



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 27 787 T2** 2006.07.13

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 051 681 B1**

(21) Deutsches Aktenzeichen: **699 27 787.6**

(86) PCT-Aktenzeichen: **PCT/US99/00279**

(96) Europäisches Aktenzeichen: **99 902 077.9**

(87) PCT-Veröffentlichungs-Nr.: **WO 1999/035593**

(86) PCT-Anmeldetag: **07.01.1999**

(87) Veröffentlichungstag
der PCT-Anmeldung: **15.07.1999**

(97) Erstveröffentlichung durch das EPA: **15.11.2000**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **19.10.2005**

(47) Veröffentlichungstag im Patentblatt: **13.07.2006**

(51) Int Cl.⁸: **G06F 17/22 (2006.01)**
G06F 17/21 (2006.01)

(30) Unionspriorität:

70720 P	07.01.1998	US
75123 P	13.02.1998	US
107941	30.06.1998	US

(73) Patentinhaber:

Microsoft Corp., Redmond, Wash., US

(74) Vertreter:

PAe Splanemann Reitzner Baronetzky
Westendorp, 80469 München

(84) Benannte Vertragsstaaten:

DE, FR, GB

(72) Erfinder:

WECKER, Dave, Bothell, US; TUNIMAN, David,
Redmond, US

(54) Bezeichnung: **KANALDEFINITIONSARCHITEKTURERWEITERUNG**

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**HINTERGRUND DER ERFINDUNG**

[0001] Die vorliegende Erfindung betrifft mobile persönliche Rechenvorrichtungen, welche im Allgemeinen als Mobilvorrichtungen bekannt sind. Insbesondere betrifft die vorliegende Erfindung ein System und ein Verfahren zum Liefern und Empfangen von Informationen auf einer Mobilvorrichtung.

[0002] Mobilvorrichtungen sind kleine elektronische Rechenvorrichtungen, welche auch oft als persönliche digitale Assistenten (PDAs) bezeichnet werden. Viele solcher Mobilvorrichtungen sind tragbare Vorrichtungen, oder handtellergröße Vorrichtungen, welche bequem in die Hand passen. Eine im Handel erhältliche Mobilvorrichtung wird unter der Handelsbezeichnung HandHeld PC (oder H/PC) verkauft und verfügt über Software, die von der Microsoft Corporation in Redmont, Washington geliefert wird.

[0003] Im Allgemeinen weist die Mobilvorrichtung einen Prozessor, einen Direktzugriffsspeicher (RAM) sowie eine Eingabevorrichtung, wie beispielsweise eine Tastatur, und eine Anzeige auf. Die Tastatur kann in die Anzeige integriert sein, wie es beispielsweise der Fall ist, wenn die Tastatur als eine berührungsempfindliche Anzeige integriert ist. Eine Kommunikationsschnittstelle ist optional bereitgestellt und wird üblicherweise zur Kommunikation mit einem Tischcomputer verwendet. Eine austauschbare oder wiederaufladbare Batterie versorgt die Mobilvorrichtung mit Energie. Optional kann die Mobilvorrichtung Energie von einer externen Energiequelle erhalten, welche Vorrang vor der eingebauten Batterie hat oder oder diese wiederauflädt.

[0004] In einigen bisherigen Anwendungen wird die Mobilvorrichtung in Verbindung mit einem Tischcomputer verwendet. Beispielsweise kann der Benutzer der Mobilvorrichtung auch auf einen Tischcomputer Zugriff haben und diesen in der Arbeit oder zu Hause, oder beidem, verwenden. Der Benutzer arbeitet typischerweise auf dem Tischcomputer und der Mobilvorrichtung mit denselben Anwendungen. Somit ist es recht vorteilhaft, wenn die Mobilvorrichtung so entworfen wurde, dass sie mit dem Tischcomputer gekoppelt werden kann, um Informationen mit dem Tischcomputer auszutauschen und zu teilen.

[0005] Eine weitere Technik zur Lieferung von Informationen an derartige Mobilvorrichtungen ist durch eine drahtlose Übertragungsverbindung. Derartige Informationen können elektronische Post (E-Mail) oder Nachrichten-, Wetter-, Sport-, Verkehrsinformationen sowie Informationen über örtliche Ereignisse beinhalten. Die Information wird typischerweise von einem mit dem Internet verbundenen Tischcomputer erhalten und über eine Drahtverbindung geliefert. Es kann jedoch erwünscht sein, derartige Informationen auch über eine drahtlose Verbindung zu liefern. Ein drahtloser Empfänger an der Mobilvorrichtung kann agieren, um Informationen zu empfangen, wenn sie zu der Mobilvorrichtung gesendet werden.

[0006] Es existiert gegenwärtig kein vernünftiger Weg, Pusch-Style-Inhalte (wie zum Beispiel Hypertext-Mark-up-Language-Inhalte (HTML-Inhalte)), welche auf einem globalen Netz wie dem Internet und dem World Wide Web bereitgestellt sind, auf drahtlose Weise und in einer offenen und verfügbaren Architektur zu derartigen Vorrichtungen zu übertragen. Die Bitrate herkömmlicher drahtloser Kanäle ist sehr niedrig. Somit ist die Lieferung eines sehr großen Inhalts (wie beispielsweise HDML-Inhalt) höchst unpraktisch.

[0007] Eine herkömmliche Herangehensweise an die Lieferung derartiger Informationen besteht darin, den Inhalt in einem vorrichtungsfreundlichen Format neu zu verfassen, wie beispielsweise HTML. Der Inhalt wird dann über ein Push-Style-Modell erhalten. Eine weitere, gegenwärtig zur Lieferung von Informationen über ein drahtloses Medium in einem geschlossenen Modell zu liefern. In einem geschlossenen Modell kann ein Inhaltslieferer nur Inhalte liefern, welche in einem Format verfasst sind, das für den Empfang durch eine bestimmte Vorrichtung geeignet ist, welche eine bestimmte Software-Art implementiert. Dies bedeutet, dass die große Mehrheit von Web-Inhalten für die Ansicht auf derartigen Vorrichtungen nicht verfügbar ist.

[0008] CASTEDO ELLERMAN: "Channel Definition Format (CDF)" CHANNEL DEFINITION FORMAT SUBMISSION 970309, 10. März 1997 (1997-03-10), XP002103294 <http://www.w3.org/TR/NOTE-CDFsubmit.html> offenbart eine Beschreibung eines Kanal-Definition-Formats, welche es einem Web Publisher erlaubt, häufig aktualisierte Informationssammlungen anzubieten.

ZUSAMMENFASSUNG DER ERFINDUNG

[0009] Es wird ein maschinenlesbarer Datenträger wie in Anspruch 1 ausgeführt sowie ein Verfahren zur Aufbereitung von Informationen auf einem Computer wie in Anspruch 10 ausgeführt bereitgestellt.

[0010] Ein Verfahren zur Aufbereitung von Informationen, wie sie beispielsweise durch das Internet auf einem Computer verfügbar sind, beinhaltet das Speichern einer Inhaltsstrukturdatei, einer Datendatei und einer Skriptdatei. Die Datendatei weist auf die Informationen hinweisende Daten auf, und die Skriptdatei weist Skriptinformationen auf, die auf eine gewünschte Form hinweisen, in der die Daten aufbereitet werden sollen. Die Inhaltsstrukturdatei, die Datendatei und die Skriptdatei sind unabhängig durch den Computer empfangbar. Die Inhaltsstrukturdatei wird gelesen, um sicherzustellen, welches Skript in der Skriptdatei den aufzubereitenden Daten zugehörig ist. Die Daten aus der Datendatei werden abgerufen, und die zugehörige Skriptdatei wird ausgeführt, um die Daten aufzubereiten. Befehle können auf einem maschinenlesbaren Datenträger geliefert werden, um das Verfahren zu realisieren.

[0011] Bevorzugt schließt die Inhaltsstrukturdatei Verweise auf Daten und Skripten in einer Hierarchie ein. Insbesondere kann die Inhaltsstrukturdatei Skriptkennzeichen einschließen, welche den Skripten in der Skriptdatei zugehörig sind, wobei die Skriptkennzeichen in der Hierarchie angeordnet sind. Wenn die Inhaltsstrukturdatei gelesen wird, um sicherzustellen, welches Skript in der Skriptdatei den aufzubereitenden Daten zugehörig ist, kann das Skript als eine Funktion der Hierarchie gewählt werden. Die Organisation der Skriptkennzeichen in einer Hierarchie erlaubt eine Kennzeichenvererbung. Insbesondere wenn kein zugehöriges Skriptkennzeichen für die Daten im unteren Abschnitt der Hierarchie vorliegt, kann ein Skript, auf welches durch ein Skriptkennzeichen in einem höheren Abschnitt der Hierarchie verwiesen wird, ausgeführt werden, um die Daten aufzubereiten. Bevorzugt wird, wenn ein Skript, auf welches in der Inhaltsstrukturdatei verwiesen wird, in der Skriptdatei nicht gefunden wird, ein Standardskript ausgeführt, um die Daten aufzubereiten.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0012] [Fig. 1](#) ist ein vereinfachtes Blockdiagramm, welches eine Mobilvorrichtung in einem erfindungsgemäßen System darstellt;

[0013] [Fig. 2](#) ist ein ausführlicheres Blockdiagramm einer in [Fig. 1](#) gezeigten Mobilvorrichtung;

[0014] [Fig. 3](#) ist eine vereinfachte piktographische Darstellung der in [Fig. 2](#) gezeigten Mobilvorrichtung;

[0015] [Fig. 4](#) ist eine vereinfachte piktographische Darstellung der in [Fig. 2](#) gezeigten Mobilvorrichtung;

[0016] [Fig. 5](#) ist ein Blockdiagramm eines Tischcomputers in Übereinstimmung mit einem Aspekt der vorliegenden Erfindung;

[0017] [Fig. 6](#) ist ein Ablaufdiagramm, welches den Betrieb einer Mobilvorrichtung in Übereinstimmung mit einem Aspekt der vorliegenden Erfindung darstellt;

[0018] [Fig. 7](#) ist ein vereinfachtes Blockdiagramm von [Fig. 6](#);

[0019] [Fig. 8](#) ist eine diagrammatische Darstellung einer graphischen Anwenderschnittstelle, welche durch die vorliegende Erfindung aufbereitet wird.

AUSFÜHRLICHE BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

[0020] [Fig. 1](#) stellt ein System **10** dar, in welchem die vorliegende Erfindung beispielhaft realisiert ist. Das System **10** weist den Inhaltslieferanten **12** auf, weiter den drahtlosen Träger **14**, den Tischcomputer **16** und die Mobilvorrichtung **18**. Der Inhaltslieferant **12** liefert jede geeignete Art von Daten von einer Datenbank oder anderen Datenquelle. Beispielsweise wird der Inhaltslieferant **12** im Folgenden als ein Lieferant von Inhalten aus dem Internet, dem World-wide-Web, beschrieben. In einem Beispiel kann der Inhalt in einem Standardformat, wie beispielsweise HTML, JPEG, GIF, wav, etc. geliefert werden.

[0021] Der Webinhalt ist bevorzugt auch in einer Inhaltsstrukturdatei beschrieben, auch allgemein als Kanaldefinitionsformat- oder Channel-Definition-Format(CDF)-Datei bekannt. Ein einzelner Inhaltsabschnitt (wie beispielsweise eine Webseite oder eine Website) wird hierin als ein Mobilkanal bezeichnet.

[0022] Ein Mobilkanal ist eine selbstbeschreibende Website, welche alle Informationen enthält, die für einen effizienten Download von Webinhalten auf die Mobilvorrichtung **18** nötig sind. Drei Komponenten sind in einem bevorzugten Mobilkanal bereitgestellt. Die Komponenten beinhalten eine CDF-Datei, einen Satz Skriptdateien zur Aufbereitung des Kanals und einen Satz aufzubereitender Datendateien. Die CDF-Dateien sind nachste-

hend ausführlicher beschrieben. Kurz gesagt ist das CDF ein Inventar an auf dem Mobilkanal enthaltenen Inhalten.

[0023] Die Skriptdateien weisen Skript auf, welches Schablonen definiert, die das Erscheinen der Daten auf dem Bildschirm der Mobilvorrichtung **18** spezifizieren. Skripten sind bevorzugt in einem Visuellen Basisskript oder Visual Basic Script (VBS) verfasst.

[0024] Die Datendateien entsprechen einer oder mehreren Skriptdateien und beinhalten Daten, welche den Direktinhalt des aufzubereitenden Kanals anzeigen. Die Daten sind in kleine und einfache Textdateien verpackt. Alle diese Informationen werden zur Definition von Webinhalten verwendet.

[0025] Der Betrieb des Systems **10** unter Verwendung separater Skript- und Datendateien ist ausführlich im US-Patent Nr. 6,311,058 beschrieben, welches am 30. Juni 1998 eingereicht wurde und den Titel "SYSTEM FOR DELIVERING DATA CONTENT OVER A LOW BIT RATE TRANSMISSION CHANNEL" trägt. Kurz gesagt ist jedoch der drahtlose Träger **14** so konfiguriert, dass er Webinhalte von dem Webinhaltslieferanten **12** über eine Wählverbindung oder eine direkte Internetverbindung oder aber eine Netzwerkverbindung empfängt. Der drahtlose Träger **14** weist auch einen drahtlosen Schiebeserver **20** auf. Der Server **20** splittet vom Inhaltslieferanten **12** empfangene Inhalte in Stücke, welche mit der speziellen Transportart kompatibel sind, die durch den drahtlosen Träger **14** verwendet wird. Beispielsweise kann der Server **20** die Daten so splitten, dass sie maximalen Paketgrößenbeschränkungen, Zeichensatzanforderungen etc. für die verwendete Kanalart oder Transportart entsprechen. vor der Übertragung werden die Daten bevorzugt in eine andere Form übersetzt. Die Übersetzung kann die Kompression, die Verschlüsselung, die Codierung und dann die Verpackung beinhalten.

[0026] Sind die Daten angemessen gesplittet, so dass sie den Transportbeschränkungen entsprechen, werden die Daten dann für die Übertragung über die Luft durch ein drahtloses Netzwerk (wie beispielsweise durch einen Paging-Kanal) konfiguriert, um direkt auf der Mobilvorrichtung **18** empfangen zu werden. Die übertragenen Daten werden durch einen drahtlosen Empfänger und die Treiberkomponente **22** auf der Mobilvorrichtung **18** empfangen, wo die Daten für die Verwendung durch die Mobilvorrichtung **18** vorbereitet werden.

[0027] Die Mobilvorrichtung **18** kann auch ein Modem **24** aufweisen. Somit können Webinhalte, anstatt durch den drahtlosen Träger **14** übertragen zu werden, direkt vom Webinhaltslieferanten **12** durch eine direkte einwählbare Modemverbindung zur Mobilvorrichtung **18** übertragen werden.

[0028] Der Tischcomputer **16** ist ebenfalls nachstehend in der Beschreibung ausführlicher beschrieben. Kurz gesagt ist jedoch der Tischcomputer **16** bevorzugt mit einem Standard-Webbrowser, wie beispielsweise dem Internet Explorer 4.0 der Microsoft Corporation in Redmont, Washington, welcher im Handel erhältlich ist, ausgestattet. Ist dies der Fall, so können die Benutzer des Tischcomputers **16** bevorzugt Kanäle auf standardmäßige Weise abonnieren, welche dem Benutzer bestimmte Kanalinhalt liefern, die Off-Line oder Online geprüft werden können. Der Tischcomputer **16** verfügt bevorzugt über einen ladbaren Transport, welcher auf die Skriptdateien zugreift und die entsprechende Datendatei (in Übereinstimmung mit dem Skript) beeinflusst, um die Inhalte aufzubereiten, wo der Tischcomputer **16** die Daten aufbereitet.

[0029] Der Tischcomputer **16** kann durch den Transport periodisch neue und aktualisierte Skript-, Daten- und CDF-Dateien entweder zur weiteren Übertragung zur Mobilvorrichtung **18** oder einfach zur Aufbereitung der Daten abrufen oder empfangen. Die Skript-, Daten- und CDF-Dateien können entweder zusammen oder unabhängig voneinander übertragen werden. Da Skriptingdateien typischerweise weit weniger häufig aktualisiert werden müssen als die Datendateien, liefert dies dem Benutzer die Möglichkeit, die Webinhalte auf dem Tischcomputer (Off-Line) zu betrachten, während nur geringe Mengen an Bandbreite für die inkrementale Aktualisierung der Datendateien nötig sind.

[0030] Der Tischcomputer **16** weist ebenfalls bevorzugt die Synchronisationskomponente **26** auf. Kurz gesagt ist die Synchronisationskomponente **26** so konfiguriert, dass sie mit einer ähnlichen Synchronisationskomponente **28** auf der Mobilvorrichtung **18** interagiert, so dass Dateien, welche das Subjekt der Synchronisation sind, vom Tischcomputer **16** zur Mobilvorrichtung **18**, oder umgekehrt, synchronisiert werden können. Sind sie synchronisiert, so enthalten beide Dateien (diejenigen auf dem Computer **16** und auf der Mobilvorrichtung **18**) aktuelle Informationen.

[0031] Insbesondere kann die Mobilvorrichtung **18** entweder mit dem Tischcomputer **16** oder mit einer anderen Mobilvorrichtung **18**, oder aber mit beiden, synchronisiert sein. In diesem Fall sind Eigenschaften von Ob-

jekten, welche in einem Objektspeicher auf der Mobilvorrichtung **18** gespeichert sind, ähnlich wie Eigenschaften anderer Fälle des selben Objekts, welches in einem Objektspeicher auf dem Tischcomputer **16** oder einer weiteren Mobilvorrichtung **18** gespeichert sind. Somit wird beispielsweise, wenn ein Benutzer einen Fall eines in einem Objektspeicher auf dem Tischcomputer **16** gespeicherten Objektes ändert, der zweite Fall dieses Objektes in dem Objektspeicher der Mobilvorrichtung **18** aktualisiert, wenn die Mobilvorrichtung **18** das nächste Mal mit dem Tischcomputer **16** verbunden ist, so dass beide Fälle des gleichen Objekts aktuelle Daten enthalten. Dies wird als Synchronisation bezeichnet.

[0032] Um Synchronisation zu erzielen, laufen die Synchronisationskomponenten **26** und **28** sowohl auf der Mobilvorrichtung **18** als auch dem Tischcomputer **16** (oder einer weiteren Mobilvorrichtung **18**). Die Synchronisationskomponenten kommunizieren miteinander durch gut definierte Schnittstellen, um die Kommunikation und Synchronisation zu verwalten.

[0033] Die Mobilvorrichtung **18** ist auch bevorzugt mit einem Skriptendolmetscher ausgestattet, welcher der gleiche oder ein ähnlicher ist wie der ladbare Transport auf dem Tischcomputer **16**. Ein derartiger Transport kann beispielsweise ein verkleinerter visueller Basisdolmetscher sein, welcher das Formatierungsskript empfängt und interpretiert. Das Skript ist einer bestimmten Datendatei zugehörig (typischerweise einer Textdatei), welche die Rohdaten für den Webinhalt enthält. Somit operiert der Skriptendolmetscher an den Daten, welche einem vorgegebenen Skript zugehörig sind, um eine Aufbereitung des Webinhaltes für den Benutzer der Mobilvorrichtung **18** zu liefern.

[0034] Durch Trennung der Skripten von den Daten im Webinhalt kann Webinhalt an die Mobilvorrichtung **18** über Kanäle mit sehr geringen Bitraten übertragen werden. Das Skript muss typischerweise nur sehr selten übertragen werden. Auch können, da eine einzelne Datei typischerweise viel kleiner ist als die Skriptdateien, die Daten relativ häufig aktualisiert werden, wodurch dem Benutzer der Mobilvorrichtung **18** aktualisierte Webinhaltsinformationen geliefert werden, ohne neue Skripten zu übertragen. Somit erlaubt die Trennung der Skripten und Daten die Übertragung von Webinhaltsinformationen auf sehr effiziente Weise über Kanäle mit geringer Bitrate.

[0035] Es sollte sich verstehen, dass, während die Mobilvorrichtung **18** mit dem Tischcomputer **16** gekoppelt werden kann, sie auch mit einer weiteren Mobilvorrichtung **18** gekoppelt werden kann. Diese Verbindung kann unter Verwendung einer geeigneten und im Handel erhältlichen Kommunikationsverbindung und unter Verwendung eines geeigneten Kommunikationsprotokolls erfolgen. Beispielsweise kann die Mobilvorrichtung **18** entweder mit dem Tischcomputer **16** oder einer anderen Mobilvorrichtung **18** mit einem physischen Kabel kommunizieren, welches unter Verwendung eines seriellen Kommunikationsprotokolls kommuniziert. Andere Kommunikationsmechanismen werden durch die vorliegende Erfindung ebenfalls unterstützt, so beispielsweise Infrarotkommunikation oder andere geeignete Kommunikationsmechanismen.

[0036] [Fig. 2](#) ist ein detaillierteres Blockdiagramm der Mobilvorrichtung **18**. Die Mobilvorrichtung **18** weist bevorzugt den Mikroprozessor **30**, den Speicher **32**, die Eingabe-/Ausgabe(I/O)-Komponenten **34**, die Tischkommunikations-Schnittstelle **36**, den drahtlosen Empfänger **37** sowie die Antenne **39** auf. Bevorzugt sind diese Komponenten der Mobilvorrichtung **10** zur Kommunikation miteinander über einen geeigneten Bus **38** gekoppelt.

[0037] Der Speicher **32** ist bevorzugt als nichtflüchtiger elektronischer Speicher, wie beispielsweise als Direktzugriffsspeicher (RAM) mit einem Batterie-Backup-Modul (nicht gezeigt) realisiert, so dass im Speicher **32** gespeicherte Informationen nicht verloren gehen, wenn die allgemeine Energie für die Mobilvorrichtung **18** abgeschaltet wird. Ein Abschnitt des Speichers **32** ist bevorzugt als adressierbarer Speicher für die Programmausführung ausgewiesen, während ein weiterer Abschnitt des Speichers **32** bevorzugt zum Speichern verwendet wird, wie beispielsweise, um eine Speicherung auf einem Diskettenlaufwerk zu simulieren.

