(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 600 27 206 T2** 2006.12.21

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 285 337 B1**
(21) Deutsches Aktenzeichen: **600 27 206.0**
(86) PCT-Aktenzeichen: **PCT/IB00/01720**
(96) Europäisches Aktenzeichen: **00 974 724.7**
(87) PCT-Veröffentlichungs-Nr.: **WO 2001/033344**
(86) PCT-Anmeldetag: **01.11.2000**
(87) Veröffentlichungstag der PCT-Anmeldung: **10.05.2001**
(97) Erstveröffentlichung durch das EPA: **26.02.2003**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **05.04.2006**
(47) Veröffentlichungstag im Patentblatt: **21.12.2006**

(51) Int Cl.[8]: ***G06F 9/44*** (2006.01)

(30) Unionspriorität:

99402721	**02.11.1999**	**EP**
00300832	**03.02.2000**	**EP**

(84) Benannte Vertragsstaaten:
AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE, TR

(73) Patentinhaber:
THOMSON Licensing, Boulogne, FR

(74) Vertreter:
Roßmanith, M., Dipl.-Phys. Dr.rer.nat., Pat.-Anw., 30457 Hannover

(72) Erfinder:
JOUET, Canal+ Technologies Societe Anonyme, Bruno, F-75516 Paris Cedex 15, FR; NGUYEN VAN HUONG, Canal+ Technologies S.A., Emile, F-75516 Paris Cedex, FR; VILLERS, Jean-Stephane, F-95800 Cergy, FR

(54) Bezeichnung: **DARSTELLUNG VON GRAPHISCHEN OBJEKTEN**

Beschreibung

[0001] Die vorliegende Erfindung betrifft graphische Benutzerschnittstellen. Gesichtspunkte der vorliegenden Erfindung betreffen ein Verfahren zur Steuerung des Aussehens eines graphischen Gegenstandes in einer graphischen Benutzerschnittstelle. Gesichtspunkte der Erfindung betreffen einen Empfänger/Dekoder, ein Fernsehsystem, ein Computerprogrammprodukt, eine durch Computer lesbares Medium und ein Signal. Gesichtspunkte der Erfindung besitzen eine besondere, jedoch nicht ausschliessliche Anwendung zur Bereitstellung einer graphischen Benutzerschnittstelle für Geräte wie Empfänger/Dekoder für digitale Fernsehsignale. Gesichtspunkte der Erfindung besitzen jedoch auch Anwendungen auf Mehrzweckcomputern und anderen Geräten.

[0002] Die meisten graphischen Benutzerschnittstellen (GUI = graphical user interface) besitzen eine gleichartige Gruppe von Grundkomponenten, die von einem Benutzer gehandhabt werden können. Diese beinhalten Merkmale wie Drucktasten, Schieberegler, Verzeichnisbehälter usw. Derartige Komponenten werden im Allgemeinen mit "Trickfenstern" (widgets) bezeichnet. Obwohl die Grundfunktion von Trickfenstern innerhalb graphischer Benutzerschnittstellen allgemein gebräuchlich ist, unterscheiden sich die Trickfenster von einer graphischen Benutzerschnittstelle zur anderen durch ihr Erscheinungsbild oder Aussehen.

[0003] Einige graphische Betriebssysteme, wie das X Fenstersystem zum Beispiel, bringen einige Einschränkungen hinsichtlich des Erscheinungsbildes der Trickfenster, die in der graphischen Benutzerschnittstelle dargestellt werden können, mit sich. Dies ermöglicht den Programmierern Anwendungen zu entwickeln, in dem sie eine Anzahl von verschiedenen Trickfenstergruppen einsetzen, wobei jede ihr eigenes unterschiedliches Erscheinungsbild hat. Darüber hinaus läuft einer grosse Zahl von Fenstermanagern unter X, die den Gesamteindruck der Fenster, die durch die Anwendungen erstellt werden, beeinflussen. Normalerweise ist es möglich etwas Kontrolle über das Erscheinungsbild einer Anwendung auszuüben, wie es in der graphischen Schnittstelle dargestellt wird, sowohl während der Entwicklung der Anwendung, als auch in einem gewissen Mass während der Laufzeit. Das Erscheinungsbild ist jedoch in beiden Fällen durch die fest programmierten Teile der Trickfenstergruppe oder von dem Fenstermanager vorgegeben. Es ist dem Benutzer nicht möglich das Aussehen einer Anwendung in erheblichem Masse zu verändern, ohne die Trickfenstergruppe, den Fenstermanager, oder beide, erneut zu kodieren. Beide dieser Optionen erfordern einen grossen Umfang an erneuter Kodierungsarbeit und einen hohen Umfang neuer, auf dem zentralen Rechner zu installierender Kodes.

[0004] Ein Vorschlag, um den hohen Umfang an erneuter Kodierarbeit zu verringern, der getan werden muss, um eine Darstellung zu aktualisieren, ist Elemente eines Fensters in einer graphischen Schnittstellenanzeige der im Speicher des zentralen Rechners abgelegten Pixelabbilder zu bauen (zum Beispiel Kanten, Begrenzungen und so weiter). Dies kann als ein unabhängiger Gesichtspunkt der Erfindung angesehen werden. Die Pixelabbilder können im Betrieb einen erheblichen Umfang an Speicherplatz erforderlich machen und stellen einen riesigen Umfang an zu übertragende Daten dar, wenn das Aussehen aktualisiert werden muss. Dies kann einen erheblichen Nachteil bedeuten, wenn die Graphikschnittstelle mit beschränkten Reserven betrieben werden muss und über eine Verbindung mit begrenzter Bandbreite aktualisiert werden muss. Ein Beispiel einer derartigen Situation entsteht, wenn eine Anwendung auf einem Empfänger/Dekoder für digitale Fernsehsignale ausgeführt werden muss. Ein derartiger Dekoder besitzt eine begrenzte Speicherkapazität im Vergleich zu einem Allzweckcomputer und das Betriebssystem (einschliesslich des Erscheinungsbildes der Graphikschnittstelle) wird durch Herunterladen der Daten über einen Kanal, der Teil des empfangenen Fernsehsignals bildet, aktualisiert.

[0005] "Generic View Handler Class." IBM Technical Disclosure Bulletin, Band 34, Nr.1, Juni 1991 (1991–06), Seiten 397–398, ISSN 0018-8689 beschreibt eine objektorientierte Betrachtungs-Steuerungsbaureihe, die optische Erscheinung für einen Gegenstand, der mit ihm verbunden ist, behandelt. EP 0798634 beschreibt ein Gerät und ein Verfahren zur Trennung des Aufbaus und des Einsatzes einer Benutzerschnittstelle von dem Aufbau und dem Einsatz des funktionalen Teils eines Softwareprogramms. Look and feel Mittel wirken wie Server für logische Objekte. Ein Look and feel Mittel steuert das Erscheinungsbild und das Verhalten der Benutzerschnittstelle, während Logikobjekte die Funktion des Softwareprogramms erfüllen. Beide Lösungen sind darin begrenzt, dass sie Probleme zeigen, wenn die Betrachtungssteuerung aktualisiert wird.

[0006] Ein ziel dieser Erfindung ist einen Anwendungsentwickler anzubieten mit der Fähigkeit das Erscheinungsbild einer Anwendung auf durchgängige und leicht steuerbare Art und Weise zu steuern mit einem Minimum an notwendiger Neukodierung und mit einem Minimum Daten, die an ein Durchführungsumfeld übertragen werden müssen.

[0007] Bei einer ersten Betrachtungsweise der vorliegenden Erfindung wird ein Verfahren bereitgestellt zur Steuerung des Erscheinungsbildes eines objektorientierten Trickfensters in einer graphischen Benutzerschnittstelle, wie im Anspruch 1 ausgeführt.

[0008] Durch genaue Bestimmung eines Betrachtungsgegenstandes anstelle eines Einschlusskodes (embedding code), der das Erscheinungsbild in einer Anwendung steuert, kann die Erfindung eine grössere Flexibilität als bislang bieten, im Hinblick darauf, wie das Erscheinungsbild eines Trickfensters gesteuert wird.

[0009] Vorzugsweise beinhaltet der Betrachtungsgegenstand Kodes oder Parameter, die bestimmen, wie das Trickfenster dargestellt wird, wobei diese Kodes oder Parameter in einem Speicher abgelegt werden. Der Betrachtungsgegenstand kann zum Beispiel mit einem objektorientierten Programmkode bestimmt werden.

[0010] Der Betrachtungsgegenstand kann durch Realisieren einer Betrachtungsgegenstandsgruppe definiert werden. Ein Betrachtungsgegenstand der auf diese Weise bestimmt wurde, kann einen Zeiger auf eine andere Betrachtungsgegenstandgruppe enthalten (eine andere als die, von der sie abgeleitet wurde). Dies kann dem Betrachtungsgegenstand ermöglichen Attribute und/oder andere Verfahren dieser anderen Betrachtungsgruppen anzusteuern. Auf diese Art und Weise kann ein Betrachtungsgegenstand seine Eigenschaften aus zwei oder mehr Ansichten beziehen, die es ermöglichen eine Ansicht zu erzeugen, die die Vereinigung von zwei verschiedenen Ansichten darstellt, oder es ermöglicht Eigenschaften den Ansichten zuzufügen.

[0011] Um den graphischen Gegenstand mit dem Betrachtungsgegenstand zu verknüpfen, kann der graphische Gegenstand ein Attribut beinhalten, um den Betrachtungsgegenstand, der mit dem graphischen Gegenstand verbunden ist, zu identifizieren.

[0012] Vorzugsweise bestimmt die Ansicht die aktuellen Farben, die spezifisch benannten Farben zugeordnet sind. Der Betrachtungsgegenstand kann zum Beispiel die aktuellen Farben, die für zumindest einer schwarzen, weissen und einer oder mehreren grauen Abstufungen zugeordnet sind, bestimmen. Auf diese Weise kann die Ansicht ein bestimmtes Farbschema bestimmen, zum Beispiel, in dem sie den Betrachtungsgegenständen, die mit der Ansicht verknüpft sind, eine bestimmte Farbschattierung gibt.

[0013] Der Betrachtungsgegenstand kann auch eine Farbentafel bestimmen, die die aktuellen zu benutzenden Farbwerte einsetzt, die zur Darstellung einer bestimmten Farbe gebraucht werden.

[0014] Um das Erscheinungsbild des Trickfensters zu ändern, kann entweder der Betrachtungsgegenstand neu bestimmt oder verändert werden, zum Beispiel durch Wechseln der Kodes oder Parameter während der Übersetzung, oder der Parameter während der Laufzeit), oder ein unterschiedlicher Betrachtungsgegenstand kann mit dem graphischen Gegenstand verknüpft werden. Das Verfahren kann daher ferner die Veränderung des Erscheinungsbildes des graphischen Gegenstandes durch Neudefinition oder durch Veränderung des Betrachtungsgegenstands oder durch Verknüpfung eines unterschiedlichen Betrachtungsgegenstands mit dem graphischen Gegenstand, beinhalten.

[0015] Wenn ein Betrachtungsgegenstand neu bestimmt oder verändert wird, dann kann es notwendig werden das Erscheinungsbild der graphischen Gegenstände, die mit jenem Betrachtungsgegenstand verknüpft sind, zu aktualisieren. Um dies zu erreichen, kann der Betrachtungsgegenstand einen Aktualisierungszähler beinhalten, dessen Wert aktualisiert wird, wenn der Betrachtungsgegenstand neu bestimmt oder verändert wird.

[0016] Vorzugsweise speichern graphische Gegenstände den Wert des Aktualisierungszählers des Betrachtungsgegenstands, den sie beeinflusst haben. Jedes Mal wenn ein graphischer Gegenstand erneut angezeigt wird, wird der vom graphischen Gegenstand gespeicherte Wert mit dem Wert des Aktualisierungszählers des Betrachtungsgegenstands verglichen. Sind die beiden unterschiedlich, dann berücksichtigt der graphische Gegenstand die Veränderungen im Betrachtungsgegenstand und speichert den neuen Wert des Aktualisierungszählers.

[0017] Der Betrachtungsgegenstand kann eine Bewertungsmaske beinhalten, die Verfahren anzeigt, die durch den Betrachtungsgegenstand abgerufen werden können, so dass Verfahren, die vom Betrachtungsgegenstand nicht zur Anwendung gebracht werden, nicht abgerufen werden. Der graphische Gegenstand kann auf die Bewertungsmaske des Betrachtungsgegenstands zugreifen, um das Bild des graphischen Gegenstands zu optimieren. Auf diese Weise können Abrufe von nicht implementierten Verfahren vermieden werden, was die Zeichnung des graphischen Gegenstands beschleunigen kann.

[0018] Unter einigen Umständen, zum Beispiel wenn ein Betrachtungsgegenstand erstmals erzeugt wird, kann der Betrachtungsgegenstand mit einem einzigen graphischen Gegenstand verknüpft werden. In einem bevorzugten Ausführungsbeispiel nist jedoch der Betrachtungsgegenstand mit einer Vielzahl von graphischen Gegenständen verknüpft. Durch Verknüpfung des Betrachtungsgegenstands mit einer Vielzahl von graphischen Gegenständen, kann für jene Gegenstände ein einheitliches Erscheinungsbild erzielt werden und der Umfang an Daten, die zur Bestimmung des Betrachtungsgegenstands erforderlich sind, können reduziert werden, im Vergleich zu dem Fall, bei dem das Erscheinungsbild jedes graphischen Gegenstand unabhängig bestimmt wurde.

[0019] Daher kann das Verfahren ein Verfahren zur Steuerung des Erscheinungsbilds einer Vielzahl von graphischen Gegenständen in einer graphischen Benutzerschnittstelle sein und kann die Verknüpfung des Betrachtungsgegenstandes mit der Vielzahl der graphischen Gegenstände beinhalten.

[0020] Unter bestimmten Umständen, wo keine graphischen Gegenstände mit einem Betrachtungsgegenstand verknüpft sind, kann es zum Beispiel wünschenswert sein den Betrachtungsgegenstand zu entfernen, um den vom Betrachtungsgegenstand belegten Speicherplatz neu zuzuordnen. Dies kann besonders wichtig bei Vorrichtungen sein, wie bei Empfängern/Dekodern, wo der Speicherplatz begrenzt sein kann. Zu diesem Zweck (unter anderen) kann der Betrachtungsgegenstand einen Zähler beinhalten, der die Anzahl graphischer Gegenstände anzeigt, die mit diesem Betrachtungsgegenstand verknüpft sind. Jedes Mal wenn ein graphischer Gegenstand mit einem Betrachtungsgegenstand verknüpft wird, wird der Zähler vorzugsweise zunehmen und jedes Mal wenn ein graphischer Gegenstand von einem Betrachtungsgegenstand gelöst wird, wird der Zähler abnehmen. Wenn der Zähler bei Null steht, kann man unterstellen, dass der Betrachtungsgegenstand sicher gelöscht werden kann.

[0021] Im Verfahren der Erfindung ist die Bestimmung eines graphischen Gegenstandes in einer graphischen Schnittstelle eingeschlossen und beinhaltet die Bereitstellung eines Betrachtungsgegenstandes, der das Erscheinungsbild des graphischen Gegenstandes steuert (zum Beispiel durch Bestimmung der Eigenschaften und/oder Verfahren, die das Erscheinungsbild des graphischen Gegenstandes steuern) und durch Bereitstellung eines Trickfenstergegenstandes, der den Ablauf des graphischen Gegenstandes steuert (zum Beispiel durch Bestimmung der Eigenschaften und/oder Verfahren, die den Betrieb des graphischen Gegenstandes steuern). Jede der oben erwähnten Eigenschaften können in Verbindung mit diesem Aspekt bereitgestellt werden.

[0022] Jedes der oben beschriebenen Verfahren beinhalten ferner vorzugsweise die Darstellung des graphischen Gegenstandes, zum Beispiel auf einem Bildschirm, wie auf einem Computerbildschirm oder auf einem Fernsehbildschirm.

[0023] Jedes der oben beschriebenen Verfahren kann von einem Empfänger/Dekoder ausgeführt werden, wie ein digitaler oder analoger Empfänger/Dekoder.

[0024] Der begriff Empfänger/Dekoder der hier verwendet wird kann einen Empfänger bedeuten zum Empfang von entweder verschlüsselten oder nicht verschlüsselten Signalen, zum Beispiel Fernseh- und/oder Rundfunksignalen, die über einige andere Mittel gesendet oder übertragen werden können. Der Begriff kann auch einen Dekoder bedeuten zur Dekodierung empfangener Signale. Ausführungsbeispiele derartiger Empfänger/Dekoder können einen im Empfänger eingebauten Dekoder zur Dekodierung der empfangenen Signale einschliessen, zum Beispiel einen Beistelldekoder, wobei nein derartiger Dekoder mit einem räumlich getrennten Empfänger zusammen arbeiten kann, oder ein derartiger Dekoder, der zusätzliche Funktionen umfasst, wie einen Web-Browser, einen Videorekorder, oder ein Fernsehgerät.

[0025] Im Hinblick auf ein gerät der Erfindung wird ein gerät bereit gestellt zur Steuerung des Erscheinungsbildes eines objektorientierten Trickfensters in einer graphischen Benutzerschnittstelle, wie im Anspruch 6 dargestellt ist.

[0026] Das gerät kann einen passend programmierten Prozessor zur Bestimmung des Betrachtungsgegenstandes und zur Verknüpfung des Betrachtungsgegenstandes mit dem graphischen Gegenstand und einen Speicher zum Ablegen des Betrachtungsgegenstandes und des graphischen Gegenstandes einschliessen. Der Betrachtungsgegenstand kann durch den objektorientierten Programmkode bestimmt werden. Der Betrachtungsgegenstand kann durch Realisieren der Betrachtungsgegenstandsgruppe bestimmt werden. Der Betrachtungsgegenstand kann einen Zeiger auf eine andere Betrachtungsgegenstandsgruppe enthalten.

[0027] Das Gerät kann zur Veränderung des Erscheinungsbildes des graphischen Gegenstandes durch Neubestimmung oder Veränderung des Betrachtungsgegenstandes, oder durch Verknüpfung eines unterschiedlichen Betrachtungsgegenstandes mit dem graphischen Gegenstand angepasst werden. Der Betrachtungsgegenstand kann einen Aktualisierungszähler (wie zum Beispiel eine Stelle im Speicher) beinhalten, dessen Wert aktualisiert wird, wenn der Betrachtungsgegenstand neu bestimmt oder verändert wird.

[0028] Der Betrachtungsgegenstand kann eine Bewertungsmaske beinhalten (zum Beispiel im Speicher abgelegt), die Verfahren, die vom Betrachtungsgegenstand abgerufen werden können, anzeigt.

[0029] Das Gerät kann ein Gerät zur Steuerung des Erscheinungsbildes einer Vielzahl von graphischen Gegenständen in einer graphischen Benutzerschnittstelle sein und kann zur Verknüpfung des Betrachtungsgegenstandes mit einer Vielzahl von graphischen Gegenständen angepasst werden.

[0030] Das Gerät kann ferner Mittel einschliessen, wie zum Beispiel einen Bildschirm (zum Beispiel ein Computerbildschirm oder eine Fernsehbildschirm) zur Darstellung der graphischen Gegenstände.

[0031] Eine Trickfenstergruppe wird zur Erstellung von Gegenständen, wie oben beschrieben, bereitgestellt, wobei die Trickfenstergruppe eine Vielzahl von Trickfensterklassen und eine oder mehrere Betrachtungsgegenstandsklassen einschliesst. Höchst typisch umfasst eine Trickfenstergruppe eine Vielzahl von Betrachtungsgegenstandsklassen, einschliesslich einer Grundklasse und einer von der Grundklasse abgeleiteten Klasse.

[0032] Um das Erscheinungsbild des Gegenstandes zu ergänzen, kann die Betrachtungsgegenstandsklasse gewechselt werden. Dies verändert die Funktion des Gegenstandes nicht, da die Funktion durch die Trickfensterklasse gesteuert wird. Darüber hinaus ist es möglich ein Beispiel für die Betrachtungsgegenstandsklasse von vielen Beispielen der Trickfensterklasse eingesetzt zu werden, wobei der Umfang des belegten Speicherplatzes minimiert wird.

[0033] Die Betrachtungsgegenstandsklasse kann in einer Bibliothek, die mit einer Anwendung während Laufzeit verknüpft ist, enthalten sein. Ein Benutzer kann daher eine bevorzugte Ansicht für den Gegenstand wählen, in dem er diejenige aus einer Reihe von Bibliotheken auswählt, die verknüpft werden soll, auswählt. Da die Betrachtungsgegenstandsklasse unabhängig von anderen Gegenständen geändert wird, wird das erneute Kodieren auf das Nötigste vereinfacht, um das neue Erscheinungsbild zu implementieren.

[0034] Wahlweise kann die Betrachtungsgegenstandsklasse in einer Bibliothek, die mit einer Anwendung während der Übersetzung verknüpft ist, enthalten sein. Dies erlaubt einem Anwendungsentwickler vollständige Kontrolle über das Erscheinungsbild auszuüben.

[0035] Die Betrachtungsgegenstandsklasse kann Graphikverfahren exportieren, die aufgerufen werden können, um einen Teil einer Komponente auf einer Benutzerschnittstellenanzeige zu zeichnen. Die Trickfensterklasse enthält typischerweise einen Kode, der Graphikverfahren aus der Betrachtungsgegenstandklasse abruft. Betrachtungsgegenstandsklasse kann auch Eigenschaften exportieren, die Daten liefert, die sich auf Elemente einer Komponente beziehen.

[0036] Die Betrachtungsgegenstandsklasse kann eine Vorgabeklasse sein, die eine vorgegebene Ansicht für Trickfenster in einer graphischen Benutzerschnittstellenanzeige liefert. Wahlweise kann die Betrachtungsgegenstandsklasse eine von der Vorgabeklasse abgeleitete Klasse sein, wobei die abgeleitete Klasse ein oder mehrere Verfahren und/oder Eigenschaften der Vorgabeklasse aufhebt. In dieser Anordnung kann ein Benutzer oder Entwickler Veränderungen mit nur einer begrenzten Anzahl von Elementen innerhalb des Erscheinungsbildes der graphischen Benutzerschnittstelle vornehmen, ohne dass er irgendwelche Teile, die unverändert bleiben sollen, nacharbeiten müsste.

[0037] Das Klassenbeispiel des Betrachtungsgegenstands kann eine Eigenschaft des Klassenbeispiels der Trickfenster sein. Ein Zeiger auf ein Beispiel der Betrachtungsgegenstandsklasse wird einer Eigenschaft des Trickfensterklassenbeispiels zugeordnet. Ein derartiger Zeiger kann als Argument an einen Konstrukteur der Trickfensterklasse ausgegeben.

[0038] Bei einem Verfahren nach den letzten vorangegangen Absatz wird ein Zeiger auf die Betrachtungsgegenstandsklasse typischerweise in einem Beispielsspeicher des Trickfensterklassenbeispiels abgelegt.

[0039] Durch Steuerung des Vorgangs ist nicht notwendigerweise damit gemeint, dass der Gegenstand funktional ist; wobei Schieberegler und Bedienknöpfe und Ähnliches in der Lage sein können, Aktionen auszuführen, wobei einige graphischen Gegenstände nur zur Dekoration dienen, wie zum Beispiel Symbole oder Teile von zusammengesetzten Gegenständen. Somit können die Eigenschaften und/oder Verfahren, die den Betrieb des Gegenstandes steuern ein wenig mehr tun, als die Grundfunktion des Gegenstandes zu bestimmen (zum Beispiel als Zeiger zu erscheinen) mit dem präzisen Erscheinungsbild, das durch den Betrachtungsgegenstand gesteuert wird.

[0040] Es versteht sich, dass die vorliegende Erfindung einzig und allein anhand von Beispielen beschrieben wurde und Veränderungen von Teilen im Rahmen des Umfangs der Erfindung gemacht werden können.

[0041] Jede in der Beschreibung offen gelegte Eigenschaft und (wo angebracht) die Ansprüche und Zeichnungen können unabhängig, oder in jeder geeigneten Kombination, zur Verfügung gestellt werden.

[0042] Wo Eigenschaften des Gerätes hier mit "Mittel für" eine bestimmte Funktion beschrieben wurden, so ist beabsichtigt, dass diese Ausdrücke breit interpretiert werden und vorzugsweise nicht für irgendein bestimmtes hier beschriebenes Ausführungsbeispiel der Erfindung einschränkend interpretiert werden. Eigenschaften des Gerätes sind bei bevorzugten Ausführungsbeispielen durch einen passend programmierten Computer oder durch Computern geliefert worden und somit sind Eigenschaften des Gerätes vorzugsweise durch sachbezogene Eigenschaften eines Computers, oder eines Produktes, das ein Computerprogramm beinhaltet, erstellt worden. Eigenschaften des Gerätes können zum Beispiel durch einen Prozessor, oder andere Teile eines Computers zur Verfügung gestellt werden, zum Beispiel ein Speicher oder eine Datenablage.

[0043] Eigenschaften einer Betrachtungsweise können bei jeder anderen Betrachtungsweise eingesetzt werden, Verfahrenseigenschaften können im Hinblick auf Geräte und umgekehrt eingesetzt werden.

[0044] Bevorzugte Eigenschaften der vorliegenden Erfindung werden nun beschrieben werden, einzig und allein durch Beispiele unter Bezug auf die begleitenden Zeichnungen, worin:

[0045] Die **Fig. 1a** einen Überblick über ein typisches digitales Fernsehsystem gibt;

[0046] Die **Fig. 1b** die hauptsächliche Architektur eines interaktiven Fernsehsystems zeigt;

[0047] Die **Fig. 2a** ein Blockdiagramm eines Empfängers/Dekoders zeigt;

[0048] Die **Fig. 2b** die Architektur eines Empfängers/Dekoders zeigt;

[0049] Die **Fig. 2c** die weitergehende Architektur eines Empfängers/Dekoders zeigt;

[0050] Die **Fig. 3** ein Diagramm eines Teiles der Hierarchie von Trickfenstern (widgets) innerhalb einer Trickfenstergruppe zeigt;

[0051] Die **Fig. 4** ein vereinfachtes Diagramm eines Trickfensters ist, das auf der graphischen Benutzerschnittstellenanzeige erscheint;

[0052] Die **Fig. 5** die Stelle im Speicher von mehreren Trickfenstern zeigt;

[0053] Die **Fig. 6a** eine Bildschirmanzeige des Web-Browser zeigt;

[0054] Die **Fig. 6b** eine Fernbedienung zum Navigieren des Web-Browsers zeigt;

[0055] Die **Fig. 7** eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0056] Die **Fig. 8** eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0057] Die **Fig. 9** eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0058] Die **Fig. 10** eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0059] Die **Fig. 11** eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0060] Die Fig. 12 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0061] Die Fig. 13 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0062] Die Fig. 14 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0063] Die Fig. 15 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0064] Die Fig. 16 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0065] Die Fig. 17 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0066] Die Fig. 18 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0067] Die Fig. 19 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0068] Die Fig. 20 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0069] Die Fig. 21 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0070] Die Fig. 22 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0071] Die Fig. 23 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0072] Die Fig. 24 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0073] Die Fig. 25 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0074] Die Fig. 27 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0075] Die Fig. 27 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0076] Die Fig. 28 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0077] Die Fig. 29 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0078] Die Fig. 30 eine weitere Bildschirmanzeige des Web-Browsers zeigt;

[0079] Die Fig. 31 ein Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0080] Die Fig. 32 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0081] Die Fig. 33 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0082] Die Fig. 34 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0083] Die Fig. 35 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0084] Die Fig. 36 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0085] Die Fig. 37 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0086] Die Fig. 38 ein weiteres Beispiel für einen graphischen Gegenstand ist, der aus Fliesen gebildet ist;

[0087] Die Fig. 39 das Verfahren des Fliesens von graphischen Gegenständen zeigt;

[0088] Die Fig. 40 einen typischen Bildpuffer zeigt;

[0089] Die Fig. 41 den Aufbau eines aus Fliesen gebildeten graphischen Gegenstandes zeigt;

[0090] Die Fig. 42 schematisch die Wirkungsweise einer virtuellen Tastatur darstellt;

[0091] Die Fig. 43 eine typische Zuordnung von Zeichen zu den Tasten auf der virtuellen Tastatur darstellt;

[0092] Die Fig. 44 Beispiele einer virtuellen Tastatur zeigt; und

[0093] Die Fig. 45 die typischen Abmessungen einer virtuellen Tastatur zeigt.

Darstellung eines digitalen Fernsehsystems

[0094] Ein Überblick eines digitalen Fernsehsystems **1** ist in der Fig. 1a dargestellt. Die Erfindung schliesst ein hauptsächlich herkömmliches digitales Fernsehsystem **2** ein, das das bekannte MPEG-2 Komprimierungssystem zur Übertragung von komprimierten digitalen Signalen benutzt. In weiteren Einzelheiten empfängt ein MPEG-2 Kompressor **3** in einem Sendezentrum einen digitalen Datenstrom (typischerweise einen Strom von Videodaten). Der Kompressor **3** ist mit einem Multiplexer und Verwürfler (scrambler) **4** über die Verbindung **5** verbunden.

[0095] Der Multiplexer **4** empfängt eine Vielzahl weiterer Eingangssignale, assembliert den Transportstrom und überträgt komprimierte digitale Signale an den Sender **6** des Sendezentrums über die Verbindung **7**, die natürlich eine breite Vielfalt an Formen, einschliesslich Telefonnetzverbindungen, einnehmen kann. Der Sender **6** überträgt elektromagnetische Signale über die Verbindungsstrecke (uplink) **8** auf einen Satellitentransponder **9**, wo sie elektronisch verarbeitet werden und über eine gedankliche Abwärtsverbindung (downlink) **10** zum Empfänger auf der Erde **12**, herkömmlich in Form einer Satellitenschüssel, die dem Endverbraucher gehört, oder gemietet ist, übertragen werden. Andere Übertragungskanäle zur Übertragung von Daten sind selbstverständlich möglich, wie zum Beispiel terrestrische Übertragung, Kabelübertragung, kombinierte Satelliten-/Kabelverbindungen, Telefonnetze usw.

[0096] Die vom Empfänger **12** empfangenen-Signale werden an einen integrierten Empfänger-Dekoder **13** weitergeleitet, der dem Endbenutzer gehört, oder den er gemietet hat und werden mit dem Fernsehgerät des Endbenutzers verbunden. Der Empfänger-Dekoder **13** dekodiert das komprimierte MPEG-2 Signal in ein Fernsehsignal für das Fernsehgerät **14**. Obwohl in der Fig. 1a ein getrennter Empfänger/Dekoder gezeigt wird, kann der Empfänger/Dekoder auch Teil eines integrierten, digitalen Fernsehgerätes sein. Wie hier als Begriff verwendet, schliesst ein Empfänger/Dekoder einen getrennten Empfänger/Dekoder ein, wie zum Beispiel einen Beistelldekoder und ein Fernsehgerät, das einen integrierten Empfänger/Dekoder darin beinhaltet.

[0097] Bei einem Mehrkanalsystem bearbeitet der Multiplexer **4** Audio- und Videoinformationen, die er von eine Anzahl paralleler Quellen empfängt und kommuniziert mit dem Sender **6**, um die Informationen weiter mit einer entsprechenden Anzahl von Kanälen zu übertragen. Zusätzlich zu den audiovisuellen Informationen können Nachrichten, oder Anwendungen, oder jede andere Art von digitalen Daten in einige oder alle dieser Kanäle einfliessen, in dem sie mit den übertragenen digitalen Audio- und Videoinformationen verschachtelt werden.

[0098] Ein bedingtes Zugangssystem (conditional access system) **15** ist mit dem Multiplexer **4** und dem Empfänger/Dekoder **13** verbunden und ist teilweise im Übertragungszentrum und teilweise im Empfänger/Dekoder untergebracht. Es gestattet dem Endverbraucher auf digitale Fernsehprogramme von einem oder von mehreren Programmanbietern zuzugreifen. Eine Chipkarte (smartcard), die in der Lage ist verschlüsselte Mitteilungen, die sich auf kommerzielle Angebote (das ist eines oder mehrere Fernsehprogramme, die von Programmanbieter verkauft werden) zu entschlüsseln, kann in den Empfänger/Dekoder **13** eingeschoben werden. Wenn man den Empfänger/Dekoder **13** und die Chipkarte einsetzt, kann der Endverbraucher kommerzielle Angebote entweder als Abonnement oder Bezahlen je nach Programm kaufen. Wie hier der Begriff Chipkarte (smartcard) gebraucht wird schliesst dieser, jedoch nicht ausschliesslich, jedes Gerät mit Chipkarten, oder einen Gegenstand mit ähnlicher Funktion und Leistungsfähigkeit ein, der zum Beispiel einen Mikroprozessor und/oder einen Speicher besitzt. Unter diesem Begriff sind auch Anordnungen einbezogen, die alternative körperliche Ausgestaltungen einer Karte haben, so zum Beispiel, schlüsselförmige Anordnungen, wie sie oft bei Fernseh-Dekodersystemen eingesetzt werden.

[0099] Wie oben erwähnt, werden durch das System übertragene Programme am Multiplexer **4** verschlüsselt, die Bedingungs- und Verschlüsselungsschlüssel für eine gegebene Sendung, die von dem Zugangskontrollsystem **15** bestimmt wird, eingesetzt. Die Übertragung von auf diese Art und Weise verschlüsselten Daten ist auf dem Gebiet von Fernsehsystemen für das Bezahlfernsehen bekannt. Verschlüsselte Daten werden typi-

scherweise zusammen mit einem Kontrollwort zum Entschlüsseln der Daten gesendet, wobei das Kontrollwort selbst durch einen so genannten Verwertungsschlüssel verschlüsselt und in verschlüsselter Form übertragen wird.

[0100] Der verschlüsselten Daten und das verschlüsselte Kontrollwort werden dann vom Dekoder **13** empfangen, der einen Zugang zu einem entsprechenden Verwertungsschlüssel besitzt, der auf einer Chipkarte gespeichert ist und die in den Dekoder gesteckt wird, damit das verschlüsselte Kontrollwort entschlüsselt wird und anschliessend die übertragenen Daten entschlüsselt. Ein Abonnent, der bezahlt hat, wird zum Beispiel in einer monatlich übertragenen Freigabenachricht ECM (entitlement control message) den notwendigen Verwertungsschlüssel empfangen, um das verschlüsselte Kontrollwort zu entschlüsseln, damit er die Übertragungen ansehen kann.

Interaktives System

[0101] Ein interaktives System **16** das ebenfalls an den Multiplexer **13** und den Empfänger/Dekoder **13** angeschlossen werden kann und wiederum teilweise im Übertragungszentrum und teilweise im Empfänger/Dekoder untergebracht sein kann, erlaubt dem Endbenutzer über einen mit einem Modem verbundenen Rückkanal **17** mit verschiedenen Anwendungen zusammen zu wirken. Der über ein Modem verbundene Rückkanal **17** kann auch für Datenübertragungen, die beim bedingten Zugangssystem **15** gebraucht werden, eingesetzt werden.

[0102] Die Fig. 1b stellt die Hauptarchitektur des interaktiven Fernsehsystems **16** des digitalen Fernsehsystems **1** dar.

[0103] Das interaktive System **16** erlaubt dem Endbenutzer zum Beispiel einen Artikel aus einem Bildschirmkatalog zu kaufen, und auf Anfrage die lokalen Nachrichten anzusehen und Wetterkarten anzuschauen und Spiele über das Fernsehgerät zu spielen.

[0104] Das interaktive System **16** beinhaltet im Überblick für Hauptelemente:

– ein Autorensystem **4004** im Übertragungszentrum oder irgendwo anderes, um einem Programmanbieter Anwendungen zu schaffen, zu entwickeln, Fehler zu beseitigen und zu testen;

– einen Anwendungs- und Datenserver **4006** im Übertragungszentrum, der mit dem Autorensystem **4004** verbunden ist, um einem Programmanbieter es zu ermöglichen Anwendungen und Daten vorzubereiten, zu authentifizieren und zu formatieren zur Lieferung an den Multiplexer und Verwürfler (scrambler) **4**, um diese in den MPEG-2 Transportstrom (typischerweise den privaten Anteil davon) einzufügen, damit dieser dann an den Endbenutzer gesendet wird.

– eine virtuelle Maschine, die eine Ablaufmaschine (RTE run time engine) **4008** beinhaltet, die einen ausführbaren Kode darstellt, der im Empfänger/Dekoder installiert ist, der dem Endbenutzer gehört, oder der gemietet ist, um dem Endbenutzer zu gestatten Anwendungen zu empfangen, zu authentifizieren, zu dekomprimieren und in den Arbeitsspeicher des Dekoders **13** zur Ausführung zu laden. Die Maschine **4008** betreibt auch residente und Allzweckanwendungen. Die Maschine **4008** ist von der Hardware und vom Betriebssystem unabhängig; und

– einen Rückkanal über ein Modem **17** zwischen dem Empfänger/Dekoder **13** und dem Anwendungs- und Datenserver **4006**, um den Signalen, die den Server **4006** anweisen, zu erlauben, Daten und Anwendungen in den MPEG-2 Transportstrom auf Anfrage des Endbenutzers einzufügen.

[0105] Das interaktive System arbeitet unter Verwendung von "Anwendungen", die die Funktionen des Empfänger/Dekoders und verschiedene darin enthaltene Vorrichtungen steuern. Anwendungen sind in der Maschine **4008** als "Ressourcendateien" enthalten. Ein "Modul" ist eine Reihe von Ressourcendateien und Daten. Ein "Speicherdatenträger" des Empfängers/Dekoders ist ein Speicherplatz für Module. Module können in den Empfänger/Dekoder **13** aus dem MPEG-2 Transportstrom heruntergeladen werden.

Empfänger/Dekoder

[0106] Unter Bezug auf die Fig. 2a werden nun die verschiedenen Elemente des Empfängers/Dekoders **13** in Form von funktionalen Blöcken beschrieben.

[0107] Der Empfänger/Dekoder **13** der zum Beispiel ein digitaler Beistelldekoder (DTSB digital set-top box) sein kann beinhaltet eine zentralen Prozessor **220**, der damit verbundene Speicherelemente einschliesst und geeignet ist, Eingangsdaten aus einer seriellen Schnittstelle **221**, einer parallelen Schnittstelle **222** (verbunden

mit dem Modem Rückkanal **17** aus der Fig. 1a), einem Modem **223** und Schaltkontakten **224** auf der Vorderseite des Dekoders zu empfangen.

[0108] Der Empfänger/Dekoder ist zusätzlich geeignet Eingaben von einer Infrarotfernbedienung **225** über die Steuereinheit **226** zu empfangen und beinhaltet zwei Chipkartenlesegeräte **227**, **228**, die geeignet sind jeweils Bank- und Abonnementchipkarten **242**, **240** zu lesen. Das Lesegerät **228** für Chipkarten arbeitet zusammen mit einer eingelegten Abonnementchipkarte **240** und mit einer Einheit für den bedingten Zugang **229** (conditional access) zur Bereitstellung des notwendigen Kontrollworts an die Demultiplexer-/Entschlüsselungseinheit **230**, um dem verschlüsselten Übertragungssignal zu ermöglichen, dekodiert zu werden. Der Dekoder beinhaltet ebenfalls einen herkömmlichen Tuner **231** und Demodulator **232**, um die Satellitenübertragung zu empfangen und zu demodulieren, bevor sie durch die Einheit **230** gefiltert und entschachtelt wird.

[0109] Die Verarbeitung von Daten innerhalb des Empfängers/Dekoders wird im Allgemeinen vom zentralen Prozessor **220** vorgenommen. Die Fig. 2b stellt die Softwarearchitektur des zentralen Prozessors **220** des Empfänger/Dekoders dar. Unter Bezug auf die Fig. 2b beinhaltet die Softwarearchitektur eine Laufzeitmaschine **4008**, einen Gerätemanager **4068** und ein Vielzahl von Geräten **4062** und Gerätetreibern **4066**, um eine oder mehrere Anwendungen **4056** durchzuführen.

[0110] Wie in dieser Beschreibung verwendet ist eine Anwendung vorzugsweise ein Teil des Computerkodes zur Steuerung von Funktionen auf hoher Ebene von vorzugsweise des Empfängers/Dekoders **13**. Wenn zum Beispiel der Endbenutzer den Fokus der Fernbedienung **225** auf ein Bedienknopfobjekt positioniert, das auf dem Bildschirm des Fernsehgerätes **14** zu sehen ist und die Bestätigungstaste drückt, läuft die Anweisungsabfolge, die mit dem Bedienknopf verknüpft ist, an.

[0111] Eine interaktive Anwendung schlägt Menüs vor und führt Befehl auf Anfrage des Endbenutzers durch und liefert Daten, die zum Zweck der Anwendung gehören. Anwendungen können entweder residente Anwendungen sein, das heisst im ROM (oder Flashspeicher oder einem anderen nicht flüchtigen Speicher) des Empfänger/Dekoders **13** abgelegt, oder übertragene und in den RAM- oder Flashspeicher des Empfänger/Dekoders **13** herunter geladene Anwendungen sein.

[0112] Anwendungen werden an Speicherorten im Empfänger/Dekoder **13** abgelegt und als Ressourcendateien geführt. Die Ressourcendateien beinhalten Dateien der Beschreibungseinheit für graphische Gegenstände, Dateien für variable Blockeinheiten, Dateien der Anweisungsabfolge, Dateien für Anwendungen und Daten, so wie dies in den oben erwähnten Patentspezifikationen in allen Einzelheiten beschrieben wurde.

[0113] Der Empfänger/Dekoder enthält Speicher, die in einen RAM Datenträger, einen FLASH Datenträger und einen ROM Datenträger aufgeteilt sind, diese physikalische Organisation ist jedoch zur logischen Organisation unterschiedlich. Der Speicher kann ferner in Speicher-Datenträger, die mit verschiedenen Schnittstellen verknüpft sind, aufgeteilt werden. Aus einer Sicht kann der Speicher als Teil der Hardware betrachtet werden, aus einer anderen Sicht kann der Speicher, getrennt von der Hardware dargestellt, als unterstützend, oder das ganze System beinhaltend betrachtet werden.

Architektur des Computerprogramms (software)

[0114] Der zentrale Prozessor **220** kann als auf die Laufzeitmaschine **4008** gerichtet betrachtet werden und bildet einen Teil einer virtuellen Maschine **4007**. Diese ist einerseits mit Anwendungen (die "high level" Seite) und andererseits (die "low level Seite), über verschiedenen dazwischen liegenden Logikeinheiten, die nachstehend beschrieben werden, mit der Empfänger-/Dekoderhardware **4061** verbunden und schlisset die verschiedenen Eingänge, wie oben beschrieben wurde, mit ein (das bedeutet zum Beispiel die serielle Schnittstelle **221**, die parallele Schnittstelle **222**, das Modem **223** und die Steuereinheit **226**).

[0115] Unter besonderem Bezug auf die Fig. 2b sind verschiedenen Anwendungen **4057** mit der virtuellen Maschine **4007** verbunden; einige der mehr gemeinsam benutzten Anwendungen können mehr oder weniger ständige im System angesiedelt sein, wie bei **4057** erwähnt wurde, während andere ins System herunter geladen werden, zum Beispiel aus dem MPEG Datenstrom oder nach Bedarf aus anderen Schnittstellen.

[0116] Die virtuelle Maschine **4007** beinhaltet zusätzlich zur Laufzeitmaschine **4008** einige residente Bibliotheksfunktionen **4006**, die eine Sammlung mit Werkzeugen (toolbox) **4058** einschliesst. Die Bibliothek enthält verschieden Funktionen in der C-Sprache, die von der Maschine **4008** benutzt wird. Diese schliesst Datenmanipulation wie Kompression, Expansion oder Vergleich von Datenstrukturen, Strichzeichnen, usw. ein. Die Bi-

bliothek **4006** beinhaltet auch Informationen über Software in Festwertspeichern (firmware) im Empfänger/Dekoder **13**, wie Hardware- und Softwarenummern und verfügbarer RAM Speicherplatz und eine Funktion, die eingesetzt wird, wenn ein neues Instrument **4062** herunter geladen wird. Funktionen können in die Bibliothek herunter geladen werden und werden im Flash- oder RAM-Speicher abgelegt.

[0117] Die Laufzeitmaschine **4008** ist mit einem Geräteverwaltungsprogramm **4068** verbunden, das mit einer Reihe von Geräten **4062** verbunden ist, die wiederum mit Gerätetreibern **4060** verbunden sind und die der Reihe nach mit den Eingängen oder Schnittstellen verbunden sind. Im weiteren Sinne kann ein Gerätetreiber zur Bestimmung einer logischen Schnittstelle betrachtet werden, so dass zwei verschiedene Gerätetreiber an einen gemeinsamen physikalischen Eingang angeschlossen werden können. Ein Gerät wird normalerweise an mehr als einen Gerätetreiber angeschlossen; wenn ein Gerät an einen einzigen Gerätetreiber angeschlossen wird, ist das Gerät normalerweise so aufgebaut, dass es die volle Funktionalität, die für den Datenaustausch notwendig ist, beinhaltet, so dass die Notwendigkeit für einen getrennten Gerätetreiber umgangen wird. Einige Geräte können unter sich selbst Daten austauschen.

[0118] Jede Funktion des Empfängers/Dekoders **13** ist als Gerät **4062** in der Softwarearchitektur des Empfängers/Dekoders vertreten. Geräte können entweder lokal oder entfernt ausgebaut sein. Lokale Geräte **4064** enthalten Chipkarten, SCART-Anschlusssignale, Modems, serielle und parallele Schnittstellen, eine MPEG Video- und Audioabspielvorrichtung und einen MPEG Sektions- und Tabellenextraktor. Die entfernt aufgebauten Geräte **4066**, die an einem entfernten Ort angewendet werden, unterscheiden sich von lokalen Geräten dadurch, dass eine Schnittstelle und der Arbeitsschritt durch die Systemleitung oder durch den Systementwickler bestimmt werden muss, als durch einen Gerätetreiber, der vom Hersteller des Empfängers/Dekoders geliefert und gebaut wurde.

[0119] Die Laufzeitmaschine **4008** läuft unter der Steuerung eines Mikroprozessors und einer gemeinsamen Schnittstelle für die Anwendungsprogrammierung (API application programming interface). Sie sind in jedem Empfänger/Dekoder **13** untergebracht, so dass alle Empfänger/Dekoder **13** von der der Anwendung her gesehen, identisch sind.

[0120] Die Maschine **4008** führt Anwendungen **4056** auf dem Empfänger/Dekoder **13** durch. Sie führt interaktive Anwendungen **4056** durch und empfängt Ereignisse von ausserhalb des Empfängers/Dekoders **13**, zeigt Graphik und Text an, ruft Geräte für Dienste auf und benutzt Funktionen der Bibliothek **4006**, die mit der Maschine **4008** für spezielle Berechnungen verbunden ist.

[0121] Die Laufzeitmaschine **4008** ist ein ausführbarer Kode, der in jedem Empfänger/Dekoder **13** eingebaut ist und schliesst einen Interpretierer zum Übersetzen der laufenden Anwendungen ein. Die Maschine **4008** kann an jedes Betriebssystem angepasst werden, einschliesslich Betriebssystemen für eine einzige Anwendung (so wie MS-DOS). Die Maschine **4008** arbeitet auf der Grundlage von Prozessablaufsteuerungseinheiten (die verschiedene Vorgänge, wie zum Beispiel ein Tastendruck, heranziehen, um verschiedene Aktionen auszuführen) und beinhaltet ihr eigenes Steuerprogramm, um Warteschlangen bei Vorgängen aus den verschiedenen Hardwareschnittstellen zu bewältigen. Sie wickelt auch die Darstellung von Graphiken und Text ab. Eine Prozessablaufsteuerungseinheit schliesst eine Reihe von Aktionsgruppen ein. Jeder Vorgang veranlasst die Prozessablaufsteuerungseinheit sich von ihrer momentanen Aktionsgruppe zu einer anderen Aktionsgruppe in Abhängigkeit der Art des Vorganges hin zu bewegen und die Aktionen der neuen Aktionsgruppe auszuführen.

[0122] Die Maschine **4008** schliesst ein Kodeladeprogramm zum Laden und Herunterladen von Anwendungen **4056** in den Speicher des Empfängers/Dekoders **13** ein. Nur der notwendige Kode wird in den RAM- oder Flashspeicher geladen, um optimalen Gebrauch sicher zu stellen. Die herunter geladenen Daten werden durch einen Authentifizierungsmechanismus überprüft, um jedwelche Veränderung einer Anwendung **4056** oder die Ausführung nicht frei gegebener Anwendungen zu verhindern. Die Maschine **4008** schliesst ferner einen Dekomprimierer ein. Wenn der Anwendungskode (eine Form des Zwischenkodes) zur Einsparung von Speicherplatz und zum schnellen Herunterladen aus dem MPEG Stroms oder über den eingebauten Empfänger-/Dekodermodus komprimiert wird, muss der Kodevor dem Laden in den RAM-Speicher dekomprimiert werden. Die Maschine **4008** schliesst auch einen Interpretierer zur Übersetzung des Anwendungskodes zur Aktualisierung verschiedener variabler Werte und zur Bestimmung von Zustandsveränderungen und einen Fehlerüberprüferkreis ein.

Architektur des Empfänger/Dekoders

[0123] Der Empfänger/Dekoder beinhält fünf Softwareschichten, die so organisiert sind, dass sie bei jedem Empfänger/Dekoder und bei jedem Betriebssystem eingesetzt werden können. Unter Bezug auf die Fig. 2c sind die verschiedenen Softwareschichten die Anwendungsschicht **250**, die Anwendungsprogrammschnittstelle (API application programming interface) **252**, die virtuelle Maschinenschicht **254**, die Geräteschicht **256** und die System-Software/-Hardwareschicht **258**.

