



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 697 23 432 T2** 2004.04.15

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 898 754 B1**

(51) Int Cl.⁷: **G06F 17/30**

(21) Deutsches Aktenzeichen: **697 23 432.0**

(86) PCT-Aktenzeichen: **PCT/GB97/01363**

(96) Europäisches Aktenzeichen: **97 923 195.8**

(87) PCT-Veröffentlichungs-Nr.: **WO 97/044747**

(86) PCT-Anmeldetag: **20.05.1997**

(87) Veröffentlichungstag
der PCT-Anmeldung: **27.11.1997**

(97) Erstveröffentlichung durch das EPA: **03.03.1999**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **09.07.2003**

(47) Veröffentlichungstag im Patentblatt: **15.04.2004**

(30) Unionspriorität:

9610505	20.05.1996	GB
96303645	22.05.1996	EP

(84) Benannte Vertragsstaaten:

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

(73) Patentinhaber:

**British Telecommunications Public Ltd. Co.,
London, GB**

(72) Erfinder:

**WEEKS, Richard, Felixstowe, Suffolk IP11 8UG,
GB; STEPHENS, Lee Michael, Ipswich, Suffolk IP4
5QT, US; DAVIES, Nicholas John, Colchester,
Essex CO4 4SX, GB; REVETT, Mike Charles,
Woodbridge, Suffolk IP12 4DL, GB; FLAVIN, Phil
Graeme, Felixstowe, Suffolk IP11 9TB, GB**

(74) Vertreter:

Beetz & Partner, 80538 München

(54) Bezeichnung: **INFORMATIONSAUFFINDUNGSSYSTEM MIT EINER CACHEDATENBANK**

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

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

Beschreibung

[0001] Die vorliegende Erfindung betrifft Verfahren und/oder Systeme zum Zugriff auf Informationen mit Hilfe eines Kommunikationssystems.

[0002] Das Internet ist ein bekanntes Kommunikationssystem auf der Grundlage von mehreren getrennten, miteinander verbundenen Kommunikationsnetzen. Es bietet eine reiche Quelle an Informationen von sehr verschiedenen Anbietern; jedoch bringt diese große Reichhaltigkeit auch das Problem des Zugriffs auf spezielle Informationen mit sich, da es keine zentrale Überwachung und Steuerung gibt.

[0003] 1982 hat sich das Volumen wissenschaftlicher, unternehmerischer, und technischer Informationen noch alle fünf Jahre verdoppelt. 1988 verdoppelte es sich alle 2,2 Jahre und 1992 alle 1,6 Jahre. Mit der Expansion des Internet und anderer Netzwerke nimmt auch die Anstiegsgeschwindigkeit weiter zu. Ein Schlüssel für die Realisierbarkeit solcher Netze ist die Fähigkeit, die Informationen zu verwalten und die Benutzer dann mit Informationen zu versehen, wenn sie diese benötigen.

[0004] Die Navigation der über das Internet verfügbaren Informationen wurde durch die Verwendung von Browsern und Sprachen, z. B. Hypertext Markup Language (HTML), möglich. So ist beispielsweise das bekannte Worldwide Web (WWW) ein Internet-Bereich, der "Browsern" mit Hilfe von Hypertext-Verbindungen zwischen den Dokumenten ermöglicht.

[0005] In der ebenfalls anhängigen internationalen Patentanmeldung Nr. GB 96/00 132, veröffentlicht am 1. August 1996 als WO 96/23265, ist ein System zum Zugriff auf Informationen, z. B. mit Hilfe des Internet, offenbart, das auf einer Gemeinschaft intelligenter Software-Agenten beruht, die Meta-Informationen über Seiten im Internet speichern. Das auf Agenten beruhende Zugriffssystem verwendet Sätze von Schlüsselworten, um die für einen bestimmten Benutzer interessanten Informationen zu lokalisieren. Darüber hinaus speichert es Benutzerprofile, so daß ein anderer Benutzer, dessen Profil zeigt, daß potentiell dafür Interesse besteht, über Seiten informiert wird, die von einem Benutzer gespeichert wurden.

[0006] Gemäß einer ersten Ausführungsform der vorliegenden Erfindung ist ein Informationszugriffssystem zum Zugriff auf Informationen angegeben, auf die mit Hilfe eines Kommunikationsnetzes zugegriffen werden kann, das umfaßt:

- a) einen Eingang zum Empfang von Anforderungen zum Zugriff auf Informationsobjekte,
- b) eine Einrichtung zum Herunterladen von angeforderten Informationsobjekten aus dem Netzwerk in einen lokalen Speicher und
- c) einen Informations-Prozessor zum Verarbeiten der heruntergeladenen Informationsobjekte, um eingebettete Verbindungen zu anderen Informationsobjekten, auf die über das Netzwerk zugegrif-

fen werden kann, zu orten, wobei der Informations-Prozessor umfaßt:

- i) eine Warteschlangen-Anordnung zur Anordnung der heruntergeladenen Informationsobjekte mit den eingebetteten Verbindungen in einer Warteschlange und
- ii) einen Satz gleichzeitig aktivierbarer Informations-Datenerfassungseinheiten zum Herunterladen von Informationsobjekten aus dem Netzwerk in einen lokalen Speicher, die durch die eingebetteten Verbindungen gekennzeichnet sind, wobei der Informations-Prozessor im Betrieb ein sich in der Warteschlange befindendes Objekt der heruntergeladenen angeforderten Information verarbeitet, um darin eingebettete Verbindungen zu orten, die andere Informationsobjekte zum Herunterladen kennzeichnen, und die im verarbeiteten Informationsobjekt georteten eingebetteten Verbindungen verschiedenen verfügbaren Datenerfassungseinheiten zuordnet, wobei dann jede Datenerfassungseinheit tätig wird und ein Informationsobjekt aus dem Netzwerk in den lokalen Speicher herunterlädt, das durch die ihr zugeordnete Verbindung gekennzeichnet ist.

[0007] Ein Vorteil der Ausführungsformen der vorliegenden Erfindung besteht darin, daß sich Verzögerungen beim Zugriff auf Informationsseiten reduzieren lassen. Es ist ebenfalls möglich, den Netzverkehr zu reduzieren, da Seiten nur einmal über das Netzwerk aufgerufen werden. Der nachfolgende Zugriff durch einen Benutzer kann auf den lokalen Speicher erfolgen.

[0008] Bevorzugt umfaßt der lokale Speicher mehr als einen Cache-Datenspeicher, nämlich, einen ersten Cache-Datenspeicher, der Informationsobjekte enthält, die durch das System abgefragt wurden, und einen zweiten Cache-Datenspeicher, der Informationsobjekte enthält, die aus dem ersten Cache-Datenspeicher übertragen wurden, wenn ein Benutzer einen Zugriff angefordert hat. Der zweite Cache-Datenspeicher kann dann in einer Weise verwaltet werden, die vom ersten Cache-Datenspeicher verschieden ist, um beispielsweise Informationen zu liefern, auf die von den Benutzern relativ häufig zugegriffen wird.

[0009] Der erste Cache-Datenspeicher kann beispielsweise wesentlich kleiner sein und kann so "beschnitten" sein, daß er nur Informationen enthält, die für Verbindungen heruntergeladen wurden, die in Seiten eingebettet sind, die erst kürzlich von Benutzern angefordert worden sind.

[0010] Im allgemeinen benötigen die Ausführungsformen der vorliegenden Erfindung Überwachungs- und Planungsmöglichkeiten, um die Warteschlangen-Verarbeitung zu steuern. Andernfalls könnte die Warteschlange einen Engpaß für die Verarbeitung bedeuten.

[0011] Gemäß einem zweiten Aspekt der vorliegenden Erfindung ist ein Verfahren zum Speichern von

Informationsobjekten angegeben, auf die mit Hilfe eines Kommunikationsnetzes zugegriffen werden kann, das die folgenden Schritte umfaßt:

- i) Empfang einer Anforderung für einen Zugriff auf ein Informationsobjekt,
- ii) Herunterladen des angeforderten Informationsobjekts aus dem Kommunikationsnetz und Speichern des heruntergeladenen angeforderten Objekts in einem lokalen Datenspeicher,
- iii) Verarbeiten des heruntergeladenen angeforderten Objekts, um irgendwelche darin eingebetteten Verbindungen zu orten, die sich auf andere Informationsobjekte beziehen, und
- iv) Herunterladen von Informationsobjekten, auf die durch eingebettete, im Schritt (iii) geortete Verbindungen verwiesen wurde, und Speichern der heruntergeladenen Objekte im lokalen Datenspeicher.

[0012] Ein Informationszugriffssystem gemäß einer Ausführungsform der vorliegenden Erfindung ist im folgenden lediglich beispielhaft unter Bezug auf die beigefügten Zeichnungen beschrieben, in denen zeigen:

[0013] **Fig. 1.** eine Umgebung, von der das Informationszugriffssystem unterstützt wird,

[0014] **Fig. 2** ein Flußdiagramm des gesamten Betriebs des Informationszugriffsystems,

[0015] **Fig. 3** ein Blockdiagramm der Bauteile eines Vorgriffsprozessors im Informationszugriffssystem von **Fig. 1**,

[0016] **Fig. 4** ein Flußdiagramm der Verarbeitung durch den Vorgriffsprozessor des Informationszugriffsystems,

[0017] **Fig. 5** eine Verarbeitungs-Warteschlange und die relevanten Informationserfassungseinheiten,

[0018] **Fig. 6** die Warteschlange und Einheiten von **Fig. 5** nach Ablauf eines Zeitintervalls und

[0019] **Fig. 7** ein zu **Fig. 2** alternatives Flußdiagramm.

