



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 60 2005 004 872 T2** 2009.02.19

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 672 502 B1**

(51) Int Cl.⁸: **G06F 11/07** (2006.01)

(21) Deutsches Aktenzeichen: **60 2005 004 872.3**

(96) Europäisches Aktenzeichen: **05 111 406.4**

(96) Europäischer Anmeldetag: **29.11.2005**

(97) Erstveröffentlichung durch das EPA: **21.06.2006**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **20.02.2008**

(47) Veröffentlichungstag im Patentblatt: **19.02.2009**

(30) Unionspriorität:

18916 20.12.2004 US

(73) Patentinhaber:

Microsoft Corp., Redmond, Wash., US

(74) Vertreter:

**Grünecker, Kinkeldey, Stockmair &
Schwanhäusser, 80802 München**

(84) Benannte Vertragsstaaten:

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

(72) Erfinder:

**Jones, Brian M., Redmond, WA 98052, US; Liu,
Carol L., Redmond, WA 98052, US; Rothschilder,
Chad B., Redmond, WA 98052, US; McCaughey,
Robert R., Redmond, WA 98052, US; Villaron,
Shawn A., Redmond, WA 98052, US; Wu, Su-Piao
Bill, Redmond, WA 98052, US**

(54) Bezeichnung: **Verfahren und computerlesbarer Datenträger zum Überprüfen und Speichern eines elektronischen Dokuments**

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**QUERVERWEIS ZU VERWANDTEN ANMELDUNGEN**

[0001] Diese Anmeldung ist mit der derzeit eingereichten US-Patentanmeldung Nr. 11/018,914 (Anwaltsregister Nr. 60001.0448US01) verwandt, welche mit „Verfahren und computerlesbares Medium zum Laden der Inhalte einer Datei“ („Method and Computer-readable medium for loading the contents of a data file“) betitelt ist.

HINTERGRUND DER ERFINDUNG

[0002] Computer werden in der heutigen Gesellschaft tiefgreifend verwendet, um einer großen Vielfalt von Aufgaben und Unterhaltungszwecken zu dienen. Zum Beispiel werden Computer heutzutage für Spiele, Kommunikation, Forschung und eine geradezu endlose Vielfalt von anderen Anwendungen eingesetzt. Eine der meistverbreiteten Verwendungen von Computern von sowohl Betrieben als auch Einzelpersonen gleichermaßen, ist die Erschaffung von elektronischen und gedruckten Dokumenten. Es gibt Computeranwendungsprogramme für die Erstellung aller Arten von elektronischen Dokumenten, inklusive Tabellen, Präsentationen, textverarbeitender Dokumente, graphischer Dokumente, so wie Diagramme und digitale Bilder, Dokumente mit computerunterstützten Entwürfen und viele andere Arten von elektronischen Dokumenten.

[0003] Elektronische Dokumente umfassen oft Inhalte, die sehr wichtig sind. Darüber hinaus ist der Inhalt eines elektronischen Dokumentes in vielen Fällen schwer oder unmöglich wiederherzustellen, wenn er verloren geht. Zum Beispiel können oft erstellte hochkomplizierte Rechts-, Geschäfts-, Marketing- und technische Dokumente nicht einfach wiederhergestellt werden, wenn die Datei oder der Daten- bzw. der Dateiordner, der die Dokumente enthält, beschädigt oder zerstört wurde. Sogar in Fällen, in denen der Inhalt eines Dokumentes einfach wiederhergestellt werden könnte, kann es sehr frustrierend für den Benutzer sein, sogar nur einen kleinen Teil der Daten zu verlieren. Dementsprechend ist es sehr wichtig, dass die in elektronischen Dokumenten beinhalteten Daten gegen Beschädigung und Zerstörung geschützt werden.

[0004] Moderne Computersysteme beinhalten Fehlerüberprüfungen und andere Mechanismen, um vor der ungewollten Beschädigung oder dem Verlust eines Systemspeichers zu schützen.

[0005] Unglücklicherweise ist es sogar mit diesen vorhandenen Mechanismen möglich, dass ein im flüchtigen Systemspeicher abgelegtes Dokument vor dem Speichern des Dokumentes in einer Datei auf ei-

nem Massenspeicher beschädigt wird. Beschädigung kann als das Resultat von einem fehlerhaften Speicher, einer fehlerhaften Speichersteuerung, Speicherverwaltungsfehlern, Ladefehlern oder beschädigten Daten, einem Absturz des Anwendungsprogrammes und aus anderen Gründen auftreten. Weil der Verlust einer jeglichen Menge von Daten für einen Benutzer frustrierend sein kann und die Zeit und der Aufwand, die zur Wiederherstellung eines beschädigten Dokumentes erforderlich ist, oft sehr groß ist, ist es wichtig, dass so viele Daten wie möglich aus einem beschädigten Dokument zurückgewonnen werden können, das vor dem Speichern des Inhaltes des Speichers in einem Massenspeichergerät in dem flüchtigen Speicher abgelegt ist.

[0006] Unter Berücksichtigung dieser und anderer Abwägungen wurden die verschiedenen Ausführungsformen der vorliegenden Erfindung gemacht.

[0007] Die Druckschrift WO 02/19075 A bezieht sich auf ein System, dass eine öffentlich zugängliche, unabhängige und sichere Betreiberinternetseite vorsieht, die dem anonymen Client-PC ein herunterladbares Vermittlungsprogramm zur Verfügung stellt, wobei mit dem Vermittlungsprogramm innerhalb des Client-PC eine Registrierungsprüfsumme generiert wird, die auf dem zu registrierenden Dokument basiert. Zusätzlich generiert der Betreiber eine Registrierungsnummer und kann ein Registrierungszertifikat erstellen, die beide auf den Client herunterladbar sind. Diese Registrierung kann dazu verwendet werden, die Integrität des Dokumentes mit Bezug auf das Datum, die Zeit und den Inhalt des Dokumentes zu einem späteren Datum und einer späteren Zeit und durch einen jeglichen Client durch einen Zugang zu sowohl dem Dokument als auch der Registrierungsnummer zu verifizieren.

KURZE ZUSAMMENFASSUNG DER ERFINDUNG

[0008] Es ist das Ziel der vorliegenden Erfindung, das obige und andere Probleme durch ein Verfahren und ein computerlesbares Medium zum Speichern des Inhaltes eines Dokumentes zu lösen, das von einer Speicherstruktur in einem flüchtigen Speicher in den auf einem Massengerät gespeicherten Dateiordner bzw. die Datei gespeichert wird. Diese Aufgabe wird durch die Erfindung, wie durch die Gegenstände der unabhängigen Ansprüche 1 und 7 dargelegt, gelöst.

[0009] Ausführungsformen sind in den abhängigen Ansprüchen definiert.

[0010] Durch die Verwendung verschiedener Ausführungsformen der vorliegenden Erfindung werden beschädigte Teile (auch „Aufzeichnungen“ oder „Einträge“ – „Records“ genannt) der Speicherstruktur während des Speicherns einer Speicherstruktur iden-

tifiziert und ein Versuch, diese Teile zu reparieren, wird unternommen, wenn die beschädigten Teile nicht repariert werden können. Die nichtbeschädigten und reparierten Teile der Speicherstruktur werden dann in einen Dateiordner bzw. eine Datei auf dem Massenspeichergerät gespeichert. Wenn Teile der Speicherstruktur nicht repariert oder übersprungen werden können, wird ein Versuch unternommen, nur die in der Speicherstruktur beinhalteten Benutzerdaten zu speichern. Auf diese Art und Weise können die in der Speicherstruktur beinhalteten Benutzerdaten sogar in Fällen einer schweren Beschädigung des Restbestandes der Datenstruktur in einem Massenspeicher gespeichert werden.

[0011] Entsprechend einem Aspekt der Erfindung ist ein Verfahren zum Speichern einer in einem flüchtigen Speicher abgelegten Datenstruktur vorgesehen, der einen oder mehrere Teile zu einer Datei auf einem Massenspeichergerät beinhaltet. Entsprechend dem Verfahren werden eine Anzahl von Speichermodi zur Verfügung gestellt. In dem „normalen“ Speichermodus wird ein Versuch unternommen, jeden Teil der Speicherstruktur auf normale Art und Weise zu speichern. Der normale Speichermodus beinhaltet eine minimale Integritätsüberprüfung von jedem der Teile der Speicherstruktur, so dass die Daten schnell gespeichert werden können. Wenn während des normalen Modus ein Abschnitt der Speicherstruktur angetroffen wird, der fehlt oder beschädigt ist, wird ein zweiter Modus verwendet, welcher der „sichere“ Speichermodus genannt wird, um zu versuchen, die Teile der Speicherstruktur zu speichern. Ein Teil der Speicherstruktur kann als beschädigt und deswegen unspeicherbar betrachtet werden, wenn er einen Fehler oder einen Absturz des Anwendungsprogrammes verursacht, das versucht ihn zu speichern, wenn der Teil einen unerwarteten Datenwert beinhaltet, wenn dem Teil Daten fehlen, wenn der Teil ungültige Aufzeichnungen oder ungültige erweiterbare Auszeichnungssprache (extensible mark-up language – „XML“) beinhaltet, und aus anderen Gründen.