[0124] Die Anwendungsschicht **250** umfasst Anwendungen die entweder resident im oder in den Empfänger/Dekoder herunter geladen wurden. Diese können interaktive Anwendungen sein, die vom Kunden benutzt werden und die zum Beispiel in Java, HTML, MHEG-5 oder anderen Sprachen geschrieben wurden, oder es können Anwendungen sein, die vom Empfänger/Dekoder benutzt werden, um solche Anwendungen zu betreiben. Diese Schicht basiert auf einer Reihe offener Anwendungsprogrammschnittstelle (APIs application programming interface), die von der virtuellen Maschineschicht bereitgestellt werden. Dieses System ermöglichtes, dass die Anwendungen in den Flash oder RAM-Speicher des Empfängers/Dekoders während der Übertragung oder auf Anfrage herunter geladen werden. Der Anwendungskode kann im komprimierten Format oder unkomprimierten Format unter Benutzung von Protokollen wie DSMCC (Data Storage Media Command and Control, Datenspeicherung für Datenträgerführung und Steuerung), NFS (Network File Server, Netzwerkdateiserver) oder anderen Protokollen.

[0125] Interaktive Anwendungen sind Anwendungen mit den der Benutzer interaktiv arbeitet, zum Beispiel und Produkte oder Dienstleistungen, wie elektronische Programmführer, Telebaninganwendungen und Spiele zu erhalten.

[0126] Verschiedene Sicherheitsmerkmale werden für die herunter geladenen Anwendungen und Daten wie folgt zur Verfügung estellt:

– Nichts kann auf den Empfänger/Dekoder herunter geladen werden, ohne zuerst im beabsichtigten Netzwerk authentifiziert zu sein, was verhindert, dass keine unregistrierte Software auf dem Empfänger/Dekoder laufen kann. Das bedeutet, dass jegliche Software, die auf dem Empfänger/Dekoder läuft, erkannt wurde und komplett getestet wurde.

– Ein Sicherheitsverwaltungsprogramm (security manager) beschränkt den Zugang zu Anwendungen in verschiedenen Speicherzonen und stellt damit Datenintegrität sicher.

– Das System kann mit jedem bedingten Zugangssystem (conditional access) gekoppelt werden, dass sich sicherer Prozessoren (zum Beispiel Chipkarten, die in den Empfänger/Dekoder eingeführt werden) bedient.

[0127] Die folgenden eingebauten Anwendungen werden zur Verwaltung der interaktiven Anwendungen benutzt:

– Start. Die Startanwendung **260** ist die erste in Gang gesetzte Anwendung, wenn der Empfänger/Dekoder eingeschaltet wird. Die Startanwendung beginnt mit den verschiedenen "Managern" (Verwaltungsprogramme) in der virtuellen Maschine, wobei die erste der Anwendungsmanager **262** ist.

– Anwendungsmanager. Der Anwendungsmanager **262** verwaltet die interaktiven Anwendungen, die auf dem Empfänger/Dekoder laufen, das heisst, er startet, stoppt, unterbricht, setz fort, wickelt Vorgänge ab und bearbeiten die Datenübertragung zwischen Anwendungen. Er erlaubt mehrfache Anwendungen zur gleichen Zeit zu betreiben und ist damit an der Zuordnung von Ressourcen innerhalb dieser Anwendungen beteiligt. Diese Anwendung ist für den Benutzer vollkommen transparent.

– Installation. Der Zweck der Installationsanwendung **264** ist den Empfänger/Dekoder zu konfigurieren, hauptsächlich, wenn er zum ersten Mal benutzt wird. Sie führt Aktionen wie die Abfrage der TV-Kanäle durch, Einstellen von Datum und Zeit, Erstellen der bevorzugten Benutzereinstellungen und so weiter. Die Installationsanwendung kann jedoch jedes Mal vom Benutzer eingesetzt werden, um die Konfiguration des Empfängers/Dekoders zu ändern.

– Zapping (Programmsuche). Die Programmsuchanwendung **268** wird eingesetzt, um die Kanäle mit den Tasten Programm "+" und Programm "–" und den Zahlentasten zu wechseln. Wenn eine andere Form des Programmwechsels gebraucht wird, zum Beispiel durch eine Anwendung mit einem Führungszeiger, wird die Programmsuchanwendung (zapping) angehalten.

– Callback (Rückfrage). Die Rückfrageanwendung (callback) wird zur Entnahme von Werten aus verschiedenen Parametern, die im Speicher des Empfänger/Dekoder abgelegt sind, eingesetzt und sie gibt diese Werte an den kommerziellen Betreiber über einen Rückkanal mit Modem **17**, oder über andere Mittel zurück.

[0128] Die API-Schicht **252** stellt "high level" Betriebsmittel zur interaktiven Anwendungsentwicklung zur Ver-

fügung. Sie schliesst mehrere Paket mit ein, die diese "high level" API kennzeichnen. Die Pakete liefern all notwendigen Funktionalitäten, um interaktive Programme zu betreiben. Die Pakete sind über die Anwendungen zugänglich.

[0129] In einem bevorzugten Ausführungsbeispiel ist die API geeignet in der Java Programmiersprache geschriebene interaktive Anwendungen auszuführen, Ferner kann es HTML und andere Formate interpretieren, wie MHEG-5. Neben diesen Übersetzern schliesst es auch andere Pakete und Dienstemodule ein, die abschaltbar und erweiterbar, so wie es die Anforderungen aufgeben, sind.

[0130] Die virtuelle Maschinenschicht **254** ist aus Sprachinterpretieren und verschiedenen Modulen und Systemen zusammengesetzt. Es besteht aus allem was zum Empfang und zur Ausführung interaktiver Anwendungen im Empfänger/Dekoder nötig ist, einschliesslich den Folgenden:

– Sprachinterpretierer. Verschiedene Interpretierer können installiert werden, um dem Typ von zu lesenden Anwendungen zu entsprechen. Dies schliesst Java, HTML, MHEG-5 und andere ein.

[0131] Diensteinformationsmaschine (SI). Die SI-Maschine lädt und überwacht gemeinsame Digitale Videoübertragung-(DVB) und Programmsysteminformationsprotokoll-(PSIP) Tabellen und lädt sie in den Cachespeicher. Sie ermöglicht den Zugang zu diesen Tabellen über Anwendungen, die die darin enthaltenen Informationen benötigen.

[0132] Steuerprogramm (scheduler) Dieses Modul erlaubt bevorrechtigte, nebenläufige Ablaufkoordination, wobei jedes Thread seine eigene Ereigniswarteschlange besitzt.

[0133] Speicherverwaltungsprogramm. Dieses Modul verwaltet den Speicherzugriff. Es komprimiert auch automatisch Daten im Speicher, falls notwendig und führt automatisch eine Speicherbereinigung durch.

[0134] Vorgangsverwaltungsprogramm. (Event Manager) Dieses Modul erlaubt Vorgänge gemäß Vorrang auszulösen. Es verwaltet den Zugriff auf Zeitgeber und Vorfälle und erlaubt den Anwendungen sich gegenseitig Vorgänge zu schicken.

[0135] Dynamischer Pogrammbinder. (Dynamic Linker) Dieses Modul erlaubt die Auflösung von Adressen, die von natürliche Java-Funktionen ausgehen, lädt von einer Java-Klasse ausgehende Verfahren, die in den RAM Speicher herunter geladen wurden und löst Anweisungen von herunter geladenen natürlichen Kodes in Richtung ROM.

[0136] Downloader (Herunterladen von Daten). Dieses Modul benutzt automatisches Herunterladen von Daten aus einem entfernten DSMCC Karussell oder über eine NFS Protokoll, mit herunter geladenen Dateien, auf die in gleicher Weise wie auf die residenten Dateien zugegriffen wird. Aufräumen des Speichers, Kompression und Authentifizierung werden ebenfalls bereitgestellt.

[0137] Class Manager (Klassenverwaltungsprogramm). Dieses Modul lädt Klassen und löst alle Problem die sich auf das Herstellen von Bezügen innerhalb der Klassen ergeben.

[0138] Dateisystem. (File System) Dieses Modul ist kompakt und zur Verwaltung eines hierarchischen Dateisystems mit mehrfachen ROM, flash, RAM und DSMCC Datenträgern optimiert. Flashintegrität ist gegen jegliche Störfälle sicher gestellt.

[0139] Sicherheitsverwaltungsprogramm (Security manager). Dieses Modul authentifiziert Anwendungen und steuert den Zugrang von Anwendungen auf empfindliche Speicherzonen und andere Zonen des Beistelldekoder.

[0140] Graphisches System graphics system). Dieses Modul ist objektorientiert und optimiert. Es beinhaltet die Verwaltung von Graphikfenstern und von Gegenständen, ebenso wie eine Vektorzeichensatzmaschine mit Mehrsprachenunterstützung.

[0141] Ferner wird das Modell zur Ressourcenmeldung DAVIC unterstützt, damit kKlientenressourcen wirkungsvoll verwaltet werden.

[0142] Die Geräteschnittstellenschicht **256** beinhaltet ein Bausteinverwaltungsprogramm und Bausteine. Bausteine sind Softwaremodule, die aus logischen Ressourcen, die für die Verwaltung von externen Vorgän-

gen und physikalischen Schnittstellen notwendig sind, bestehen. Die Bausteinschicht verwaltet Übertragungskanäle zwischen Treibern und Anwendungen und liefert verbesserte Prüfung von Fehlerausnahmebedingungen. Einige Beispiel für verwaltete Bausteine sind: Chipkartenlesegeräte, Modems, Netzwerke, PCMCIA (Personal Computer Memory Card International Association), LED Anzeige und so weiter. Programmierer müssen mit diesen Schichten nicht direkt umgehen, da die API-Schicht die obigen Bausteine steuert.

[0143] Die System-Software/Hardwareschicht **258** wird vom Hersteller des Empfänger/Dekoders zur Verfügung gestellt. Wegen des modularen Aufbaus des System und wegen der durch das OS (OS Operating System = Betriebssystem) bereitgestellten Dienste (so wie Vorgangsplanung und Speicherverwaltung), die Teil der virtuellen Maschine sind, sind die höheren Schichten nicht an ein besonderes in Echtzeit arbeitendes System (RTOS) oder an einen besonderen Prozessor gebunden.

Trickfenster-Gruppen

[0144] In einem bevorzugten Ausführungsbeispiel wird eine Trickfenstergruppe zum Einsatz in einer graphischen Benutzerschnittstelle (GUI graphical user interface) bereitgestellt. Eine besondere Anwendung einer solchen Trickfenstergruppe (widget set) ist Trickfenster in einer GUI-Anzeige eines Empfänger/Dekoders für das digitale Fernsehen bereit zu stellen. Jedes Trickfenster ist als objektorientiertes Modul implementiert, so dass für jedes Trickfenster es eine entsprechende Trickfensterklass gibt. Somit kann irgendein Trickfenster aus einfacheren Bausteintrickfenstern durch Übernahme oder Vereinigung von Klassen anderer Trickfenster aufgebaut werden.

[0145] Die **Fig. 3** ist ein vereinfachtes Diagramm der Hierarchie von Trickfenstern innerhalb einer Trickfenstergruppe. Bei diesem Ausführungsbeispiel enthält die Trickfenstergruppe eine Gruppe einfacher Trickfensterklassen **410**, einschliesslich, unter anderem, Fenster- und Dialogboxenrahmen, einen Schieberegler, eine Drucktaste, ein Ankreuzfeld, ein Textfeld und ein Textbearbeitungsfeld. Bei einer nächsten Schwierigkeitsstufe gibt es die Klassen **420**, die mehrere einfache Trickfensterklassen verbinden, oder das verhalten eines einfachen Trickfenster verändern. Ein Trickfenster, zum Beispiel so eines wie ein Verzeichnisfeld kann bearbeitbare Listenpositionen aus einer Textbearbeitungsfeldklasse erzeugen und dem Benutzer ermöglichen durch die Liste zu blättern, in dem er eine Bildlaufleiste, die von einer Schiebereglerklasse abgeleitet wird, benutzt. Bei einer noch höheren Schwierigkeitsstufe enthält die Trickfenstergruppe verknüpfte Trickfenster **430**, wie zum Beispiel eine Dialogbox für die Dateiauswahl, die Druckknöpfe beinhaltet, Listen zum Blättern, Textfelder und Textbearbeitungsfelder, die insgesamt in anderen Klassen von Trickfenstergruppen festgelegt sind.

[0146] Jede Trickfensterklasse setzt Verfahren und Vorgangssteuerungsprogramme zur Steuerung des Trickfensterbetriebs ein. Die Trickfensterklassen können auch Verfahren zum Zeichnen eines Teils der Trickfenster beinhalten. Mit dem Ziel ein bestimmtes Erscheinungsbild oder einen bestimmten "Look" für das Trickfenster bereit zu stellen, beinhalten die Trickfensterklassen Zeichenverfahren einer Betrachtungsgegenstandsklasse, mit der die Trickfensterklasse verknüpft ist. Dies wird weiter unten in weiteren Einzelheiten beschrieben werden.

Erscheinungsbildklassen – Öffentliche Verfahren und API (look class public methods and API)

[0147] Damit die Betrachtungsgegenstandsklassen und Trickfensterklassen zusammen arbeiten können, ist es für die Betrachtungsgegenstandsklassen notwendig, eine konsistente Reihe öffentlicher Methoden zu haben, die garantiert zum Einsatz durch die Trickfensterklasse zur Verfügung stehen. Die Betrachtungsgegenstandsklasse muss insbesondere eine Standard API bereitstellen, die Verfahren enthalten, die die Trickfensterklasse aufrufen kann, um sich selbst auf einer graphischen Benutzerschnittstellenanzeige zu zeichnen.

[0148] Die API, die bei Trickfenstern eingesetzt wird ist in einer Basisklasse bestimmt, aus der alles Ansichten (looks) abgeleitet werden. Die API beinhaltet die folgenden Elemente:

1. Allgemeine Darstellungsverfahren
2. Besondere Darstellungsverfahren
3. Steuerung der Erstellung und Zerstörung von Vorgängen
4. Steuerung der Ränder
5. Steuerung der Änderungen.

[0149] Allgemeine Darstellungsverfahren sind diejenigen, die für alle Trickfenster zur Verfügung stehen, während besondere Darstellungsverfahren bestimmten Typen von Trickfenstern zueigen sind.

[0150] Ansichten werden unter Benutzung einer hierarchischen Architektur gebaut. Eine neue Betrachtungsklasse wird durch Übernahme der Attribute, Verfahren und Standardwerte der Klasse, von der sie abgeleitet wurden, und dann durch Hinzufügen neuer Attribute, Verfahren und Standardwerten, oder durch Überschreiben einiger oder aller Übernommenen erzeugt.

[0151] Eine Betrachtungsklasse wird als Tabelle, die Zeiger für öffentliche Verfahren enthält, organisiert. Eine Betrachtungsklasse die von einer anderen Betrachtungsklasse abgeleitet wurde kann deshalb ein Verfahren durch Wechseln des relevanten Zeigers, so dass er auf ein anderes Verfahren zeigt umdefinieren. Eine Betrachtungsklasse implementiert typischerweise nur einige der verfügbaren öffentlichen Verfahren.

[0152] Bei einer anderen praktischen Anwendung werden Betrachtungsklassen mit Validierungsmasken versehen. Eine Validierungsmaske bestimmt, welche Verfahren durch die Betrachtungsklasse aufgerufen werden können, so das Verfahren, die nicht implementiert sind nicht aufgerufen werden. Ein Trickfenster (widget) kann auf die Validierungsmaske einer Betrachtungsklasse zugreifen, um die Zeichnung des Trickfensters zu optimieren. In diesem Fall ist kennt das Trickfenster die Verfahren, die nicht implementiert sind und so wird das Trickfenster die Erzeugung von Aufrufen derartiger Verfahren vermeiden. Auf diese Art und Weise ist es möglich Zeit durch Aufruf unechter Verfahren zu vergeuden.

[0153] Eine Betrachtungsklasse kann aus zwei oder mehr weiteren Klassen (Mehrfachvererbung) abgeleitet werden. Das kann es gestatten eine Betrachtungsklasse zu schaffen, die eine Verbindung aus zwei oder mehreren anderen Ansichten bildet. Wie oben erwähnt wurde nimmt, wenn eine Ansicht erstellt wurde, diese die Attribute, Verfahren und Standardwerte der Betrachtungsklasse, aus der sie begleitet wurde, an. Um Mehrfachvererbungen zu implementieren, enthält die Ansicht auch einen oder mehrere Zeiger für die zusätzlichen Klassen, aus der sie Attribute, Verfahren und Standardwerte ableitet. Die Ansicht kann dann auf diese Attribute zugreifen, ohne sie selbst kopieren oder selbst erzeugen zu müssen.

[0154] In anderen Ausführungsbeispielen nimmt, wenn eine Ansicht erstellt wurde, diese die Attribute, Methoden und Standardwerte aller Betrachtungsklassen an, von der sie abgeleitet wurden.

[0155] Das Prinzip der Mehrfachvererbung ist auch nützlich in Situationen, wo nicht standardisierte Trickfenster entworfen werden, was die Ansicht veranlassen kann, nicht standardisierte Verfahren einzusetzen. Der Zeiger in einer Ansicht kann auf eine zweite Betrachtungsklasse hinweisen, die nicht standardisierte Verfahren enthält, die zur Darstellung des Trickfensters erforderlich sind.

[0156] Es ist von Bedeutung, dass die verschiedenen Betrachtungsklassen, aus denen eine Ansicht abgeleitet wurde, nicht miteinander in Konflikt geraten. Dies kann durch Sicherstellung von zusätzlichen Betrachtungsklassen, die nur Methoden beinhalten, die sich nicht in der Haupt-Betrachtungsklasse befinden, aus denen die Ansicht abgeleitet wurde, erreicht werden, oder durch Ausgeben einer Rangfolge an die verschiedenen Klassen.

[0157] Ein Beispiel der öffentlichen Verfahren einer Betrachtungsklasse wird unten dargelegt:

```
/*Initialisierung des Vorganges*/
MhwWgtLookInitDefault (MhwWgtLookclass* MhwWgtLookAtts*);
MhwWgtLookInitClass (Void)
MhwWgtLookResetDefault (MhwWgtLookclass*);

MhwWgtLookAttsGetDefault (MhwWgtLookClass*, MhwWgtLookAtts*);
MhwWgtLookAttsInit (MhwWgtLookAtts*);
/*Abfragen und Setzen der Grenzabmessungen*/
MhwWgtLookAttsGetBorderwidthBottom (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsGetBorderwidthLeft (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsGetBorderwidthRight (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsGetBorderwidthTop (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsSetBorderwidthBottom (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsSetBorderwidthLeft (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsSetBorderwidthRight (MhwWgtLookAtts*, Card8*);
MhwWgtLookAttsSetBorderwidthTop (MhwWgtLookAtts*, Card8*);
/*Abfragen und Setzen der Farben*/
MhwWgtLookAttsGetColorBackground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorBlack (MhwWgtLookAtts*, MhwWgtColor*);
```

MhwWgtLookAttsGetColorGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorForeground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorHighlight (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorLightGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorMapAndVisual (MhwWgtLookAtts*, MhwWgtColorMapId*, MhwWgtVisual*);
MhwWgtLookAttsGetColorMiddleGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorTransparent (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorVeryLightGrey (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorWhite (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorBackground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorBlack (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorDarkGrey (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorForeground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorHighlight (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorLightGrey (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorMapAndVisual (MhwWgtLookAtts*, MhwWgtColorMapId, MhwWgtVisual);
MhwWgtLookAttsSetColorMiddleGrey (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorTransparent (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorBackground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorVeryLightGrey (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorWhite (MhwWgtLookAtts*, MhwWgtColor*);
/*Abfragen und Setzen der Übernahmedaten*/
MhwWgtLookAttsGetHeritageData1 (MhwWgtLookAtts*, Void**);
MhwWgtLookAttsSetHeritageData1 (MhwWgtLookAtts*, Void**);
/*Konstruktor*/
MhwWgtLookNew (MhwWgtLookAtts*);
/*Destruktor*/
MhwWgtLookDelete (ein Objekt)
/*Standard API*/
MhwWgtLookDrawAnchor (anObject, aWidget, aX, aY, aW, aH, aText, aLength, anAscent, aState)
MhwWgtLookDrawBackground (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookCheckSymbol (anObject, aWidget, aX, aY, aW, aH, aState aSymbol)
MhwWgtLookDrawChoice (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawCross (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawCursor (anObject, aWidget, aX, aY, aW, anAscent, aH)
MhwWgtLookDrawForeground (anObject, aWidget, ax1 aY1 aW, aH)
MhwWgtLookDrawFocus (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawHighlight (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawInset (anObject, aWidgetl aX, aY, aW, aH)
MhwWgtLookDrawItermr (anObject, aWidget, aX, aY, aW, aH, aTextLength, anAscent, aState)
MhwWgtLookDrawOutset (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawRelief (anObject, aWidget, aX, aY, aW, aH, aRelief)
MhwWgtLookDrawSelectedBG (anObject, aWidget, aX, aY, aW, aH
MhwWgtLookDrawSlidArrow (anObject, aWidget, aX, aY, aW, aH, aDirection)
MhwWgtLookDrawSlidLift (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawString (anObject, aWidget, aX, aY, aText, aLength, anAscent)
MhwWgtLookGetBorderWidth (anObject, aBorder)
MhwWgtLookGetClassIddth (anObject)
MhwWgtLookGetClassName (anObject)
MhwWgtLookGetItemBorderWidth (anObject)
MhwWgtLookGetMethodMask (anObject)
MhwWgtLookGetPreferredSizeArrow (anObject)
MhwWgtLookGetPreferredSizeCheck (anObject)
MhwWgtLookGetPreferredsSizeChoice (anObject)
MhwWgtLookGetPreferredSizeCross (anObject)
MhwWgtLookGetUpdateCounter (anObject)
MhwWgtLookIsInstantof (anObject, aClassId)
MhwWgtLookReDrawItema (anObject, aWidget, ax1 aY, aW, aH, aText, aLength, anAscent, aState)
MhwWgtLookRef (anObject)
MhwWgtLookSetBorderWidther (anObject, aBorder, aWidth)

MhwWgtLookUnDrawCross (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookGetUpdateCounter (anObject)
MhwWgtLookUnDrawCross (anObject, aWifget, aX, aY, aW, aH)
MhwWgtLookUnDrawCursor (anObject, aWidget, aX, aY, anAscent, aH)
MhwWgtLookunDrawFocusr (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookunDrawHighlight (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookUnDrawRelief (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookUnRef (anObject)
MhwWgtLookGetBackground (anObject)
MhwWgtLookGetColorBlack (anObject)
MhwWgtLookGetColorDarkGray (anObject)
MhwWgtLookGetColorHighlight (anObject)
MhwWgtLookGetColorLighGray (anObject)
MhwWgtLookGetColorMap (anObject)
MhwWgtLookGetColorMiddleGray (anObject)
MhwWgtLookGetColorTransparent (anObject)
MhwWgtLookGetColorVeryLightGray (anObject)
MhwWgtLookGetColorWhite (anObject)
MhwWgtLookGetColorForeground (anObject)
MhwWgtLookGetColorHeritageData1 (anObject)
MhwWgtLookGetColorHeritageLink1 (anObject)

Erstellen und Anzeigen eines Trickfensters

[0158] Wenn eine Anwendung verlangt, dass ein Trickfenster auf der Anzeige einer graphischen Benutzerschnittstelle (GUI graphical user interface) erscheint, ist der erste Vorgang, den die Anwendung ausführen muss, das Erstellen einer Instanz der Trickfensterklasse. Während der Bildung der Trickfensterinstanz, wird eine Betrachtungsklasseninstanz mit der Trickfensterklasseninstanz. Das spezielle Aussehen wird wie folgt gewählt:

1. wenn eine Betrachtungsinstanz von der Anwendung an den Konstruktor geleitet wird, dann benutze ihn.
2. Anderenfalls benutze die Standardansicht, die für die zu schaffende Trickfensterklasse spezifiziert ist, wenn es eine gibt.
3. Anderenfalls benutze die Standardansicht, die für den Trickfensterkontext spezifiziert ist, wenn es einen gibt.
4. Anderenfalls benutze die Standardansicht für die Trickfenstergruppe.

[0159] Wenn einmal die Trickfensterklasse realisiert ist, kann die Anwendung eine passende ihrer öffentlichen Methoden (public method) aufrufen, um sie anzuzeigen.

[0160] Die Trickfensterklasse stellt vorzugsweise auch eine öffentliche Methode zur Verfügung, die mit einem Zeiger zu einer Betrachtungsklasseninstanz aufgerufen werden kann und diese Betrachtungsklasseninstanz wird dann mit der Trickfensterklasseninstanz verknüpft. Dies führt dazu, dass sich das Erscheinungsbild des Trickfensters im Einklang mit der neu verknüpften Betrachtungsklassenmethode verändert. Es sollte verstanden werden, dass "Verknüpfung" in Wirklichkeit nichts weiter bedeutet, als den Wert eines Feldes innerhalb der Trickfensterklassenmethode zu setzen. Um das Trickfenster mit einer unterschiedlichen Betrachtungsklasse zu verknüpfen, kann dies, im einfachsten Ausführungsbeispiel, einfach durch Erstellen einer Zuordnung zum Feld gemacht werden. (Siehe dazu jedoch weiter unten die Bemerkungen in Bezug auf die Speicherverwaltung und die Methode MhwWgtXxxSetLook.) Viele Trickfensterklasseninstanzen können jedoch mit einer Betrachtungsklasseninstanz verknüpft werden. Dies ist in Diagrammform in der **Fig. 5** dargestellt.

[0161] Wenn eine Trickfensterklassenmethode aufgerufen wird, um das Trickfenster auf einem Bildschirm der graphischen Benutzerschnittstelle (GUI) darzustellen, baut sie das Bild des Trickfensters in der folgenden Reihenfolge auf:

1. Der Hintergrund des Trickfensters (zum Beispiel eine Hintergrundfarbe oder ein Bild)
2. Hintergrundüberzug (zum Beispiel ein Logo)
3. Vordergrund des Trickfensters
4. Vordergrundüberzug (zum Beispiel ein Logo)
5. Der Rand des Trickfensters
6. Hervorhebung
7. Eingabefeld (input focus = blinkender Eingabe-Cursor)

[0162] Bei einem vorgegebenen Trickfenster können bestimmte Teil fehlen. Das Vorhandensein oder Fehlen eines Teils des Trickfensters hängt von den folgenden Kriterien ab:

1. Hard coding (Kode ist nicht flexibel). Einige Teile sind für bestimmte Trickfensterklassen nicht definiert.
2. Optionale Teile. Zum Beispiel der Fokus, das Relief und das Hervorheben können nach Belieben durch öffentliche Attribute des Trickfensters gesperrt sein.
3. Bestimmung des Aussehens. Ein Aussehen kann ein oder mehrere Teil weglassen.

[0163] In einem typischen Beispiel werden die folgenden Schritte durchgeführt.

[0164] Zuerst wird der Hintergrund durch die Trickfensterklassenmethode selbst gezeichnet, zum Beispiel durch Ausmalen einer Hintergrundfarbe, eines Hintergrundmusters oder eines Bildes. Der Hintergrundüberzug wird dann durch Aufruf der öffentlichen Methode der verknüpften Betrachtungsinstanz MhwWgtLookDrawBackground, in dem geeignete Argumente für die Höhe, die Breite und die Lage des Trickfensters auf dem Bildschirm spezifiziert werden. Das Aussehen des Hintergrundes wird dann zum Beispiel durch das Aussehen durch Überlagerung mit einem Logo verändert.

[0165] Die Trickfensterklassenmethode muss dann den Vordergrund des Trickfensters aufbauen; das ist sozusagen die Schaffung der visuellen Objekte, die gegenwärtig durch einen Benutzer oder durch Bildschirminformationen gehandhabt werden, wenn sich das Trickfenster im Einsatz befindet. Das Trickfenster könnte zum Beispiel ein Ankreuzfeld implementieren, in welchem Fall es die Betrachtungsklassenmethode MhwWgtLookDrawCheckSymbol aufruft. Das Aussehen kann dann den Vordergrund verändern, zum Beispiel durch Überlagerung mit einem Logo.

[0166] Der Rahmenbezirk des Trickfensters wird dann gezeichnet, wie unten beschrieben werden wird.

[0167] Wenn die Trickfensterklasse bestimmt, dass eines der Objekte innerhalb des Trickfensters Eingangsfokus besitzt, ruft es die Betrachtungsklassenmethode MhwWgtLookDrawFocus auf, um dies in dem gezeigten Trickfenster abzubilden. Auf die gleiche Weise ruft die Trickfensterklasse, wenn ein Teil des Trickfensters hervorgehoben werden muss, die Betrachtungsklassemethode MhwWgtLookDrawHighlight auf.

Verwaltung des Trickfensterrahmens

[0168] Ein besonderes Beispiel der Art in der die Ansicht das Erscheinungsbild des Trickfensters auf einem GUI-Bildschirm (GUI = graphische Benutzerschnittstelle) bei der Verwaltung der Rahmen steuert. Das Aussehen eines Trickfensters mit Rahmen in seiner herkömmlichsten Form auf einen GUI-Bildschirm, ist in der Fig. 4 dargestellt. Das Trickfenster **500** belegt ein rechteckiges Gebiet in der Darstellung einer graphischen Benutzerschnittstelle GUI) Das gebiet, welches vom Trickfenster belegt wird, schliesst zwei Bereiche ein: den inneren Anwendungsbereich **510**, der durch einen Randbereich umgeben ist.

[0169] Der Randbereich trägt typischerweise zur Funktion des Trickfensters bei (obwohl es in einigen Fällen von einem Benutzer eingesetzt werden könnte, um das Trickfenster zu bewegen und/oder das Trickfenster in der Größe anpassen möchte). Daher gibt es einen beträchtlichen Umfang an Gestaltungsvarianten des Randbereiches im Einklang mit den Wünschen des Benutzers. Die Farbe, die Breite, das Hintergrundmuster können alle ausgewählt werden, um dem Benutzer zu gefallen und um ein durchgehendes Erscheinungsbild zu schaffen. Somit wird die Verantwortung zum Zeichnen der Ränder an die Betrachtungsklasse gegeben.

[0170] Die Ansicht unterstützt 4 Dimensionen zur Spezifikation der breite der Ränder. Diese spezifizieren den Abstand vom linken, rechten, oberen und unteren des Anwendungsbereichs **510** zur Umrandung des Trickfensters. Diese Dimensionen werden jeweils mit L, R, T, B in der Fig. 4 bezeichnet. Werte dieser Dimensionen werden in der Standardansicht spezifiziert. Eine Anwendung kann eine Klasse definieren, die von einer Standardansicht abgeleitet wurde, in der die Werte überschrieben werden, um eine Ansicht zu schaffen, die ein Trickfenster mit einer Umrandung mit einer nicht standardisierten Breite erzeugt. Eine Anwendung (zum Beispiel eine Ansichtenverwaltung) kann ebenfalls die Werte zur Laufzeit wechseln durch, in dem die Ansichtmethoden MhwWgtLookAttsBorderwidthBottom, MhwWgtLookAttsBorderwidthLeft, MhwWgtLookAttsBorderwidthRight, MhwWgtLookAttsBorderwidthTop aufgerufen werden.

[0171] Innerhalb der Ansichtenklasse gibt es einen Kode, der das detaillierte Layout einer Umrandung, gemäss den Werten, die der Ansichtenklasse über die Trickfensterklasse übermittelt wurden.

Farbverwaltung für Trickfenster

[0172] Eine Ansichtenklasse beinhaltet eine Bestimmung der Farben, so dass eine Trickfensterinstanz, die mit einer bestimmten Ansichteninstanz verknüpft ist, die in dieser Ansichteninstanz definierten Farben einsetzen wird. In einem Ausführungsbeispiel bestimmt eine Ansicht die folgenden Farben:

- schwarz
- dunkelgrau
- mittelgrau
- hellgrau
- sehr helle Grau
- weiss
- durchsichtig
- Farbe hervorheben

[0173] Die Farbenbestimmung in der Ansicht beim Zeichnen eines Trickfensters gebraucht. Wenn zum Beispiel eine schwarze Linie gezogen werden soll beim Darstellen eines Trickfensters, wird die Farbe, die als "schwarz" bestimmt wurde benutzt werden. Wenn die Ansicht zum Beispiel eine rote Farbnuancierung haben soll, dann kann "schwarz" als dunkel definiert werden und "weiss" als hellrosa definiert werden mit verschiedenen roten Schatten, die dazwischen bestimmt werden. Bei diesem Beispiel würde normalerweise ein Zeichenvorgang eine schwarze Linie anstelle einer dunkelroten ziehen, und so weiter.

[0174] Zusätzlich bestimmt die Ansicht eine Farbübersichtstafel, die die aktuellen zu benutzenden Farbwerte beim Anzeigen irgendeiner speziellen Farbe auf dem Bildschirm der graphischen Benutzerschnittstelle (GUI) einsetzt.

Erzeugung eines modifizierten Trickfensters

[0175] Unterstellen wir, dass die Standardansichtmethode MheWgtLookDrawCheckSymbol einen rechteckigen Kasten zeichnet, der entweder leer ist, oder der ein kleines Häkchensymbol enthält in Abhängigkeit seines Zustandes und dass dies die Ansicht eines normalen Eingabefelds in einer graphischen Benutzerschnittstelle (GUI) bestimmt. Unterstellen wir nun, dass ein anderes Trickfenster erforderlich ist, in dem entweder ein Häkchen oder ein Kreuz dargestellt wird. Das Erscheinungsbild wird vollständig durch die Ansichtklasse gesteuert, so muss also nur die Ansichtklasse verändert werden. Darüber hinaus kann eine neue Ansichtklasse zur Implementierung dieses Verhaltens aus der bestehenden Ansichtklasse abgeleitet werden und nur eine Methode MhwWgtLookDrawCheckSymbol zur Verfügung stellen, um die Methode mit demselben Namen in der Basisansichtklasse zu überschreiben. Darüber hinaus wird mit diesem Argument, wenn die Methode MhwWgtLookDrawCheckSymbol aufgerufen wird, ein Zustand auf zutreffend gesetzt und die Methode MhwWgtLookDrawCheckSymbol der Basisklasse kann dann aufgerufen werden, um ein Häkchen zu zeichnen. Ein neuer Kode muss geschrieben werden, nur um den Fall zu behandeln, wo ein Kreuz gezeichnet werden muss. Damit kann ein neues Trickfenster erstellt werden mit einem Minimum an Programmieraufwand.

[0176] Es sollte klar sein, dass dieses Vorgehen das Erscheinungsbild des ursprünglichen Ankreuzfeldesfensters nicht verändert; diese Trickfenster benutzt die Basisansichtklasse, die nicht abgeändert wurde. Um einen Wechsel des Erscheinungsbildes des ursprünglichen Ankreuzfeldfensters in einer Anwendung auszuüben, muss die Methode MhwWgtLookDrawCheckSymbol in der Basisansichtklasse verändert werden. Alle Ankreuzfeldfenster, die von dieser Klasse abgeleitet wurden, werden sich dann ihr Erscheinungsbild bei nächster Gelegenheit verändern (während der Übertragungszeit oder je nachdem während der Laufzeit) nach der die Ansichtklasse dann mit der Anwendung verbunden ist.

[0177] Im Grundsatz wird die Trickfensterklasse zusammenwirken, um mit jeder Klasse ein Trickfenster zu erstellen, die einen geeigneten Satz öffentlicher Methoden und Eigenschaften wie eine Ansichtklasse besitzt. Es gibt jedoch einen Vorteil alle Ansichtklassen aus einer möglichst kleinen Zahl gemeinsamer Basisklassen abzuleiten und idealerweise gerade aus einer einzigen Basisklasse. Diejenigen, die mit objektorientierter Programmierung vertraut sind werden verstehen, dass dies den Einsatz von Speicherplatz und anderer Ressourcen durch die Ansichtklassen minimiert. Eine abgeleitete Klasse besitzt einen Zeiger auf seine Basisklasse, so dass sie auf einen Methodenkode und auf statische Daten der Basisklasse zugreifen kann, ohne eine solchen Kode, oder Daten im Speicher zu verdoppeln.

Versionssteuerung

[0178] Es ist möglich, dass einige Trickfensterinstanzen ein sehr langes Leben haben. Zum Beispiel das Hauptfenster eines Fensterverwaltungsprogramms, ein Trickfenster für eine Taskleiste bei einem Bildschirmarbeitsplatz, und so weiter. In solchen Fällen gibt es eine grosse Wahrscheinlichkeit, dass die Ansichtklassen während der Laufzeit des Trickfensters aktualisiert werden könnten. Man muss die Trickfensterklass dazu bringen sich selbst neu zu zeichnen, wenn dies eintritt.

[0179] Eine Möglichkeit dieses zu erreichen ist, jeder Ansichtsklasse einen Aktualisierungszähler zuzuordnen, der als Gemeingut (public property) exportiert wird oder über eine öffentliche Methode (public method) zugänglich ist. Wenn eine Trickfensterklasse realisiert wird, fragt die Trickfensterklasseninstanz den Aktualisierungszähler der verknüpften Ansicht und speichert den Wert des Aktualisierungszählers im Speicher dezentral zur Trickfensterklasseninstanz. Wenn die Instanz der Ansichtklasse danach aktualisiert wurde, kann die Trickfensterklasseninstanz diesen Wechsel durch Vergleich der Werte mit den Werten des Aktualisierungszählers in der Ansichtsklasseninstanz, die in ihrem lokalen Speicher abgelegt sind, erkennen. Wenn die Ansichtsklasseninstanz aktualisiert wurde, kann das Trickfenster sich neu entwerfen, in dem es die Methoden der Ansichtklasse einsetzt.

Konstruktion und Destruktion der Ansichtklasseninstanzen

[0180] Im Allgemeinen wird es weniger Instanzen jeder Ansichtklasse als von jeder Trickfensterklasse geben. In einigen Fällen kann es gerade eine Instanz eine Ansichtbasisklasse geben, auf die sich alle Trickfensterklasseninstanzen bei einer Anwendung beziehen. Es kann auch Instanzen aus abgeleiteten Ansichtklassen geben, auf die sich einige Trickfensterklasseninstanzen einer Anwendung beziehen. Eine Trickfensterklasse kann jedoch nicht unterstellen, dass es immer eine Ansichtklasseninstanz zu der zeit geben wird, in der die Trickfensterklasse realisiert wird; die Trickfensterinstanz könnte die erste sein, die eine Verknüpfung mit einer speziellen Ansichtklasse erfordert.

[0181] Es wird deshalb vorgeschlagen, dass während der Realisierung jeder Trickfensterklasse, der Trickfensterklassenkonstruktor die verknüpfte Ansichtsklasse MhwWgtLookNew aufruft. Falls keine Instanz der Ansichtsklasse besteht, wird eine neue Instanz geschaffen. Ein Wert von 1 wird dann gespeichert in einem Referenzzähler, der im lokalen Speicher der Ansichtsklasseninstanz gehalten wird. Wenn eine Instanz der Ansichtsklasse bereits besteht, führt der Ansichtsklassenkonstruktor den Zeiger auf sie zurück und erhöht den Referenzzähler.

[0182] Während der Destruktion jeder Trickfensterklasseninstanz, ruft der Destruktor der Trickfensterklasse den Destruktor MhwWgtLookDelete für die verknüpfte Ansichtsklasseninstanz auf. Der Destruktor MhwWgtLookDelete verringert den Referenzzähler. Falls der Zähler grösser als 0 bleibt, läuft der Destruktor ganz einfach zurück. Falls der Destruktor jedoch 0 erreicht, dann sind keine Trickfensterklasseninstanzen (mit Ausnahme der, die der Destruktion ausgesetzt ist) verknüpft mit dieser Ansichtsklasseninstanz, wobei in diesem Fall der Ansichtsklassendestruktor fortfährt, die Ansichtklasseninstanz aus dem Speicher zu löschen.

[0183] Die Trickfensterklassenmethode MhwWgtXxxSetLook kann aufgerufen werden, um die Ansicht, mit der eine spezifische Trickfensterklasseninstanz verknüpft ist, zu wechseln. Innerhalb dieser Methode wird zuerst der Destruktor der abgehenden Ansichtsklasseninstanz aufgerufen und dann wird ein Aufruf an die Bezugsfunktion der neuen Ansichtsklasse gemacht, um einen Zeiger zu einer Klasseninstanz zu erhalten. Dies stellt sicher, dass die Bezugszähler der Ansichtklassen korrekt aktualisiert werden.

[0184] Es müssen auch Vorkehrungen für eine neu zu schaffende Instanz eine Ansichtsklasse getroffen werden, selbst wenn bereits eine Instanz besteht. Dies gestattet es einer Anwendung mehr als eine Instanz irgendeiner gegebenen Ansichtsklasse zu haben und verschiedene Attribute in den verschiedenen Instanzen zu setzen. Es können zum Beispiel zwei Instanzen der gleichen Ansichtsklasse, die in jeder Richtung identisch sind, vorhanden sein, ausser eine besitzt alle verbleibenden Attribute in Übereinstimmung mit ihren Standardeinstellungen und die anderen besitzen unterschiedliche Werte, die einem oder mehreren ihrer Attribute zugeordnet sind (Randbreite, Farbe und so weiter).

Ansichtenverwaltungsprogramm

[0185] Es versteht sich, dass das System der Ansichtsklassen und Instanzen eine sehr genaue Steuerung über die Gesamtansicht einer Anwendung erlaubt. Nur ein Attribut irgendeiner Ansichtsklasse kann zum Bei-

spiel verändert werden, um einen geringfügigen Wechsel der Ansicht einer Anwendung zu bewirken. Dementsprechend kann eine Ansichtenverwaltungsprogrammanwendung (look manager) bereitgestellt werden, um dem Benutzer zu erlauben, diese Attribute, wie erforderlich, zu verändern. Eine derartige Anwendung schliesst typischerweise eine GUI-Anzeige (graphical user interface) mit ein, die Trickfenster beinhaltet, die die Erfindung verkörpern, damit der Benutzer sofort die Auswirkung durch den Wechsel der Attribute der Ansicht beim Erscheinen des Trickfensters erkennen kann.

Web Browser

[0186] Die Internetnavigatorschnittstelle wird nun beschrieben unter Bezug auf die beigefügten Zeichnungen.

[0187] Die Fig. 6a zeigt eine Bildschirmaufnahme des Hauptbildes der Navigatoranzeige eines Internetbrowsers. Das Hauptbild zeigt eine vertikale Kette **1100**, das das Hauptmenü umfasst und verschiedene Schaltflächen als verknüpftes Verzeichnis beinhaltet. Die Schaltflächen sind mit Verbindungselementen der Kette verbunden. Die in der Kette dargestellten Schaltflächen **1100** der Fig. 6a beinhalten eine NEU LADEN/STOP Schaltfläche **1110**, eine VORHER Schaltfläche **1120**, eine WEITER Schaltfläche **1130**, eine VERLAUF Schaltfläche **1140**, eine LESEZEICHEN Schaltfläche **1150**, EINSTELLUNGEN Schaltfläche **1160** und die BEENDEN Schaltfläche **1170**.

[0188] Die Hauptmenükette **1100** ist so angeordnet, um das HTML Dokument (HTML hyper-text markup language = Hypertext-Auszeichnungssprache), das auf dem Bildschirm **1101** dargestellt werden soll, zu überlagern. In der Fig. 6a wird kein HTML Dokument gezeigt und der Bildschirm **1101** ist, abgesehen von der Hauptmenükette, leer.

[0189] Der Web Browser schliesst einige Präferenzen, die vom Benutzer gesetzt werden können, ein. Der Browser schliesst eine Möglichkeit zum bestimmen mehrere Benutzerprofile ein.

[0190] Der Benutzer hat eine Steuerung mit der er zwischen Objekten auf dem Bildschirm **1101** navigieren, Objekte hervorheben und Objekte wählen kann. Im vorliegenden Beispiel ist die benutze Steuerung eine Fernbedienung **1180**. Die Zahlentasten **1181** werden zur Eingabe von Daten benutzt; das Schreibmarkentastenfeld (cursor keypad) **1182** wird zu Navigieren und die Bildfläche herum benutzt. Das Schreibmarkentastenfeld **1182** beinhaltet eine AUF Taste **1183**, eine AB taste **1184**, eine LINKS Taste **1185**, eine RECHTS Taste **1187**. Das Schreibmarkentastenfeld **1182** beinhaltet auch eine WAHL Taste **1186** mit der Objekte auf dem Bildschirm angewählt werden können.

[0191] Die AUF Taste **1183** und die AB Taste **1184** werden zur Bewegung des Blickfeldes (focus) verwendet, in diesem Beispiel eine Hervorhebung nach oben und unten in der Kette **1100**, um einzeln die Schaltflächen **1110**, **1120**, **1130**, **1140**, **1150**, **1160** und **1170** hervorzuheben. Wenn eine Schaltfläche hervorgehoben wurde, kann sie durch Benutzen der Taste **1186** angewählt werden.

[0192] Wenn eine HTML-seite angezeigt wird, ruft jede Taste der Fernbedienung die Kette **1100** (Symbolleiste) auf. Die Kette **1100** kann auch durch den Benutzer ein-und ausgeblendet werden. Bei einer Einstellungsoption wird die Kette **1100** automatisch versteckt, wenn eine HTML Seite angezeigt wird und der Benutzer wählt AUF in der Kette **110**, wenn er zu einer anderen HTML-Seite wechseln will.

[0193] Die Fig. 7 zeigt die Bildschirmanzeige der Fig. 6a mit einem geöffneten HTML Dokument. Informationen zu dem offenen Dokument werden im Textfeld **1112** gegeben, das in der Kette mit der Schaltfläche NEU LADEN/STOP **1110** verknüpft ist. Man kann erkennen, dass die Verbindungen mit der Kette **1114** zwischen den Schaltflächen dem Benutzer visuell anzeigen, dass er sich zwischen den Schaltflächen in der Richtung der Verbindungen hin und her bewegen kann.

[0194] Die Fig. 6a zeigt die hervorgehobene NEU LADEN/STOP Schaltfläche (das hervorgehobene Symbol (icon) für NEU LADEN/STOP ist weiss auf dunklem Hintergrund anstelle von dunkel auf weissem Hintergrund wie in der Fig. 7, wo es nicht hervorgehoben ist). Das HTML Dokument kein durch Drücken des WAHL Knopfes **1186**, wenn die NEU LADEN/STOP Schaltfläche hervorgehoben ist, neu geladen werden.

[0195] Der Benutzer bewegt die Markierung (highlight) die Kette **1100** nach unten unter Benutzung des AB Knopfes **1184**. In der Fig. 7 wird dann die VORHER Schaltfläche **1120** hervorgehoben. Die Fig. 8 zeigt, wenn die Markierung auf der VORHER Schaltfläche **1120** ist, erscheint ein "Tooltip" (Quickinfo) einschliesslich einer Textbox **1122** auf dem Bildschirm. Im vorliegenden Beispiel erscheint der Tooltip sobald das entsprechende

Symbol hervorgehoben wird. Eine Präferenz kann eingestellt werden, so dass der Tooltip nach einer Verzögerung erscheint, wenn die Schaltfläche hervorgehoben wird. Die Textbox **1122** schliesst das Wort VORHER ein, um die Funktion der VORHER Schaltfläche **1120** anzuzeigen. Durch Betätigen der VORHER Schaltfläche durch Drücken des WAHL Knopfes **1186** bewegt sich der Browser zu der zuvor angezeigten Seite.

[0196] In der Fig. 9 wird die Markierung abwärts zur WEITER Schaltfläche **1130** bewegt und nach kurzer zeit erscheint ein Tooltip einschliesslich einer Textbox **1132** mit dem Wort "WEITER", um den Benutzer zu unterstützen.

[0197] In der Fig. 10 wird die Schaltfläche VERLAUF **1140** hervorgehoben und der damit verbundene Tooltip **1142** erscheint mit dem Wort "VERLAUF". Die VERLAUF Schaltfläche **1140** hat mehr als nur eine Funktion und so ruft beim Aktivieren des Knopfes WAHL **1186** auf dem Steuertastenfeld eine Unterkette **1144** hervor, die weitere Optionen in Bezug auf die VERLAUF Funktion anbietet. Die Unterkette **1144** ist in der Fig. 11 dargestellt. Die Unterkette **1144** schliesst zusätzliche Schaltflächen, einschliesslich einer ANZEIGE Schaltfläche **1146** und einer HINZUFÜGEN Schaltfläche **1148** ein. Der Benutzer bewegt sich entlang der Unterkette **1144**, in dem er die Knöpf RECHTS und LINKS **1187**, **1185** benutzt. Auf der Bildfläche der Fig. 11 ist die Schaltfläche ANZEIGE hervorgehoben und ein Tooltip (Quickinfo) **1147** erscheint, um dem Benutzer mitzuteilen, dass die Schaltfläche ANZEIGE hervorgehoben wurde. Es ist anzumerken, dass die Tooltips **1147**, die mit der Hauptkette **1100** verknüpft sind, auf der rechten Seite der Kette **1100** erschienen sind; die Tooltips für die Unterkette erscheinen über der Unterkette.

[0198] Die Grösse des Kastens für den Tooltip ist an die Länge des Wortes oder der Wörter, die angezeigt werden, angepasst. Wo verschiedene Sprachpräferenzen eingestellt werden können, wird die Grösse des Kastens für den Tooltip vorzugsweise an die Länge des Wortes in der gewählten Sprache angepasst.