[0020] Die spezielle, im folgenden beschriebene Ausführungsform der vorliegenden Erfindung wird hier als "Casper", für: Cached Access to Stored Pages with Easy Retrieval" (Cache-Zugriff auf gespeicherte Seiten mit einfachem Wiederabruf), bezeichnet. Das Casper-System ist insbesondere für einen Zugriff auf WorldWide-Web-Seiten über das Internet Global Communications Network ausgebildet. Andere Ausführungsformen der Erfindung können natürlich ebenfalls verwendet werden, um auf andere Informationssysteme zuzugreifen, in denen Dateneinheiten ("Informationsobjekte") darin eingebettete Verbindungen zu anderen Dateneinheiten aufweisen.

[0021] Die Informationen, auf die mit Hilfe des Web zugegriffen werden kann, sind als Seiten in HTML angegeben. Innerhalb eines Dokumentes können Wortfolgen oder andere Kennzeichnungen hervorgehoben sein. Wenn der Benutzer, während er ein Dokument anschaut, eine hervorgehobene Kette von Worten oder eine Kennung auswählt und darauf unter

Verwendung der Maustaste klickt, liefert der hervorgehobene Text eine Verbindung mit einem anderen Dokument. Ein Klick auf den hervorgehobenen Text triggert das System für einen Aufruf des relevanten Dokuments über das Internet, damit es der Benutzer auf dem Bildschirm ansehen kann. Er ersetzt das Dokument, das der Benutzer vorher angesehen hat.

[0022] In der obengenannten, ebenfalls anhängigen Patentanmeldung ist ein System beschrieben, das zum Speichern von Metainformationen über Seiten verwendet werden kann, die durch Anklicken von Hypertext-Verbindungen über das Internet ausgewählt wurden. Durch die Verwendung von Schlüsselwort-Sätzen macht das System andere interessierte Benutzer einer Benutzergruppe auf ein neues Dokument aufmerksam, für das Metainformationen gespeichert worden sind.

[0023] In einem Casper-System gemäß einer Ausführungsform der vorliegenden Erfindung wird das System, wenn eine Seite ausgewählt wurde, um beispielsweise deren Metainformationen zu speichern, automatisch diese Seite nach Hypertext-Verbindungen zu anderen Dokumenten überprüfen, die über das Internet verfügbar sind. Gibt es solche Verbindungen im Dokument, und die verbundenen Dokumente sind nicht bereits lokal gespeichert, dann schiebt das Casper-System die neue Seite in eine Seiten-Verarbeitungsschlange. Die Verbindungen der Seite werden, wenn sie verfügbar werden, den Seitenerfassungs-Softwareeinheiten zugeordnet. Für jede Verbindung wird das eine Verbindung aufweisende Dokument dann in einen lokalen "Vorgriffs"-Cache-Datenspeicher gelesen.

[0024] Das hat den Effekt, das ein lokaler Speicher von Seiten entsteht, die klar auf Seiten bezogen sind, an denen der Benutzer interessiert ist. Wenn der Benutzer sich entschließt, über eine Seite, auf die ursprünglich zugegriffen wurde, hinauszugehen, kann er die entsprechenden Seiten einfach aus ihren lokalen Daten-Cache-Speichern herausholen, ohne Verbindungen im Internet aufrufen zu müssen. Dadurch wird die Informationsabfrage für den Benutzer schneller und der Verkehr im Internet reduziert sich.

[0025] In **Fig. 1** umfaßt die Hardware-/Software-Umgebung, die die Ausführungsformen der vorliegenden Erfindung unterstützt, folgende Elemente:

- i) einen WWW-Viewer **100**, das über einen Proxy-Server mit einem Netzwerk, beispielsweise dem Internet, verbunden ist.
- ii) einen Haupt- und einen Vorgriffs-Cache-Speicher **120**, **125** und einen damit verbundenen Cache-Controller **115**.
- iii) einen Vorgriffsprozessor **130**, der zwischen den Cache-Controller **115** und den Vorgriffs-Cache-Speicher **125** geschaltet ist und der einen direkten Zugriff zum Internet **110** hat.

[0026] Die Casper unterstützende Umgebung ist einem allgemein bekannten Typ zuzuordnen. Der WWW-Viewer **100** bietet über den Proxy-Server **105**

die Browse-Möglichkeit im Internet **110** an. Beim Proxy-Server **105** handelt es sich um ein Software-System eines bekannten Typs, das Anfragen nach Informationen von den Browsern des Benutzers auffangen und diese verarbeiten kann, bevor es sie an die Informationsquelle im Internet weiterleitet. Bei dem Haupt- und dem Vorgriffs-Cache-Speicher **120**, **125** handelt es sich um lokale Datenspeicher zum Speichern von WWW-Seiten, und der Cache-Controller **115** bildet zu ihnen eine Schnittstelle, die Seiten in die Cache-Speicher schreibt und diese wieder abrufen und die Inhalte protokolliert.

[0027] Die Bereiche, in denen Casper primär unterschiedlich ist, betreffen das Vorsehen des zweiten Cache-Speichers, des Vorgriffs-Speichers **125** und des Vorgriffsprozessors **130**.

[0028] Es ist bekannt, Cache-Server zu verwenden, wobei es sich bei Cache um einen temporären Speicher handelt, in dem beispielsweise Seiten gespeichert gehalten werden, auf die erst kürzlich von einer Gruppe von Benutzern zugegriffen wurde. Wenn der Cache-Speicher voll ist, werden Seiten gelöscht, wenn neue Seiten gespeichert werden. Bei Casper ist es der Haupt-Cache-Speicher **120**, der die Seiten speichert, auf die von Benutzern zugegriffen wurde. Ferner gibt es jedoch den Vorgriffs-Speicher **125**, der Seiten speichert, auf die selbst nicht zugegriffen wurde, die aber mit Seiten verbunden sind, auf die Benutzer zugegriffen haben. Dieser Vorgriffs-Speicher **125** wird gefüllt, indem die Verbindungen genommen werden, die in diesen, von Benutzern aufgerufenen Seiten eingebettet sind und indem diese Seiten wegen der Verbindungen heruntergeladen werden.

[0029] Diese Cache-Speicher **120**, **125** werden dann verwendet, wenn ein Benutzer eine Seite anfordert. Casper verwendet den Proxy-Server **105**, um die Anfrage aufzufangen und um zunächst zu überprüfen, ob sich die Seite bereits im Haupt-Cache-Speicher **120** oder im Vorgriffs-Cache-Speicher **125** befindet. Ist die Seite bereits in einem der Cache-Speicher **120**, **125** vorhanden, holt Casper sie aus dem Cache-Speicher. Wenn nicht, muß schließlich die relevante Informationsquelle im Internet kontaktiert und die Seite von einem entfernt stehenden Server **135** angefordert werden.

[0030] In **Fig. 2** weist das Flußdiagramm, das sich auf den oben beschriebenen Ablauf bezieht, die folgenden Schritte auf:

[0031] Schritt 300: Der Proxy-Server **105** des Casper-Systems erhält eine Benutzeranfrage **300**, bei der an einem entfernt stehenden Standort im Internet, beispielsweise einem entfernt stehenden Server **135**, ein Universal-Ressourcenlokalisator (URL) eingesetzt ist. Bei der Anfrage kann es sich um eine Anfrage nach einer von mehreren Alternativen handeln. So kann beispielsweise der Benutzer die Ansicht einer Seite anfordern und/oder er kann für Aktualisierungszwecke das Wiederladen einer Seite vom entfernt stehenden Server **135** anfordern. Alternativ dazu kann der Benutzer ein Programm aufrufen, das

dynamisch läuft und für das eine Ablage im Cache-Speicher nicht zweckmäßig wäre. Wenn eine Anfrage nach einer Seite erfolgt, kann das bedeuten, daß die im Cache-Speicher abgelegte Version der Seite nicht akzeptabel ist und daß das System daher die Seite von dem entfernt stehenden Ursprungs-Server **135** aus liefern soll.

[0032] Schritt 305: Zur Bestimmung des Charakters der Anfrage und irgendwelcher Einschränkungen überprüft der Proxy-Server **105** die Benutzeranfrage. Die Benutzeranfrage kann optionale, vom Benutzer gewählte Einschränkungen enthalten, beispielsweise "Neustart", was bedeutet, daß die im Cache-Speicher abgelegte Version einer Datei nicht akzeptabel ist, sie kann aber auch darin eingebettete Einschränkungen enthalten, z. B. in Bezug auf den Universal-Ressourcenlokalisator, den sie enthält. Casper ist mit einer Konfigurations-Datei ausgestattet, die folgende Angaben machen kann:

- Universal-Ressourcenlokalisatoren (URL) sind auszuschließen, da sie beispielsweise einen zweifelhaften Inhalt aufweisen
- URL, die selbst einen Neustart erzwingen, weil beispielsweise bekannt ist, daß sich ihr Inhalt häufig ändert, oder weil der verwendete Server ein lokaler Server oder ein schneller Server ist und weil eine Ablage im Cache-Speicher daher nicht zweckmäßig ist.