[0038] Der Speicher **32** weist das Betriebssystem **40**, ein Anwenderprogramm **42** (wie beispielsweise einen persönlichen Informationenmanager oder PIM) sowie einen Objektspeicher **44** auf. Während des Betriebes wird das Betriebssystem **40** bevorzugt durch den Prozessor **30** vom Speicher **32** ausgeführt. Das Betriebssystem **40** ist ein Betriebssystem der Marke Windows CE, welches im Handel von der Microsoft Corporation erhältlich ist. Das Betriebssystem **40** ist bevorzugt für Mobilvorrichtungen entworfen und realisiert Datenbankmerkmale, welche durch den PIM **42** durch einen Satz exponierter Anwenderprogrammier-Schnittstellen und -Verfahren verwendet werden können. Die Objekte im Objektspeicher **44** werden bevorzugt durch den PIM **42** und das Betriebssystem **40** gewartet, wenigstens teilweise auf Aufrufe an die exponierten Anwender-Programmier-Schnittstellen und -Verfahren hin.

[0039] Die I/O-Komponenten **34** können vorgesehen sein, um Eingabe- und Ausgabeoperationen von einem Benutzer der Mobilvorrichtung **18** zu vereinfachen. Die I/O-Komponenten **34** sind ausführlicher mit Bezug auf [Fig. 3](#) und [Fig. 4](#) beschrieben.

[0040] Die Tischkommunikations-Schnittstelle **36** ist optional als beliebige geeignete Kommunikationsschnittstelle vorgesehen. Die Schnittstelle **36** wird bevorzugt zur Kommunikation mit dem Tischcomputer **16**, dem Inhaltslieferanten **12**, dem drahtlosen Träger **14** und optional einer weiteren Mobilvorrichtung **18**, wie mit Bezug auf [Fig. 1](#) beschrieben, verwendet. Somit weist die Kommunikationsschnittstelle **36** bevorzugt Synchronisationskomponenten **28** zur Kommunikation mit dem Tischcomputer **16** sowie ein Modem **24** zur Kommunikation mit dem Inhaltslieferanten **12** auf. Der drahtlose Empfänger und Treiber **22** werden zur Kommunikation mit dem drahtlosen Träger **14** verwendet.

[0041] [Fig. 3](#) ist eine vereinfachte piktographische Darstellung einer Mobilvorrichtung **10**, welche in Übereinstimmung mit der vorliegenden Erfindung verwendet werden kann. Die Mobilvorrichtung **10**, wie in [Fig. 3](#) dargestellt, kann ein Desktop-Assistent sein, welcher unter der Bezeichnung H/PC vertrieben wird und über Software verfügt, die von der Microsoft Corporation geliefert wird. Die Mobilvorrichtung **18** weist eine Miniatur-Tastatur **43**, die Anzeige **45** und den Griffel **46** auf. In [Fig. 3](#) ist die Anzeige **45** eine Flüssigkristallanzeige (LCD), welche einen kontaktempfindlichen Anzeigebildschirm in Verbindung mit dem Griffel **46** verwendet. Der Griffel **46** wird verwendet, um an ausgewiesenen Koordinaten auf die Anzeige zu drücken oder diese zu berühren, um bestimmte Benutzer-Eingabefunktionen zu erzielen. Die Miniatur-Tastatur **43** ist bevorzugt als alphanumerische Miniatur-Tastatur mit jeglichen geeigneten und erwünschten Funktionstasten realisiert, welche ebenfalls zur Erzielung bestimmter Benutzer-Eingabefunktionen bereitgestellt sind.

[0042] [Fig. 4](#) ist eine weitere vereinfachte piktographische Darstellung der Mobilvorrichtung **18** in Übereinstimmung mit der vorliegenden Erfindung. Die Mobilvorrichtung **18**, wie in [Fig. 4](#) dargestellt, weist einige Teile auf, welche denjenigen ähneln, die mit Bezug auf [Fig. 3](#) beschrieben wurden, und welche ähnliche Bezugszeichen tragen. Beispielsweise weist die Mobilvorrichtung **18**, wie in [Fig. 4](#) gezeigt, ebenfalls den berührungsempfindlichen Bildschirm **45** auf, welcher in Verbindung mit dem Griffel **46** verwendet werden kann, um bestimmte Benutzer-Eingabefunktionen zu erzielen. Es sollte sich verstehen, dass die Anzeige **45** für die Mobilvorrichtung, wie in [Fig. 3](#) und [Fig. 4](#) gezeigt, die gleiche Größe aufweisen kann, oder dass sie unterschiedliche Größen aufweisen können, typischerweise jedoch viel kleiner wäre als eine herkömmliche, mit einem Tischcomputer verwendete Anzeige. Beispielsweise können in [Fig. 3](#) und [Fig. 4](#) gezeigte Anzeigen durch eine Matrix von nur 240 × 320 Koordinaten, oder 160 × 160 Koordinaten, oder jede andere passende Größe, definiert sein.

[0043] Die in [Fig. 4](#) gezeigte Mobilvorrichtung **18** weist auch eine Reihe von Benutzer-Eingabetasten oder -knöpfen (wie beispielsweise Rollknöpfe **47**) auf, welche es dem Benutzer erlauben, die Menüoptionen aufzurollen, oder aber andere Anzeigooptionen, welche auf der Anzeige **45** angezeigt werden, oder welche es dem Benutzer erlauben, Anwendungen zu ändern, ohne die Anzeige **45** zu berühren. Zusätzlich weist die auch in [Fig. 4](#) gezeigte Mobilvorrichtung **18** bevorzugt auch einen Energieknopf **49** auf, welcher verwendet werden kann, um die allgemeine Energie an die Mobilvorrichtung **18** an- und abzuschalten.

[0044] Es sollte sich auch verstehen, dass in [Fig. 4](#) die Mobilvorrichtung **18** einen Handschriftbereich **51** aufweist. Der Handschriftbereich **51** kann in Verbindung mit dem Griffel **46** so verwendet werden, dass der Benutzer Nachrichten schreiben kann, welche im Speicher **42** zur späteren Verwendung durch die Mobilvorrichtung **18** gespeichert werden. Die handschriftlichen Nachrichten werden einfach in handschriftlicher Form gespeichert und können durch den Benutzer abgerufen und auf dem Anzeigebildschirm **45** angezeigt werden, so dass der Benutzer die handschriftlichen Nachrichten, welche in die Mobilvorrichtung **18** eingegeben wurden, erneut betrachten kann. Die Mobilvorrichtung **18** verfügt über ein Zeichen-Erkennungsmodul, so dass der Benutzer alphanumerische Informationen durch Schreiben mit dem Griffel **46** auf dem Bereich **51** in die Mobilvorrichtung **18** eingeben kann. In diesem Fall erkennt das Zeichen-Erkennungsmodul in der Mobilvorrichtung **18** die alphanumerischen Zeichen und wandelt die Zeichen in computererkennbare alphanumerische Zeichen um, welche durch die Anwenderprogramme **42** in der Mobilvorrichtung **18** verwendet werden können.

[0045] [Fig. 5](#) und die zugehörige Erklärung sollen eine kurze, allgemeine Beschreibung eines geeigneten Tischcomputers **16** liefern, in welchem Teile der Erfindung realisiert werden können. Obgleich dies nicht erforderlich ist, wird die Erfindung, zumindest teilweise, im allgemeinen Kontext computerausführbarer Anweisungen, wie beispielsweise Programmmodule, beschrieben, welche durch einen Personalcomputer **16** oder die Mobilvorrichtung **18** ausgeführt werden. Im Allgemeinen weisen Programmmodule Routineprogramme, Objekte, Komponenten, Datenstrukturen, etc. auf, welche bestimmte Aufgaben durchführen oder bestimmte abstrak-

te Datentypen realisieren. Darüber hinaus wird es sich für Fachleute verstehen, dass der Tischcomputer **16** mit anderen Computer-Systemkonfigurationen realisiert sein kann, einschließlich Mehrprozessorsystemen, mikroprozessorbasierter oder programmierbarer Verbraucherelektronik, Netzwerk-PCs, Minicomputern, Zentralrechnern und ähnlichem. Die Erfindung kann auch in verteilten Rechenumgebungen ausgeführt werden, wo Aufgaben durch entfernte Verarbeitungsvorrichtungen durchgeführt werden, welche durch ein Kommunikationsnetzwerk verbunden sind. In einer verteilten Rechenumgebung können Programmmodule sowohl in lokalen als auch entfernten Speichervorrichtungen angeordnet sein.

[0046] Mit Bezug auf [Fig. 5](#) weist ein beispielhaftes System zur Realisierung des Tischcomputers **16** eine Rechenvorrichtung für allgemeine Zwecke in Form eines herkömmlichen Personalcomputers **16**, einschließlich der Verarbeitungseinheit **48**, eines Systemspeichers **50** sowie eines Systembusses **52** auf, welcher unterschiedliche Systemkomponenten einschließlich des Systemspeichers **50** mit der Verarbeitungseinheit **48** koppelt. Der Systembus **52** kann ein beliebiger mehrerer Arten von Busstrukturen sein, einschließlich eines Speicherbusses oder einer Speicherregelung, eines Peripheriebusses und eines lokalen Busses, welcher eine beliebige einer Vielzahl von Busarchitekturen verwendet. Der Systemspeicher **50** weist den Nur-Lese-Speicher (ROM) **54** sowie einen Direktzugriffsspeicher (RAM) **55** auf. Ein Basis-Eingabe/Ausgabe-System (Basic Input/Output System, BIOS) **56**, welches das Basisprogramm umfasst, das den Informationstransfer zwischen Elementen innerhalb des Tischcomputers **16** unterstützt, wie beispielsweise während des Hochfahrens, ist im ROM **54** gespeichert. Der Tischcomputer **16** weist weiter ein Festplattenlaufwerk **57** zum Lesen aus und Schreiben in eine Festplatte (nicht gezeigt) auf, weiter ein Magnetplattenlaufwerk **58** zum Lesen von oder Schreiben auf die entnehmbare Magnetplatte **59** und ein optisches Plattenlaufwerk **60** zum Lesen von oder Schreiben auf eine entnehmbare optische Platte **61**, wie beispielsweise eine CD-ROM oder andere optische Medien. Das Festplattenlaufwerk **57**, das Magnetplattenlaufwerk **58** und das optische Plattenlaufwerk **60** sind mit dem Systembus **52** durch eine Festplattenlaufwerk-Schnittstelle **62**, eine Magnetplattenlaufwerk-Schnittstelle **63** bzw. eine optische Laufwerk-Schnittstelle **64** verbunden. Die Laufwerke und zugehörigen computerlesbaren Medien liefern nichtflüchtige Speicherung computerlesbarer Anweisungen, Datenstrukturen, Programmmodule und anderen Daten für den Tischcomputer **16**.

[0047] Obgleich die hierin beschriebene beispielhafte Umgebung eine Festplatte, eine entnehmbare Magnetplatte **59** und eine entnehmbare optische Platte **61** verwendet, sollte sich für Fachleute verstehen, dass andere Arten computerlesbarer Medien, welche Daten speichern können, auf die durch einen Computer zugegriffen werden kann, wie beispielsweise Magnetkassetten, Flash-Memory-Karten, digitale Videoplatten (DVDs), Bernoullikassetten, Direktzugriffsspeicher (RAMs), Nur-Lese-Speicher (ROMs) und ähnliches ebenfalls in der beispielhaften Betriebsumgebung verwendet werden können.

[0048] Eine Reihe von Programmmodulen können auf der Festplatte, der Magnetplatte **59**, der optischen Platte **61**, dem ROM **54** oder dem RAM **55** gespeichert werden, einschließlich eines Betriebssystems **65**, eines oder mehrerer Anwenderprogramme **66** (welche PIMS umfassen können), anderer Programmmodule **67** (welche die Synchronisationskomponente **26** umfassen können) sowie der Programmdateien **68**. Ein Benutzer kann Befehle und Informationen in den Tischcomputer **16** durch Eingabevorrichtungen, wie beispielsweise eine Tastatur **70**, eine Zeigevorrichtung **72** und ein Mikrophon **74**, eingeben. Andere Eingabevorrichtungen (nicht gezeigt) können einen Joystick, ein Game-Pad, eine Satellitenschüssel, einen Scanner oder ähnliches umfassen. Diese und andere Eingabevorrichtungen sind oft mit der Verarbeitungseinheit **48** durch eine Seriellanschluss-Schnittstelle **76** verbunden, welche mit dem Systembus **52** gekoppelt ist, jedoch mit anderen Schnittstellen, wie beispielsweise einer Soundkarte, einem Parallelanschluss, einem Game-Port oder einem universellen seriellen Bus (USB) gekoppelt sein kann. Ein Monitor **77** oder eine andere Art von Anzeigevorrichtung ist ebenfalls mit dem Systembus **52** über eine Schnittstelle, wie beispielsweise einen Videoadapter **78**, verbunden. Zusätzlich zum Monitor **77** können Tischcomputer typischerweise andere periphere Ausgabevorrichtungen, wie beispielsweise den Lautsprecher **75** und Drucker, aufweisen.

[0049] Der Tischcomputer **16** kann in einer vernetzten Umgebung unter Verwendung logischer Verbindungen zu einem oder mehreren entfernten Computern (außer der Mobilvorrichtung **18**), wie beispielsweise einem entfernten Computer **79**, arbeiten. Der entfernte Computer **79** kann ein weiterer Personalcomputer, ein Server, ein Router, ein Netzwerk-PC, eine gleichwertige Vorrichtung oder ein anderer Netzwerkknoten sein, und weist typischerweise viele oder alle der vorstehend im Zusammenhang mit dem Tischcomputer **16** beschriebenen Elemente auf, obgleich nur eine Speichervorrichtung **80** in [Fig. 4](#) dargestellt ist. Die in [Fig. 4](#) dargestellten logischen Verbindungen umfassen ein lokales Netz (Local Area Network, LAN) **81** und ein globales Netz (Wide Area Network, WAN) **82**. Derartige Netzwerkumgebungen sind in Büros, unternehmensweiten Computernetzwerk-Intranetzen und dem Internet allgemein üblich.

[0050] Bei Verwendung in einer LAN-Netzwerkumgebung ist der Tischcomputer **16** mit dem lokalen Netz **81** durch eine Netzwerkschnittstelle oder einen Adapter **83** verbunden. Bei Verwendung in einer WAN-Netzwerkumgebung weist der Tischcomputer **16** typischerweise ein Modem **84** oder eine andere Vorrichtung zum Aufbau von Kommunikationen über das globale Netz **82**, wie beispielsweise das Internet, auf. Das Modem **84**, welches intern oder extern sein kann, ist mit dem Systembus **52** über die Seriellanschluss-Schnittstelle **76** verbunden. In einer Netzwerkumgebung können Programmmodule, welche in Verbindung mit dem Tischcomputer **16** dargestellt sind, oder Teile davon, in den entfernten Speichervorrichtungen gespeichert sein. Es versteht sich, dass die gezeigten Netzwerkverbindungen beispielhaft sind und andere Arten der Herstellung einer Kommunikationsverbindung zwischen den Computern verwendet werden können.

[0051] Der Tischcomputer **16** führt das Betriebssystem **65** aus, welches typischerweise im nichtflüchtigen Speicher **54** gespeichert ist und auf dem Prozessor **48** ausgeführt wird. Ein geeignetes Betriebssystem ist ein Betriebssystem der Marke Windows, welches durch die Microsoft Corporation vertrieben wird, wie beispielsweise Windows 95 oder Windows NT, Betriebssysteme oder abgeleitete Versionen von Betriebssystemen der Marke Windows oder ein anderes geeignetes Betriebssystem. Andere geeignete Betriebssysteme sind unter anderem Systeme wie beispielsweise Macintosh OS, welches durch die Apple Corporation vertrieben wird, und der OS/2-Präsentationsmanager, welcher durch International Business Machines (IBM) aus Armonk, New York, vertrieben wird. Anwenderprogramme sind bevorzugt im Programmmodul **67** in dem flüchtigen oder nichtflüchtigen Speicher gespeichert oder können von einer Diskette **59** oder einem CD-ROM-Laufwerk **61** in eine beliebige der in [Fig. 5](#) gezeigten Komponenten geladen werden, oder aus einem Netzwerk über den Netzwerkadapter **83** heruntergeladen werden, oder unter Verwendung eines anderen geeigneten Mechanismus geladen werden.

[0052] [Fig. 6](#) ist ein Blockdiagramm, welches den funktionellen Aufbau der Mobilvorrichtung **18** zeigt. [Fig. 6](#) zeigt ähnliche Teile wie die vorstehend in der Beschreibung aufgezeigten. Ähnliche Teile sind mit ähnlichen Bezugszeichen bezeichnet. [Fig. 6](#) stellt dar, dass die Mobilvorrichtung **18** webinhaltinformationen entweder über die Synchronisationskomponente **26**, den drahtlosen Empfänger (Funkempfänger und Treiber) **22** oder das Modem **24** empfängt. In jedem dieser Fälle werden schließlich CDF-Dateien **201** sowie Skriptenschablonen und Datendateien, welche durch die Blöcke **204** und **202** dargestellt sind, dem Zwischenspeicher **206** geliefert. Dort, wo die Webinhaltinformationen durch die Synchronisationskomponente **26** empfangen werden, können die CDF-Dateien **201**, die Skriptenschablonen **204** und die Datendateien **202** nicht auf die gleiche Weise verschlüsselt oder codiert oder anderweitig formatiert werden wie sie es für die Übertragung über einen drahtlosen Kanal oder Modemkanal werden. Daher werden die CDF-Dateien **201**, die Skriptenschablonen **204** und die Datendateien **202** direkt an die Zwischenspeicher-Verwaltung **208** geliefert. Die Zwischenspeicher-Verwaltung **208** empfängt die CDF-Dateien **201**, die Skriptenschablonen **204** und die Datendateien **202** und liefert sie an den Zwischenspeicher **206**. Die Zwischenspeicher-Verwaltung **208** weist Speicher manipulations- und Zeitgebungskomponenten sowie Datentransferkomponenten auf, welche geeignet sind, um die CDF-Dateien **201**, die Skriptenschablonen **204** und die Datendateien **202** zu einer bestimmten Position im Zwischenspeicher **206** zu transportieren und diese Position aufzuspüren.

[0053] Wenn andererseits die Webinhalte über einen drahtlosen Empfänger und Treiber **22** oder das Modem **24** empfangen werden, müssen zusätzliche Verarbeitungsschritte unternommen werden, ehe die Daten zwischengespeichert werden. Der drahtlose Empfänger und Treiber **22** ist eine physikalische Schicht, welche Nachrichten empfängt und filtert und Alarmereignisse an die Mobilvorrichtung **18** erzeugt. Die übertragenen Informationen werden vor der Übertragung zunächst übersetzt (beispielsweise komprimiert, verschlüsselt, codiert und verpackt). Somit müssen die Daten vor der weiteren Verwendung durch die Mobilvorrichtung **18** zurück in ihre ursprüngliche Form übersetzt werden. Daher werden die Daten zunächst an den Nachrichten-Router **210** geliefert. Der Nachrichten-Router **210** agiert, um die Nachricht aufzunehmen und die empfangene Nachricht an eine Übersetzungsschicht **209** zu leiten. In [Fig. 6](#) weist die Übersetzungsschicht **209** eine Auspack- und Fügekomponente **212**, eine Gruppe zusätzlicher Übersetzer, welche kollektiv mit **214** bezeichnet sind, sowie eine weitere Routing-Komponente **216** auf.

[0054] Der Auspack- und Fügeblock **212** agiert, um eine Gruppe von übertragenen Paketen zu empfangen, auszupacken und zu ordnen. Die Auspackvorrichtung fügt Pakete jeglicher langer Nachrichten, welche durch den drahtlosen Träger **14** aufgesplittet wurden, wieder zusammen. Die geordneten Daten werden zu den Übersetzungskomponenten **214** geliefert.

[0055] Die Übersetzungskomponenten **214** agieren, um die Daten neu zu formatieren oder zu übersetzen, so dass sie eine geeignete Form aufweisen, um durch den Inhaltshandhaber **216** gehandhabt zu werden. Beispielsweise können, wenn die Pakete, welche eine Nachricht enthalten, durch die Auspack- und Fügevorrich-

tung **212** ausgepackt und wieder zusammengefügt wurden, die Übersetzungskomponenten **214** diese Pakete typischerweise dekomprimieren, entschlüsseln und decodieren.

[0056] Der Inhaltshandhaber **216** liefert die ausgepackte, zusammengefügte und übersetzte Nachricht an das geeignete registrierte Ziel (d.h. die geeignete Anwendung oder einen anderen Funktionsblock) auf der Mobilvorrichtung **18**. In der in [Fig. 5](#) dargestellten Ausführungsform liefert der Inhaltshandhaber **216** die Informationen an die Zwischenspeicher-Verwaltung **208**, welche sie im Zwischenspeicher **206** speichert.

[0057] Wünscht der Benutzer, Off-Line die im Zwischenspeicher **206** gespeicherten Webinhalte durchzusehen, so startet der Benutzer ein geeignetes Anwenderprogramm, welches durch den Kanalbrowser-Block **218** in [Fig. 6](#) angezeigt ist. Der Kanalbrowser **218** erzeugt bevorzugt passende Anwenderschnittstellen auf der Anzeige **45**, welche dem Benutzer die Möglichkeit liefern, einen bestimmten zu betrachtenden Kanal zu wählen.

[0058] Der Kanalbrowser **218** ist so konfiguriert, dass er mit einem ladbaren Transport **220** interagiert, welcher wiederum mit der Zwischenspeicher-Verwaltung **208** gekoppelt ist. Als Antwort auf die Anfrage des Benutzers nach Betrachtung der über den gewählten Kanal gelieferten Informationen fordert der ladbare Transport **220** die Zwischenspeicher-Verwaltung **208** auf, die entsprechenden Webinhaltsinformationen (in Form von Skriptenschablonen und Datendateien) aus dem Zwischenspeicher **206** abzurufen. Die gewünschten Skriptenschablonen **204** und Datendateien **202** werden von dem Zwischenspeicher **208** zu dem ladbaren Transport **220** geliefert.

[0059] Der Skriptendolmetscher im Transport **220** ist bevorzugt ein visueller Basisskriptendolmetscher, welcher die Skriptenschablonen **204** dolmetscht und auf die Datendateien **202** als eine Funktion auch der CDF-Datei **201** wirkt, um eine erwünschte Aufbereitung der Webinhalte zu liefern. In [Fig. 6](#) werden die Webinhalte als eine Seite **224** mit herkömmlicher Hypertext-Mark-Up(HTML)-Sprache aufbereitet. Der ladbare Transport **220** liefert dann die HTML-Seiten-Aufbereitung an den Kanalbrowser **218** zur Betrachtung durch den Benutzer der Mobilvorrichtung **18** auf der Anzeige **45**.

