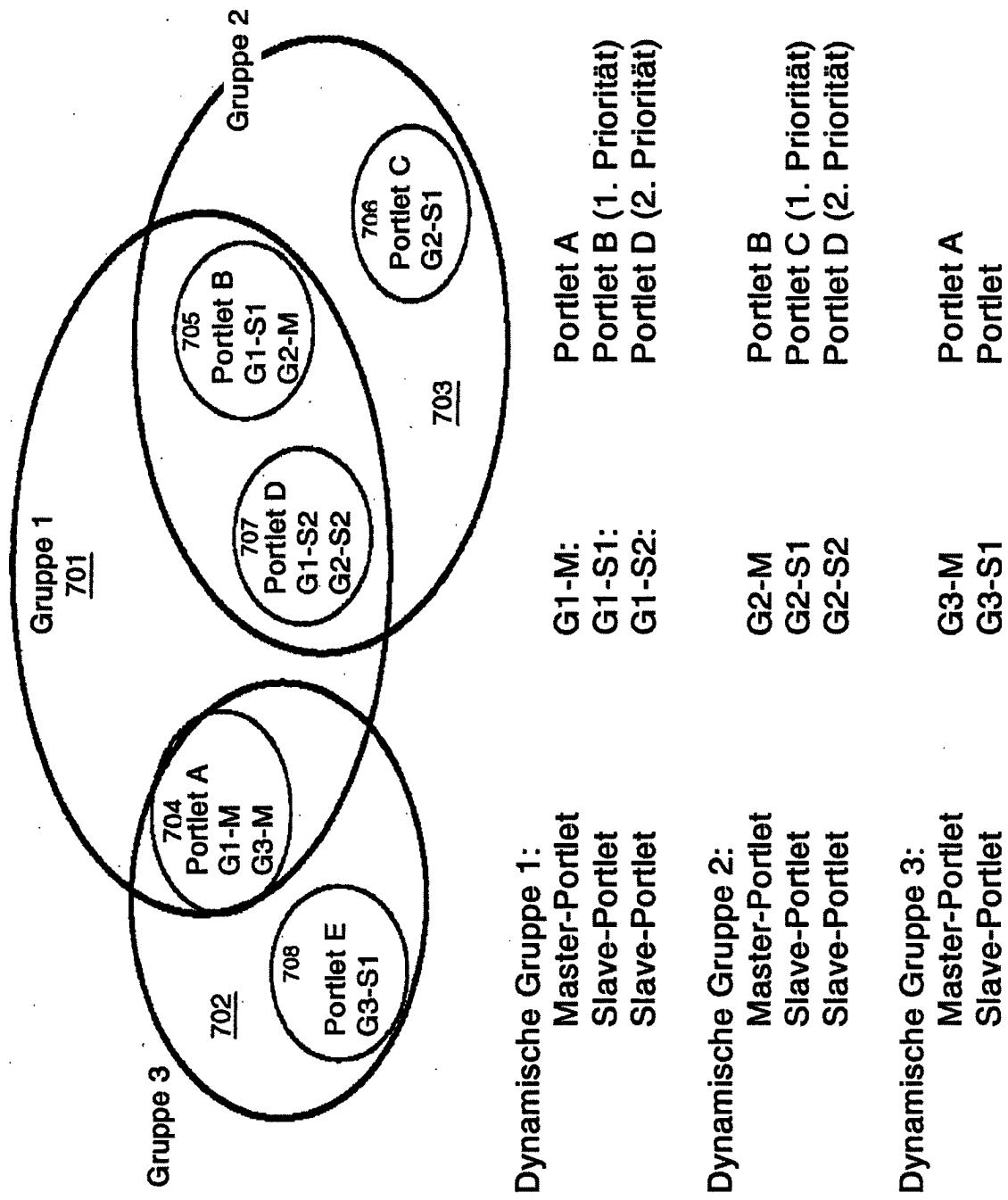


FIG. 7

Beispiel von Gruppen mit dynamischem Kontext für Portlets



Initialisierung einer Portlet-Anwendung für dynamischen Kontext gemäß Spezifikation in der Definitionsinstanz