[0033] Die Anfragenüberprüfung kann daher beispielsweise eine Reihe der im folgenden aufgelisteten Überprüfungen sein:

- erzwingt URL einen Neustart?
- ist URL eine zugelassene Stelle?
- hat der Benutzer einen Neustart angegeben?
- hat der Benutzer auf andere Weise angegeben, daß eine im Cache-Speicher vorhandene Version nicht akzeptabel ist?

[0034] In Abhängigkeit vom Ergebnis geht der Proxy-Server entweder zu Schritt 310 oder zu Schritt 330. Hat der Benutzer "Neustart" angeklickt, oder wenn URL den "Neustart" erzwingt, wird der Weg "Cache-Speicher-Version nicht akzeptabel" zu Schritt 310 verfolgt. Wenn es sich beim URL nicht um eine erlaubte Stelle handelt, kommt eine Meldung "Zugriff abgelehnt" zum Benutzer zurück und es erfolgt keine weitere Bearbeitung. Ansonsten wird der Weg "Cache-Speicher-Version akzeptabel" zu Schritt 330 verfolgt.

[0035] Schritt 310: Wenn im Schritt 305 festgestellt wurde, daß eine im Cache-Speicher aufgenommene Version nicht akzeptabel ist, initiiert der Proxy-Server **105** eine Verbindung mit dem relevanten entfernt stehenden Server **135**. Wie aus **Fig. 1** ersichtlich ist, hat der Proxy-Server **105** eine direkte Schnittstelle zum Internet **110**. Im Schritt 315 prüft der Proxy-Server **105**, ob das Objekt der Anfrage existiert. Ist das nicht der Fall, erzeugt er in den Schritten 320, 355 eine entsprechende Nachricht für den Benutzer und liefert

dieselbe.

[0036] Schritt 323: Wenn das Objekt der Anfrage existiert, prüft der Proxy-Server **105**, ob sie im Cache-Speicher abgelegt werden soll. Wenn das Objekt der Anfrage existiert und eine HTML-Seite ist, aber eine Ablage im Cache-Speicher nicht zweckdienlich ist, z. B. wenn eine Casper-Konfigurations-Datei den URL als eine Stelle definiert, die den Neustart erzwingt, geht der Poxy-Server **105** zu den Schritten 350, 355 weiter, um den Vorgriffsprozessor auszulösen und die Seite zum Browser des Benutzers zu senden. Wenn das Objekt der Anfrage existiert, es sich um eine HTML-Seite handelt und eine Ablage im Cache-Speicher zweckdienlich ist, geht der Proxy-Server **105** zu Schritt 325 weiter und schreibt die entsprechende Seite in den Haupt-Cache-Speicher **120**. Sodann geht der Proxy-Server **105** zu den Schritten 350, 355, wie oben beschrieben.

[0037] Es kann der Fall eintreten, daß zur Zeit weder eine Ablage im Cache-Speicher noch eine Vorgriffs-Verarbeitung zweckdienlich sind, beispielsweise in dem Fall, in dem ein Programm dynamisch läuft. In diesem Fall kann der Proxy-Server **105**, wie oben beschrieben, zu den Schritten 350, 355 gehen, der Vorgriffsprozessor **130** findet aber keine eingebetteten Verbindungen, die er verarbeiten könnte. Andererseits kann auch eine zusätzliche Überprüfung erfolgen (in **Fig. 2** nicht gezeigt), ob eine Vorgriffs-Verarbeitung zweckdienlich ist. Ist das nicht der Fall, könnte der Proxy-Server **105** die Schritte 350, 355 auslassen und das Programm einfach an den Browser **100** des Benutzers liefern.

[0038] Schritt 330: Wenn im Schritt 305 bestimmt wurde, daß eine Aufnahme im Cache-Speicher zweckdienlich ist, der Benutzer keinen Neustart festgelegt hat, oder daß eine im Cache-Speicher aufgenommene Version nicht akzeptabel ist, überprüft der Proxy-Server **105** über den Cache-Controller **115**, ob sich das Objekt der Anfrage, normalerweise eine Web-Seite, im Haupt-Cache-Speicher **120** befindet. Wenn das der Fall ist, ist der Proxy-Server **105** bereit, zu Schritt 345 zu gehen.

[0039] Wenn sich eine angeforderte Web-Seite nicht im Haupt-Cache-Speicher **120** befindet, überprüft der Proxy-Server **105** über den Cache-Controller **115**, ob sie sich bereits im Vorgriffs-Cache-Speicher **125** befindet. Wenn das der Fall ist, geht der Proxy-Server **105** zu Schritt 340 und überträgt die Seite in den Haupt-Cache-Speicher **120** und kann dann zu Schritt 345 gehen.

[0040] Wenn die angeforderte Web-Seite auch nicht im Vorgriffs-Cache-Speicher **125** ist, geht der Proxy-Server **105** zu Schritt 310 und die Verarbeitung wird fortgesetzt, wie im Schritt 310 beschrieben.

[0041] Schritt 345: Der Proxy-Server holt die angeforderte Seite aus dem Haupt-Cache-Speicher **120** und geht zu Schritt 350, wobei er den Vorgriffsprozessor **130** triggert. Nach dem Triggern des Prozessors geht der Proxy-Server zu Schritt 355, wo er der(den) angeforderten Seite(n) Casper-Sei-

ten-Deckblätter hinzufügt und sie an den Browser **100** des Benutzers schickt.

[0042] Der Vorgriffsprozessor **130** überprüft in der Zwischenzeit, wie im einzelnen im folgenden beschrieben ist, die angeforderte(n) Seite(n), um darin eingebettete Verbindungen zu erkennen und auf die Seiten mit den eingebetteten Verbindungen zuzugreifen und sie in den Vorgriffs-Cache-Speicher **125** herunterzuladen.

[0043] Der in **Fig. 3** gezeigte Vorgriffsprozessor **130** umfaßt Software-Verarbeitungen und einen Speicher, die auf einer Unix-Maschine läuft. Er enthält drei Asynchronverarbeitungen, einen Seiten-Prozessor **255**, einen Schlitz-Prozessor **260** und einen Shelf-Prozessor **265**.

[0044] Der Seiten-Prozessor **255** umfaßt: einen Nachrichteneingang **200** zum Empfang von Nachrichten, die beispielsweise den URL einer zu verarbeitenden Seite enthalten, einen HTML-Parser **205** für den Abruf der Inhalte eines URL und zur gründlichen Syntax-Analyse derselben, um alle darin eingebetteten HTML-Verbindungen zu erkennen, einen Verbindungs-Assessor **210**, um zu bestimmen, welche der erkannten Verbindungen "vorab geholt" werden sollen, und eine Seiten-Warteschlange **215**, die die am Nachrichteneingang **200** eingegangenen Seitendaten zur Verarbeitung als Verarbeitungsanfragen speichert. Der Parser **205** und der Verbindungs-Assessor **210** haben auf den Haupt-Cache-Speicher **120** Zugriff und der Verbindungs-Assessor **210** hat auf den Vorgriffs-Chache-Speicher **125** Zugriff. Mit der Seiten-Warteschlange **215** ist ein Zeitgeber **235** gekoppelt, der irgendwelche Verbindungen mit "Ignorieren" kennzeichnet, die zu lange auf eine Verarbeitung gewartet haben (ein konfigurierbarer Parameter). Dadurch wird ein Mechanismus für eine Beschränkung der Länge der Seitenschlange geschaffen.

[0045] Der Schlitz-Prozessor **260** dient allgemein dem Zweck, die Seitendaten der Seitenschlange **215** so rasch wie möglich zu verarbeiten. Zu diesem Zweck kann er parallel eine Reihe von Unix-Unterverarbeitungen mit Hilfe mehrerer Datenerfassungsschlitze **225** durchlaufen. Er ist mit einem Schlitzfüller **220** versehen, der den Zustand der Schlitze **225** überwacht und versucht, irgendwelche freien Schlitze der Seitenschlange **215** zu bestücken. Jedesmal, wenn ein freier Schlitz neu gefüllt ist, wird ein Verbindungsprozessor **230** als Unix-Unterverarbeitet gestartet, der versucht, eine Verbindung mit dem Server herzustellen, der von einem URL in einer eingebetteten Verbindung genannt wurde. War das erfolgreich, liest ein Leseprozessor **240** die betreffende Seite von Server und leitet sie zu einem Cache-Writer **245**, einem Teil des Cache-Controllers **115**, der die Daten in den Vorgriffs-Cache-Speicher **125** schreibt und den Shelf-Prozessor **265** benachrichtigt. Beide, der Verbindungs- und der Leseprozessor **230**, **240** sind mit einem Zeitgeber **235** ausgestattet, damit verhindert wird, daß die Datenerfassungsschlitze **225** in ineffizi-

enter Weise blockiert werden.