[0012] Im sicheren Modus wird eine ausführliche Integritätsprüfung von jedem Teil der Datenstruktur durchgeführt. In dem sicheren Speichermodus kann auch ein Versuch unternommen werden, die beschädigten Teile der Speicherstruktur zu reparieren. Alle Teile, die repariert werden können, werden dann abgespeichert. Wenn ein Teil der Speicherstruktur in dem sicheren Speichermodus angetroffen wird, der fehlt oder beschädigt ist und der nicht reparierbar ist, wird das Speichern des nicht reparierbaren Abschnittes übersprungen. Wenn Teile der Datenstruktur angetroffen werden, die nicht reparierbar sind und für welche eine Speicherung nicht übersprungen werden kann, wird ein dritter sicherer Modus verwendet, genannt der „minimale“ Speichermodus bzw. Modus minimalen Abspeicherns, um zu versuchen, bestimmte

Abschnitte der Datenstruktur zu speichern.

[0013] In dem minimalen Speichermodus werden nur die Teile der Speicherstruktur gespeichert, die Benutzerdaten beinhalten. Zum Beispiel können Benutzerdaten Textdaten oder numerische Daten umfassen, die von einem Benutzer eingegeben wurden. Als ein Beispiel, wenn die Speicherstruktur Daten für eine Tabelle beinhaltet, wird ein Versuch im minimalen Speichermodus unternommen, um nur die in den Zellen der Tabelle beinhalteten Daten zu speichern. Kein Versuch wird in dem minimalen Modus unternommen, um andere Typen von Daten zu speichern, die in der Speicherstruktur beinhaltet sein können, so wie eingebettete Objekte, Pivot-Tabellen, automatische Filter, Graphiken, Stile, Formate und Anwendungs- oder Benutzereinstellungen.

[0014] Gemäß anderen Ausführungsformen der Erfindung wird ein computerlesbares Medium auch zur Verfügung gestellt, auf welchem von einem Computer ausführbare Instruktionen gespeichert sind. Wenn die computerausführbaren Instruktionen von einem Computer ausgeführt werden, veranlassen sie den Computer dazu, einen ersten Speichermodus zum Speichern des Inhaltes einer Speicherstruktur zur Verfügung zu stellen, die einen oder mehrere Teile hat. In dem ersten Speichermodus wird eine minimale Integritätsprüfung der Teile der Speicherstruktur durchgeführt, während sie in einen Dateiordner auf einem Massenspeichergerät gespeichert werden. Die computerausführbaren Instruktionen veranlassen den Computer auch dazu, einen zweiten Speichermodus zum Speichern der Speicherstruktur zur Verfügung zu stellen, in dem eine ausführlichere Integritätsüberprüfung der Teile der Speicherstruktur als in dem ersten Speichermodus durchgeführt wird. In dem zweiten Speichermodus kann auch ein Versuch unternommen werden, Teile zu reparieren, die nicht speicherbar sind. Darüber hinaus wird in dem zweiten Speichermodus das Speichern von jeglichen nicht speicherbaren Daten übersprungen.

[0015] Die computerausführbaren Instruktion veranlassen den Computer auch dazu, mit dem Speichern der Inhalte einer Speicherstruktur in dem ersten Speichermodus zu beginnen. Wenn ein Teil der Speicherstruktur in dem ersten Speichermodus als nicht speicherbar festgelegt wird, schaltet der Computer in den zweiten Speichermodus und versucht, die Speicherstruktur in diesem Speichermodus zu speichern. Wenn in dem zweiten Speichermodus ein nicht speicherbarer Teil angetroffen wird, der repariert werden kann, wird der nicht speicherbare Teil repariert und gespeichert. Wenn der nicht speicherbare Teil nicht repariert werden kann, wird das Speichern des nicht speicherbaren Teils übersprungen.

[0016] Gemäß einer Ausführungsform der Erfindung veranlassen die computerausführbaren Instruk-

tionen den Computer auch, einen dritten Speichermodus zur Verfügung zu stellen, worin nur die Abschnitte der Speicherstruktur gespeichert werden, die Benutzerdaten beinhalten. Wenn in dem zweiten Speichermodus festgelegt wird, dass ein Teil der Speicherstruktur nicht speicherbar ist und dass der nicht speicherbare Teil nicht repariert oder übersprungen werden kann, wird ein Versuch unternommen, den Inhalt der Speicherstruktur in dem dritten Speichermodus zu speichern.

[0017] Die Erfindung kann als ein Rechen- bzw. Computerprozess, ein Computersystem oder jedem herstellbaren Artikel implementiert werden, so wie einem Computerprogrammprodukt oder computerlesbaren Medien. Das Computerprogrammprodukt kann ein Computerspeichermedium sein, das von einem Computersystem lesbar ist und das ein Computerprogramm aus Instruktionen zum Ausführen eines Computerprozesses enkodiert. Das Computerprogrammprodukt kann auch ein ausgebreitetes Signal auf einem Träger sein, der von einem Computersystem lesbar ist und ein Computerprogramm aus Instruktionen zum Ausführen eines Computerprozesses enkodiert.

[0018] Diese und andere Merkmale sowie Vorteile, welche die vorliegende Erfindung charakterisieren, werden durch das Lesen der folgenden detaillierten Beschreibung und einer Betrachtung der begleitenden Zeichnung offensichtlich.

KURZE BESCHREIBUNG DER VERSCHIEDENEN ANSICHTEN DER ZEICHNUNGEN

[0019] [Fig. 1](#) ist ein Diagramm der Architektur eines Computersystems, das ein Computersystem darstellt, welches in einer der verschiedenen Ausführungsformen der Erfindung verwendet und zur Verfügung gestellt wird;

[0020] [Fig. 2](#) ist ein Blockdiagramm, das Aspekte einer Speicherstruktur und der verschiedenen Speichermodi darstellt, die von den Ausführungsformen der Erfindung zur Verfügung gestellt werden; und

[0021] [Fig. 3A–Fig. 3B](#) sind Flussdiagramme, die den veranschaulichenden Prozess zum Speichern einer Speicherstruktur gemäß den verschiedenen Ausführungsformen der Erfindung zeigen.

DETAILLIERTE BESCHREIBUNG DER ERFINDUNG

[0022] Mit Bezug auf die Zeichnungen, in denen die gleichen Bezugszeichen die gleichen Elemente repräsentieren, werden nun verschiedene Aspekte der vorliegenden Erfindung beschrieben. Insbesondere [Fig. 1](#) und die dazugehörige Diskussion sind dazu vorgesehen, eine kurze generelle Beschreibung ei-

ner passenden Computerumgebung zur Verfügung zu stellen, in der Ausführungsformen der Erfindung implementiert werden können. Während die Erfindung in einem generellen Kontext von Programmmodulen beschrieben wird, die auf einem Betriebssystem auf einem Personalcomputer ausgeführt werden, werden Fachmänner erkennen, dass die Erfindung auch in Kombination mit anderen Typen von Computersystemen und Computerprogrammen implementiert werden kann.

[0023] Generell beinhalten Programmmodule Routinen, Programme, Komponenten, Datenstrukturen und andere Typen von Strukturen, die bestimmte Aufgaben ausführen oder bestimmte abstrakte Datentypen implementieren. Darüber hinaus werden es Fachmänner anerkennen, dass die Erfindung mit anderen Konfigurationen von Computersystemen ausgeführt werden kann, die in der Hand gehaltene Geräte (hand-held devices), Multiprozessorsysteme, Mikroprozessorbasierte und programmierbare Benutzerelektronik, Minicomputer, Großrechner und dergleichen beinhalten. Die Erfindung kann auch in verteilten Computerumgebungen ausgeführt werden, worin Aufgaben von fernab liegenden Verarbeitungsgeschäften ausgeführt werden, die durch ein Kommunikationsnetzwerk miteinander verbunden sind. In einer verteilten Computerumgebung können Programmmodule sowohl in lokalen als auch fernab liegenden Massenspeichergeräten angeordnet sein.