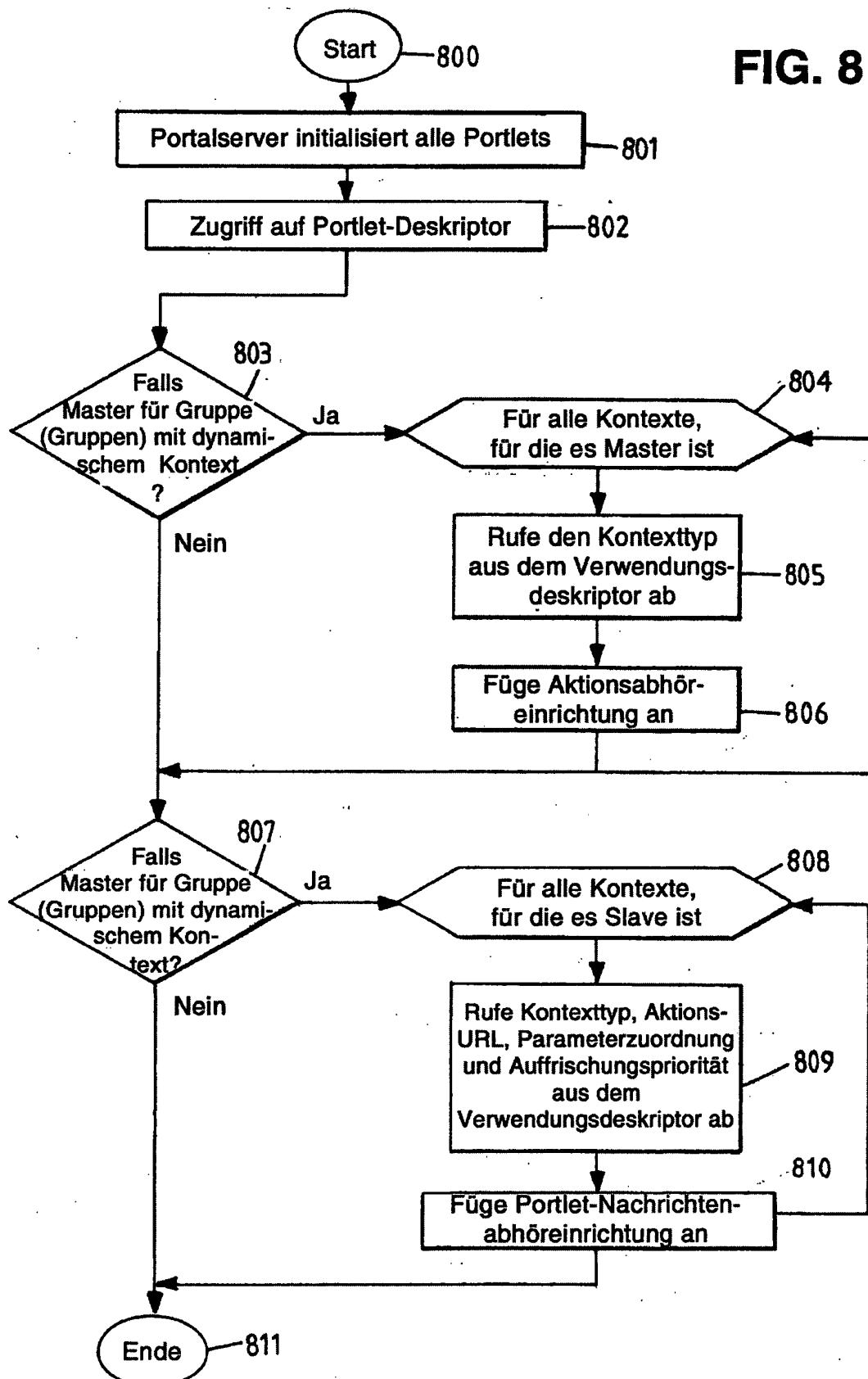


FIG. 8

FIG. 9a

Ablauf der Laufzeit einer Portlet-Gruppe für dynamischen Kontext