[0046] Der Shelf-Prozessor **265** schränkt die Größe des Vorgriffs-Cache-Speichers **125** ein, indem er Dateien nach Ablauf einer bestimmten Zeit entfernt. Eine Liste von Dateien bleibt gespeichert, die zum Vorgriffs-Cache-Speicher **125** gesendet wurden, indem Datei-Kennungen in Zeitschlitze in einem Datenspeicher **250** eingegeben werden, der mit "Timeout Shelf" (Zeitablauf-Shelf) bezeichnet ist. Dieser ist mit einem Zeitgeber **235** versehen, der die Zeitschlitze in effektiver Weise durch den Datenspeicher schiebt, wobei jeder Zeitschlitz eine Löschung aller mit ihm gekoppelter Dateien aus dem Vorgriffs-Cache-Speicher **125** auslöst, wenn er das Ende des Datenspeichers erreicht hat.

[0047] Die maximal zulässige Länge der Seitenschlange und die Anzahl der Datenerfassungsschlitze sollen konfigurierbar sein, damit die Betriebsbedingungen erfüllt werden. Zur Dimensionierung eines Vorgriffsprozessors **130** sind die unten beschriebenen Überlegungen zweckdienlich.

[0048] Bezüglich einer Seitenschlangenlänge besteht bei den Softwarebedingungen kein ernstlicher Mehraufwand darin, die Seitenschlange beliebig lang zu machen. Die Beschneidung ihrer Länge ist jedoch eine Maßnahme zur Steuerung der Leistung des Systems. Der Grund dafür, Seiten am Ende der Schlange fallen zu lassen, ist der, daß der Benutzer an Seiten, die sich am Ende der Schlange befinden, ohne daß sie vollständig verarbeitet wurden, wahrscheinlich zu diesem Zeitpunkt ohnehin das Interesse verloren hat. Es besteht daher keine Notwendigkeit mehr, solche Seiten weiter zu verarbeiten.

[0049] Es wäre daher eine mögliche Ausgangsstrategie, mit einer kurzen Seiten-Warteschlange zu beginnen, und ihre Länge solange zu steigern, bis sie, in seltenen Fällen, überläuft. Wenn jedoch Seiten zu lange, beispielsweise mehr als eine Minute, in der Warteschlange verbleiben, kann das ein Zeichen dafür sein, daß die Länge der Schlange verkürzt werden sollte. Alternativ dazu sollte die Anzahl der Datenerfassungsschlitze erhöht werden; siehe unten.

[0050] Die Sache ist wegen der Tatsache etwas kompliziert, daß es in Abhängigkeit von der Länge des "Shelf", d. h. der zulässigen Lebenszeit der Seiten im Vorgriffs-Cache-Speicher, zwei unterschiedliche Betriebsweisen des Vorgriffs-Cache-Speichers geben kann.

[0051] Wenn der einzige Zweck zur Aufnahme in den Vorgriffs-Cache-Speicher darin besteht, den den Benutzern während ihrer unmittelbaren Browsing-Sitzungen eine gesteigerte Ansprechgeschwindigkeit zu bieten, dann gibt es keinen Grund, Seiten mehr als einige wenige Minuten im Cache-Speicher zu halten. Wenn Seiten bedeutend länger, etwa einige Tage, im Vorgriffs-Cache-Speicher gehalten werden, besteht die Möglichkeit, daß der Cache-Speicher auch die zweite Anforderung einer Seite erfüllen kann. Wenn der Vorgriffs-Cache-Speicher auf diese Weise betrieben würde, würde das das obige Argument für eine

Festlegung der Seitenschlangenlänge zugunsten einer längeren Schlange ändern, da ein Vorteil darin besteht, Seiten im vorhinein abzurufen, auch wenn das mehrere Minuten in Anspruch nimmt.

[0052] Hinsichtlich der Anzahl der Datenerfassungsschlitze auf dem einfachsten Niveau kann gesagt werden: je mehr Schlitze verfügbar sind, desto besser ist es. Wenn nämlich ständig freie Datenerfassungsschlitze vorhanden sind, können die Seiten mit einem Minimum an Verzögerung aufgerufen werden.

[0053] In der Praxis zieht eine Unixmaschine die Grenzen für die Anzahl der zulässigen Unterprozesse. Innerhalb dieser Grenzen kann eine große Anzahl von Unterprozessen, beispielsweise mehr als zwanzig, laufen, da keiner von ihnen viel Raum im Zentralprozessor oder in der Speicherquelle benötigt. Die meisten der Unterprozesse "leben", d. h. sie warten einfach darauf, daß Daten verfügbar werden.

[0054] Wenn die Last am Vorgriffsspeicher höher ist, als eine Maschine handhaben kann, können natürlich mehrere Maschinen eingesetzt werden.

[0055] Eine Alternative zu den Unterprozessen wäre, mehrere Teilprozesse im Vorgriffsprozessor laufen zu lassen. Dieses Verfahren würde weniger zusätzliche Betriebssysteme, aber möglicherweise einen noch effizienteren Betrieb bedeuten, die Wartung könnte sich aber schwieriger gestalten. Der Vorgriffsprozessor würde dann nur in einem Programm und nicht in getrennten Teilen implementiert, die gewartet werden müssen. Bei der Zuverlässigkeit des Vorgriffsprozessors könnte man auch einen Kompromiß eingehen: Die Anordnung für eine Mehrfach-Unterverarbeitung ist tolerant in Bezug auf einen Ausfall beliebiger Unterverarbeitungen; solche Ausfälle beeinträchtigen den Vorgriffsprozessor selbst nicht, der aber dennoch zuverlässig und kontinuierlich laufen muß.

[0056] Gemäß **Fig. 4** kann ein Flußdiagramm des grundlegenden Betriebs des Vorgriffsprozessors wie folgt beschrieben werden:

[0057] Schritt 400: Der Proxy-Server **105** liefert dem Benutzer entweder aus seinem Haupt-Cache-Speicher oder aus dem Internet eine Seite und instruiert den Vorgriffsprozessor **130**, die Seite zu bearbeiten, indem er dem Prozessor, der den URL für die betreffende Seite enthält, eine Nachricht übermittelt.

[0058] Schritt **405**: Der Vorgriffsprozessor **130** erhält die Inhalte des URL aus dem Haupt-Cache-Speicher **120** und unterzieht den Inhalt der Seite einer Syntaxanalyse, um die Verbindungen zu anderen Seiten festzustellen. (Das heißt, er übersetzt die HTML der Seite, um Informationen über miteinander verbundene Seiten zu gewinnen. Dafür ist natürlich das einprogrammierte Wissen der HTML-Syntax vonnöten.) Schritt 410: Die Existenz der so ermittelten Kind-Verbindungen wird in Bezug auf den Haupt-Cache-Speicher **120** und den Vorgriffs-Cache-Speicher **125** untersucht.

[0059] Schritt 415: Wenn eine Kind-Verbindung nicht bereits im Cache-Speicher abgelegt ist, und es sich nicht um eine Seite oder eine Kind-Seite handelt,

die bereits verarbeitet wird, wird sie der Kind-Liste der Seite hinzugefügt.

[0060] Schritt 420: Wenn festgestellt wird, daß die zu verarbeitende Seite irgendwelche Kind-Verbindungen hat, die nicht aufgerufen werden müssen, wird die Seite in den Anfang der Seitenverarbeitungsschlange aufgenommen. Jeder Eintrag in der Seitenverarbeitungsschlange **215** ist mit der Seiten-URL und, für jedes der Kind-Verbindungen, mit einem Verbindungs-URL und einem Verbindungszustand gekoppelt. Letzterer wird aus: "ignoriert", wenn eine Ablehnung durch den Verbindungs-Assessor **210** erfolgt ist, "erledigt", wenn eine vollständige Verarbeitung erfolgt ist, oder "abgebrochen" ausgewählt, wenn die Zeit abgelaufen ist, oder aus einer Schlitzzahl ausgewählt, wenn gerade eine Verarbeitung läuft.

[0061] Schritt 425: Wenn die Seitenverarbeitungsschlange **215** zu lang geworden ist, was ein Anzeichen für einen Engpaß ist, wird die letzte Seite aus der Schlange entfernt und irgendwelche ausstehenden Datenerfassungen, die mit den Kindern der Seite gekoppelt sind, werden abgebrochen.

[0062] Schritt 430: Es wird ein Versuch unternommen, jedes Kind der neuen Seite zu verarbeiten, und irgendwelche noch nicht verarbeiteten Kinder aus anderen Seiten in der Seitenschlange, indem sie inaktiven Datenerfassungsschlitzten der Seiten zugeordnet werden. Der Schlitzfüller **220** überwacht den Zustand der Datenerfassungsschlitzte **225**; jeder von ihnen kann den Zustand "frei", "verbunden" oder "lesend" aufweisen, und bestückt die freien Schlitzte aus der Seitenschlange **215**, beginnend mit der ersten, nicht verarbeiteten Verbindung in der neu hinzugefügten Seite.

[0063] Schritt 435: Die neu gefüllten Datenerfassungsschlitzte der Seite werden aktiviert, und ein "Verbindungsprozessor" **230** wird als eine Unix-Unterverarbeitung, wie oben beschrieben, gestartet.

[0064] Schritt 440: Irgendwelche Datenerfassungsschlitzte einer Seite, die für den Verbindungsversuch zu lange gebraucht haben, oder die aus irgendwelchen anderen Gründen ausgefallen sind, werden freigegeben und als inaktiv bezeichnet, worüber dem Schlitzfüller **220** Mitteilung gemacht wird.