[0199] Die Fig. 12 zeigt die Bildschirmanzeige wie man sie erhält, wenn die ANZEIGE Schaltfläche ausgewählt wurde. Ein ANZEIGE VERLAUF Fenster **1141** erscheint auf dem Bildschirm mit einer Überschrift **1143** (hier wurde die französische Sprachoption für die Überschrift ausgewählt) und zeigt Einzelheiten **1145** von vorangegangenen Seiten, die vom Benutzer betrachtet wurden. Der Benutzer kann im Text nach oben und nach unten blättern und eines der Details **1145** hervorheben, indem er den Knopf WAHL **1186** drückt. Die Knöpfe RECHTS und LINKS werden zum Hervorheben der OK oder LÖSCHEN Knöpfe **1149**, **1149'** benutzt.

[0200] Die Fig. 13 zeigt die hervorgehobene HINZUFÜGEN Schaltfläche **1148** und ihren verknüpften Tooltip (Quickinfo). Die HINZUFÜGEN Schaltfläche wird benutzt, um die gegenwärtig gezeigte Seite der Verlaufsliste hinzuzufügen.

[0201] Die Fig. 14 zeigt die hervorgehobene LESEZEICHEN Schaltfläche **1150** und den mit ihr verknüpften Tooltip **1151**. In der Fig. 15 wurde die LESEZEICHEN Schaltfläche **1150** ausgewählt und die LESEZEICHEN Unterkette **1152** ist einschliesslich des Tooltips **1151**, der ANZEIGE, HINZUFÜGEN und EDITIEREN Schaltflächen **1153**, **1154**, **1155**, **1156** dargestellt. In der Fig. 15 ist die ANZEIGE Schaltfläche **1153** und ihr Tooltip dargestellt. Die ANZEIGE Schaltfläche **1153** ist ausgewählt und das ANZEIGE Fenster **1157** wird gezeigt (siehe Fig. 16) (Man kann erkennen, dass, aus Gründen der Übersichtlichkeit, die Lesezeichen Unterkette **1152** nicht dargestellt wird, wenn das Anzeige-Fenster **1157** auf der Bildfläche erscheint). Das ANZEIGE Fenster **1157** beinhaltet eine Überschrift und eine Textbox zum Blättern und listet die Lesezeichen auf. Was das ANZEIGE/VERLAUF Fenster anbetrifft, so schliesst das Fenster die OK und LÖSCHEN Tasten ein. Das Schreibmarkentastenfeld **1182** wird eingesetzt, um um das Fenster herum zu navigieren und ein Lesezeichen auszuwählen, wenn erwünscht.

[0202] In der Fig. 17 ist die HINZUFÜGEN Schaltfläche hervorgehoben und ihr Tooltip ist dargestellt. Wenn die HINZUFÜGEN Schaltfläche ausgewählt wird, wird das HINZUFÜGEN Fenster gezeigt (sieh Fig. 18). Das HINZUFÜGEN Fenster **1158** beinhaltet zwei Kasten zur Texteingabeund zur Eingabe der URL des Lesezeichens und ihren Titel. Daten werden unter Verwendung der Nummerntasten **1181** eingegeben (zum Beispiel unter Benutzung des Bildschirmtastenfeldes, das hier beschrieben wird). Das Fenster schliesst auch die OK und LÖSCHEN Tasten, wie oben beschrieben, ein. Der Benutzer navigiert zwischen den Texteingabefeldern und den OK und LÖSCHEN Tasten, in dem er den Kursor **1182** (Schreibmarke) benutzt.

[0203] Die Fig. 19 zeigt die hervorgehobene LÖSCHEN Schaltfläche **1155** und ihren Tooltip. Durch Auswahl der LÖSCHEN Schaltfläche **1155** können Lesezeichen gelöscht werden.

[0204] Die Fig. 20 zeigt die hervorgehobene EDITIER Schaltfläche **1156** und ihren Tooltip. Durch Auswahl

der EDITIER Schaltfläche **1156** können Lesezeichen editiert werden.

[0205] Die Fig. 21 zeigte eine wechselnde Form der Lesezeichen Unterkette **1152**, bei der der Lesezeichen Tooltip **1151** nicht dargestellt wird. Dies kann Platz auf dem Bildschirm sparen, insbesondere wenn die Unterkette lang ist. Das Erscheinungsbild des Tooltips ist eine Option die vom Benutzer gewählt werden kann.

[0206] Die Fig. 22 die hervorgehobene Schaltfläche EINSTELLUNGEN **1160** und ihren Tooltip **1161**. Wenn EINSTELLUNGEN ausgewählt wird, wird das Authentifizierungsfenster **1165** dargestellt (siehe Fig. 22), das den Benutzer auffordert sich zu identifizieren und das Benutzerpasswort einzugeben, bevor die Einstellungen geändert werden können. Das Authentifizierungsfenster **1165** schliesst zwei Texteingabefelder für die Eingabe des Benutzernamens und des Passwortes ein, die mit den Nummerntasten **1181** und den tasten OK und LÖSCHEN eingegeben werden können. Wenn einmal der korrekter Benutzername und das Passwort in das Authentifizierungsfenster **1165** eingegeben worden sind und die OK Taste gedrückt wurde, wird die Unterkette der Einstellungen **1162** angezeigt, siehe dazu die Fig. 24

[0207] Die Unterkette **1162** der Einstellungen schliesst eine MODEM Schaltfläche **1163** und eine BROWSER Schaltfläche **1164** ein. Die Fig. 24 zeigt die hervorgehobene MODEM Schaltfläche **1163** mit dem verknüpften Tooltip. Durch Auswahl der der MODEM Schaltfläche **1163** können die Einstellungen für das Modem geändert werden. Die Fig. 25 zeigt die hervorgehobene BROWSER Schaltfläche **1164** und der dazugehörige Tooltip. Wenn die BROWSER Schaltfläche **1164** ausgewählt wurde, wird das Browserfenster **1166** angezeigt, siehe dazu die Fig. 26. Wieder navigiert der Benutzer um die Objekte im Fenster mit den Kursortasten **1182** herum. Die Objekte im Browserfenster schliessen eine Klappfenstertabelle **1167** für FARBE. Durch hervorheben der Tabellenüberschrift und durch Wahl mit den Kursortasten **1182**, erscheinen die Elemente in der Tabelle und der Benutzer kann den Kursor in der Tabelle nach oben und nach unten bewegen und eine neue Browserfarbe mit dem Kursor wählen. Auf ähnliche Weise kann die Textsprache des Browsers in der Klappfenstertabelle **1168** gewechselt werden. Durch Bewegen des Hervorhebungszeichens auf Tooltipauswahl und durch Drücken der Wahltaste **1186**, kann der Tooltip ein- und ausgeschaltet werden. Das Fenster schliesst wie zuvor die OK und WAHL Schaltflächen ein.

[0208] Die Fig. 27 zeigt die hervorgehobene VERLASSEN Schaltfläche **1170** und den dazugehörigen Tooltip **1171**. Wenn die VERLASSEN Schaltfläche **1170** ausgewählt wurde, erscheint die Unterkette für VERLASSEN **1172** (siehe dazu die Fig. 28). In der Fig. 28 ist die BESTÄTIGEN Schaltfläche **1173** hervorgehoben und der dazugehörige Tooltip wird angezeigt. Wenn der Benutzer den Webbrowser verlassen möchte, wählt er die BESTÄTIGEN Schaltfläche **1173**. Wenn der Benutzer die Auswahl für VERLASSEN löschen möchte, dann wählt er die LÖSCHEN Schaltfläche **1174** aus, die in der Fig. 20 mit dem Tooltip hervorgehoben ist.

[0209] Alternative Gestaltungen der Hauptmenükette **110** können eingesetzt werden, wobei die Form der einzelnen Schaltflächen und die Textur verändert werden können, vorzugsweise ohne die Gesamtform der Kette zu verändern. Die Farbe kann ebenfalls geändert werden. Die Veränderungen können als Optionen im EINSTELLUNGEN – Menü verfügbar gemacht werden, wobei die Form und die Textur der Kette als Ganzes geändert werden, um dem Browser ein einheitliches Aussehen (Oberfläche) zu verleihen.

[0210] Die Schaltflächen können zum Beispiel eckig, rund, rautenförmige sein oder auch andere Formen haben, vorzugsweise mit einer Textur, um eine dreidimensionales Erscheinungsbild zu erzeugen. Das Erscheinungsbild der Verknüpfungen zwischen den Schaltflächen können so gestaltet werden, dass sie mit dem Erscheinungsbild der Schaltflächen übereinstimmen oder alternativ diese ergänzen. Das Erscheinungsbild der Verknüpfungen kann auch so gewählt werden, dass sie eine strukturelle Einheitlichkeit ausstrahlen, um die Wahrnehmung des Benutzers zu erhöhen und zu ihm zu zeigen, dass die Schaltflächen sinnvoll untereinander verknüpft sind.

[0211] Das bogenförmige Aussehen der Kette, wie sie in den **Fig.** 6–Fig. 30 gezeigt wird, wird vom Entwickler gewählt und ist vorzugsweise vom Benutzer nicht zu verändern. Andere Konfigurationen der Schaltflächenkette und der Unterketten sind möglich, so wie zum Beispiel eine gerade Kette, oder eine halbkreisförmige Kette. In anderen Ausführungsbeispielen kann die Schaltflächenfolge und Konfiguration der Schaltflächenkette und der Unterketten durch den Benutzer verändert werden.

[0212] Der Empfänger/Dekoder ermöglicht Internetnavigation und auch das Lesen von Emails.

[0213] Nun soll die die Graphikwerkstatt zum Modellieren eines Navigators behandelt werden.

[0214] Die Graphikwerksatt zur Modellierung eines Navigators ist eine Einheit oder eine Sammlung von elementaren Graphikobjekten. Jedes Graphikobjekt ist die bildliche Darstellung einer der Funktionen des Navigators auf einem Bildschirm. Jede Funktion des Navigators kann durch ein Graphikobjekt dargestellt werden, oder durch eine Abfolge von Bildern oder Graphikobjekten (ein Animation), oder einer Ansammlung von Graphikobjekten (zum Beispiel ein Bild im Hintergrund der Bildschirmfläche oder ein Bild im Hintergrund eines Dialogfensters dem andere Graphikobjekte hinzugefügt werden können). Es gibt zwei interne Formate für Bilder: MPEG2 und PIXMAP-GRL.

[0215] Das PNG Format wird bei elementaren Graphikobjekten, die die Funktionalität des "Navigationssystems" darstellen, eingesetzt: Laden, Verbinden, Vorheriges Dokument, Nächstes Dokument, Verlassen, und so weiter.

[0216] Um ein nicht reckeckiges Bild auf der Graphikebene zu drucken, ist es notwendig eine Ausschnittmaske zu verwenden, die die sichtbaren (bedeutenden) Zonen bestimmt. Diese Maske muss vom Designer in Form einer Bitmap zur Verfügung gestellt werden: diese Maske wird aus Gründen der Leistungsbeschränkung nicht vom Programm geplant.

[0217] Die beiden Stufen zum Hinzufügen einer Ausschnittsmaske ist zuerst die Stufe, wo ein Bild dargestellt wird und dann auf der graphischen Ebene, das Ausfüllen eines Rechtecks der gleichen Abmessung und Position des Bildes mit der Folienfarbe, während die Ausschnittmaske angewendet wird, um den nutzbaren Teil des Bildes sichtbar zu machen.

[0218] Das PIXMAP-GRL Bildformat wird bei Graphikobjekten, die die Navigatorressourcen oder die Benutzerschnittstelle anzeigen, verwendet: vertikale Bildlaufleiste, Tabellen, Einfachwahl, Mehrfachwahl, und so weiter.

[0219] Die Objekte vom PIXMAP-GRL Typ haben variable Abmessungen (jedes Graphikobjekt oder Modell ist in einfache, elementare Objekte zerlegt) und kann mit Farben versehen werden (wechselnde Farben).

[0220] Das PIXMAP-GRL Bildformat kann man erhalten, in dem man irgendwelche anderen Graphikformate, die bestens bekannte Verfahren (wie zum Beispiel BMP, JPEG, GIF, PNG, und so weiter) einsetzen, konvertiert.

[0221] Die Zerlegung von Graphikobjekten in graphische Elemente wird nach einer Matrix durchgeführt (zum Beispiel 3 × 3, 4 × 4, oder 1 × 4), was von dem dazustellenden Objekttyp abhängt.

[0222] Die Palette enthält 256 Farben. Diese Palette wird bei Graphikobjekten des Typs PIXMAP und PNG eingesetzt. In einer Palette gibt es zwei Teile. Der erste Teil besteht aus 26 Farben, die zur Erleichterung der Darstellung und dem Design der Anwendungen vorgesehen sind. Der zweite Teil besteht aus 230 nicht veränderbaren Farben, die zum Einsatz bei Anwendungen zur Verfügung stehen.

[0223] Die maximale Grösse des Bildschirms beträgt 720 Pixel in der Breite und 576 in der Höhe. Um die Sicht bei einfachen Fernsehgeräten sicher zu stellen, ist es notwendig die Grösse auf 592 Pixel in der Breite und 480 Pixel in der Höhe zu beschränken. Damit hochwertige Fernsehgeräte voll genutzt werden können, wird der Benutzer die Option zur Anpassung der Grösse des Bildschirms haben. Bei Internetnavigatoren sind die Seiten im Allgemeinen mit 600 Pixeln in der Breite und 400 Pixeln in der Höhe konzipiert.

[0224] Die allgemeinen Attribute eines Graphikobjekts werden nun im Einzelnen erörtert.

[0225] Ein Graphikobjekt (gemäß dessen, was es darstellt) hat eine genaue Grösse. Die genaue Grösse wird vom Designer bestimmt und dient als Leitlinie für die Darstellung.

[0226] Jedes Graphikobjekt kann in der Grösse angepasst werden. Je nach Art des Graphikobjekts ist es möglich die Höhe und die Breite anzupassen. Das Verfahren zur Anpassung der Grösse eines Graphikobjekts nach PIXMAP-GRL folgt den Empfehlungen der Zerlegung des Graphikobjekts. Das Verfahren zum Zeichnen von Graphikobjekten mit variabler Grösse wird später erörtert werden.

[0227] Das Bild des Graphikobjekts auf dem Bildschirm wird durch eine mehrfarbige Form wiedergegeben; das Hintergrundbild sollte wenn möglich einen Farbbereich haben (flüssig, der Effekt von Knetmasse).

[0228] Das Bild des Graphikobjekts muss nicht nach seiner Position (Koordinaten), oder nach der Reihenfolge des folgenden oder vorhergehenden Objektes auf dem Bildschirm, gezeichnet werden: der Begriff einer fliessenden Position des Objekts. Jedes Objekt ist definitionsgemäss von anderen Objekten unabhängig (ausser einem Hintergrundbild).

[0229] Der Text wird durch das Programm gemäss der gewählten Sprache gedruckt. Dies unterstellt, dass kein Bild Text enthalten sollte. Die Gestaltung eines Objektes liegt in den Händen des Designers. Das allgemeine Erscheinungsbild kann einem spezifischen Thema entsprechen (zum Beispiel Startrek, 007, die Simpsons).

[0230] Der Aspekt des Blickfelds (focus) kann durch mehrere Mittel dargestellt werden: ein rechteckiger Fokus auf dem Graphikobjekt; einem Fokus der den Hintergrund (mit einer anderen Farbe) des Graphikobjekts hervorhebt, oder mit einem Fokus der die Form des Graphikobjekts färbt.

[0231] Der Normalzustand (ohne Fokus, aktiv, nicht gedrückt) ist die Basis des Graphikobjekts.

[0232] Der gesperrte Zustand eines Graphikobjekts kann durch verschieden Mittel dargestellt werden: Die form des Objekts in grau (oder mit Zierleiste); Überlegen mit einem markanten Verbotszeichen auf dem in Frage stehenden Graphikobjekt; Einstellen des Hintergrunds des Objekts auf eine Farbe oder das Objekt unsichtbar machen.

[0233] Der unterdrückte Zustand des Graphikobjekts ist die graphische Darstellung eines Objekts das den Fokus aufgrund eines Klicks erhält, bevor jedoch die Schaltfläche freigegeben ist. Die Darstellung kann eine umgekehrte Darstellung des Objekts sein, oder es kann derselbe wie der fokussierte Zustand sein.

[0234] In Bezug auf einen Flip-Flop Darstellungseffekt kann ein Bild oder ein Piktogramm (icon) zwei Sichteffekte beinhalten: einen der den Text darstellt (zum Beispiel die Hinterseite einer Euromünze), der andere, der ein Symbol zeigt (die Vorderseite oder Kopfseite). Dieser visuelle Effekt wird animiert durch ein Programm, das über einem Taktgeber läuft; ein Taktgeber wird gestartet bei der Darstellung des ersten Piktogramms und sobald der Taktgeber den vorbestimmten Wert erreicht, findet der Wechsel des Piktogramms statt: entweder wird ein zweites Piktogramm oder eine Folge von Piktogrammen, die einen progressiven Übergang darstellen, gezeigt.

[0235] Nun wird die Zerlegung eines Graphikobjekts des Typs PIXMAP-GRL beschrieben. unter Bezug auf die Fig. 31 bis Fig. 38. Diese Figuren zeigen Beispiele der elementaren graphischen Objekte, die bei dem Matrix- (**1201**–**1209**), (**1211**–**1219**, und so weiter) eingesetzt werden und dem entsprechenden Graphikobjekt, das gebildet wird, wenn die elementaren Objekte in der passenden Art und Weise (**1210**, **1220**, und so weiter) kombiniert werden. Die Figuren wurden ungefähr mit Faktor 4 vergrössert.

[0236] Um ein Graphikobjekt in der Grösse anzupassen (vergrössern oder verkleinern), wird jedes Graphikobjekt (ein Design, welches vom Graphikkünstler erstellt wurde) graphische Elemente in Matrixform wie Puzzleteile aufgeteilt. Jedes Element wird dann gekennzeichnet entsprechend der 4 Haupteckpunkte und dem Zentrum (Nord, Süd, West, Ost, Zentrum, Nord-Ost, Nord-West, Süd-Ost. Süd-West. Mittelpunkt des Zentrums). Die Breite und Höhe der Matrix hängen von der Art des Objekts ab.

[0237] Bestimmte graphische Elemente (Puzzleteile) werden auf einmal gedruckt (die Ecken). Um das Objekt breiter oder höher zu machen, werden bestimmte Elemente auf wiederholte Art und weise gedruckt (n Mal Breite oder Höhe des jeweiligen Elements).

[0238] Graphikobjekte, die durch das Matrixzerlegungsverfahren (oder durch Fliesen) gebildet wurden, oder einen teil davon bilden, werden nun aufgelistet. Diese Graphikobjekte werden in dem HTML-Bereich gezeichnet.

– Schaltflächenbreite/ohne Text im aktiven Zustand (**1210**): 3 × 3 Matrix (**1201**–**1209**); Grösse der Elemente: 4 Pixel breit und hoch; Elemente zur Anpassung der Breite: Nord – Zentrum (**1202**), Zentrum (**1205**), Süd – Zentrum (**1208**); Elemente zur Anpassung der Höhe: West – Zentrum (**1204**), Zentrum (**1215**), Ost – Zentrum (**1216**).

– Schaltfläche mit/ohne Text im inaktiven, im grau gefärbten Zustand (**1230**): 3 × 3 Matrix (**1221**–**1229**); Grösse der Elemente: 4 Pixel breit und hoch; Elemente zur Anpassung der Breite: Nord – Zentrum (**1222**), Zentrum (**1225**), Süd – Zentrum (**1228**); Elemente zur Anpassung der Höhe: West – Zentrum (**1224**), Zentrum (**1225**), Ost – Zentrum (**1226**).

– "Ankreuzfeld", dargestellt mit/ohne Fokus und durchkreuzt/nicht durchkreuzt (Fig. 34): 1 × 1 Matrix; Grösse der Elemente: 16 Pixel breit und hoch; Elemente zur Anpassung der breite: keine; Elemente zur Anpassung der Höhe: keine.
– Liste der Optionen, für eine Einzel- oder Mehrfachwahl (**1252**, **1253**, **1254**): 3 × 3 Matrix (**1241**–**1249**); Grösse der Elemente: 4 Pixel breit und hoch; Elemente zu Anpassung der Breite: Nord – Zentrum (**1242**), Zentrum (**1245**), Süd – Zentrum (**1248**); Elemente zur Anpassung der Höhe: West – Zentrum **1244**), Zentrum **1245**), Ost – Zentrum (**1246**); kann einen Auf (**1250**) und/oder ab Indikator (**1251**) beinhalten; Position: x. y Ausgangspunkt + Breite und Höhe des Auf Indikators. Auf Indikator (**1250**): 1 × 1 Matrix, Grösse der Elemente: 16 Pixel breit und 8 Pixel hoch; Elemente zur Anpassung der Breite: keine; Elemente zur Anpassung der Höhe: keine; Position; y Ausgangspunkt, in der Höhe zentriert.

[0239] Ab Indikator (**1251**): 1 × 1 Matrix; Grösse der Elemente: 16 Pixel breit und 8 Pixel hoch; Elemente zur Anpassung der Breite: keine; Elemente zur Anpassung der Höhe: keine; Position: y Ausgangspunkt + Höhe des Auf Indikators + Höhe der Liste, in der Höhe zentriert.
– Tabelle (zum Zeichnen der Formen) (**1260**) 3 × 3 Matrix (**1270**); Grösse der Elemente: 2 Pixel breit und hoch; Elemente zur Anpassung der Breite: Nord – Zentrum, Zentrum, Süd – Zentrum; Elemente zur Anpassung der Höhe: West – Zentrum, Zentrum, Ost – Zentrum; kann eine Zelle beinhalten.
– Zelle (zum Schreiben eines Eintrags in ein Formular) (**1280**) 3 × 3 Matrix (**1290**); Grösse der Elemente: 2 Pixel breit und hoch; Elemente zur Anpassung der Breite: Nord – Zentrum, Zentrum, Süd – Zentrum; Elemente zur Anpassung der Höhe: West – Zentrum, Zentrum, Ost – Zentrum; Kann Text oder ein Bild beinhalten; Position: x, y Ausgangspunkt + Dicke des Tabellenrahmens.
– Text mit Rahmen (Textfeld) (**1300**) 3 × 3 Matrix (**1310**); Grösse der Elemente: 2 Pixel breit und hoch; Elemente zur Anpassung der Breite: Nord – Zentrum, Zentrum, Süd – Zentrum; Elemente zur Anpassung der Höhe: West – Zentrum, Zentrum, Ost – Zentrum; kann Text beinhalten.
– Rahmen (**1320**) 3 × 3 Matrix (**1325**); Grösse der Elemente: 4 Pixel breit und hoch; Elemente zur Anpassung der Breite: Nord – Zentrum, Zentrum, Süd – Zentrum; Elemente zur Anpassung der Höhe: West – Zentrum, Zentrum, Ost – Zentrum.
– Vertikale Bildlaufleiste (**1330**) 1 × 3 Matrix (**1270**); Grösse der Elemente: 8 Pixel breit und hoch; Elemente zur Anpassung der Breite: keine; Elemente zur Anpassung der Höhe: Zentrum; mit dem Rahmen verbunden (abhängig von der Position des Rahmenobjekts); beinhaltet das Indikatorgraphikobjekt des Indexes im Verhältnis zur Höhe.
– Horizontale Bildlaufleiste (**1340**) 3 × 1 Matrix; Grösse der Elemente: 8 Pixel breit und hoch; Elemente zur Anpassung der Höhe: keine; Elemente zur Anpassung der Breite: Zentrum; mit dem Rahmen verbunden (abhängig von der Position des Rahmenobjekts); beinhaltet das Indikatorgraphikobjekt des Indexes im Verhältnis zur Breite.
– Horizontale Linie: 1 × 1 Matrix, Grösse der Elemente: 4 Pixel breit und hoch.
– Vertikale Linie: 1 × 1 Matrix, Grösse der Elemente: 4 Pixel breit und hoch.

[0240] Eine Zusammenfassung aller Graphikobjekte in der Web-Browserschnittstelle und der mit ihr verknüpften Funktion werden nun aufgeführt.

[0241] Was nun folgt ist eine nicht erschöpfende Aufstellung der Graphikobjekte, die zur Konstruktion eines Navigatormodells im Dekoder notwendig sind. Die Tabelle, die hier gezeigt wird, führt die Objekte Element für Element und die Liste der Objekte, die aus mehreren graphischen Elementen zusammengesetzt sind, auf.

Elementare Graphikobjekte	Funktionalität	Type	Kommentare
BTN_Connecter	Verbinden / trennen des Modemanschlusses	Anklickbares PNG Bild	Startet die Modemverbindung mit der definierten Profilinformation
BTN_Connecter	Verbinden / trennen des Modemanschlusses	Anklickbares PNG Bild	Schaltet das Modem ab
BTN_Information_Profil	Zugang zur Authentifizierungsinformation für das Verbindungsprofil	Anklickbares PNG Bild	Zeigt eine Dialogbox zur Konfigurierung des Abonnentenprofils: Anmeldung, Passwort und Telefonnummer
BTN_Information_Serveur	Schnellzugang zur Proxy-Serverinformation	Anklickbares PNG Bild	Zeigt eine Dialogbox zur Konfigurierung der Parameter der Proxyserver
BTN_Configuration_navigateur	Schnellzugriff auf die Navigatoroptionen	Anklickbares PNG Bild	Zeigt eine Dialogbox zur Konfigurierung der Navigatoroptionen
BTN_Accès_Email	Schnellzugriff auf die Email - Anwendung	Anklickbares PNG Bild	Startet den Email-Klienten
BTN_Document_Précédent	Vorherige Seite	Anklickbares PNG Bild	Zeigt die unmittel-bar vorhergehende HTML Seite
BTN_Document_Suivant	Nächste Seite	Anklickbares PNG Bild	Zeigt die unmittel-bar folgende HTML Seite
BTN_Stop_Chargement	Hält das Laden des laufenden Dokuments an	Anklickbares PNG Bild	Hält das Laden des laufenden Dokuments an
BTN_Annuaire	Schnellzugriff auf Lesezeichen	Anklickbares PNG Bild	Zeigt eine Dialogbox mit einer Lesezeichenliste an
BTN-Quitter	Verlässt die Anwendung und kehrt zu ZV zurück	Anklickbares PNG Bild	Verlässt die Anwendung
BTN_Saisie_Adresse	Schnellzugriff zur Wahl einer Verbindung	Anklickbares PNG Bild	Öffnet die Dialogbox, von der man auf eine Seitenadresse (URL) zugreifen kann, oder

			von der man eine erstellte URL auswählen kann
IMG_Logo_Navigateur	Logo	Anklickbares PNG Bild	Kreisabfolge von Bildern zur Anzeige der laufenden Ladeaktivität
IMG_Logo_Opérateur	logo	Anklickbares PNG Bild	
IMG_Diode	Zustands- oder Tätigkeitsindikator	Nicht anklick-bares PNG Bild	Farbdioden (LEDs): rot, grün, blau, schwarz, gelb
BTN_Flèche_haut	Eine Reihe nach oben	Anklickbares und nicht anklickbares PNG Bild	Anklickbare Bilder zeigen an: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. Nicht anklickbare Bilder: verbotener und normaler Zustand
BTN_Flèche_bas	Kursor eine Reihe nach unten bewegen	Anklickbares und nicht anklickbares PNG Bild	Anklickbare Bilder zeigen an: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. Nicht anklickbare Bilder: verbotener und normaler Zustand
BTN_Flèche_droite	Kursor eine Reihe nach vorwärts bewegen	Anklickbares und nicht anklickbares PNG Bild	Anklickbare Bilder zeigen an: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. Nicht anklickbare Bilder: verbotener und normaler Zustand
BTN_Fléche_gauche	Kursor nach hinten bewegen	Anklickbares und nicht anklickbares PNG Bild	3 anklickbare Bilder: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. 2 nicht anklickbare

			Bilder: verbotener und normaler Zustand
BTN_Page_haut	Bewege Kursor mehrere Reihen nach oben (eine halbe Seite oder vorherige Seite)	Anklickbares und nicht anklickbares PNG Bild	3 anklickbare Bilder: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. 2 nicht anklickbare Bilder: verbotener und normaler Zustand
BTN_Page_bas	Bewege Kursor mehrere Reihen nach unten (eine halbe Seite oder folgende Seite)	Anklickbares und nicht anklickbares PNG Bild	3 anklickbare Bilder: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. 2 nicht anklickbare Bilder: verbotener und normaler Zustand
BTN_Frame_Suivante	Gehe zum nächsten Frame	Anklickbares PNG Bild	3 anklickbare Bilder: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. Dieses Bild erscheint, wenn ein Frame folgt; andernfalls wird das Bild nicht gezeigt, um den Bildschirm nicht zu überladen
BTN_Frame_Précédente	Gehe zum vorherigen Frame	Anklickbares PNG Bild	3 anklickbare Bilder: unterdrückter Zustand, nicht unterdrückter Zustand und verbotener Zustand. Dieses Bild erscheint, wenn ein Frame folgt; andernfalls wird das Bild nicht gezeigt, um den Bildschirm nicht zu überladen

BTN_Choix_Croix	Indikator, ob eine Option innerhalb einer Liste geprüft ist oder nicht	Anklickbares und nicht editierbares PNG Bild	Wie eine Box zum Eintragen eines Kreuzes. Mehrfachwahl möglich. Demzufolge gibt es 2 Bilder: geprüft oder nicht geprüft
BTN_Choix_Simple	Indikator, ob eine Option innerhalb einer Liste geprüft ist oder nicht	Anklickbares und nicht editierbares PNG Bild	Wie eine Box zum Eintragen eines Kreuzes. Mehrfachwahl möglich. Demzufolge gibt es 2 Bilder: geprüft oder nicht geprüft
BTN_Ok	Bestätigt eine Frage oder Information	Anklickbares und nicht editierbares PNG Bild	Generisches Bestätigungsbild (ohne Text). Für den Einsatz in einer Dialogbox...
BTN_Annuler	Löscht eine Frage oder eine Information	Anklickbares und nicht editierbares PNG Bild	Generische Löschbild (ohne Text. Für den Einsatz in einer Dialogbox
BTN_Oui		Anklickbares und nicht editierbares PNG Bild	Generische Löschbild (ohne Text. Für den Einsatz in einer Dialogbox
BTN_Non		Anklickbares und nicht editierbares PNG Bild	Generische Löschbild (ohne Text. Für den Einsatz in einer Dialogbox
IMG_Curseur_Normal_Souris	Zeiger / graphischer Kursor	Normales Anzeigebild für Graphikkursor	Wie ein Pfeil mit einem Brennpunkt
IMG_Curseur_Attente_S	Zeiger im betriebsamen Modus	Zeiger in Betrieb	Wie eine Eieruhr oder Chronometer
IMG_Curseur_Texte	Position des Graphikkursors in einem text während des Zugriffs	Zeiger	
IMG_Statusline	Textanzeige in einer Reihe mit 4/5 des Grösse des Bildschirms	Hintergrundbild zur Anzeige von nicht editierbarem Textes	Bild ohne Text. Der Text wird im abgeschnittenen Format angezeigt. Der Text nicht mehr als 40 Zeichen.
IMG_Indéfinie	Generische Anzeige des noch nicht	Anklickbares und nicht anklickbares Bild	Bild das ein Bild im HTML

	geladenen Bildes		Dokument anzeigt, das noch nicht in den Speicher geladen wurde, oder dessen Ladung unterbrochen wurde.
BDG_Login_Password	Generische Dialogbox zur Anzeige oder Eingabe der Anmeldung und des Passworts zum Zugriff eine sicheren Seite	Bildschirmhintergrundbild mit graphischen Rahmen	Hintergrundbild, das ein Gebiet mit informellem Text abgrenzt, Anmeldungsfeld oder Passwortfeld. Diese Dialogbox ist für den Zugriff auf eine sichere Seite erforderlich. Das Programm maskiert das Passwort
BDG_Information_Profil	Generische Dialogbox zur Anzeige oder zur Eingabe der Anmeldung, des Passworts und der Telefonnummer für einen Fernverbindungsserver	Bildschirmhintergrundbild mit graphischem Rahmen	Hintergrundbild, das ein Gebiet mit informellem Text abgrenzt, Anmeldungsfeld oder Passwortfeld oder Telefonnummernfeld. Diese Dialogbox ist vor der Anwahl des Zugangsservers erforderlich. Das Programm maskiert das Passwort und die Bestätigung des Passworts wird verlangt.
BDG_Confirmation-Password	Zur Bestätigung der Eingabe eines Passworts.	Dialogbox mit einem Hintergrundbild mit graphischen Rahmen	Hintergrundbild, das ein Gebiet mit informellem Text und ein Passwortfeld abgrenzt. Diese Dialogbox ist notwendig, wenn der Benutzer ein neues Passwort eingibt oder verändert. Das Programm maskiert das Passwort.
IMG Telecommande filigrane	Filigrane (umrahmt	Umrahmung	Die Umrahmung

	die Tasten) der Fernbedienung für erweiterte Eingabe unter Benutzung des virtuellen Tastatur		der Fernbedienung ist dem Bild der virtuellen Tastatur überlagert, um die Tasten der Fernbedienung visuell dem Entwurf der Tasten der virtuellen Tastatur anzugleichen. Schnellzugrif f durch visuellen Speicher und nicht durch dokumentarisc hen Speicher. Die Umrahmung der Fernbedienung stasten besteht aus keinen Buchstaben oder Symbolen.
BKG_Clavier_virtuel	Virtuelle Tastatur für erweitere Eingabe über die Fernbedienung	Hintergrundbild einer Tastatur	Die virtuelle Tastatur ist auf dem Bildschirm wie die eines PC abgebildet, aber ohne Buchstaben oder Zeichen auf den Tasten darzustellen. Buchstaben auf jede Taste zu bringen wird vom Programm vorgenommen: die erlaubt die Bestimmung einer generischen international en Sprache (azerty qwerty oder andere). Das Umrandungsbil d einer Fernbedienung ist auf der Tastatur überlagert und folgt dem Fokus: dies erfüllt den Zweck visuell darzustellen, dass das Drücken einer Fernbedienung staste einem

			Tastendruck auf der Tastatur entspricht, ohne dass man sich die Übersetzung im Voraus merken muss. Die Escape-(Entrinnen-) Taste weist die virtuelle Tastatur ab. Bestimmte Tasten dienen als Funktionstasten: "http://www", "fr", "com", "org", usw. und andere haben bestimmte Funktionen: "Eingabe", "Rücktaste", Löschen", usw.
BKG_Toolbar_Navigation	Im Symbolleisten-hintergrund dargestellt	Hintergrundbild	
BKG_Toolbat_System_Configuration	Im Symbolleisten-hintergrund dargestellt	Hintergrundbild	
BKG_Annuaire	Im Hintergrund der Lesezeichen-Dialog-box dargestellt	Hintergrundbild	
BKG_Information_Profil	Im Hintergrund der Dialogbox der Konfiguration des Abonnentenprofils abgebildet	Hintergrundbild	
BKG_Information_Serveur	Im Hintergrund der Dialogbox des Proxyservers abgebildet		

[0242] Die folgende Tabelle beschreibt die verschiedenen elementaren Graphikobjekte, die ein Graphikobjekt bilden, das eine veränderliche Grösse der graphischen Abbildung haben kann. Die Nebeneinanderstellung elementarer Objekte zur Herstellung eines komplexen Objekts wird durch das Programm (Wiederherstellung des Puzzles) bewerkstelligt.

Elementare Graphikobjekte	Funktionalität	Type	Kommentare
TBL_Vertical_Gauche	Zur Darstellung des Rahmens einer Tabelle mit veränderlicher Grösse	Bild	Zum Zeichnen eines Rahmens auf einer Tabelle mit veränderlicher Grösse. Dieses Objekt zeigt eine vertikale Streckung der linken Rahmenseite
TBL_Vertical_Droite	Zur Darstellung des Tabellenrahmens	Bild	Zum Zeichnen eines Rahmens auf einer Tabelle mit veränderlicher Grösse. Dieses Objekt zeigt eine vertikale Streckung der rechten Rahmenseite
TBL_Horizontal_Haut	Zur Darstellung des Tabellenrahmens	Bild	Zum Zeichnen eines Rahmens auf einer Tabelle mit veränderlicher Grösse. Dieses Objekt zeigt eine horizontale Streckung der oberen Rahmenseite
TBL_Horizontal_Bas	Zur Darstellung des Tabellenrahmens	Bild	Zum Zeichnen eines Rahmens auf einer Tabelle mit veränderlicher Grösse. Dieses Objekt zeigt eine horizontale Streckung der unteren Rahmenseite
TBL_Angle_Haut_Gauche	Zur Darstellung der Ecke des Tabellenrahmens	Bild	Zum Zeichnen der oberen, linken Ecke eines Tabellenrahmens
TBL_Angle_Haut_Droite	Zur Darstellung der Ecke des Tabellenrahmens	Bild	Zum Zeichnen der oberen, rechten Ecke eines Tabellenrahmens
TBL_Angle_Bas_Gauche	Zur Darstellung der Ecke des Tabellenrahmens	Bild	Zum Zeichnen der unteren, linken Ecke eines Tabellenrahmens
TBL_Angle_Bas_Droite	Zur Darstellung der Ecke des Tabellenrahmens	Bild	Zum Zeichnen der unteren, rechten Ecke eines Tabellenrahmens
CEL_Vertical_Gauche	Zur Darstellung des Rahmens einer Zelle	Bild	Zum Zeichnen des Rahmens einer Zelle. Dieses Objekt zeigt eine vertikale Streckung der linken Seite des Rahmens. Eine Zelle ist Fach in einer Tabelle... Die Dicke eines Zellrahmens sollte schmäler als der einer Tabelle sein
CEL_Vertical_Droite	Zur Darstellung des Rahmens einer Zelle	Bild	Zum Zeichnen des Rahmens einer Zelle. Dieses Objekt zeigt eine vertikale Streckung der rechten Seite des Rahmens

CEL_Horizontal_Haut	Zur Darstellung des Rahmens einer Zelle	Bild	Zum Zeichnen des Rahmens einer Zelle. Dieses Objekt zeigt eine horizontale Streckung der oberen Seite des Rahmens
CEL_Horizontal_Bas	Zur Darstellung des Rahmens einer Zelle	Bild	Zum Zeichnen des Rahmens einer Zelle. Dieses Objekt zeigt eine vertikale Streckung der unteren Seite des Rahmens
CEL_Image	Zur Darstellung des Rahmens eines nicht geladenen Bildes in einer Zelle	Generisches Bild	Generisches Bild, das die Platzierung eines nicht geladenen Bildes in einer Zelle zeigt.
BDG_Vertical_Gauche	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen des Rahmens eine Dialogbox mit variabler Grösse. Dieses Objekt zeigt eine vertikale Streckung der linken Seite des Rahmens
BDG_Vertical_Droite	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen des Rahmens eine Dialogbox mit variabler Grösse. Dieses Objekt zeigt eine vertikale Streckung der rechten Seite des Rahmens
BDG_Horizontal_Haut	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen des Rahmens eine Dialogbox mit variabler Grösse. Dieses Objekt zeigt eine horizontale Streckung der oberen Seite des Rahmens
BDG_Horizontal_Bas	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen des Rahmens eine Dialogbox mit variabler Grösse. Dieses Objekt zeigt eine horizontale Streckung der unteren Seite des Rahmens
BDG_angle Haut_Gauche	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen der oberen linken Ecke des Rahmens einer Dialogbox mit variabler Grösse.
BDG_Angle_Haut_Droite	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen der oberen rechten Ecke des Rahmens einer Dialogbox mit variabler Grösse.
BDG_Angle_Bas_Gauche	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen der unteren linken Ecke des Rahmens einer Dialogbox mit variabler Grösse.
BDG_Angle_Bas_Droite	Zur Darstellung Des Rahmens einer Dialogbox	Generisches Bild	Zum Zeichnen der unteren rechten Ecke des Rahmens einer Dialogbox mit variabler Grösse.
LST_Vertical_Gauche	Zur Darstellung des	Bild	Zum Zeichnen des

	Rahmens einer vertikalen Liste auswählbarer Optionen		Rahmens einer vertikalen Liste mit variabler Grösse. Dieses Objekt zeigt eine vertikale Streckung der linken Seite des Rahmens
LST_Vertical_Droite	Zur Darstellung des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zum Zeichnen des Rahmens einer vertikalen Liste mit variabler Grösse. Dieses Objekt zeigt eine vertikale Streckung der rechten Seite des Rahmens
LST_Bas_Gauche	Zur Darstellung des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zum Zeichnen des Rahmens einer vertikalen Liste mit variabler Grösse. Dieses Objekt zeigt eine horizontale Streckung der oberen Seite des Rahmens
LST_Bas_Droite	Zur Darstellung des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zum Zeichnen des Rahmens einer vertikalen Liste mit variabler Grösse. Dieses Objekt zeigt eine horizontale Streckung der oberen Seite des Rahmens
LST_Angle_Haut_Gauche	Zur Darstellung der Ecke des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zur Zeichnung der oberen linken Ecke des Rahmens einer vertikalen Liste mit variabler Grösse
LST_Angle_Haut_Droite	Zur Darstellung der Ecke des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zur Zeichnung der oberen rechten Ecke des Rahmens einer vertikalen Liste mit variabler Grösse
LST_Angle_Bas_Gauche	Zur Darstellung der Ecke des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zur Zeichnung der unteren linken Ecke des Rahmens einer vertikalen Liste mit variabler Grösse
LST_angle_Bas_Droite	Zur Darstellung der Ecke des Rahmens einer vertikalen Liste auswählbarer Optionen	Bild	Zur Zeichnung der unteren rechten Ecke des Rahmens einer vertikalen Liste mit variabler Grösse
ASC_Haut	Zur Darstellung einer vertikalen Bildlaufleiste ("Aufzug")	Anklickbares und nicht anklickbares Bild	Zur Zeichnung des oberen Endes einer vertikalen Bildlaufleiste ("Aufzug")
ASC_Bas	Zur Darstellung einer vertikalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Zeichnung des unteren Endes einer vertikalen Bildlaufleiste
ASC_Cage	Zur Darstellung einer vertikalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Zeichnung einer Stufe ("Etage") einer vertikalen Bildlaufleiste
ASC_Ascenseur	Zur Darstellung einer vertikalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Anzeige der Position der vertikalen Bildlaufleiste
SCR_Gauche	Zur Darstellung einer horizontalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Zeichnung des linken Endes einer horizontalen Bildlaufleiste eines

			Fensters, Textfeldes, eines Frames, usw.
SCR_Droite	Zur Darstellung einer horizontalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Zeichnung des rechten Endes einer horizontalen Bildlaufleiste eines Fensters, Textfeldes, eines Frames, usw.
SCR_Cage	Zur Darstellung einer horizontalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Zeichnung eines Fachs ("Schrittes") einer horizontalen Bildlaufleiste eines Fensters, Textfeldes, eines Frames, usw.
SCR_Position	Zur Darstellung einer horizontalen Bildlaufleiste	Anklickbares und nicht anklickbares Bild	Zur Anzeige der Position des Kursors in der horizontalen Bildlaufleiste eines Fensters, Textfeldes, eines Frames, usw.
IMG_Gauche	Zur Darstellung einer Textzeile	Nicht anklickbares Bild	Das ist das Bild auf der linken Seite des nicht editierbaren, gezeigten Textes. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Droite		Nicht anklickbares Bild	Das ist das Bild auf der rechten Seite des nicht editierbaren, gezeigten Textes. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Caractere		Nicht anklickbares Bild	Das ist das Hintergrundbild auf dem ein nicht editierbares Zeichen gedruckt wird. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Edition_Gauche	Zur Darstellung einer zu editierenden Textzeile	Editierbares und anklickbares Bild	Dies ist das Bild auf der linken Seite einer editierbaren und auswählbaren Textzone. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Edition_Droite	Zur Darstellung einer zu editierenden Textzeile	Editierbares und anklickbares Bild	Dies ist das Bild auf der rechten Seite einer editierbaren und auswählbaren Textzone. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Edition_Caractere	Zur Darstellung	Editierbares und	Dies ist das

	einer zu editierenden Textzeile	anklickbares Bild	Hintergrundbild auf dem ein Zeichen einer editierbaren und auswählbaren Textzone gedruckt wird. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Multi_Edition_Gauche	Zur Darstellung der Bearbeitung von mehrzeiligem Text	Editierbares und anklickbares Bild	Dies ist das Bild auf der linken Seite einer editierbaren und auswählbaren Textzone. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Multi_Edition_Droite	Zur Darstellung der Bearbeitung von mehrzeiligem Text	Editierbares und anklickbares Bild	Dies ist das Bild auf der rechten Seite einer editierbaren und auswählbaren Textzone. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen
IMG_Multi_Edition_Caractere	Zur Darstellung der Bearbeitung von mehrzeiligem Text	Editierbares und anklickbares Bild	Dies ist das Hintergrundbild auf dem ein Zeichen einer editierbaren und auswählbaren Textzone gedruckt wird. Die gewählte Grösse muss mit dem gewählten Zeichensatz übereinstimmen

Navigatorfunktionen

Elementare Graphikobjekte	**Beschreibung**	**Aktiviert durch**	**Graphikobjekte**	**Kommentare**
Connect / disconnect	Modemverbindung mit dem definierten Profil einschalten Modem ausschalten	BTN_Connecter BTM_Déconnecter		. Starte die Modemverbindung, wenn es nicht angeschlossen ist, wenn die Information zum Verbindungsprofil ausgefüllt ist, wenn die URL angeklickt wird
Konfigurieren und Verändern der Authentification auf Information des Verbindungsprofils hin	Zeigt eine Dialogbox zur Konfigurierung des Abonnentenprofils : Anmelden, Passwort, Telefon	BTN_Information_Profil	BDG_Information_Profil	. Eingabe, Ändern, Bestätigung, Speichern im Flashspeicher, Passwort und Telefonnummer des Zugangsservers . Zugang löschen
Bestätigung der Passworteingabe	Aufruf das Passwort einzugeben	Nach der Änderung eines Passwortes	BDG_Confirmation_Password	. Eingabe, Bestätigung, Löschen . Jedes eingegebene

				Zeichen wird durch ein Maskierungszeichen (Stern) beim Drucken ersetzt . Vergleiche die Eingabe mit dem veränderten Passwort
Drucke oder gebe Anmeldung und Passwort ein, um auf die sichere Seite zuzugreifen	Anfrage der Authentifizierung von der entfernten Seite (site)	In Folge eines Versuches auf die sichere Seite zuzugreifen	BDG_Login_Passwort	. Eingabe, Bestätigung, Löschen, Senden der Information an die Seite (site) . Jedes Zeichen des Passworts wird durch ein Maskierungszeichen (Stern) beim Drucken ersetzt
Verwalten einer virtuellen Tastatur	Virtuelle Tastatur für erweiterte Eingabe mit einer Fernbedienung	Wenn der Fokus auf dem editierbaren Objekt liegt	BKG_Clavier_Virtuel IMG_Telecommande_filigrane	Die virtuelle Tastatur wird auf dem Bildschirm wie die eines PC, aber ohne Zeichen auf dem Tasten zu zeigen, dargestellt. Das Aufbringen von Zeichen auf die tasten erfolgt durch das Programm: dies erlaubt die Definition einer generischen, internationalen Tastatursprache(azerty, qwerty oder andere). Das Umrandungsbild einer Fernbedienung ist auf der Tastatur überlagert und folgt dem Fokus: dies erfüllt den Zweck visuell darzustellen, dass das Drücken einer Fernbedienungstaste einem Tastendruck auf der Tastatur entspricht, ohne dass man sich die Übersetzung im Voraus merken muss. Die Escape-(Entrinnen-) Taste weist die virtuelle Tastatur ab. Bestimmte Tasten dienen als Funktionstasten: "http://www", "fr", "com", "org", usw. und andere haben bestimmte Funktionen: "Eingabe", "Rücktaste", Löschen", usw.

[0243] Die Java API für den Webbrowser wird nun beschrieben.

[0244] Hier folgt nun die Aufstellung der Java Pakete, die auf der Ebene der Navigatoranwendung im Dekoder

gebraucht werden. Die Liste ist in zwei Teile aufgeteilt: in die AWT (Abstract Window Toolkit – abstrakter Fenster-Werkzeugkasten) Klassen der JDK 1.1 und die Java Schnittstellenklassen der verschiedenen Dienste, die im nativen C-Kode geschrieben sind.