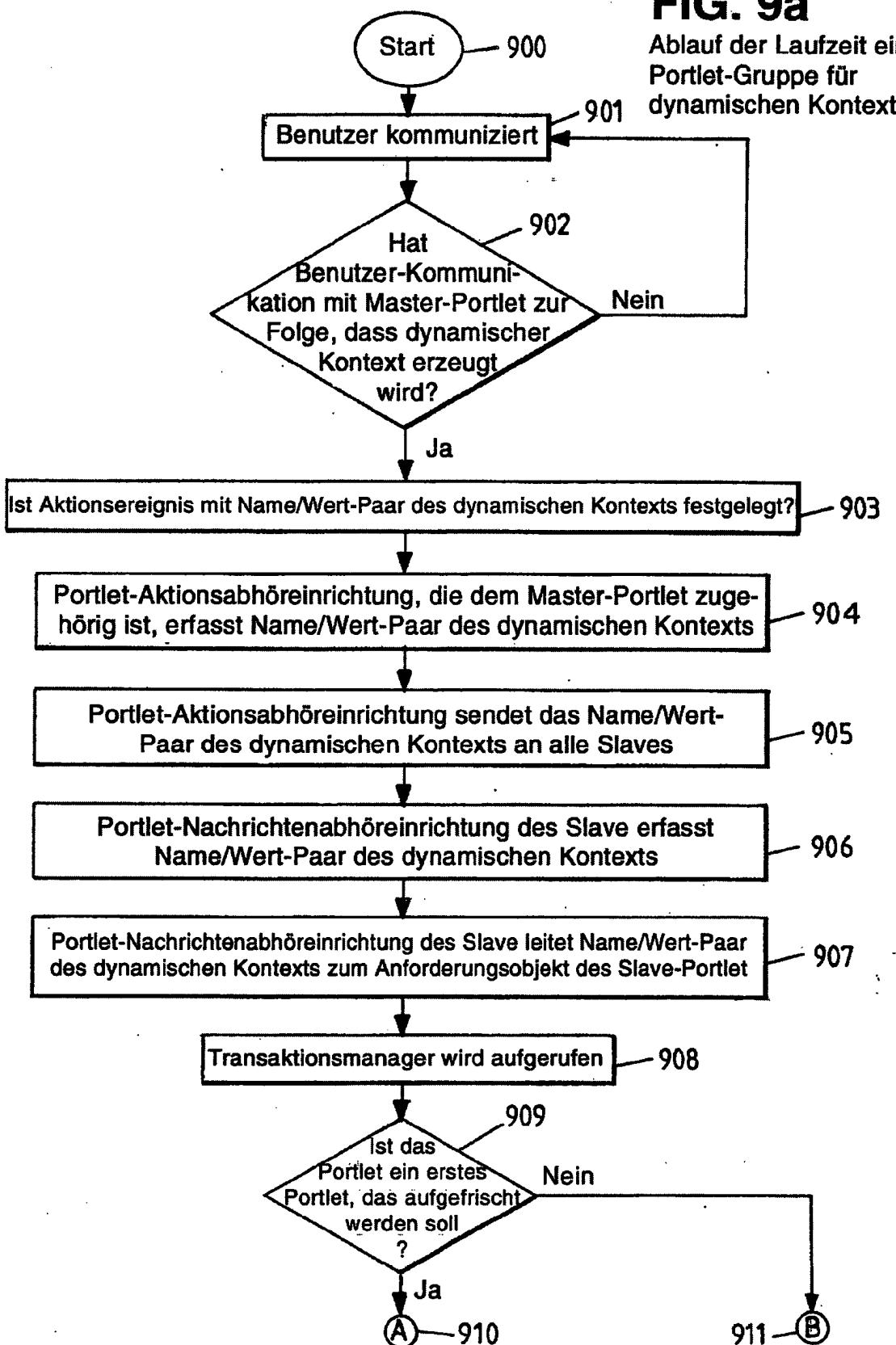


FIG. 9b

Ablauf der Laufzeit einer Portlet-Gruppe
für dynamischen Kontext