[0065] Schritt 445: Irgendwelche Datenerfassungsschlitzte für Seiten, die erfolgreich mit den Servern **135** verbunden wurden, werden vom "Verbindungs"-Zustand in den "Lesen"-Zustand aktualisiert. Es wird von einem Leseprozessor **240** gelesen, der die gleiche Unix-Unterverarbeitung wie die Verbindungsphase verwendet. Der Schlitzprozessor **130** liest die relevanten Kind-Verbindungsdaten aus dem Netzwerk und übermittelt sie einem Cache-Writer im Cache-Controller **115**. Dieser schreibt die Daten in den Vorgriffs-Cache-Speicher **125** und macht dem Shelf-Prozessor **265** Mitteilung.

[0066] Schritt 450: Irgendwelche Seiten-Datenerfassungsschlitzte **225**, die für ihren Versuch, aus den Servern zu lesen, zu lange gebraucht haben, werden freigegeben und als inaktiv markiert, worüber wieder-

um, wie in der Verbindungsphase, dem Schlitzfüller **220** Mitteilung gemacht wird.

[0067] Schritt 455: Seiten, die zu lange im Vorgriffs-Speicher geblieben sind, werden als unerwünscht angesehen und aus dem Cache-Speicher entfernt. Das ist praktisch, denn irgendwelche Dateien, auf die von einem Benutzer nach dem Vorabruf zugegriffen wurde, sind von einer getrennten Cache-Controller-Anordnung zum Haupt-Cache-Speicher **120** übertragen worden. Obwohl es außerordentlich einfach ist, den Vorgriffs-Cache-Speicher **125** abzugleichen, indem er regelmäßig auf veraltete Dateien abgetastet wird und diese entfernt werden, wird doch durch diese Technik allzu viel Rechenzeit verbraucht und macht einen Zugriff mit einer hohen Zahl von Disketten erforderlich. Daher ist es bevorzugt, einen Shelf-Prozessor **265**, wie oben beschreiben, zu verwenden.

[0068] Die obigen Verarbeitungsschritte wiederholen sich auf unbegrenzte Zeit und nicht notwendigerweise in der beschriebenen Reihenfolge. Natürlich können insbesondere die Schritte 440, 445, 450 und 455 mit einer Häufigkeit ausgeführt werden, die in Relation zu dem Rest des Verfahrens als zweckmäßig erachtet wurde; diese Schritte sind durch gepunktete Linien dargestellt, um anzuzeigen, daß sie nicht immer dann vorhanden müssen, wenn der Vorgriffsprozessor **130** beauftragt wird, eine Seite zu bearbeiten.

[0069] Eine Realzeitanzeige des Systems könnte so aussehen, wie in **Fig. 5** gezeigt ist. In diesem Beispiel sind vier Seiten in der Schlange **215** und es gibt zehn einzelne Datenerfassungsschlitzte **505**. Hier wurde angenommen, daß eine neue Seite (Seite 1) gerade angekommen ist, und daß einige Datenerfassungsschlitzte **505** soeben frei geworden sind, da die Verarbeitung einiger Kinder von Seite 4 gerade abgeschlossen worden ist, dann sind einige Neuzuweisungen von Datenerfassungsschlitzten **505** erforderlich. Zwei Kinder von Seite 2 werden gerade verarbeitet. Die anderen zwei Kinder von Seite 2 sind zwar auch zur Verarbeitung bereit, jedoch könnte die Ankunft der neuen Seite 1 einen Einfluß darauf haben, was mit ihnen geschieht. Seite 3 hat vier Kind-Verarbeitungen, eine Verbindung wird aufgebaut, die anderen Verbindungen sind bereits hergestellt und gelesen. Seite 4 ist fertig, alle ihre Kinder sind entweder gelesen oder die Verarbeitung wurde abgebrochen, da zuviel Zeit benötigt wurde.

[0070] Der Vorgriffsprozessor **130** wird nun die Kinder der neu angekommenen Seite 1, den Schlitzten **2, 6** und **7**, und ein Kind der Seite 2 dem Schlitz **8** zuweisen. Seite 4 wird aus der Warteschlange entfernt, da sie keine ausstehenden Operationen aufweist.

[0071] Aus **Fig. 6** können die Ergebnisse der Neuzuordnung entnommen werden.

[0072] **Fig. 7** zeigt, daß auch eine etwas andere Version der vorliegenden Erfindung bevorzugt werden könnte: In den Schritten 345, 350, umgeht das System den Vorgriffsprozessor **130** in einem Fall, in

dem eine Seite bereits im Haupt-Cache-Speicher **120** gefunden worden ist. Dieser Version kann dann der Vorzug gegeben werden, wenn alle Seiten bereits vom Vorgriffsprozessor **130** verarbeitet worden sind, bevor sie in den Haupt-Cache-Speicher **120** geladen werden.

[0073] Eine Folge der obigen, in **Fig. 7** gezeigten Variation ist es, daß das System direkt den Vorgriffsprozessor **130** (Schritt 350) auslöst, nachdem eine Seite vom Vorgriffs-Cache-Speicher **125** in den Haupt-Cache-Speicher **120** übertragen wurde. In diesem Falle muß das System, wenn es die Seite in den Haupt-Cache-Speicher überträgt, die Seite auch zum Vorgriffsprozessor **130** übertragen.

[0074] Es ist möglich, die Ausführungsformen der Erfindung in einer objektbezogenen Technologie zu gestalten. So können beispielsweise die drei Prozessoren, der Seiten-, Schlitze- und Shelf-Prozessor, als Objekte auf dem höchsten Niveau angesehen werden. Jeder kann eine unabhängige Einheit sein, die mit den anderen mit Hilfe von Nachrichten anstelle eines direktem Datenzugriffs oder einer Datenmodifizierung kommuniziert. Auf einem niedrigeren Niveau haben die Elemente der Seitenschlange und der Datenerfassungsschlitze objektähnliche Merkmale: Jedes Element ist eine Umschreibung eines Prototyps (ein Seitenschlangen-Element oder ein Datenerfassungsschlitze) und jedes verfügt über zugehörige Daten und Zustände.

[0075] Wie beschrieben, greift Casper auf alle Seiten zu, die mit einer ausgewählten Seite verbunden sind. Es ist möglich, das Prinzip der ebenfalls anhängigen Patentanmeldung, auf die oben Bezug genommen wurde, zu nutzen und die Seiten auszuwählen, auf die in Übereinstimmung mit einem Benutzerprofil beispielsweise auf der Grundlage des Interesses und des Kontextes zugegriffen wurde.

Patentansprüche

1. Informationssystem zum Zugriff auf Informationen, auf die mit Hilfe eines Kommunikationsnetzes zugegriffen werden kann, das umfaßt:

- a) einen Eingang zum Empfang von Anforderungen zum Zugriff auf Informationsobjekte,
 - b) eine Einrichtung zum Herunterladen von angeforderten Informationsobjekten aus dem Netzwerk in einen lokalen Speicher und
 - c) einen Informations-Prozessor zum Verarbeiten der heruntergeladenen Informationsobjekte, um eingebettete Verbindungen zu anderen Informationsobjekten, auf die über das Netzwerk zugegriffen werden kann, zu orten,
- wobei der Informations-Prozessor umfaßt:

- i) eine Warteschlangen-Anordnung zur Anordnung der heruntergeladenen Informationsobjekte mit den eingebetteten Verbindungen in einer Warteschlange und
- ii) einen Satz gleichzeitig aktivierbarer Informations-Datenerfassungseinheiten zum Herunterladen

von Informationsobjekten aus dem Netzwerk in einen lokalen Speicher, die durch die eingebetteten Verbindungen gekennzeichnet sind,

wobei der Informations-Prozessor im Betrieb ein sich in der Warteschlange befindendes Objekt der heruntergeladenen angeforderten Information verarbeitet, um darin eingebettete Verbindungen zu orten, die andere Informationsobjekte zum Herunterladen kennzeichnen, und die im verarbeiteten Informationsobjekt georteten eingebetteten Verbindungen verschiedenen verfügbaren Datenerfassungseinheiten zuordnet, wobei dann jede Datenerfassungseinheit tätig wird und ein Informationsobjekt aus dem Netzwerk in den lokalen Speicher herunterlädt, das durch die ihr zugeordnete Verbindung gekennzeichnet ist.

2. Informationssystem nach Anspruch 1, wobei der lokale Speicher mindestens zwei Cache-Datenspeicher umfaßt, einen ersten Cache-Datenspeicher, in den die Datenerfassungseinheiten im Betrieb gekennzeichnete Informationsobjekte heruntergeladen, und einen zweiten Cache-Datenspeicher zum Speichern von Informationsobjekten, die auf eine am Eingang erhaltene Anforderung heruntergeladen wurden.

3. Informationssystem nach Anspruch 2, das ferner eine Einrichtung umfaßt, die auf den Empfang einer Anforderung zum Zugriff auf ein Informationsobjekt am Eingang anspricht, das im ersten Cache-Datenspeicher gespeichert ist, um das angeforderte Informationsobjekt vom ersten Cache-Datenspeicher zum zweiten Cache-Datenspeicher zu übertragen.