[0024] Mit Bezug auf [Fig. 1](#) wird nun eine veranschaulichende Architektur eines Computers **2** beschrieben, der in verschiedenen Ausführungsformen der Erfindung verwendet wird. Die in [Fig. 1](#) gezeigte Computerarchitektur stellt einen konventionellen Desktop- oder Laptopcomputer dar, der eine zentrale Prozessoreinheit **5** (central processing unit – „CPU“), einen Systemspeicher **7**, beinhaltend einen Zufallszugriffsspeicher **9** (random access memory – „RAM“) und einen nur lesbaren Speicher (read-only memory – „ROM“) **11**, und eine Systemdatenübertragungseinrichtung **12** (system bus) beinhaltet, die den Speicher mit der CPU **5** verbindet. Ein rudimentäres Ein- und Ausgabesystem (basic input/output system), das in dem ROM **11** gespeichert ist, beinhaltet die rudimentären Routinen, die dabei behilflich sind, Informationen zwischen Elementen innerhalb des Computers zu übertragen, so wie während des Hochfahrens. Der Computer **2** beinhaltet des Weiteren ein Massenspeichergerät **14** zum Speichern eines Betriebssystems **16** sowie zum Speichern von Anwendungsprogrammen und anderen Programmmodulen, die weiter unten ausführlicher beschrieben werden.

[0025] Das Massenspeichergerät **14** ist mit der CPU **5** über eine Massenspeichersteuerung (mass storage controller – nicht gezeigt) mit dem Bus **12** verbunden. Das Massenspeichergerät **14** und seine assoziierten Computerlesbaren Medien stellen einen

nichtflüchtigen Speicher für den Computer **2** zur Verfügung. Obwohl die Beschreibung der computerlesbaren Medien, die hierin enthalten ist, sich auf Massenspeichergeräte bezieht, so wie Festplatten oder CD-ROM-Laufwerke, sollte es von Fachmännern verstanden werden, dass computerlesbare Medien alle zur Verfügung stehenden Medien sein können, auf die von dem Computer **2** zugegriffen werden kann.

[0026] Nur beispielhaft und nicht einschränkend kann ein computerlesbares Medium Computerspeichermedien und Kommunikationsmedien umfassen. Computerspeichermedien beinhalten flüchtige und nichtflüchtige, entnehmbare und nicht entnehmbare Medien, welche durch jegliche Verfahren oder Technologien zum Speichern von Informationen implementiert werden, so wie computerlesbare Instruktionen, Datenstrukturen, Programmmodule oder andere Daten. Computerspeichermedien beinhalten, aber sind nicht beschränkt auf RAM, ROM, EPROM, EEPROM, Flash-Speicher oder andere Festspeichertechnologien, CD-ROM, digital versatile disks („DVD“) oder andere optische Speicher, magnetische Kassetten, magnetische Bänder, magnetische Scheibenspeicher oder andere magnetische Speichervorrichtungen oder jegliche andere Medien, die dazu verwendet werden können, die gewünschten Informationen zu speichern, auf welche mit dem Computer **2** zugegriffen werden kann.

[0027] Gemäß den verschiedenen Ausführungsformen der Erfindung kann der Computer **2** in einer vernetzten Umgebung betrieben werden, die logische Verbindungen zu fernab liegenden Computern durch ein Netzwerk **18**, so wie das Internet, verwendet. Der Computer **2** kann sich mit dem Netzwerk **18** durch eine Netzwerkschnittstelleneinheit **20** verbinden, die mit dem Bus **12** verbunden ist. Es sollte anerkannt werden, dass die Netzwerkschnittstelleneinheit **20** auch dazu verwendet werden kann, andere Typen von Netzwerken von abliegenden Computersystemen anzubinden. Der Computer **2** kann auch einen Kontrollschaltkreis **22** für die Ein- und Ausgabe beinhalten (input/output controller), um Eingaben von einer Anzahl von anderen Geräten zu empfangen und zu verarbeiten, inklusive einer Tastatur, Maus oder eines elektronischen Eingabestiftes (nicht in [Fig. 1](#) gezeigt). Ebenso kann der Eingabe-/Ausgabekontrollschaltkreis (input/output controller) **22** eine Ausgabe an einen Bildschirm, einen Drucker oder andere Typen von Ausgabegeräten zur Verfügung stellen.

[0028] Wie oben kurz erwähnt wurde, können eine Anzahl von Programmmodulen und Dateiordnern in dem Massenspeichergerät **14** und dem RAM **9** auf dem Computer **2** gespeichert sein, inklusive eines Betriebssystems **16**, das zum Steuern des Betriebes eines vernetzten Personalcomputers geeignet ist, so wie das WINDOWS XP Betriebssystem von der MIC-

RO-SOFT CORPORATION of Redmond, Washington. Das Massenspeichergerät **14** und das RAM **9** können auch eines oder mehrere Programmmodule speichern. Insbesondere können das Massenspeichergerät **14** und das RAM **9** ein Tabellenkalkulationsanwendungsprogramm **10** speichern. Wie es Fachmännern bekannt sein wird, ist das Tabellenkalkulationsanwendungsprogramm **10** betriebsbereit bzw. betreibbar, um die Funktionalität zum Erschaffen und Editieren von elektronischen Tabellen zur Verfügung zu stellen.

[0029] Entsprechend einer Ausführungsform der Erfindung beinhaltet das Tabellenkalkulationsanwendungsprogramm **10** das EXCEL-Tabellenkalkulationsanwendungsprogramm von der MICROSOFT CORPORATION. Es sollte allerdings erkannt werden, dass andere Tabellenkalkulationsanwendungsprogramme von anderen Herstellern verwendet werden können, um die verschiedenen Aspekte der vorliegenden Erfindung zu verkörpern. Es sollte auch anerkannt werden, dass, obwohl die Ausführungsformen der Erfindung, die hierin beschrieben sind, in dem Kontext eines Tabellenkalkulationsanwendungsprogrammes vorgestellt werden, die Erfindung jedoch mit jedem anderen Typ eines Anwendungsprogrammes verwendet werden kann, das Daten in einen Dateiordner bzw. eine Datei speichert. Zum Beispiel können die hierin beschriebenen Ausführungsformen der Erfindung innerhalb eines Textverarbeitungsanwendungsprogrammes, eines Präsentationsanwendungsprogrammes, eines Zeichnungs- oder computerunterstützten Designanwendungsprogrammes oder eines Datenbankanwendungsprogrammes angewendet werden.

[0030] Wie in [Fig. 1](#) gezeigt ist, können Teile des Tabellenkalkulationsanwendungsprogrammes **10** während der Ausführung in den flüchtigen RAM **9** gespeichert werden. Darüber hinaus kann das Tabellenkalkulationsanwendungsprogramm **10** in Verbindung mit der Erschaffung und dem Editieren von Tabellen bzw. Tabellenkalkulationsdokumenten einen Teil des RAM **9** verwenden, um das Dokument zu speichern. Insbesondere das Tabellenkalkulationsanwendungsprogramm **10** kann einen oder mehrere Speicherstrukturen **25** verwenden, um das Tabellenkalkulationsdokument repräsentierende Daten zu speichern. Von Zeit zu Zeit ist das Tabellenkalkulationsanwendungsprogramm **10** entweder als Reaktion auf eine Benutzeranfrage oder auf eine automatisierte Art und Weise betriebsfähig bzw. betriebsbereit, den Inhalt der Speicherstruktur **25** in einen auf dem Massenspeichergerät **14** gespeicherten Dateiordner bzw. eine Datei **24** zu speichern. Die Datei **24** beinhaltet Daten, die verschiedene Aspekte des Tabellenkalkulationsdokumentes beinhalten, so wie Benutzerdaten, die den Inhalt der Zellen der Tabelle beinhalten, Anwendungseinstellung, Formatinformationen und andere Daten, die zu den verschiedenen

Merkmale gehören, die von dem Tabellenkalkulationsanwendungsprogramm **10** zur Verfügung gestellt werden. Wie weiter unten mit Bezug auf die [Fig. 2–Fig. 3B](#) noch ausführlicher beschrieben wird, wird ein Verfahren zum Speichern der Inhalte der Speicherstruktur **25** in die Datei **24** von dem Tabellenkalkulationsanwendungsprogramm **10** verwendet, das die Wahrscheinlichkeit für eine Beschädigung in der Speicherstruktur **25** berechnet und das versucht, die Menge der Benutzerdaten zu maximieren, die in der Datei **24** gespeichert werden, sogar wenn die Speicherstruktur **25** beschädigt wurde.