Klassen	Verfahren
java.awt.Choice	(Add(String) GetItemCount() GetSelectedIndex() Remove(String) SetSelectedIndex(int)
java.awt.Component	AddKeyListener (KeyListener) AddFocusListener (FocusListener) AddMouseListener (MouseListener) Contains (int,int) enableEvents(long) setLocation() getSize() setBackground(Color) setBackground(ImageMask) setBackground(int) setLocation(int) setSize(int,int)
java.awt.Graphics	drawImage getSize setLocation
jawa.awt.Image	createImage(string) getHeight() getWidth()
java.awt.ImageMask extends java.awt.Image	setMask(bitmap)
java.awt.List	List(int) add(String) getItem(num) getItemCount() getSelectItem() remove(String) replace() setMultipleMode(boolean)
jaja.awt.Panel	setLayout(layout)
java.awt.Point	
java.awt.Toolkit	loadLut(String) getDefaultToolkit()
java.awt.TextField	addActionListener(ActionListener) setEchoChar(char) setSecretMode()
java.awt.Window	setModal(boolean)
java.awt.event.FocusEvent	
java.awt.event.FocusListener (Schnittstelle)	void focusGained(FocusEvent) void focusLost(FocusEvent)
java.awt.event.KeyEvent	
java.awt.event.KeyListener (Schnittstelle)	void keyPressed(KeyEvent) void keyReleased(KeyEvent) void keyTyped(KeyEvent)
java.awt.event.MouseEvent	
java.awt.event:MouseListener (Schnittstelle)	void mouseClicked(MouseEvent) void mouseEntered(MouseEvent) void mouseExited(MouseEvent) void mousePressed(MouseEvent)d void mouseReleased(MouseEvent)

[0245] Das Navigatorpaket, das Browserpaket genannt wird, fasst verschiedene Pakete zusammen: das Browser-Zeichenpaket, das Dienste anbietet, die es erlauben ein HTML Dokument abzurufen und innerhalb des Dokumentbrowsers zu navigieren; und das Mediawebtv Paket, das den Aufbau einer Internetverbindung mit der Authentifizierung des Benutzers erlaubt.

[0246] Nun wird die Struktur der browser.drawer.MhwBookmark Klasse beschrieben.

[0247] Eine Lesezeichenliste ist einem Benutzer zugeordnet. Es gibt innerhalb der Lesezeichenliste keine Hierarchie.
Constructor: MhwBookmark(subscriberId): öffnet eine bestehende Lesezeichenliste
Constructor: MhwBookmark(subscriberId): erzeugt eine neue Lesezeichenliste
deleteBookmark(): löscht eine Lesezeichenliste.

add(URL, Name): fügt eine Eingabe hinzu
remove(itemNumber): löscht eine Eingabe
modify(itemNumber, URL, Name): ändert eine bestehende Eingabe
getList(): gibt eine Liste der Eingaben in die Lesezeichenliste (keine Hierarchie) zurück
getItemCount(): gibt die Anzahl der Eingaben in die Lesezeichenliste
isFull(): boolesch – ist Liste voll?
isEmpty(): boolesch – ist Liste leer?
setHomePage(itemNumber)
setHomepage(): itemNumber
goToURL(sessionNumber): lädt das Dokument, das der gewählten Eingabe entspricht

[0248] Im Falle dass es einen Fehler gibt, wird eine Fehlermeldung durch die Methoden add(), remove(), modify(), getList(), setHomepage() zurückgegeben (oder einem Vorfall, wenn die Ausprägung asynchron ist).

[0249] Die browser.drawer.MhwHistory Klasse erlaubt das Navigieren von einem Dokument zum anderen innerhalb einer Liste von zuvor gezeigten Dokumenten. Es gibt in der Verlaufsliste keine Hierarchie. Die Einzelheiten der Klasse werden nun aufgeführt.
Constructor: MhwHistory(sessionNumber)
getList(): gibt die Verlaufsliste zurück (keine Hierarchie)
getCurrent(): erhält die gegenwärtige URL
setCurrent(indexNumber): verändert die gegenwärtige URL
getNext(): erhält die URL der nächsten Eingabe
getPrevious(): erhält die URL der vorangegangenen URL
getItemCount(): gibt die Anzahl der Eingaben im Verlauf zurück
addEventsRegister(): abonniert Fehlerereignisse aus der Verlaufsliste
removeEventsRegister()

[0250] Im Falle, dass es einen Fehler gibt, wird eine Fehlermeldung durch die Methoden getList(), getNext(), getPrevious(), setCurrent() zurückgegeben (oder einem Vorfall, wenn die Ausprägung asynchron ist).

[0251] Die Ereignisse sind: addEventsRegister(sessionNumber): [abonniert die Fehlerereignisse der Verlaufsliste]; and removeEventsRegister(sessionNumber) [abbestellen].

[0252] Die browser.drawer:MhwDocument Klasse erlaubt das laden und Anzeigen eines HTML Dokumentes im Dekoder. Die Einzelheiten der Klasse werden nun erläutert.
Constructor: MhwDocument(sessionNumber)
freeze(): hält die Anzeige des aktuellen Dokuments (Laden des Dokuments geht weiter)
unfreeze(): startet die Anzeige des aktuellen Dokuments erneut
isPending(): falls das Dokument gerade geladen wird
stop(): hält das Laden des aktuellen Dokuments an
reload(): lädt das Dokument erneut
getDocumentInfo(): gibt den Titel und die URL des Dokumentes zurück
addStatusRegister(): abonniert die Information zum Zustand und Ende des Ladens
goToURL(url): lädt eine Webseite
submit(login, password, URL): reicht Authentifizierung zum Laden einer Webseite ein
getStatisticsDocument(): gibt die Anzahl der laufenden Anfragen und der URL des gerade zu ladenden Dokuments zurück

[0253] Die browser.mediawebtv.MhwConnection Klasse vereinigt die Benutzerverbindungs- und Authentifizierungsfunktionalität. Die Einzelheiten der Klasse werden nun erörtert:
Constructor: MhwConnection(subscriberId)
start(): fragt Verbindung an
stop(): fragt Verbindungsende an
cancel(): löscht die Verbindung
setAuthentificationType(type): setzt den
Authentifizierungsmodus auf den CANAL+ Modus (msd/Passwort) oder durch Anmeldung/Passwort)
getAttributes(): führt die Verbindungsattribute zurück
setAttributes(attributes): verändert die Verbindungsattribute
setPassword(password): verändert das Passwort
setPassword(password): erhält das Passwort

setRutoCheckPassword(bAutoCheck): legt fest, ob das Passwort automatisch mit dem Bestätigungspasswort geprüft werden muss
getAutoCheckPassword(bAutoCheck): liest fest, ob das Passwort automatisch mit dem Bestätigungspasswort geprüft werden muss
getIPClient(ipaddress, netmask): liest die gewählte IP und das Netzwerkmaskenpaar
setIPClient(ipaddress, netmask): verändert die gewählte IP und das Netzwerkmaskenpaar
getDNS(dns1, dns2): liest die primären und sekundären DNS Adressen
setDNS(dns1, dns2): verändert die primären und sekundären DNS Adressen
getURLConfigServer(url): liest die Adresse des Konfigurationsservers
setURLConfigServer(url): verändert die Adresse des Konfigurationsservers
getQueryCommand(queryCmd, typeOfQuery): liest nach Type die Anfrage, die an den Konfigurationsservers gesendet werden soll
setQueryCommand(queryCmd, typeOfQuery): verändert die Anfrage nach typequeryAcquisitions(tableAcquisitions, typeOfAcquisitions, NumberOfAcquisitions): liest die Liste der Akquisitionen.
startAcquisition(acquisitionId): startet einen Zukauf (Daten/Video)
stopAcquisition(acquisitionId): hält einen Zukauf (Daten/Video) an
addStatusRegister(): Abonnement ist über den Verbindungsstatus zu informieren.

[0254] Die Ereignisse sind: Verbindungsverlust; laufende Verbindung; aufgebaute Verbindung; Anforderung Verbindungsbestätigung; Verbindungsfehler; Modemstatus ein/aus; Initialisierung im Gange; Wählvorgang; Fehler, aber Modem ist an; Serverstatus; ungültiger Anschluss; ungültige URL; Anmeldefehler; und ungültiges Passwort.
removeStatusRegister(): stellt Netzwerkverbindungsstatus ab
isConnected subscriberId(): gibt boolesch zurück und dem Modem Status für verbunden/trennen
isPending(): gibt boolesch für Modem beim Aufbau der Verbindung zurück
getExtendedProviderUrl(providerUrl): liest den momentan abonnierten Anbieter
setExtendedProviderUrl(providerUrl): ändert den momentan abonnierten Anbieter

[0255] Die browser.mediawebtv.MhwConfiguration Klasse verwaltet das profil jedes Benutzers und seine oder ihre Präferenzen. Die Einzelheiten der Klasse werden nun aufgeführt.
Constructor: MhwConfoguration()
readprofile(subscriberId): lese das Profil
writeProfile(subscriberId, profile): schreibe das Profil
readDefaultProfile(): lese das Standardprofil
writeDefaultProfile(profile): verändert das Standardprofil
getUserCount(): Anzahl der Nutzer
newUser(profile): Identifizierung eines neuen Benutzers
getLastConnect(): der letzte verbundene Nutzer

[0256] Die höchste Anzahl von Profilen ist momentan auf 5 festgelegt, aber dies stellt keine strenge Begrenzung dar; grösser Anzahl von Profilen können bei Bedarf gespeichert werden.

[0257] Im Falle eines Fehlers wird eine Fehlermeldung durch die Methoden WriteProfile() und writeDefaultProfile() (oder ein Ereignis, wenn das momentane Auftreten asynchron ist) zurück geschickt.

[0258] Die browswer.mediawebtv.MhwMultiSession Klasse erlaubt es eine Navigatorsession auszuwählen. Eine Session ist eine Instanz des Navigators, die automatisch durch eine andere Instanz eingeleitet wurde. Wenn der Navigator startet, wird eine Session erzeugt nach dem Aufbau einer authentifizierten Verbindung. Die Einzelheiten der Klasse werden nun erläutert.
Constructor: MhwMultiSession
getCurrentSessionNumber()
setCurrentSessionNumber(int number)
getPreviousSession()
addSession(): giert die Anzahl der erzeugten Sessions zurück
removeSession(int numSession)

[0259] Das Navigatormodell für den Dekoder wird nun in weiteren Einzelheiten beschrieben unter Bezug auf die **Fig. 6a** und **Fig. 7**–**Fig. 30**.

[0260] Das hier vorgestellte Modell ist ein einfaches Beispiel eines Navigators und gibt eine allgemeine Idee

der grundsätzlichen Funktionen. Es gibt vollkommene Freiheit auf der Ebene der graphischen Darstellung. Die einzige wichtige Eigenschaft sind die Zonen und Bilder innerhalb denen Funktionen der gleichen Art gruppiert sind und die allgemeine Benutzerschnittstelle.

[0261] Der Navigator verwendet alle verfügbaren Graphiken (MPEG, PIXMAP) des graphischen Studios. Die Bildschirmdarstellungen dieses Modells sind in Baumform organisiert und jede bildet zusammen eine Ansammlung unverzichtbarer Funktionen. Jede Funktion oder Option auf dem Bildschirm wird durch Bewegen eines Fokus angesteuert mit Hilfe von Pfeiltasten (auf der Fernbedienung) oder durch Einsatz der Tastatur mit einem Kursor/Zeiger. Die Auswahl einer Aktion wird durch einen Kontrollklick oder durch eine vorbestimmte Taste (zum Beispiel "OK") erzielt.

[0262] Wenn keine physische Tastatur vorhanden ist, um Text mit der Fernbedienung einzugeben, ist es notwendig, eine virtuelle Tastatur bereit zu stellen. Dies wird durch Bewegung des Fokus erreicht mit einer Möglichkeit schnell einzutippen durch Abbilden der Tasten der Fernbedienung auf der virtuellen Tastatur; mit anderen Worten: das Bild der Tasten auf der Fernbedienung wird sichtbar (in nachgezeichneter Form) mit Umrandung oder leicht undurchlässig auf dem Bild der virtuellen Tastatur. Die virtuelle Tastatur wird in weiteren Einzelheiten in diesem Dokument erfolgen.

[0263] Die Fig. 30 zeigt die Kette der obersten Ebene der Navigatorfunktionsschaltflächen (**1410**) mit einem Teil einer Webseite die daneben sichtbar ist (**1411**).

[0264] Der Hauptnavigatorbildschirm wird nun beschrieben unter besonderem Bezug auf die Fig. 30.

[0265] Die Navigatorfunktionen sind in mehreren Schichten zusammen gefasst. Der Hauptbildschirm zeigt eine vertikale Leiste (Hauptmenü) (**1410**), das aus einer Reihe von Schaltflächen zusammengesetzt ist: Neu laden/Stop (**1401**), vorherige Seite (**1402**), nächste Seite (**1403**), Verlauf kürzlich besuchter Seiten, Lesezeichen, verbinden/trennen, Konfiguration, Navigator verlassen (**1408**).

[0266] Die Navigator GUI (graphical user interface) graphische Benutzerschnittstelle (Hauptbildschirm) wird angezeigt, wenn eine Funktionstaste gedrückt wird (entweder auf der Fernbedienung oder auf der Tastatur). Wenn der GUI auf dem TV-Bildschirm erscheint, wird das HTML Dokument (**1411**) (welches der GUI überdeckt) in den Speicher geladen, aber die Aktualisierung des Dokumentes wird angehalten, um das Anzeigeverhalten des GUI nicht zu beeinflussen. Das HTML Dokument wird erneut angezeigt, wenn der Navigator GUI verlassen wird. Diese Einschränkung wird aufgehoben, wenn die Leistungsmerkmale eventuell zufrieden stellend sind, wobei die gleichzeitige Darstellung des HTML Dokuments und der graphischen Benutzerschnittstelle (GUI) ermöglicht wird.

[0267] Die graphische Benutzerschnittstelle (GUI) wird von der TV Bildschirmfläche auf Tastendruck (auf der Fernbedienung oder der Tastatur) verschwinden, oder durch Anklicken ausserhalb der GUI Schaltflächenbereiche. Die Anzeige des gegenwärtig zu ladenden oder im Cache-Speicher zu speichernden HTML Dokuments wird dann gestartet oder erneut begonnen.

[0268] Eine Schaltfläche ist in Wirklichkeit ein rechteckiger oder quadratischer Bereich (zum Beispiel 32 × 32 Pixel). Wenn der Graphikkursor einen Bereich berührt, erhält dieser Bereich (Fenster) den Fokus (vergleiche die Funktion EnterNotify(WindowId)).

[0269] Wenn die Schaltflächengraphik zum Beispiel einen Reifen darstellt, muss festgestellt werden, ob die gegenwärtige Position des Kursors tatsächlich die Pixel des Reifens abdeckt. Es ist daher notwendig, den Wert des Pixels im Kernbereich des Mauszeigers in der Ausschnittsmaske der Schaltfläche zu finden (durch Berechnung der relativen Position, getpixel() in der Ausschnittsmaske, dann den Pixelwert prüfen). Dieses Erkennungsverfahren erlaubt es die Prüfung zu verbessern, ob ein Klick ausgeführt, oder vor dem Beginn der Schaltflächenfunktion nicht ausgeführt wurde.

[0270] Wenn der Mauszeiger den rechteckigen oder quadratischen Bereich verlässt, dann verliert der Bereich den Fokus (vergl. LeaveNotify(WindowId)).

[0271] Die erklärende Schaltflächenübersicht wird nun im Einzelnen beschrieben.

[0272] Wenn der Mauszeiger oder der rechteckige Fokus mit der Schaltfläche, die eine Funktion darstellt, zusammenfällt, wird ein kurzer Satz (tooltip) angezeigt entweder waagerecht oder senkrecht, der die Funktion

dieser Schaltfläche erläutert. Wenn die Schaltfläche entweder durch einen Mausklick oder durch eine Funktionstaste ausgewählt wird, erscheint eine Liste mit Schaltflächen, die die Untermenüoptionen enthalten. Das System der kurzen erklärenden Sätze (tooltips) wird auch bei den Schaltflächen der Untermenüs eingesetzt.

[0273] Navigation der Menüoptionen wird mit den Pfeiltasten auf der Fernbedienung oder der Tastatur erreicht. Die letzte Schaltfläche auf dem Hauptbildschirm, auf der der Fokus lag, wird für die nächste Anzeige des Hauptbildschirms gespeichert.

[0274] Nun wird die virtuelle Tastatur in weiteren Einzelheiten unter Bezug auf die Fig. 42–Fig. 45 erläutert.

[0275] Die Fig. 42 zeigt schematisch, wie die im Moment sichtbare virtuelle Tastatur (**1501**) auf dem zugrunde liegenden Gitter der virtuellen Tastatur (**1501**, **1506**) abgebildet wird.

[0276] Die Fig. 43 zeigt eine typische Darstellung von Zeichen auf den Tasten einer virtuellen Tastatur.

[0277] Die Fig. 44 zeigt die Bilder, die bei der virtuellen Tastatur benutzt werden mit dem ersten (**1545**) und zweiten (**1546**) Nummernblock, die jeweils den Fokus besitzen (und auch verschiedene Arten von Fernbedienungen **1542** und **1543**) zeigen.

[0278] Die Fig. 45 zeigt eine typische virtuelle Tastaturausführung mit typischen überlagerten Dimensionierungen.

[0279] Zuerst wird die virtuelle Tastatur als ein Werkzeug wahrgenommen, das von der Anwendung, bei der sie eingesetzt wird, unabhängig ist. Man kann sie innerhalb des Webbrowseranwendung und gleichermassen bei der Mailanwendung benutzen. Darüber hinaus ist ihr Aussehen vollständig von der betroffenen Anwendung unabhängig.

[0280] Die virtuelle Tastatur wird vom Zeitpunkt an, zu dem der Benutzer ein editierbares Gebiet auf dem Bildschirm wählt, angezeigt, wenn er keine wirkliche Tastatur, oder eine Fernbedienung mit Tastatur besitzt. Der Fokus wird auf das Ende des Textes in dem editierbaren Gebiet gelegt. Beim Drücken von "OK" (auf der Fernbedienung oder der virtuellen Tastatur), oder "Löschen (auf der virtuellen Tastatur) verschwindet er.

[0281] Die virtuelle Tastatur, die auf dem Bildschirm (**1501**) sichtbar ist, besteht aus drei Blöcken mit 10 Tasten (die drei Nummernblöcke der Fernbedienung darstellen), die Seite an Seite beieinander liegen (**1502**, **1503**, **1504**). Der Benutzer kann den Fokus (**1505**) mit den Pfeiltasten auf der Fernbedienung von einem Block zum anderen verschieben. Nachdem ein Block ausgewählt wurde, wird das entsprechende Zeichen, das auf der virtuellen Tastatur aufgedruckt ist, durch Drücken einer Taste auf dem Nummernblock der Fernbedienung eingegeben.

[0282] Der Benutzer kann auch die AUF- und AB-Pfeiltasten benutzen. Dies bringt die gleiche virtuelle Tastatur auf den Bildschirm, jedoch mit unterschiedlichen Zeichen auf den tasten (**1506**). Somit kann man durch Hin- und Herschalten zwischen einer Reihe von 5 virtuellen Tastaturen alle Zeichen eines westlichen Computers darstellen. Es ist auch möglich weitere Tastaturen hinzuzufügen, falls dafür Bedarf entstehen sollte.

[0283] Unter Bezug auf die Fig. 44 und um eine sofortige Beziehung zwischen den Nummernblöcken der Fernbedienung und dem Fokus der virtuellen Tastatur herzustellen, zeigt ein Überlagerungsbild der Fernbedienung den Fokus (**1542**, **1542**). Somit kann der Benutzer leicht erkennen, dass nur ein Teil der Tastatur den Fokus besitzt und dass der Rest der Zeichen durch Hin- und Herbewegen des Fokus mit den Pfeiltasten erreicht werden kann. Die Tastatur ist gedanklich mit den letzteren Punkten aufgebaut.

[0284] Die Lösung mit einer virtuellen Tastatur nimmt wenig Bildschirmhöhe in Anspruch und erlaubt leicht die Anzahl verfügbarer Zeichen zu erweitern. Im Standardbetrieb wird die virtuelle Tastatur mit einem Miniaturalphabet dargestellt.

[0285] Bestimmte Schaltflächen haben wichtige Sonderfunktionen:

– "OK" auf der Fernbedienung zur Bestätigung der aktuellen Wahl (wenn das Feld nur eine Zeile hat, bestätigt auch das ¶ Zeichen (**1521**) oder kann alternativ so gewählt werden, dass es keine Funktion besitzt; andernfalls entspricht es nur zur Zeilenumschaltung).

– "Löschen" (**1522**) auf der virtuellen Tastatur, zum Verlassen des Werkzeugs ohne Bestätigung (die Veränderungen, die nach dem Öffnen der Tastatur gemacht wurden, gehen verloren).

– "Rücktaste" (**1523**) auf der virtuellen Tastatur, zum Löschen des zuletzt eingegeben Zeichens.
– Die Auf, Ab, Links, Rechts Pfeiltasten, um in der Editierzone sich hin- und herzubewegen.
– "Tab" (**1520**) auf der virtuellen Tastatur, zum Einfügen einer konfigurierbaren Anzahl von Zwischenräumen "in einem Rutsch" (vier standardmässig).

[0286] Die Tastatur befindet sich immer im "Einfügen" Modus.

[0287] Ein Beispiel einer Tastatur ist in der Fig. 44 dargestellt und wird später erläutert.

[0288] Mit den 5 Tastaturen (5 × 3), die in der Fig. 43 gezeigt werden und den beiden Schriftarten, die für die Webbrowseranwendung (Arialweb und Courier) installiert sind, können alle Zeichen einer herkömmlichen Tastatur abgedeckt werden. Die Abmessungen der Tastatur auf dem Bildschirm betragen 272 Pixels breit mal 184 Pixel hoch.

[0289] Die virtuelle Tastatur und die funktionalen Verbindungen in Bezug auf ihren Einsatz bei den verschiedenen Anwendungen wurden innerhalb des "canalplus.virtualkd" Paketes entwickelt.

[0290] Die Klassen die innerhalb des Paketes "MhwVirtualKbd" (die graphische Beschreibung der virtuellen Tastatur und der Verhaltensklasse), "MhwVkTextField" (die aus "java.awt.TextField" abgeleitet ist und die eine virtuelle Tastatur zulässt, die innerhalb der globalen Anwendung definiert ist und das Textfeld zur Steuerung der Ereignisse gemeinsam benutzt) und MhwTextArea" (eine Klasse die von "java.awt.Textarea" geerbt wurde und die eine virtuelle Tastatur zulässt, die innerhalb der globalen Anwendung definiert ist und die Textfeldklasse zur Steuerung der Ereignisse benutzt).

[0291] Die MhwVirtualKbd Klasse wird nun in weiteren Einzelheiten beschrieben.

[0292] Der Konstruktor der MhwVirtualKbd" Klasse wird als "privat" definiert. Daher kann nur eine einzigartige virtuelle Tastatur konstruiert werden, wenn die Hauptanwendung, die sie einsetzen könnte, gestartet wird zum Beispiel wenn eine wirkliche Tastatur fehlt). Das Ziel ist daher eine Tastatur vorzustellen, die besonders für die laufende Anwendung konfiguriert ist und die erscheint, wenn der Benutzer ein Textfeld eingibt (Einfach- oder Mehrfachzeile).

[0293] Wenn die Tastatur erzeugt wird, werden die vier (statischen) Hauptvariablen, die konfiguriert werden können, gesetzt sein:
– Ursprung (parent): Container, "Ursprung" der virtuellen Tastatur, die selbst vorhanden sein muss zum Zeitpunkt, zu dem die Tastatur geschaffen wird. Sie wird mit dem "setParent"-Verfahren gesetzt, das eine NullPointerException" zurückspielt, wenn das "parent" im Argument "Null" beträgt
– beschreibende Datei (descriptive file (die ASCIII Datei, die die Tastatur sowohl in Bezug auf die Graphiken hinter den Bildern, die die verschiedenen Tastaturen betreffen, und die man erhält, wenn die virtuelle Tastatur benutzt wird, beschreibt, als auch die auf die Tasten gedruckte Beschriftung. Die Zeichen sind durch ihren Unicode Kode spezifiziert. Der Name der beschreibenden Datei kann gesetzt werden, in dem man das "setScreenFile"-Verfahren benutzt.
– Anzahl der Bildschirmdarstellungen: Anzahl der von der virtuelle Tastatur initialisierten und benutzten "Nummernblöcke". Diese Anzahl, die durch "setScreensNumber" gesetzt wird, entspricht der Anzahl an Tastaturen, deren Eigenschaften in der oben genau erläuterten beschreibenden Datei eingespeichert sind.
– Eingangskoordinaten: dies sind die Koordinaten der oberen linken Ecke des Hintergrundbildes der Tastatur in dem Ursprungscontainer (parent container) (der oben beschrieben wurde). Dies wird eingestellt, in dem man das "setCoordInit" Verfahren benutzt.

[0294] Sobald die Tastatur eingestellt ist, kann man entscheiden, ob sie gebraucht wird, oder auch nicht, in dem man das "getInstance" Verfahren einsetzt, das die Tastatur der laufenden Anwendung findet, falls sie besteht (falls die virtuelle Tastatur der Anwendung noch nicht besteht und wenn die Anwendung sie benutzt, dann schafft das "getInstance" Verfahren eine und benutzt dazu die Variablen [die vier zuvor Beschriebenen], die man eingesetzt hätte).

[0295] Die Ereignisverwaltung wird nun beschrieben.

[0296] Gemäß den vorhergehenden Beschreibungen funktioniert die virtuelle Tastatur, wenn einmal angezeigt, alleine durch Auswertung der Vorgänge, die an sie übertragen werden durch: den Nummernblock, die "OK"-Taste und die vier Richtungspfeiltasten auf der Fernbedienung. Diese Tasten werden besondere Rollen

für die im Gebrauch befindliche Tastatur übernehmen.

[0297] Die "OK" Taste besitzt eine wichtige Rolle, da sie dem Benutzer erlaubt zwei Dinge auszuführen: "Zurück" zum Textfeld, um Informationen einzugeben und dann den Betrieb der virtuellen Tastatur anzuzeigen und zu starten und dann "Verlassen", um das Textfeld "zu verlassen" und die Änderungen zu speichern.

Pfeiltasten

[0298] Die "rechten" und die "linken" Pfeile gestatten das Bild der Fernbedienung (in dem der Nummernblock, der den Fokus besitzt, angezeigt wird) auf den drei Nummernblöcken, die auf der virtuellen Tastatur dargestellt sind, hin- und herzubewegen. Die Tasten des Nummernblocks auf der Fernbedienung sind folglich "vorherbestimmt", abhängig von der Tastatur mit dem Fokus, um die verschiedenen Zeichen darzustellen.

[0299] Im häufigsten Fall verursacht das "Drücken" dieser Knöpfe, wenn die virtuellen Tastatur aktive ist, automatisches Einfügen des Zeichens auf dieser Taste in das laufende Textfeld an der durch den Kursor bezeichneten Stelle.

[0300] Von den Zeichen können sechs als "besondere" Zeichen betrachtet werden und veranlassen das auf der Taste dargestellte Zeichen nicht direkt auch im Textfeld angezeigt zu werden;
Rücktaste (BackSpace): "< (**1523**): wenn die Taste auf der Fernbedienung, die diesem Zeichen entspricht, gedrückt wird, wird das Zeichen unmittelbar links von der Kursorposition im laufenden Feld gelöscht.
Tab: ">>" (**1520**): wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, wird eine konfigurierbare Anzahl von Zwischenräumen (' '), standardmäßig 4, an der laufenden Kursorposition eingefügt.
Eingabe: "¶" (**1521**): wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, wird ein Zeilenvorschub (line feed) an der Kursorposition eingefügt. Eigentlich ist das laufende Textfeld eine Instanz der "MhwVkTextField" Klasse, das bedeutet eine, die nur eine editierbare Zeile hat, Drücken dieser Taste hat entweder keine Auswirkung, oder verursacht Bestätigung diese Feldes. wenn im Gegensatz dieses Textfeld eine Instanz der "MhwVkTextArea" Klasse ist, besteht sie aus mehreren editierbaren Zeilen und dieses Zeichen verursacht einen Zeilenvorschub (wenn der Kursor sich auf der letzten editierbaren Zeile befindet, hat das Drücken dieser Taste keine Auswirkung).
Löschen: "¢ (**1522**): wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, werden alle Veränderungen, die in dem laufenden Textfeld, im Anschluss an die Öffnung der virtuellen Tastatur, vorgenommen wurden, annulliert. Mit anderen Worten: der Inhalt wird auf den Wert zurückgeführt, den er hatte, bevor die Veränderungen vorgenommen wurden und die virtuelle Tastatur wird ausgeblendet.
Linker Pfeil: wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, wird der Kursor im laufenden Textfeld einen Platz nach links bewegt. Ist der Kursor bereits in der "Null" Position (es kann keine Bewegung mehr nach links vorgenommen werden), hat die Taste keine Auswirkung.
Rechter Pfeil: wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, wird der Kursor im laufenden Textfeld einen Platz nach rechts bewegt. Befindet sich der Kursor bereits hinter dem letzten Zeichen im Textfeld (und kann nicht weiter nach rechts bewegt werden), dann hat die Taste keine Auswirkung.
"Auf" Pfeil: wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, wird der Kursor im laufenden Textfeld einen Platz nach oben bewegt. Befindet sich der Kursor bereits in der ersten Zeile des Textfeldes (oder wenn das laufende Textfeld nur eine Zeile hat: MhwVkTextField)), hat die Taste keine Auswirkung.
"Ab" Pfeil: wenn der Knopf auf der Fernbedienung, der diesem Zeichen entspricht, gedrückt wird, wird der Kursor im laufenden Textfeld einen Platz nach unten bewegt. Befindet sich der Kursor bereits in der letzten Zeile des Textfeldes (oder wenn das laufende Textfeld nur eine Zeile hat: MhwVkTextField)), hat die Taste keine Auswirkung.

[0301] Die "findLocation" Methode bestimmt die Lage der virtuellen Tastatur auf dem Bildschirm und versucht die "bebaute" Fläche zu minimieren.

[0302] Die MhwVkTextField Klasse ist einfach eine Spezialisierung der "TextField" Klasse im "java.awt" Paket. Sie verwaltet zusätzlich einen booleschen Wert, der den Gebrauch (oder auch nicht) der virtuellen Tastatur verwaltet.

[0303] Wenn der boolesche Wert "wahr" ist, wird eine "base" Instanz der "TextField" Klasse geschaffen und ein virtueller Tastaturempfänger, der innerhalb der laufenden Anwendung zur Verfügung steht, wird gleichzeitig

hinzugefügt, wobei die "addKeyListener" Methode angewendet wird. Wenn nicht, dann wird eine "normale" TextArea erzeugt.

[0304] Wenn das Textfeld den Fokus besitzt, dann wenn der Benutzer "OK" drückt und der Boolesche Verknüpfung den Einsatz der virtuellen Tastatur spezifiziert, wird die virtuelle Tastatur angezeigt und erhält den Fokus. Es verwaltet alle Ereignisse und kann das Textfeld ausfüllen. Wenn der Benutzer abermals "OK" drückt, wird sein Text bestätigt und die Tastatur verlässt den Fokus. Wenn der Gebrauch der virtuellen Tastatur nicht beabsichtigt ist (boolesch = unwahr), dann hat das Textfeld das gleiche Verhalten wie ein Standardtextfeld in "java.awt".

[0305] Die MhwVkTextArea Klasse ist einfach eine Spezialisierung der "TextArea" Klasse im "java.awt" Paket. Sie verwaltet zusätzlich einen booleschen Wert, der den Gebrauch (oder auch nicht) der virtuellen Tastatur verwaltet.

[0306] Die Konstruktoren sind genau die gleichen, wie jene der "TextArea" Klasse in dem "java.awt" Paket, mit einem einfachen zusätzlichen Argument: eine Boolesche, die den Gebrauch der virtuellen Tastatur spezifiziert.

[0307] Wenn der boolesche Wert "wahr" ist, wird eine "base" Instanz der "TextField" Klasse geschaffen und ein virtueller Tastaturempfänger, der innerhalb der laufenden Anwendung zur Verfügung steht, wird gleichzeitig hinzugefügt, wobei die "addKeyListener" Methode angewendet wird. Wenn nicht, dann wird eine "normale" TextArea erzeugt.

[0308] Wenn das Textfeld den Fokus besitzt, dann wenn der Benutzer "OK" drückt und der Boolesche Verknüpfung den Einsatz der virtuellen Tastatur spezifiziert, wird die virtuelle Tastatur angezeigt und erhält den Fokus. Es verwaltet alle Ereignisse und kann das Textfeld ausfüllen. Wenn der Benutzer abermals "OK" drückt, wird sein Text bestätigt und die Tastatur verlässt den Fokus. Wenn der Gebrauch der virtuellen Tastatur nicht beabsichtigt ist (boolesch = unwahr), dann hat das Textfeld das gleiche Verhalten wie ein Standardtextfeld in "java.awt".

[0309] Die folgende Sektion beschreibt ferner die Implementierung von Eigenschaften, die oben beschrieben wurden und insbesondere die Wiederherstellung graphischer Objekte (zum Beispiel Textfelder, Schaltflächen, Schieberegler, Listen, Ankreuzfelder, Auswahlmöglichkeiten, und so weiter) aus einer Reihe graphischer Elemente.

[0310] Die **Fig. 39** zeigt eine Schaltfläche, der zum Beispiel aus 9 Elementen besteht: den 4 Ecken (NW 2100, NO 2101, SW 2102, SO 2103), den 4 Seiten (N 2104, O 2105, W 2106 und S 2107) und dem Zentrum (C2108).

[0311] Typischerweise, aber nicht notwendigerweise, ist jedes der neun Elemente quadratisch (4 × 4 Pixel, 8 × 8 Pixel, 16 × 16 Pixel) Die Blöcke N, O, W, S und C Elemente sind innerhalb dem erforderlichen Gebiet gekachelt (bestimmt durch die Grösse der Komponente zusammen mit dem Rand durch das Aussehen bestimmt). Dieses Gebiet muss nicht ein Mehrfaches der Grösse der Elemente sein. Gegenwärtig verläuft die Kachelung von links nach rechts oder von oben nach unten, somit erscheinen alle unvollständig gezeichneten Elemente am unteren Rand und an den rechten Seiten. Man kann ins Auge fassen "zentriert" und "rechts" bündig ausgerichtete Kachelregeln hinzuzufügen, wenn man dies als nützlich empfindet. Im Allgemeinen haben die NW, NO, SW und SO Elemente die gleiche Grösse wie der Rand und daher ist die Kachelung nicht notwendig. Falls sie jedoch kleiner als erforderlich sind, wird die Kachelung automatisch vorgenommen.

[0312] Eine kurze Beschreibung folgt, wie dies implementiert wird. Die Arbeit wird in drei Gebiete aufgeteilt: die Vorrichtung, die in dem nativen WGT Modul eingebaut ist, erlaubt die Erstellung und Zuordnung verschiedener Ansichten mit verschiedenen WGT Trickfenstern; die Entwicklung des nativen "LookPixmap" Moduls, in dem die Funktionen zum Zeichnen der verschiedenen graphischen Merkmale der Komponenten eingeschlossen sind; und der Entwicklung des mhw.awt Java Paketes zur Verbindung mit dem nativen Kode.

[0313] Diese drei Entwicklungsgebiete werden in den folgenden Sektionen beschrieben.

[0314] Der WGT Modul Ansichtsvorrichtung wird nun beschrieben unter Bezug auf die **Fig. 40** und **Fig. 41**.

[0315] Das Ziel dieser Sektion ist zu beschreiben, wie der WGT Modul verändert wurde, so dass er in der Lage ist das Aussehen der verschiedenen verwalteten graphischen Objekte zu steuern. Neue Ansichten kön-

nen in neuen Klassen definiert werden, die, wenn nötig, hinzugefügt werden können, in dem der WGT Modul unberührt bleibt. Eine Standard WGT Ansicht (LookWgt) wird standardmässig zur Verfügung gestellt, umgesetzt und den Trickfenstern zugeordnet, so dass diese Anwendungen, die diese Möglichkeit nicht nutzen wollen, dies nicht müssen.

[0316] Die folgenden Eigenschaften können erforderlich sein: eine Anzahl verschiedener Ansichten kann definiert werden für jeden Objekttyp; eine verschiedene Ansicht kann unabhängig jedem Objekt gegeben werden, und standardmässig kann die Ansicht, die durch den aktuellen WGT definiert ist bei den Objekten eingesetzt werden.

[0317] Zwei verschiedene Techniken zum Zeichnen der Ansicht eines Objekts müssen möglich sein: bitmap und vektorielle Technik.

[0318] Bei der Bitmap Technik wird das Objekt durch Kombination einer Anzahl vorbestimmter Bitmap Elemente gezeichnet, um eine einzige Bitmap zu erzeugen, die dann gezeichnet wird. Um Objekte mit variabler Grösse zu erzeugen, können Elemente wiederholt werden entlang der Seite der Elemente, damit man ein Objekt der erforderlichen Grösse aufbauen kann.

[0319] Weder die Grösse noch die genaue Anordnung jeder dieser Elemente ist definiert. Die Idee ist, dies so offen wie möglich zu lassen, damit die Erzeugung eines neuen Aussehens durch die WGT nicht allzu sehr eingeschränkt ist. Die Betrachtungsklasse muss die Bitmap Elemente bestimmen und die Regeln, um sie zusammen als eine Funktion der erforderlichen Objektgrösse zu montieren.

[0320] Die LookPixmap (nativ) und die PixmapLook (Java) Klassen, die später beschrieben werden, benutzen ausschliesslich diese Methode. Es gibt nichts, um die Entwickler daran zu hindern, ihren eigenen Look zu schaffen, der von der LookPixmap, oder auch nicht, abgeleitet ist, die die vektorielle Methode benutzt.

[0321] Bei der vektoriellen Technik wird das Objekt unter Benutzung einer Reihe grundlegender Zeichenoperationen gezeichnet, so wie zum Beispiel DrawRectancle(), DrawArc(). Die Betrachtungsklasse muss die Regeln zur Zeichnung des Objekts als Funktion der erforderlichen Objektgrösse bestimmen.

[0322] Es ist durchaus möglich eine Betrachtungsklasse durch Kombination von sowohl der Bitmap- als auch der vektoriellen Technik in Betracht zu ziehen. Der Bitmap Ansatz, zum Beispiel, kann für die Grundform einer Schaltfläche eingesetzt werden, während die der vektorielle Ansatz für die Hervorhebung eingesetzt werden könnte.

[0323] Der WGT Modul Look Mechanismus wird nun in mehr Einzelheiten erläutert.

[0324] Die folgenden Mechanismen wurden aufgenommen:

– Unterteilung der aktuellen Zeichenmethode für jedes Objekt in sieben Funktionen, wovon bestimmte in der Betrachtungsklasse, die dem Objekt zugeordnet sind, enthalten sind: Drawbackground(): MhwWgtLookDrawBackground(); DrawForeground(); MhwWgtLookDrawForeground(); MhwWgtLookDrawRelief(); MhwWgtLookDrawFocus(); MhwWgtLookHighlight()

– Erzeugung einer abstrakten Klasse MhwWgtLook

– Erzeugung einer Klasse MhwWgtLookWgt, abgeleitet von MhwWgtLook und realisiert wenn WGT gestartet wird.

– Hinzufügung einer g_TheDefaultLook globalen Variablen, um die Ansicht einzurichten, die jedem Objekt, wenn es erzeugt wird, zugeordnet werden wird, wenn kein spezifischer Look MhwWgtXXXAttsSetLook nicht zugeordnet wurde.

– Hinzufügung einer öffentlichen Methode MhwWgtSetDefaultLook(context), um den standardmässigen Look für Objekte zu wechseln.

– Hinzufügung von zwei öffentlichen Methoden zu den Objektklassen, MhwWgtSetXXXAttrLook(*object, *look) und MhwWgtGetXXXAttsLook(*object).

[0325] Diese Aspekte werden in den folgenden Sektionen dargestellt, beginnend mit der Methode zum Zeichnen.

[0326] Jede Zeichenfunktion ruft nun die folgenden Methoden auf, wenn sie beansprucht wird: DrawBackground(); MhwWgtDrawBackground(); DrawForeground(); MhwWgtLookDrawForeground(); MhwWgtLookDrawRelief(); MhwWgtLookDrawFocus(); und MhwWgtLookHighlight().

[0327] Die beiden MethodenFrawBackground() und DrawBackground() sind Teil von Wgt und werden unberücksichtigt des Aussehens aufgerufen. Die anderen sind eigentlich Zeiger für die entsprechenden Funktionen in der Look Klasse, die mit dem in Frage stehenden Trickfenster verknüpft sind. Auf diese Art und Weise implementiert die Look Klasse die Zeichenfunktionen für diese Teile.
Background (Hintergrund) Dieser ermöglicht der Ansicht (look) hinter dem gesamten Trickfenster zu zeichnen.
Foreground (Vordergrund) Dieser kann benutzt werden um ein Bild zu zeichnen mit einer anderen Graphik über dem Mittelteil des Trickfensters (wobei die Rahmen ausgeschlossen sind).
Relief Dies wird aufgerufen, wenn die relief flag des Trickfensters gesetzt ist und wird eingesetzt um einen Rahmen oder relief für das Trickfenster zu zeichnen.
Focus (Fokus) Dieser wird aufgerufen, wenn das Trickfenster den Fokus besitzt. Es kann benutzt werden um dies graphisch anzuzeigen.
Highlight (Hervorhebung) Diese wird aufgerufen, wenn das Trickfenster hervorgehoben wird. es kann benutzt werden, um dies graphisch anzuzeigen.

[0328] Die abstrakte Klasse MhwWgtLook wird bestimmt und enthält folgendes:
WgtCoreLookClassMethod; WgtCoreLookClassField; WgtCoreLookClass; WgtCoreLookPart; und WgtCoreLookObject.

[0329] Diese sind unten beschrieben.

Felder		
MhwWgtLookClass	*extends ;	
Int32	callsId ;	
String	ClassName ;	
Card16	mask ;	
card8	borderwidthTop ;	
card8	borderwidthBottom ;	
card8	borderwidthLeft ;	
card8	borderwidthRight ;	
MhwWgtColorMapId	Colormap:	
MhwWgtVisual	visual ;	
Card32	blackPixel ;	
Card32	whitePixel ;	
Card32	transparentPixel ;	
MhwWgtColor	verylightGray ;	
MhwWgtColor	lightGray ;	
MhwWgtColor	middleGray ;	
MhwWgtColor	darkGray ;	
MhwWgtColor	highlight ;	
MhwWgtColor	blackColor ;	
MhwWgtColor	whiteColor ;	
METHOD TABLE	set to point to the overloaded methods in derived classes):	
MhwWgtError	(*delete)	(MhwWgtLook) ;
MhwWgtError	(*free)	(MhwWgtLook) ;
MhwWgtError	(*reference)	(MhwWgtLook) ;
Card8	(*getBorderWidth)	(MhwWgtLook, MhwWgtWidget, MhwWgtLookBorder);
MhwWgtError	(*getBorderWidth)	(MhwWgtLook, (MhwWgtLookBorder, Card8) ;
Bool	(*isInstanceOf)	(MhwWgtLook, Int32) ;
MhwWgtError	(*drawBackground)	(MhwWgtLook, (MhwWgtWidget, Int16, Int16, Card6, Card16)
MhwWgtError	(*drawForeground)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawRelief)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16, MhwWgtLookRelief) ;
MhwWgtError	(*undrawRelief	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16);
MhwWgtError	(*drawFocus)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*undrawFocus)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawHighlight)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*undrawHighlight)	(MhwWgtLook, MhwWgtWidget, Int16, Int16,

		Card16, Card16) ;
MhwWgtError	(*drawInset)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawOutset)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawSelectBG)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawCheckSymbol)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16, Bool, MhwWgtCheckStyle) ;
MhwWgtError	(*drawChoiceSymbol)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawAnchor)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16, String, Card32, Int8, MhwWgtLookItemFlag, MhwWgtColor) ;
MhwWgtError	(*drawCross)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*undrawCross)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawItem)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16, String, Card 32, Int8, MhwWgtLookItemFlag);
MhwWgtError	(*redrawItem)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16, String, Card 32, Int8, MhwWgtLookItemFlag, Bool) ;
MhwWgtError	(*drawSlidArrow)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16, MhwWgtSlidDirection);
MhwWgtError	(*drawSlidLift)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16);
MhwWgtError	(*drawCursor)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*undrawCursor)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, Card16, Card16) ;
MhwWgtError	(*drawString)	(MhwWgtLook, MhwWgtWidget, Int16, Int16, String Card32, Int8) ;
MhwWgtDimension	(*getPreferredSizeArrow)	(MhwWgtLook) ;
MhwWgtDimension	(*getPreferredSizeCheck)	(MhwWgtLook) ;
MhwWgtDimension	(*getPreferredSizeChoice)	(MhwWgtLook) ;
MhwWgtDimension	(*getPreferredSizeCross)	(MhwWgtLook) ;
Card8	(*getItemBorderWidth)	(MhwWgtLook) ;

[0330] (Die Card8, Card16 und so weiter, Datentypen sind Pseudonyme für Nummerntypen, die die angegebene Anzahl an Bits haben, zum Beispiel Card8 entspricht einem "char", Card16 entspricht einem "short", usw.)

[0331] Jede Zeichenmethode ist für jeden Objekttyp identisch. Für Ansichten, die eine unterschiedliche Methode für jedes Verfahren benötigen, oder zumindest für bestimmte Methoden, muss die Methode in der Betrachtungsklasse den Trickfenstertyp identifizieren und entsprechend vorgehen.

[0332] Beachten Sie bitte den Zweck der DrawNothing() Methode. Sie gibt einfach OK zurück, wenn sie aufgerufen wird. Bestimmte Eigenschaften sind bei einer gegebenen Ansicht nicht notwendigerweise implementiert, so dass der WGT nicht notwendigerweise das Vorhandensein einer gegebenen Funktion prüfen muss, jede nicht implementierte Funktion sollte auf diese Methode verweisen.

[0333] Beachten Sie auch Mask. Dies ist ein privater, nur lesbarer (read-only) boolescher Datenbereich, wo

jedes Element einer der obigen Methoden entspricht. Wenn ein Element auf 1 gesetzt wird, dann wird die entsprechende Methode neu definiert. Andernfalls wird die Methode nicht neu definiert. Auf diese Art und Weise kann, wenn gewünscht, WGT in einem einzigen Arbeitsgang herausfinden, welche Methoden aufgerufen werden müssen.

[0334] Die Betrachtungsklasse wird zur Bestimmung der Schnittstelle zwischen irgendeiner Betrachtungsbestimmung und WGT benutzt. WGT benutzt nur diese Methoden, um die gewünschte Ansicht darzustellen. Wenn zusätzlichen Funktionalitäten einer Ansicht erwünscht sind, können diese in einer erweiterten Ansichtenstruktur einbezogen werden. aber es liegt an der Anwendung und nicht an WGT, diese Methoden/Parameter zu berücksichtigen. Auf diese Art und Weise können zusätzliche Attribute hinzugefügt werden.

[0335] Eine abgeleitete Ansichtsstruktur muss alle diese Methoden und Attribute enthalten und sie kann ebenfalls ihre eigenen hinzufügen. WGT wird jedoch nur jene Methoden berücksichtigen, die in der MhwWgtLook Struktur definiert sind.

[0336] Nun wird die MhwWgtLookWgt Klasse erläutert.

[0337] Damit bestehende Anwendungen nicht verändert werden müssen, um mit der veränderten Version der WGT kompatibel zu bleiben, wird eine Basisansichten-Klasse erzeugt, die die Ansicht, die WGT Objekte im Moment besitzen, bestimmt und durch WGT realisiert.

[0338] Es ist eine Unterklasse von MhwWgtLook und wird MhwWgtLookWgt genannt. Wenn diese Klasse realisiert wird, werden die Werte aller Zeiger in der Struktur so gesetzt, dass sie auf die WGT-bestimmten Methoden hinweisen.

[0339] Dies Basisklasse enthält sonst nichts anderes – sie definiert ganz einfach die Ansicht, die WGT momentan liefert.

FELDER	
Card8	reliefWidth
MhwWgtColor	reliefColorWhite ;
MhwWgtColor	reliefColorBlack ;
Card8	focusWidth ;
MhwWgtColor	focusColor ;
Card8	highlightWidth ;
MhwWgtColor	highlightColor ;
MhwWgtColor	anchorColorBGNormal ;
MhwWgtColor	anchorColorFGVisited :
MhwWgtColor	anchorColorBGCurrent :
Card16	mask ;
Card8	reliefWidth ;
MhwWgtColor	reliefColorWhite ;
MhwWgtColor	reliefColorBlack ;
Card8	focusWidth ;
MhwWgtColor	focusColor ;
Card8	highlightWidth ;
MhwWgtColor	highlightColor ;
MhwWgtColor	anchorColorBGNormal ;
MhwWgtColor	anchorColorFGVisited ;
MhwWgtColor	anchorColorBGCurrent ;
MhwWgtColorMapId	colorMap ;
Card32	blackPixel ;
Card32	whitePixel ;
Card32	transparentPixel ;
MhwWgtColor	verlylightGray ;
MhwWgtColor	lightGray ;
MhwWgtColor	middleGray ;
MhwWgtColor	darkGray ;
MhwWgtColor	highlight ;
MhwWgtColor	blackColor ;
MhwWgtColor	whiteColor ;

METHOD TABLE :		
MhwWgtError	(*setReliefWidth)	(MhwWgtLkWgt, Card8) ;
MhwWgtError	(*setReliefColorBlack)	(MhwWgtLkWgt, MhwWgtColor) ;
MhwWgtError	(*setReliefColorWhite	(MhwWgtLkWgt, MhwWgtColor) ;
MhwWgtError	(*setFocusWidth	(MhwWgtLkWgt, Card8) ;
MhwWgtError	(*setFocusColor)	(MhwWgtLkWgt, MhwWgtColor) ;
MhwWgtError	(*setHighlightWidth)	(MhwWgtLkWgt, Card8) ;
MhwWgtError	(*setHighlightColor)	(MhwWgtLkWgt, MhwWgtColor) ;
MhwWgtError	(*setAnchorColorBGNormal)	(MhwWgtLkWgt, MhwWgtColor) ;
MhwWgtError	(*setAnchorColorFGVisited)	(MhwWgtLkWgt, MhwWgtColor) ;
MhwWgtError	(*setAnchotColorBGCurrent)	(MhwWgtLkWgt, MhwWgtColor);

[0340] In Bezug auf die WGT Initialisierung muss eine Instanz der MhwWgtLookwgt Klasse geschaffen werden, wenn der WGT gestartet wird. WGT wird dadurch Zugriff auf diese Methoden bekommen, wenn die Anwendung keine unterschiedliche Ansicht spezifiziert. Die g_TheDefaultLook global Variable (unten beschrieben) muss anfänglich so gesetzt werden, dass auf diese Ansicht hinweist.