4. Informationssystem nach Anspruch 2 oder 3, wobei jede empfangene Anforderung einen Ortungsindikator enthält, der den Ort des angeforderten Informationsobjekts im Netzwerk anzeigt, und das System ferner ein Register zum Speichern der Ortungsindikatoren zusammen mit verwandten begrenzten Daten sowie eine Einrichtung umfaßt, die auf den Empfang einer Anforderung zum Zugriff auf ein Informationsobjekt am Eingang anspricht, um auf das Register zuzugreifen, um beliebige begrenzte Daten zu erhalten, die in bezug auf einen in der empfangenen Anforderung enthaltenen Ortungsindikator gespeichert sind.

5. Informationssystem nach einem der vorhergehenden Ansprüche, das ferner eine Warteschlangenlängen-Steuereinrichtung zum Streichen von Informationsobjekten aus der Warteschlangen-Anordnung umfaßt.

6. Informationssystem nach Anspruch 5, wobei die Warteschlangenlängen-Steuereinrichtung so ausgebildet ist, daß sie Informationsobjekte aufgrund der Länge der Zeit streicht, die sich ein Informationsobjekt in der Warteschlangen-Anordnung be-

findet.

7. Informationssystem nach einem der vorhergehenden Ansprüche, das ferner eine Datenerfassungseinheit-Überwachungseinrichtung umfaßt, die erfaßt, wann Datenerfassungseinheiten verfügbar werden, und um die Zuordnung von in Informationsobjekten vorhandenen Verbindungen zu verfügbaren Datenerfassungseinheiten zu triggern, die sich in der Warteschlangen-Anordnung befinden.

8. Informationssystem nach Anspruch 7, wobei die Datenerfassungseinheit-Überwachungseinrichtung die Zuordnung von Verbindungen in einer Prioritätenfolge triggert, wobei die Prioritätenfolge in Übereinstimmung mit dem Zeitraum bestimmt wird, den sich ein Informationsobjekt in der Warteschlangen-Anordnung befunden hat.

9. Informationssystem nach Anspruch 8, wobei das höchste Prioritätsniveau mit den Verbindungen verknüpft ist, die sich in dem Objekt mit der kürzesten Zeit in der Warteschlangen-Anordnung befinden.

10. Informationssystem nach einem der vorhergehenden Ansprüche, wobei die Warteschlangen-Anordnung so ausgebildet ist, daß sie jedes Informationsobjekt durch objektbezogenes Speichern – einer Kennung für das Objekt und – einer Kennung für eine oder mehrere im Objekt eingebettete Verbindungen in der Warteschlange ordnet.

11. Informationssystem nach Anspruch 10, wobei die Warteschlangen-Anordnung ferner so ausgebildet ist, daß eine Zustandsinformation in bezug auf jede der Verbindungskennungen, insbesondere den Herunterlade-Zustand für jede Verbindung, gespeichert wird, die anzeigt, ob das jeweilige gekennzeichnete Informationsobjekt durch eine Informations-Datenerfassungseinheit heruntergeladen wurde.

12. Informationssystem nach einem der vorhergehenden Ansprüche, das einen Proxy-Server umfaßt, der so ausgebildet ist, daß er eine Zugriffsanforderung eines Benutzers auf ein Informationsobjekt zum Herunterladen des angeforderten Informationsobjektes aus dem Netzwerk in den lokalen Speicher und zum Triggern der Funktion des Informations-Prozessors in bezug auf das heruntergeladene angeforderte Informationsobjekt empfängt.

13. Informationssystem nach Anspruch 12, wobei im Fall der Abhängigkeit von Anspruch 2 der Proxy-Server so ausgebildet ist, daß er das heruntergeladene angeforderte Informationsobjekt im zweiten Cache-Datenspeicher speichert.

14. Informationssystem nach Anspruch 13, wobei der Proxy-Server ferner so ausgebildet ist, daß er beim Empfang einer Zugriffsanforderung auf ein Informationsobjekt bevorzugt vor einem Herunterladen des angeforderten Objekts über das Kommunikationsnetz auf mindestens einen, den ersten oder den zweiten Cache-Datenspeicher zugreift, um das angeforderte Informationsobjekt wiederzugewinnen.

15. Verfahren zum Speichern von Informationsobjekten, auf die mit Hilfe eines Kommunikationsnetzes zugegriffen werden kann, das die folgenden Schritte umfaßt:

- i) Empfang einer Anforderung für einen Zugriff auf ein Informationsobjekt,
- ii) Herunterladen des angeforderten Informationsobjekts aus dem Kommunikationsnetz und Speichern des heruntergeladenen angeforderten Objekts in einem lokalen Datenspeicher,
- iii) Verarbeiten des heruntergeladenen angeforderten Objekts, um irgendwelche darin eingebetteten Verbindungen zu orten, die sich auf andere Informationsobjekte beziehen, und
- iv) Herunterladen von Informationsobjekten, auf die durch eingebettete, im Schritt (iii) geortete Verbindungen verwiesen wurde, und Speichern der heruntergeladenen Objekte im lokalen Datenspeicher.

16. Verfahren nach Anspruch 15, wobei Schritt (iii) umfaßt: Anordnen der heruntergeladenen angeforderten Informationsobjekte durch Speichern, bezogen auf jedes Objekt, einer Kennung für das Objekt und einer Kennung für jede geortete, im Objekt eingebettete Verbindung in einer Warteschlange.

17. Verfahren nach Anspruch 15 oder 16, wobei Schritt (iv) umfaßt: Zuordnen von eingebetteten Verbindungen, die im Schritt (iii) geortet wurden, zu verschiedenen Datenerfassungseinheiten eines Satzes von gleichzeitig aktivierbaren Datenerfassungseinheiten und Herunterladen der Informationsobjekte mit den zugeordneten Verbindungen mit Hilfe der Datenerfassungseinheiten.

18. Verfahren nach Anspruch 17, wobei im Fall der Abhängigkeit von Anspruch 16 der Schritt (iii) ferner umfaßt: Speichern von Zustandsinformationen in bezug auf jede Kennung der eingebetteten Verbindungen, die ihren Zustand in bezug auf das Herunterladen durch die Datenerfassungseinheiten anzeigen.

19. Verfahren nach Anspruch 17 oder 18, wobei der lokale Datenspeicher mindestens zwei Cache-Datenspeicher, einen ersten Cache-Datenspeicher, in den die Datenerfassungseinheiten im Betrieb gekennzeichnete Informationsobjekte herunterladen, und einen zweiten Cache-Datenspeicher zum Speichern von Informationsobjekten umfaßt, die aufgrund einer am Eingang erhaltenen Anforderung herunter-

geladen wurden, wobei das Verfahren ferner den Schritt umfaßt: (v) Übertragen des angeforderten Informationsobjekts vom ersten Cache-Datenspeicher zum zweiten Cache-Datenspeicher nach dem Empfang einer Anforderung zum Zugriff auf ein Informationsobjekt, das im ersten Cache-Datenspeicher gespeichert ist.