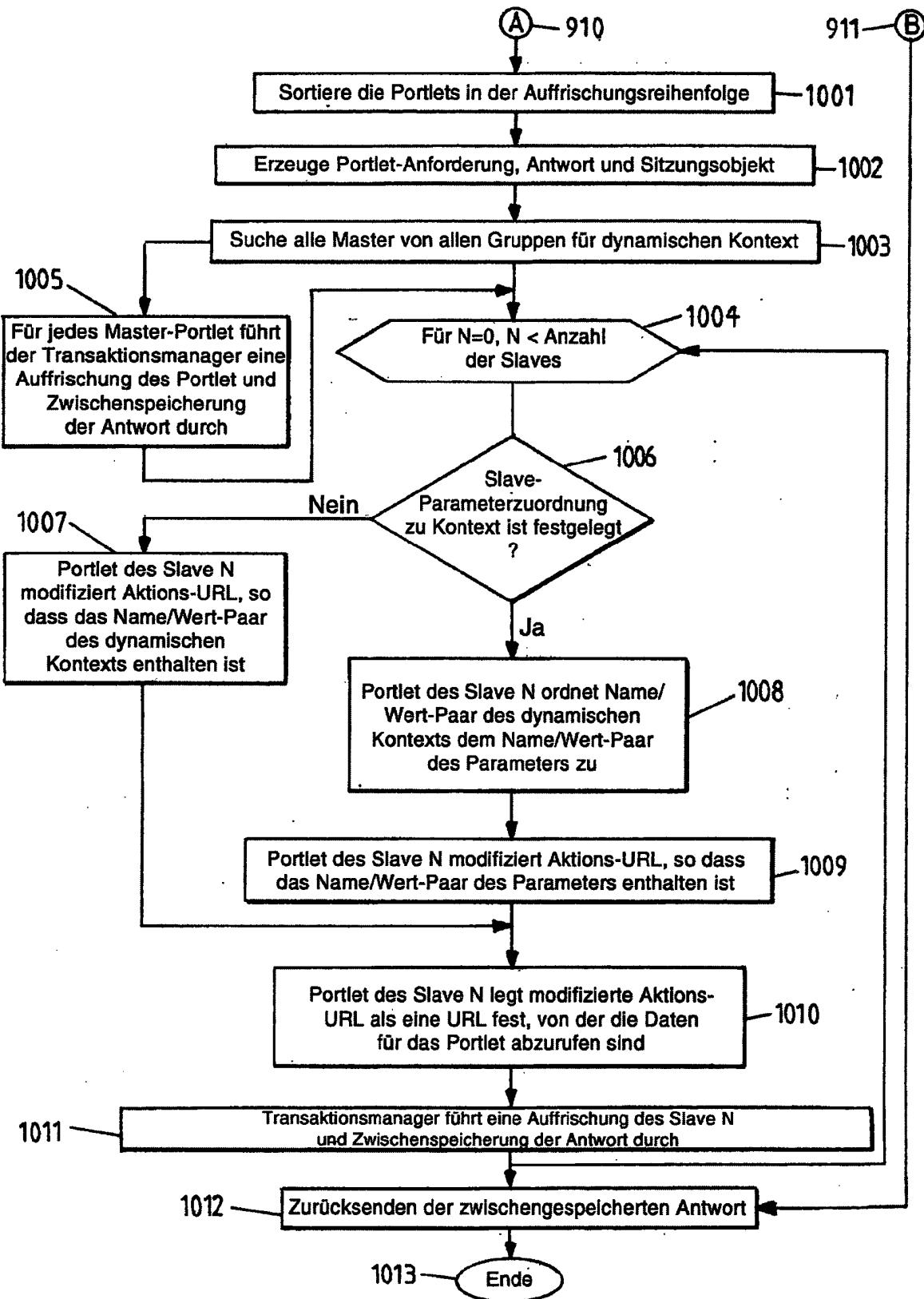
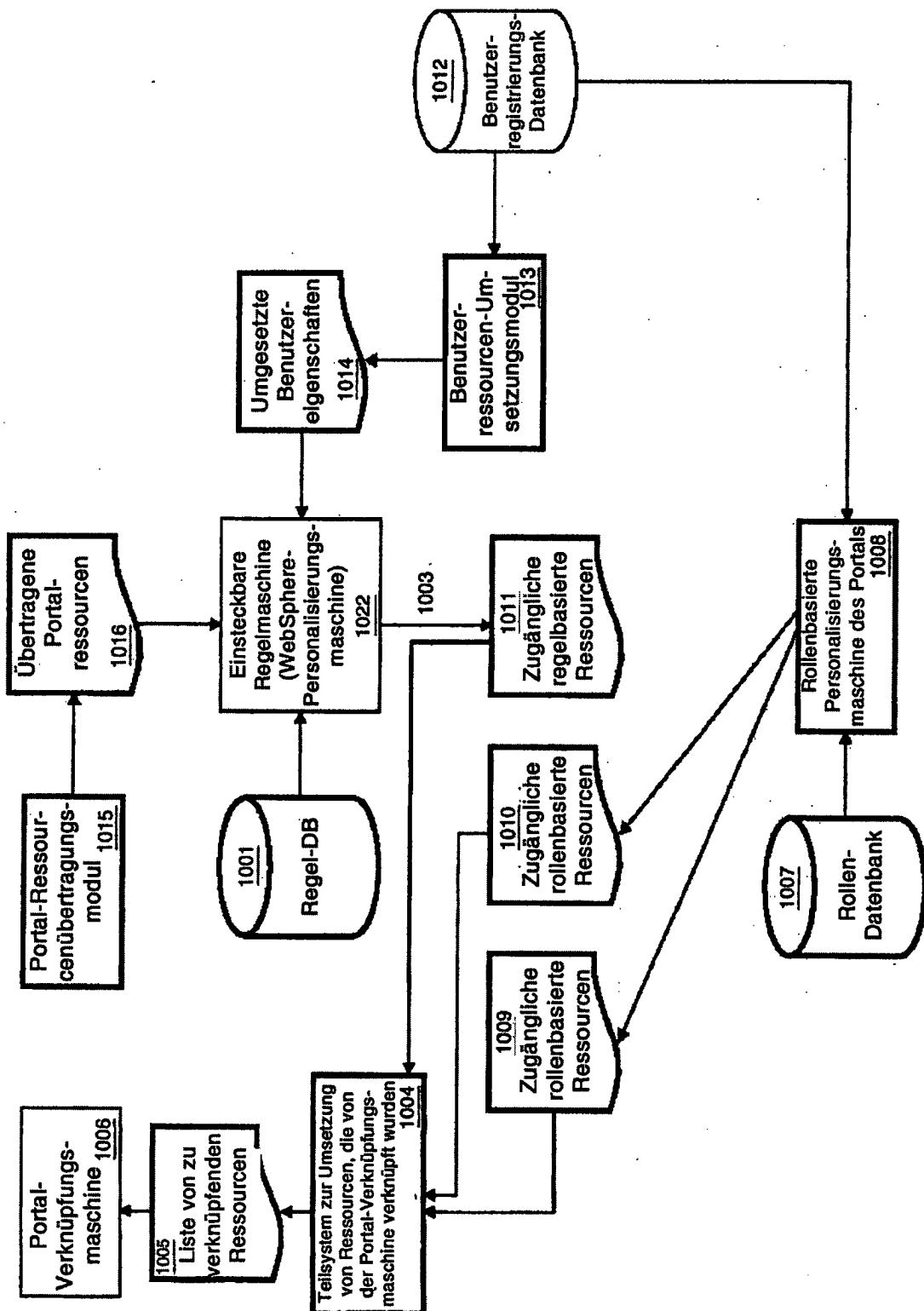
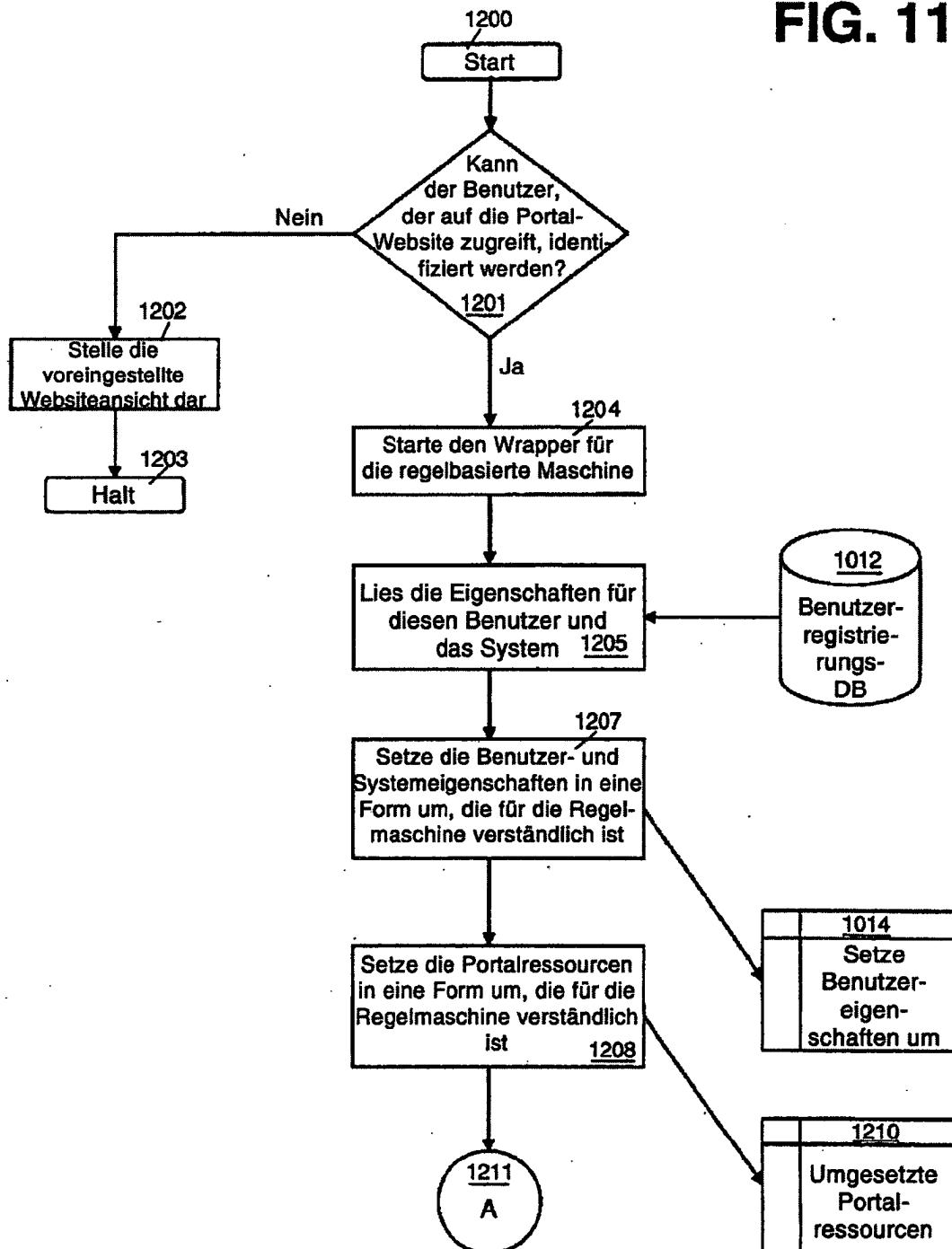