[0031] Sich nun [Fig. 2](#) zuwendend, werden zusätzliche Details zur Verfügung gestellt, welche den Inhalt der Speicherstrukturen **25** und den Betrieb der Speichermechanismen betrachten, die von dem Tabellenkalkulationsanwendungsprogramm **10** verwendet werden. Wie in [Fig. 2](#) gezeigt ist, sind die Speicherstrukturen **25** in eine Anzahl von Teilen **26A–26N** unterteilt. Jeder der Teile **26A–26N** wird zum Speichern von Informationen verwendet, die sich auf eines oder mehrere Merkmale beziehen, die von dem Tabellenkalkulationsanwendungsprogramm **10** unterstützt werden. Darüber hinaus können die Informationen von verschiedenen, aber miteinander verwandten Merkmalen in einem einzelnen der Teile **26A–26N** gespeichert werden. Zum Beispiel werden die Daten für die Merkmale A–C wie in [Fig. 2](#) gezeigt in dem Teil **26A** gespeichert. Die Daten für das Merkmal D werden im Teil **26B** gespeichert. Die Daten für die Merkmale E–G werden in dem Teil **26C** gespeichert und so weiter. Benutzerdaten können in jedem der Teile **26A–26N** gespeichert sein. Es sollte gewürdigt werden, dass die in der Speicherstruktur **25** gespeicherten Daten in einer nicht kontinuierlichen Art und Weise gespeichert werden können und dass die Daten für miteinander verwandte Merkmale in separaten Speicherorten gespeichert sein können.

[0032] Wie oben kurz beschrieben und in [Fig. 2](#) gezeigt ist, ist es für die innerhalb der Teile **26A–26N** beinhalteten Daten möglich, beschädigt zu werden. Eine Beschädigung (corruption) kann das Ergebnis eines fehlerhaften Speichers, einer fehlerhaften Speichersteuerung bzw. eines fehlerhaften Speicherkontrollschaltkreises, von Speicherverwaltungsfehlern, Ladefehlern oder beschädigten Daten, eines Absturzes des Anwendungsprogrammes oder aus anderen Gründen sein. Die Daten für einen bestimmten Abschnitt können auch als verlorengegangen bestimmt werden. Ein Teil der Speicherstruktur **25** kann als beschädigt und deswegen unspeicherbar angesehen werden, wenn der Teil einen Fehler in dem oder einen Absturz des Anwendungsprogrammes verursacht, das versucht, ihn zu speichern, wenn der Abschnitt einen nicht erwarteten Datenwert beinhaltet, wenn dem Abschnitt Daten fehlen, wenn der Abschnitt eine ungültige Aufzeichnung oder ungültige erweiterbare Auszeichnungssprache (extensible

markup language – „XML“) beinhaltet oder aus anderen Gründen. In den illustrativen Speicherstrukturen **25**, die in [Fig. 2](#) gezeigt sind, wurden die Teile **26B** und **26C** beschädigt.

[0033] Wie hierin beschrieben wird, sind Teile der Speicherstruktur **25** von dem Tabellenkalkulationsanwendungsprogramm **10** speicherbar, abgesehen von der Beschädigung der Teile **26B** und **26D**. [Fig. 2](#) zeigt auch diesen Speicherprozess, der die illustrativen Speicherstrukturen **25** verwendet. Im Besonderen beginnt das Tabellenkalkulationsanwendungsprogramm **10** mit dem Speichern der Speicherstruktur **25** in einem normalen Speichermodus. In dem normalen Speichermodus wird eine minimale Integritätsüberprüfung der Teile **26A–26N** der Speicherstruktur **25** durchgeführt. Wenn ein beschädigter Teil der Speicherstruktur **25** während des Speicherns im normalen Modus angetroffen wird, schaltet das Tabellenkalkulationsanwendungsprogramm **10** in einen sicheren Speichermodus und fängt damit an, die Speicherstruktur **25** von vorn zu speichern. Zum Beispiel wenn der beschädigte Teil **26B**, wie in [Fig. 2](#) gezeigt ist, in dem normalen Speichermodus angetroffen wird, wird in den sicheren Speichermodus gewechselt und die Speicherung beginnt von vorn am Anfang der Speicherstruktur **25**. Es sollte gewürdigt werden, dass entsprechend den Ausführungsformen der Erfindung, das Speichern von zusätzlichen Teilen der Speicherstruktur **25** nicht an den Anfang zurückzukehren braucht.

[0034] In dem sicheren Speichermodus wird gegenüber dem normalen Speichermodus eine zusätzliche Integritätsprüfung der Teile **26A–26N** der Speicherstruktur **25** durchgeführt. Wenn ein beschädigter Abschnitt während des Speicherns im sicheren Modus angetroffen wird, wird zusätzlich ein Versuch unternommen, den beschädigten Teil zu reparieren. Wenn der beschädigte Teil repariert werden kann, wird dieser Teil gespeichert. Wenn der beschädigte Teil nicht repariert werden kann, dann wird das Speichern des beschädigten Teils übersprungen. Zum Beispiel, wie in [Fig. 2](#) gezeigt ist, ist der Teil **26B** beschädigt und kann nicht gespeichert werden. Deswegen wird das Speichern des Teils **26B** übersprungen und der Teil **26C** wird gespeichert.

[0035] Nachdem der Teil **26C** gespeichert wurde, wird dann ein Versuch unternommen, den Teil **26D** zu speichern. Allerdings, wie in [Fig. 2](#) gezeigt ist, ist der Teil **26D** beschädigt. Dementsprechend wird ein Versuch unternommen, den Teil **26D** zu reparieren. Wenn der Teil **26D** nicht gespeichert werden kann, wird das Speichern des Teils **26D** übersprungen und dieser Prozess wird fortgesetzt, bis die übrig bleibenden Teile repariert oder übersprungen wurden. Gemäß einer Ausführungsform der Erfindung kann die Speicherung der Speicherstrukturen **25** an den Anfang der Speicherstruktur **25** zurückkehren, nachdem

ein nicht speicherbarer Teil angetroffen wurde und für nicht reparierbar erklärt wurde. Dies ist in [Fig. 2](#) dargestellt. Wieder an den Anfang der Speicherstruktur **25** auf diese Weise zurückzukehren, erlaubt das Speichern von anderen Teilen der Datei **24**, die im Zusammenhang mit einem nicht speicherbaren Teil stehen, der zu überspringen ist, sogar wenn die in Zusammenhang stehenden Teile nicht beschädigt sind.

[0036] Wenn während des Speicherns der Speicherstruktur **25** ein Teil angetroffen wird, der nicht speicherbar und nicht reparierbar ist, kann das Tabellenkalkulationsanwendungsprogramm **10** in einen dritten Speichermodus umschalten, welcher der minimale Speichermodus bzw. Modus minimalen Abspeicherns genannt wird. In dem minimalen Speichermodus wird ein Versuch unternommen, nur die Benutzerdaten aus den Speicherstrukturen **25** zu speichern. Im Besonderen wird unter Betrachtung von einem Textdokument ein Versuch unternommen, nur den Text des Dokumentes zu speichern. Unter Bezug auf ein Tabellen- bzw. Tabellenkalkulationsdokument wird ein Versuch unternommen, den Inhalt der Zellen des Tabellenkalkulationsdokumentes zu speichern, welche die Dateneingaben von dem Benutzer, Formeln und von Formeln generierte Daten beinhalten. Auf diese Art und Weise können, sogar wenn Teile der Speicherstruktur beschädigt sind, einige oder alle der Benutzerdaten zurück gewonnen und gespeichert werden. Dieser Prozess ist durch die gepunktete Linie in [Fig. 2](#) dargestellt und würde ausgeführt werden, wenn der Teil **26D** für nicht speicherbar und nicht reparierbar bestimmt wird. Zusätzliche Details, welche diesen Prozess betrachten, werden unten mit Bezug auf die [Fig. 3A–Fig. 3B](#) zur Verfügung gestellt.

[0037] Mit Bezug auf die [Fig. 3A–Fig. 3B](#) wird nun die Routine **300** erklärt werden, welche einen von dem Tabellenkalkulationsanwendungsprogramm **10** zum Speichern des Inhalts einer Datenstruktur **25** verwendeten Prozess darstellt. Wenn die Diskussion der Routinen, die hierin präsentiert werden, gelesen wird, sollte es gewürdigt werden, dass die logischen Operationen von verschiedenen Ausführungsformen der vorliegenden Erfindung als eine Sequenz von computerimplementierten Schritten oder Programmmodulen implementiert (1) werden, die auf einem Computersystem und/oder (2) als miteinander verbundene maschinelle logische Schaltkreise oder Schaltkreismodule innerhalb des Computersystems laufen. Die Implementation ist abhängig von den Anforderungen an die Arbeitsleistung des Computersystems auswählbar, das die Erfindung implementiert. Entsprechend werden die logischen Operationen, die in den [Fig. 3A–Fig. 3B](#) dargestellt sind und welche die Ausführungsformen der darin beschriebenen vorliegenden Erfindung ausmachen, verschiedenartig als Operationen, strukturelle Geräte, Schrit-

te oder Module bezeichnet. Es wird von einem Fachmann erkannt werden, dass diese Operationen, strukturellen Geräte, Schritte und Module als Software, in firmware, in digitaler Logik für spezielle Zwecke und als eine Kombination davon implementiert werden können, ohne von dem Wesen und dem Ziel der vorliegenden Erfindung abzuweichen, wie es in den hierin vorgegebenen Ansprüchen ausgedrückt wird.