[0341] Die Vorgänge zur Bestimmung des Aussehens und zum Setzen der Standardansichten werden nun erläutert.

[0342] WGT ist nicht zur Bestimmung und Realisierung neuer Ansichtenobjekte zuständig. Anwendungen müssen dies selbst vornehmen. Alle Ansichtenobjekte müssen sich in eine MhwWgtLook Struktur einbetten lassen. Siehe auch unten unter die Verwaltung von Ansichten.

[0343] Bezüglich der Standardansichten muss ein Feld:
MhwWgtLook *DefaultLook

[0344] Zum MhwWgtContext Objekt hinzugefügt werden, unter Hinweis auf die Instanz von MhwWgtLook, die bei jedem neuen Objekt, das von diesem Kontext erzeugt wurde, angewendet werden muss. Wenn ein neuer WGT Kontext erzeugt wurde, muss das Feld so gesetzt werden, dass es auf WgtBasicLook hinweist.

[0345] Bezüglich der Einrichtung der Standardansicht für einen Kontext, wird eine öffentliche Methode zur Verfügung gestellt:
MgwWgtSetDefaultLook(MhwWgtContext, aContext, MhwWgtLook aLook)
zur Einrichtung des DefaultLook Feldes in aContext um auf den aLook hinzuweisen.

[0346] Zur Verknüpfung einer Ansicht mit einem Objekt, wird das folgende Attribut zur coreAtts Struktur in der core Klasse hinzugefügt:
MhwWgtLook *Look

[0347] Dieses Attribut wird auf diese Art und Weise für jedes gestaltete Objekt erzeugt. Wann auch immer ein Objekt realisiert wird, wird Look eingerichtet, um auf die DefaultLook globale Variable hinzuweisen.

[0348] Zwei neue öffentliche Methoden sollten zur Core-Klasse hinzugefügt werden, um der Look Instanz, die mit dem zu Objekt verknüpft ist, zu ermöglichen, verändert zu werden:
MhwWgtSetXXXAttsLook(MhwWgtWidget anObject, MhwWgtLook aLook) und MhwWgtGetXXXAtts-Look(MhwWgtWidget an Object, MhwWgtLook *aLook).

[0349] Die Verwaltung der Ansichten wird nun erläutert.

[0350] WGT liefert keine Ansichtenverwaltung. Um eine Ansicht, ausser der Standardansicht, zu benutzen, muss eine Anwendung zuerst sicher stellen, dass eine oder mehrere Look Klassen realisiert und initialisiert sind und dass immer wenn sie ein neues WGT Objekt erzeugt, die MhWWgtSetCoreAttsLook() Methode einsetzt, um das Objekt mit der gewünschten Ansicht zu verbinden. Wenn sie eine gegebene Ansicht für alle zukünftigen Trickfenster einsetzen möchte, kann sie die MhwWgtSetDefaultLook() Methode, wie oben beschrieben, benutzen.

[0351] Eine Anwendung, die irgendeine Ansicht, ausser der durch die Standardeinstellung bestimmte, einsetzen möchte, ist für die Erzeugung und Realisierung der Ansicht zuständig. In einem anderen Ausführungsbeispiel der Erfindung kann eine Anwendung Ansichten aus entfernter Quelle herunterladen. In diesem Fall muss jedoch die Anwendung selbst die erforderlichen Klassen, die von MhwWgtLook abgeleitet sind, zur Verfügung stellen.

[0352] Eine Ansicht soll erst durch die Anwendung, die sie erzeugt hat, zerstört werden, bis alle Trickfenster, die sie benutzen, zerstört sind. Dies verlangt die Hinzufügung eines refCounter Feldes, um die Anzahl der "Klienten" zu zählen.

```
Look=MhwNewLook()
MhwLookRef(Look);
...
MhwLookUnref(Look);
Look=0
```

[0353] Dies ersetzt free (look). Die Ansicht (look) wird augenblicklich zerstört, wenn ihr refCounter Feld 0 entspricht.

```
SetXxxLook(widget), look) (
If(widget->core.look)
MhwLookUnref(widget->core.look);
Widget->core.look=look;
If(look)
MhwLookRef(look);
```

)

[0354] Die WGT Modulliste der APIs zur Implementierung von Looks folgt hier:

```
extern MhwWgtError MhwWgtLookInitDefault     (MhwWgtLookClass*,
                                              MhwWgtLookAtts*);
extern MhwWgtError MhwWgtLookInitClass        (void) ;
extern MhwWgtError MhwWgtLookAttsGetBorderWidthBTLR (MhwWgtLookAtts*,
                                    Card8*, Card8*, Card8*, Card8*);
extern MhwWgtError MhwWgtLookAttsGetBorderWidthBottom  (MhwWgtLookAtts*,
                                                 Card8*)
extern MhwWgtError MhwWgtLookAttsGetBorderWidthLeft(MhwWgtLookAtts*,
                                              Card8*)
extern MhwWgtError MhwWgtLookAttsGetBorderWidthRight(MhwWgtLookAtts*,
                                              Card8*)
extern MhwWgtError MhwWgtLookAttsGetBorderWidthTop(MhwWgtLookAtts*,
                                              Card8*)
extern MhwWgtError MhwWgtLookAttsGetDeafult  (MhwWgtLookClass*,
                                              MhwWgtLookAtts*)

extern MhwWgtError MhwWgtLookAttsInit         (MhwWgtLookAtts*);
extern MhwWgtError MhwWgtLookAttsSetBorderWidthBTLR (MhwWgtLookAtts*,
                                    Card8*, Card8*, Card8*, Card8*);
extern MhwWgtError MhwWgtLookAttsSetBorderWidthBottom
                                              (MhwWgtLookAtts*,   Card8*);
extern MhwWgtError MhwWgtLookAttsSetBorderWidthLeft
                                              (MhwWgtLookAtts*,   Card8*);
extern MhwWgtError MhwWgtLookAttsSetBorderWidthRight
                                              (MhwWgtLookAtts*,   Card8*);
extern MhwWgtError MhwWgtLookAttsSetBorderWidthTop
                                              (MhwWgtLookAtts*,   Card8*);
```

[0355] Der Look/LookPixmap Modul wird nun in weiteren Einzelheiten beschrieben.

[0356] Die MhwWgtLookPixmap Klasse wird aus der MhwWgtLook Klasse abgeleitet, wie oben beschrieben wurde. Im Wesentlichen arbeitet sie durch Rekonstruktion der verschiedenen Elemente jeder Komponente, um ein graphische Bild der erforderlichen Grösse zu erzeugen, wie oben beschrieben wurde.

[0357] Diese Bilder werden für Folgendes benötigt: Hintergründe für Schaltflächen; Relief (das ist zum Beispiel der Rand um die aktive Zone von Textfeldern); ein Symbol für die Auswahl Komponente; Ankreuzfelder; Schiebregler; und Schiebereglerlifte.

[0358] Unter Bezug auf die Fig. 40 werden Bilder nicht komprimiert, um Bereitstellzeit zu vermindern, jedoch in einem besonderen Format gespeichert, das so ausgelegt ist, um so weit wie möglich belegten Speicherplatz zu minimieren. Die Farbe aller Pixel (**2152**) wird in einem einzigen Bit beschrieben, welche die Indexnummer (**2151**) der Farbe in der aktuellen Farbtabelle darstellt. Die Fig. 40 zeigt einen beispielhaften Puffer, image, der ein 4 × 4 Bild (**2153**) enthält.

[0359] Das Bild (**2153**) in der Fig. 40 würde folgendermassen gespeichert werden:
Card8 slidLiftSeVrImage4 [4] [4] =
(
(0, 0, 0, 1),
(0, 0, 1, 2),
(0, 1, 2, 3),
(1, 2, 3, 4)

);

[0360] Die LookPixMap Bildstruktur wird nun erläutert.

[0361] Zur Identifizierung der Grösse eines Bildpuffers, wird eine Struktur, LookPixmapImage, definiert, einschliesslich dem oben beschriebenen Bildpuffer, zusammen mit der Breite und Höhe des Bildes. Die Struktur, wie unten beschrieben, wird eingesetzt, um die Daten für jedes graphische Element aufzunehmen.

```
Type struct
(
        Card8    *imageData;          Puffer mit Bild. Es ist eine zweidimensionale
                                      Anordnung der Card8, wobei jedes Element die
                                      Indexnummer in der Palette der Farben, die auf
                                      der Position in der Anordnung dargestellt werden
                                      sollen, enthält.
        Card8    *maskData;           Puffer, der die Maske enthält. Es ist eine
                                      eindimensionale Anordnung der Card8, mit einem
                                      Bit pro Pixel. (Format, so wie es vom Bildmodul
                                      zurückgespielt wird)
        Card16    width;              Bildbreite
        Card16    height;             Bildhöhe
        Card8    isOpaque;            0 ist, wenn das Bild keine transparenten
                                      Elemente enthält, 1, wenn vollständig
                                      undurchsichtig.
)
LookPixmapImage;
```

[0362] Bilder können verschiedene Grössen haben, obwohl sie für einen gegebenen Elementetyp hauptsächlich gleich sein werden. Das zentrale Element (xxxxxC), jedoch, hat oft die Grösse 1 × 1. Die MhwWgtLookPixmapAllImages Struktur fasst alle Bildelemente, wie unten dargestellt, zusammen:

```
Typedef struct
(
LookPixmapImage *relnoNW;          Nordwest Ecke des Reliefs – Status Normal
LookPixmapImage *relnoSW,          Südwest Ecke des Reliefs – Status Normal
```

```
LookPixmapImage *relnoNE;          Nordost Ecke des Reliefs – Status Normal
LookPixmapImage *relnoSE;          Südost Ecke des Reliefs – Status Normal
LookPixmapImage *relnoN;           Nord Rand des Reliefs – Status Normal
LookPixmapImage *relnoW;           West Rand des Reliefs – Status Normal
LookPixmapImage *relnoE;           Ost Rand des Reliefs – Status Normal
LookPixmapImage *relnoS;           Sued Rand des Reliefs – Status Normal
LookPixmapImage *relnoC;           Mittelfläche des Reliefs – Status Normal
LookPixmapImage *relfoNW;          Status – nur Fokus
LookPixmapImage *relfoSW;
LookPixmapImage *relfcNW;
LookPixmapImage *relfoSE;
LookPixmapImage *relfoN;
LookPixmapImage *relfoW;
LookPixmapImage *relfoE;
LookPixmapImage *relfoS;
LookPixmapImage *relfoC;
LookPixmapImage *relhiNW;          Status – nur Hervorhebung
LookPixmapImage *relhiSW;
LookPixmapImage *relhiNE;
LookPixmapImage *relhiSe;
LookPixmapImage *relhiN;
LookPixmapImage *relhiW;
LookPixmapImage *relhiE;
LookPixmapImage *relhiE;
LookPixmapImage *relhiS;
LookPixmapImage *relhiC;
LookPixmapImage *relfhNW;          Status – Fokus und Hervorhebung
LookPixmapImage *relfhSW;
LookPixmapImage *relfhNE;
LookPixmapImage *relfhSE;
LookPixmapImage *relfhN;
LookPixmapImage *relfhW;
LookPixmapImage *relfhE;
 LookPixmapImage *relfhS;
LookPixmapImage *relfhC;
 LookPixmapImage *butnoNW;         Schaltflächenelemente
LookPixmapImage *butnoSW;
LookPixmapImage *butnoNE;
LookPixmapImage *butnoSE;
LookPixmapImage *butnoN;
LookPixmapImage *butnoW;
LookPixmapImage *butnoE;
LookPixmapImage *butnoS;
LookPixmapImage *butnoC;
LookPixmapImage *butfoNW;
LookPixmapImage *butfoSW;
LookPixmapImage *butfoNE;
LookPixmapImage *butfoSE;
LookPixmapImage *butfoN;
LookPixmapImage *butfow;
LookPixmapImage *butfoE;
```

```
LookPixmapImage *butfoS;
LookPixmapImage *butfoC;
LookPixmapImage *buthiNW;
LookPixmapImage *buthiSW;
LookPixmapImage *buthiNE;
LookPixmapImage *buthiSE;
LookPixmapImage *buthiN;
LookPixmapImage *buthiW;
LookPixmapImage *buthiE;
LookPixmapImage *buthiS;
LookPixmapImage *buthiC;
LookPixmapImage *butfhNW;
LookPixmapImage *butfhSW;
LookPixmapImage *butfhNE;
LookPixmapImage *butfhSE;
LookPixmapImage *butfhN;
LookPixmapImage *butfhW;
LookPixmapImage *butfhS;
LookPixmapImage *butfhS;
LookPixmapImage *butfhC;
LookPixmapImage *choice;          Wahlsymbol
LookPixmapImage *chcklna;         Ankreuzfeldsymbol – Typ 1 nicht gewählt, kein Fokus
LookPixmapImage *chck1a;          Ankreuzfeldsymbol – Typ 1 ausgewählt, kein Fokus
LookPixmapImage *chck1flna;       Ankreuzfeldsymbol – Typ 1 nicht gewählt, Fokus
LookPixmapImage *chck1fla;        Ankreuzfeldsymbol – Typ 1 ausgewählt, Fokus
LookPixmapImage *chck2na;         Ankreuzfeldsymbol – Typ 2 nicht gewählt, kein Fokus
LookPixmapImage *chck2a;          Ankreuzfeldsymbol – Typ 2 ausgewählt, kein Fokus
LookPixmapImage *chckf2na;        Ankreuzfeldsymbol– Typ 2 nicht gewählt, Fokus
LookPixmapImage *chckf2a;         Ankreuzfeldsymbol – Typ 2 ausgewählt, Fokus
LookPixmapImage *slidNeVr;        Schieberegler Hintergrundelemente
LookPixmapImage *slidEVr;
LookPixmapImage *slidSeVr;
LookPixmapImage *slidSwHr;
LookPixmapImage *slidSHr;
LookPixmapImage *slidSeHr;
LookPixmapImage *slidSeVrHr;
LookPixmapImage *slidLiftNeVr;
 LookPixmapImage *slidLiftEVr;
LookPixmapImage *slidLiftSeVr;
LookPixmapImage *slidLiftSwHr;
LookPixmapImage *slidLiftSHr;
LookPixmapImage *slidLiftSeHr;
)
MhwWgtLookPixmapAllImages;
```

[0363] Diese Sektion beschreibt das LookPixmap Modul einschliesslich der LookPixmap Klasse, die geschaffen wurde, um eine Reihe verschiedener Ansichten beim Webbrowser einzusetzen.

[0364] Dieser Modul enthält die folgenden Ressourcendateien: MhwWgtLookPixmap.h; MhwWgtLookPixmapStruct.h; MhwWgtLookPixmapClass. c; MhwWgtLookPixmapPrivate. c; MhwWgtLookPixmapImages. h; MhwWgtLookPixmapImages2. h; MhwWgtLookPixmapImages3.h; MhwWgtLookPixmapImages4.h; MhwWgtLookPixmapImages5. h; und MhwWgtLookPixmapImages6. h;

[0365] Der LookPixmap Modul wird nun beschrieben einschliesslich Einzelheiten der bevorzugten Methode zur Bildung und Gebrauch von LookPixmap Objekten.

[0366] Jede Software die WGT zur Schaffung und Verwaltung von Trickfenstern einsetzt, kann das LookPixmap Modul benutzen, um alternative Ansichten den WGT Trickfenstern zu liefern. Für eine Anwendung zum Einsatz der LookPixmap Ansicht, muss ein LookPixmap Objekt geschaffen werden. Dies kann unter Einsatz des folgenden Kodes erledigt werden:

```
MhwWgtLkWebClass                    PixmapLook;
MhwWgtLkWeb                         PixmapLookObject;
MhwWgtLkWebAtts                     LookPixmapValues;
MhwWgtError                         WgtErr;

 WgtErr = MhwWgtLkWebInitClass();
WgtErr = MhwWgtLkWebAttsInit(LookPxmapValues);
WgtErr = MhwWgtLkWebInitDefault/&PixmapLook, &LookPixmapValues);
PicmapLookObject = MhwWgtLkWebNew(6LookPixmapValues);
```

[0367] Eine Methode zur Einrichtung der Standardansicht wird nun beschrieben.

[0368] Eine Anwendung kann ein gegebenes Ansichtobjekt standardmässig nutzen. Standardmässig ist die Standardansicht das LookWgt Objekt, das durch WGT erzeugt wurde. Um eine andere Standardansicht einzurichten, unter der Voraussetzung, dass sie, wie oben beschrieben, bereits erstellt wurde, kann die folgende Funktion verwendet werden:
MhwwgtSetDefaultLook((MhwWgtLook) PixmapLookObject);

[0369] Alle anschliessend erzeugten WGT Trickfenster werden mit der LookPixmap Ansichtenklasse verknüpft und nicht die WGT Standard LookWgt.

[0370] Eine Anwendung kann eine Ansicht für einen gegebenen Typ von Trickfenster, oder ein gegebenes Trickfenster wählen oder einrichten, wie nun beschrieben werden wird.

[0371] Eine Anwendung kann die Ansicht für ein gegebenes Trickfenster einrichten, wenn das Objekt durch Aufruf der folgenden Funktion erzeugt wird, kurz bevor das Trickfenster erzeugt wird:
MhwWgtXXXAttsSetLook(MhwWgtXXXAtts*, MhwWgtLook);

[0372] Sie kann die Ansicht eines Objektes nach Erzeugung einrichten, in dem die folgende Funktion benutzt wird:
MhwWgtXXXSetLook(MhwWgtXXXWidget*, MhwWgLook);
(wobei xxx der Typ des Trickfensters ist, zum Beispiel LIST).

[0373] Die Methode zum Einsatz der LookPixmap Bilder wird nun erläutert.

[0374] Ein einzelnes LookPixmap Objekt setzt einen einzigen Satz Bilder ein. Man kann ohne Zweifel die Ansicht dramatisch ändern, nur durch Wechseln der Bilder.

[0375] Man kann die Bilder, die für ein gegebenes LookPixmap Objekt benutzt werden, ändern, in dem die folgenden Funktionen aufgerufen werden:
MhwWgtLookPixmapSetImages(MhwWgtLookPixmap*, MhwWgtLookPixmapAllImages*);

[0376] Richtet die Bilder, die für alle Trickfenster benutzt werden, die das spezifizierte LookPixmap Objekt einsetzen, in der spezifizierten Bilderreihe ein.
MhwWgtLookPixmapSetDefaultImages(MhwWgtLookPixmap*);

[0377] Richtet die Bilder, die für alle Trickfenster, die das spezifizierte LookPixmap Objekt einsetzen, in die spezifizierte Standardbilderreihe ein.

[0378] Wenn man verschiedene Bilder für verschiedene Trickfenster benutzen will, muss man ein LookPixmap Objekt für jede erforderliche Bilderreihe schaffen. Jedes Bild wird dann mit der entsprechenden Ansicht zugeordnet und man verknüpft dann jede Ansicht mit dem entsprechenden Trickfenster.

[0379] Die API des LookPixmap Moduls wird nun beschrieben unter Bezug auf die **Fig. 39** und **Fig. 41**.

[0380] Die folgenden öffentlichen APIs stehen zur Verfügung:

MhwWgtLookPixmapSetImages()

Prototyp

MhwWgtError MhwWgtLookPixmapSetImages(MhwWgtLkWeb aLook, MhwWgtLookPixmapAllImages* someImages);

Beschreibung:

[0381] Richtet die Bilderreihen, die von aLook benutzt werden bei den Bilderreihen ein, die auf someImages hinweisen.

Parameter:

aLook	Das MhwWgtLkWeb Objekt, mit dem die Bilder verknüpft werden sollen, someImages
someImages	Die Bilderreihe, die mit aLook verknüpft werden sollen

[0382] Meldet zurück:
MHW_WGT_SUCCESS

MhwWgtLookPixmapSetImagesID()

Prototyp

MhwWgtError MhwWgtLookPixmapSetImagesID (MhwWgtLkWeb aLook, Card8 anImageID;

Beschreibung:

[0383] Richtet die Bilderreihen, die von aLook benutzt werden bei den Bilderreihen ein, die in MhwWgtLookPixmap programmiert sind und durch anImageID identifiziert werden.

Parameter:

aLook	Das MhwWgtLkWeb Objekt, mit dem die Bilder verknüpft werden sollen, und durch anImageID identifiziert sind.
anImageID	Die Kennung der Bilderreihe, programmiert in MhwWgtLookPixmap zur Verknüpfung mit aLook

[0384] Meldet zurück:
MHW_WGT_SUCCESS

MhwWgtLookPixmapSetDefaultImages()

Prototyp:

MhwWgtError MhwWgtLookPixmapSetDefaultImages (MhwWgtLkWeb aLook

Beschreibung:

[0385] Richtet die Bilderreihen, die von aLook benutzt werden bei den Bilderreihen ein, die in MhwWgtLookPixmap programmiert sind und durch die Kennung 1 identifiziert werden.

Parameter:

aLook	Das MhwWgtLkWeb Objekt, mit dem die Bilder verknüpft werden sollen, und durch anImageID identifiziert sind.

[0386] Meldet zurück:
MHW_WGT_SUCCESS

MhwWgtLookPixmapLoadImages()

Prototyp:

MhwWgtError MhwWgtLookPixmapSetLoadImage (MhwWgtLkWeb aLook, Int32 anElementID, Int32 aWidth, Int32 aHeight, Card8* anImageBuffer);

Beschreibung:

[0387] Kommt zum Einsatz zum Wechseln eines einzelnen Bildes durch die momentane MhwWgtLookPixmapAllImages Struktur, um auf das festgelegte Bild hinzuweisen. Erzeugt eine LookPixelmapImage Struktur und richtet die laufende MhwWgrLookPixmapAllImages Struktur ein, auf die durch aLook hingewiesen wurde, um auf diese LookPixmapImage für das Element hinzuweisen, spezifiziert durch anElementID.

Parameter:

aLook	Das MhwWgtLkWeb Objekt mit dem die durch anImageID identifizierten Bilder verknüpft werden.
anElementID	Die Kennung der zu wechselnden Elemente
aWidth	Die Breite in Pixel des neuen Bildes
aHeight	Die Höhe in Pixel des neuen Bildes
anImageBuffer	Der Puffer der die neuen Bilddaten enthält

[0388] Meldet zurück:
MHW_WGT_SUCCESS

LookPixmapMakeImageFromElements()

Prototyp:

MhwWgtError LookPixmapMakeImageFromElements (LookPixmapImage* elemN

LookPixmapImage* elemE,
LookPixmapImage* elemW,
LookPixmapImage* elemS,
LookPixmapImage* elemNW,
LookPixmapImage* elemE,
LookPixmapImage* elemSW,
LookPixmapImage* elemSE,
LookPixmapImage* elemC, Card16 anX,
Card16 aY, Card16 aWidth, Card16
aHeight, MHWWindowID aWindow,
LookPixmapDrawMode aDrawMode);

Beschreibung:

[0389] Nimmt die 9 Bilder des elemX und zeichnet sie in das bestimmte MHW Fenster. Die Regeln zum Aufbau des Bildes sind durch aDrawMode bestimmt (momentan besteht nur MHW_WGT_LIKWEB_DRAW_NORMAL). Das endgültige Bild wird in dem Fenster mit der oberen linken Ecke bei (anX, aY) positioniert und mit einer Grösse aWidth × aHeight gezeichnet.

[0390] Wenn eines oder mehrere Elemente Null Grösse haben (entweder elemXX.width oder elemXX.height

ist null) wird dieses Element nicht gezeichnet.

Parameter:

elemN, elemE, elemW, elemS, elemNW, elemE, elemSW, elemC, elemSE:	das jeweils zu zeichnende Bild oben, rechts, links, unten, obere linke Ecke, obere rechte Ecke, unten linke Ecke, unten rechte Ecke und Mitte.
anX	Die x-Position im Fenster, aWindow zum Zeichnen des endgültigen Bildes.
aY	Die y-Position im Fenster, aWindow zum Zeichnen des endgültigen Bildes.
aWidth	Die Breite, in Pixel, des neuen Bildes
aHeight	Die Höhe, in Pixel, des neuen Bildes
aWindow	Das Fenster, in dem das gebaute Bild gezeichnet werden soll.
aDrawMode	Der Modus in dem das Bild aufgebaut werden soll. MHW_WGT_LIKWEB_DRAW_NORMAL: Setzt die NW (**2100**), NO (**2101**, SW (**2102**), und S = (**2103**) Elemente in den vier Ecken (**2200**, **2201**, **2202**, **2203**) ohne Kacheln. Kachelt horizontal die N und S Elemente. Kachelt vertikal die W und O Elemente. Kachelt sowohl horizontal als auch vertikal die C Elemente. Obwohl es für alle Bildgrössen funktioniert, wird nur zugesichert, dass es das Gebiet korrekt kachelt, vorausgesetzt dass das zentrale Gebiet (**2208**) rechteckig ist.

[0391] Meldet zurück:
MHW_WGT_SUCCESS

[0392] Das Mhw.awt Java Schnittstellenpaket wird nun erläutert.

[0393] Drei Java Klassen wurden entwickelt, so dass der Look Mechanismus, der durch WGT bestimmt ist durch Java Anwendungen ausgeschöpft werden kann. Diese sind: mhw.awt.WgtLook; und mhw.awt.PixmapLook.

[0394] Die Look Klasse ist die abstrakte Klasse, die der MhwWgtLook Klasse entspricht, die oben beschrieben wurde.

[0395] Die WgtLook Klasse wird eingesetzt, um Instanzen der WGT Klasse MhwWgtLookWgt zu erzeugen und zu bearbeiten.

[0396] Die PixmapLook Klasse wird eingesetzt, um Bilder, die von der WgtLook Klasse benutzt werden zu speichern.

[0397] Die Mhw.awt.PixmapLook API wird nun beschrieben, beginnend mit Einzelheiten der Konstruktoren.

PixmapLook

public PixmapLook()

[0398] Erzeugt eine neue Instanz eines PixmapLook Objekts mit den auf Standardbilder voreingestellten Bildern (ID=1).

PixmapLook

public PixmapLook(int imageID)

[0399] Erzeugt eine neue Instanz eines PixmapLook Objekts mit Bildern die zu den durch imageID spezifi-

zierten Bildern voreingestellt sind.

[0400] Die Methoden werden nun beschrieben.

SetImages

Public void SetImages()

[0401] Richten die laufenden Bilder für dieses PixmapLook Objekt auf den Standardwert (ID=1) ein.

LoadImage

```
public void        LoadImage          int elementID
                                      int width ,
                                      int height ,
                                      Byte [] buffer )
```

[0402] Lädt ein festgelegtes Bildelement. Jedes PixmapLook Objekt hat einen Satz von (94) Bildern, die damit verknüpft sind. Diese Bilder stellen die Elemente der graphischen Komponenten wie folgt dar:
0 relnoNW; 1 relnoSW; 2 relnoNE; 3 relnoSE; 4 relnoN; 5 relnoW; 6 relnoE; 7 relnoS; 8 relnoC; 9 relfoNW; 10relfoSW; 11relfoNE; 12relfoSE; 13relfoN; 14relfoW; 15relfoE; 16relfoS; 17relfoC; 18relhiNW; 19relhiSW; 20relhiNE; 21relhiSE; 22relhiN; 23relhiW; 24relhiE; 25relhiS; 26relhiC; 27relfhNW; 28relfhSW; 29relfhNE; 30relfhSE; 31relfhN; 32relfhW; 33relfhE; 34relfhS; 35relfhC; 36 butnoNW; 37 butnoSW; 38 butnoNE; 39 butnoSE; 40 butnoN; 41 butnoW; 42 butnoE; 43 butnoS; 44 butnoC; 45 butfoNW; 46 butfoSW; 47 butfoNE; 48 butfoSE; 49 butfoN; 50 butfoW; 51 butfoE; 52 butfoS; 53 butfoC; 54 buthiNW; 55 buthiSW; 56 buthiNE; 57 buthiSE; 58 buthiN; 59 buthiW; 60 buthiE; 61 buthiS; 62 buthiC; 63 butfhNW; 64 butfhSW; 65 butfhNE; 66 butfhSE; 67 butfhN; 68 butfhW; 69 butfhE; 70 butfhS; 71 butfhC; 72 choice; 73 chcklno; 74 chcklse; 75 chcklfo; 76 chcklfs; 77 chck2no; 78 chck2se; 79 chck2fo; 80 chck2fs; 81 slidNeVr; 82 slidEVr; 83 slidSeVr; 84 slidSwHr; 85 slidSHr; 86 slidSeHr; 87 slidSeVrHr; 88 slidLiftNeVr; 89 slidLiftEvr; 90 slidLiftSeVr; 91 slidLifSwHr; 92 slidLiftSHr; 93 slidLiftSeHr.
width spezifiziert die Breite des genehmigten Bildes
height spezifiziert die Höhe genehmigten Bildes
buffer enthält die Bilddaten. Diese sind in Form einer Byteanordnung, wobei jedes Byte den Farbpalettenindex, der für jeden Pixel eingesetzt werden muss, darstellt. Der Index der für Pixel (x, y) benutzt wird ist buffer[buffer(y*width) + x].

MakeImageFromElements

```
Public void        akeImageFromElements(int [] widths,
                   Int [] heights,
                   Byte [] [] buffers,
                   Int anX,
                   Int aY
                   Int aWidth,
                   Int aHeight,
                   Java.awt.Component aComponent
```

[0403] Baut ein Bild, basierend auf 9 Elementen/N, E, W, S, NW, NO, SW, SO, C) und zeichnet es in ein Fenster, das mit der Komponente aComponent verknüpft ist, mit der oberen linken Ecke bei (anX, aY9 und mit der Grösse aWidth × aHeight. Die Bildpuffer werden als zweidimensionale Anordnung weiter gegeben, wobei eine Dimension die Bildnummer angibt (0–8 entspricht N, O, W, S, NW. NO. SW, SO, C) und die andere enthält die Daten. Die Breiten und die Höhen jedes Puffers werden in die Breiten- und Höhenfelder gegeben.

DownloadLookDir

Public java.lang.String DownloadLookDir()

[0404] sLädt ein "look" Verzeichnis aus dem MPEG Strom herunter. Meldet eine Zeichenkette, die Daten für jede Ansichtenbilder, die heruntergeladen werden können, enthält und die durch Zeilenvorschubzeichen getrennt sind. Die Zeilennummer eines Titels (0 bis n – 1) entspricht der Kennung (ImageSet), die mit Image Sek-

tion DownLoadLookImages(intImageSet) benutzt wird.

[0405] Das Verzeichnis ist in der Tat eine einfache Textdatei, die die Zeichenkette, die zurückgemeldet wird, enthält. Der Dateipfad ist fest programmiert in der Quelle – gegenwärtig /home/users/mstoddar/mh-plus/util/looks/images.dir.

[0406] Dies kann nach Bedarf geändert werden. Diese kann daher im Dekoder benutzt, um automatisch aus dem MPEG Strom herunter zu laden.

[0407] Das Format der Datei ist:
<Image Set Title 1><\t><Image Set Description><\t><URL Resource><\t><URL Preview><\n>
<Image Set Title 2><\t><Image Set Description><\t><URL Resource><\t><URL Preview><\n>
<Image Set Title 3><\t><Image Set Description><\t><URL Resource><\t><URL Preview><\n>
<Image Set Title 4><\t><Image Set Description><\t><URL Resource><\t><URL Preview><\n>

[0408] Meldet "" zurück, wenn nicht erfolgreich.

DownloadLookImages

public void DownloadLookImages(int ImageSet)

[0409] Lädt eine neue Reihe Bilder aus dem MPEG Strom herunter, die durch die Zeilennummer (0 bis n – 1) einer der Eingänge, die in DownLoadLookDir() zurückgegeben wurden, identifiziert sind und ordnet sie dieser Ansicht zu.

[0410] Die Datei, die die Daten enthält hat das folgende Format:
WWWWHHHHWWWWHHHH ... eine Reihe 4-Bitketten (leading spaces), die die dezimalwerte der Breiten und Höhen aller 94 Bilder beinhalten (in der gleichen Reihenfolge wie in der Methode LoadImage()). Die Datenpuffer für jedes folgende Bild, auch im selben Format wie LoadImage(). Es wird keine Angleichung zwischen den Bildern vorgenommen, der Beginn des nächsten Bildes startet beim Byte das dem vorhergehenden folgt.

[0411] Der Dateipfad ist fest programmiert in der Quellegegenwärtig
/home/users/mstoddar/mhplus/util/looks/images.<ImageSet>.

[0412] Dies kann nach Bedarf geändert werden. Diese kann daher im Dekoder benutzt, um automatisch aus dem MPEG Strom herunter zu laden.
public void DownloadLookImages(string ImageURL)

[0413] Lädt eine Reihe neuer Bilder aus dem MPEG Strom herunter, durch die spezifizierte URL identifiziert und ordnet sie der Ansicht zu.

[0414] Die Datei, die die Daten enthält hat das oben angegebene Format.

[0415] Die Struktur der Dateien wird nachstehend aus Vereinfachungsgründen im C Syntax aufgeführt:

Card8 relnoNWwidth [4]	Zeichenfolgedarstellung mit Dezimalwert
Card8 relnoSWheight [4]	
Card8 relnoSWwidth [4]	
Card8 relnoSWheight [4]	
Card8 relnoNEwidth [4]	
Card8 relnoNEheight [4]	
Card8 relnoNheight [4]	
Card8 relnoWwidth [4]	
Card8 relnoWheight [4]	
Card8 relnoEwidth [4]	
Card8 relnoEheight [4]	
Card8 relnoSwidth [4]	

Card8 relnoSheight [4]
Card8 relnoCwidth [4]
Card8 relnoCheight [4]
Card8 relfoNWwidth [4]
Card8 relfoNWheight [4]
Card8 relfoSWwidth [4]
Card8 relfoNEwidth [4]
Card8 relfoNEheight [4]
Card8 relfoSEwidth [4]
Card8 relfoSEheight [4]
Card8 relfoNwidth [4]
Card8 relfoNheight [4]
Card8 relfoWwidth [4]
Card8 relfoWheight [4]
Card8 relfoEwidth [4]
Card8 relfoEheight [4]
Card8 relfoSwidth [4]
Card8 relfoSheight [4]
Card8 relfoCwidth [4]
Card8 relfoCheight [4]
Card8 relhiNWwidth [4]
Card8 relhiNWheight [4]
Card8 relhiSWwidth [4]
Card8 relhiNEwidth [4]
Card8 relhiNEheight [4]
Card8 relhiSEwidth [4]
Card8 relhiSEheight [4]
Card8 relhiNwidth [4]
Card8 relhiNheight [4]
Card8 relhiWwidth [4]
Card8 relhiWheight [4]
Card8 relhiEwidth [4]
Card8 relhiEheight [4]
Card8 relhiSwidth [4]
Card8 relhiSheight [4]
Card8 relhiCwidth [4]
Card8 relhiCheight [4]
Card8 relfhNWwidth [4]
Card8 relfhNWheight [4]
Card8 relfhSWwidth [4]
Card8 relfhNEwidth [4]
Card8 relfhNEheight [4]
Card8 relfhSEwidth [4]
Card8 relfhSEheight [4]
Card8 relfhNwidth [4]
Card8 relfhNheight [4]
Card8 relfhWwidth [4]

Card8 relfhwheight[4]
Card8 relfhEwidth [4]
Card8 relfhEheight [4]
Card8 relfhSwidth [4]
Card8 relfhSheight [4]
Card8 relfhCwidth [4]
Card8 relfhCheight [4]
Card8 butnoNWwidth [4]
Card8 butnoNWheight [4]
Card8 butnoSWwidth [4]
Card8 butnoNEwidth [4]
Card8 butnoNEheight [4]
Card8 butnoSEwidth [4]
Card8 butnoSEheight [4]
Card8 butnoNwidth [4]
Card8 butnoNheight [4]
Card8 butnoWwidth [4]
Card8 butnoWheight [4]
Card8 butnoEwidth [4]
Card8 butnoEheight [4]
Card8 butnoSwidth [4]
Card8 butnoSheight [4]
Card8 butnoCwidth [4]
Card8 butnoCheight [4]
Card8 butfoNWwidth [4]
Card8 butfoNWheight [4]
Card8 butfoSWwidth [4]
Card8 butfoNEwidth [4]
Card8 butfoNEheight [4]
Card8 butfoSEwidth [4]
Card8 butfoSEheight [4]
Card8 butfoNwidth [4]
Card8 butfoNheight [4]
Card8 butfoWwidth [4]
Card8 butfoWheight [4]
Card8 butfoEwidth [4]
Card8 butfoEheight [4]
Card8 butfoSwidth [4]
Card8 butfoSheight [4]
Card8 butfoCwidtht [4]
Card8 butfoCheight [4]
Card8 buthiNWwidtht [4]
Card8 buthiNWheight [4]
Card8 buthiSWwidtht [4]
Card8 buthiNEwidtht [4]
Card8 buthiNEheight [4]

Card8 buthiSEwidth [4]
Card8 buthiSEheight [4]
Card8 buthiNwidth [4]
Card8 buthiNheight [4]
Card8 buthiWwidth [4]
Card8 buthiWheight [4]
Card8 buthiEwidth [4]
Card8 buthiEheight [4]
Card8 buthiSwidth [4]
Card8 buthiSheight [4]
Card8 buthiCwidth [4]
Card8 buthiCheight [4]
Card8 butfhNWwidth [4]
Card8 butfhNWheight [4]
Card8 butfhSWwidth [4]
Card8 butfhNEwidth [4]
Card8 butfhNEheight [4]
Card8 butfhSEwidth [4]
Card8 butfhSEheight [4]
Card8 butfhNwidth [4]
Card8 butfhNheight [4]
Card8 butfhWwidth [4]
Card8 butfhWheight [4]
Card8 butfhEwidth [4]
Card8 butfhEheight [4]
Card8 butfhSwidth [4]
Card8 butfhSheight [4]
Card8 butfhCwidth [4]
Card8 butfhCheight [4]
Card8 choicewidth [4]
Card8 choiceheight [4]
Card8 chcklnowidth [4]
Card8 chckinoheight [4]
Card8 chcklewidth [4]
Card8 chcklseheight [4]
Card8 chcklfowidth [4]
Card8 chcklfoheight [4]
Card8 chcklfswidth [4]
Card8 chcklfsheight [4]
Card8 chck2nowidth [4]
Card8 chck2noheight [4]
Card8 chck2sewidth [4]
Card8 chck2seheight [4]
Card8 chck2fowidth [4]
Card8 chck2foheight [4]
Card8 chck2fswidth [4]
Card8 chck2fsheight [4]
Card8 slidNeVrwidth [4]
Card8 slidNeVrheight [4]
Card8 slidEVrwidth [4]
Card8 slidVrheight [4]

Card8 slidSeVrwidth [4]
Card8 slidSeVrheight [4]
Card8 slidSwHrwidth [4]
Card8 slidSwHrheight [4]
Card8 slidSHrwidth [4]
Card8 slidSHrheight [4]
Card8 slidSeHrwidth [4]
Card8 slidSeHrheight [4]
Card8 slidSeVrHrwidth [4]
Card8 slidSeVrHrheight [4]
Card8 slidLiftNeVrwidth [4]
Card8 slidLiftNeVrheight [4]
Card8 slidLiftEVrwidth [4]
Card8 slidLiftVrheight [4]
Card8 slidLiftSeVrwidth [4]
Card8 slidLiftSeVrheight [4]
Card8 slidLiftSwHrwidth [4]
Card8 slidLiftSwHrheight [4]
Card8 slidLiftSHrwidth [4]
Card8 slidLiftSHrheight [4]
Card8 slidLiftSeHrwidth [4]
Card8 slidLiftSeHrheight [4]
Card8 RelnoNWbuffer [width × height]
Card8 RelnoSWbuffer [width × height]
Card8 RelnoNEbuffer [width × height]
Card8 RelnoSEbuffer [width × height]
Card8 RelnoNbuffer [width × height]
Cardß RelnoWbuffer [width × height]
Card8 RelnoEbuffer [width × height]
Card8 RelnoSbuffer [width × height]
Card8 RelnoCbuffer [width × height]
Card8 RelfoNWbuffer [width × height]
Card8 RelfoSWbuffer [width × height]
Card8 RelfoNEbuffer [width × height]
Card8 RelfoSEbuffer [width × height]
Card8 RelfoNbuffer [width × height]
Card8 RelfoWbuffer [width × height]
Card8 RelfoEbuffer [width × height]
Card8 RelfoSbuffer [width × height]
Card8 RelfoCbuffer [width × height]
Card8 RelhiNWbuffer [width × height]
Card8 RelhiSWbuffer [width × height]
Card8 RelhiNEbuffer [width × height]
Card8 RelhiSEbuffer [width × height]
Card8 RelhiNbuffer [width × height]
Card8 RelhiWbuffer [width × height]
Card8 RelhiEbuffer [width × height]
Card8 RelhiSbuffer [width × height]
Card8 RelhiCbuffer [width × height]
Card8 RelfhNWbuffer [width × height]
Card8 RelfhSWbuffer [width × height]

Card8 RelfhNEbuffer [width × height]
Card8 RelfhSEbuffer [width × height]
Card8 RelfhNbuffer [width × height]
Card8 RelfhWbuffer [width × height]
Card8 RelfhEbuffer[width × height]
Card8 RelfhSbuffer [width × height]
Card8 RelfhCbuffer [width × height]
Card8 ButnoNWbuffer [width × height]
Card8 ButnoSWbuffer [width × height]
Card8 ButnoNEbuffer [width × height]
Card8 ButnoSEbuffer [width × height]
Card8 ButnoNbuffer [width × height]
Card8 ButnoWbuffer [width × height]
Card8 ButnoEbuffer [width × height]
Card8 ButnoSbuffer [width × height]
Card8 ButnoCbuffer [width × height]
Card8 ButfoNWbuffer [width × height]
Card8 ButfoSWbuffer [width × height]
Card8 ButfoNEbuffer [width × height]
Card8 ButfoSEbuffer [width × height]
Card8 ButfoNbuffer [width × height]
Card8 ButfoWbuffer [width × height]
Card8 ButfoEbuffer [width × height]
Card8 ButfoSbuffer [width × height]
Card8 ButfoCbuffer [width × height]
Card8 ButhiNWbuffer [width × height]
Card8 ButhiSWbuffer [width × height]
Card8 ButhiNEbuffer [width × height]
Card8 ButhiSEbuffer [width × height]
Card8 ButhiNbuffer [width × height]
Card8 ButhiWbuffer [width × height]
Card8 ButhihiEbuffer [width × height]
Card8 ButhiSbuffer [width × height]
Card8 ButhiCbuffer [width × height]
Card8 ButfhNWbuffer [width × height]
Card8 ButfhSWbuffer [width × height]
Card8 ButfhNEbuffer [width × height]
Card8 ButfhSEbuffer [width × height]
Card8 ButfhNbuffer [width × height]
Card8 ButfhWbuffer [width × height]
Card8 ButfhEbuffer [width × height]
Card8 ButfhSbuffer [width × height]
Card8 ButfhCbuffer [width × height]
Card8 Choicebuffer [width × height]
Card8 chcklnobuffer [width × height]
Card8 chcklsebuffer [width × height]
Card8 chcklfobuffer [width × height]
Card8 chcklfsbuffer [width × height]
Card8 chck2nobuffer [width × height]
Card8 chck2sebuffer [width × height]
Card8 chck2fobuffer [width × height]

Card8 chck2fsbuffer [width × height]
Card8 slideNeVrbuffer [width × height]
Card8 slideEVrbuffer [width × height]
Card8 slideSeVrbuffer [width × height]
Card8 slideSwHrbuffer [width × height]
Card8 slideSHrbuffer [width × height]
Card8 slideSeHrbuffer [width × height]
Card8 slideSeVrbuffer [width × height]
Card8 slideNeVrbuffer [width × height]
Card8 slideLiftNeVrbuffer [width × height]
Card8 slideLiftEVrbuffer [width × height]
Card8 slideLiftSeVrbuffer [width × height]
Card8 slideLiftSwHrbuffer [width × height]
Card8 slideLiftSHrbuffer [width × height]
Card8 slideLiftSeHrbuffer [width × height]

Zum Beispiel

	Breite	**Höhe**		**Breite**	**Höhe**		**Breite**	**Höhe**
0000000		8		8	8		8	
0000020		8		8	8		8	
0000040		8		8	8		8	
0000060		8		8	8		8	
0000100		8		8	8		8	
0000120		8		8	8		8	
0000140		8		8	8		8	
0000160		8		8	8		1	
0000200		8		8	1		8	
0000220		8		8	8		8	
0000240		8		8	8		8	
0000260		8		8	8		8	
0000300		8		8	8		8	
0000320		1		1	8		8	
0000340		8		8	8		8	
0000360		8		8	8		8	
0000400		8		8	8		8	
0000420		8		8	1		1	
0000440		8		8	8		8	
0000460		8		8	8		8	
0000500		8		8	8		8	
0000520		8		8	8		8	
0000540		1		1	8		8	
0000560		8		8	8		8	
0000600		8		8	8		8	
0000620		8		8	8		8	
0000640		8		8	1		1	
0000660		8		8	8		8	
0000700		8		8	8		8	
0000720		8		8	8		8	
0000740		8		8	8		8	
0000760		1		1	8		8	
0001000		8		8	8		8	
0001020		8		8	8		8	
0001040		8		8	8		8	
0001060		8		8	1		1	
0001100	1	6	1	6	1	6	1	6
0001120	1	6	1	6	1	6	1	6
0001140	1	6	1	6	1	6	1	6
0001160	1	6	1	6	1	6	1	6
0001200	1	6	1	6		8		8
0001220		8		8		8		8
0001240		8		8		8		8
0001260		8		8		1		1
0001300		2		2		2		2
0001320		2		2		2		2
0001340		2		2		2		2

Start des Puffers 1 data. (relnNW) (8 × 8 Bytes)

```
0001360  \0   \0   \0   \0   \0   266  004  004  \0   \0   \0   266  \n   \r   \r   \r
0001400  \0   \0   \n   \r   017  \r   \v   001  \0   266  \r   017  \r   001  001  001
0001420  \0   \n   017  \r   001  001  001  001  265  \r   \r   001  001  001  001  001
0001440  004  \r   \v   001  001  001  001  001  004  \v   001  001  001  001  001  001
```

Start des Puffers 2 data. (relnSW) (8 × 8 Bytes)

```
0001460  004  004  263  \0   \0   \0   \0   \0   \r   \r   \v   004  \a   \0   \0   \0
0001500  001  001  001  270  004  262  \0   \0   001  001  001  001  \n   004  004  \0
0001520  001  001  001  001  001  \n   \a   \0   001  001  001  001  001  001  270  265
0001540  001  001  001  001  001  270  \t   \a   006  001  001  001  001  001  270  006
```

Start des Puffers 3 data. (relnNE) (8 × 8 Bytes)

```
0001560  004  \v   001  001  001  001  001  006  \b   001  001  001  001  001  001  001
0001600  006  \n   001  001  001  001  001  001  \0   265  \n   001  001  001  001  001
0001620  \0   006  \t   \rL  001  001  001  001  \0   \0   261  \t   \n   \n   \n   \n
0001640  \0   \0   \0   006  022  265  265  \t   \0   \0   \0   \0   \0   \a   006  006
```

Start des Puffers 4 data. (relnSE) (8 × 8 Bytes)

```
0001660  \r   001  001  001  001  270  \t   006  001  001  001  001  001  270  \t   006
0001700  001  001  001  001  001  \n   265  261  001  001  001  001  \n   266  261  \0
0001720  001  001  001  \n   266  \b   261  \0   270  270  004  266  \b   261  \0   \0
0001740  \t   \t   \b   261  261  \0   \0   \0   006  006  261  \0   \0   \0   \0   \0
```

reDrawAll

public void reDrawAll()

[0416] Findet das Trickfenster mit dem Fokus, dann die Vorgänger, bis keine mehr vorliegen. Oberstes Fenster wird dann auf unsichtbar gesetzt und wieder auf sichtbar. Dann sollte das ganze Fenster neu gezeichnet werden.

[0417] Die verschiedenen, oben beschriebenen, Verfahren zur Darstellung eines oder mehrerer graphischer Objekte und zur Navigation zwischen einer Vielzahl derartiger Objekte, oder zum Empfangen von Eingaben durch den Benutzer, können auch vorzugsweise, jedoch nicht ausschliesslich, auf anderen Gebieten im Rahmen des Empfangs von Rundfunksendungen durch Rundfunksenderbetreiber eingesetzt werden. Jede Funktionalität eines Beistelldekoders, der über visuelle Interaktivität mit dem Benutzer arbeitet, kann im Allgemeinen solche Verfahren einsetzen.