FIG. 10 Strukturelle Abbildung von regelbasierten dynamischen Verknüpfungskomponenten



Ablaufplan der regelbasierten dynamischen Verknüpfung von Komponenten

FIG. 11



Ablaufplan der regelbasierten dynamischen Verknüpfung

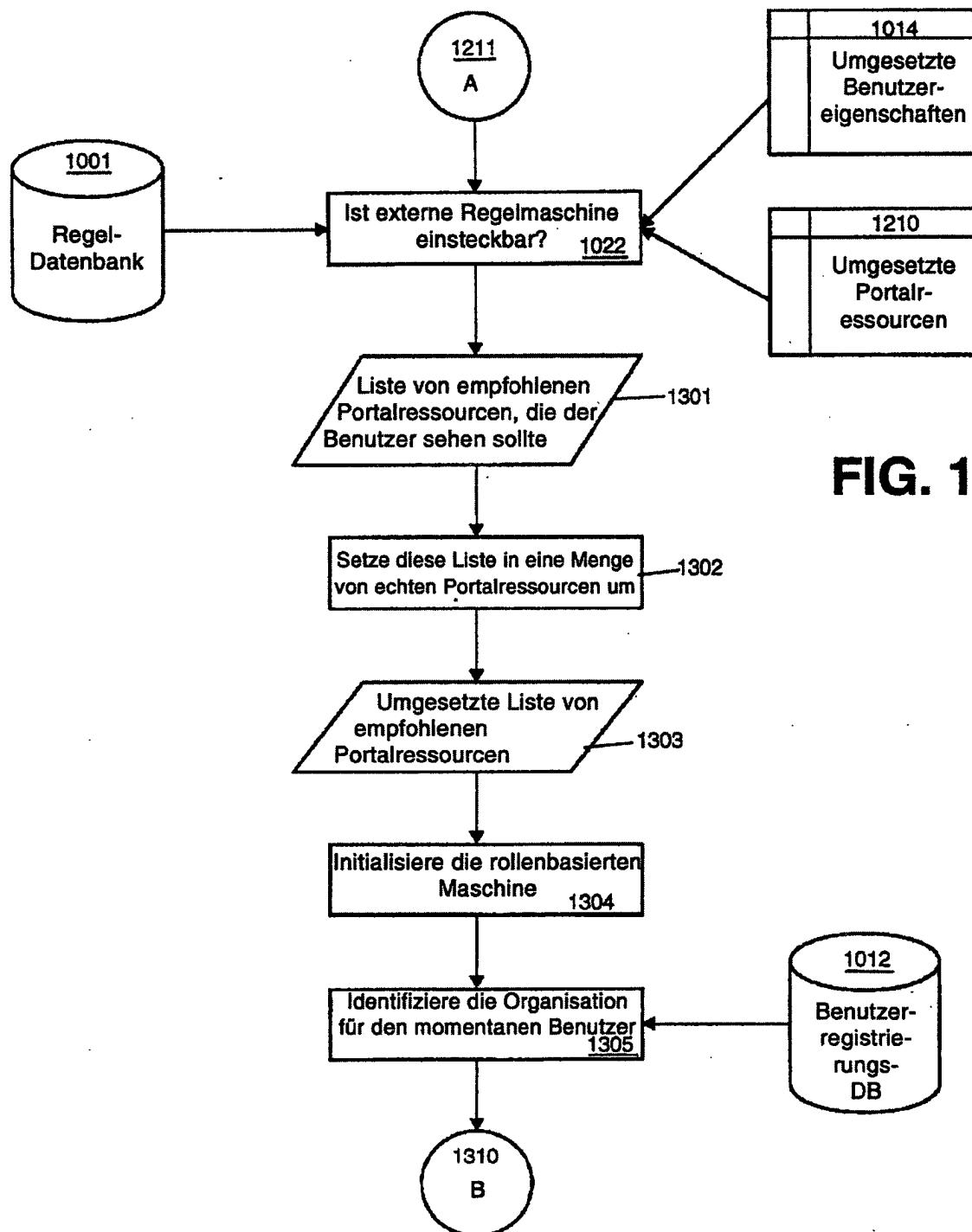


FIG. 12b

Ablaufplan der regelbasierten dynamischen Verknüpfung

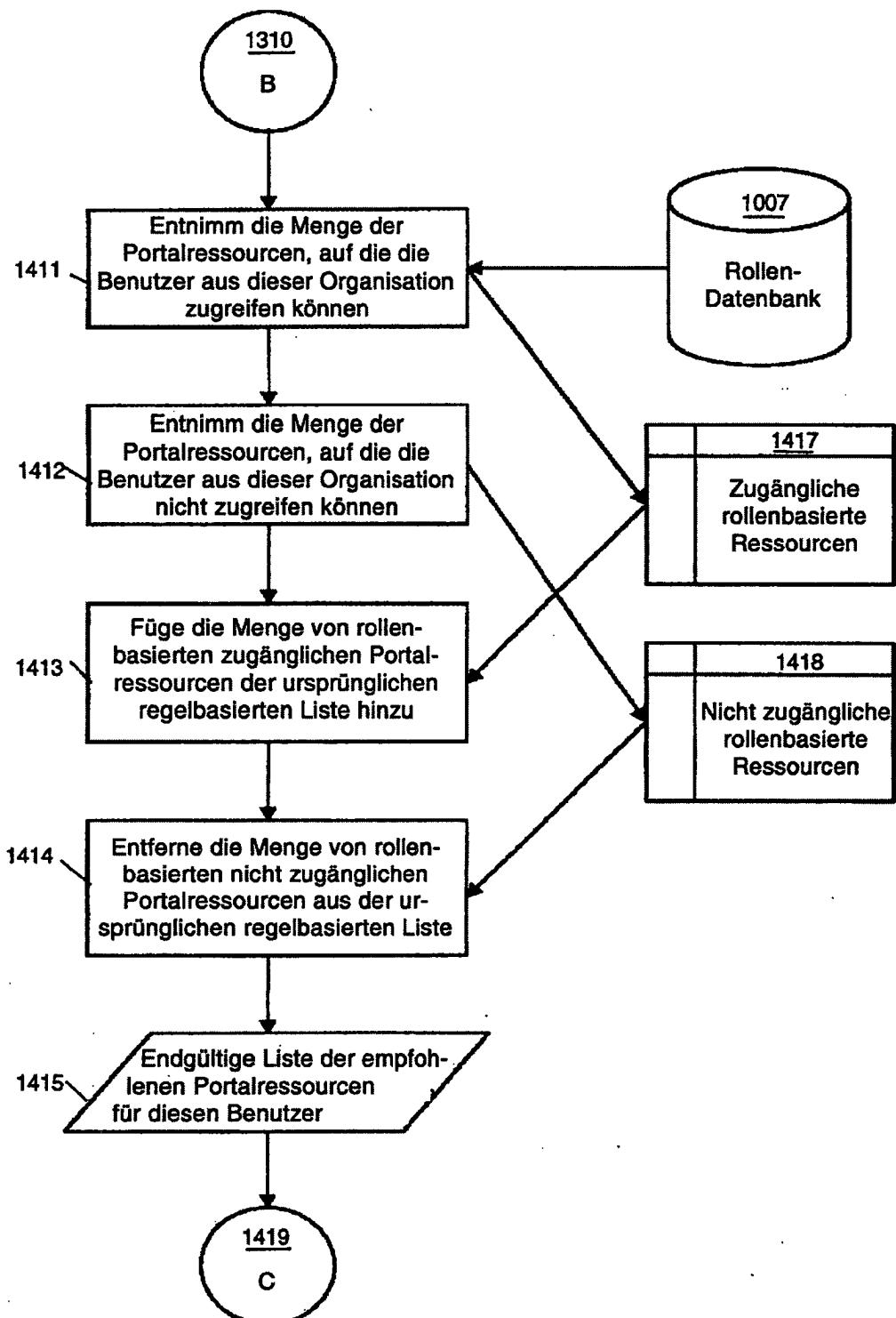
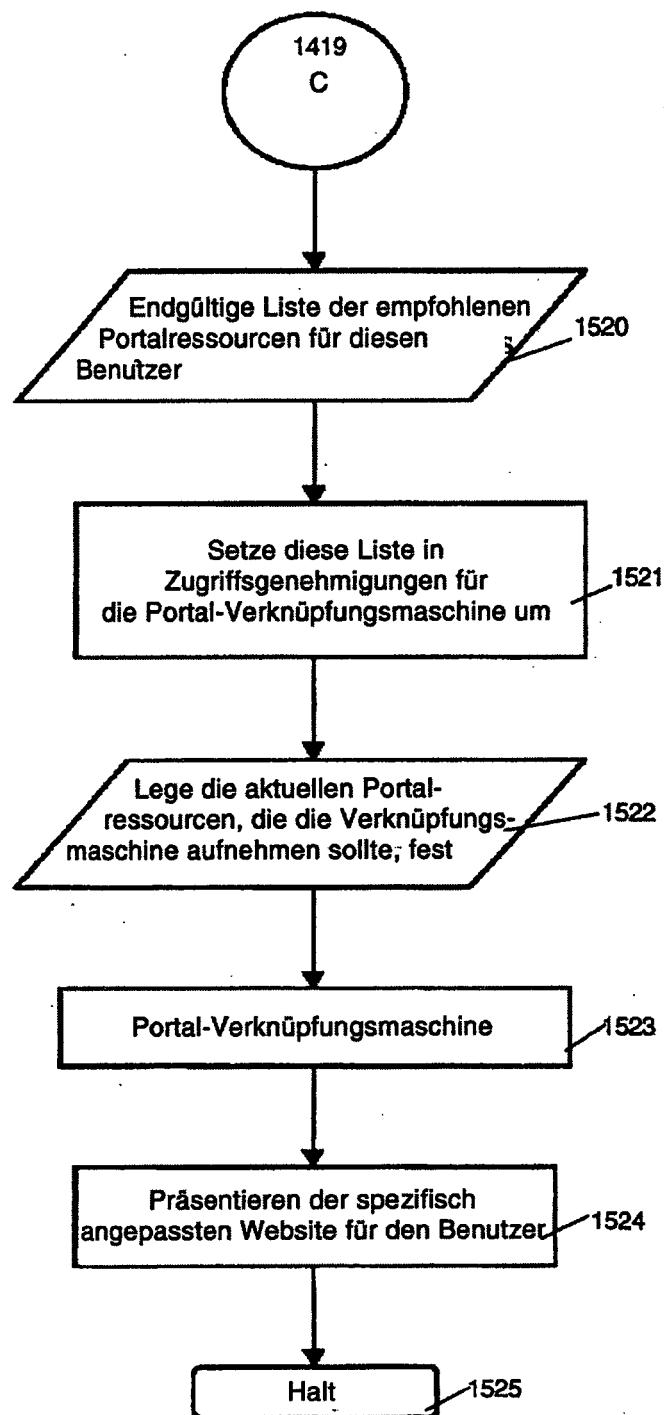