[0060] Das System **10** ermöglicht eine Protokollierung gewünschter Informationen zur Verwendung durch den Inhaltslieferanten **12**. Anders gesagt, können die Inhaltslieferanten durch Liefern eines Eintrags in die CDF-Datei **201** bestimmte Elemente kennzeichnen, welche sie aufspüren möchten (d.h. sie können bestimmte Elemente kennzeichnen, für welche sie wissen möchten, wann und wie lange diese Punkte durch einen vorgegebenen Benutzer betrachtet wurden).

[0061] Wenn der Benutzer beispielsweise den Kanalbrowser **218** startet und Informationen vom ladbaren Transport **220** anfordert, bestimmt der ladbare Transport **220**, ob die angeforderten Informationen das entsprechende CDF-Kennzeichen umfassen, welches anzeigt, dass der Inhaltslieferant Informationen bezüglich der Zeit und Dauer der Betrachtung der Informationen protokollieren möchte. Ist dies der Fall, so protokolliert der ladbare Transport **220** Informationen, welchen die Zeit und Dauer der Betrachtung der Informationen durch den Benutzer darstellen. Diese Informationen werden im Zwischenspeicher **206** an einer Position gespeichert, welche diesen speziellen Webinhaltsinformationen entspricht.

[0062] Wenn die Mobilvorrichtung **18** das nächste Mal mit dem Tischcomputer **16** synchronisiert wird, wird nicht nur die Mobilvorrichtung **18** mit den momentanen, durch den Tischcomputer **16** empfangenen Webinhalten aktualisiert, sondern der Tischcomputer **16** wird mit den momentanen Protokollinformationen, die durch die Mobilvorrichtung **18** aufrecht erhalten werden, aktualisiert. Auf ähnliche Weise wird das nächste Mal, wenn der Browser auf dem Tischcomputer **16** auf den entsprechenden Webinhalt vom Inhaltslieferanten **12** zugreift, die Protokollinformation von dem Tischcomputer **16** auf den Inhaltslieferanten **12** übertragen. Da der Browser auf dem Tischcomputer **16** der Internet Explorer 4.0 ist, werden Protokollinformationen, welche auf den Tischcomputer **16** synchronisiert wurden, zum Inhaltslieferanten **12** übertragen, wenn der Scheduler des Internet Explorer 4.0 auf dem Tischcomputer **16** das nächste Mal aufgerufen wird.

[0063] [Fig. 7](#) ist ein vereinfachtes Blockdiagramm eines Abschnitts von [Fig. 6](#), welches sich auf diejenigen Komponenten konzentriert, die verwendet werden, um auf Daten zuzugreifen, welche auf der Mobilvorrichtung **18** gespeichert sind, um sie dem Benutzer durch die Anzeige **45** anzuzeigen. In [Fig. 7](#) sind die nachstehend beschriebenen CDF-Dateien **201**, die Daten **202**, die Skriptenschablonen **204** und die Vorzüge **230** als Teil des Zwischenspeichers **206** gezeigt.

[0064] Ein Aspekt der vorliegenden Erfindung beinhaltet die Verwendung der CDF-Dateien **201** zur Konstruktion von HTML-Seiten **224**. Insbesondere kombinieren HTML-Seiten **224** die Skriptenschablonen **204** und Da-

ten **202** unter Verwendung von in den CDF-Dateien **201** enthaltenen Strukturinformationen. Beispielsweise bestimmt eine spezielle Skriptenschablone **204**, welcher Subkanal einer CDF-Datei **201** angezeigt wird und erhält den Titel und das Logo für diesen Subkanal von der CDF-Datei **201**, wobei sie in die HTML-Seite **224** eingebaut werden. Zusätzlich können bestimmte Datenelemente wie beispielsweise ein Artikel oder Bild und/oder zusätzliche Subkanäle innerhalb des Subkanals von der CDF-Datei **201** erhalten werden, um einen Index an den angezeigten Subkanal darzustellen. Typischerweise können, da Elemente sich beim Erhalt neuer Daten häufig ändern, Elementtitel für die HTML-Seite **224** direkt von den in den Daten **202** gespeicherten Daten erhalten werden. Obgleich dieses Verfahren des Mischens von Skriptenschablonen **204**, Daten **202** und CDF-Dateien **201** komplexer ist als herkömmliche Systeme und Verfahren, bei denen vollständige HTML-Seiten von einem Server auf einen Client übertragen werden, ist dieses Verfahren vorteilhaft. Insbesondere ist das vorliegende System und Verfahren in der Lage, Inhalte in kleinen Datensegmenten anstelle vollständiger HTML-Seiten zu liefern. Diese inkrementale Herangehensweise macht es effizient und ökonomisch, zeitkritische Informationen, wie beispielsweise Aktienmarktwerte, während Spitzenzeiten im Netz zu aktualisieren. Das System und Verfahren erlaubt auch die Verwendung im Zwischenspeicher **206** gespeicherter Standard-Skriptenschablonen **232**. Die Standard-Skriptenschablonen **232** können verwendet werden, um Kanalinhalt (d.h. Subkanäle und Daten) aufzubereiten, wenn eine bestimmte Skriptenschablone fehlt.

[0065] CDF ist ein Standard zur Erzeugung von "Kanälen" zur Verwendung mit Browsern, wie beispielsweise dem Internet Explorer 4.0. Der CDF-Standard basiert auf XML, der Extensible Markup Language. Ein Aspekt der vorliegenden Erfindung enthält zusätzliche Kennzeichen zur Erweiterung des CDF für eine verbesserte Verwendung auf Mobilvorrichtungen. Insbesondere werden die zusätzlichen Kennzeichen verwendet, um durch die CDF-Dateien **201** zu navigieren, wenn Inhalte auf der Anzeige **45** anzuzeigen sind, um den nötigen Speicherplatz auf der Mobilvorrichtung **18** zu reduzieren. Die nachstehende Tabelle 1 ist ein Beispiel einer CDF-Datei in Übereinstimmung mit der vorliegenden Erfindung.

TABELLE 1

K	<pre> <!--Declare item scripts --> <ITEM HREF="Http://www.microsoft.com/test/script1.mcs" ID="IS1"> <USAGE VALUE="None"/> </ITEM> </pre>
	<pre> <ITEM HREF="Http://www.microsoft.com/test/script2.mcs" ID="IS2"> <USAGE VALUE="None"/> </ITEM> </pre>
J	<pre> < -- Declare channel scripts --> <ITEM HREF="Http://www.microsoft.com/test/script3.mcs" ID="CS1"> <USAGE VALUE="None"/> </ITEM> </pre>
	<pre> <ITEM HREF="Http://www.microsoft.com/test/script4.mcs" ID="CS2"> <USAGE VALUE="None"/> </ITEM> </pre>
A	<pre> <CHANNEL HREF = "mctp://www.microsoft.com/test/test.cdf" ID="test" </pre>
C	<pre> BASE ="http://www.microsoft.com/test/ </pre>
B	<pre> SELF ="http://www.microsoft.com/test.cdf/> </pre>
E	<pre> <TITLE VALUE = "Test Channel" /> <!--IS1 is the general item script to use within the channel --> <ITEMSCRIPT VALUE="IS1"/> <!--CS1 is the general channel script to use within the channel --> <CHANSRIPT VALUE = "CS1"/> </pre>
G	<pre> <ITEM HREF = "Http://www.microsoft.com/test/AD1.mad" ID="AD1"> <USAGE VALUE = "MobileAd"/> </ITEM> <ITEM HREF = "Http://www.microsoft.com/test/AD2.mad" ID="AD2"> <USAGE VALUE = "MobileAd"/> </ITEM> </pre>
H	<pre> <CHANNEL ID = "C1" DEFAULTPREF = "ON"> <TITLE VALUE = "Test Subchannel 1" /> -This subchannel is also rendered by the general channel script CS1 --> <ITEM HREF="http://www.microsoft.com/test/item-a.mcd" ID="ITA"> <!--This item is rendered by the item-specific IS2 script --> <ITEMSCRIPT VALUE="IS2"/> <USAGE VALUE="MOBILECHANNEL"/> </ITEM> </CHANNEL> <CHANNEL ID = "C2" DEFAULTPREF = "OFF"> <TITLE VALUE = "Test Subchannel 2" /> -This subchannel is rendered by the channel-specific script CS2 --> <CHANSRIPT VALUE = "CS2" /> <ITEM HREF="http://www.microsoft.com/test/item-b.mcd" ID="ITB"> <!--This item is also rendered by the general IS1 item script --> <USAGE VALUE="MOBILECHANNEL"/> </ITEM> </CHANNEL> </CHANNEL> </pre>

[0066] Mit Bezug auf Tabelle 1 ist der Pfad des CDF in zwei Attributen des Spitzenpegel-"CHANNEL"-Kennzeichens vorhanden, welches den Standard in CDF-Dateien darstellt. Insbesondere verweist das "HREF"-Attribut, welches mit "A" bezeichnet ist, auf den CDF-Pfad unter Verwendung des Transportprotokolls des ladbaren Transports **220** ([Fig. 6](#)), beispielsweise kann ein Mobilkanal-Transportprotokoll (MCTP) verwendet werden. Das HREF-Attribut verwendet den MCTP-Präfix zur Anzeige, dass die CDF-Datei für eine Mobilvorrichtung **18** ist. Wenn die CDF-Datei auf dem Tischcomputer **16** vorhanden ist ([Fig. 1](#)) und nicht auf der Mobilvorrichtung **18**, kann der MCTP-Präfix eine besondere Verarbeitung aufrufen, wenn durch den auf dem Tischcomputer **16** verwendeten Browser darauf verwiesen wird. Anders als das durch Browser verwendete HREF-Attribut, wie beispielsweise der Internet Explorer 4.0 auf dem Tischcomputer **16**, zeigt die URL nicht direkt die aufzubereitende Seite an. Vielmehr verweist das Attribut auf den Spitzenpegelkanal als durch die CDF-Datei spezifiziert.

[0067] Das "SELF"-Attribut "B" verweist auf den CDF-Pfad unter Verwendung des Standard HTTP-Präfix. Anders als die Standardrealisierung des CDF auf dem Tischcomputer **16** wird das SELF-Attribut bevorzugt mit dem Spitzenpegel-CHANNEL-Kennzeichen verwendet.

[0068] Das "BASE"-Attribut "C" weist die selbe Funktionalität in der CDF-Datei der vorliegenden Erfindung auf wie für die Verwendung in einem Browser auf dem Tischcomputer **16**. Seine URL ist eine HTTP-URL. In dem dargestellten Beispiel ist das HREF-Attribut "A" für das Spitzenpegel-CHANNEL-Kennzeichen das einzige, welches eine Transport-Protokollstil-URL aufweist. Alle anderen HREF-Attributwerte sind im HTTP-Stil.

[0069] Mehrere Attribute und Attributwerte, welche in den CDF-Dateien **201** gemäß der vorliegenden Erfindung auftreten, treten in Standard-CDF-Kennzeichen auf. Die Kennzeichen sind in der nachstehenden Tabelle beschrieben.

TABELLE 2

ID	Eine kurze String-Identifizierung für die Elemente CHANNEL, ITEM und LOGO
DEFAULTPREF	Ein Boolean-Operator, welcher die vorgeschlagene Präferenzeinstellung für ein CHANNEL-Element anzeigt. Werte u.a. "An" oder "Aus".
USAGE	Neue Verwendungswerte "Mobilkanal", "Mobiltischcomputer" und "MobilAD".
CHANNEL	Das CHANNEL-Element kann ein USAGE-Kennzeichen annehmen, welches entweder die vorstehend definierten "Mobilkanal"- oder "Mobiltischcomputer"-USAGE-Werte spezifiziert. Es ist erforderlich für das Spitzenpegel-CHANNEL-Element einer mobilen Tischkomponente.

[0070] Jedes Kennzeichen oder Attribut ist nachstehend detailliert beschrieben:

"ID"

[0071] Ein ID-Kennzeichen ist ein Text-String, welcher als ein Attribut zur Identifizierung des spezifizierten Elements verwendet wird. Ein ID-Kennzeichen muss für alle Elemente CHANNEL, ITEM und LOGO in einem Mobilkanal bereitgestellt werden.

ID = "ChanId"

ID = "ItemId"

ID = "LogId"

[0072] Ein ID-Kennzeichen wird für kurze und schnelle Verweise sowohl innerhalb der CDF-Datei **201** als auch innerhalb der Skriptenschablonen **204** verwendet. Innerhalb der CDF-Datei **201** wird das ID-Kennzeichen als ein Wert für CHANSCRIPT und ITEMSCRIPT-Kennzeichen verwendet, welche nachstehend beschrieben werden, welche auf das zugehörige ITEM-Kennzeichen verweisen sollen, das die Skriptenschablone **204** darstellt.

[0073] Innerhalb der Skriptenschablone **204** wird das ID-Kennzeichen, zusammen mit der nachstehend beschriebenen MCTP, verwendet um einzigartige URLs zu bilden. Das ID-Kennzeichen wird durch den ladbaren Transport **220** verwendet, um auf einzigartige Weise auf einen Kanal oder ein Element zu verweisen. MCTP-Verweise weisen die folgende Form auf:

"mctp://Cdfid/ChanID" für einen Kanal oder

"mctp://CDFid/ItemID" für ein Element.

[0074] Die String-Länge eines ID-Kennzeichens wird auf dem Minimum gehalten, welches nötig ist, um es innerhalb des CDF über die Zeit zu definieren. Die ID-String-Länge kurz zu halten ist wichtig, um Netzbandbreite und Speicherplatz zu sparen.

[0075] In der CDF-Datei **201** wird das ID-Kennzeichen des Spitzenpegelkanals als Handgriff zum Kanal verwendet. Die maximale Länge des ID-Strings beträgt 64 Zeichen, doch ein Handgriff von zwischen 6 und 10 Zeichen ist empfohlen, damit das Spitzenpegel-ID einzigartig ist. Nachstehend drei Code-Beispiele.

```
<CHANNEL ID      = " Sports" >
<ITEM HREF       = " www.microsoft.com/test/sports/article001.mcd"
      ID          = " Art1">
<LOGO HREF       = " www.microsoft.com/test/sports/sportslogo.gif"
      STYLE       = " IMAGE"
      ID          = " L_Sports" >
```

[0076] Das ID-Kennzeichen ist für jedes Elternelement bevorzugt und kann nur einmal auftretend sein.

"USAGE"

[0077] Für das USAGE-Kennzeichen werden drei neue Werte geliefert:

(1) MobileChannel

[0078] Die Aussage

```
<USAGE Value = "MobileChannel"/>
```

spezifiziert den Kanal als einen Kanal oder ein Datenelement als zur Verwendung mit der Mobilvorrichtung **18**. Dem Spitzenpegelkanal sollte als ein USAGE-Wert von "MobileChannel" verliehen werden. Wenn der USAGE-Wert auf "MobileChannel" eingestellt wird, werden Elemente durch den Kanalbrowser **218** erkannt und auf der Anzeige **45** der Mobilvorrichtung **18** angezeigt, jedoch nicht durch einen Kanal-Browser auf dem Tischcomputer **16** erkannt. Dieses Merkmal ermöglicht es, spezielle Datenelemente (mit der Dateiendung ".mcd") auf geeignete Weise auf der Mobilvorrichtung **18** anzuzeigen und sie zu ignorieren, wenn die CDF-Datei **201** mit dem Tischcomputer **16** verwendet wird. Z.B.:

```
<ITEM HREF=" http://www.microsoft.com/test1.mcd"      ID=" T1 "
      <USAGE VALUE = "MobileChannel"/>
</ITEM>
<ITEM HREF=" http://www.microsoft.com/test2.md"       ID=" T2 "
      <USAGE VALUE = "None"/>
</ITEM>
```

[0079] Das vorstehende Element T1 ist ein Datenelement für die Mobilvorrichtung **18** und wird durch den Kanalbrowser **218** erkannt, jedoch nicht durch den Kanalbrowser auf dem Tischcomputer **16**. Element T2 ist ein Skript und wird nicht durch die Kanalbrowser für die Mobilvorrichtung **18** oder den Tischcomputer **16** erkannt, da der Wert gleich "Nichts" ("None") ist

(2) MobileDesktopComponent

[0080] Die Aussage

```
<USAGE VALUE = " MobileDesktopComponent"/>
```

spezifiziert den Kanal als eine auf der Anzeige **45** in Verbindung mit anderen Anwendungen, wie beispielsweise persönlichen Informationsverwaltungen, z.B. Scheduling und E-Mail-Anwendungen, sichtbare Komponente.

(3) MobileAd

[0081] Die Aussage

<Usage value = "MobileAd"/>

spezifiziert ein Element als eine auf der Anzeige **45** sichtbare Werbung. Für das nachstehende Beispiel:

```
<ITEM HREF = http://www.microsoft.com/AD1.mad">
    <USAGE VALUE = "MobileAd">
</ITEM>
```

spezifiziert "AD1.mad" die Anzeige der Werbung gemäß den Skriptenschablonen **204**, welche nachfolgend beschrieben werden. In dem Beispiel aus Tabelle 1 sind zwei Werbungen im Spitzenpegelkanal bei "E" definiert. Wenn der Spitzenpegelkanal auf der Anzeige **45** gemäß der Skriptenschablone **204** angezeigt wird, können eine oder beide Werbungen ebenfalls angezeigt werden. Die Skriptenschablone **204** kann die Anzeige der Werbungen abwechseln, wenn mehr als eine vorhanden sind. In dem Beispiel aus Tabelle 1 sind die Werbungen im Hinblick auf den Spitzenpegelkanal definiert; jedoch erlaubt ein Merkmal der vorliegenden Erfindung, dass die Werbungen auch angezeigt werden können, wenn die Subkanäle "C1" oder "C2" ebenfalls angezeigt werden, ohne dass die Syntax für die Werbungen in der Subkanal-Definition wiederholt werden muss. Wie hierin verwendet, wird dieses Merkmal als "Erbe" bezeichnet. Wenn gewünscht, können neue Werbungen in den Subkanälen definiert werden.

DEFAULTPREF

[0082] Das DEFAULTPREF-Kennzeichen kann wie folgt verwendet werden:

```
<CHANNEL
ID = "ChanId"
DEFAULTPREF = "ON" | "OFF"
>
```

[0083] In dem Beispiel aus Tabelle 1 werden die DEFAULTPREF-Kennzeichen bei "G" und "H" verwendet.

[0084] Das DEFAULTPREF-Kennzeichen markiert einen Subkanal mit spezifischen Standardpräferenzen. Anders gesagt regelt das DEFAULTPREF-Kennzeichen, für welche Subkanäle ein Benutzer standardmäßig Inhalte empfangen wird. Standardmäßig werden, wenn Inhalte für einen neuen Kanal an die Mobilvorrichtung **18** geliefert werden, Elemente innerhalb von Subkanälen, welche nicht mit dem Attribut DEFAULTPREF = "OFF" markiert sind, nicht übertragen.

[0085] Das Merkmal erlaubt es dem Inhaltslieferanten **12**, einen Kanal zu schaffen, welcher mehr Inhalte bietet als vernünftigerweise durch die auf der Mobilvorrichtung **18** verfügbaren begrenzten Speicherressourcen untergebracht werden können, und überschwemmt dennoch standardmäßig nicht die Mobilvorrichtung **18** mit allen Inhalten des Kanals. In einer Ausführungsform werden die DEFAULTPREF-Kennzeichen überprüft und bei **230** gespeichert, wenn der Kanal erstmals der Mobilvorrichtung **18** geliefert wird. Der Benutzer kann seine Präferenzen **230** wie gewünscht speichern, um mehr oder weniger Inhalte einzuschließen als die DEFAULTPREF-Einstellungen erlauben.

[0086] Das DEFAULTPREF-Kennzeichen kann Werte von entweder "ON" oder "OFF" aufweisen. Wenn das DEFAULTPREF-Attribut nicht spezifiziert ist, behandelt die Mobilvorrichtung **18** den Subkanal, als wäre er mit DEFAULTPREF = "ON" markiert. Das DEFAULTPREF-Kennzeichen kann nur einmal in einem Kanalelement erscheinen.

[0087] Zusätzlich zu den vorstehend beschriebenen Kennzeichen, welche in CDF-Dateien des Standes der Technik vorhanden sind, jedoch jetzt neue Attribute aufweisen, beinhalten die CDF-Dateien **201** auch neue Kennzeichen zur Verwendung mit den Skriptenschablonen **204** und Daten **202**. Die neuen Kennzeichen sind in der nachfolgenden Tabelle beschrieben.

TABELLE 3

Kennzeichen	Beschreibung
CHANSRIPT	Identifiziert die ID der Skriptendatei, um den Kanal und Subkanäle aufzubereiten.
ITEMSCRIPT	Identifiziert die ID der Skriptendatei, um die Elementendaten-Datei aufzubereiten.
ITEMFORMAT	Definiert die Dateistruktur für Datendateien.

[0088] Jedes Kennzeichen wird nachfolgend ausführlich beschrieben:

CHANSRIPT

[0089] Das CHANSRIPT-Kennzeichen weist im Allgemeinen folgende Form auf

<CHANSRIPT VALUE = "ChannelID"/>

wobei "ChannelID" die URL spezifiziert. In dem Beispiel aus Tabelle 1 wird das Skript für die Anzeige eines Kanals oder eines Subkanals mit einer Extension ".mcs" wie bei "J" angezeigt gekennzeichnet. In einer Ausführungsform wird "Kennzeichenerbe" geliefert, wo der CHANSRIPT-Kennzeichen-Wert auf alle Kanäle oder "Kinder"-Kanäle des momentanen Kanals oder Subkanals zutrifft. Das Subkanal-CHANSRIPT-Kennzeichen übertrifft alle CHANSRIPT-Werte, welche vorher durch ein höheres oder "Eltern"-Kanal-Element definiert waren. Das VALUE-Attribut spezifiziert die ID des ITEM-Elements, welches dem Skript entspricht, das ausgeführt werden soll, um diesen Pegel des Kanals aufzubereiten.

[0090] In Tabelle 1 spezifiziert beispielsweise

<CHANSRIPT VALUE = CS1"/>

die Verwendung einer bestimmten Skriptenschablone, wo das durch "CS1" identifizierte Kanalskript im Anfangsabschnitt der CDF-Datei definiert wurde.

[0091] Das Spitzenpegel-CHANNEL-Element kann mindestens ein CHANSRIPT-Kennzeichen als Kind-Element aufweisen. Jeder Subkanal kann maximal ein derartiges Kennzeichen aufweisen.