[0418] Eine Kette mit Symbolen, möglicherweise mit Unterketten, die hinzugefügt wurden und über die man navigieren kann, können bei einer Anwendung für den Einkauf über Fernsehangebote (home shopping) zum Einsatz kommen, um dem Benutzer die Anzeige von Artikeln zu erlauben, Preise anzusehen, Aufträge abgeben oder auf andere Weise mit der Anwendung interaktiv zu arbeiten. Das graphische Objekt, auf dem man einen Auftrag erteilen kann, könnte, wenn es hervorgehoben wird, in der oben beschriebenen Weise automatisch zwischen den Kaufsymbolen (zum Beispiel ein Dollarzeichen, $) und Text hin- und herspringen und den bis dahin ausgegebenen Betrag, oder das Wort "Kaufen" in der Sprache des Benutzers anzeigen. Alternativ oder zusätzlich könnte ein graphisches Objekt, das das Wort "Kaufen" in der Sprache des Abonnenten beinhaltet, erscheinen, immer dann wenn das "Kaufen" Symbol gewählt wird.

[0419] Das "Kaufen" Symbol im obigen Beispiel kann neben ein Symbol gesetzt werden, das beim Anklicken die Liste der bislang getätigten Einkäufe aufzeigt und ein anderes Symbol, das beim Anklicken Lieferoptionen für das gerade gekaufte Produkt anzeigt, um so eine logische Abfolge der Symbole in der Kette, in der der Benutzer navigieren kann, anzubieten. Wenn das "Kaufen" Symbol ausgewählt wurde, kann eine Unterkette mit verschiedenen Unteroptionen erscheinen, die verschiedene Kreditpläne für teurere Produkte einschliesst. Irgendwelche Information die vom Benutzer verlangt werden, wie zum Beispiel Strasse, Lieferadresse, können mit der virtuellen Tastatur eingegeben werden.

[0420] Bei einem elektronischen Programmführer können ähnliche Verfahren benutzt werden, zum interaktiven Browsern und zur Darstellung verschiedener Kanäle, Themen und Zeit und Datum. Eine weitere kundenspezifische Anpassung kann für die Umgruppierung der graphischen Optionen in der Kette nach Vorliebe des Kunden denkbar sein; im Fall einer Kette von Kanälen können die vom Benutzer bevorzugten Kanäle an die Spitze der Kette programmiert werden. Eine solche Präferenz kann vom Benutzer angegeben werden oder vom Programm abgeleitet werden.

[0421] Weitere Anwendungen für die oben beschriebene Verfahren schliessen Online-Kataloge ein, Nachrichten und Wetterinformationen auf Anfrage, Spiele und die allgemeine Verwaltung des Beistelldekoders (Verwalten der Konfiguration, usw.) Im Fall von Spielen kann der Kopf/Schwanz Flip-Flop Effekt benutzt werden, um in-game Animationen bereit zu stellen, ohne notwendigerweise zusätzliche zu schreibende Verfahren erforderlich zu machen und die virtuelle Tastatur kann als alternative Form zur Steuerung fortgeschrittener Spielarten benutzt werden.

[0422] Es sollte auch gewürdigt werden, dass alle Verfahren zur Interaktion über die Fernbedienung, wie hier beschrieben wurde, ersetzt, oder ergänzt werden können durch eine Maus, (oder andere Richtungssteuerungen, wie zum Beispiel ein Rollerball oder Steuerknüppel (Joystick)) und oder durch eine Tastatur (oder durch andere Vorrichtungen, die eine Vielzahl von Tasten besitzen), entweder durch Simulation der Tasten einer Fernbedienung (zum Beispiel durch Einsatz der Nummerntasten 0–9; der Pfeiltasten und der Rücklauftaste auf einer Tastatur) oder direkt (zum Beispiel unter Einsatz der Maus, um auf die Schaltflächen zu klicken und der Tastatur, um direkt Text einzugeben, anstatt die virtuelle Tastatur zu benutzen).

[0423] Die oben beschriebene virtuelle Tastatur kann bei jedem Gerät eingesetzt werden, das eine Vielzahl von Tasten besitzt, wie zum Beispiel Spielgeräte oder Mobiltelefone, zum Beispiel. Im letzteren Fall kann die virtuelle Tastatur im Wesentlichen, wie beschrieben wurde, auf dem Bildschirm des Telefons (bei Telefonen mit einer genügend grossen Bildfläche) abgebildet werden, oder in einer komprimierten Form (für Telefone mit kleineren Anzeigeflächen). Eine derartige Kompression der virtuellen Tastatur könnte dazu führen, dass nur der Symbolnummernblock zu einer bestimmten Zeit gezeigt wird, vorzugsweise mit einer Empfehlung der Symbole oder Art der Symbole auf die durch Bedienen der links, rechts, auf- und ab-Tasten zugegriffen werden kann (oder deren Entsprechungen, zum Beispiel im Fall von Richtungssteuerungen der rollerartigen Ausführung). Die komprimierte virtuelle Tastatur kann bei anderen Anwendungen eingesetzt werden, insbesondere dort, wo es wenig Platz zur Anzeige der Tastatur gibt.

[0424] Der Begriff "Ankreuzfeld" (check box) kann sich auf ein graphisches Objekt jeglicher Form beziehen, zum Beispiel rund, die in der Lage ist verschiedene Zustände anzuzeigen, vorzugsweise zwei Zustände, die "kontrolliert" und "nicht kontrolliert" entsprechen, aber möglicherweise mehr als zwei Zustände und die ihren Zustand in durchgängiger Art und Weise verändern, wenn sie vom Benutzer angeklickt, oder angewählt werden. Der "kontrollierte" Zustand kann durch ein Häkchen, Kreuz oder eine andere Ausschmückung in dem Kästchen angezeigt werden.

[0425] Zur Erleichterung der Bezugnahme haben die unten benutzten Begriffe die folgenden bevorzugten Bedeutungen:
HTML: HyperTextMarkup Language, eine Sprache die Dokumente beschreibt, die im Internet ausgetauscht werden. Das Dokument kann Bezüge zu Seiten, Formatierungsinformationen, Ton und Bilder, usw. einschliessen.
HTTP: HyperText Transport Protocol, ein Protokoll zur Kommunikation zwischen Internetservern, die HTML Dokumente und Navigationsanwendungen enthalten, die das HTML Dokument abbilden.
MPEG-2: Motion Picture Expert Group, ein Kodierverfahren bewegter Bilder und Ton in Echtzeit.
PPP: Point-to-Point Protocol, ein Verbindungsprotokoll für Fernzugriff, das zwei Computern erlaubt über ein Modem vernetzt zu werden.
PROXY SERVER: Eine auf dem Server befindliche Anwendung, die es erlaubt, sichere Internetverbindungen aufzubauen und die auch HTTP und FTP Anfragen puffert.
SESSION: Eine Instanz einer Art Verbindung oder einer Anwendung im Speicher zu einer gegebenen Zeit.
URL: Uniform Resource Locator, eine Adresse, die zur Lokalisierung einer Datei oder von Ressourcen im Internet dient. Die Verbindung zu einer Seite bezeichnet die Adresse der Ressource, die in der Webseite enthalten ist.
WWW: World Wide Web, Internet Netzwerk, das lokale oder entfernte Dokumente benutzt. Ein Web Dokument ist eine Web-Seite und die Verbindungen in der Seite erlauben die Navigation zwischen verschiedenen Seiten und zwischen verschiedenen Themenkreisen, unberücksichtigt ob diese in einem lokalen oder entfernten Netzwerk angesiedelt sind.

GUI Graphical User Interface. Graphische Benutzerschnittstelle.
WGT: Widget Toolkit. "Werkzeugkasten" für Trickfenster.

[0426] Es ist wohlverstanden, dass die vorliegende Erfindung oben einzig und allein anhand von Beispielen beschrieben wurde und dass Veränderungen von Einzelheiten innerhalb des Bereichs der Erfindung erfolgen können.

[0427] Jede offen gelegte Eigenschaft in der Beschreibung und (wo angezeigt) der Ansprüche und Zeichnungen, können unabhängig oder in jeder passenden Kombination vorgesehen werden.

[0428] In jeder oder allen zuvor erwähnten Eigenschaften, wurden bestimmte Eigenschaften der vorliegenden Erfindung durch Einsatz von Rechnerprogrammen implementiert. Es wird jedoch gewiss dem Fachmann deutlich werden, dass jede dieser Eigenschaften durch Einsatz von Hardware, oder einer Kombination von Hardware und Software, implementiert werden können. Ferner ist leicht zu verstehen, dass die Funktionen, die von der Hardware, der Computersoftware und dergleichen vollzogen werden, mit elektrischen und ähnlichen Signalen durchgeführt werden.

[0429] Eigenschaften, die sich auf die Speicherung von Informationen beziehen, können auf geeigneten Speicherorten oder Speicherplätzen implementiert werden. Eigenschaften, die sich auf die Datenverarbeitung beziehen, können durch passende Prozessoren oder Steuerungsmittel implementiert werden, entweder in Software oder in Hardware oder in einer Kombination von beiden.

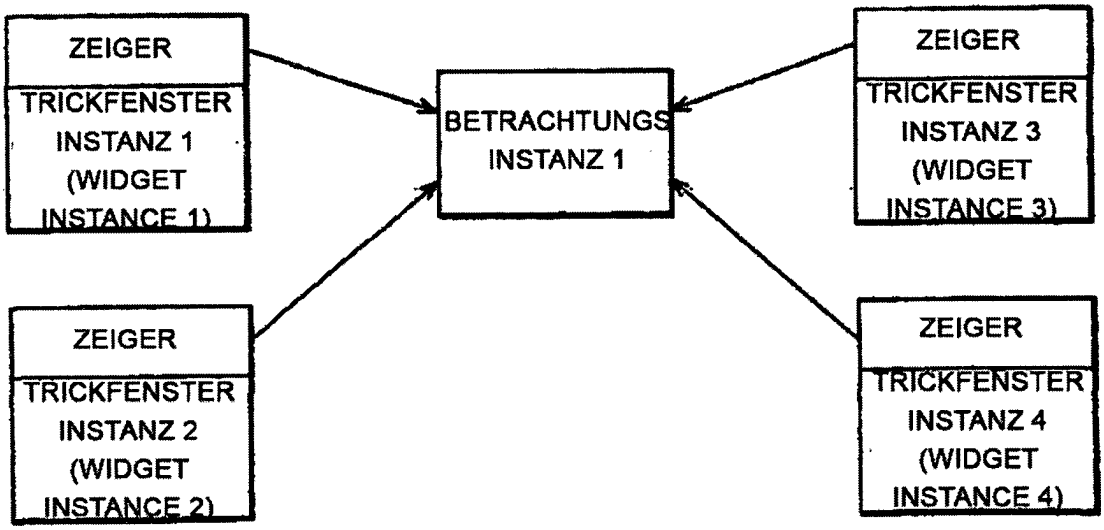
[0430] In irgendeiner oder allen der zuvor erwähnten Formen, wobei die Erfindung in jeder Form ausgeführt werden kann, können einige oder alle der folgend Formen ausgeführt werden: in einem Verfahren zum Betrieb eines Rechnersystems; im Rechner selbst; in einem Rechnersystem, wenn programmiert mit, oder angepasst oder eingerichtet ist, das Verfahren zum Betrieb dieses System auszuführen; und/oder in einem durch einen Rechner lesbaren Speichermittel, das ein darauf aufgenommenes Programm besitzt, das geeignet ist, nach dem Verfahren zum Betrieb des System zu arbeiten.

[0431] Wie hier durchgehend als Begriff verwendet wurde, kann "Computer System" durch "Computer", "Rechner", "System", "Ausrüstung", "Gerät", "Maschine" oder ähnliche Begriffe ersetzt werden.

[0432] Referenznummern, die in den Ansprüchen auftauchen, sind nur zur Illustration und sollten keine einschränkende Auswirkung auf den Umfang der Ansprüche darstellen.

[0433] Die Antragsteller erklären hiermit, zur Abwendung von Einwänden, dass sie die Urheberechte in den beigefügten Zeichnungen beanspruchen.

Titel: Darstellung graphischer Objekte



[0434] Kurzdarstellung: Schwerpunkte der vorliegenden Erfindung betreffen ein Verfahren zur Steuerung des Erscheinungsbildes eines graphischen Objekts in einer graphischen Benutzerschnittstelle. In einem Ausführungsbeispiel der Erfindung schliesst ein Objekt, wie zum Beispiel ein Trickfenster (widget) in einer graphi-

schen Benutzerschnittstelle, eine Instanz einer Trickfensterklasse (widget class) ein, in der Eigenschaften und/oder Verfahren bestimmt sind, die die Bedienung des Objekts steuern und eine damit verknüpfte Instanz eine Betrachtungsobjektsklasse, in der die Eigenschaften und/oder Verfahren bestimmt sind, die das Erscheinungsbild des Objekts steuern.

Patentansprüche

1. Ein Verfahren zur Steuerung des Aussehens eines objektorientierten Trickfensters (widget) (**430**; **500**) bei einer graphischen Benutzerschnittstelle (graphical user interface GUI), die beinhaltet:
Bestimmung eines Betrachtungsgegenstands; und
Verknüpfung des Betrachtungsgegenstands mit dem Trickfenster (**430**; **500**);
wobei der Betrachtungsgegenstand einen Kode oder Kodeparameter enthält, die bestimmen, wie das Trickfenster (**430**; **500**) dargestellt wird und
wobei der Betrachtungsgegenstand die Erscheinung des Trickfensters (**430**; **500**) bestimmt,
dadurch gekennzeichnet, dass
der Betrachtungsgegenstand ein Aktualisierungszählwerk beinhaltet, dessen Wert aktualisiert wird, wenn der Betrachtungsgegenstand neu definiert oder verändert wird.

2. Verfahren nach Anspruch 1, wobei der Betrachtungsgegenstand durch einen objektbezogenen Programmkode bestimmt wird.

3. Verfahren nach Anspruch 1, wobei das Trickfenster ein Attribut zur Kennzeichnung des Betrachtungsgegenstands beinhaltet, der mit dem Trickfenster (**430**; **500**) verknüpft ist.

4. Verfahren nach Anspruch 1, das ferner die Veränderung der Erscheinung des Trickfensters (**430**; **500**) durch Neudefinition oder Veränderung des Betrachtungsgegenstands, oder durch Verknüpfung eines anderen Betrachtungsgegenstands mit einem Trickfenster (**430**; **500**) beinhaltet.

5. Verfahren nach Anspruch 1, das ferner die Steuerung des Aussehens einer Vielzahl von Trickfenstern (**430**; **500**) in einer graphischen Benutzerschnittstelle beinhaltet, durch Verknüpfung des Betrachtungsgegenstands mit der Vielzahl von Trickfenstern (**430**; **500**).

6. Vorrichtung (**13**) zur Steuerung des Aussehens eines objektorientierten Trickfensters (**430**; **500**) in einer graphischen Benutzerschnittstelle, die beinhaltet:
Mittel (**220**) zur Bestimmung eines Betrachtungsgegenstands; und
Mittel (**220**) zur Verknüpfung des Betrachtungsgegenstands mit dem Trickfenster (**430**; **500**);
wobei der Betrachtungsgegenstand einen Kode oder Kodeparameter beinhaltet, die bestimmen, wie das Trickfenster (**430**; **500**) dargestellt wird und
wobei der Betrachtungsgegenstand das Aussehen des Trickfensters (**430**; **500**) bestimmt
dadurch gekennzeichnet, dass
der Betrachtungsgegenstand ein Aktualisierungszählwerk beinhaltet, dessen Wert aktualisiert wird, wenn der Betrachtungsgegenstand neu definiert oder verändert wird.

Es folgen 41 Blatt Zeichnungen

Anhängende Zeichnungen

FIG. 1a

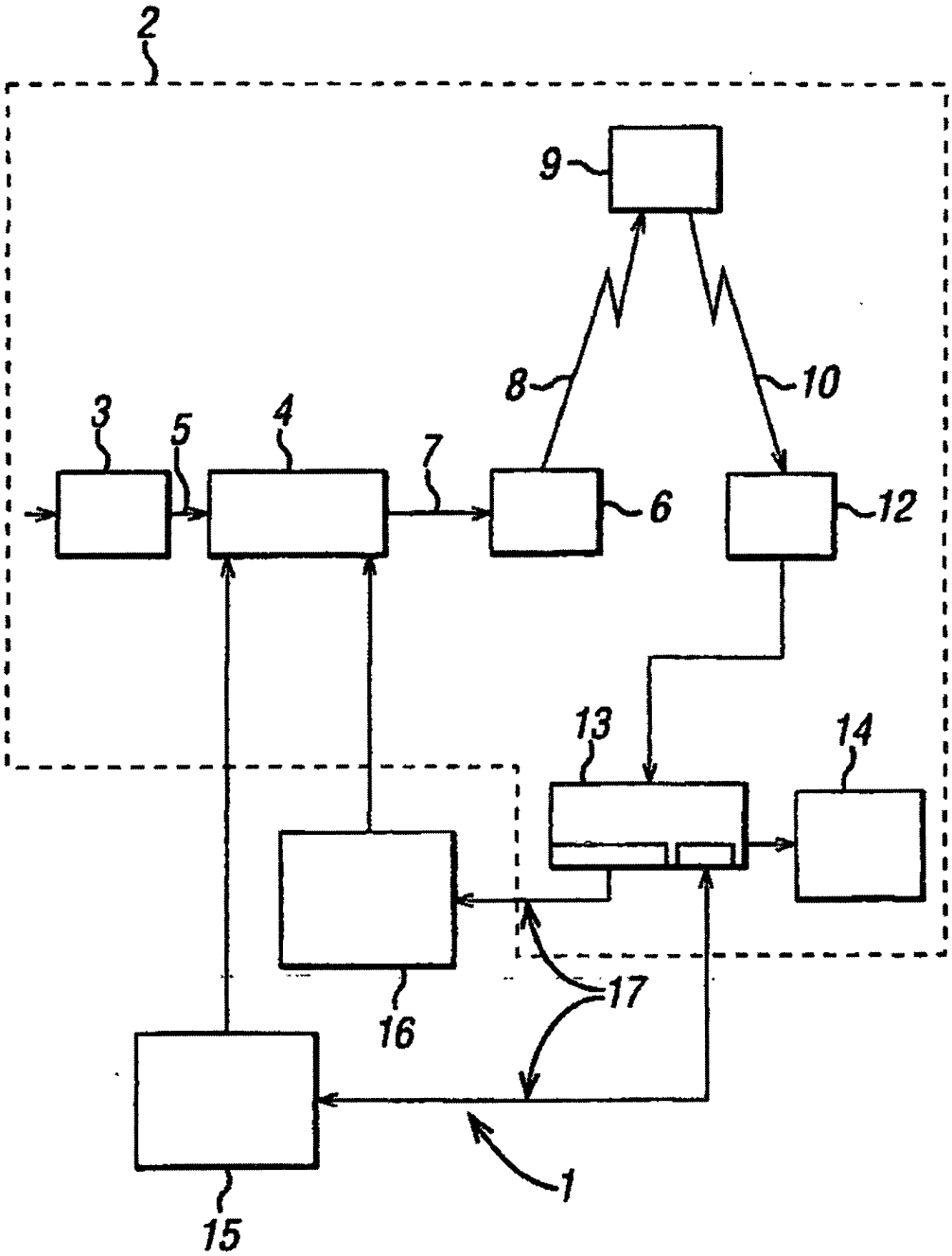
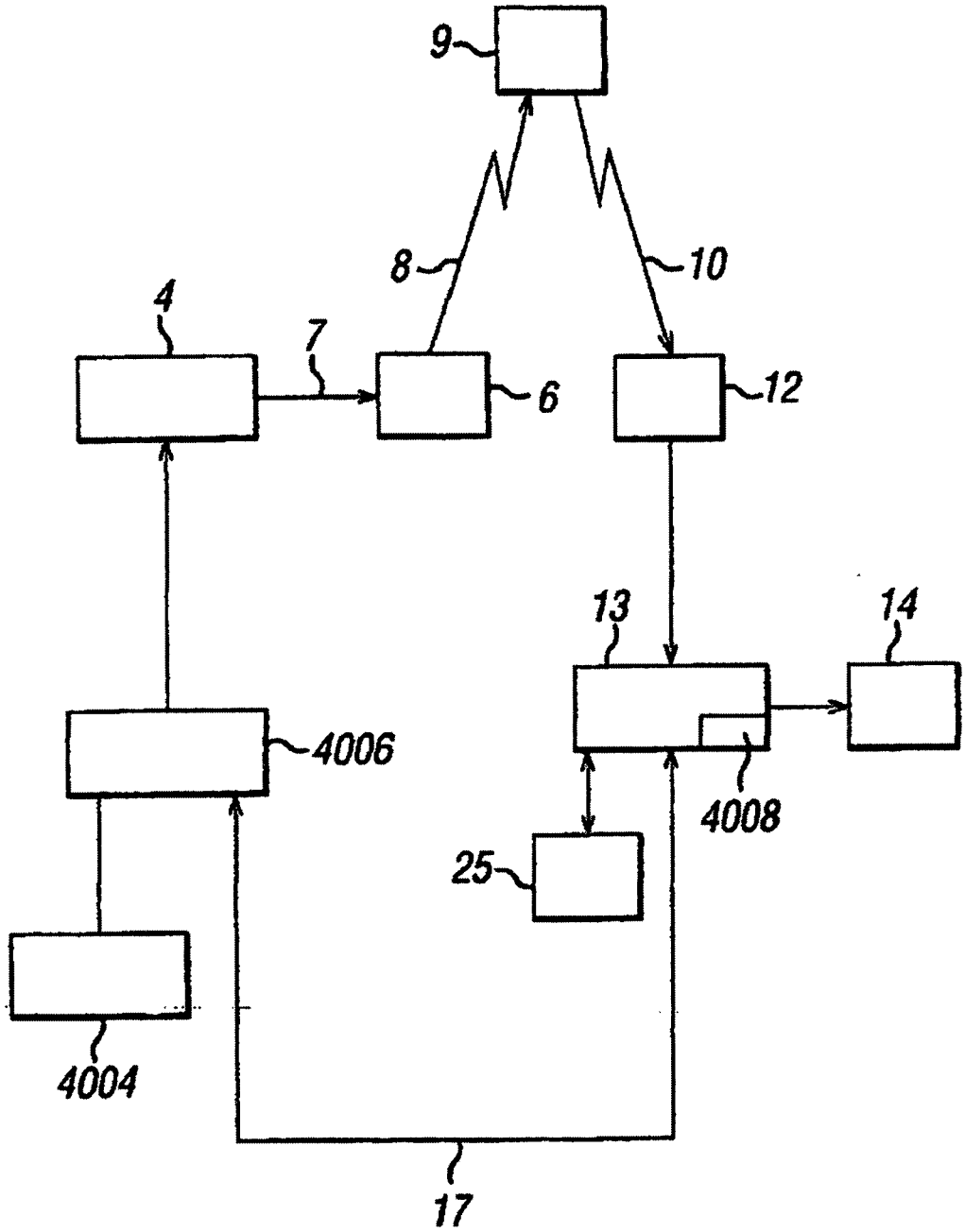


FIG. 1b



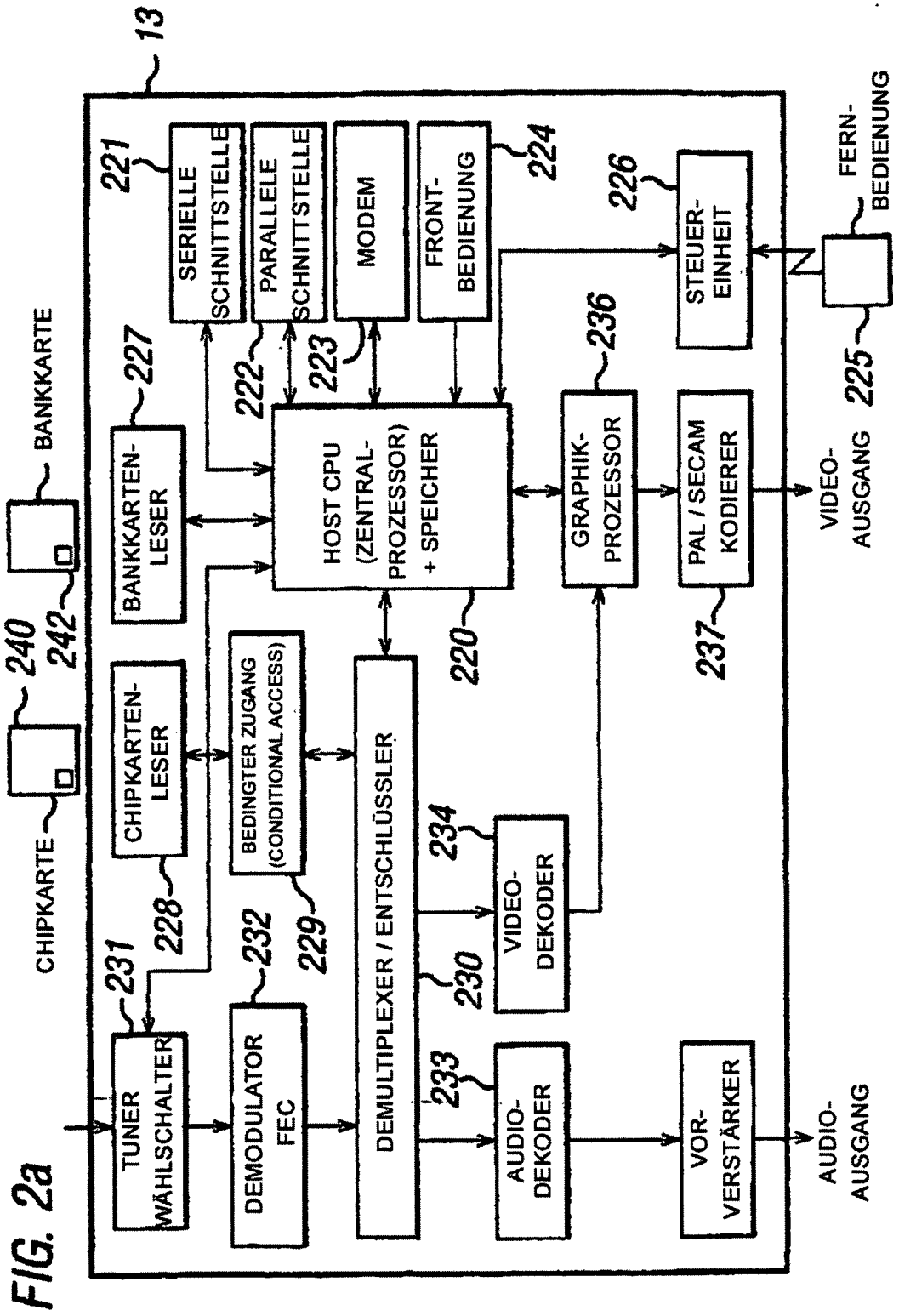

13
BANKKARTE
240
242
CHIPKARTE
CHIPKARTEN-LESER
228
BANKKARTEN-LESER
227
SERIELLE SCHNITTSTELLE
221
PARALLELE SCHNITTSTELLE
222
MODEM
223
FRONT-BEDIENUNG
224
STEUER-EINHEIT
226
FERN-BEDIENUNG
225
HOST CPU (ZENTRAL-PROZESSOR) + SPEICHER
220
GRAPHIK-PROZESSOR
236
PAL / SECAM KODIERER
237
VIDEO-AUSGANG
BEDINGTER ZUGANG (CONDITIONAL ACCESS)
229
DEMULTIPLEXER / ENTSCHLÜSSLER
230
VIDEO-DEKODER
234
TUNER WÄHLSCHALTER
231
DEMODULATOR FEC
232
AUDIO-DEKODER
233
VOR-VERSTÄRKER
AUDIO-AUSGANG

FIG. 2a

FIG. 2b

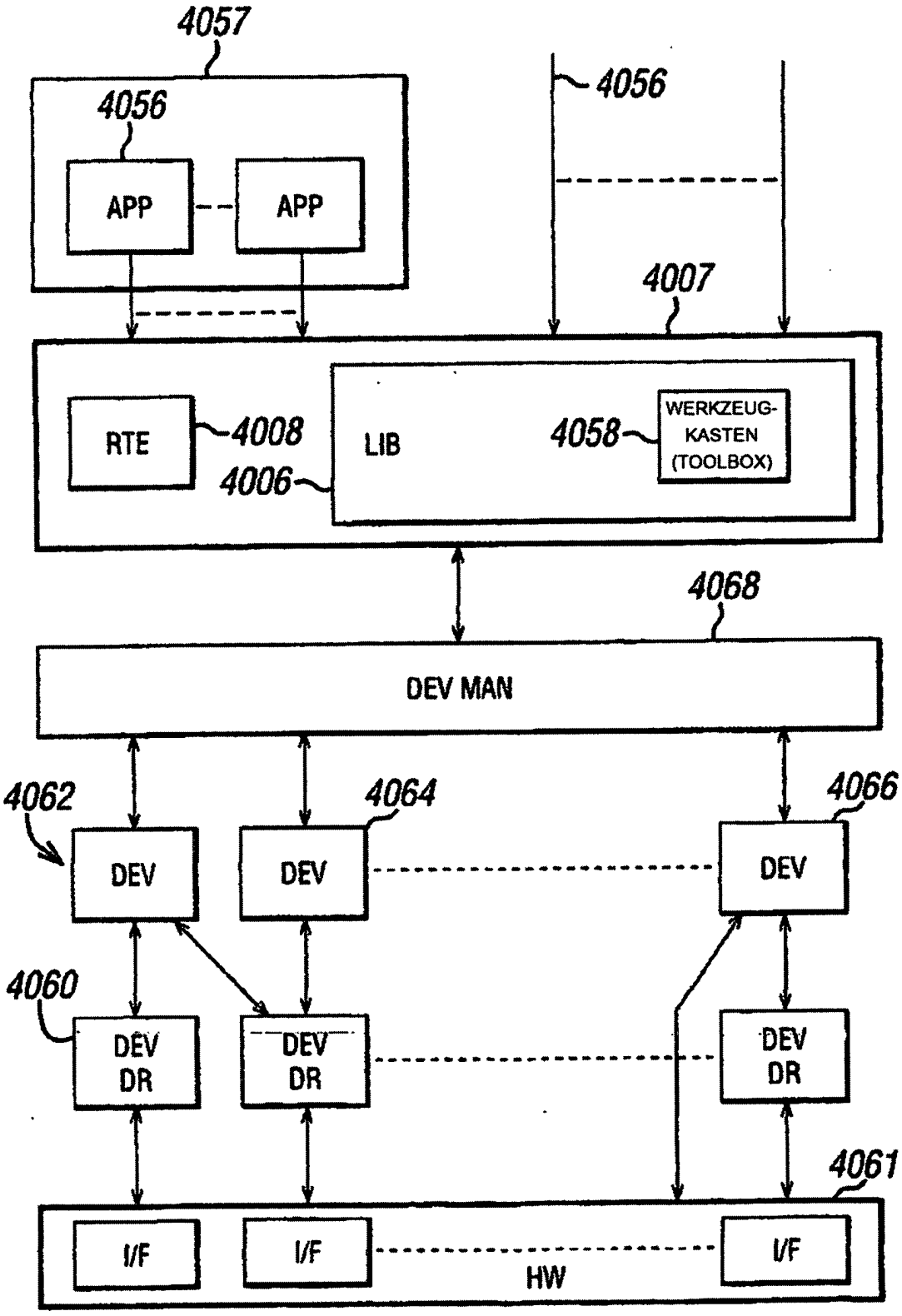

4057
4056
APP
APP
4056
4007
RTE
4008
4006
LIB
4058
WERKZEUG-
KASTEN
(TOOLBOX)
4068
DEV MAN
4062
4064
4066
DEV
DEV
DEV
4060
DEV
DR
DEV
DR
DEV
DR
4061
I/F
I/F
I/F
HW

FIG. 2c

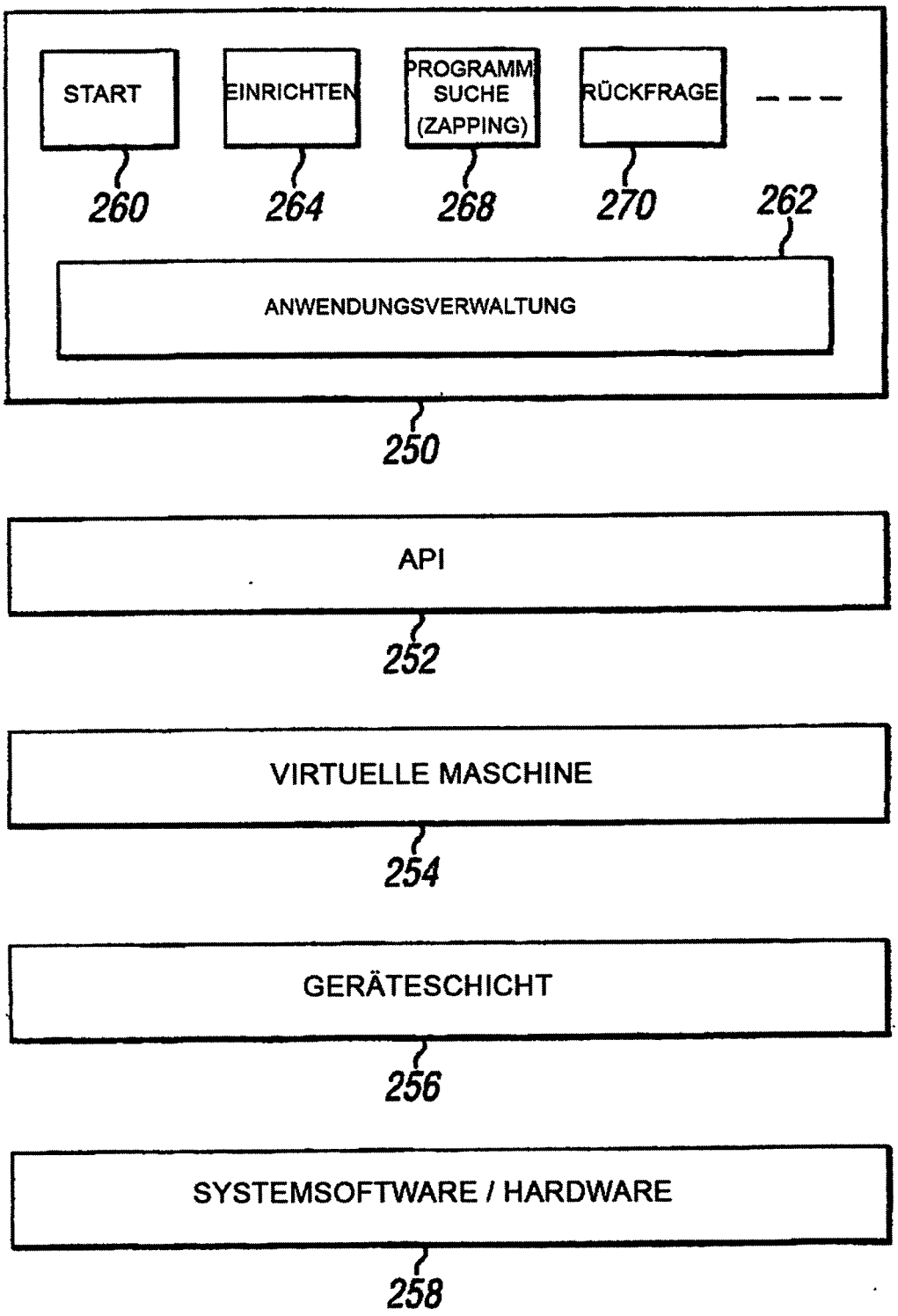

START
EINRICHTEN
PROGRAMM SUCHE (ZAPPING)
RÜCKFRAGE
260
264
268
270
262
ANWENDUNGSVERWALTUNG
250
API
252
VIRTUELLE MASCHINE
254
GERÄTESCHICHT
256
SYSTEMSOFTWARE / HARDWARE
258

FIG. 3

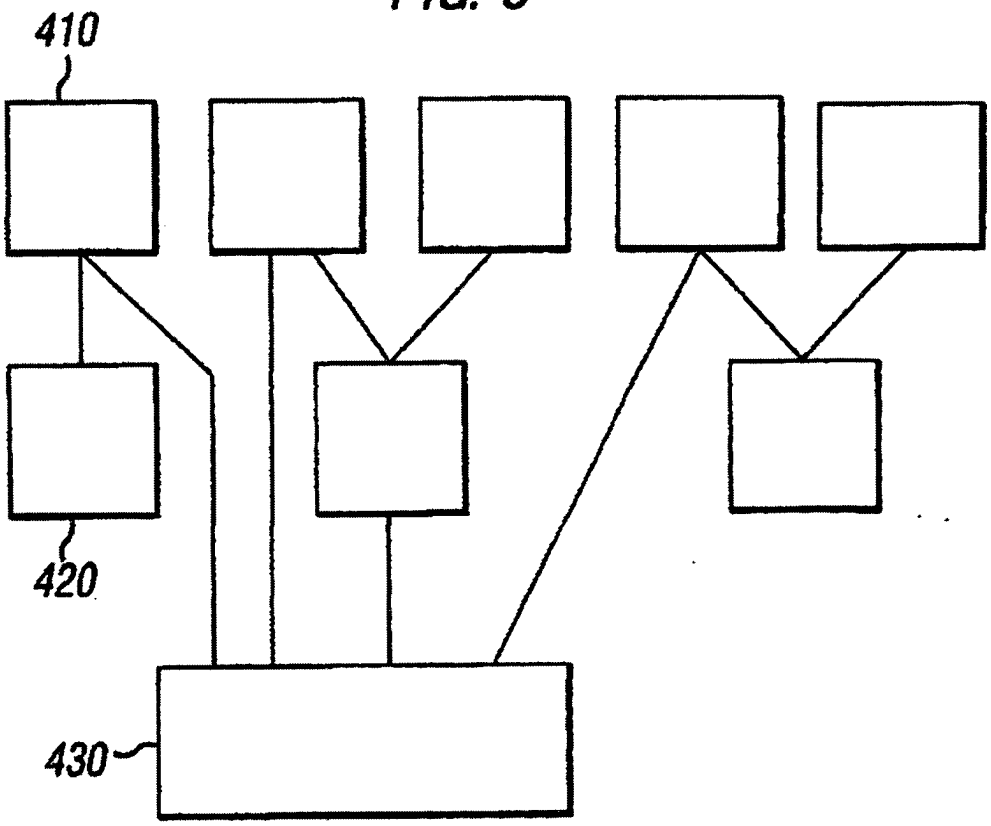


FIG. 6b

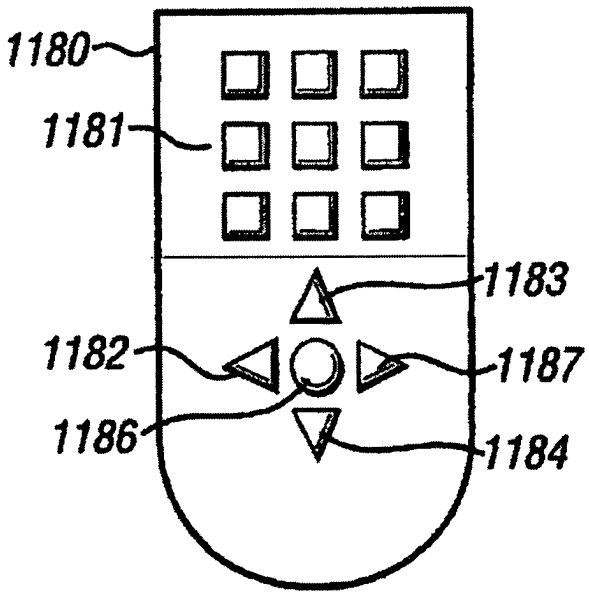


FIG. 4

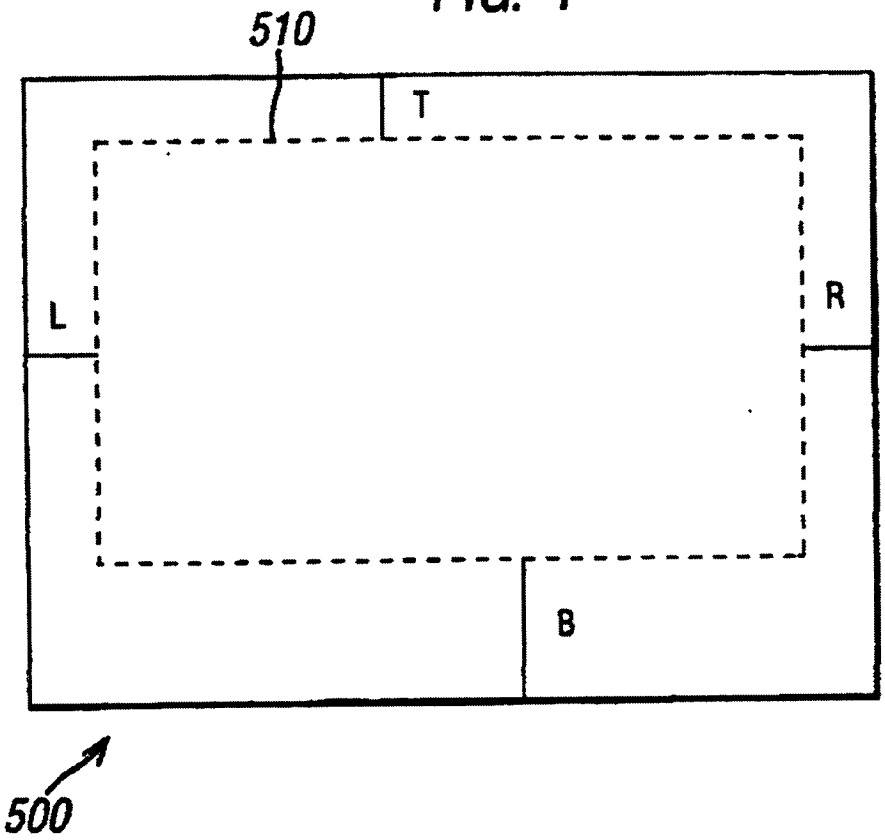

510
T
L
R
B
500

FIG. 5

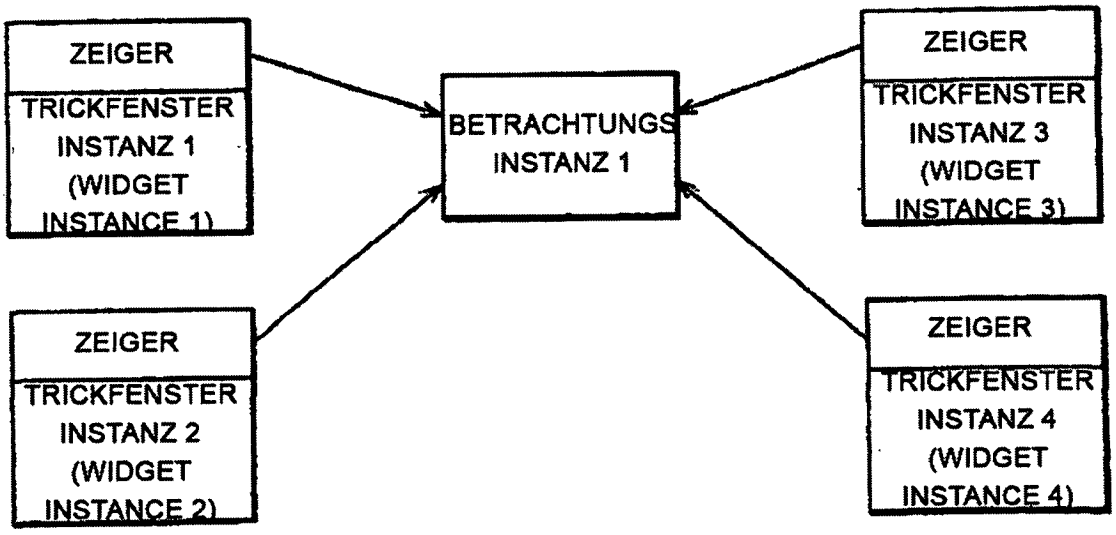
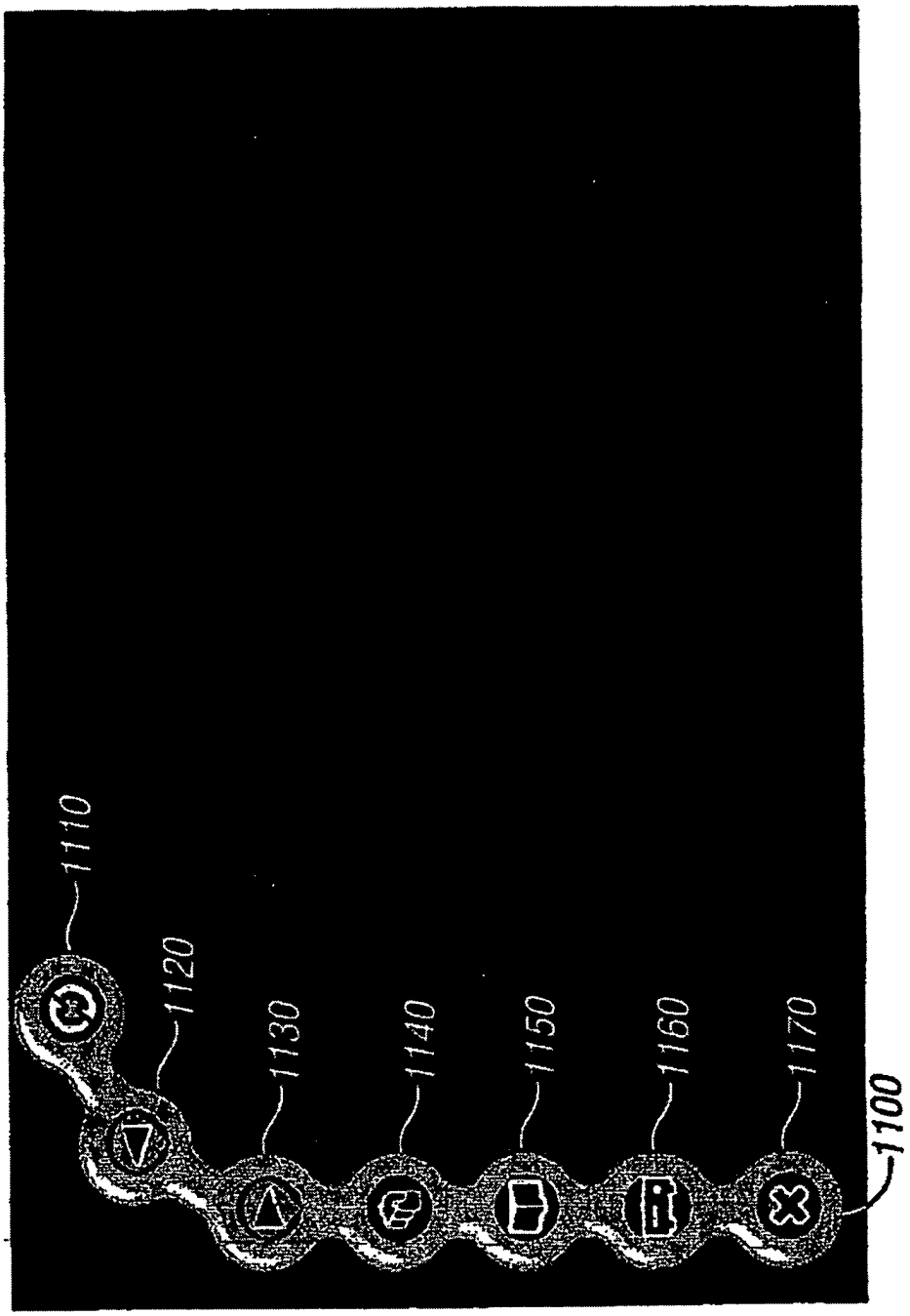

ZEIGER
TRICKFENSTER INSTANZ 1 (WIDGET INSTANCE 1)
ZEIGER
TRICKFENSTER INSTANZ 2 (WIDGET INSTANCE 2)
BETRACHTUNGS INSTANZ 1
ZEIGER
TRICKFENSTER INSTANZ 3 (WIDGET INSTANCE 3)
ZEIGER
TRICKFENSTER INSTANZ 4 (WIDGET INSTANCE 4)