FIG 12c

Ablaufplan der regelbasierten dynamischen Verknüpfung



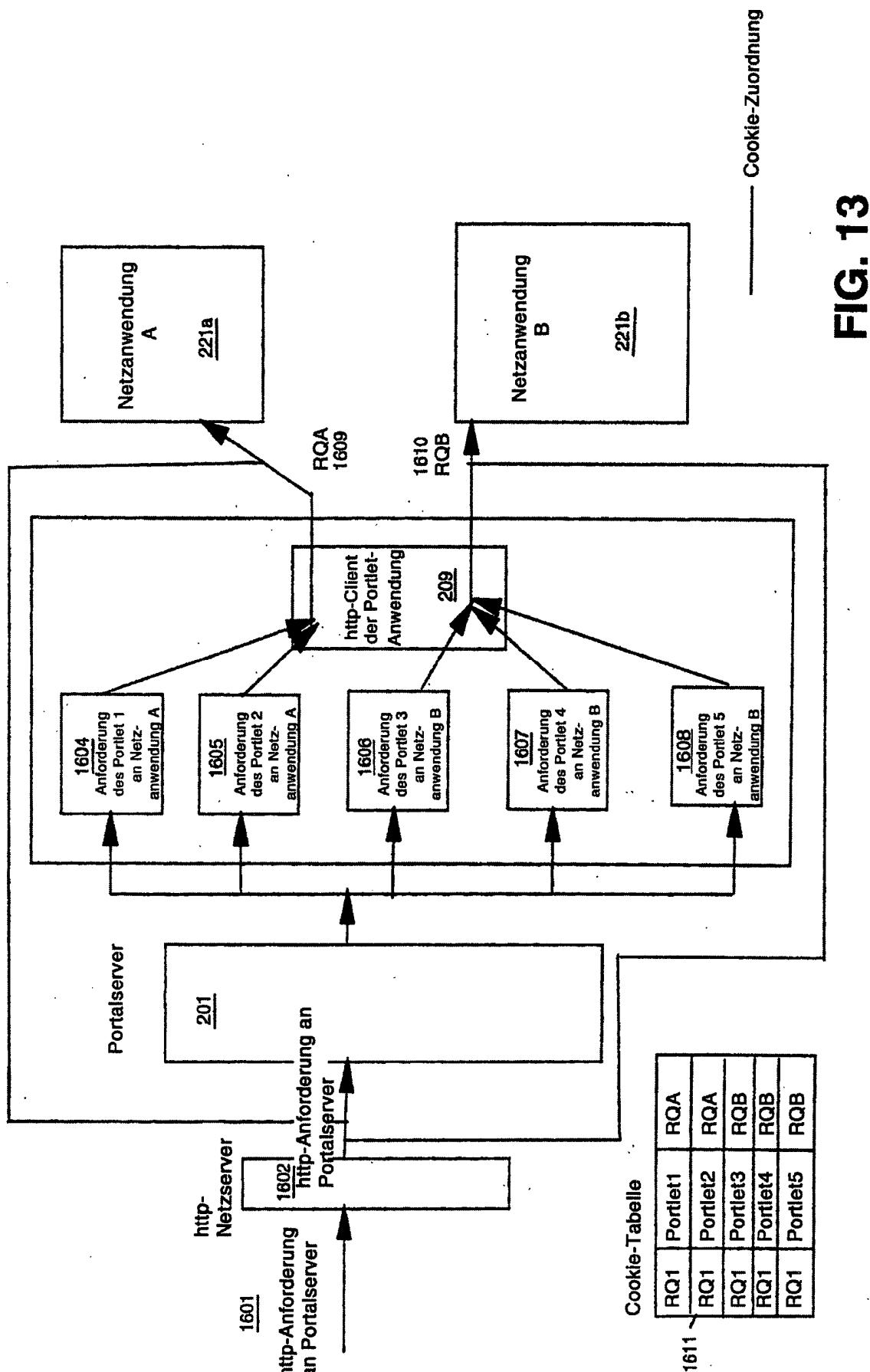


FIG. 13