ITEMSCRIPT

[0092] Das ITEMSCRIPT-Kennzeichen weist im Allgemeinen folgende Form auf

<ITEMSCRIPT VALUE = "ItemID"/>

wobei "ItemID" die URL spezifiziert. In dem Beispiel aus Tabelle 1 ist das Skript für die Anzeige eines Elements mit einer Extension ".mcs" gekennzeichnet, wie bei "K" angezeigt. In einer Ausführungsform ist "Kennzeichenerbe" bereitgestellt, wobei der ITEMSCRIPT-Kennzeichen-Wert auf alle Kind-Elemente des momentanen Kanals oder Subkanals zutrifft. Das Subkanal-ITEMSCRIPT-Kennzeichen übertrifft jeden vorher durch ein Eltern-CHANNEL-Element definierten ITEMSCRIPT-Wert. Das VALUE-Attribut spezifiziert die ID des ITEM-Elements, welches dem auszuführenden Skript entspricht, um diesen Pegel des Kanals aufzubereiten. In Tabelle 1 spezifiziert beispielsweise

<ITEMSCRIPT VALUE = "IS1"/>

eine bestimmte Skriptenschablone, wobei das durch "IS1" identifizierte Skript vorher in der CDF definiert war. Es sollte sich verstehen, dass wenn die nachfolgende Aussage:

<USAGE VALUE = "NONE"/>

in Verbindung mit einer Skriptendatei auftritt, das VALUE-Attribut "None" für das USAGE-Kennzeichen verhindert, dass die Skriptendatei durch den Kanalbrowser **218** angezeigt wird.

[0093] Das oberste CHANNEL-Element kann mindestens ein ITEMSCRIPT-Kennzeichen aufweisen. Bei allen anderen Kanalpegeln kann höchstens ein derartiges Kennzeichen vorhanden sein.

ITEMFORMAT

[0094] Das ITEMFORMAT-Kennzeichen weist im Allgemeinen folgende Form auf

<ITEMFORMAT VALUE = "header_block; repeat_block"/>

[0095] Das Kennzeichen spezifiziert das Format einer Klasse von Datenelementen durch Identifikation der zugehörigen Dateistruktur. Die Datenelemente sind einfache Textdateien, welche einen einzigartigen Anfangsblock und eine sich wiederholende Blockstruktur für speicherorientierte Daten aufweisen können. Assistenzfunktionen sind in der Skriptingumgebung bereitgestellt, um auf die Dateninhalte unter Verwendung von in dem ITEMFORMAT-Kennzeichen enthaltenen Informationen zuzugreifen. Sowohl header block als auch repeat block sind optional. Doch mindestens das eine oder das andere, also header block oder repeat block, muss existieren. Existiert repeat block, so sollte ihm ein Abgrenzer vorangehen, beispielsweise ein Semikolon (;), wie vorstehend angezeigt. Der Anfangsblock enthält typischerweise die Beschreibung über die Datenelemente. Und der wiederholbare Datenblock enthält eine Beschreibung über bestimmte Datenelemente darin. Der header block und repeat block weisen folgende Form auf:

$V_i[T_i]$, für $i = 1$ bis n .

[0096] "Vi" stellt hier den Feldnamen des Blockwertes dar, und "Ti" ist der optionale Typ des Blockwertes. Wenn "Ti" weggelassen wird, wird ein Standardwert "HTML" angenommen. Gültige Typen sind in der nachfolgenden Tabelle aufgelistet:

TABELLE 4

Typ	Beschreibung
HTML	HTML-Text einschließlich Markup.
TEXT	s. HTML.
IMG	ID von Bildelement in CDF-Dateien.
HREF	URL zu einer Seite, beispielsweise Datendatei und Kanalskript

[0097] Ein Datenblock ist lediglich eine Gruppe von Werten. Es existiert ein Wert pro Zeile. Jegliche bedeutende Datendatei sollte mindestens einen Datenblock aufweisen. Drei Datenblöcke können beispielsweise verwendet werden, um ein Portfolio von drei Aktien zu zeigen. In dem nachfolgenden Beispiel zeigt ein "Markt"-Kanal Aktienwerte an, welche in der "Stocks.mcd"-Datei gelistet sind. Der Anfangsblock gibt den Titel an und zeigt die Daten der gezeigten Aktien an. Die zu listenden Daten schließen den Namen, den niedrigen Preis, den hohen Preis und Abschlusspreise jeder Aktie ein.

```
<ITEM HREF="http://www.market.com/Stocks.mcs"ID="Stock_S"
  <USAGE VALUE="None"
</ITEM>
.....
<CHANNEL ID= "Stock_C">
  <TITLE VALUE="Market"/>
  <ITEM HREF="http://www.market.com/stocks.mcd"
    ID="Stock_D">
    <USAGE VALUE="MobileChannel"/>
    <ITEMSCRIPT VALUE="Stock_S/>
    <ITEMFORMAT VALUE="Title,Date,Picture=IMG;
      Name, Low, High, Close"/>
  </ITEM>
</CHANNEL>
```

[0098] Der Anfangsblock weist hier drei Werte "Titel", "Datum" und "Bild" auf, und der Datenblock weist vier auf: "Name", "Niedrig", "Hoch" und "Abschluss".

[0099] IMG zeigt das Feld an, welches ein Bild, wie beispielsweise JPG oder GIF repräsentiert. Der Feldwert ist der Identifikator des Elements, welches die URL des Bildes definiert. Das eingebaute Elementenskript erschafft einen IMG-Wert, um dieses Element anzuzeigen.

[0100] Der Datenblock kann wiederholt werden, um eine Tabelle von Aktienwerten aufzubauen. Wenn die Datei Stocks.mcd einen einzelnen Datenblock enthält, zeigt das Skript eine einzige Aktie pro Seite an. Wenn sie eine Vielzahl von Datenblöcken enthält, könnte das Skript eine Tabelle von Aktien anzeigen.

[0101] Der Wiederholungsblock kann weggelassen werden, wie in dem nachfolgenden Beispiel gezeigt, welches einen Nachrichtenartikel mit einem Titel, einem Bild und dem Textkörper darstellt:
`<ITEMFORMAT VALUE = "TITLE, PICTURE = IMG, BODY"/>`

[0102] Der Anfangsblock kann auch weggelassen werden, wie in dem nachfolgenden Beispiel gezeigt, welches eine Aktienseite mit einer Auflistung von Aktien darstellt. Es versteht sich, dass der Semikolon (;) vorhanden ist, um die Werteliste anzuzeigen. Es ist daher ein Wiederholungsblock und nicht ein Anfangsblock.
`<ITEMFORMAT VALUE = "; Name, Low, High, Close"/>`

[0103] Es sollte sich verstehen, dass nicht alle Standardkennzeichen unterstützt und auf der Mobilvorrichtung **18** verwendet werden.

[0104] Insbesondere müssen die CDF-Dateien **201** keine Softwareaktualisierungs-Kanalkennzeichen (EAR-LIESTIME-, INTERVALTIME- und LASTTIME-Kennzeichen) unterstützen, wenn der Kanalbrowser **218** nur in einem "Off-Line"-Modus betrieben wird. Zusätzlich kann das LOGIN-Kennzeichen ignoriert werden. Während jedoch das EAR-LIESTIME-, INTERVALTIME- und LASTTIME-Kennzeichen auf der Mobilvorrichtung **18** ignoriert werden, wenn diese Off-Line betrieben wird, werden sie unterstützt, wenn die Mobilvorrichtung **18** in einem "Online"-Modus betrieben werden kann. Zusätzlich werden, wenn die Strukturen der CDF-Datei **201**, der Skriptenschablonen-Dateien **204** und der Datendatei **202** auf dem Tischcomputer **16** oder einer anderen ähnlichen Vorrichtung verwendet werden, die Software-Aktualisierungskennzeichen erkannt und verwendet, um den Kanal von dem Inhaltslieferanten **12** herunterzuladen.

Mobilkanal-Datendateien

[0105] Die Datendateien **202** werden verwendet, um Inkrementaldaten für einen Kanal zu übertragen. Dies sind einfache Textdateien, welche Daten mit einem Datenelement pro Zeile in der Datei enthalten. Innerhalb der CDF-Datei **201** kann eine Struktur für die Datendatei unter Verwendung eines ITEMFORMAT-Kennzeichens wie vorstehend beschrieben erklärt werden.

[0106] In dem Beispiel weist jede Datendatei eine Extension ".mcd" auf. Die CDF-Datei **201** weist ein ITEM-Kennzeichen auf, um die "mcd"-Datei zu definieren. Innerhalb des ITEM-Kennzeichens wird das ID-Attribut als schnelle Art des Verweises auf die "mcd"-Datei von innerhalb des Skriptes verwendet, ohne dass auf ihre vollständige URL verwiesen werden muss.

[0107] Anders als die konventionelle Herangehensweise von Skriptenlaufwerken, wo Skriptdateien ausgeführt werden und Daten für die Anzeige anfordern, wird im vorliegenden System das Gegenteil für die Anzeige inkrementaler Daten durchgeführt. Da neue Informationen zu jeder Zeit innerhalb einer neuen "mcd"-Datei hereinkommen können, ist es effizienter, die Skriptenschablone **204** zu aktivieren, um Daten **202** anzuzeigen, wenn sie eintreffen. Diese Herangehensweise, bei welcher Daten Skripten auslösen, hat einen zusätzlichen Vorteil darin, dass sie das Erbe von Skriptenschablonen erlaubt.

[0108] Das Dateiformat für Datendateien **202** ist flexibel. Es ist einfach eine Textdatei, welche die Daten enthält, wie beispielsweise Verweise auf Bilder mit jedem Element auf einer eigenen Zeile. Das vorliegende System weist Verfahren innerhalb der Skriptingumgebung zum Lesen dieses Inhalts von der Datendatei **202** auf.

[0109] Das ITEMFORMAT-Kennzeichen wird verwendet, um die Datenart zu spezifizieren, welche in der Datei vorhanden ist. Im Allgemeinen weisen Dateien das folgende Format auf:

[Anfangsblock]
 [Datenblock 1]

[Datenblock 2]

...

[Datenblock n]

[0110] Wie vorstehend beschrieben wird die Verwendung bestimmter Skriptenschablonen **204** in den CDF-Dateien **201** mittels der CHANNELSKRIPT- und ITEMSCRIPT-Kennzeichen identifiziert. Die Skriptenschablonen **204** werden auf datengetriebene Weise aufgerufen. Wenn es Zeit ist, eine bestimmte Datendatei **202** anzuzeigen, wird auf die Datendatei **202** verwiesen, und die geeignete Elementen-Skriptenschablone wird lokalisiert, um sie anzuzeigen. Auf ähnliche Weise wird, wenn es Zeit ist, Kanalinhalt zu zeigen, wie beispielsweise eine Listung von Subkanälen oder Datenelementen, auf den Kanal verwiesen und die geeignete Kanal-Skriptenschablone wird lokalisiert, um sie anzuzeigen.

ITEMSKRIPT-SELEKTION

[0111] Wenn es nötig ist, eine bestimmte Datendatei **202** durch den Kanalbrowser **218** aufzubereiten, muss die geeignete Skriptenschablone **204** ausgewählt werden. Elementen-Skriptenschablonen sind verantwortlich für die Aufbereitung der Datendateien und werden als Ergebnis des Verweises auf die URL der Datendatei **202** aufgerufen. Die geeignete Skriptenschablone **204** wird basierend auf der Nähe eines ITEMSCRIPT-Kennzeichens zu der bestimmten Datendatei **202** ausgewählt, um aufbereitet zu werden.

[0112] In der dargestellten Ausführungsform weist eine Elementen-URL, wie sie in Skripten erscheint, die folgende Form auf:

Mctp://CDFid/ItemID

[0113] Hier spezifiziert mctp die Verwendung des Mobilkanal-Transportprotokolls, um die URL aufzulösen und die Skriptingmaschine aufzurufen. CDFid ist das ID-Kennzeichen des Spitzenpegel-CHANNEL-Elements und wird verwendet, um die korrekte CDF-Datei in den CDF-Dateien **201** zu wählen. ItemId ist das ID-Kennzeichen des ITEM-Elements für die aufzubereitende Datendatei **202**.

[0114] Die Datendatei erscheint in einem ITEM-Element innerhalb der Kanalhierarchie. Die Position des ITEM-Elements relativ zu einem ITEMSCRIPT-Kennzeichen bestimmt, welche Skriptenschablone **204** verwendet wird, um die Daten aufzubereiten. Die Skriptenschablone **204** wird durch Abgleich des ID-Wertes in dem ITEMSCRIPT-Kennzeichen mit dem ID-Wert eines ITEM-Elements, welches das Element für die Skriptenschablone **204** ist, innerhalb der CDF-Datei **201** identifiziert.

[0115] ITEMSCRIPT-Kennzeichen können Kinder entweder von CHANNEL-Elementen oder ITEM-Elementen sein. Ein ITEMSCRIPT-Kennzeichen bestimmt die Skriptenschablone **204**, die für alle Elemente des momentanen Kanals und seiner Subkanäle zu verwenden ist, wenn keine weiteren ITEMSCRIPT-Kennzeichen mit den Subkanälen in Zusammenhang stehen. Eine Elementenskript-Schablone **204**, wie durch ITEMSCRIPT identifiziert, für ein CHANNEL- oder ITEM-Element übertrifft jeden vorher definierten ITEMSCRIPT-Wert.

[0116] Somit wird ein Erbmodell verwendet. Wenn es nötig ist, ein bestimmtes Datenelement aufzubereiten, wird das nächstliegende ITEMSCRIPT-Element in der Hierarchie verwendet, um zu bestimmen, welche Skriptenschablone **204** das Datenelement aufbereiten sollte. Für den Fall, dass die geeignete Skriptenschablone **204** auf der Mobilvorrichtung **18** nicht verfügbar ist, wird eine eingebaute Skriptenschablone zur Aufbereitung des Datenelementes verwendet. Die Standard-Elementen-Skriptenschablone ist bei **232** gespeichert und zählt einfach durch alle Felder, welche in dem ITEMFORMAT-Kennzeichen spezifiziert sind, und zeigt für jedes die geeigneten Daten aus der spezifizierten Datendatei an. Wenn die Datendatei einen Wiederholungsblock enthält, werden alle Blockwerte auf der Seite in einer Liste aufgezählt, bis das Ende der Datendatei erreicht wird.

[0117] Üblicherweise werden Elementen-Skriptenschablonen auf dem Spitzenpegel eines Kanals spezifiziert, gewöhnlich an der Spitze der Datei, wie in Tabelle 1 gezeigt. Jeder Elementen-Skriptenschablone wird eine einmalige ID zugeordnet. Auf das Skript kann dann unter Verwendung eines ITEMSCRIPT-Elements von jeder beliebigen Position in dem CDF-ladbaren Transport **220** verweisen, anstelle des gewöhnlichen HTTP-Protokolls.

[0118] Es sollte sich verstehen, dass aufgrund einer Begrenzung der Art und Weise, wie Standard-Kanalbrowser für den Tischcomputer **16** die URLs für Bilder handhaben, damit ein Bild auf dem Tischcomputer **16** aufbereitet wird, auf Bilder unter Verwendung von Standard-HTTP-Verweisen verwiesen werden sollte, anstelle des ladbaren Transports. Wenn erlaubt wird, dass der Kanal gesichtet wird, ist der Tischcomputer **16**

nicht von Bedeutung, so können URLs der ladbaren Transportart verwendet werden.

KANAL-SKRIPTEN-AUSWAHL

[0119] Zusätzlich zur Aufbereitung von Daten aus Datendateien, wie beispielsweise ein Nachrichtenartikel, ist es üblicherweise notwendig, die momentane Position innerhalb der CDF aufzubereiten, so dass der Benutzer zu den erwünschten Daten navigieren kann. Wenn es notwendig ist, eine CDF-Navigationsseite (d.h. den Kanal und/oder die Subkanäle) auf der Anzeige **45** aufzubereiten, so muss die geeignete Skriptenschablone ausgewählt werden. Kanal-Skriptenschablonen sind verantwortlich für die Aufbereitung der CDF-Navigationsseiten, welche durch den Inhaltslieferanten **12** geliefert werden. Wie bei einer Elementen-Skriptenschablone führt der Verweis auf die URL des Subkanals zur Aufrufung einer Kanal-Skriptenschablone. Die geeignete Skriptenschablone wird basierend auf der Nähe eines CHANSRIPT-Kennzeichens zu dem bestimmten CHANNEL-Element in der URL gewählt.

[0120] Eine Kanal-URL, wie sie in Skriptenschablonen auftritt, weist die folgende Form auf:
mctp://CDFid/ChanID

[0121] Hier spezifiziert mctp die Verwendung des ladbaren Transportprotokolls zur Auflösung der URL und Aufrufung der Skriptingmaschine. "CDFid" ist das ID-Kennzeichen des Spitzenpegel-CHANNEL-Elements. Es wird verwendet, um die korrekte CDF-Datei auszuwählen. "ChanId" ist das ID-Kennzeichen des CHANNEL-Elements für den Subkanal innerhalb der aufzubereitenden CDF. Wenn ein Benutzer durch den Kanal navigiert, bewegt er sich effektiv auf- und abwärts durch die CDF-Kanalhierarchie und greift auf Datendateien zu. Auf jedem Pegel in der Kanalhierarchie ist es möglich, eine Skriptenschablone zuzuordnen, um den Kanalinhalt anzuzeigen, welcher gewöhnlich eine Liste von Subkanälen oder verfügbaren Elementen ist.

[0122] Das CHANSRIPT-Kennzeichen identifiziert die zu verwendende Skriptenschablone, um die momentane Kanalposition auf eine Weise aufzubereiten, welche ähnlich ist, wie die, in der das ITEM-SCRIPT-Kennzeichen die Skriptenschablone identifiziert, um Daten aufzubereiten. Die Position eines CHANNEL-Elements relativ zu einem CHANSRIPT bestimmt, welche Skriptenschablone verwendet wird, um den Subkanal aufzubereiten. Die Skriptenschablone wird durch Abgleich des ID-Wertes in dem CHANSRIPT-Element mit dem ID-Wert eines ITEM-Elements, d.h. das ITEM für die Skriptenschablone, innerhalb der CDF identifiziert.

[0123] CHANSRIPT-Kennzeichen sind Kinder von CHANNEL-Elementen. Ein CHANSRIPT-Kennzeichen bestimmt die Skriptenschablone, welche für den momentanen Kanal und seine Subkanäle zu verwenden ist. Ein CHANSRIPT-Kennzeichen, welches für ein CHANNEL-Element spezifiziert ist, übertrifft jeden vorher definierten CHANSRIPT-Wert.

[0124] Somit wird ein Erbmodell verwendet. Wenn es nötig ist, einen bestimmten Subkanal aufzubereiten, wird das nächststehende CHAN-SCRIPT-Kennzeichen aufwärts in der Hierarchie verwendet, um zu bestimmen, welche Skriptenschablone die Daten aufbereiten soll. In dem Fall, dass die geeignete Skriptenschablone nicht auf der Mobilvorrichtung **18** verfügbar ist, wird eine eingebaute Skriptenschablone **232** verwendet, um den Kanal so gut sie kann aufzubereiten.

SKRIPTING

[0125] Die Skriptenschablone spezifiziert das Layout und Verhalten von HTML-Seiten. Skriptensegmente sind zwischen den Abgrenzungspaaren "<%" und "%>" oder "<%= " und "%>" eingeschlossen. Das zweite Paar von Abgrenzungen bringt eine spezielle Verwendung und Bedeutung in der Skriptenschablone mit sich. In jedem Skriptensegment muss mindestens eine gültige ausführbare oder nichtkommentierende Aussage vorliegen. Es muss auch mindestens ein Skriptingsegment in der mcs-Datei vorliegen. Jedes leere Skriptensegment erzeugt einen Syntax-Fehler. Skriptingsegmente können frei mit Standard-HTML-Text vermischt sein, vorausgesetzt, dass die skriptengenerierte HTML-Ausgabe die gültige Syntax innerhalb des Kontextes des Standard-HTML-Anzeigecodes aufweist. Anhang A liefert ein Beispiel eines CHANNELSCRIPT und eines ITEM-SCRIPT. Anhang B ist eine Beschreibung einer Skriptingumgebung zur Verwendung in dem vorliegenden System.

[0126] [Fig. 8](#) stellt ein Beispiel einer Indexbetrachter-Benutzerschnittstelle **240** zur Aufbereitung von Inhaltsinformationen unter Verwendung der CDF-Dateien **201**, der Skriptenschablonen **204** und der Daten **202** dar. Die Indexbetrachter-Benutzerschnittstelle **240** kann die vollständige Anzeige **45** bedecken, wenn die Informationen aufbereitet sind, oder alternativ kann die Indexbetrachter-Benutzerschnittstelle **240** ein "Fenster" einer

größeren auf der Anzeige **45** präsentierten graphischen Benutzerschnittstelle sein.

[0127] In [Fig. 5](#) ist der Spitzenpegelkanal bei **242** angezeigt. Der Spitzenpegelkanal **242** weist allgemeine Subkanal-Kategorien auf, so beispielsweise "Nachrichten und Technologie", "Sport", "Wirtschaft", "Unterhaltung", "Lifestyle und Reisen", "Das Microsoft-Netzwerk" und "MSNBC". Die Subkanäle **244**, **245**, **246**, **247**, **248**, **249** und **250** entsprechen den Subkanälen des ersten Pegels "Test-Subkanal 1" und "Test-Subkanal 2", welche in dem Beispiel aus Tabelle 1 gezeigt sind. Jeder der Subkanäle **244** bis **250** führt zu weiteren Subkanälen und/oder Datenelementen. In dem Beispiel aus [Fig. 5](#) weist der Subkanal **245** weitere Subkanäle **252**, **253**, **254** und **255** auf. In diesem Beispiel weist der Subkanal **252** noch weitere Subkanäle auf, welche allgemein bei **257** angezeigt sind, und welche bei Zugriff den neuesten Artikel anzeigen, der zu der gelisteten Subjektüberschrift gehört. Ein Anzeigeknopf **260** kann dem Benutzer bereitgestellt werden, wenn zusätzliche Informationen vorhanden sind, aber nicht gezeigt werden. In diesem Beispiel zeigt der Anzeigeknopf **260** an, dass zusätzliche Subkanäle zu dem Spitzenpegelkanal **242** vorhanden sind. Ein Abschnitt **264** der Benutzerschnittstelle **240** kann für Werbung beiseite gestellt werden.