Es folgen 6 Blatt Zeichnungen

Fig.1

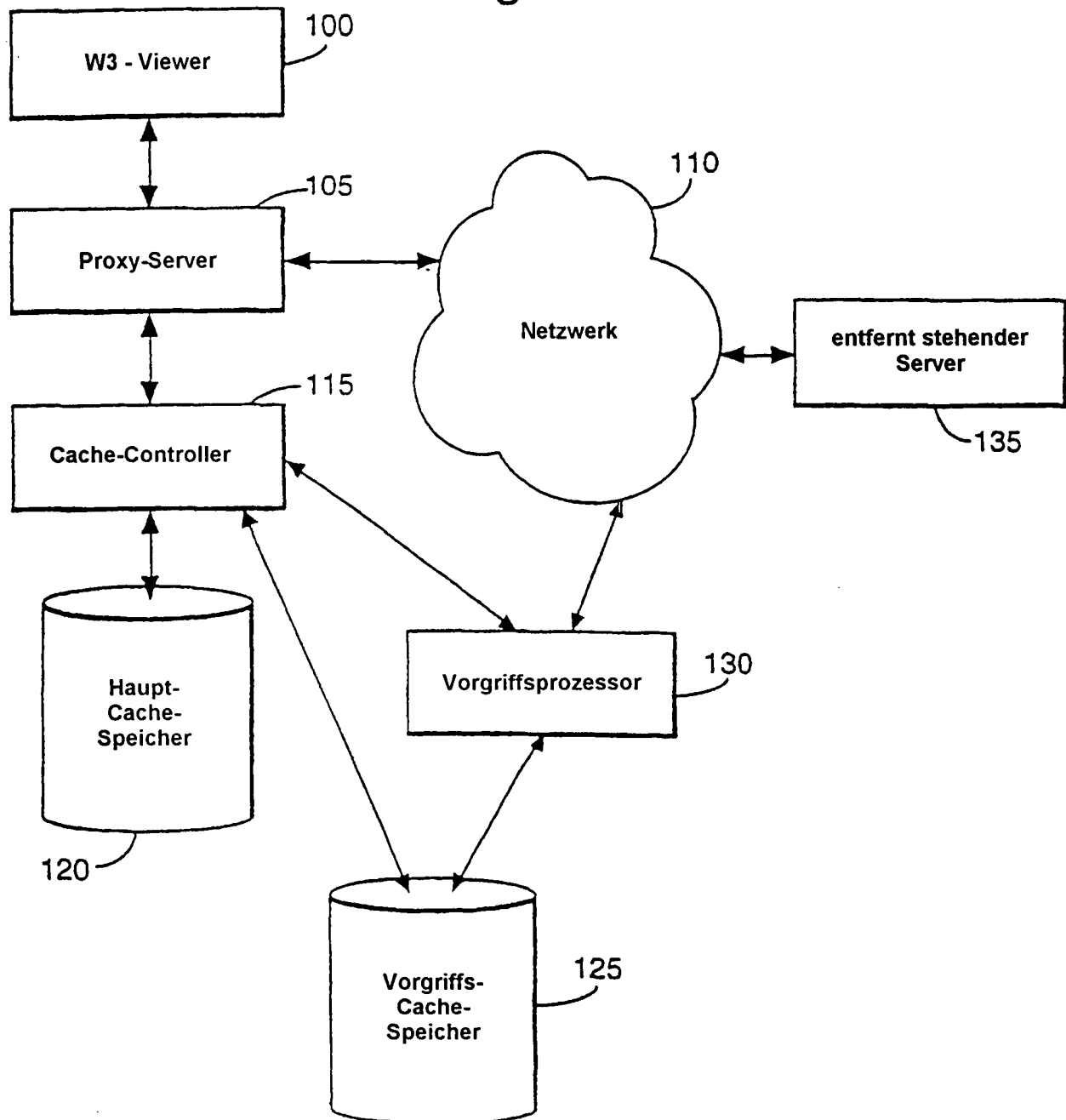
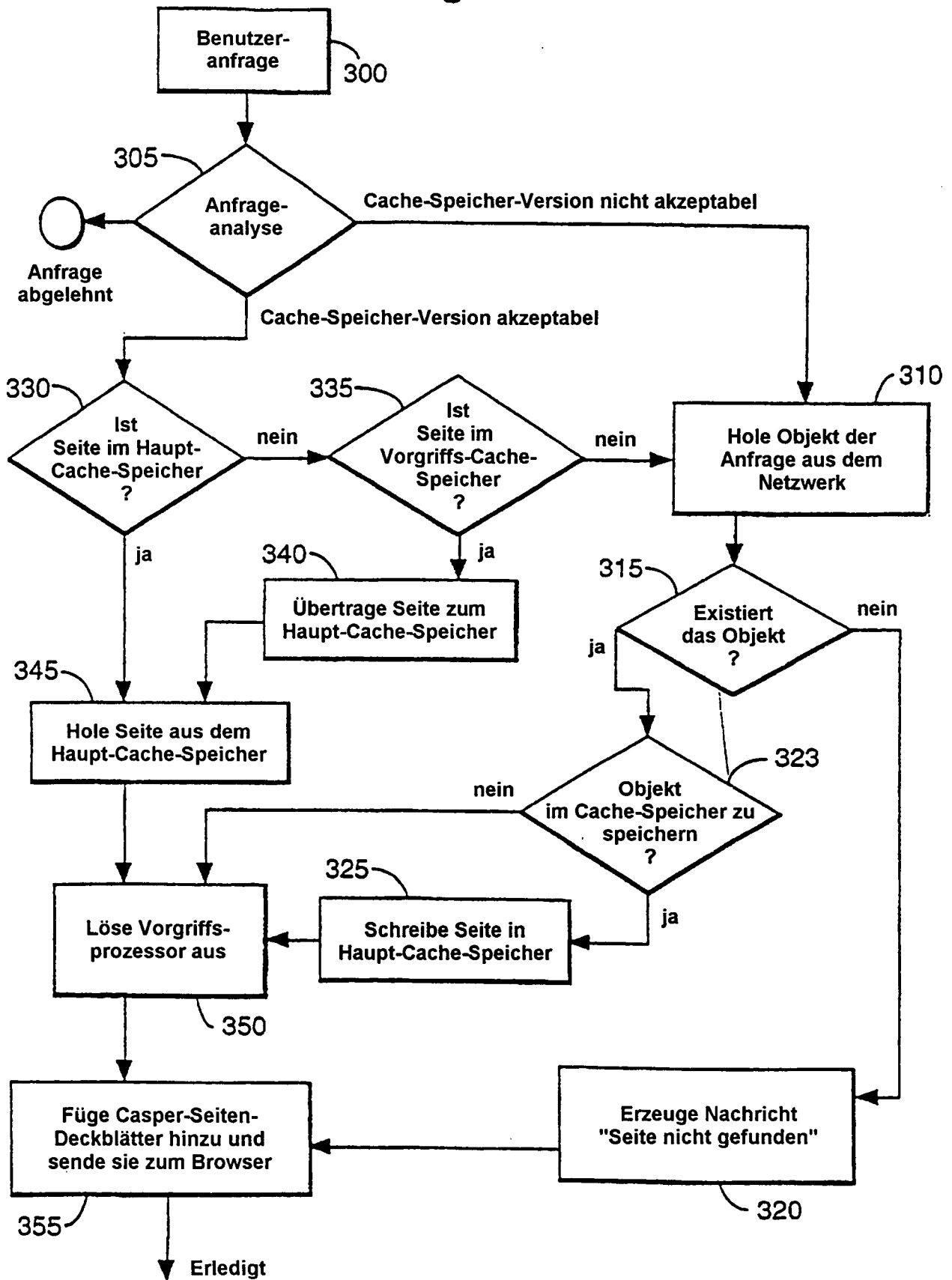