Synchronisationsmodell-Darstellung
FIG. 14

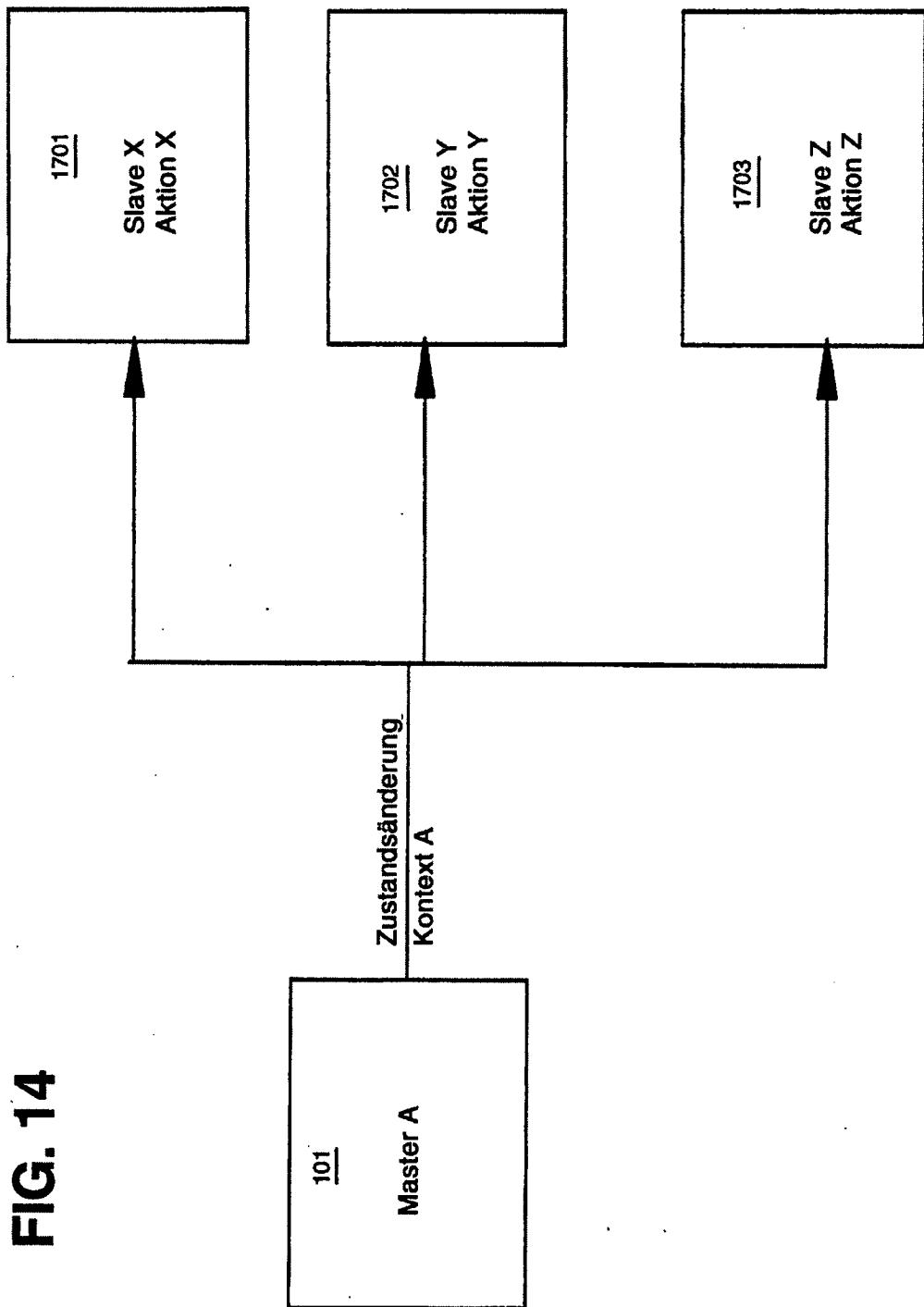


FIG. 15

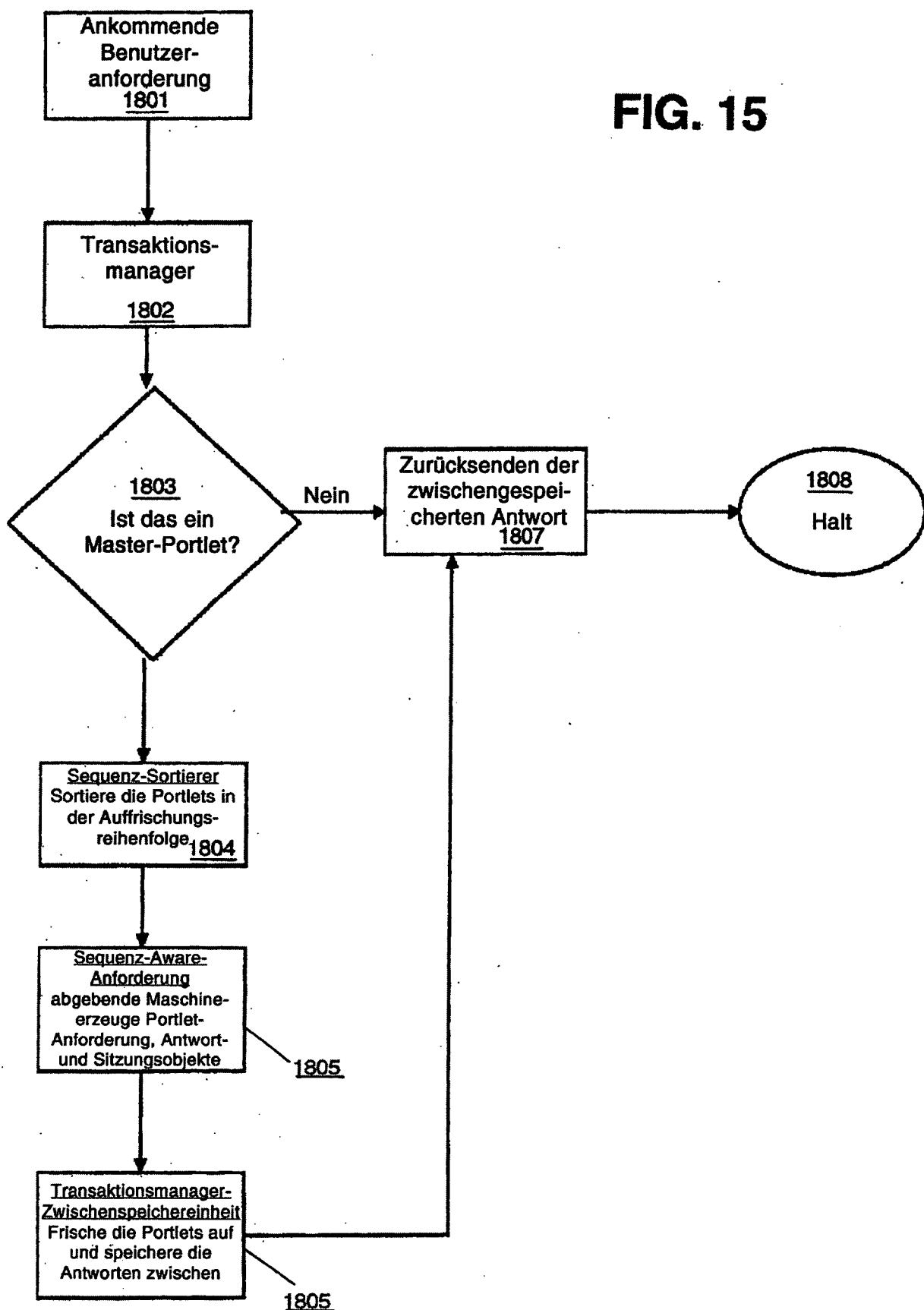