[0128] Zur Anzeige der Inhaltsinformationen des Spitzenpegelkanals **242** greift der Kanalbrowser **218** ([Fig. 6](#)) auf den ladbaren Transport **220** zu, um die CDF-Datei für den Spitzenpegelkanal **242** in den gespeicherten CDF-Dateien **201** zu lokalisieren. Durch Untersuchung der CDF-Datei nach dem Spitzenpegelkanal **242** bestimmt der ladbare Transport **220**, welche Skriptendatei von den Skriptenschablonen **204** ausgeführt wird, um die Titel des in [Fig. 8](#) dargestellten Subkanals **244** bis **250** aufzubereiten. Es sollte sich verstehen, dass die Titel der Subkanäle **244** bis **250** nicht in der Skriptendatei enthalten sind, welche ausgeführt wird, sondern vielmehr einen Teil der CDF-Datei für den Spitzenpegelkanal **242** bilden. Somit untersucht die zur Aufbereitung des Spitzenpegelkanals **242** verwendete Skriptendatei die CDF-Datei nach dem Spitzenpegelkanal **242**, um die HTML-Seite **224** wie durch den Kanalbrowser **218** geliefert zu erzeugen. Dies erlaubt es allgemeinen Skriptendateien, verwendet zu werden, anstatt nur bestimmten Skriptendateien für jeden Abschnitt der aufbereiteten Informationen. Wie vorstehend beschrieben, kann wenn ein Subkanal, wie beispielsweise der Subkanal **245**, aufbereitet wird, eine andere Skriptendatei aufgerufen werden, oder wenn eine nicht aufgerufen wird, die für die Erzeugung des Spitzenpegelkanals **242** verwendete Skriptendatei verwendet werden. Wiederum untersuchen, wenn ein beliebiger der Subkanäle **252** bis **255** oder **257** aufzubereiten sind, die ausgeführten Skripten die CDF-Datei nach dem Spitzenpegelkanal **242**, um relevante Information zu erhalten, welche angezeigt werden, wie auch zu bestimmen, wie die Information in der Hierarchie anzuzeigen ist. Zusammen mit den Skriptenschablonen **204** und den Daten **220** können die CDF-Dateien wie gewünscht aktualisiert werden.

[0129] Obgleich die vorliegende Erfindung mit Bezug auf bevorzugte Ausführungsformen beschrieben wurde, werden Fachleute erkennen, dass Änderungen in Form und Detail vorgenommen werden können, ohne vom Schutzzumfang der Erfindung abzuweichen.

Beispielhaftes Kanalskript

```

<html>
<%
    Set MC = Server.CreateObject("MobileChannels.Utilities")
    URL = Request.ServerVariables("URL")
    DataID = Request.QueryString("DATAID")
    Pieces = Split(URL, "/")
    ChanID = Pieces(2)

    'get logo and title of channel
    TopElem = MC.Locate(ChanID)
    ChanTitle = " "
    LogoHref = 0
    If TopElem Then
        ChanTitle = MC~Title(TopElem)
        LogoElem = TopElem
        LogoElem = MC.Navigate(LogoElem, "INMATCH", "LOGO")
        Do While LogoElem
            LogostyleElem = MC.Navigate(LogoElem, "IMMATCH", "STYLE")
            If LogoStyleElem Then
                If Strcomp(MC.value(LogoStyleElem), "IMAGE", 1) = 0 Then
                    LogoHref = MC.Href(LogoElem)
                    If LogoHref Then
                        If MC.HrefExists(LogoHref) Then
                            Exit Do
                        Else
                            LogoHref = 0
                        End If
                    End If
                End If
            End If
        End While
        LogoHref = 0
    End If
End If

```

```

LogoElem = MC.Navigate(LogoElem, "NEXT")
    If LogoElem Then
        LogoElem = MC.Navigate(LogoElem, "MATCH", "LOGD")
    End If

Loop
End If
NeedTitle = 1
Response.Write("<head><title>" & ChanTitle & "</title></head>")
If LogoHref Then
    Response.Write("<body><a href=mnctp://" & ChanID & ">
        </a><br>&nbsp; <br>")
ElseIf ChanTitle And Len(ChanTitle) Then
    Response.Write("<body><a href=mnctp://" & ChanID & ">
        <h3>" & ChanTitle & "</h3></a>")
    NeedTitle = 0
Else
    Response.Write("<body>")
End If
'decide whether you need a title for this chan/subchan
    If DataID And (DataID <> ChanID) Then
        NeedTitle = 1
    Else
DataID = ChanID
    Else If
        SubTitle = 0
        SubElem = MC.Locate(DataID)
        If SubElem Then
            SubTitle = MC.Title(SubElem)
            If SubTitle And NeedTitle Then
                Response.Write("<b>" & SubTitle & "</b><br>&nbsp; <br>")
            End If
        End If
'display contents of chan/subchan
        Response.Write("<table border=0 cellpadding=-2 cellspacing=-2>")

```

```

ChildElem = MC.Navigate(SubElem, "In")
Do While ChildElem
    ShowIt = 1
    IsChan = 0
    If MC.Tag(ChildElem) = "CHANNEL" Then
        IsChan = 1
        If Not MC.IsSubscribed(ChildElem) Then
            ShowIt = 0
        End If
    ElseIf MC.Tag(ChildElem) = "ITEM" Then
        VisparElem = MC.Navigate(ChildElem, "InMatch", "USAGE")
        If VisparElem Then
            Usage = MC.Value(VisFarElem)
            If usage Then
                If StrComp(Usage, "None", 1) = 0 Then
                    ShowIt = 0
                End If
            End If
        End If
    End If
    'be sure item exists
    If ShowIt Then
        ChildHref = MC.Href(ChildElem)
        If ChildHref Then
            If Not MC.HrefExists(ChildHref) Then
                ShowIt = 0
            End If
        Else
            ShowIt = 0
        End If
    End If
Else
    ShowIt = 0
End If
If ShowIt Then

```

'be sure you can get the ID

IDVal = 0

IDElem = MC.Navigate (ChildElem, "InMatch", "ID")

If IDElem Then

IDVal = MC.Value(IDElem)

End If

If Not IDVal Then

ShowIt = 0

End If

End If

'get title

If ShowIt Then

ItemTitle = MC.Title(ChildElem)

If Not ItemTitle Or (Len)ItemTitle = 0)

Then ShowIt = 0

Else

If Len(ItemTitle) > 26 Then

ItemTitle = Mid(ItemTitle,0,25) & "..."

End If

End If

End If

'You know what it is and are going to try to show it

If ShowIt Then

If IsChan Then

Response.Write("<tr><td>*")

Else

Response Write ("<tr><td>")

End If

Response.Write("<td> <a href=mctp://" & ChanID & "/"

& IDVal & ">" & ItemTitle & "")

End If

ChildElem = MC.Navigate(ChildElem,"Next")

Loop

Response.Write("</table>")

Else

Response.Write("Data ID not found.")

End If

%>

<hr>

Note: This page was automatically generated because the correct scripts could not be found. If this problem persists after synchronization, please contact the content provider.

</body>

</html>

Beispielhaftes Elementenskript

```

<html>
<%
    Set MC    = Server.CreateObject("MobileChannels.Utilities")
    URL       = Request.ServerVariables ("URL")
    DataID    = Request.QueryString("DATAID")
    Pieces    = Split(URL,"/")
    ChanID    = Pieces(2)

' get logo and title of channel
    TopElem = MC.Locate(ChanID)
    ChanTitle = " "
    LogoHref = 0
    If TopElem Then
        ChanTitle = MC.Title(TopElem)
        LogoElem = TopElem
        LogoElem = MC.Navigate(LogoElem,"INMATCH","LOGO")
    Do While LogoElem
        LogoStyleElem = MC.Navigate(LogoElem,"INMATCH","STYLE")
        If LogoStyleElem Then
            If StrComp(MC.Value(LogoStyleElem),"IMAGE,1") = 0 Then
                LogoHref = MC.Href(LogoElem)
                If LogoHref Then
                    If MC.HrefExists(LogoHref) Then
                        Exit Do
                    Else
                        LogoHref = 0
                    End If
                End If
            End If
        End If
    End While
    LogoElem = MC.Navigate(LogoElem,"NEXT")
    If LogoElem Then
        LogoElem = MC.Navigate(LogoElem,"MATCH","LOGO")
    End If
%>

```



```
End If
Loop
End If

Response.Write("<head><title>" & ChanTitle & "</title></head>")
If LogoHref Then
    Response.Write("<body><a href=mctp://" & ChanID & ">
        </a><br>")
ElseIf ChanTitle and Len(ChanTitle) Then
    Response.Write("<body><a href=mctp://" & ChanID & ">
        <h3>" & ChanTitle & "</h3></a>")
Else
    Response.Write("<body>")
End If
```

```

' dump article out best we can
ArtElem = 0
If DataID Then
    ArtElem = MC.Locate(DataID)
End If
If ArtElem Then
    For Blk=0 To 100
        Data = MC.Data(ArtElem,Blk)
        If Not Data.Count Then
            Exit For
        End If
        For Field=0 To Data.Count - 1
            Tag = Data(Field).Tag
            Val = Data(Field).Value
            Type = Data(Field).Type
            If Val And Len(Val) Then
                If (StrComp(Type,"Html",1) = 0) Or (StrComp(Type,"Text",1)
                    = 0) Then
                    ' output text in standard html
                    If Tag And Len(Tag) Then
                        Response.Write("<b>" & Tag & ": </b>")
                    End If
                    Response.Write(Val & "<br>")
                ElseIf StrComp(Type,"Img",1) = 0 Then
                    ' try to create an image
                    ImgElem = MC.Locate(Val)
                    If ImgElem Then
                        ImgElem = MC.Href(ImgElem)
                        If ImgHref Then
                            Response.Write("<img scr=" & ImgHref
                                & "><br>")
                        End If
                    End If
                ElseIf StrComp(Type,"Href",1) = 0 Then

```

```

' write an href
    Response.Write("<a href=" & Val & ">" & Tag & "</a><br>")
    End If
End If

Next
Next
End If
%>

<br>&nbsp; <br><hr>
<b>Note:</b> This page was automatically generated because the correct
scripts could not be found. If this problem persists after
synchronization,
please contact the content provider.
</body>
</html>

```

ANHANG B

MOBILKANAL-SKRIPTINGUMGEBUNG

[0130] Das Mobilkanal-Skriptingmodell basiert auf Active Server Pages (ASP), gemäß Definition in IIS. ASP-Code ist in VBS geschrieben. In Mobilkanälen werden sowohl ASP als auch VBS reduziert, um den Beschränkungen von Windows-CE-Vorrichtungen zu entsprechen. Das angepasste ASP ist auch als Pocket ASP (pASP) bekannt. Zusammen werden pASP und VBS als Mobilkanal-Skriptingumgebung bezeichnet. Die hier angeführten Beschreibungen konzentrieren sich auf die Unterschiede zwischen pASP/VBS für Mobilkanäle und ASP/VBS für Aktivkanäle.

Arten

[0131] Es gibt drei legale Datentypen für Mobilkanalskripting: STRING, NUMERISCH und BOOLEAN. Jedoch wird nur STRING intern unterstützt. Die beiden anderen sind von STRING abgeleitet. String-Symbole können unter Verwendung des zweifachen Anführungszeichens (") spezifiziert sein, um den Ausdruck einzuklammern. Numerische Strings können ohne Anführungszeichen spezifiziert sein. Zahlen können nur aus ganzen Zahlen bestehen, und ihr Wert zwischen -32.768 und 32767 liegen. Boolean-Ausdrücke evaluieren zu 1 für wahr und 0 für falsch. Sie können nicht wie in Visual Basic WAHR oder FALSCH zugeordnet werden. Beispielsweise

Datentyp	Wert	Beschreibung
STRING	"Beispiel-String-Symbol"	
NUMERISCH	Ergebnis=3+4	Das Ergebnis evaluiert zu 7. Doch Ergebnis wird als ein String-Wert gespeichert.
BOOLEAN	(a) 3 = 3, (b) 3 = 5	(a) evaluiert zu 1 und (b) zu 0.

Datenstrukturen

Mobilkanäle unterstützen folgende Datenstrukturen.

Datenstruktur	Beschreibung
Variable	Elementare Datenstruktur der einfachen vorstehend präsentierten Datentypen. Variablennamen sind alphanumerisch und müssen mit einem Alphazeichen beginnen. Das Unterstreichzeichen kann mit Ausnahme des führenden Zeichens verwendet werden. Variablennamen sollten kurz sein, um Speicher zu sparen und können in jedem Fall nicht länger als 255 Zeichen sein.
Feld	Eine geordnete Sammlungen mit numerischen Tasten. Der Index zählt von Null (0). Beispielsweise Ergebnis = a(0) + a(1). Das Verfahren, Array.Count führt die Gesamtanzahl von Elementen in dem Feld zurück.

Schlüsselwörter

[0132] Die folgenden Schlüsselwörter sind reserviert und können nicht als Variablennamen verwendet werden:

If, Then, Else, Elseif, End If
 For, Next, Do While, Loop, Exit For, Exit While
 Set, Response, Request, MobileChannels
 Now, LocDate, Len, Mid, Split, Asc, Chr, StrComp, Random

Kommentare

[0133] Kommentare werden mit dem einfachen Anführungszeichen ()(dies ist ein Beispiel-Kommentar.) begonnen und können überall auf einer Zeile erscheinen. REM von VBS ist nicht für Mobikanal-Skripting unterstützt. Das folgende ist ein Beispiel für einen Kommentar.

Operatoren und Präzedenz

Operator	Typ	Präzedenz	Beschreibung
.	Numerisch	1	Multiplikation
/	Numerisch	1	Division
Mod	Numerisch	1	Modulo-Division
+	Numerisch	2	Addition
-	Numerisch	2	Subtraktion
&	String	2	Konkatenation
<	Boolean	3	Weniger als
<=	Boolean	3	Weniger als oder gleich
>	Boolean	3	Größer als
>=	Boolean	3	Größer als oder gleich
=	Boolean	3	Gleich
<>	Boolean	6	Nicht gleich
And	Boolean	4	Logisches UND
Or	Boolean	4	Logisches ODER
Not	Boolean	5	Logisches NICHT

[0134] Ausdrücke werden gemäß der Operatoren-Präzedenz evaluiert. Operatoren von höherer Präzedenz (wobei 1 die höchste ist) werden zuerst evaluiert. Operatoren des gleichen Pegels werden von links nach rechts evaluiert. Die Präzedenz kann übergangen werden, indem Klammern verwendet werden, welche auch ineinander angeordnet sein können. Die innerste Klammer wird zuerst evaluiert.

[0135] Anders als in VBS werden alle Ausdrücke innerhalb einer Aussage immer evaluiert. In dem nachstehenden Beispiel werden, wenn arr.count nicht größer als null ist, arr(1) und arr(2) evaluiert, und die Verweise auf arr(j) führen zu einem Fehler.

```
If arr.count >0 und arr(1) = "foo" then
    arr(2) = "bar"
End If
```

[0136] Wenn der erste logische Ausdruck falsch ist, sind die sich ergebenden Ausdrücke ungültig. Die korrekte Realisierung sollte wie folgt sein:

```
If arr.count > 0 then
    If arr(1) = "foo" then
        arr(2) = "bar"
    End If
End If
```

Umschaltende Sonderzeichen

[0137] Sonderzeichen, wie beispielsweise das doppelte Anführungszeichen, können innerhalb eines String-Symbols "umgeschaltet" werden, indem ihnen das Back-Slash-Zeichen (\) vorangestellt wird. Das Back-Slash-Zeichen kann in einem String eingeschlossen werden indem es selbst umgeschaltet wird. Zum Beispiel:

"Dieser String enthält\"doppelte Anführungszeichen\"."

"Dieser String enthält Back-Slashes wie in einem Dateipfad: \\c:\\windows."

Aussagen

[0138] In der Mobilkanal-Skriptingumgebung gibt es fünf Klassen von Aussagen:

Zuordnung

[0139] Die Zuordnungsangabe weist die folgende Form auf:
<Variable> = <Ausdruck>

Konditional

[0140] Die If-Aussage liefert einen konditionalen Regelfluss. Der End-If-Teil ist erforderlich. Die Aussagen nach einem logischen Ausdruck werden nicht evaluiert, außer der logische Ausdruck evaluiert zu wahr (1). Die konditionale Aussage kann eine der folgenden Formen aufweisen:

```
If <logischer Ausdruck> Then
    <Aussage>
End If
```

oder

```
If <logischer Ausdruck> Then
    <Aussage1>
Else
    <Aussage2>
End If
```

oder

```
If <logischer Ausdruck1> Then
    <Aussage1>
ElseIf <logischer Ausdruck2> Then
    <Aussage2>
End If
```

oder

```
If <logischer Ausdruck1> Then
    <Aussage1>
ElseIf <logischer Ausdruck2> Then
    <Aussage2>
Else
    <Aussage3>
End If
```

Schleife

[0141] Es gibt zwei Arten von Schleifenaussagen: For/Next und Do/While:

Die For-Schleife wiederholt sich durch die Schleife, indem die Variable anfänglich auf den numerischen Ausdruck1 gesetzt und dieser Wert bei jedem Durchgang durch die Schleife um die Step-Menge (Ausdruck 3) erhöht wird. Wenn die optionale Step-Klausel weggelassen wird, wird die Standardklausel von Step 1 aufgerufen. Die Schleife endet, wenn die Variable einen Wert erreicht, der größer ist als Ausdruck2.

```
For <Variable>=<Ausdruck1> To <Ausdruck2> [Step <Ausdruck3>]
    <Aussage1>
    (Exit For) 'Optional
    <Aussage2>
Next
```

[0142] Die Do-While-Schleife fährt fort, bis der logische Ausdruck logAusdruck falsch (0) wird. Die Exit-Aus-

sagen liefern einen Weg, eine Schleife zu beenden ohne die normalen Beendigungskriterien zu erfüllen. Wenn auf Exit getroffen wird, bricht die Schleife, und die Ausführung wird an der unmittelbar auf die Schleife folgenden Aussage wieder aufgenommen. Exit wird gewöhnlich in Verbindung mit einer Konditional-Aussage verwendet.

```
Do While <logAusdruck>
    <Aussage1>
    (Exit While) 'Optional
    <Aussage2>
```

Loop

Aktivserver

[0143] Aktivserver-Aussagen beziehen sich auf die Verfahren von pASP-Objekten, wie beispielsweise Response und Request. Die Response.Write-Aussage führt eine Ausgabe zu dem HTML-Strom zurück, beispielsweise

Response.Write(" Klicken Sie hier für Sport ").

[0144] Die Mobilkanal-Skriptingumgebung exponiert bestimmte Servervariablen. Die Request.ServerVariables-Aussage kann verwendet werden, um die Servervariablen anzufragen. Es nimmt einen Namens-Stringausdruck und führt einen mit dem Namen in Verbindung stehenden Wert-Stringausdruck zurück.

```
newURL = Request.ServerVariables("URL")
erhält also die Wurzel-URL für den Kanal der Seite. Und
platStr = Request.ServerVariables("Platform")
führt den Platform-String als einen der folgenden zurück:
```

String	Platform
"WIN32_CE"	Windows CE
"WIN32_WINDOWS"	Windows 95/Windows 98
"WIN32_NT"	Windows NT

[0145] Auf ähnliche Weise führt die Request.QueryString-Aussage den Wert eines spezifizierten Arguments zurück, welches zu der Seite als Teil der URL geleitet wird. Wenn beispielsweise die URL für eine Seite als "MCTP://msnbc/ch2 ? city = seattle" bezeichnet wird, so weist die Aussage theCity = Request.QueryString("city") der theCity-Variablen Seattle zu.

Set

[0146] Die Set-Aussage weist einem Fall eines Objekts eine Variable zu. Die Mobilkanal-Skriptingumgebung unterstützt jedoch nur das MobileChannels.Utilities-Pseudoobjekt. Somit ist die einzige Verwendung der Set-Aussage, ein MobileChannels.Utilities-Objekt zu schaffen und weist es einer Fallvariablen zu:

```
Set mc_variable = Server.Create("MobileChannels.Utilities")
Zeilenumbrüche werden bei der Evaluation einer Aussage ignoriert. Somit können Aussagen sich über mehr
als eine Zeile erstrecken. Das Aussagen-Fortsetzungszeichen ("_") wird empfohlen, doch nicht verbindlich.
Zum Beispiel
```

```
MyVar = "Dies ist ein Beispiel für" & _
"eine Aussage, die sich" &
"über mehrere Zeilen erstreckt." & MyVar
```

Funktionen

[0147] Die folgenden Funktionen werden in der Mobilkanal-Skriptingumgebung exponiert.

Now

[0148] Führt das momentane Datum und die Zeit zurück und nimmt kein Argument an. Z.B.: Response.Write("Das heutige Datum ist " & Now).

LocDate

[0149] Führt das Datum unter Verwendung der momentanen Regionaleinstellungen zurück, um das Datum zu formatieren. Z.B.:
 Response.Write("Datum:" & LocDate)

Len (<string>)

[0150] Führt die Länge eines Strings zurück. Z.B.:
 Len ("Hallo?")
 führt 6 zurück.

Mid(aStringExpression, startNumExpression, [length])

[0151] Führt einen Sub-String eines existierenden Strings zurück. Der resultierende Sub-String ist length Zeichen lang und beginnt bei der Start-Zeichenzahl (zählend von eins, nicht null) in dem ursprünglichen String-Ausdruck. Z.B.:
 Foo = Mid("This is my string", 9, 2).
 Foo wird nun auf "my" gesetzt.

Split(aStringExpression, delimiterStringExpression)

[0152] Parst einen String in Substrings basierend auf einer spezifizierten Abgrenzung. Das Ergebnis wird als ein Stringfeld zurückgeführt. Z.B..
 Names = Split("Bob;Fred;Joe;", ";")
 führt zu folgenden Sub-Strings:
 Names (0) = "Bob"
 Names (1) = "Fred"
 Names (2) = "Joe"
 Names (3) = " "

Asc(aStringExpression)

[0153] Wandelt einen Zeichen-String in seinen numerischen ASCII-Wert um und führt einen numerischen Ausdruck zurück. Wenn die aStringExpression länger als ein Zeichen ist, führt die Funktion nur den ASCII-Wert des ersten Zeichens zurück.

Chr(numericExpression)

[0154] Wandelt einen numerischen ASCII-Wert in das zugeordnete Zeichen um und führt einen String-Ausdruck mit einer Länge von einem Zeichen zurück. Um beispielsweise einen String mit nur einem Neuzeilenzeichen zu erzeugen, wird folgendes verwendet:
 str = Chr(10)

StrComp (S1, S2[,Compare])

[0155] Diese Funktion vergleicht zwei Strings, S1 und S2, wobei optional der Vergleichsmodus, Compare, spezifiziert wird. Das Compare-Argument kann 0 oder 1 sein. Wenn Compare weggelassen wird, wird ein binärer Vergleich durchgeführt.