[0038] Es sollte gewürdigt werden, dass die Routine **300** während ihrer Ausführung verschiedene Variablen verwendet. Insbesondere hält die Variable „mode“ mit dem aktuellen Speichermodus Schritt. Diese Variable kann entweder zu „safe“ („sicher“), „normal“ oder „minimal“ gesetzt werden. Die Variable „skip counter“ („Überspringzähler“) hält mit den Speicherstrukturen **25** Schritt, die übersprungen werden sollten, wenn das Speichern der Speicherstrukturen **25** an den Anfang zurückkehrt, nachdem ein beschädigter Teil angetroffen wurde. Eine Variable „number of records to skip“ („Anzahl der zu überspringenden Aufzeichnungen“) beschreibt die aktuelle Anzahl von Abschnitten, die während des aktuellen Speicherversuchs übersprungen werden sollten. Eine Variable „current record“ („aktuelle Aufzeichnung“) identifiziert den aktuellen Abschnitt innerhalb der Daten, während dieser verarbeitet wird. Es sollte gewürdigt werden, dass mehr oder weniger Variablen zum Ausführen der gleichen Aufgabe verwendet werden können. Darüber hinaus sollte erkannt werden, dass die in den [Fig. 3A](#) und [Fig. 3B](#) dargestellte Routine **300** nur eine mögliche Implementation der Erfindung repräsentiert und dass viele andere Implementationen für einen Fachmann offensichtlich sein werden.

[0039] Die Routine **300** beginnt bei jeder Operation **302**, **304** oder **306**. Insbesondere kann entsprechend den Ausführungsformen der Erfindung eine Benutzerschnittstelle zur Verfügung gestellt werden, die es einem Benutzer erlaubt, auszuwählen, ob ein Dokument normal gespeichert wird (Operation **304**), ob es im sicheren Speichermodus gespeichert wird (Operation **302**) oder ob es im minimalen Speichermodus (**306**) gespeichert wird. Diese Benutzerschnittstelle kann dem Benutzer angezeigt werden, wenn der Benutzer fordert, das Dokument zu speichern. Basierend auf der Auswahl des Benutzers innerhalb der Benutzerschnittstelle beginnt die Routine **300** ihren Betrieb an einer der Operationen **302**, **304** oder **306**.

[0040] Wenn das Speichern im sicheren Modus zu beginnen ist, beginnt die Routine **300** bei der Operation **302**, wobei die Variable für den Modus auf „safe“ gesetzt wird. Die Routine **300** macht dann mit der Operation **308** weiter. Wenn mit dem Speichern im normalen Speichermodus zu beginnen ist, beginnt die Routine **300** bei der Operation **304**, wobei die Variable für den Modus zu „normal“ gesetzt wird. Die Routine **300** macht dann von der Operation **304** bis zur Operation **308** weiter. Wenn mit dem Speichern

im minimalen Speichermodus zu beginnen ist, beginnt die Routine bei der Operation **306**, wobei die Variable für den Modus zu „minimal“ gesetzt wird. Von der Operation **306** an setzt sich die Routine **300** bis zur Operation **348** fort, wie unten beschrieben ist.

[0041] Bei der Operation **308** wird die Variable des Zählers der Übersprünge bzw. Auslassungen initialisiert, um anzuzeigen, dass keine Aufzeichnungen übersprungen bzw. ausgelassen werden sollten. Die Routine **300** macht dann mit der Operation **310** weiter, wobei die aktuelle Aufzeichnung zu der ersten Aufzeichnung in der Speicherstruktur gesetzt wird. Die Variable der Anzahl der Aufzeichnungen, die zu überspringen sind, ist der Anzahl der zu überspringenden Aufzeichnungen gleichgesetzt. Bei dem ersten Durchlauf setzt dies die Anzahl der zu überspringenden Aufzeichnungen gleich Null. Von der Operation **310** fährt die Routine **300** mit der Operation **312** fort.

[0042] Bei der Operation **312** wird ein Versuch unternommen, die aktuelle Aufzeichnung im aktuellen Modus zu speichern. Zum Beispiel wenn die Variable für den Modus gleich „normal“ ist, wird eine minimale Integritätsüberprüfung der zu speichernden Abschnitte ausgeführt. Wenn die Variable gleich „safe“ gesetzt ist, wird eine zusätzliche Integritätsüberprüfung durchgeführt. Von der Operation **312** fährt die Routine **300** mit der Operation **314** fort, wobei eine Festlegung gemacht wird, ob die aktuelle Aufzeichnung nicht speicherbar ist (d. h. entweder beschädigt oder fehlend). Wenn die aktuelle Aufzeichnung speicherbar ist, dann zweigt die Routine **300** zur Operation **316** ab, wo eine Festlegung gemacht wird, ob mehr Aufzeichnungen verbleiben, die zu speichern sind. Wenn mehr Aufzeichnungen vorhanden sind, zweigt die Routine **300** von der Operation **316** zur Operation **318** ab, wobei die Variable für die aktuelle Aufzeichnung zu der nächsten Aufzeichnung in der Speicherstruktur **25** gesetzt wird. Die Routine **300** fährt dann mit der Operation **312** fort, wo die nächste Aufzeichnung gespeichert wird. Wenn bei der Operation **316** festgelegt wird, dass keine zusätzlichen Aufzeichnungen überbleiben, um gespeichert zu werden, zweigt die Routine **300** zur Operation **320** ab, wo sie endet. Auf diese Art und Weise werden alle Aufzeichnungen im aktuellen Modus gespeichert, wenn keine beschädigten oder fehlenden Aufzeichnungen existieren.

[0043] Es sollte erkannt werden, dass in Ausführungsformen der Erfindung einige Integritätsüberprüfungen (integrity checks) auf der Merkmalsebene (feature level) durchgeführt werden können, die der Aufzeichnungsebene (record level) gegenübersteht.

[0044] Um solche Integritätsüberprüfungen auf der Merkmalsebene durchzuführen, wird ein Versuch unternommen, alle Aufzeichnungen für ein bestimmtes

Merkmal zu speichern. Dann wird eine Festlegung gemacht, ob die Daten für das Merkmal gültig sind. Wenn die Daten ungültig sind, werden die zu überspringenden Daten an die Aufzeichnungen für das zu überspringende Merkmal aktualisiert und ein weiterer Versuch wird unternommen, die Datei zu speichern. Konsistenzüberprüfungen auf Dateiebenen (file-level consistency checks) können auch auf gleiche Art und Weise gemacht werden.

[0045] Wenn bei der Operation **314** festgelegt wird, dass die aktuelle Aufzeichnung nicht speicherbar ist, fährt die Routine **314** mit der Operation **322** fort, wo eine Festlegung getroffen wird, ob der aktuelle Modus der normale Modus ist. Wenn der aktuelle Modus der normale Modus ist, zweigt die Routine **300** zur Operation **324** ab, wo die Zählervariable für das Überspringen aktualisiert wird, welche anzeigt, dass ein Teil der Speicherstruktur **25** identifiziert wurde, der vielleicht übersprungen werden muss. Die Routine **300** fährt dann mit der Operation **324** fort, wo die Variable für den Modus zu „safe“ gesetzt wird. Auf diese Art und Weise wird der Speichermodus auf das Antreffen des nicht speicherbaren Teils der Speicherstruktur **25** hin von dem normalen Modus zu dem sicheren Modus umgeschaltet. Die Routine **300** kehrt dann zur Operation **310** zurück, wo die Verarbeitung des Dateiordners bzw. der Datendatei an den Anfang zurückkehrt.

[0046] Wenn bei der Operation **322** festgelegt wird, dass der aktuelle Speichermodus nicht der normale Modus ist, fährt die Routine **300** bei der Operation **328** fort, wo eine Festlegung gemacht wird, ob der aktuelle Speichermodus der sichere Modus ist. Weil nur der normale oder der sichere Speichermodus mögliche Werte in diesem Abschnitt der Routine **300** sein sollten, zweigt die Routine zur Operation **330** ab, wo ein Fehler zurückgegeben wird, wenn der aktuelle Speichermodus nicht der sichere Modus ist. Die Routine **300** fährt dann von der Operation **330** an zur Operation **320** fort, wo sie endet. Wenn allerdings bei der Operation **328** festgelegt wird, dass der aktuelle Modus der sichere Modus ist, fährt die Routine **300** zu der Operation **332** fort.