FIG. 16

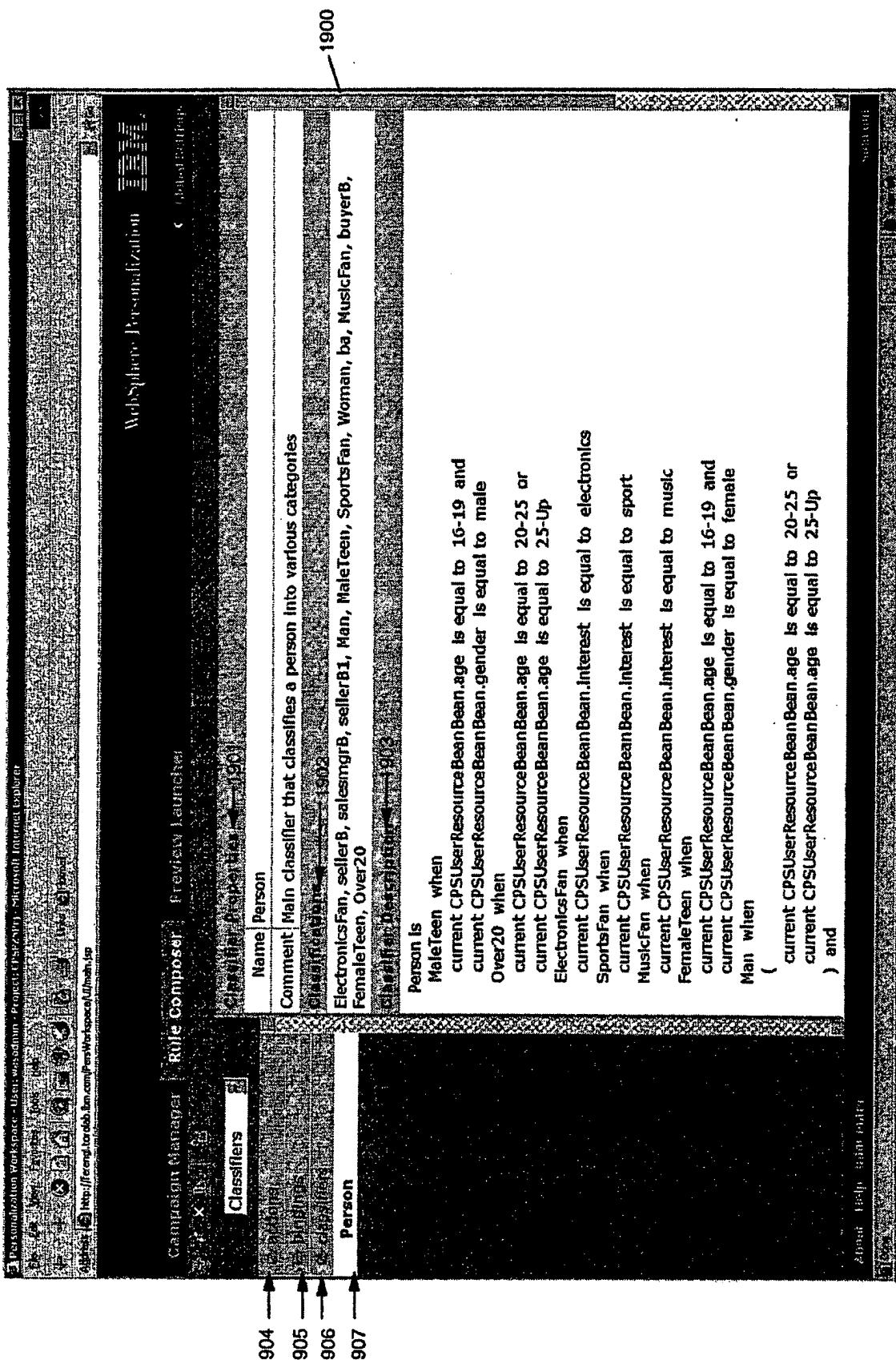


FIG. 17

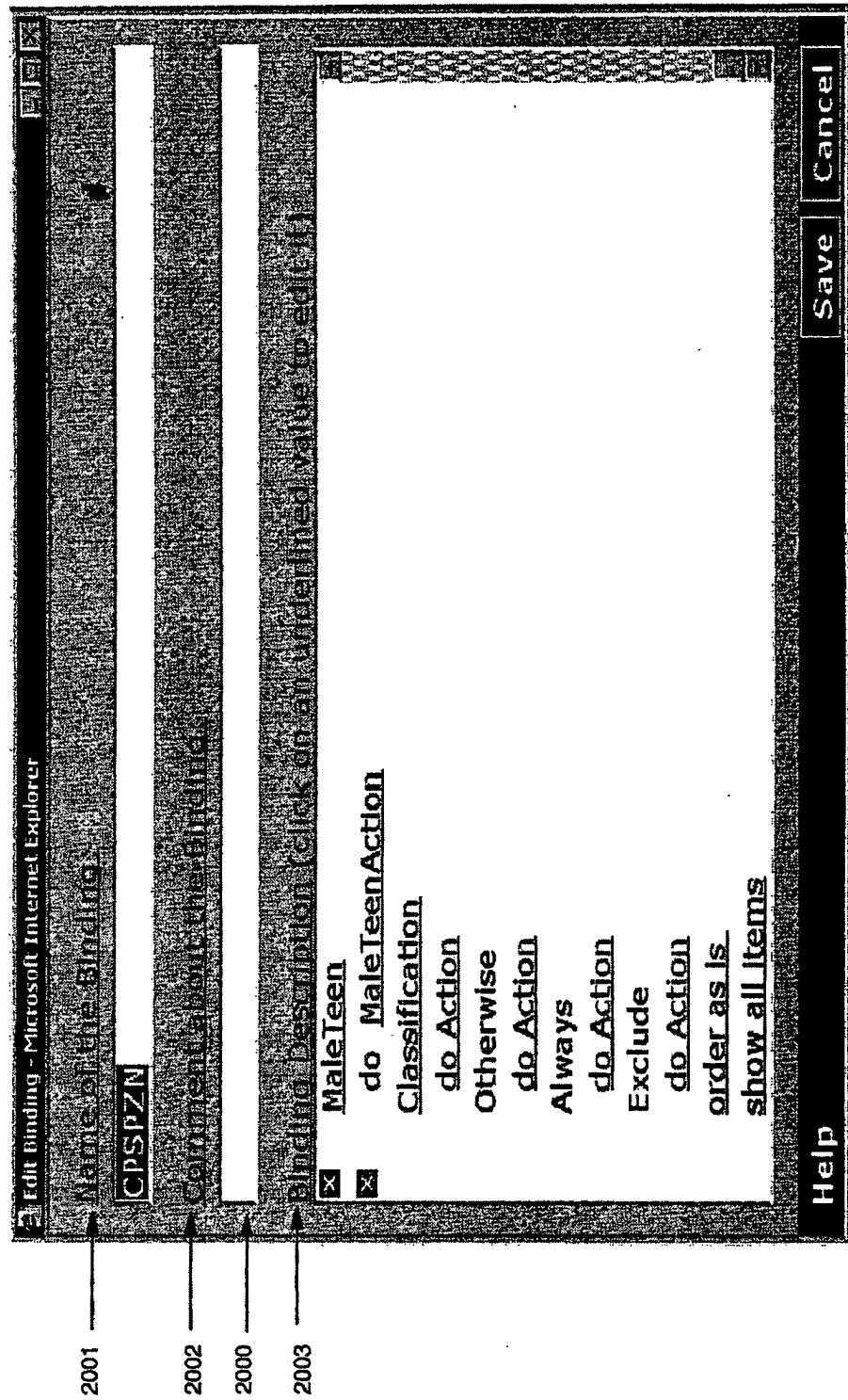


FIG. 18