[0156] Diese Funktion führt einen der folgenden Werte zurück:

Bedingung	Rückführwert
S1 ist kleiner als S2	-1
S1 ist gleich S2	0
S1 ist größer als S2	1

Random(range)

[0157] Die Funktion erzeugt eine willkürliche Zahl im Bereich 0 bis eins weniger als range. Z.B.:

num = Random(10)
erzeugt willkürliche Zahlen von 0 bis 9 einschließlich.

Mobilkanal-Skriptingobjekt

[0158] MobileChannels.Utilities ist ein Pseudoobjekt in der Mobilkanal-Skriptingumgebung, welches Unterstützung für Navigation und Manipulation von Objekten innerhalb einer CDF-Datei liefert. Das Utilities-Objekt liefert eine Anzahl von Verfahren für Mobilkanal-Skripting. Diese Verfahren sind in der nachfolgenden Tabelle zusammengefasst.

Verfahren	Beschreibung
Data	Liest einen Datenblock von einem Datenelement
Debug	Emitteert Fehlerbehebungs-Ausgangssignale um die Skriptenentwicklung zu unterstützen
Href	Führt die HREF eines Elements zurück
HrefExists	Führt wahr zurück, wenn ein Element im Zwischenspeicher existiert
IsSubscribed	Führt den geordneten Status eines Kanals/Subkanals zurück
IsUnread	Führt den gelesenen/ungelesenen Status eines Elements oder Kanals/Subkanals zurück
LibraryCall	Greift auf eine spezielle DLL-Funktion zu
Locate	Springt zu einer spezifizierten ID innerhalb der CDF
Navigate	Überquert eine CDF-Datei
Tag	Führt das Kennzeichen eines Elements in einer CDF-Datei zurück
Title	Führt den Titel eines Elements zurück
Value	Führt den Wert eines Elements in einer CDF-Datei zurück

[0159] Zur Nutzung dieser Dienste muss das Utilities-Objekt zunächst unter Verwendung der Set-Funktion wie folgt angepasst werden:

```
Set MC = Server.Create("MobileChannels.Utilities")
```

[0160] MC wird nachfolgend als Abkürzung für das MobileChannels.Utilities-Skripting-Objekt verwendet, jedoch kann das Objekt jedem Variablennamen zugeordnet werden

NAVIGIERVERFAHREN DES UTILITIES-OBJEKTS

[0161] Der MC.Navigate()-Befehl ist ein starkes, häufig verwendetes und bei weitem das bedeutendste Verfahren in pASP. Es ist dazu ausgelegt, die Untersuchung der Struktur eines Mobilkanals, wie in CDF dargestellt, beim Ablauf zu unterstützen. Um das Verhalten dieses Befehls zu verstehen, ist eine kurze Beschreibung des Hintergrundes und der Terminologie hilfreich.

[0162] Der Grund-Operand des Navigate-Befehls ist ein Element, welches die kleinste Informationseinheit in einer CDF-Datei ist. Jedes Element weist ein Kennzeichen (tag) auf, sowie optional einen Wert (value). Die MC.Tag()- und MC.Value()-Verfahren von pASP können verwendet werden, um diese Strings für jedes Element abzuholen. Die Elemente werden in einer Baumstruktur wie durch XML spezifiziert angeordnet, und die CDF-Datei wird geparst. Der Navigate-Befehl erlaubt es, sich zu bestimmten Elementen innerhalb des Baumes zu bewegen, und sich zwischen Elementen mit bestimmten Beziehungen zu bewegen. Diese Information kann sehr hilfreich für die Kanalskripten sein, welche CDF verwenden, um dynamisch die HTML-Seiten für den Kanal bei Betriebszeiten zu erzeugen.

[0163] Die nachstehenden Beschreibungen werden sich häufig auf die Beispiel-CDF-Datei und ihren zugeordneten Parse-Baum beziehen, welche am Ende dieses Dokumentes bereitgestellt sind. Der Parse-Baum zeigt die interne Repräsentation der Beispiel-CDF-Datei. Jede Zeile des Parse-Baumes entspricht einem Element, und alle beginnen mit dem Kennzeichen für das Element. Die eingerückten Elemente sind Kinder ihrer weniger eingerückten Eltern. Elemente mit dem selben Einrückungsgrad sind Geschwister. In der CDF-Datei ist beispielsweise das BASE-Element ein Kind des Spitzenpegel-CHANNEL-Elements. Das erste HREF-Element ist ein Geschwister des BASE-Elements. Das INTERVALTIME-Element ist ein Kind des SCHEDULE-Element.

ments.

[0164] Von vielen Elementen wird angenommen, dass sie einen Standardwert aufweisen. Diese sind in dem Parse-Baum durch einen "[string]"-Ausdruck nachfolgend auf das Kennzeichen des Elements angezeigt. Der Standardwert wird im folgenden Schema bestimmt. Zuerst ist, wenn das betrachtete Element einen ihm direkt zugehörigen String aufweist, der String der Standardwert. Als nächstes wird, wenn ein Kinder-VALUE-Element existiert, der Wert des Kindes der Standard. Wenn kein VALUE-Element bereitgestellt ist, jedoch ein Kinder-HREF-Element gefunden wird, sein Wert der Standardwert. Kann nichts von diesen gefunden werden, ist der Wert VALUE leer. So ist beispielsweise der Wert des ersten ID-Kennzeichens ein direkter String, das TITLE-Kennzeichen weist ein ausdrückliches VALUE-Element auf, also wird dieses verwendet, und der Wert des ersten LOGO-Kennzeichens ist sein HREF. Das SCHEDULE-Kennzeichen weist keinen direkten String, VALUE- oder HREF-Kinder auf, also ist sein Wert leer.

[0165] Die Navigate-Funktion weist die folgende Syntax auf:
 NewElem = MC.Navigate(StartElem, NavAction,[,Match])

[0166] Die Funktion führt ein neues Element zurück, oder aber 0, wenn der Befehl das spezifizierte Element nicht finden konnte. Dieser Rückführwert kann unter Verwendung von Standard-VBS-Vergleichen getestet werden, beispielsweise folgende:

```
IF NOT NewElem THEN
    ' not found
END IF
```

[0167] Der StartElem-Parameter ist das Startelement, von welchem relative Bewegungsbefehle zu basieren sind. Wenn der absolute Bewegungsbefehl "Jump" verwendet wird, muss " " für den StartElem-Parameter verwendet werden. Doch in allen anderen Fällen muss ein gültiges Element von einem vorangehendem Navigate()-Befehl zurückgeführt werden.

[0168] Der NavAction-Parameter muss einer der folgenden Strings sein:

"Jump"

[0169] Die "Jump"-Aktion ist der erste Befehl, der verwendet wird um ein Startelement zu erhalten. Er entspricht dem MC.Locate()-Befehl (s. unten). Der StartElem-Parameter muss ein leerer String sein. Die "Jump"-Aktion navigiert unmittelbar zu einem bestimmten Element in der CDF, wie durch die gelieferte ID spezifiziert. In der folgenden Aussage beispielsweise springt
 NewElem = MC.Navigate("", "Jump", "D1")
 zum ersten Datenelement in der Beispiel-CDF-Datei. Das NewElem wird das ITEM-Elternelement zu dem ID-Element ("D1", auf etwa halber Höhe in der Beispiel-CDF-Datei) sein.

"First"

[0170] Die "First"-Aktion bewegt sich zum ersten Element auf einem vorgegebenen Pegel. Von dem ID-Element des ersten LOGO-Elements in der Beispiel-CDF-Datei bewegt sich beispielsweise eine "First"-Aktion zu dem STYLE-Kennzeichen dieses LOGO. Praktischere Szenarien sollen diese Aktion verwenden, um zurück zum Beginn der Liste von ITEMS unter einem Subkanal zu gehen.
 NewElem = MC.Navigate(StartElem, "First")

"Out"

[0171] Die "Out"-Aktion bewegt sich zu dem Elternelement des momentanen Elements oder zur linken Einrückung in dem Parse-Baumdiagramm. Beispielsweise führt die "Out"-Aktion zu der Bewegung von dem TITLE-Element der Beispiel-CDF zu dem Spitzenpegel-Element.
 NewElem = MC.Navigate(StartElem, "Out")

"In"

[0172] Die "In"-Aktion bewegt sich zu dem ersten Kinder-Element unterhalb des momentanen Elements. Beispielsweise führt die "In"-Aktion zu einer Bewegung von dem ersten USAGE-Element in der Beispiel-CDF-Da-

tei zu dem VALUE-Element.

```
NewElem = MC.Navigate(StartElem,"In")
```

"Prev"

[0173] Die "Prev"-Aktion bewegt sich zu dem Element auf dem gleichen Pegel, welches dem momentanen Element unmittelbar vorangeht. Findet sie kein vorangehendes Element auf dem selben Pegel, so kehrt sie zu 0 zurück; sie rückt nicht zu dem Elternelement aus. Beispielsweise führt die "Prev"-Aktion zu einer Bewegung von dem BASE-Element in der Beispiel-CDF-Datei zu dem HREF-Element rechts vor ihm. Das nochmalige Aufrufen von "Prev" führt zu 0 zurück, da keine weiteren Geschwister auf diesem Pegel existieren.

```
NewElem = MC.Navigate(StartElem,"Prev")
```

"Next"

[0174] Die "Next"-Aktion bewegt sich zu dem nächsten Element auf dem selben Pegel. Wie bei der "Prev"-Aktion kehrt sie, wenn sie keine derartigen Geschwisterelemente findet, zu null zurück. Beispielsweise führt die "Next"-Aktion zu einer Bewegung von dem ersten LOGO-Kennzeichen in der Beispiel-CDF-Datei zu dem zweiten LOGO-Element.

```
NewElem = MC.Navigate(StartElem,"Next")
```

"Match"

[0175] Die "Match"-Aktion versucht, ein Geschwisterelement mit einem Kennzeichen zu finden, welches zu dem spezifizierten Match-String passt. Sie überquert so viele Geschwister wie sie benötigt, bis sie entweder ein Gegenstück findet oder keine weiteren Geschwister finden kann. Wenn die "Match"-Aktion von einem passenden Element startet, führt sie einfach das momentane Element zurück. Um über das momentane Element hinaus zu passen muss die "Match"-Aktion auf eine "Next"-Aktion folgen.

```
NewElem = MC.Navigate(StartElem,"Match","TagToMatch")
```

"InMatch"

[0176] Die "InMatch"-Aktion entspricht dem vorstehenden "Match" mit der Ausnahme, dass sie ihre Suche beim ersten Kind des momentanen Elements beginnt. Dies kann nützlich bei der Suche nach bestimmten Subkennzeichen sein, welche das einschließende Element modifizieren. Beispielsweise suchen die folgenden Aussagen

```
UsageElem = MC.Navigate( StartElem,"InMatch","USAGE")
```

```
If UsageElem Then
```

```
    UsageVal = MC.Val( UsageElem )
```

```
    'test for specific usage...
```

```
End If
```

nach dem USAGE-Kennzeichen für ein bestimmtes Element.

[0177] Die einzigen Aktionen, die den optionalen dritten Parameter verwenden, sind "Match" und "InMatch".

ANDERE VERFAHREN DES UTILITIES-OBJEKTS Tag

[0178] Dieses Verfahren führt den Kennzeichentag)-Namen eines Elements zurück.

```
tagString = MC.Tag(elementID)
```

Value

[0179] Dieses Verfahren führt den Wert eines Elements zurück.

```
valString = MC.Value(elementID)
```

Data

[0180] Dieses Verfahren erhält Daten von einer Mobilkanal-Datendatei und führt ein Feld von Na-

men-Wert-Paaren, basierend auf der momentanen Position und der spezifizierten Blockzahl, zurück. Die Namen sind die Feldnamen, wie spezifiziert in einer ITEMFOR-MAT-Aussage, und die Werte sind die Datenelemente (Zeilen) wie von der Datendatei abgeholt. In dem nachfolgenden Beispiel ist dataltems ein Feld zum Halten der Datenelemente,

```
dataltems = MC.Data(elementID,blockNum)
```

wobei elementID die momentane Position innerhalb der CDF-Datei ist, beispielsweise das ID-Element für die MCD-Datei, und block-Num ist die Blockzahl innerhalb der Datei. Blöcke beginnen mit null. Also ist der erste Wiederholungsblock, wenn vorhanden, immer Blockzahl eins (selbst wenn kein Anfangsblock existiert). Das resultierende Feld, dataltems, enthält ein Element für jedes Element (Zeile) innerhalb des Blocks. Die Elemente in einem Block zählt von null.

[0181] Jedes Datenelement ist tatsächlich ein Objekt, welches die Tag-, Type- und Value-Verfahren unterstützen, um seine eigenen Eigenschaften zu exponieren.

Tag

[0182] dataltems(index).Tag führt den Feldnamen für die Feldposition des Index zurück, wie in dem <ITEMFORMAT>-Element erklärt.

Value

[0183] dataltems(index).Value führt den Wert des Feldes für die Feldposition des Index zurück.

Type

[0184] dataltems(index).Type führt den Typ wie spezifiziert in der <ITEMFORMAT>-Aussage zurück. Ist kein Typ gelistet oder fehlt das <ITEMFORMAT>-Kennzeichen, so führt das Type-Verfahren "HTML" zurück. Andere Typen schließen "TEXT", "IMG" und "HREF" ein.

Typ	Beschreibung
HTML	Das Zeilenelement ist HTML-formatierter Inhalt. Dies ist der Standardtyp.
HREF	Die Zeile ist eine URL, (entweder http:// oder mctp://).
IMG	Die Zeile enthält die ID eines Bildelements in der CDF-Datei.
TEXT	wie HTML.

Locate

[0185] Die Funktion weist die folgende Form auf:

```
newElem = MC.Locate("ID")
```

und ist eine abgekürzte Form für die "Jump"-Aktion des Navigate-Verfahrens:

```
newElem = MC.Navigate("", "Jump", "ID")
```

LibraryCall

[0186] Diese Funktion erlaubt einem Skript Zugriff auf eine Maß-DLL, um Funktionen durchzuführen, welche durch Standard-Skripting nicht verfügbar sind. Das Verfahren weist die folgende Form auf:

```
Result = MC.LibraryCall(LibName,FuncName[,param]*)
```

[0187] Zunächst untersucht dieses Verfahren die Sicherheit, um sicherzustellen, dass die DLL angemessen für Zugriff über pASP-Skripting registriert wurde. Eine zugängliche DLL muss einen Registereintrag in \HKLM\Software\Microsoft\Mobile Channels\Components aufweisen, welcher dem Namen der DLL entspricht.

[0188] Dann lädt das LibraryCall-Verfahren dynamisch die spezifizierte DLL durch Aufrufen der GetProcAddress-Funktion zum Suchen der spezifizierten Funktion. Jegliche zusätzlichen Parameter werden dann aufgereiht, ehe sie an die DLL-Funktion weitergeleitet werden. Bis zu 8 optionale Parameter können passiert werden.

[0189] Die DLL-Funktion muss einen LPWSTR-Wert zurückführen. Wenn der Rückführwert NULL beträgt,

führt LibraryCall null (0) zurück. Andernfalls führt sie den String selbst als Standard-pVBS-Stringwert zurück.

Debug(Mesg)

[0190] Das Debug-Verfahren ermöglicht es, dass ein Fehlerbehebungs-String während der Ausführung des Skripts ausgestellt wird. Dies kann während der Entwicklung nützlich sein, um die Untersuchung des Programmflusses zu unterstützen. Die Fehlerbehebungs-Nachrichten erscheinen im Konsolenfenster jedes angeschlossenen Debuggers, ähnlich wie der OutputDebugString API.

[0191] Die Funktion führt keinen Wert zurück.

HrefExists(Href)

[0192] Dieses Verfahren testet zur Bestimmung, ob die spezifizierte URL im Zwischenspeicher gefunden werden kann. Dies ermöglicht es, dass ein Skript fehlende Bilder, Datenelemente oder andere Komponenten, die durch das Skript benötigt werden, vorsichtig handhabt. Die URL muss eine voll qualifizierte URL im http-Stil sein.

[0193] Dieses Verfahren führt 1 zurück, wenn es im Zwischenspeicher gefunden wurde, andernfalls 0.

Href(Elem)

[0194] Dieses Verfahren führt die volle URL für das spezifizierte Element zurück, wenn es in der CDF-Datei spezifiziert ist. Es führt 0 zurück, wenn keine URL gefunden werden kann.

IsSubscribed(ChanElem)

[0195] Dies testet, um zu sehen, ob das spezifizierte Kanal- oder Subkanalelement momentan vom Benutzer angefordert ist. Es führt 1 zurück, wenn der Kanal angefordert ist, oder 0, wenn er nicht gefunden wird oder nicht angefordert ist.

[0196] N.B.: Dies funktioniert nicht bei Elementen oder Kanälen. Weiterhin führt es immer 1 zurück, wenn es auf dem Tischcomputer läuft (in IE4).

Title

[0197] Dieses Verfahren weist die folgende Form auf:

titleString = MC.Title(ElemString)

und versucht, den Titel eines gegebenen Elements auf folgende Weise zu entziffern:

Wenn ein explizites TITLE-Kennzeichen für dieses Element existiert, wird der Wert davon zurückgeführt,

Wenn es ein .mcd-Datenelement mit einem ITEMFORMAT ist, welches ein TITLE-Feld spezifiziert, wird das Datenelement geöffnet und der Titel daraus extrahiert,

Wenn ein ID-Element bereitgestellt ist, wird sein Wert zurückgeführt,

Ansonsten wird NULL zurückgeführt.

[0198] Es versteht sich, dass dieses Verfahren kein Datenelement als gelesen ("Read") markiert, wenn es den Titel abholt. Dies unterscheidet sich von der Verwendung von Navigate, um den Titel zu erhalten. Das letztere Verfahren markiert das Element als gelesen ("Read"), selbst wenn der Benutzer es nicht tatsächlich gesehen hat.

IsUnread

[0199] Dieses Verfahren führt ein Boolean-Zeichen zurück, welches anzeigt, ob das zugeordnete Element oder der Kanal gelesen wurde.

newContent = MC.IsUnread(Elem)

[0200] Die Funktion führt einen Nicht-Null-Wert zurück, wenn es direkt auf ein ungelesenes Element gerufen wird. Wenn es auf einen Subkanal gerufen wird, führt es Nicht-Null zurück, falls irgendwelche Elemente oder andere Subkanäle innerhalb des Subkanals nicht gelesen wurden.

SetUnread

[0201] Dieses Verfahren setzt den Gelesen-/Ungelesen-Status für ein Element und führt keinen Wert zurück. Und es weist das folgende Format auf:
SetUnread(Elem[,Flag])

[0202] Der Elem-Parameter sollte ein gültiges Element von einem früheren Navigate()- oder Locate()-Aufruf sein. Der optionale Flag-Parameter ist ein Boolean-Zeichen, welches zur Markierung des Status von Elem: 0 für "ungelesen" und 1 für "gelesen" dient. Der Standardwert von Flag ist "ungelesen".

[0203] N.B.: Aufgrund einer Einschränkung der Implementierungen von Version 1.0 der Mobilkanäle werden Bildelemente nicht automatisch als "gelesen" markiert (wie dies bei MDC-Elementen der Fall ist). Dies führt dazu, dass das Bild als "ungelesen" markiert bleibt, obgleich es gelesen wurde. Weiterhin zeigen alle Eltern-Subkanäle ebenfalls als "ungelesen", so lange irgendwelche Bilder darin ungelesen sind. Um dieser Situation abzuweichen, sollte der Skriptenautor manuell jedes Bild als "ungelesen" markieren, wann immer es angezeigt wird. Die SetUnread()-Utility ist die richtige Weise, um dies zu erreichen.

Kanal-Browser und Active Desktop HTML Extensionen

[0204] Mehrere HTML-Extensionen liefern eine zusätzliche Funktionalität zum Schreiben fortgeschrittenerer Skripten für den Active Desktop und zum Regeln von Seiten-Aktualisierungen im Kanal-Browser.

Anwendungs-Links

[0205] Windows CE Active Desktop unterstützt eine spezielle HTML Href zum Starten einer Anwendung von einem Hyperlink. Das Format ist folgendes:

Launch Text

appname ist der Name der zu startenden Anwendung, wenn der Link angeklickt wird.

[0206] Die Anwendung muss registriert worden sein, indem ein Wert selben Namens als .exe in dem Register bei \HKLM\Software\Microsoft\Mobile Channels\Components.META Tags angeordnet wird.

META-Kennzeichen

[0207] Der Kanal-Browser und Active Desktop erkennen die nachfolgenden speziellen META-Kennzeichen. Die Eingliederung dieser META-Kennzeichen in den Anfangsblock einer Seite, entweder direkt oder über Skripting kann dazu führen, dass die Seite automatisch auf eine bestimmte Art und Weise gehandhabt oder aktualisiert wird. Es versteht sich, dass diese META-Kennzeichen (mit Ausnahme von Refresh) durch IE4 ignoriert werden.

[0208] Die META-Kennzeichen sind in der nachstehenden Tabelle zusammengefasst.

Http_Equiv	Beschreibung	Unterstützung
Notify	Fangen von Zwischenspeicher oder Datenbank-Updates	Active Desktop und Kanal-Browser
Refresh	Neuladen nach Zeitintervall	Active Desktop
LaunchApp	Ausführen einer Anwendung für Desktop-Komponente	Active Desktop
Autosize	Regelung der Bildskalierung	Kanal-Browser

[0209] Nachstehend einige ausführliche Beschreibungen jedes Kennzeichens.

Notify

[0210] Dieses META-Kennzeichen ermöglicht die automatische Aktualisierung einer Seite, wenn ein Update zu einer bestimmten Datenbank existiert oder wenn ein bestimmtes Element im Zwischenspeicher aktualisiert wird. Dies kann verwendet werden, um eine Seite automatisch zu regenerieren, wenn eine neue Version davon

(oder eine ihrer Komponenten) über Sync oder einen anderen Mechanismus hereinkommt. Die beiden Formen dieses META-Kennzeichens sind folgende:

```
<META HTTP-EQUIV="Notify"
  CONTENT="DBUPDATE=[DBname];URL=[RefreshUrl]">
<META HTTP-EQUIV="Notify"
  CONTENT="CACHEUPDATE=[WatchUrl];URL=[RefreshUrl]">
```

[0211] DBname ist der Name der für Updates zu überwachenden Datenbank.

[0212] WatchUrl ist die URL eines für Zwischenspeicher-Updates darauf zu beobachtenden Elements.

[0213] RefreshUrl ist die zu ladende URL, wenn ein Update detektiert wird.

Refresh

[0214] Dieses META-Kennzeichen verursacht, dass eine Seite nach einem spezifizierten Zeitintervall automatisch neu geladen wird. Die Form ist folgende:

```
<META HTTP-EQUIV="Refresh"CONTENT="[secs]; URL=[RefreshUrl]">
```

secs setzt die Anzahl von Sekunden, bis die Seite neu geladen wird.