Fig.2



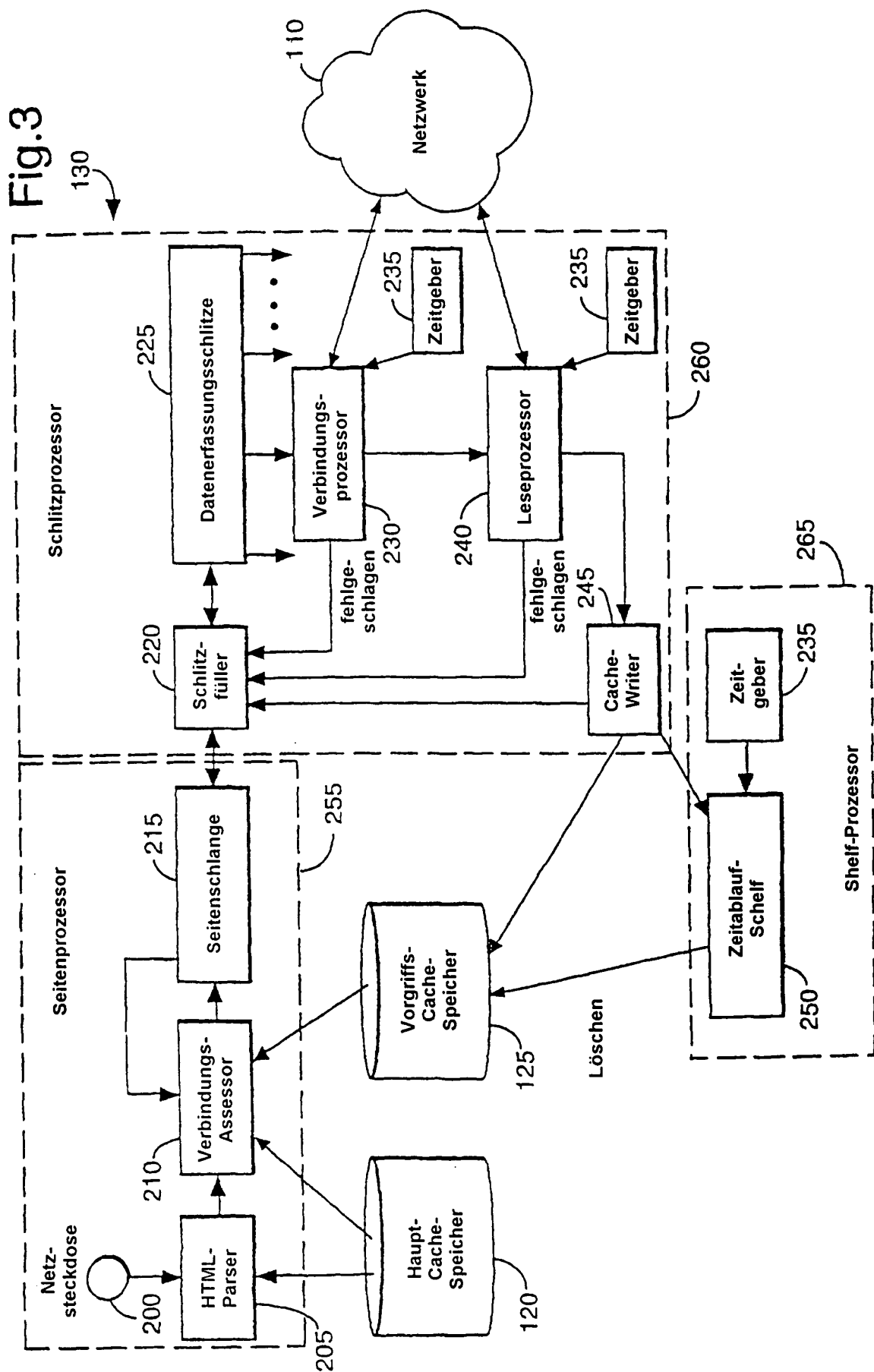


Fig.4

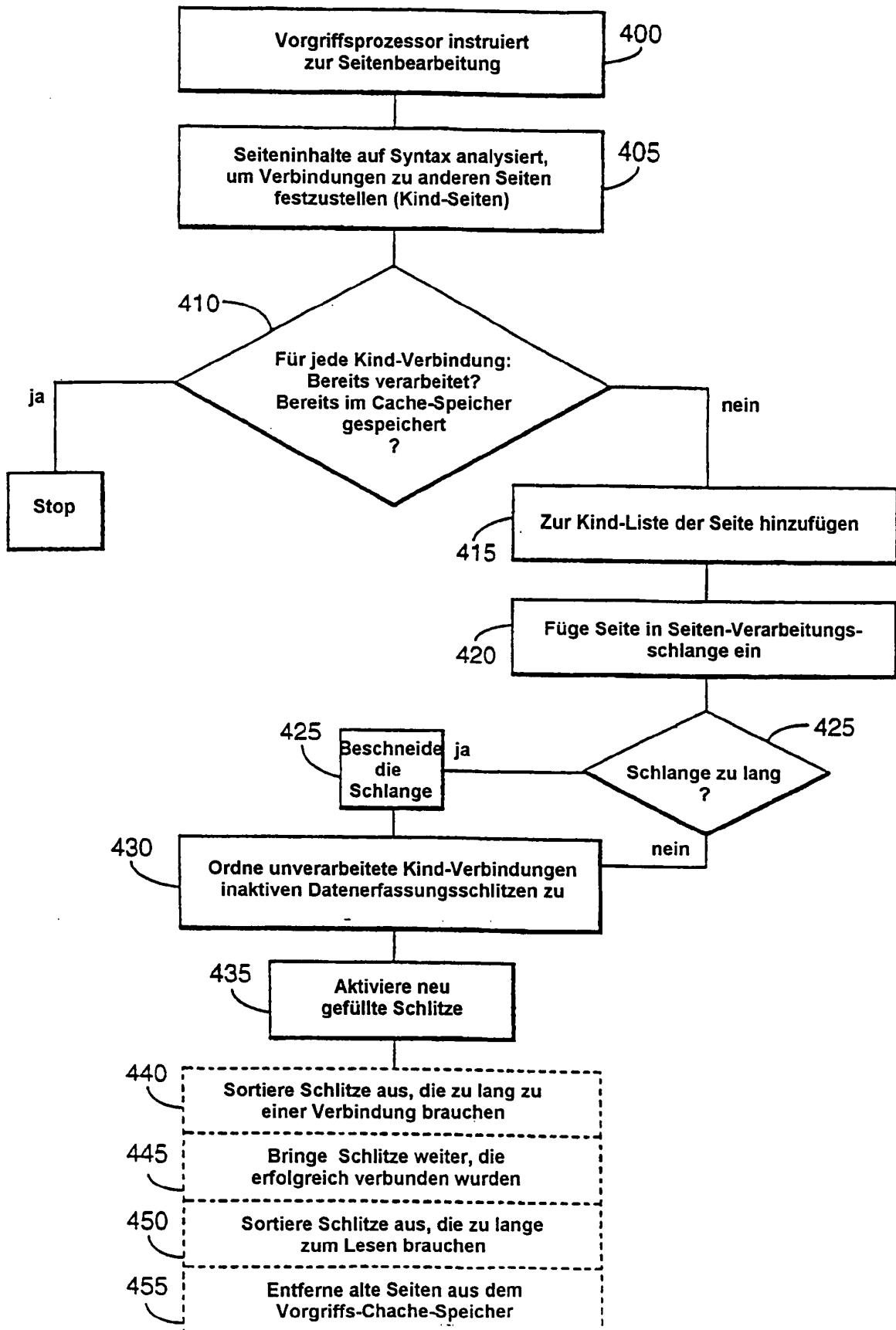
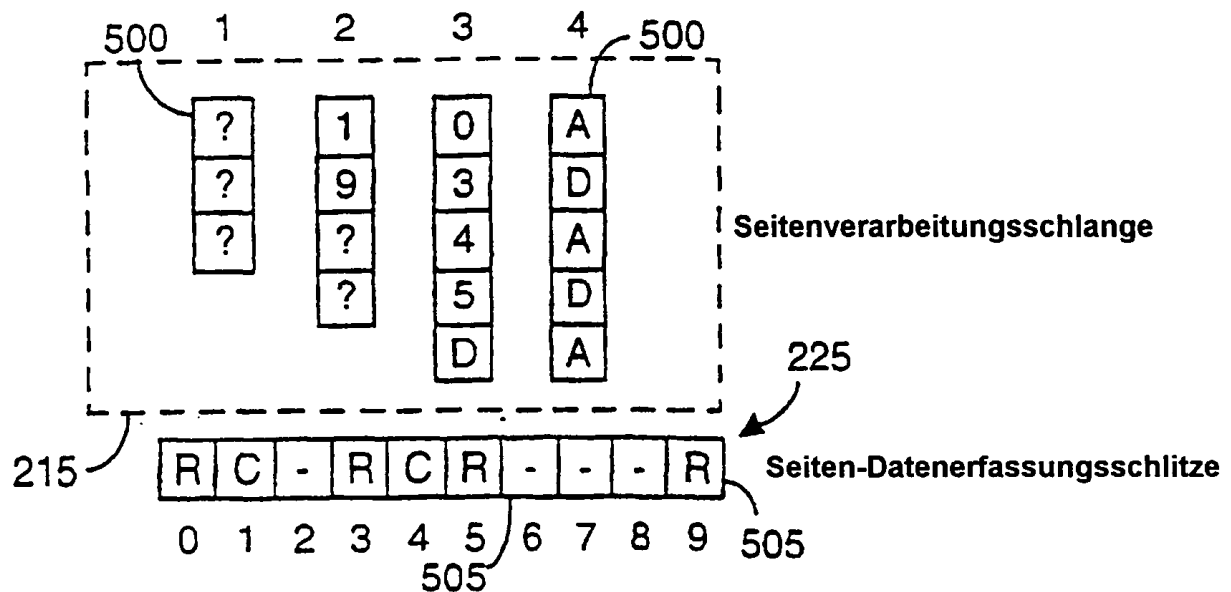


Fig.5



Warteschlangenschlüssel:

? = Nicht zugewiesen

= Schlitznummer

D = Erledigt

A = Abgebrochen

Schlitzschlüssel:

- = Inaktiv

C = Verbindungsaufbau

R = Beim Lesen

Fig.6

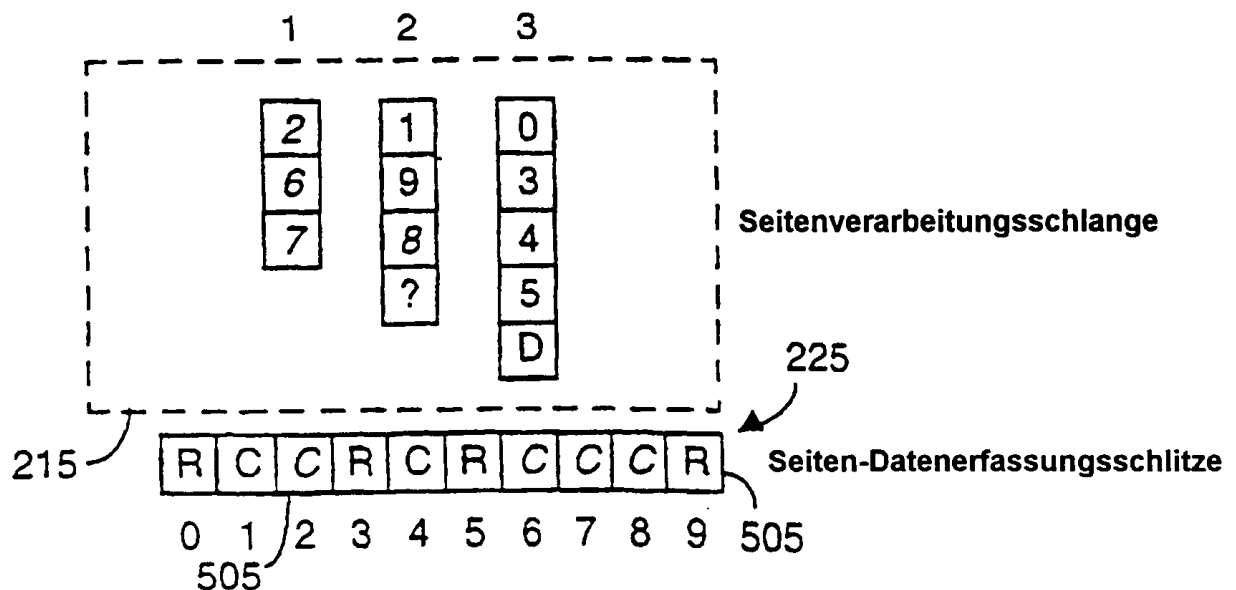


Fig.7