FIG. 6a



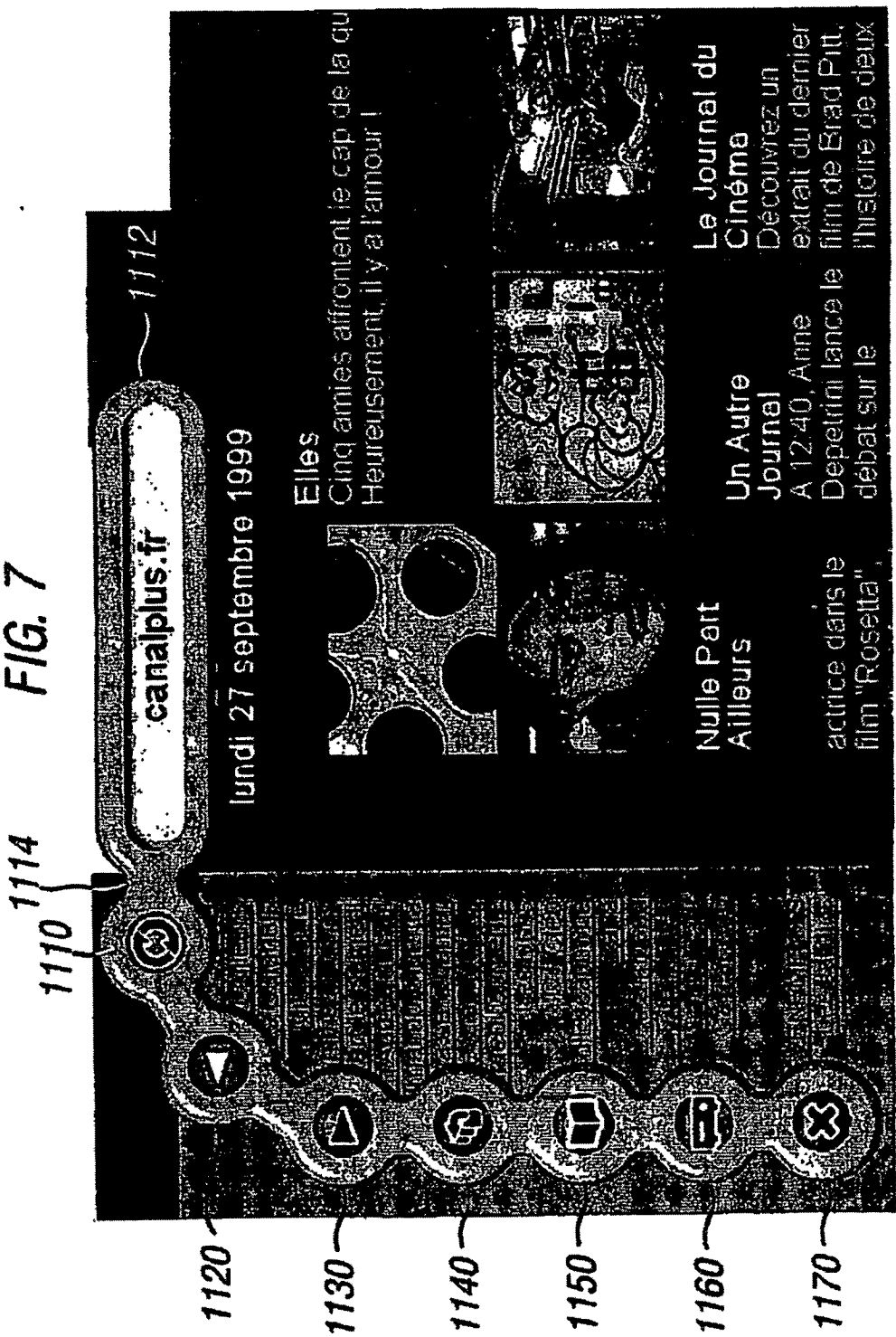

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Un Autre Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
Nulle Part Ailleurs
actrice dans le
film "Rosetta",
1110
1112
1114
1120
1130
1140
1150
1160
1170
1100

FIG. 7

FIG. 8

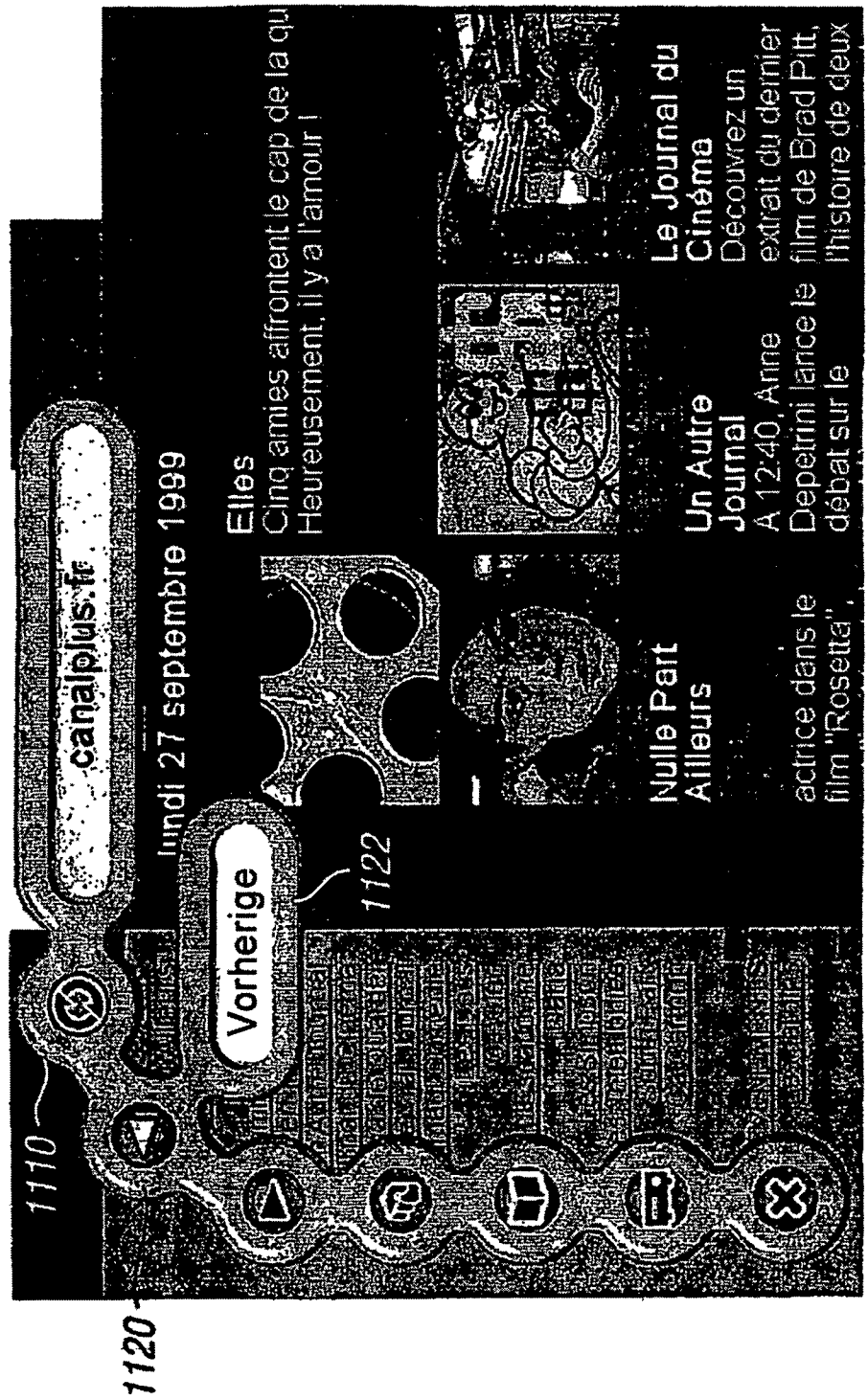

1110
canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Vorherige
1122
Nulle Part Ailleurs
Un Autre Journal
A 12:40, Anne
Depetrini lance le
débat sur le
actrice dans le
film "Rosetta",
Le Journal du Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
1120

FIG. 9

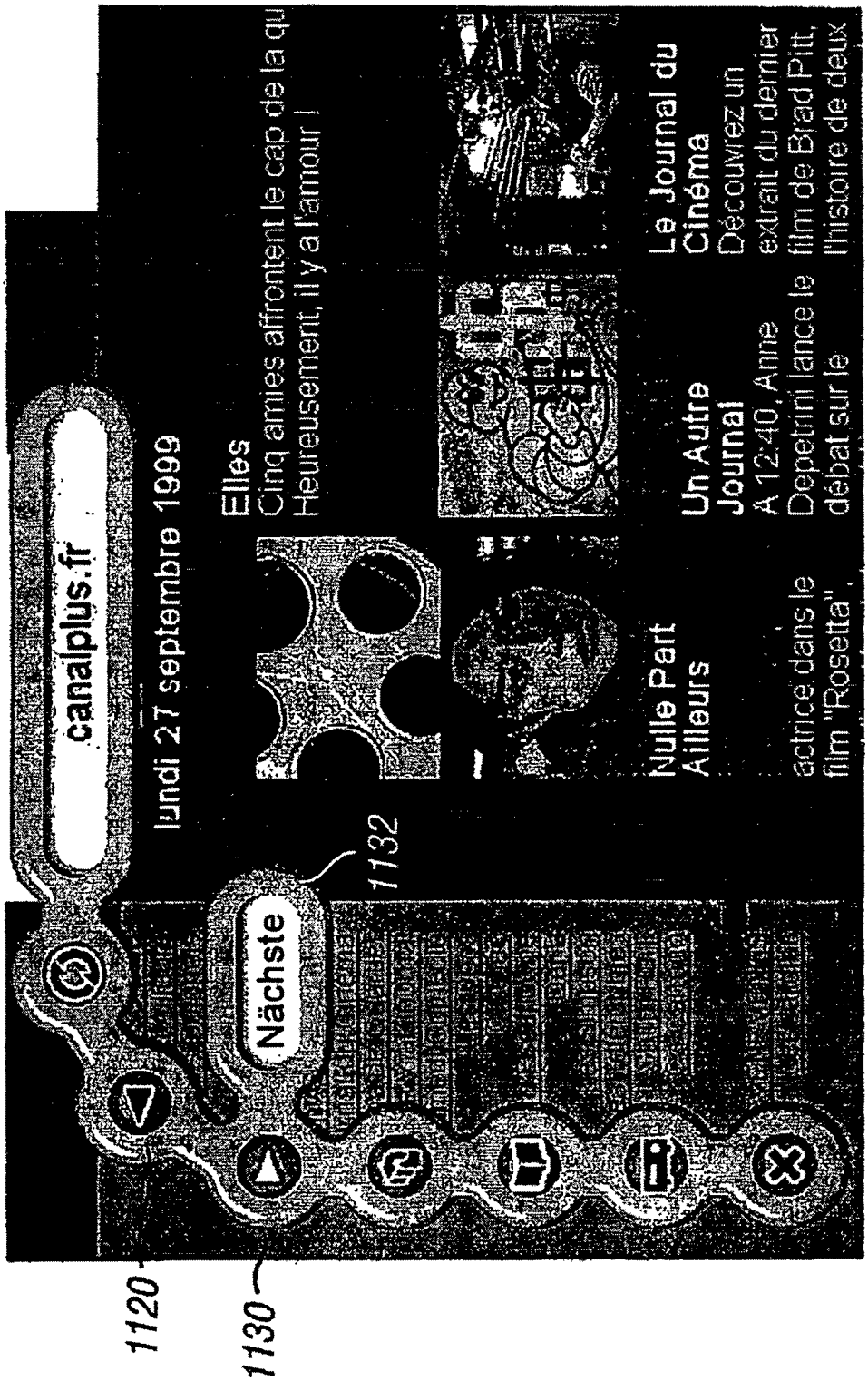

canalplus.fr
lundi 27 septembre 1999
Nächste
1132
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
1120
1130

FIG. 10

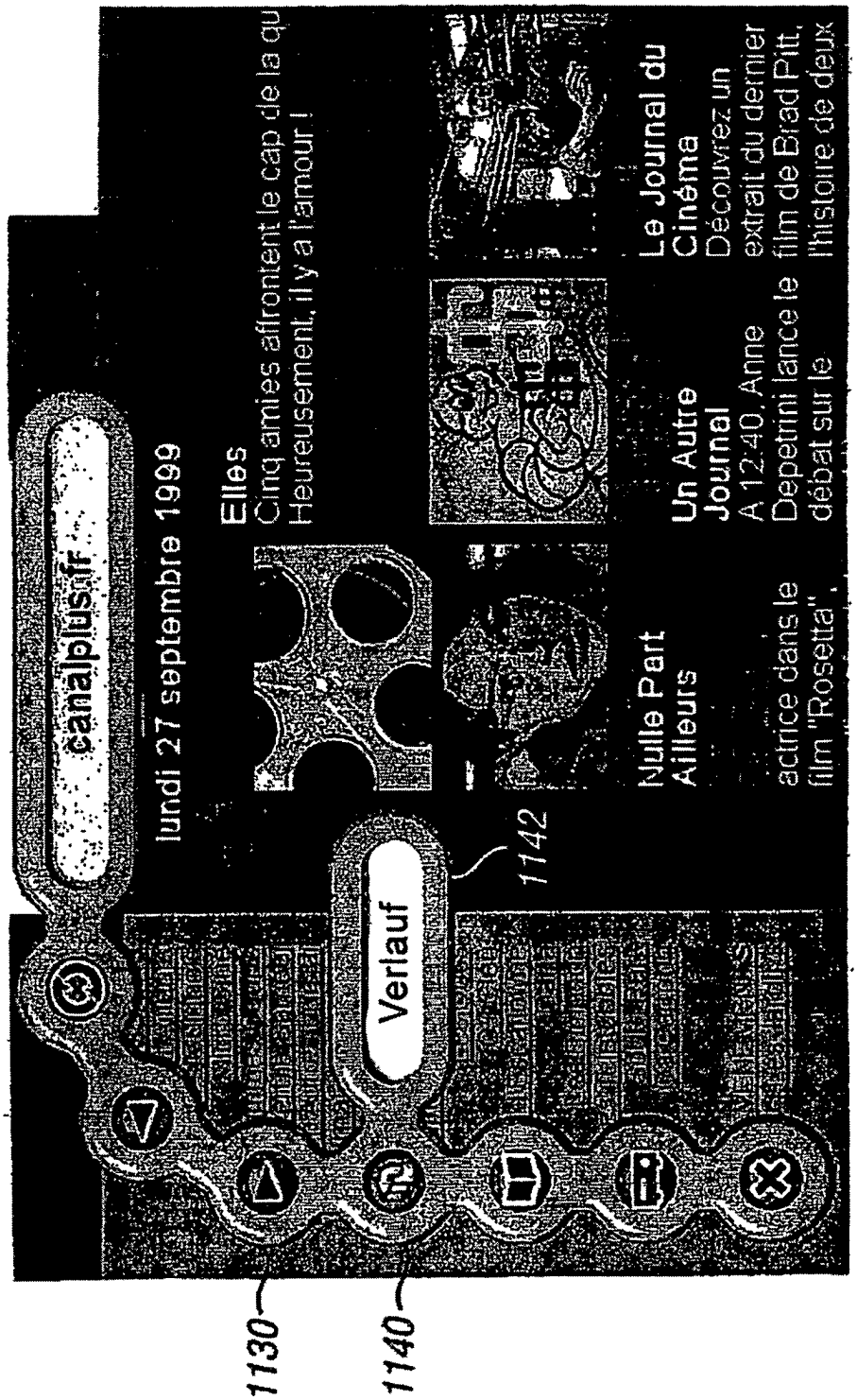

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Verlauf
1142
1130
1140

FIG. 11

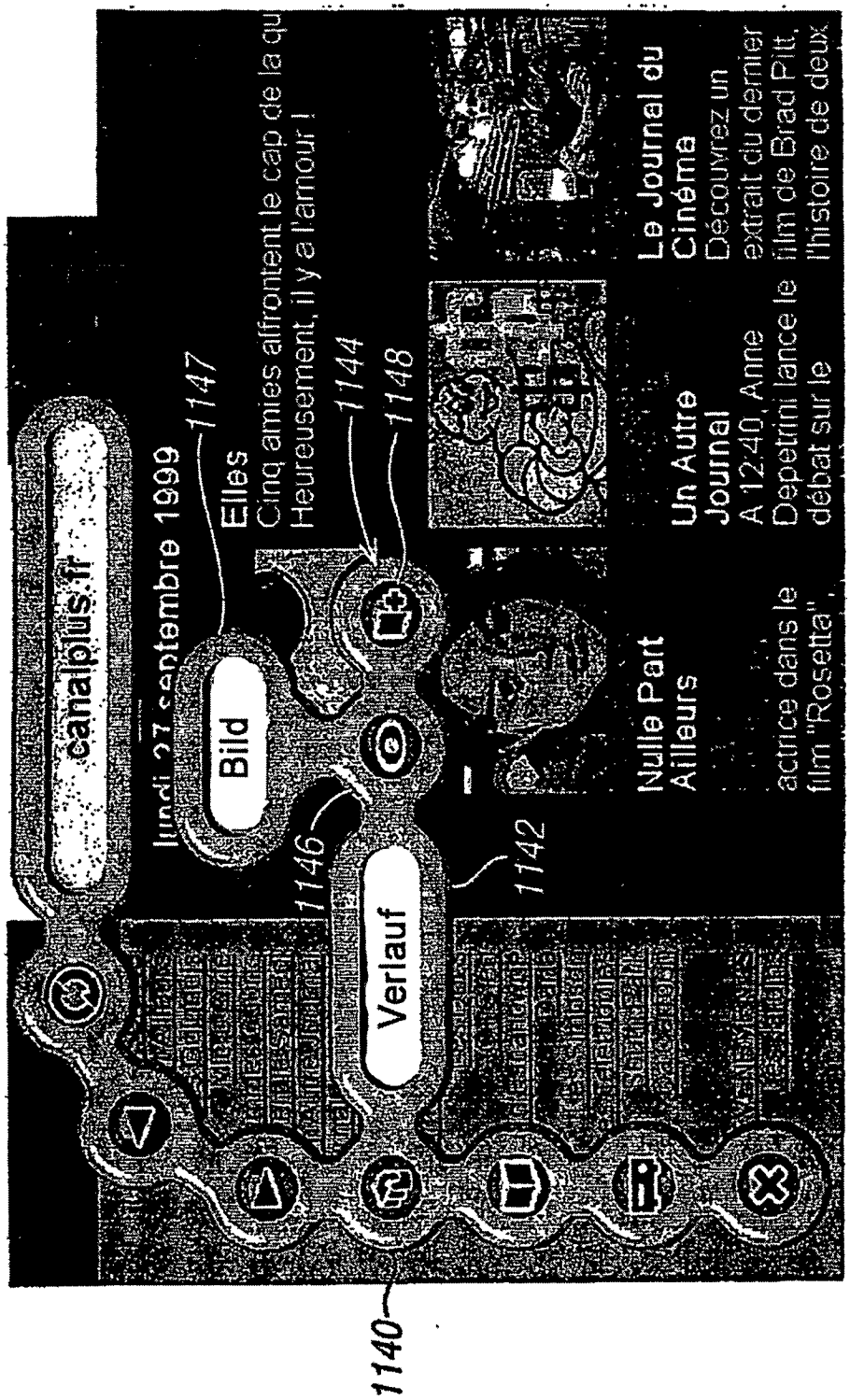

canalplus.fr
lundi 27 septembre 1999
Bild
Verlauf
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part
Ailleurs
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
actrice dans le
film "Rosetta",
1140
1142
1144
1146
1147
1148

FIG. 12

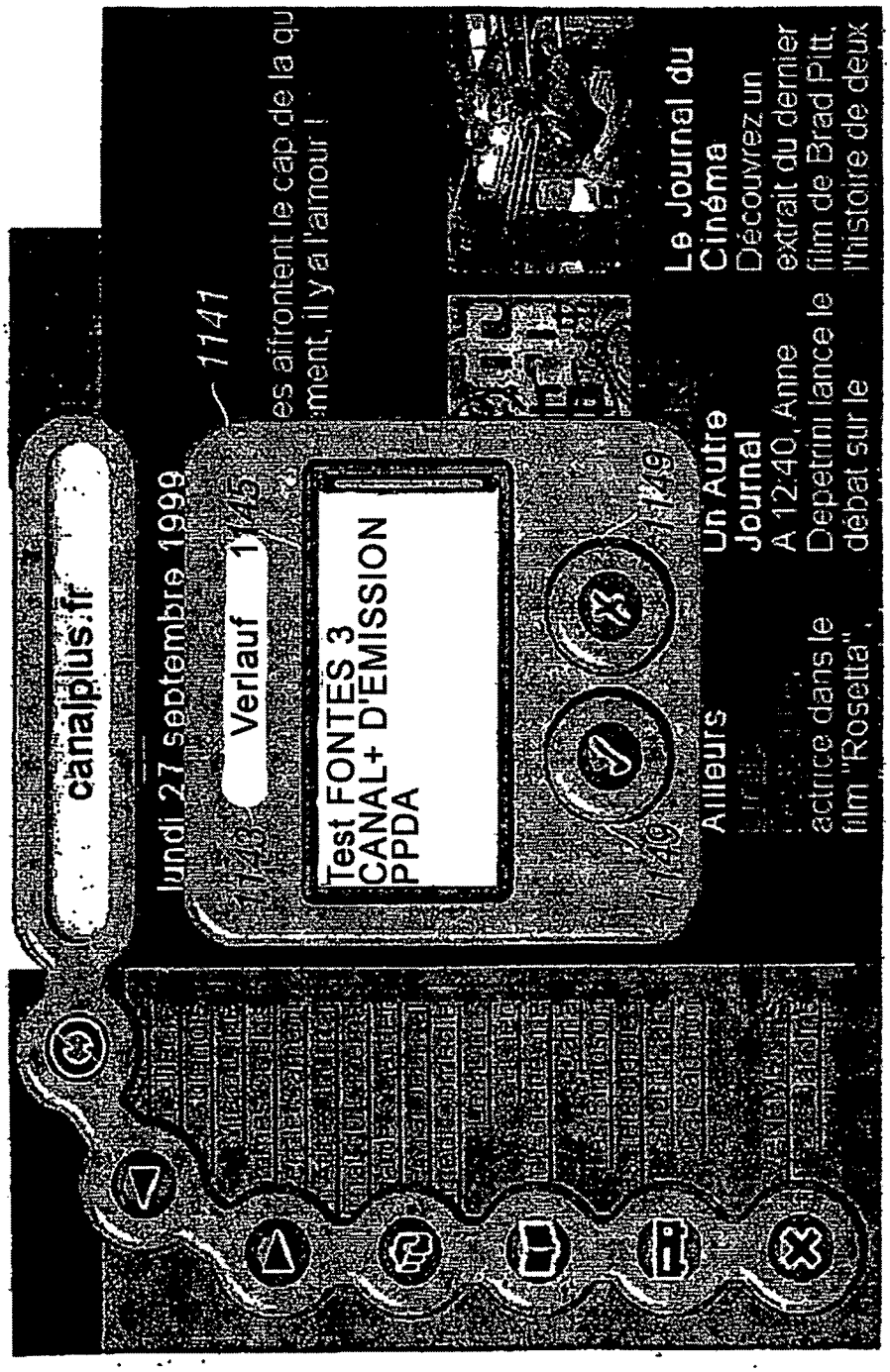

canalplus.fr
lundi 27 septembre 1999
Verlauf
1145
1141
Test FONTES 3
CANAL+ D'EMISSION
PPDA
Ailleurs
Un Autre Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux

FIG. 13

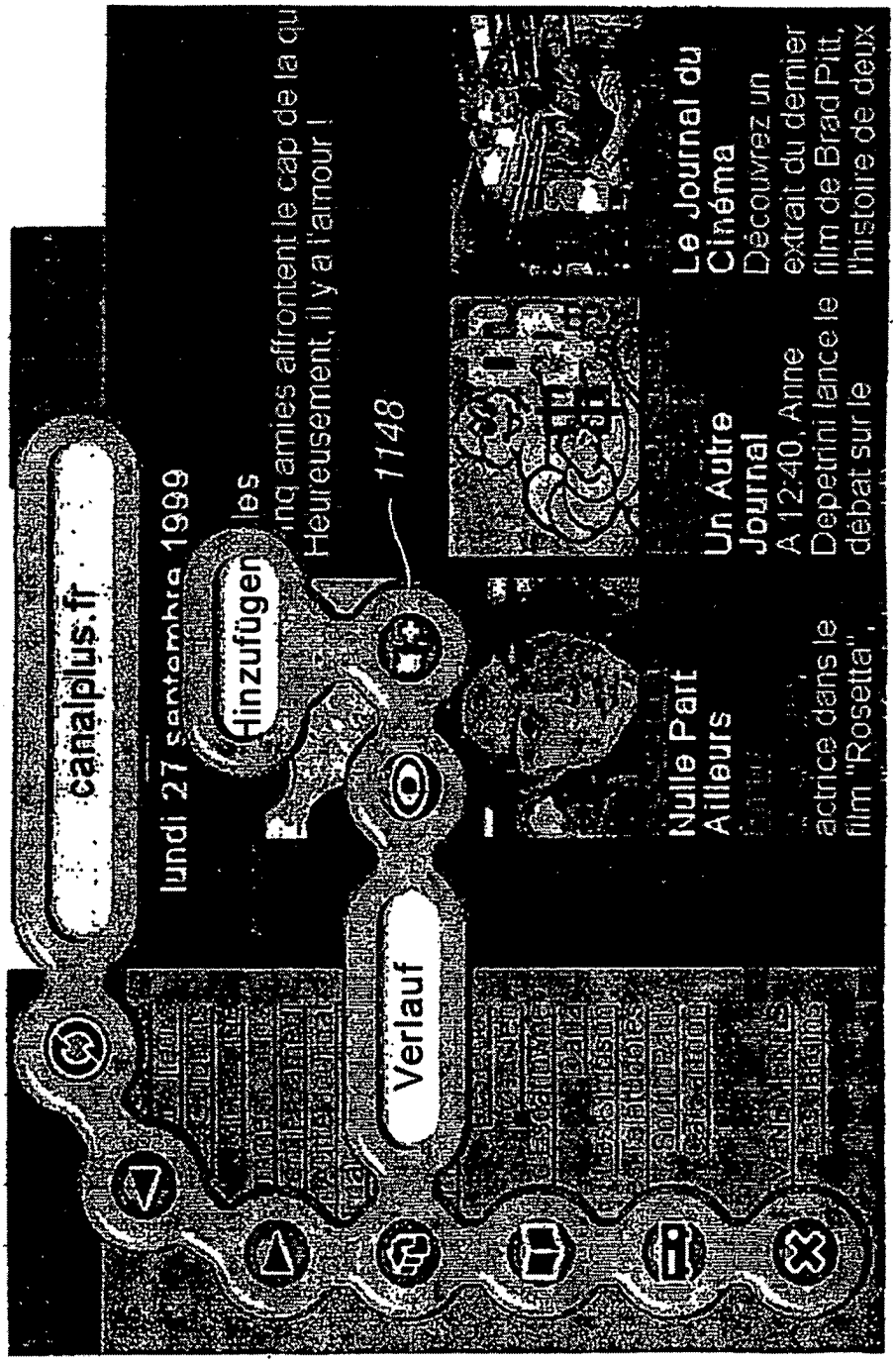

canalplus.fr
lundi 27 septembre 1999
Hinzufügen
Verlauf
1148
Heureusement, il y a l'amour !
Nulle Part Ailleurs
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
actrice dans le film "Rosetta",

FIG. 14

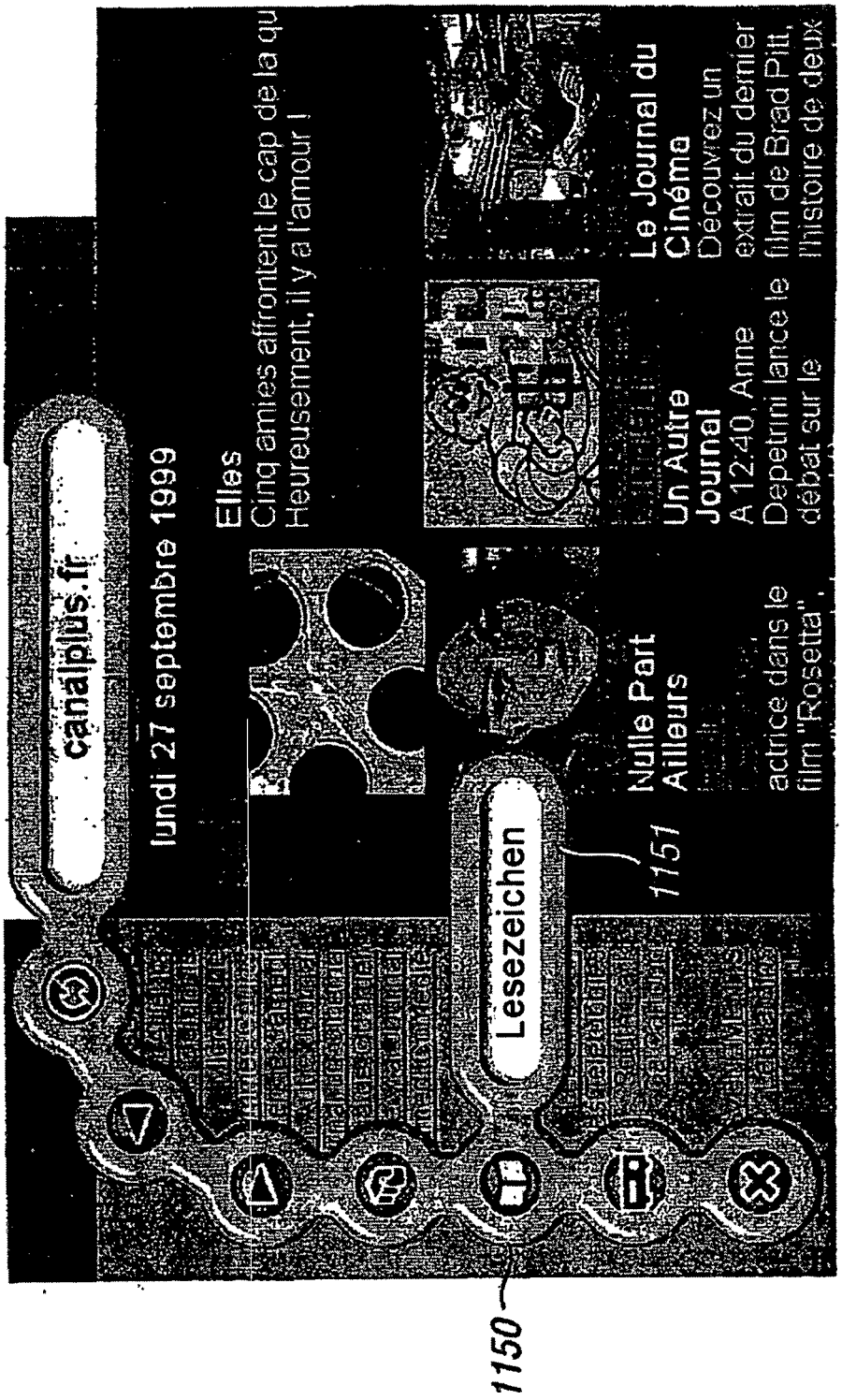

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Lesezeichen
1151
1150

FIG. 15

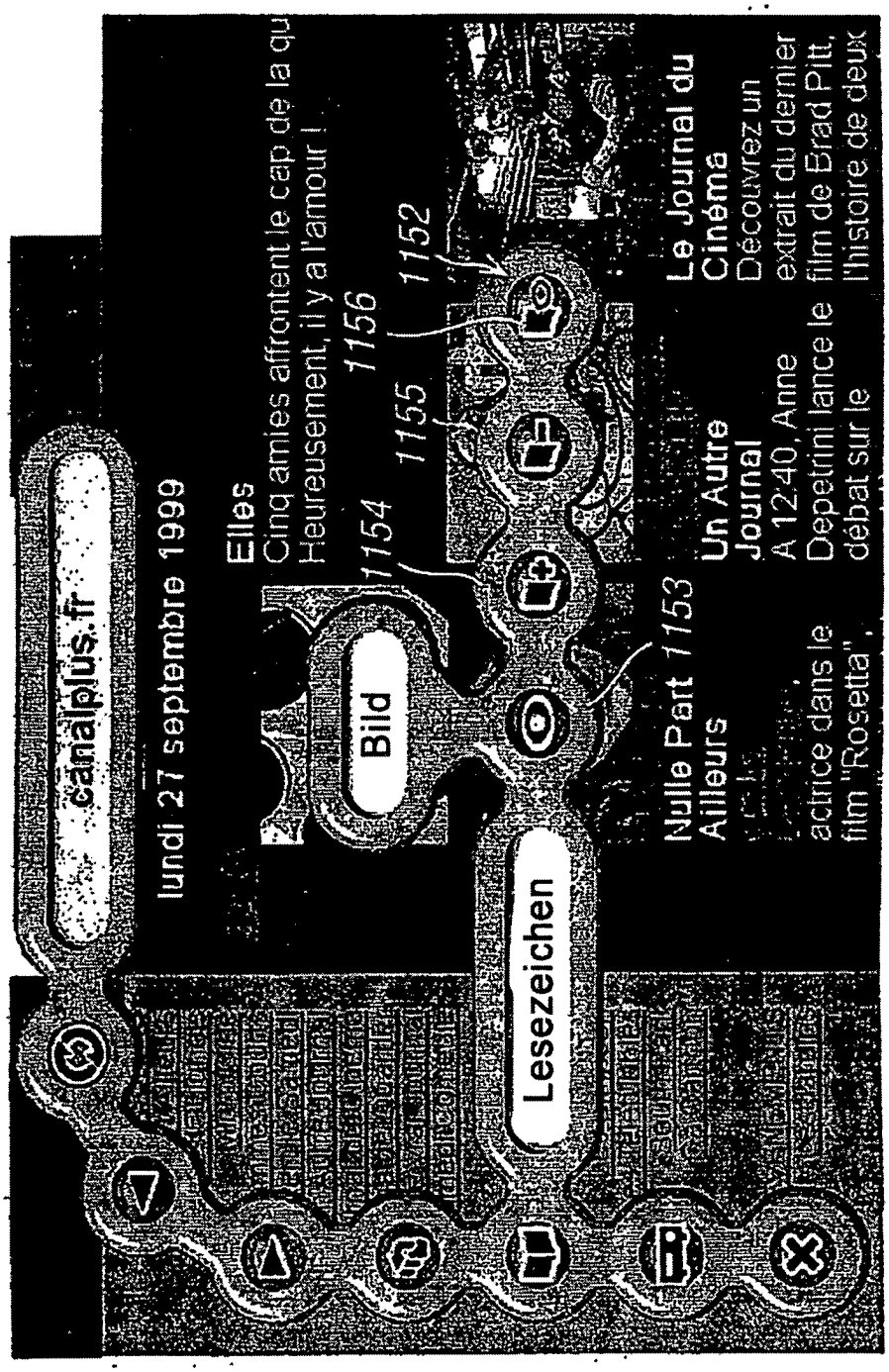

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
1156
1155
1152
1154
Bild
Nulle Part 1153
Ailleurs
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
actrice dans le
film "Rosetta",
Lesezeichen

FIG. 16

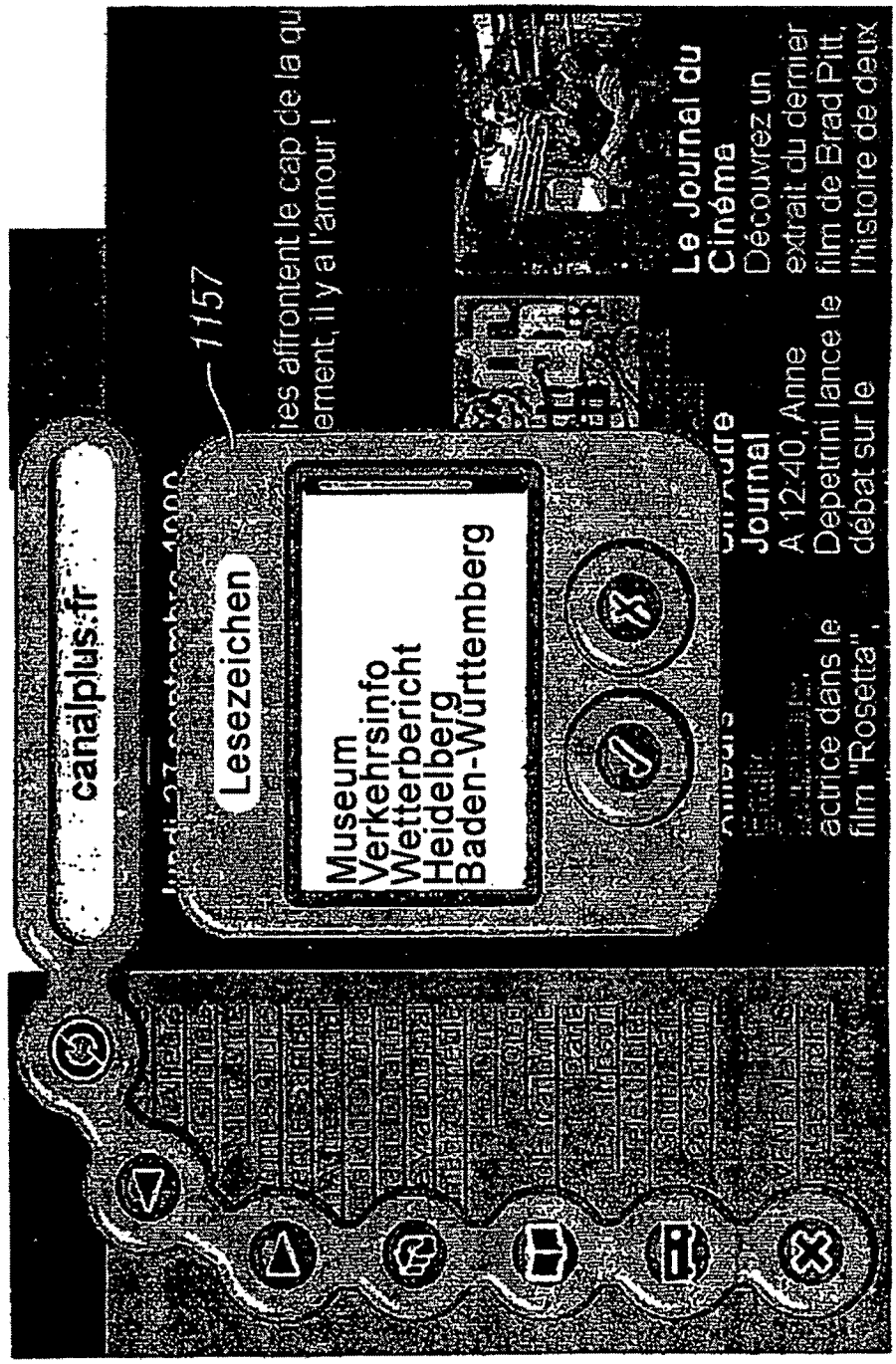

canalplus.fr
Lesezeichen
Museum
Verkehrsinfo
Wetterbericht
Heidelberg
Baden-Württemberg
1157
les affrontent le cap de la qu
ement, il y a l'amour !
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
actrice dans le
film "Rosetta",

FIG. 17

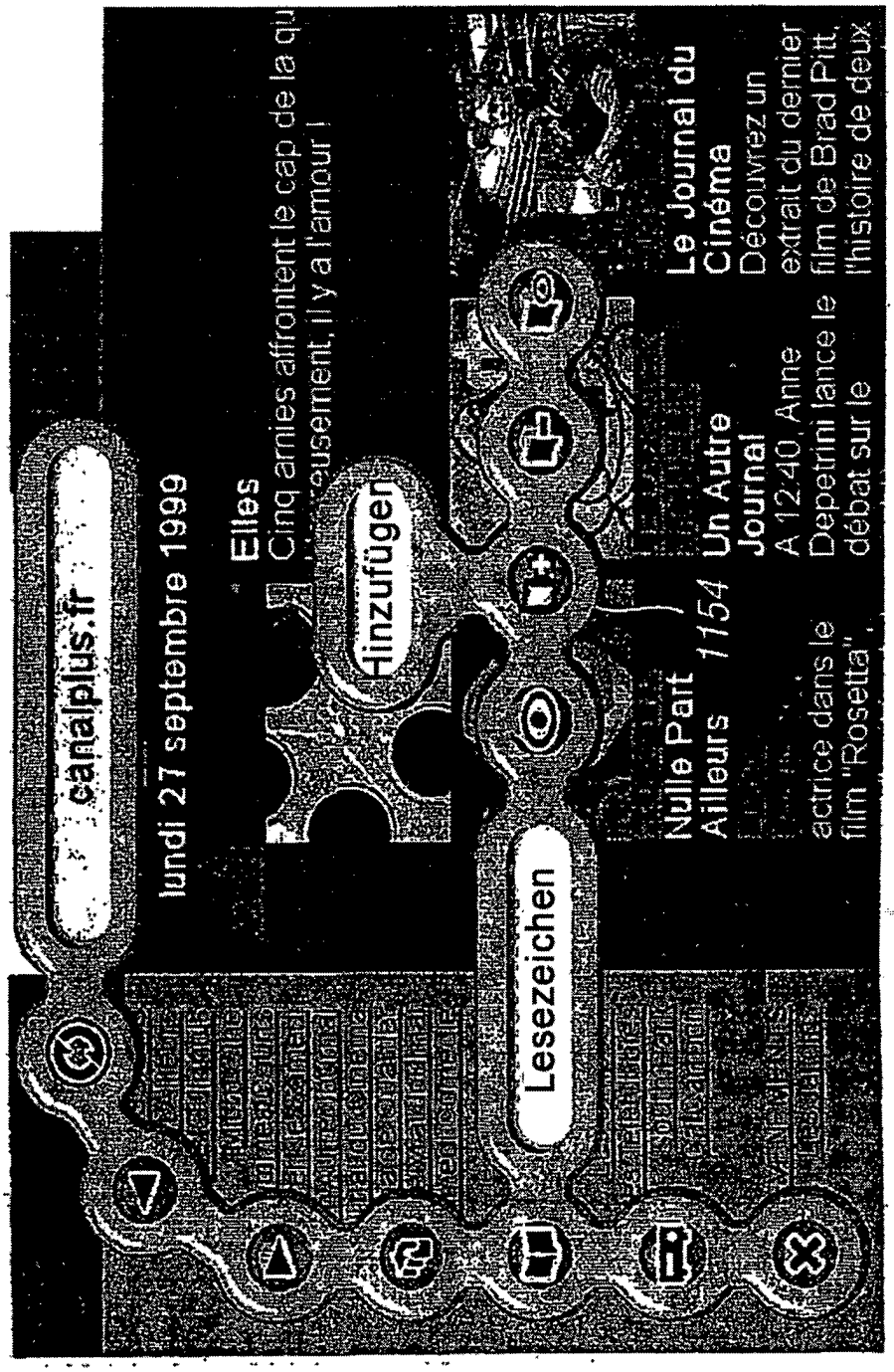

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Hinzufügen
Lesezeichen
Nulle Part Ailleurs 1154
Un Autre Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
actrice dans le
film "Rosetta",

FIG. 18

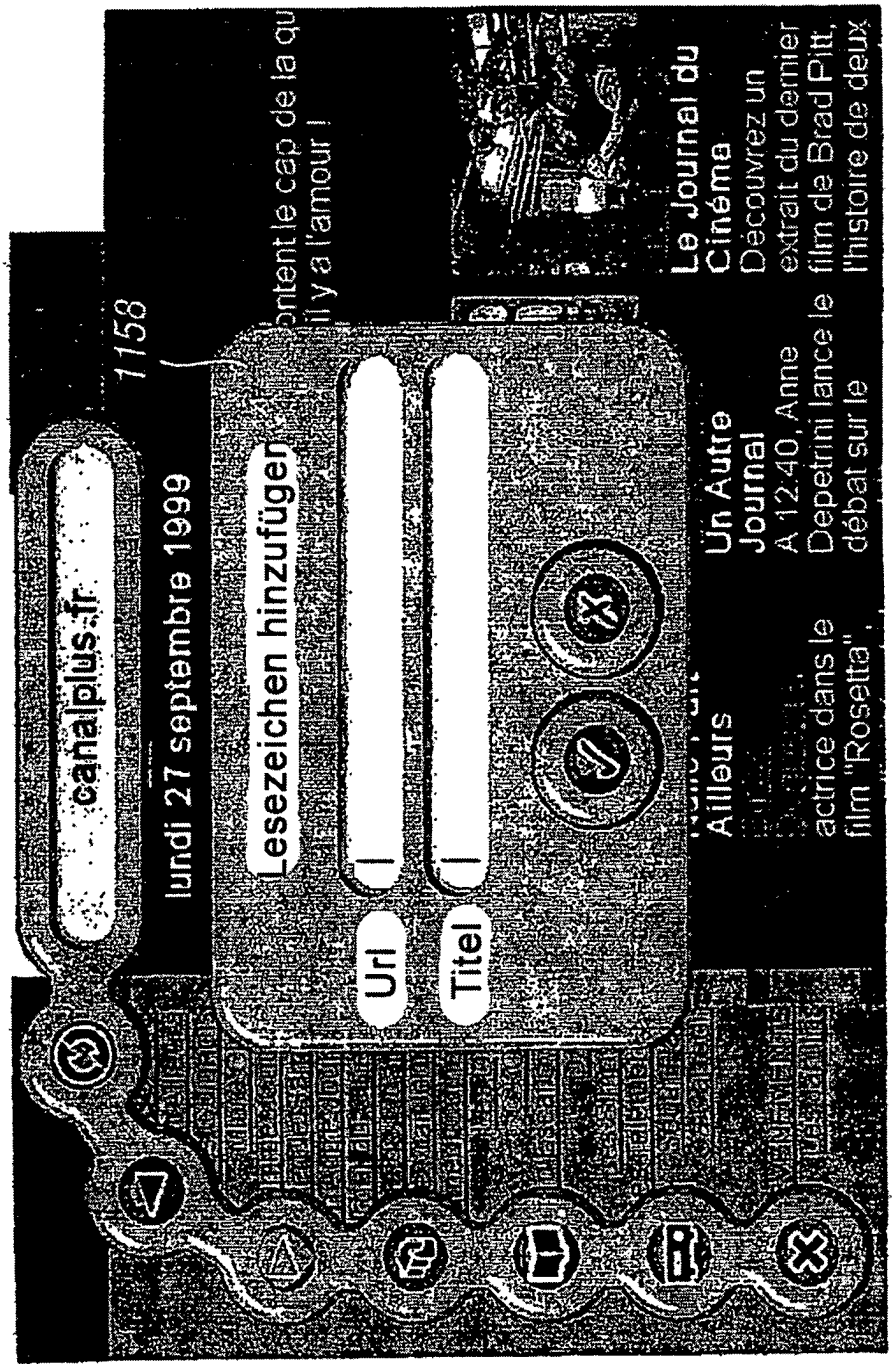

canalplus.fr
lundi 27 septembre 1999
1158
Lesezeichen hinzufügen
Url
Titel
Ailleurs
Un Autre Journal
A 12.40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux

FIG. 19

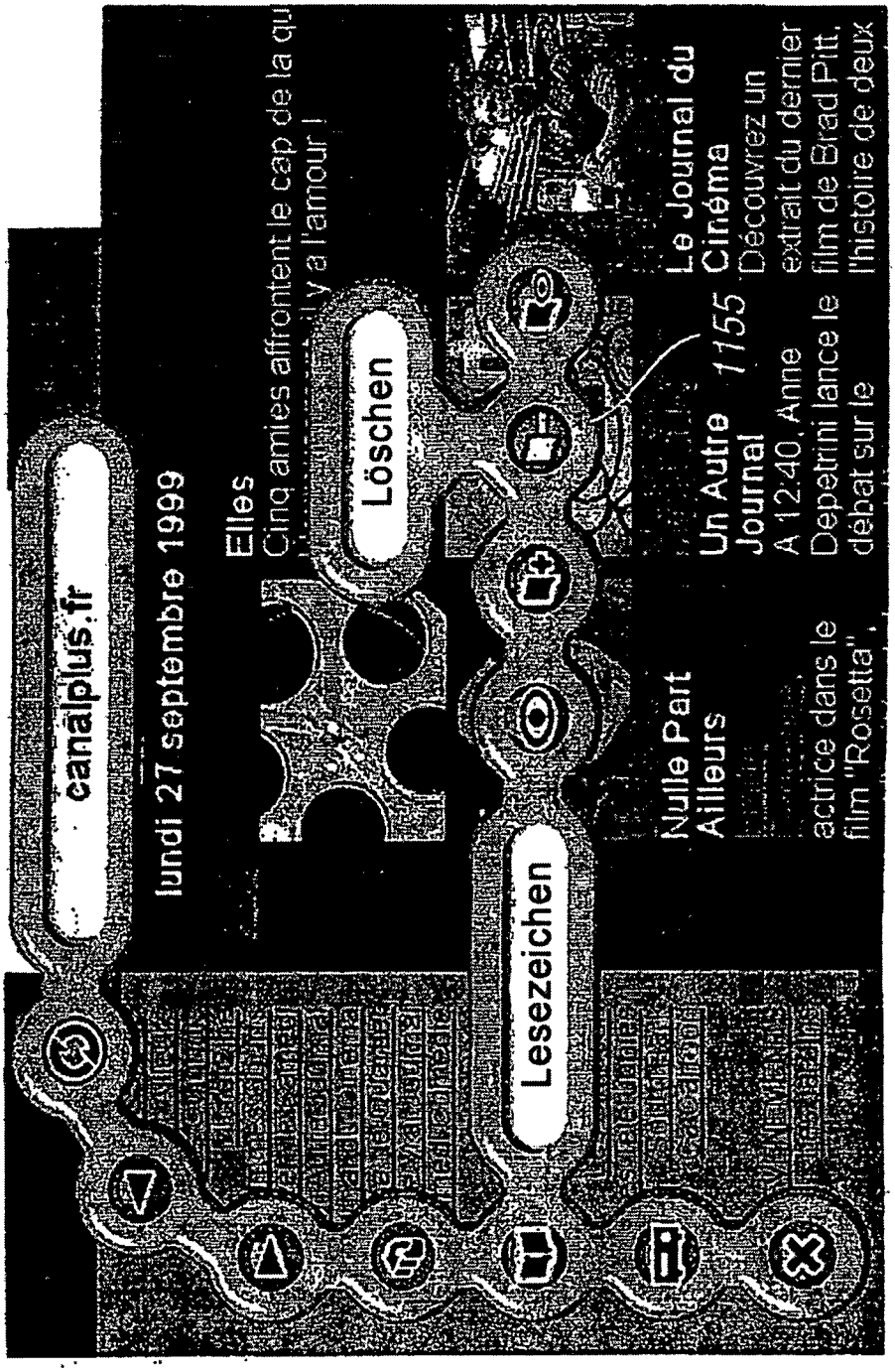

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
il y a l'amour !
Löschen
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
1155
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Lesezeichen