[0215] RefreshUrl ist die nach dem spezifizierten Zeitintervall zu ladende URL.

LaunchApp

[0216] Dieses META-Kennzeichen ermöglicht es, dass eine Anwendung durch Klicken auf den Anfangsblock einer Komponente des Active Desktop auf der Vorrichtung gestartet wird. Die Form ist folgende:

```
<META HTTP-EQUIV="LaunchApp"CONTENT="[appname][?params]">
```

appname ist der Name des zu startenden ausführbaren Programms.

params ist eine optionale durch Komma abgetrennte Liste von params, welche nach Aufrufung an die Anwendung weiterzuleiten sind.

[0217] Die Anwendung muss durch Anordnen eines Wertes des selben Namens wie die .exe in \HKLM\Software\Microsoft\Mobile Channels\Components registriert worden sein.

Autosize

[0218] Dieses META-Kennzeichen ermöglicht es, dass das Standardbild-Skalierungsverhalten für eine bestimmte Seite außer Kraft gesetzt wird. Die HTML-Regelung wird standardmäßig versuchen, Bilder für die Anzeige auf dem kleineren Bildschirm zu skalieren. Wenn dieses META jedoch in dem Anfangsblock der Seite spezifiziert ist, werden die Bilder mit voller Größe angezeigt, wodurch wenn nötig Scrollmenüs erscheinen. Die Form ist folgende.

```
<META HTTP-EQUIV="Autosize" CONTENT="Off">
```

[0219] Es versteht sich, dass, da der Standardwert immer "Ein" ist, kein anderer Wert in dem CONTENT-Feld benötigt wird.

Beispiel-CDF-Datei

```

<?XML version="1.0"?>
<CHANNEL
  HREF="mctp://mySite/34droad/34droad.cdf "
  BASE="http://mySite/" ID="34droad">
  <SELF HREF="http://mySite/34droad/34droad.cdf" />
  <SCHEDULE> <INTERVALTIME MIN="40"/> </SCHEDULE>
  <USAGE VALUE="MobileChannel"/>
  <TTITLE>3 4 D Road</>
  <ABSTRACT>Things to think about while you're away...</>
  <LOGO STYLE="IMAGE" HREF="34droad/34logo.gif" ID="LOGO"/>
  <LOGO STYLE="ICON" HREF="34droad/34icon.gif" ID="ICON"/>
  <CHANSRIPT VALUE="SS"/>
  <ITEMSCRIPT VALUE="SS"/>
  <ITEM HREF="34droad/34.mcs" ID="SS">
    <ABSTRACT>Things to think about while you're away...</>
  </ITEM>
  <ITEM HREF="cgi-bin/deep1.mcd?1" ID="D1">
    <USAGE VALUE="MobileChannel"/>
    <LOG VALUE="document:view"/>
  </ITEM>
  <ITEM HREF="cgi-bin/deep1.mcd?2" ID="D2">
    <USAGE VALUE="MobileChannel"/>
    <LOG VALUE="document:view"/>
  </ITEM>
  <ITEM HREF="cgi-bin/deep1.mcd?3" ID="D3">
    <USAGE VALUE="MobileChannel"/>
    <LOG VALUE="document:view"/>
  </ITEM>
  <ITEM HREF="34droad/34logo.gif" ID="LOGO">
    <USAGE VALUE="None"/>
  </ITEM>
  <ITEM HREF="34droad/34icon.gif" ID="ICON">
    <USAGE VALUE="None"/>
  </ITEM>
  <ITEM HREF="34droad/34main.gif" ID="MGIF">
    <USAGE VALUE="None"/>
    <LOG VALUE="document:view"/>
  </ITEM>
</CHANNEL>

```

Parse-Baum der Beispiel-CDF-Datei

CHANNEL = mctp://mySite/34droad/34droad.cdf
HREF = mctp://mySite/34droad/34droad.cdf
BASE = http://mySite/
ID = 34droad
SELF = http://mySite/34droad/34droad.cdf
HREF = http://mySite/34droad/34droad.cdf
SCHEDULE
INTERVALTIME
MIN = 40
USAGE = MobileChannel
VALUE = MobileChannel
TITLE = 3 4 D Road
VALUE = 3 4 D Road
ABSTRACT = Things to think about while you're away...
VALUE = Things to think about while you're away...
LOGO = 34droad/34logo.gif
STYLE = IMAGE
HREF = 34droad/34logo.gif
ID = LOGO
LOGO = 34droad/34icon.gif
STYLE = ICON
HREF = 34droad/34icon.gif
ID = ICON
CHANSRIPT = SS
VALUE = SS
ITEMSCRIPT = SS
VALUE = SS
ITEM = 34droad/34.mcs
HREF = 34droad/34.mcs
ID = SS
ABSTRACT = Things to think about while you're away...
VALUE = Things to think about while you're away...
ITEM = cgi-bin/deep1.mcd?1
HREF = cgi-bin/deep1.mcd?1
ID = D1
USAGE = MobileChannel
VALUE = MobileChannel
LOG = document:view
VALUE = document:view

```

ITEM = cgi-bin/deep1.mcd?2
  HREF = cgi-bin/deep1.mcd?2
  ID = D2
  USAGE = MobileChannel
    VALUE = MobileChannel
  LOG = document:view
    VALUE = document:view
ITEM = cgi-bin/deep1.mcd?3
  HREF = cgi-bin/deep1.mcd?3
  ID = D3
  USAGE = MobileChannel
    VALUE = MobileChannel
  LOG = document:view
    VALUE = document:view
ITEM = 34droad/34logo.gif
  HREF = 34droad/34logo.gif
  ID = LOGO
  USAGE = None
  VALUE = None
ITEM = 34droad/34icon.gif
  HREF = 34droad/34icon.gif
  ID = ICON
  USAGE = None
  VALUE = None
ITEM = 34droad/34main.gif
  HREF = 34droad/34main.gif
  ID = MGIF
  USAGE = None
  VALUE = None
LOG = document:view
  VALUE = document:view

```

Patentansprüche

1. Maschinenlesbarer Datenträger, mit von einem Computer (16, 18) lesbaren Instruktionen, welche bei Implementierung den Computer (16, 18) veranlassen, Informationen von einem von dem Computer entfernt angeordneten Server (12) zu bearbeiten, indem er die folgenden Schritte ausführt:
 Speichern einer Inhaltsstrukturdatei (201), einer Datendatei (202) und einer Skriptdatei (204) auf dem Computer (16, 18), wobei die Datendatei (202) Daten aufweist, die die Informationen anzeigen, und die Skriptdatei (204) Skriptinformationen aufweist, die eine gewünschte Form anzeigen, in der die Daten aufbereitet werden sollen, wobei die Inhaltsstrukturdatei (201), die Datendatei (202) und die Skriptdatei (204) unabhängig voneinander von dem Computer (16, 18) empfangen und gespeichert werden;
 Lesen der Inhaltsstrukturdatei (201) zum Bestimmen, welches Skript in der Skriptdatei (204) mit den aufzubereitenden Daten in Zusammenhang steht;
 Abrufen der Daten aus der Datendatei (202) und Ausführen des zugehörigen Skriptbefehls in der Skriptdatei (204) zum Aufbereiten der Daten; und
 Anzeigen der aufbereiteten Daten auf einem von einem Computer (16, 18) gesteuerten Bildschirm.

2. Maschinenlesbarer Datenträger nach Anspruch 1, welcher von einem Computer lesbare Instruktionen aufweist, die bei Implementierung den Computer veranlassen, Informationen zu bearbeiten, indem er die folgende Schritte ausführt:

Speichern einer ersten Skriptdatei (**204**) und einer zweiten Skriptdatei (**232**) auf dem Computer (**16, 18**); und wobei das Lesen der Inhaltsstrukturdatei (**201**) zum Bestimmen, welches Skript das Durchsuchen der ersten Skriptdatei (**204**) nach dem zugehörigen Skript einschließt, und wobei die Durchführung des Ausführens eines weiteren Skriptbefehls aus der zweiten Skriptdatei (**232**) einschließt, wenn das zugehörige Skript nicht gefunden wird.

3. Maschinenlesbarer Datenträger nach Anspruch 1, welcher computerlesbare Instruktionen aufweist, die bei Implementierung den Computer veranlassen, Informationen zu bearbeiten, indem er folgende Schritte ausführt:

Speichern einer Präferenzdatei (**230**) auf dem Computer (**18**), welche Informationen bezüglich der in der Datendatei (**202**) zu speichernden Daten aufweist; und zeitweilig aussetzendes Empfangen aktualisierter Daten und Lesen der Präferenzdatei (**230**) zum Bestimmen, ob die aktualisierten Daten in der Datendatei (**202**) gespeichert werden sollen.

4. Maschinenlesbarer Datenträger nach Anspruch 3, dadurch gekennzeichnet, dass die Inhaltsstrukturdatei (**201**) Präferenzkennzeichen einschließt, die Standardpräferenzen zugehörig sind, und dadurch gekennzeichnet, dass das Speichern einer Präferenzdatei (**230**) das Lesen der Präferenzkennzeichen einschließt.

5. Maschinenlesbarer Datenträger nach Anspruch 4, welcher computerlesbare Instruktionen einschließt, die bei Implementierung den Computer (**16, 18**) veranlassen, Informationen zu bearbeiten, indem er den folgenden Schritt ausführt:

Ändern der Informationen in der Präferenzdatei (**230**).

6. Maschinenlesbarer Datenträger nach Anspruch 1, dadurch gekennzeichnet, dass die Inhaltsstrukturdatei (**201**) Verweise auf Daten und Skripten in einer Hierarchie einschließt.

7. Maschinenlesbarer Datenträger nach Anspruch 6, dadurch gekennzeichnet, dass das Lesen der Inhaltsstrukturdatei (**201**) zum Bestimmen, welches Skript in der Skriptdatei (**204**) den aufzubereitenden Daten zugehörig ist, das Auswählen des Skripts als eine Funktion der Hierarchie einschließt.

8. Maschinenlesbarer Datenträger nach Anspruch 1, dadurch gekennzeichnet, dass das Ausführen des Skripts das Aufbereiten der Daten in einer prozessorunabhängigen Form aufweist.

9. Maschinenlesbarer Datenträger nach Anspruch 8, dadurch gekennzeichnet, dass der Computer eine mobile Vorrichtung (**18**) aufweist, und dadurch gekennzeichnet, dass das Speichern der Inhaltsstrukturdatei (**201**), Datendatei (**202**) und Skriptdatei (**204**) das Empfangen der Inhaltsstrukturdatei (**201**), Datendatei (**202**) und Skriptdatei (**204**) von einem Tischcomputer (**16**) einschließt.

10. Verfahren zum Anzeigen von Informationen von einem Server (**12**) auf einem Bildschirm eines Computers (**16, 18**), welcher entfernt von dem Server (**12**) angeordnet ist, wobei das Verfahren die folgenden Schritte aufweist:

Speichern einer Inhaltsstrukturdatei (**201**), einer Datendatei (**202**) und einer Skriptdatei (**204**) auf dem Computer (**16, 18**), wobei die Datendatei (**202**) auf die Informationen hinweisende Daten aufweist, und die Skriptdatei (**204**) Skriptinformationen aufweist, die auf die gewünschte Form, in der die Daten aufbereitet werden sollen, hinweist, wobei die Inhaltsstrukturdatei (**201**), Datendatei (**202**) und Skriptdatei (**204**) unabhängig voneinander von dem Computer (**16, 18**) empfangen und gespeichert werden;

Lesen der Inhaltsstrukturdatei (**201**) zum Bestimmen, welches Skript in der Skriptdatei (**204**) den aufzubereitenden Daten zugehörig ist;

Abrufen der Daten aus der Datendatei (**202**) und Ausführen des zugehörigen Skriptbefehls in der Skriptdatei (**204**) zur Aufbereitung der Daten; und

Anzeigen der aufbereiteten Daten auf dem Computer-Bildschirm.

11. Verfahren nach Anspruch 10, welches weiterhin die folgenden Schritte aufweist:

Speichern einer ersten Skriptdatei (**204**) und einer zweiten Skriptdatei (**232**) auf dem Computer (**16, 18**); und wobei das Lesen der Inhaltsstrukturdatei (**201**) zum Bestimmen, welches Skript das Durchsuchen der ersten Skriptdatei (**204**) nach dem zugehörigen Skript, und die Verarbeitung des Ausführens eines weiteren Skriptbefehls aus der zweiten Skriptdatei (**232**) einschließt, wenn das zugehörige Skript nicht gefunden wird.

12. Verfahren nach Anspruch 10, welches weiterhin folgende Schritte aufweist:

Speichern einer Präferenzdatei (**230**) auf dem Computer (**16, 18**), welche Informationen bezüglich der in der

Datendatei (**202**) zu speichernden Daten aufweist; und zeitweilig aussetzendes Empfangen aktualisierter Daten und Lesen der Präferenzdatei (**230**) zum Bestimmen, ob die aktualisierten Daten in der Datendatei (**202**) gespeichert werden sollen.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass die Inhaltsstrukturdatei (**201**) Präferenzkennzeichen einschließt, die Standardpräferenzen zugehörig sind, und wobei Speichern einer Präferenzdatei (**230**) das Lesen der Präferenzkennzeichen einschließt.

14. Verfahren nach Anspruch 13, welches weiterhin das Ändern der Informationen in der Präferenzdatei (**230**) aufweist.

15. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass die Inhaltsstrukturdatei (**201**) Verweise auf Daten und Skripten in einer Hierarchie einschließt.

16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, dass das Lesen der Inhaltsstrukturdatei (**201**) zum Bestimmen, welches Skript in der Skriptdatei (**204**) den aufzubereitenden Daten zugehörig ist, das Auswählen des Skripts als eine Funktion der Hierarchie einschließt.

17. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass die Inhaltsstrukturdatei (**201**) Skriptkennzeichen einschließt, die Skripten in der Skriptdatei (**204**) zugehörig sind, wobei die Skriptkennzeichen in der Hierarchie sind.

18. Verfahren nach Anspruch 17, dadurch gekennzeichnet, dass das Auswählen der Skriptdatei (**204**) das Auswählen eines Skripts einschließt, auf das ein Skriptkennzeichen in einem höheren Abschnitt der Hierarchie für in einem niedrigeren Abschnitt der Hierarchie aufzubereitenden Daten verweist, wenn es kein zugehöriges Skriptkennzeichen für die Daten in dem niedrigeren Abschnitt der Hierarchie gibt.

19. Verfahren nach Anspruch 18, dadurch gekennzeichnet, dass das Skript das Aufbereiten der Daten in einer prozessorunabhängigen Form aufweist.

20. Verfahren nach Anspruch 19, dadurch gekennzeichnet, dass der Computer eine mobile Vorrichtung (**18**) aufweist, und wobei das Speichern der Inhaltsstrukturdatei (**201**), Datendatei (**202**) und Skriptdatei (**204**) das Empfangen der Inhaltsstrukturdatei (**201**), Datendatei (**202**) und Skriptdatei (**204**) von einem Tischcomputer einschließt.