[0047] Bei der Operation **332** wird ein Versuch unternommen, die aktuelle Aufzeichnung zu reparieren. Bei der Operation **334** wird eine Festlegung gemacht, ob die aktuelle Aufzeichnung reparierbar war. Wenn die Aufzeichnung reparierbar war, zweigt die Routine **300** zur Operation **336** ab, wo die aktuelle Aufzeichnung gespeichert wird. Bei der Operation **336** wird die Zählervariable für das Überspringen auch aktualisiert, um anzuzeigen, dass das Speichern der aktuellen Aufzeichnung nicht übersprungen werden sollte, weil die Aufzeichnung reparierbar war. Von der Operation **336** zweigt die Routine **300** zurück zur Operation **316** ab, wo die Überbleibenden bzw. die Reste der Aufzeichnungen der Datenstruktur **25** in

der oben beschriebenen Art und Weise verarbeitet werden.

[0048] Wenn bei der Operation **334** festgelegt wird, dass die aktuelle Aufzeichnung nicht repariert werden konnte, zweigt die Routine **300** zur Operation **338** ab. Bei der Operation **338** wird eine Festlegung gemacht, ob die Anzahl der Aufzeichnungen, die zu überspringen sind, gleich Null ist. Dies würde der Fall sein, wo das Speichern im normalen Modus begonnen wurde und wo die erste beschädigte Aufzeichnung angetroffen wurde und die Aufzeichnung nicht reparierbar ist. In diesem Fall zweigt die Routine **300** zur Operation **340** ab, wo die Zählervariable für das Überspringen aktualisiert wird, um anzuzeigen, dass die Aufzeichnung übersprungen werden sollte. Die Routine **300** kehrt dann zur Operation **310** zurück, wo die Verarbeitung der Speicherstruktur **25** in der oben beschriebenen Art und Weise an den Anfang zurückkehrt.

[0049] Wenn bei der Operation **338** festgelegt bzw. bestimmt wird, dass die Anzahl der Variablen zu überspringender Aufzeichnungen nicht gleich Null ist, fährt die Routine **300** bei der Operation **342** fort, wo ein Versuch unternommen wird, das Speichern der aktuellen Aufzeichnung zu überspringen. Bei der Operation **344** wird eine Festlegung gemacht, ob das Speichern der aktuellen Aufzeichnung übersprungen werden sollte. Wenn das Speichern der aktuellen Aufzeichnung übersprungen werden kann, zweigt die Routine **300** zur Operation **346** ab, wo die Aufzeichnung in der Variablen für die zu überspringenden Aufzeichnungen gekennzeichnet wird. Die Routine fährt dann zur Operation **316** fort, wie oben beschrieben ist.

[0050] Wenn bei der Operation **344** bestimmt wird, dass die aktuelle Aufzeichnung nicht übersprungen werden kann, fährt die Routine **300** zur Operation **306** fort, wo die Variable für den Modus zu „minimal“ gesetzt wird. Die Routine **300** fährt dann zur Operation **348** fort, wo ein Versuch unternommen wird, die Speicherstrukturen **25** im minimalen Modus zu speichern. Wie oben beschrieben wurde, werden im minimalen Modus nur Benutzerdaten gespeichert. Darüber hinaus wird ein Versuch unternommen, so viel wie möglich der Benutzerdaten zu speichern, wenn die Benutzerdaten auch beschädigt sind. Die Routine **300** fährt dann mit der Operation **320** fort, wo sie endet.

[0051] Basierend auf dem Vorangegangenen sollte gewürdigt werden, dass die verschiedenen Ausführungsformen der Erfindung ein Verfahren, ein System, eine Vorrichtung und computerlesbare Medien zum Speichern der Inhalte eines Dokumentes umfassen, das in einer Struktur in einem flüchtigen Speicher in eine Datei bzw. einen Daten- oder Dateionder gespeichert wird, der auf einem Massenspeicher-

gerät gespeichert ist. Die obigen Spezifikationen, Beispiele und Daten stellen eine komplette Beschreibung der Herstellung und Verwendung der Zusammensetzung der Erfindung dar.

Patentansprüche

1. Verfahren zum Sichern des Inhalts einer Speicherstruktur (**25**), die in einem flüchtigen Speicher gespeichert ist und einen oder mehrere Abschnitte (**26A**, **26B**, **26C**, **26D**, ..., **26N**) hat, in eine Datendatei in einer Massenspeichervorrichtung (**14**), wobei das Verfahren umfasst:

Unternehmen des Versuchs, jeden Abschnitt der Speicherstruktur in einem ersten Sicherungsmodus zu sichern, wobei in dem ersten Modus minimale Integritätsprüfung an jedes der Abschnitte durchgeführt wird, um festzustellen, ob ein Abschnitt der Datendatei beschädigt ist; und

in Reaktion darauf, dass festgestellt wird, dass ein Abschnitt beschädigt ist, Unternehmen des Versuchs, den Inhalt der Speicherstruktur in einem zweiten Sicherungsmodus zu sichern, wobei umfangreichere Integritätsprüfung jedes der Abschnitte durchgeführt wird, um festzustellen, ob ein beschädigter Abschnitt repariert werden kann, und

in Reaktion darauf, dass festgestellt wird, dass ein beschädigter Abschnitt nicht repariert werden kann, Auslassen der Sicherung des beschädigten Abschnitts.

2. Verfahren nach Anspruch 1, wobei, wenn versucht wird, den Inhalt der Speicherstruktur in dem zweiten Sicherungsmodus zu sichern, eine Feststellung dahingehend getroffen wird, ob ein beschädigter Abschnitt repariert werden kann, und in Reaktion darauf, dass festgestellt wird, dass ein beschädigter Abschnitt repariert werden kann, Reparieren des beschädigten Abschnitts und Sichern des reparierten Abschnitts in die Datendatei.

3. Verfahren nach Anspruch 2, das des Weiteren umfasst:

Feststellen, ob das Sichern eines beschädigten Abschnitts, der nicht zu reparieren ist, ausgelassen werden kann; und

in Reaktion darauf, dass festgestellt wird, dass das Sichern eines beschädigten Abschnitts, der nicht zu reparieren ist, nicht ausgelassen werden kann, Unternehmen eines Versuchs, den Inhalt der Speicherstruktur in einem dritten Sicherungsmodus zu sichern, wobei in dem dritten Sicherungsmodus nur Abschnitte der Speicherstruktur, die Benutzerdaten entsprechen, in die Datendatei gesichert werden.

4. Verfahren nach Anspruch 3, wobei die Benutzerdaten von einem Benutzer eingegebene Textdaten umfassen.

5. Verfahren nach Anspruch 3, wobei die Benut-

zerdaten von einem Benutzer eingegebene numerische Daten umfassen.

6. Computerlesbares Medium, auf dem durch Computer ausführbare Befehle gespeichert sind, die, wenn sie von einem Computer ausgeführt werden, den Computer veranlassen:

einen ersten Sicherungsmodus zum Sichern des Inhalts einer Speicherstruktur bereitzustellen, die einen oder mehrere Abschnitte (**26A**, **26B**, **26C**, ..., **26N**) hat, von einem flüchtigen Speicher in eine Datendatei in einer Massenspeichervorrichtung (**14**), wobei minimale Integritätsprüfung jedes der Abschnitte durchgeführt wird, um festzustellen, ob ein Abschnitt der Datendatei beschädigt ist, wenn in dem ersten Sicherungsmodus gesichert wird;

einen zweiten Sicherungsmodus zum Sichern des Inhalts der Speicherstruktur in die Datendatei bereitzustellen, wobei umfangreichere Integritätsprüfung jedes der Abschnitte durchgeführt wird, um festzustellen, ob ein beschädigter Abschnitt repariert werden kann, wobei in dem zweiten Sicherungsmodus das Sichern des beschädigten Abschnitts in Reaktion darauf ausgelassen wird, dass festgestellt wird, dass der beschädigte Abschnitt nicht repariert werden kann,

mit dem Sichern des Inhalts der Speicherstruktur in dem ersten Sicherungsmodus zu beginnen; und in Reaktion darauf, dass festgestellt wird, dass ein Abschnitt beschädigt ist, zu dem zweiten Sicherungsmodus zu wechseln.

7. Computerlesbares Medium nach Anspruch 6, wobei in dem zweiten Sicherungsmodus eine Feststellung dahingehend getroffen wird, ob ein beschädigter Abschnitt repariert werden kann, und in Reaktion darauf, dass festgestellt wird, dass ein beschädigter Abschnitt repariert werden kann, der beschädigte Abschnitt repariert wird und der reparierte Abschnitt in die Datendatei gesichert wird.