Es folgen 8 Blatt Zeichnungen

Anhängende Zeichnungen

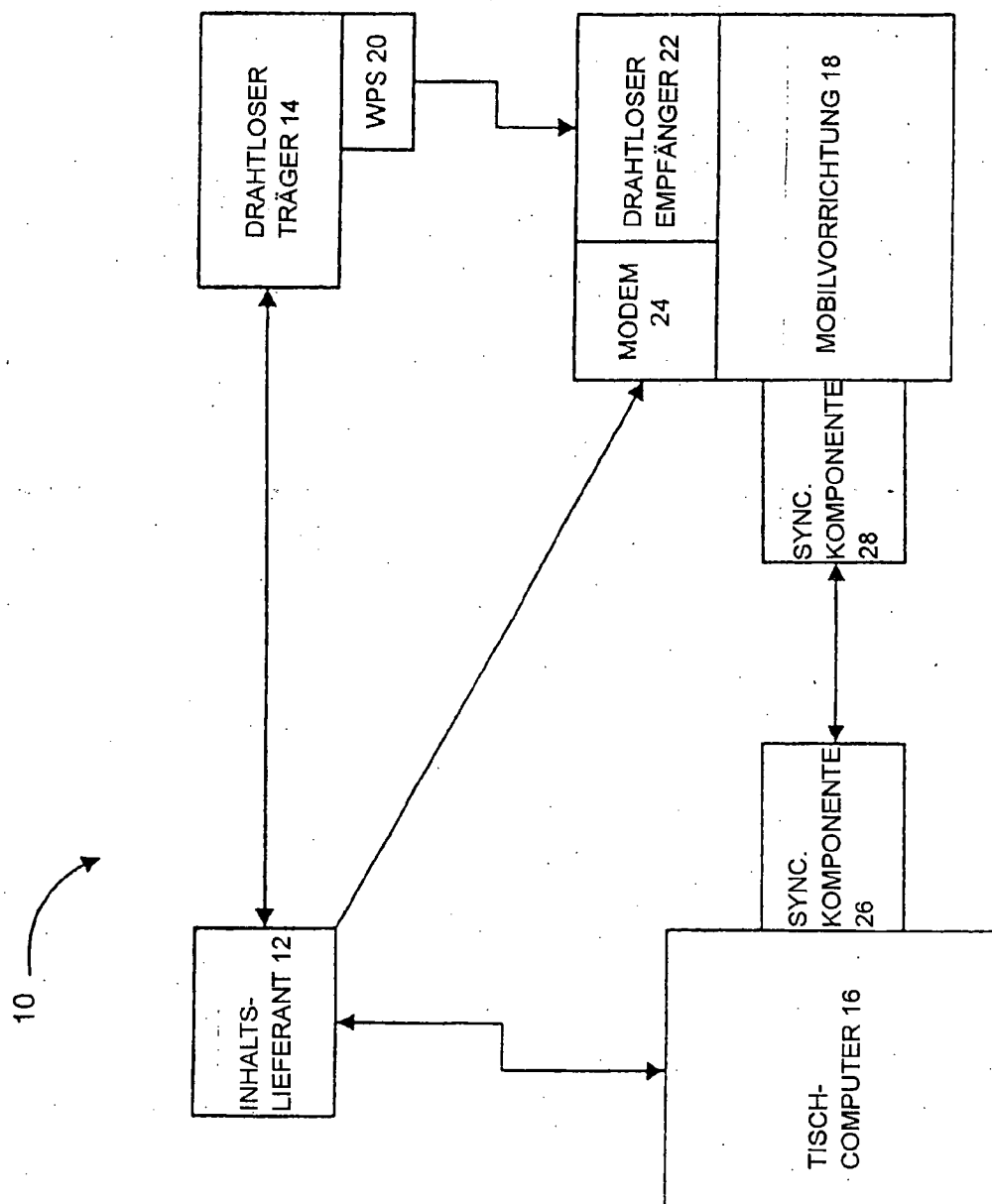


FIG. 1

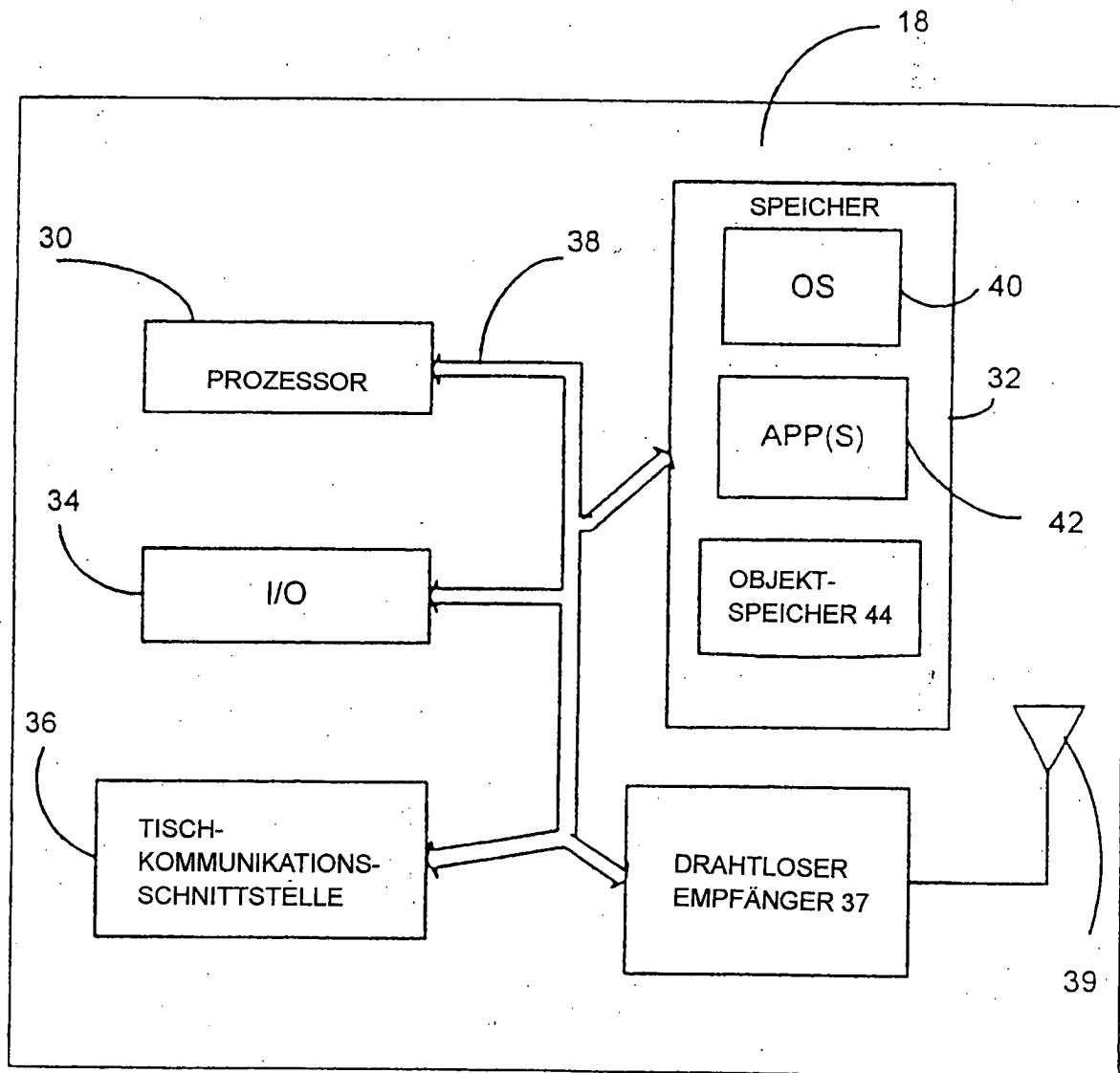


FIG.2

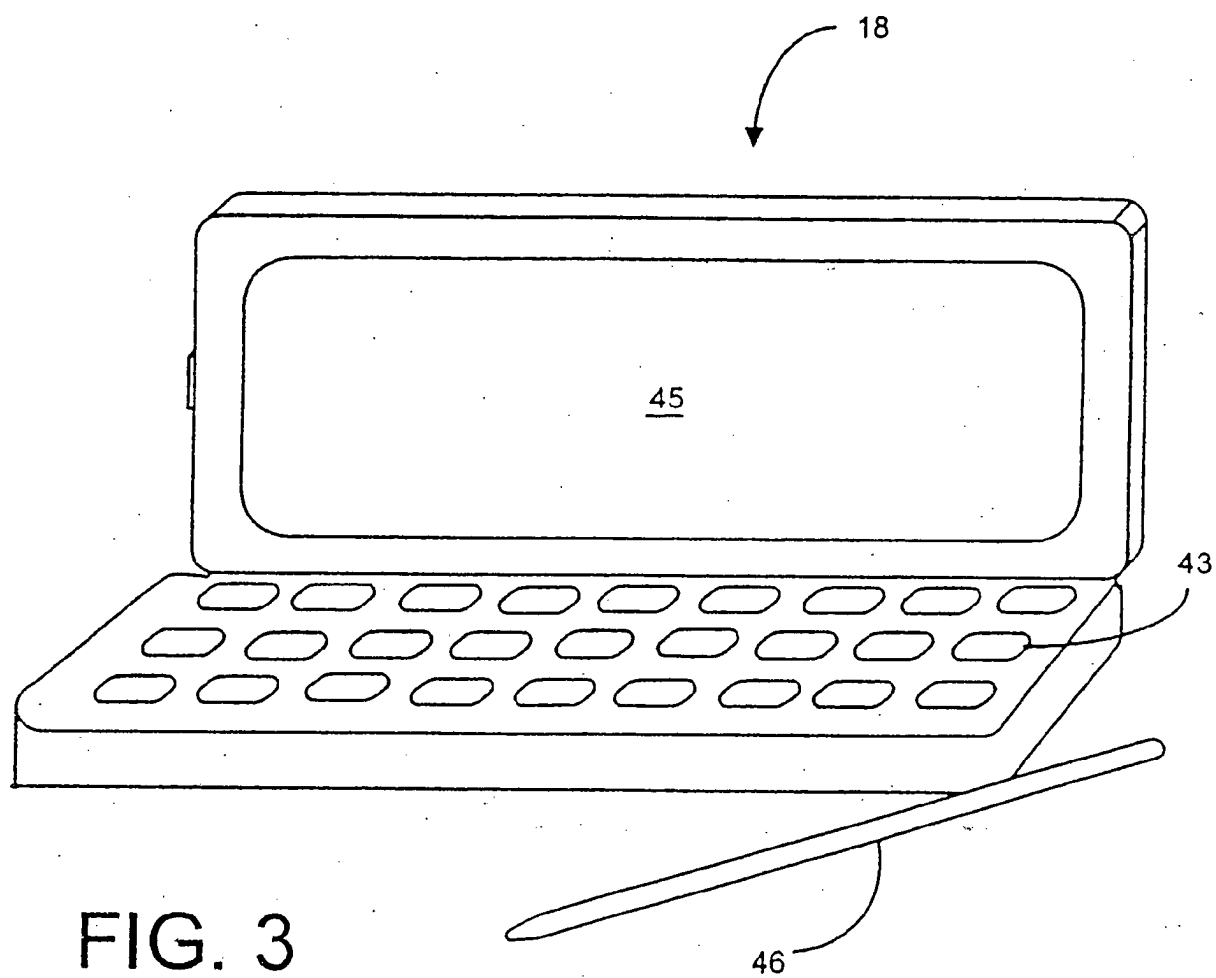


FIG. 3

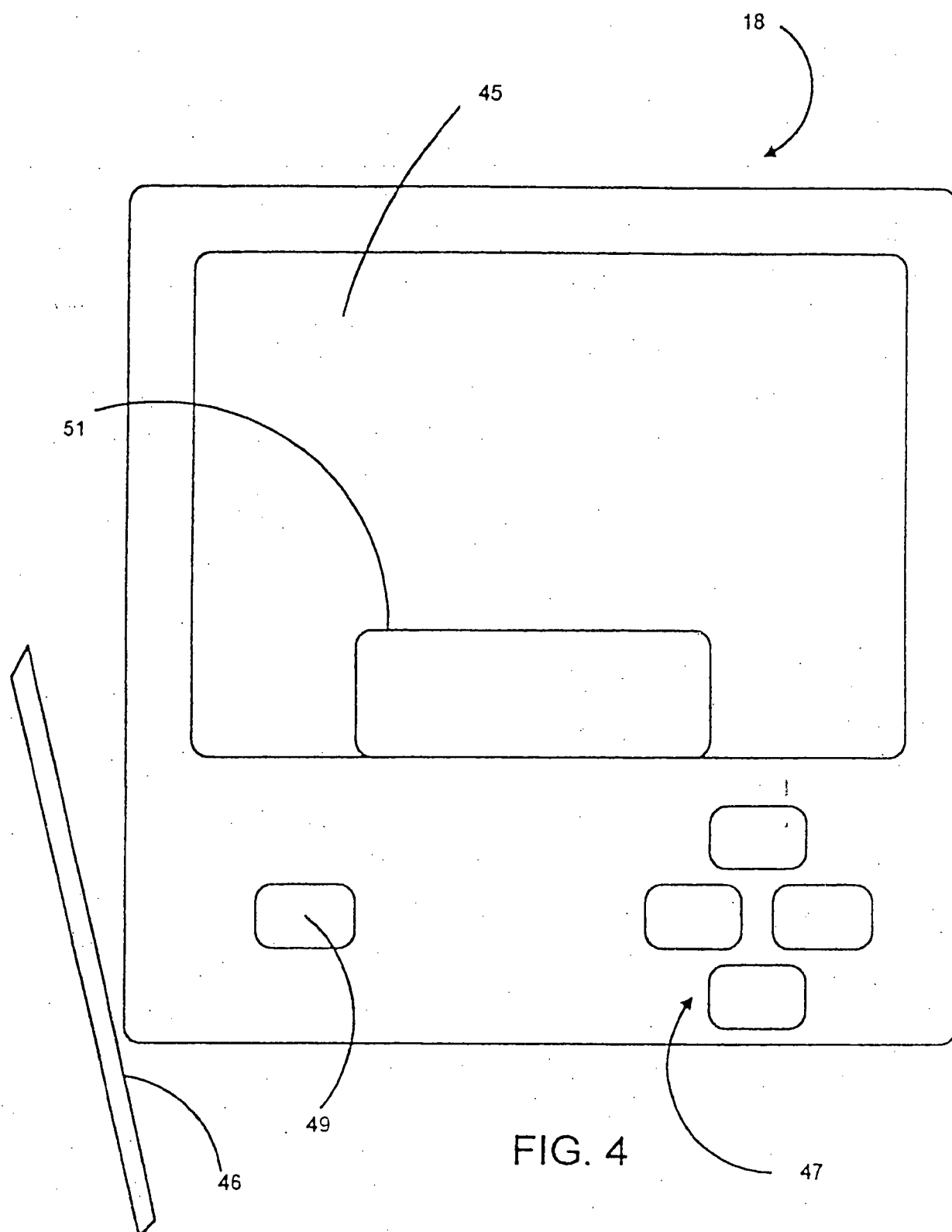


FIG. 5

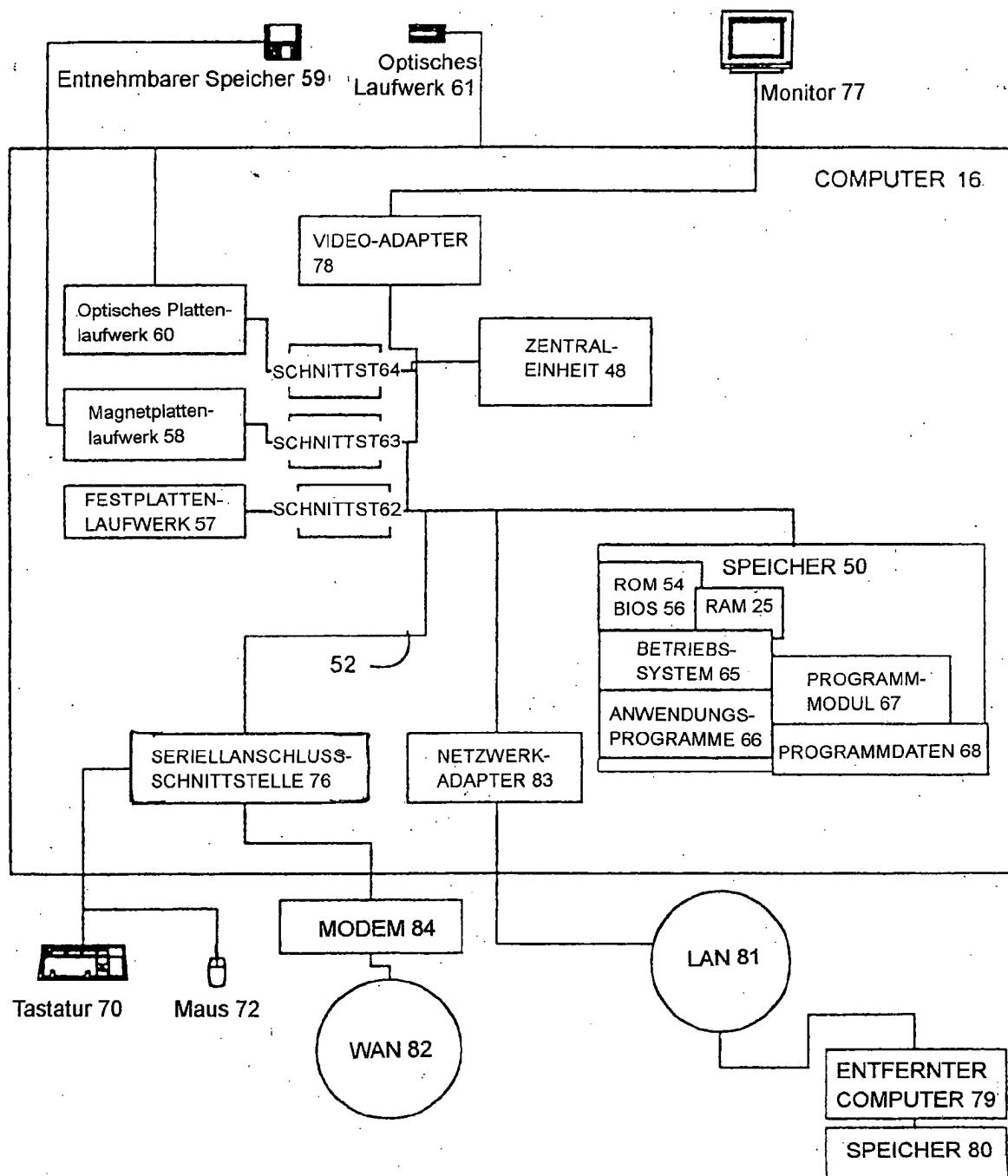


FIG. 6

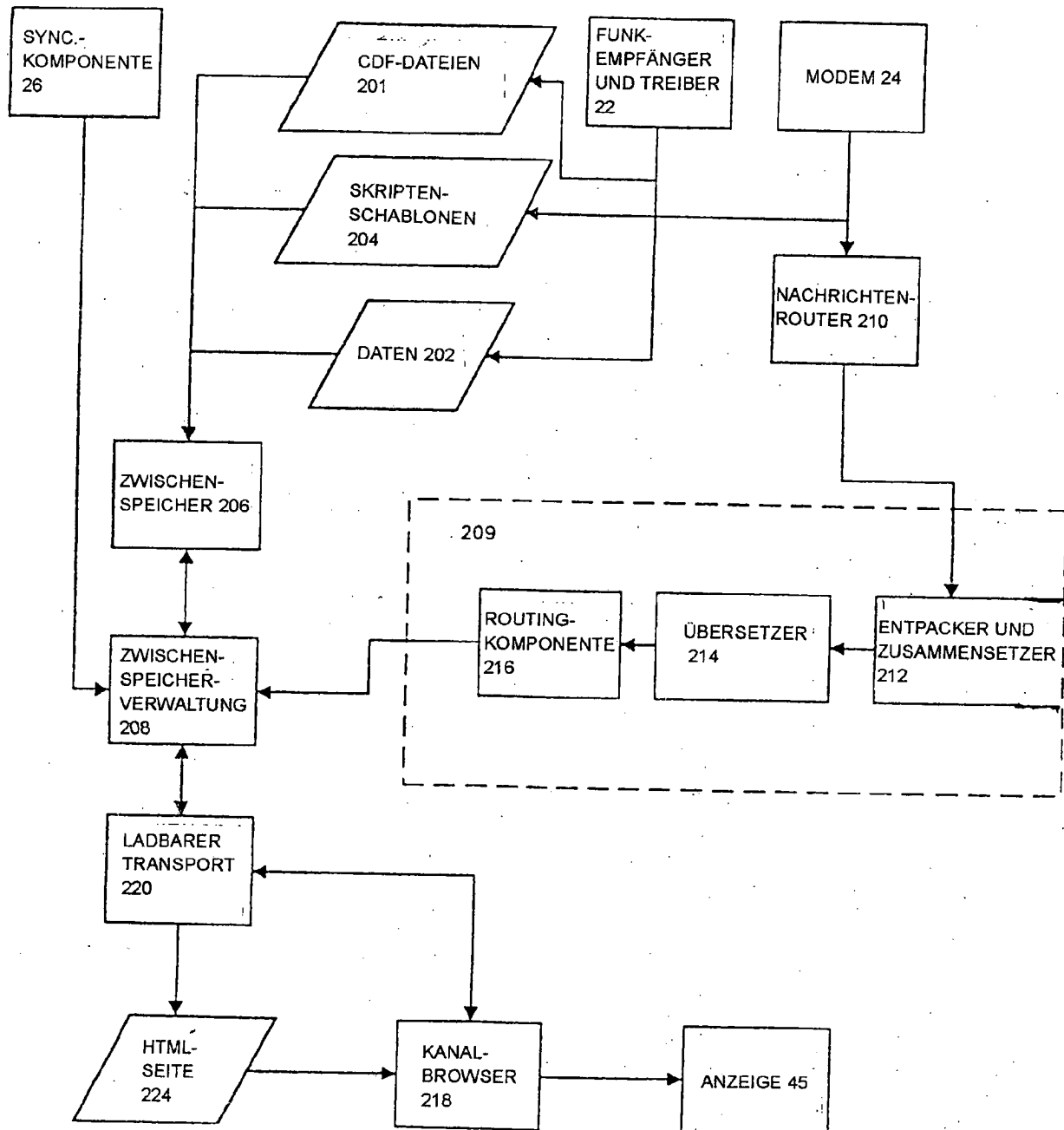
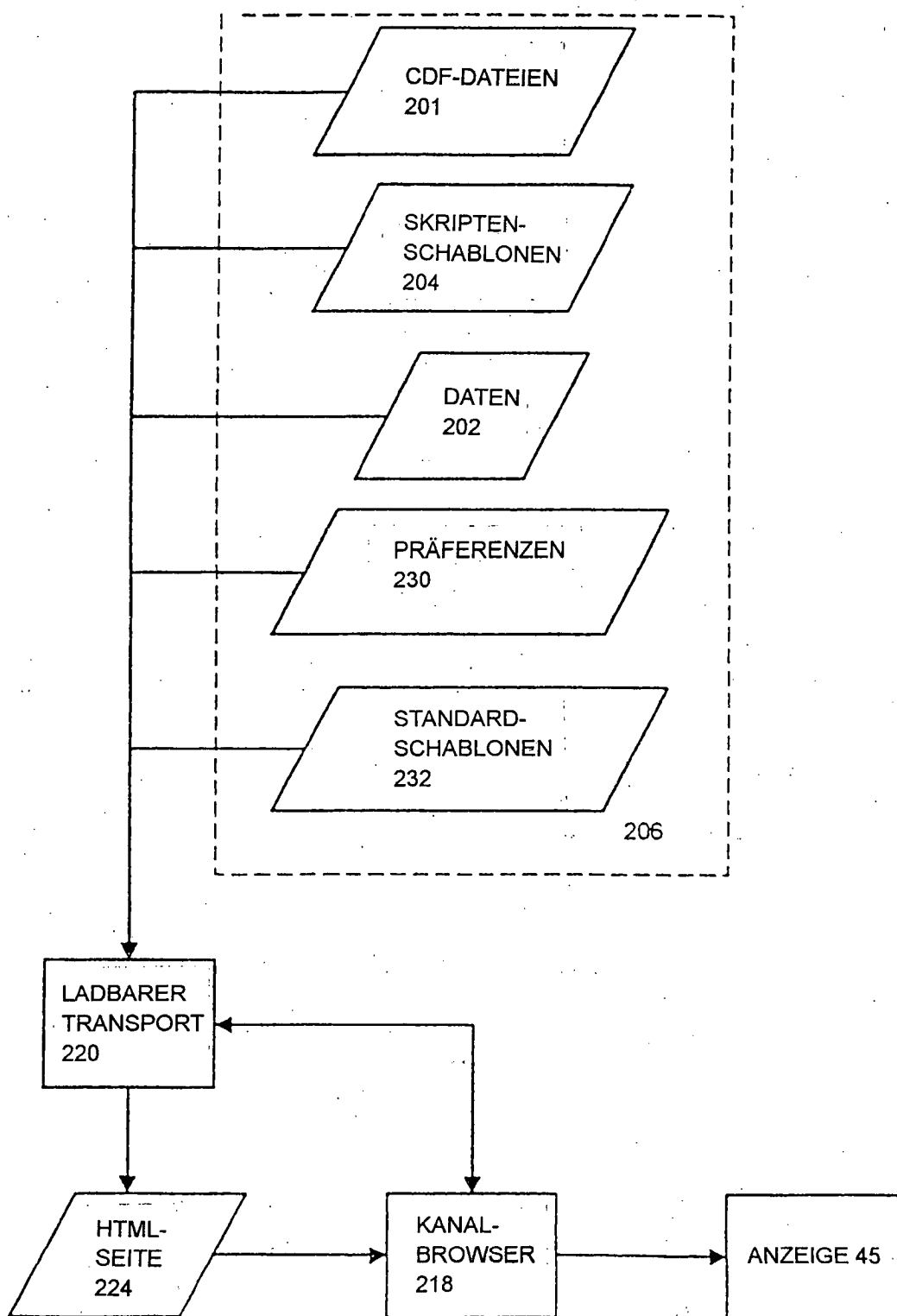


FIG. 7



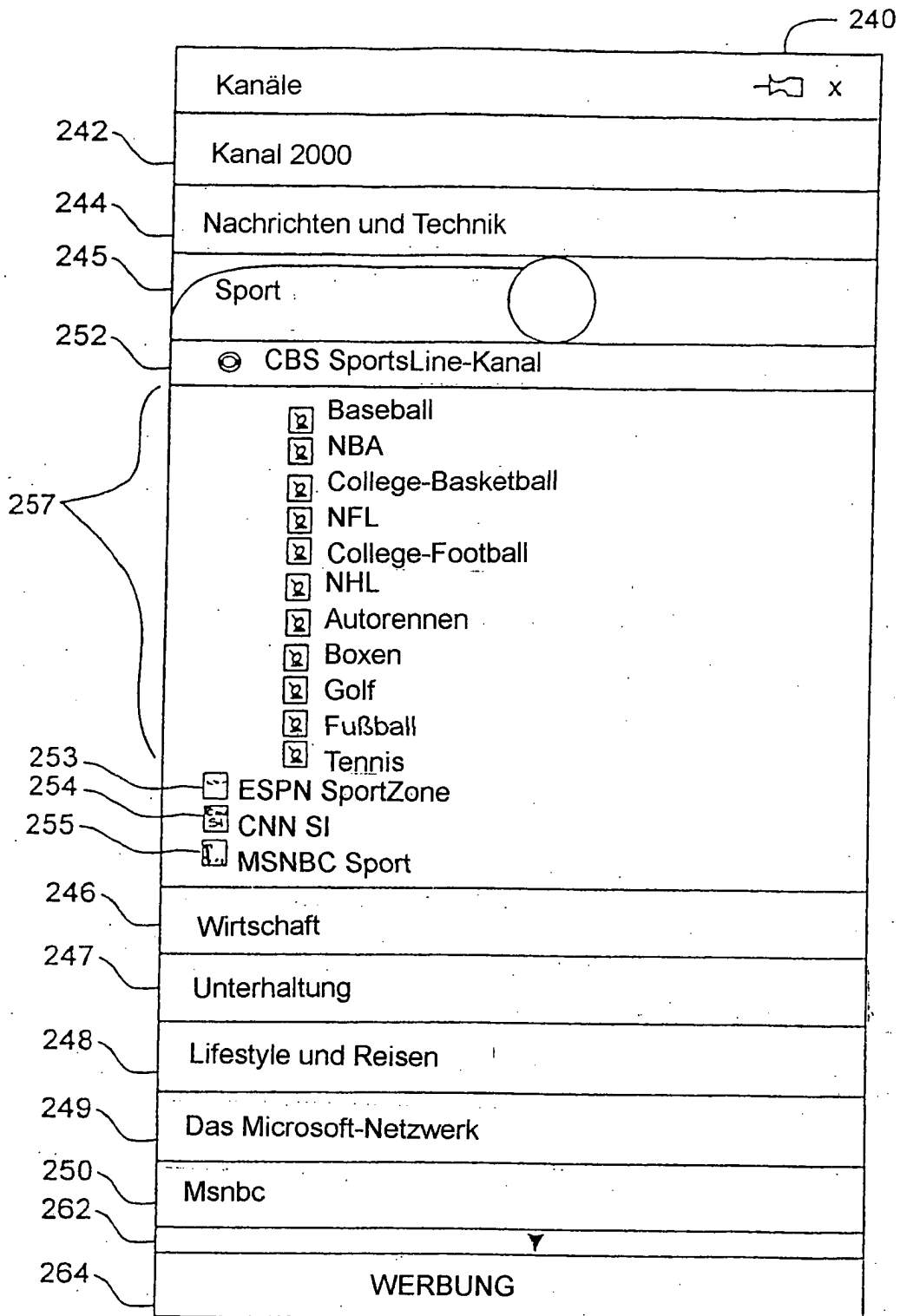


FIG. 8