8. Computerlesbares Medium nach Anspruch 7, das des Weiteren durch Computer lesbare Befehle umfasst, die, wenn sie von dem Computer ausgeführt werden, den Computer veranlassen:

einen dritten Sicherungsmodus bereitzustellen, wobei in dem dritten Modus nur Abschnitte der Speicherstruktur, die Benutzerdaten entsprechen, in die Datendatei gesichert werden;

während er in dem zweiten Sicherungsmodus arbeitet, festzustellen, ob das Sichern eines beschädigten Abschnitts, der nicht zu reparieren ist, ausgelassen werden kann; und

in Reaktion darauf, dass festgestellt wird, dass das Sichern eines beschädigten Abschnitts, der nicht zu reparieren ist, nicht ausgelassen werden kann, zu dem dritten Sicherungsmodus zu wechseln.

9. Computerlesbares Medium nach Anspruch 8, wobei die Benutzerdaten von einem Benutzer einge-

gebene Textdaten umfassen.

10. Computerlesbares Medium nach Anspruch 9, wobei die Benutzerdaten von einem Benutzer eingegebene numerische Daten umfassen.

11. Computerlesbares Medium nach Anspruch 6, wobei ein Abschnitt beschädigt sein kann, wenn der Abschnitt einen unerwarteten Datenwert enthält.

12. Computerlesbares Medium nach Anspruch 11, wobei ein Abschnitt des Weiteren beschädigt sein kann, wenn in dem Abschnitt Daten fehlen.

13. Computerlesbares Medium nach Anspruch 12, wobei ein Abschnitt des Weiteren beschädigt sein kann, wenn der Abschnitt ungültige XML (Extensible Markup Language)-Daten enthält.

14. Computerlesbares Medium nach Anspruch 13, wobei ein Abschnitt des Weiteren beschädigt sein kann, wenn Sichern des Abschnitts durch ein Anwendungsprogramm bewirkt, dass das Anwendungsprogramm abstürzt.

15. Computerlesbares Medium nach Anspruch 14, wobei ein Abschnitt des Weiteren beschädigt sein kann, wenn der Abschnitt fehlt.

Es folgen 4 Blatt Zeichnungen

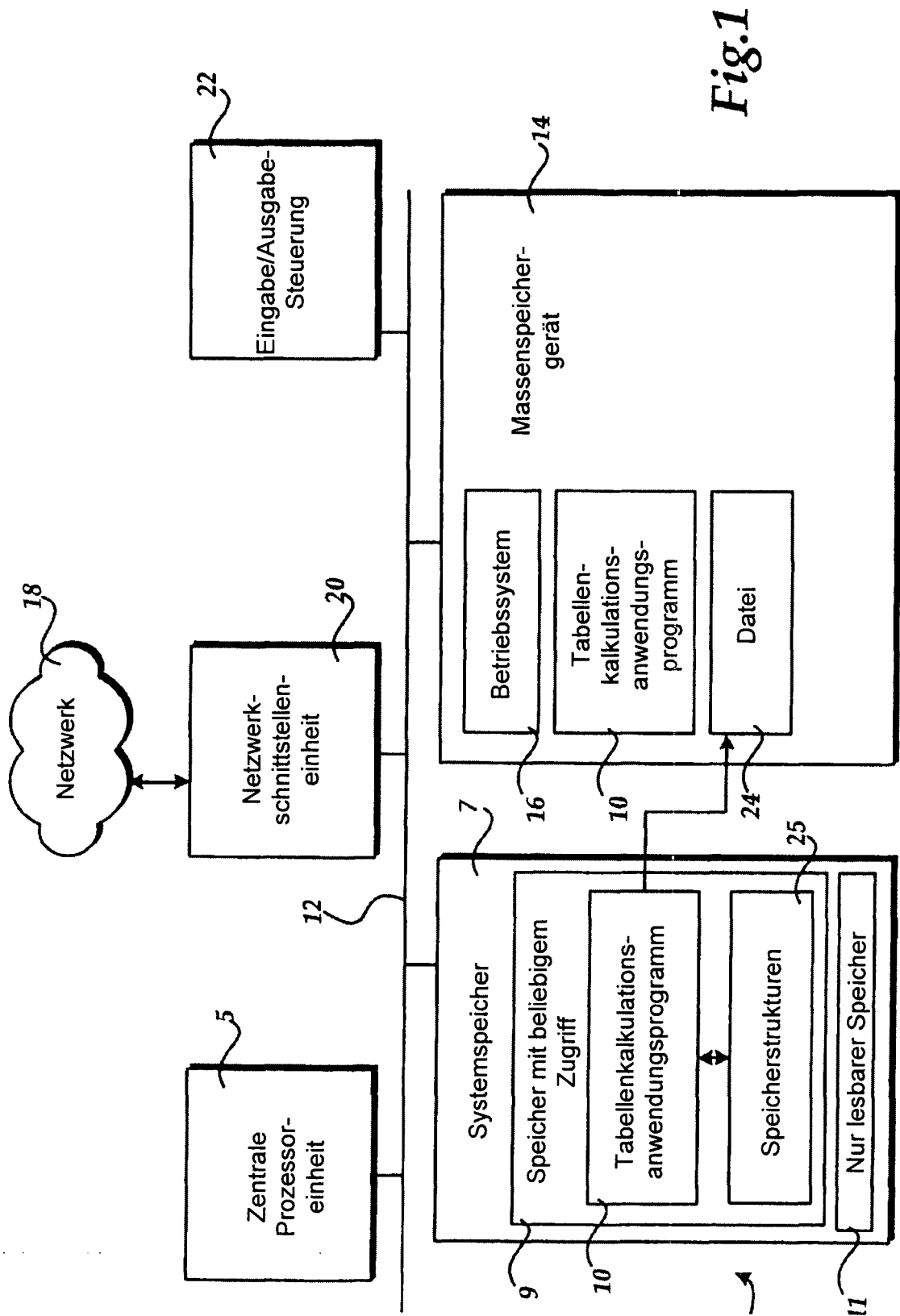


Fig.1

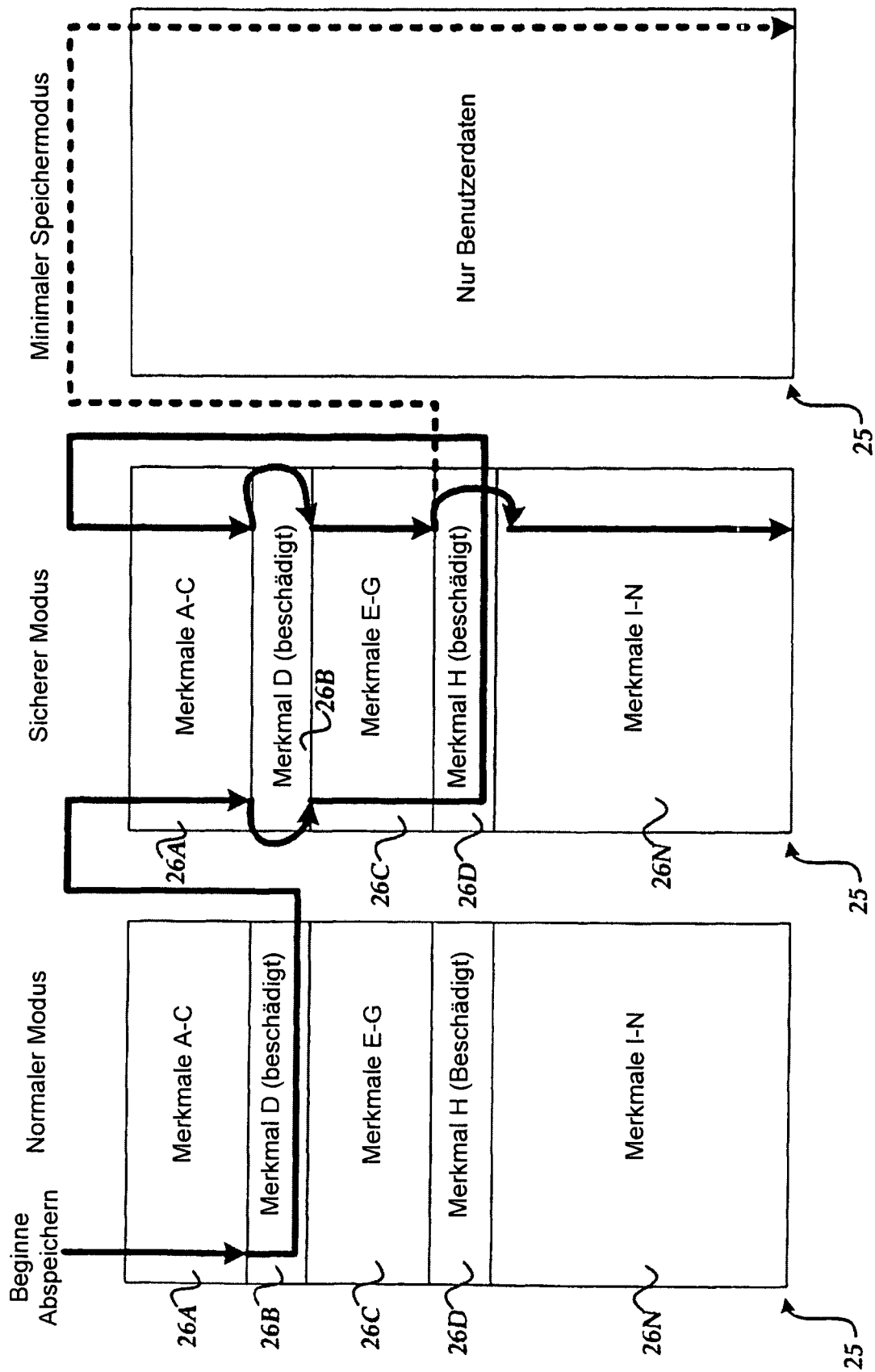
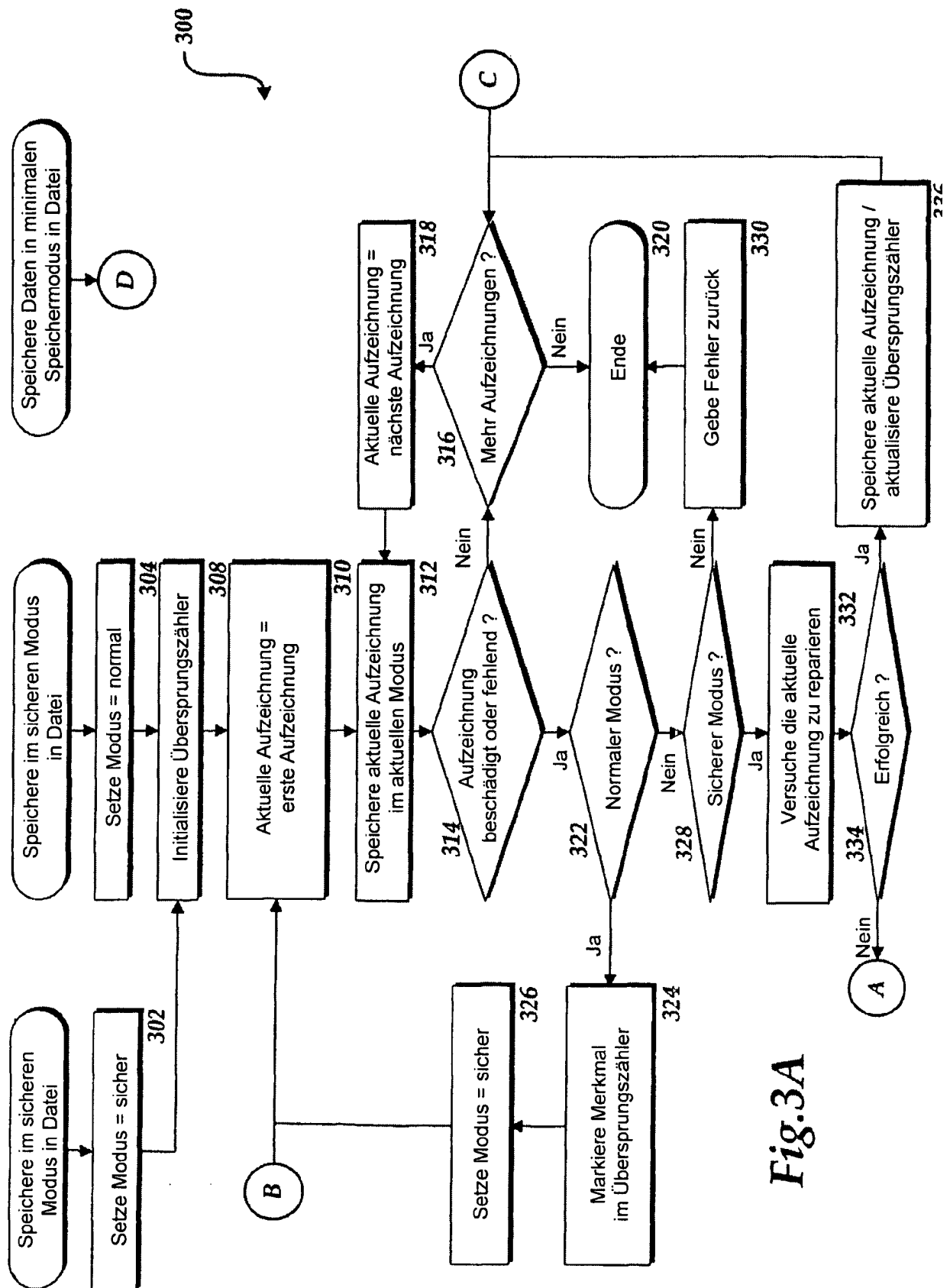


Fig.2



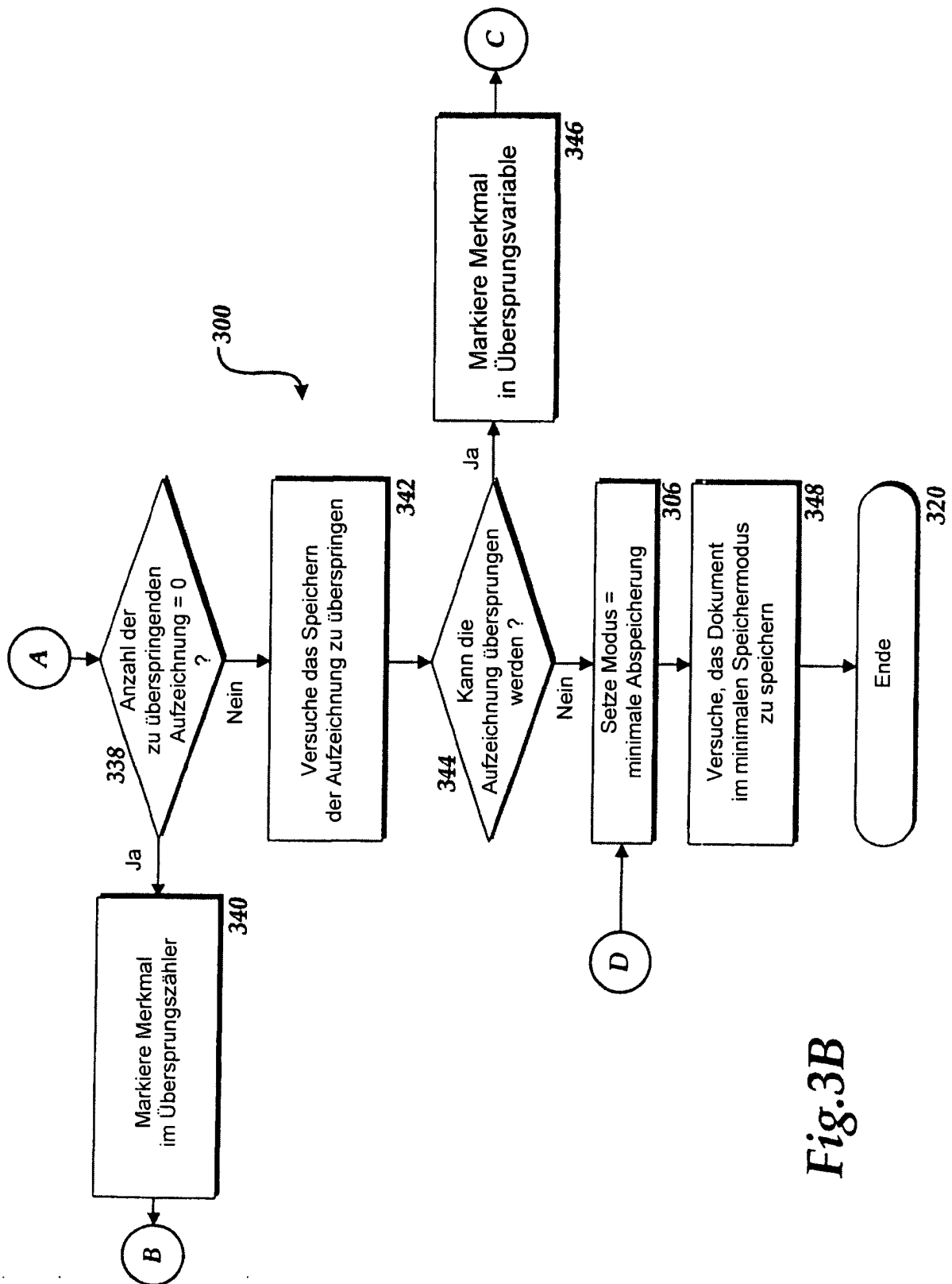


Fig.3B