FIG. 20

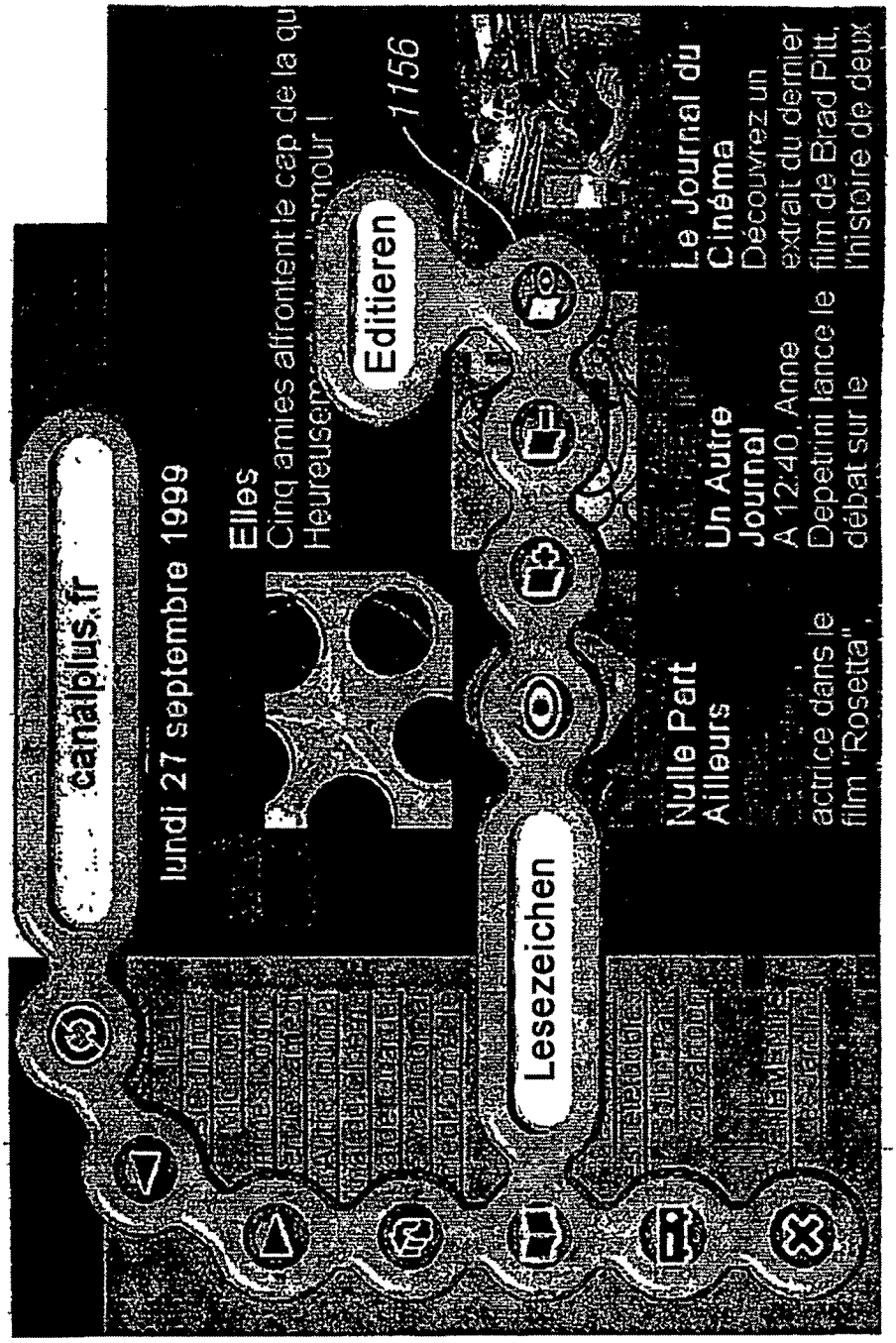

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Editieren
1156
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Lesezeichen

FIG. 21

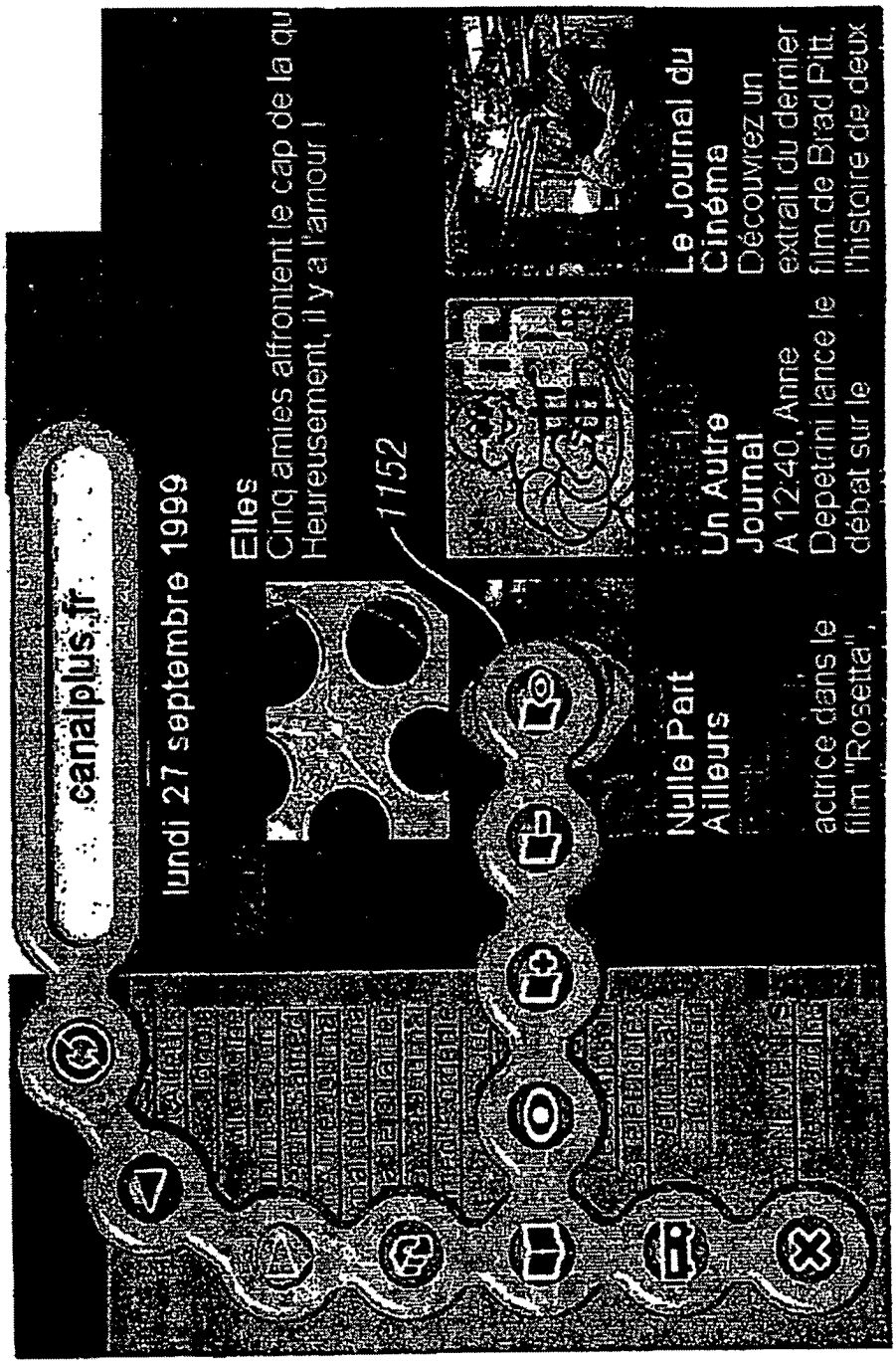

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
1152
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le film "Rosetta",

FIG. 22

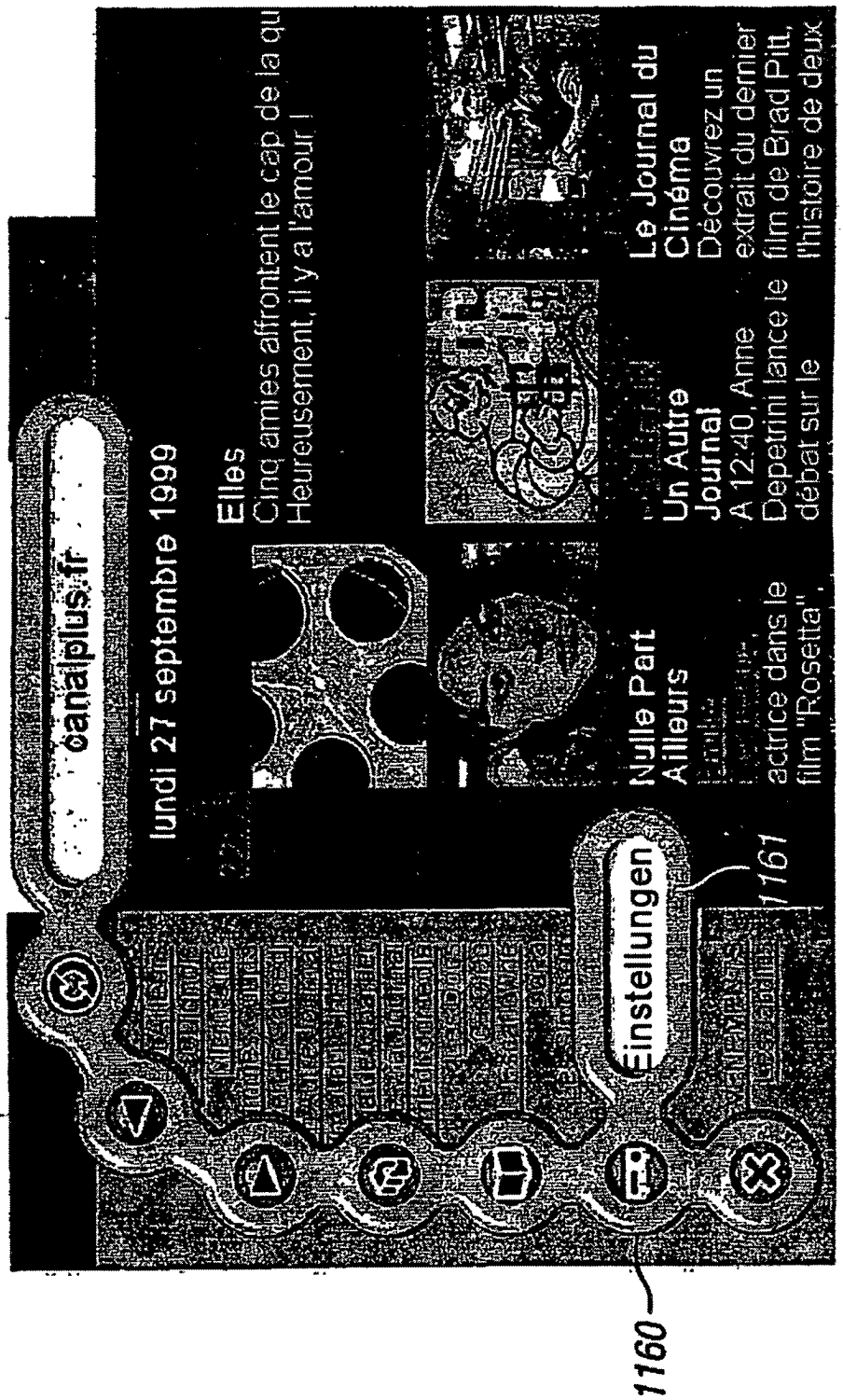

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Einstellungen
1160
1161

FIG. 23

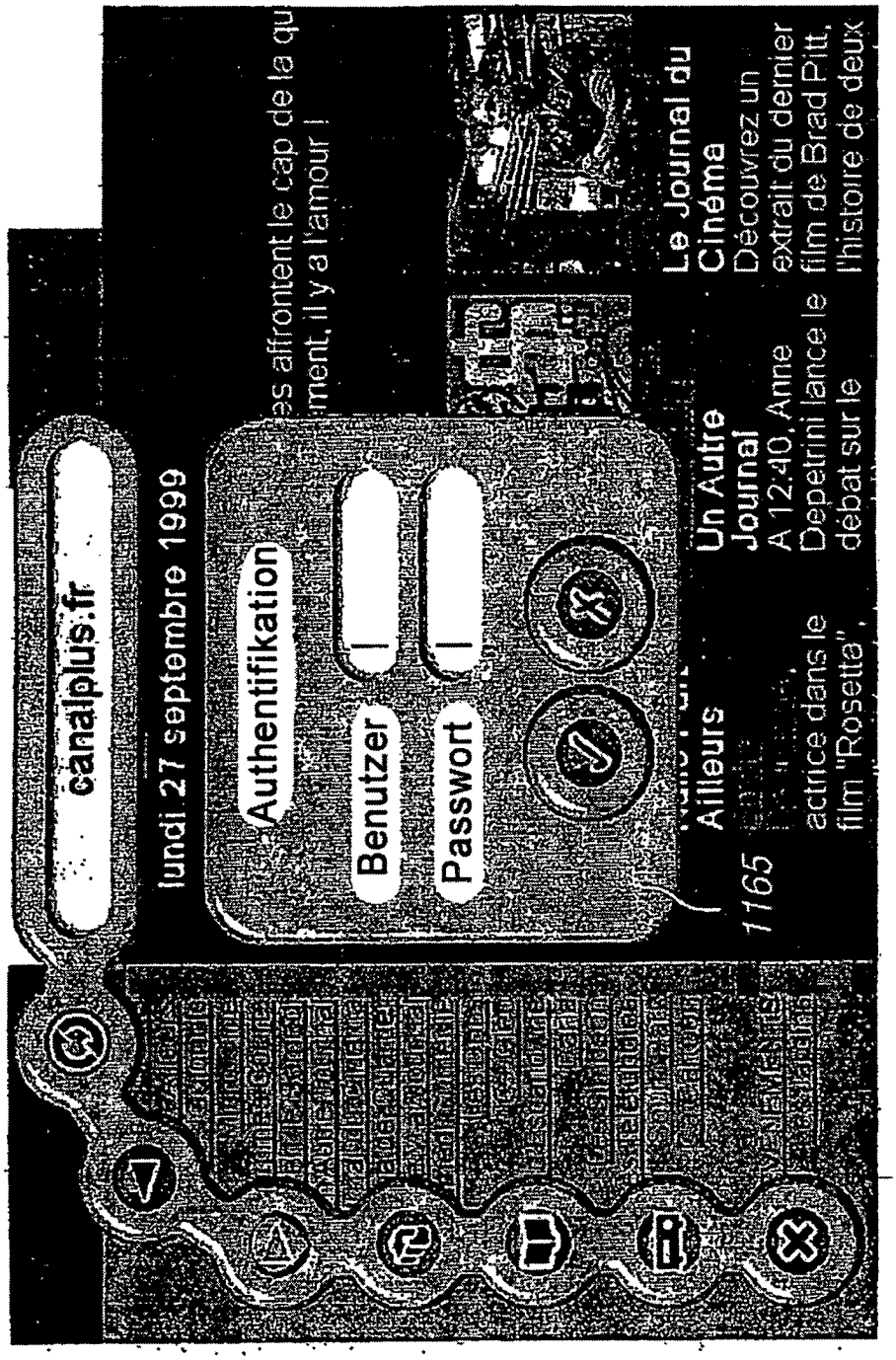

canalplus.fr
lundi 27 septembre 1999
Authentifikation
Benutzer
Passwort
Un Autre Journal
A 12.40, Anne Depetrini lance le debat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux

FIG. 24

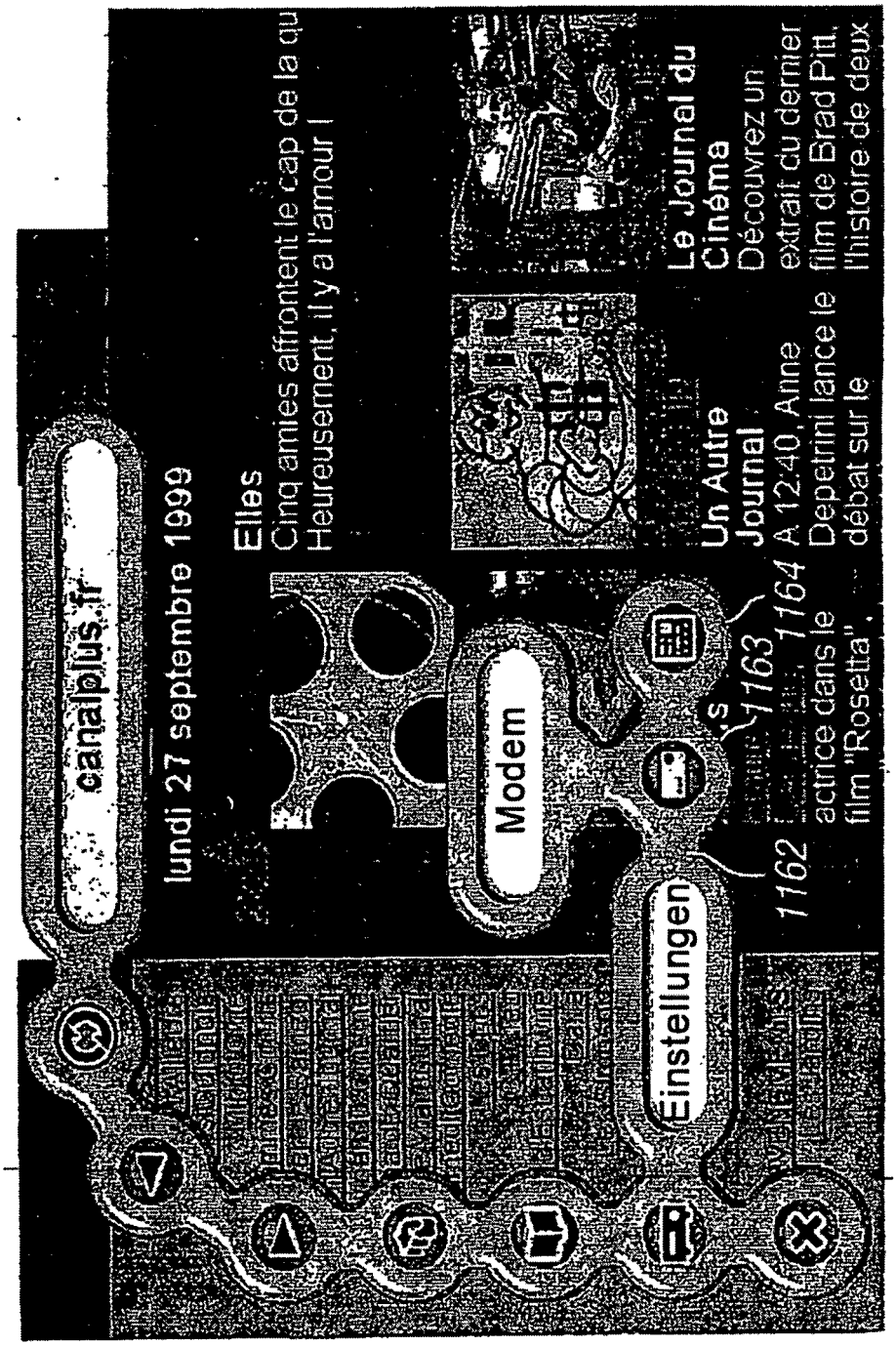

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film "Rosetta".
1162
1163
1164
Modem
Einstellungen

FIG. 25

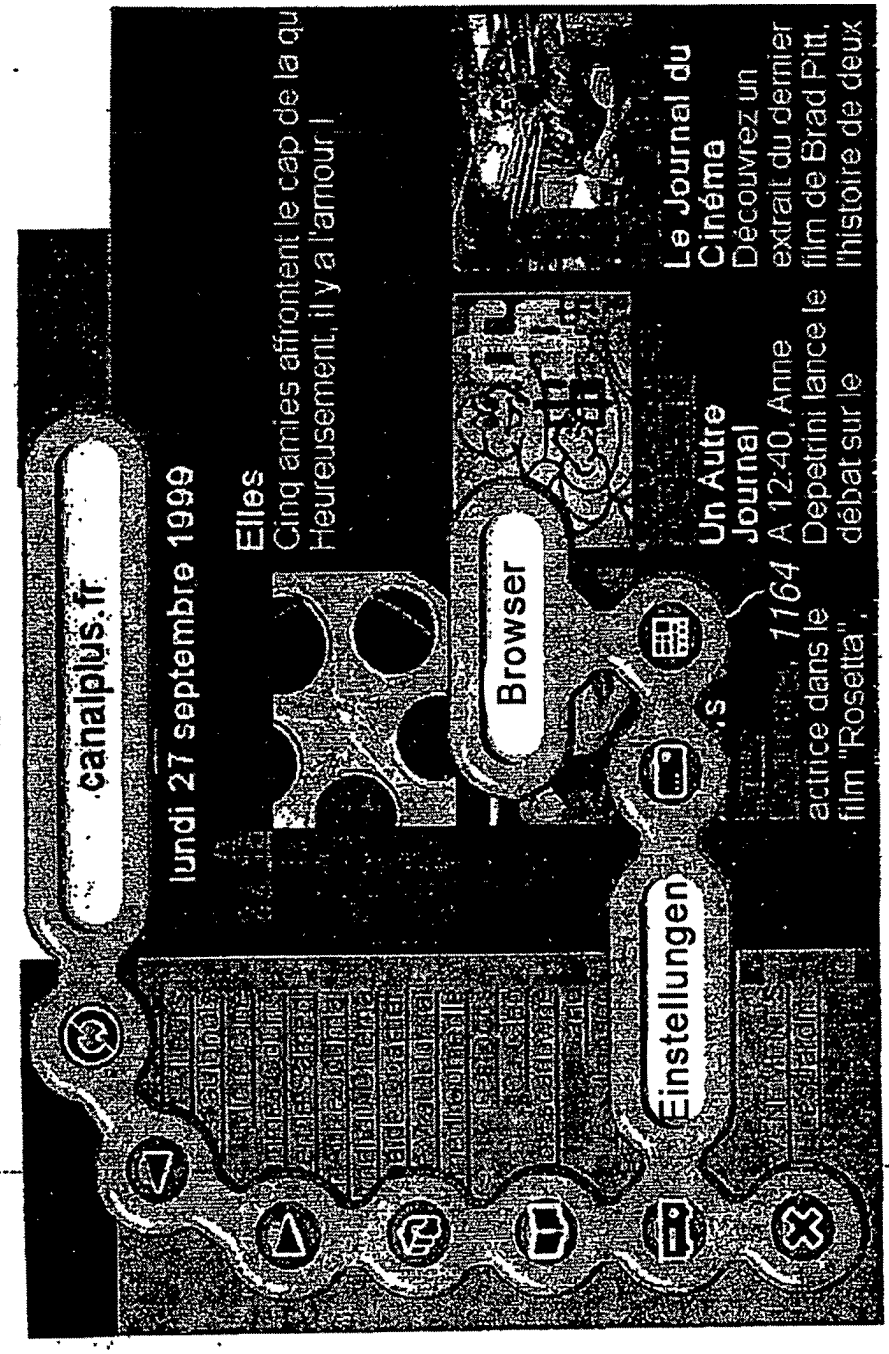

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Browser
Einstellungen
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux

FIG. 26

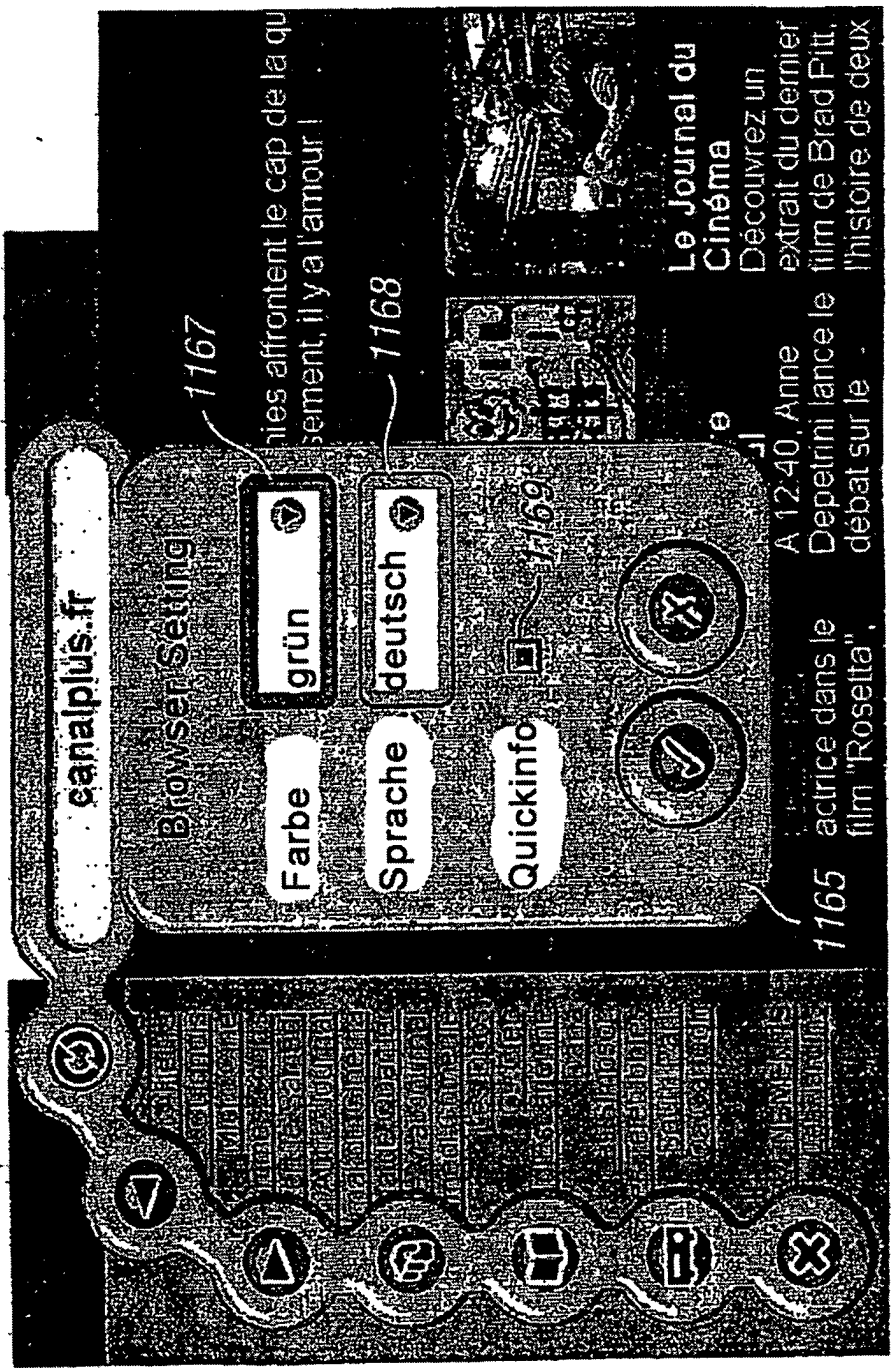

canalplus.fr
Browser setting
1167
1168
Farbe
grün
Sprache
deutsch
Quickinfo
1169
1165
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film "Rosetta",

FIG. 27

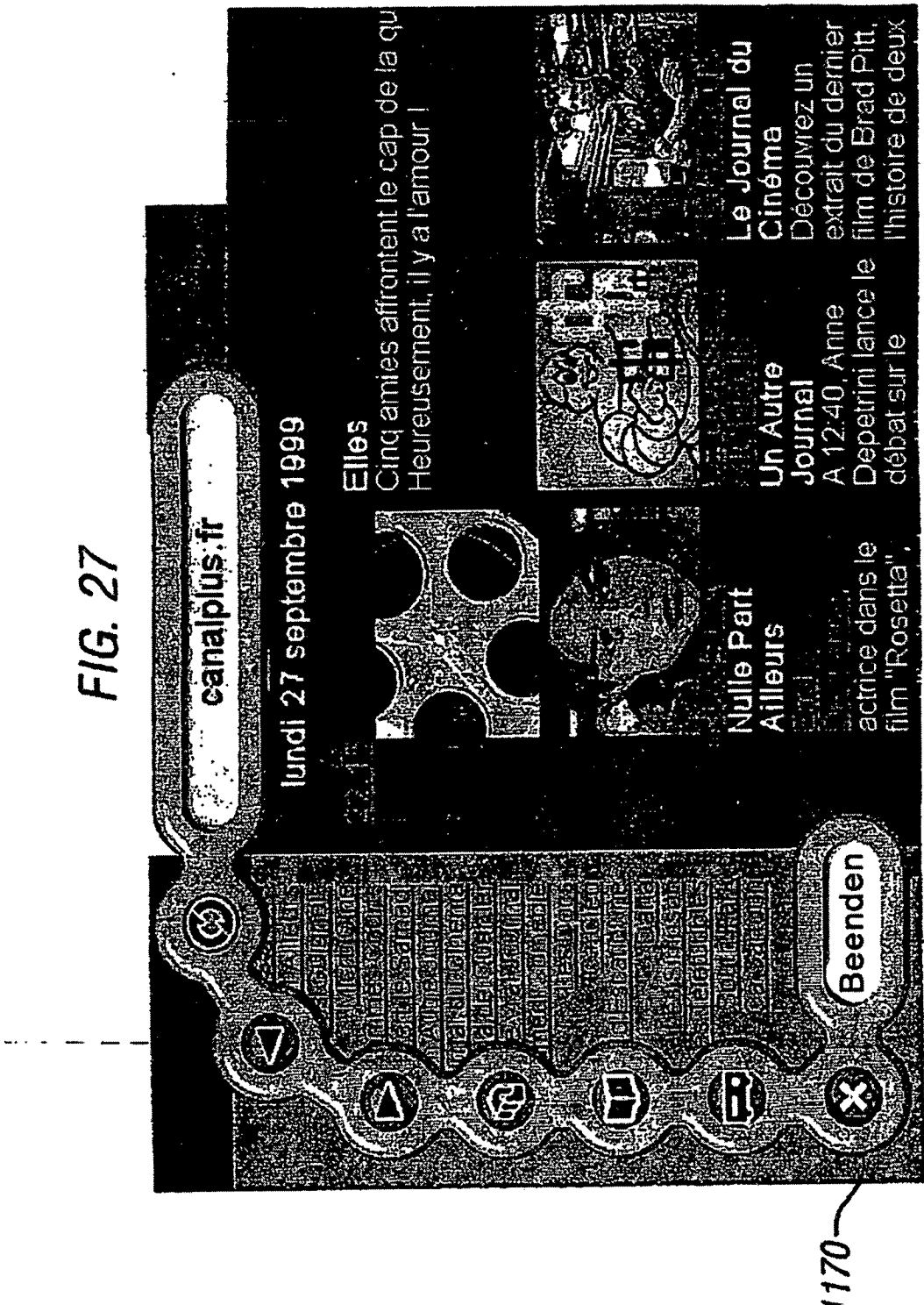

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
actrice dans le film "Rosetta",
Beenden
1171
1170

FIG. 28

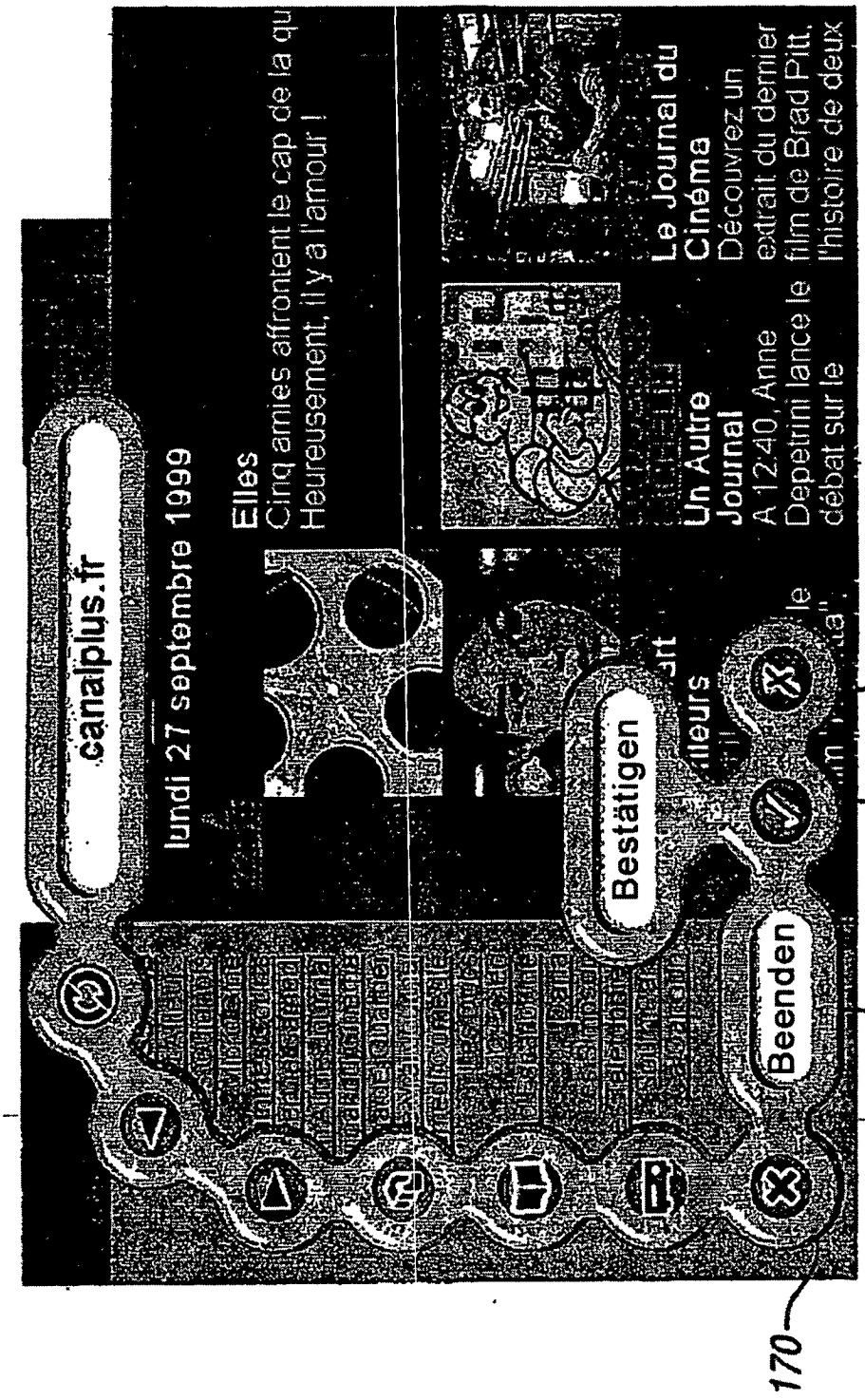

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Bestätigen
Beenden
170
1171
1172
1173

FIG. 29

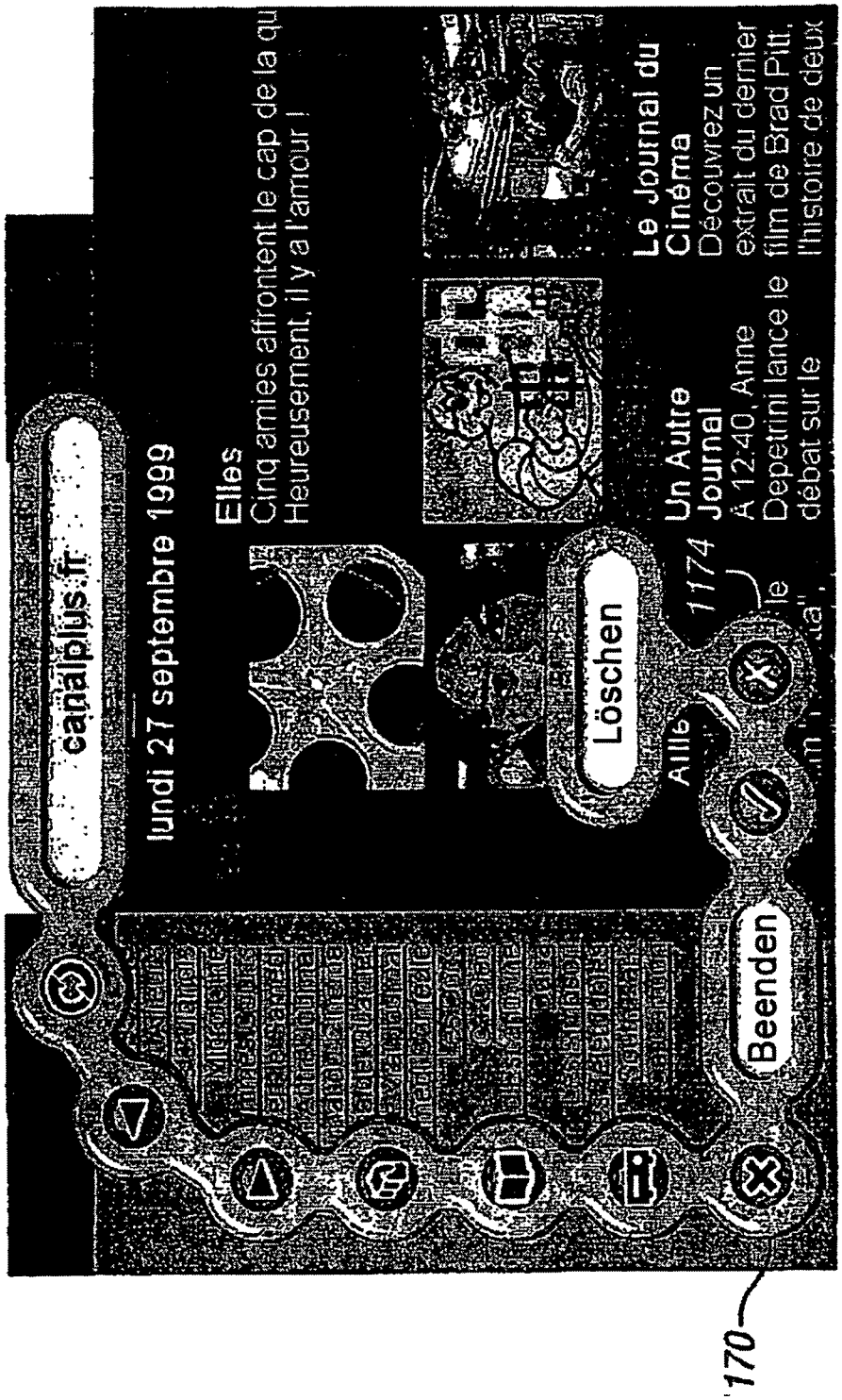

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
À 12:40, Anne Depetrini lance le débat sur le
Löschen
Alle
1174
Beenden
170

FIG. 30

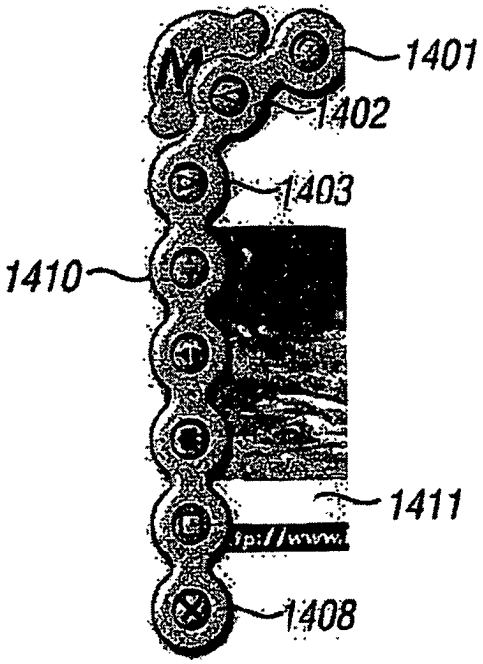

1401
1402
1403
1410
1411
1408

FIG. 31

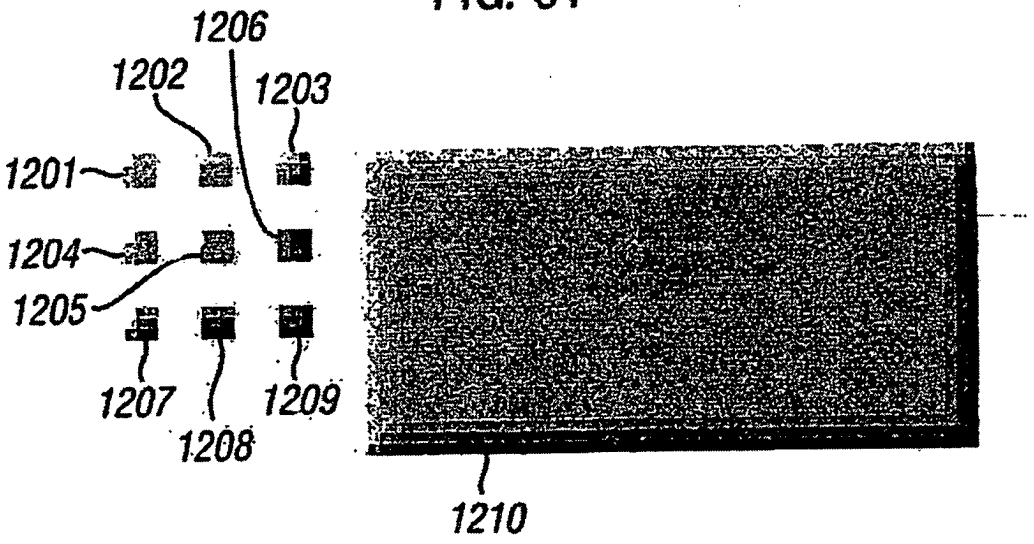

1206
1202
1203
1201
1204
1205
1207
1208
1209
1210

FIG. 32

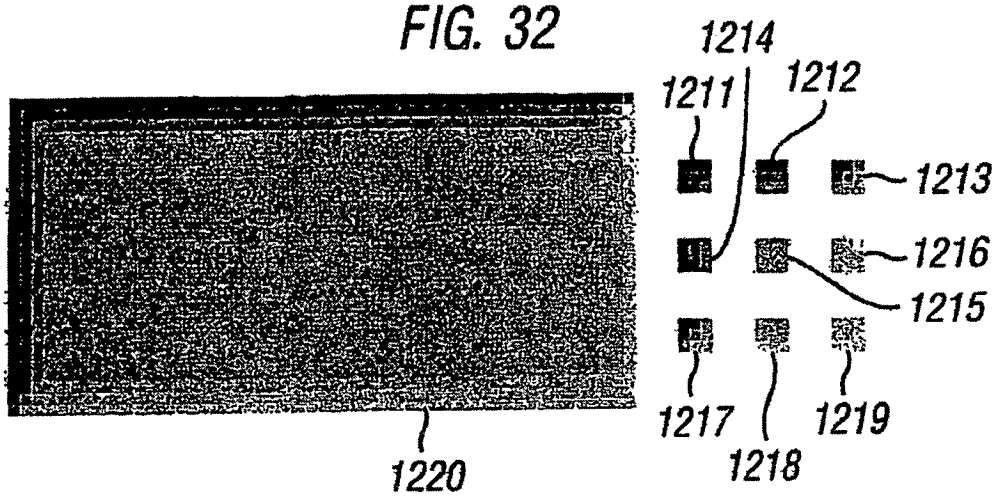


FIG. 33

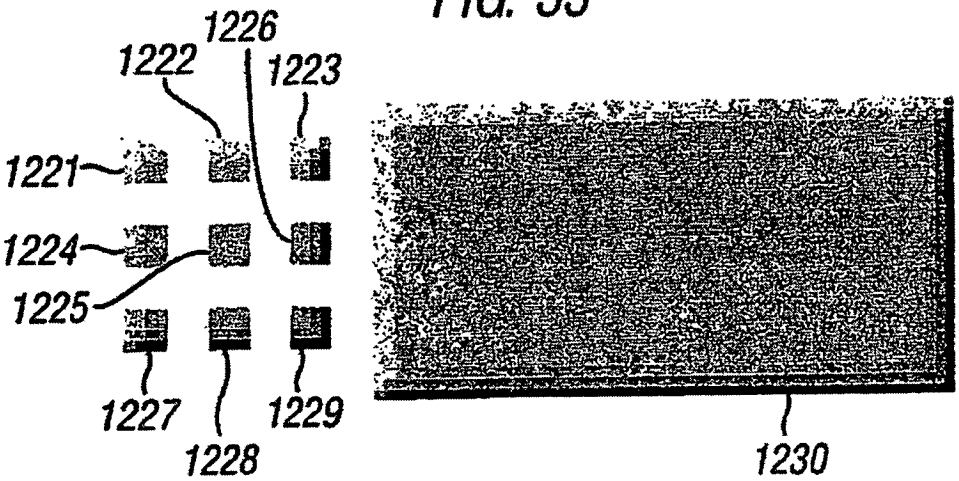
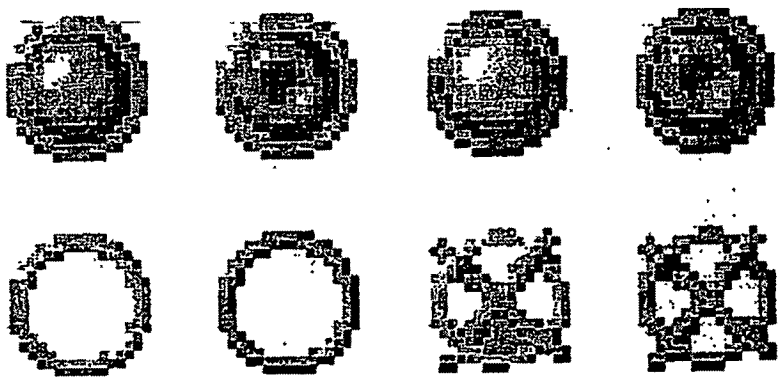


FIG. 34



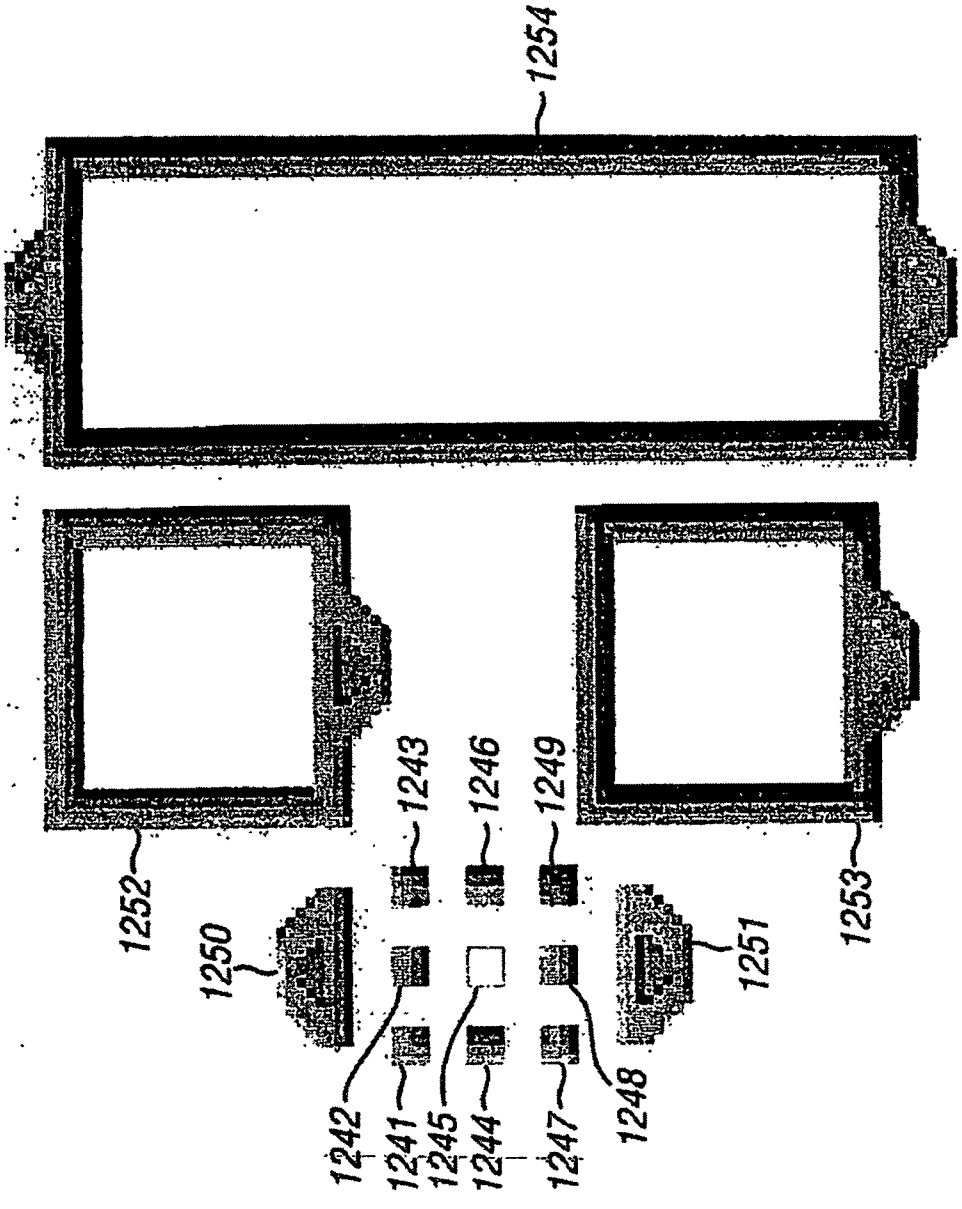

1254
1252
1250
1243
1246
1249
1253
1251
1242
1241
1245
1244
1247
1248

FIG. 35

FIG. 36

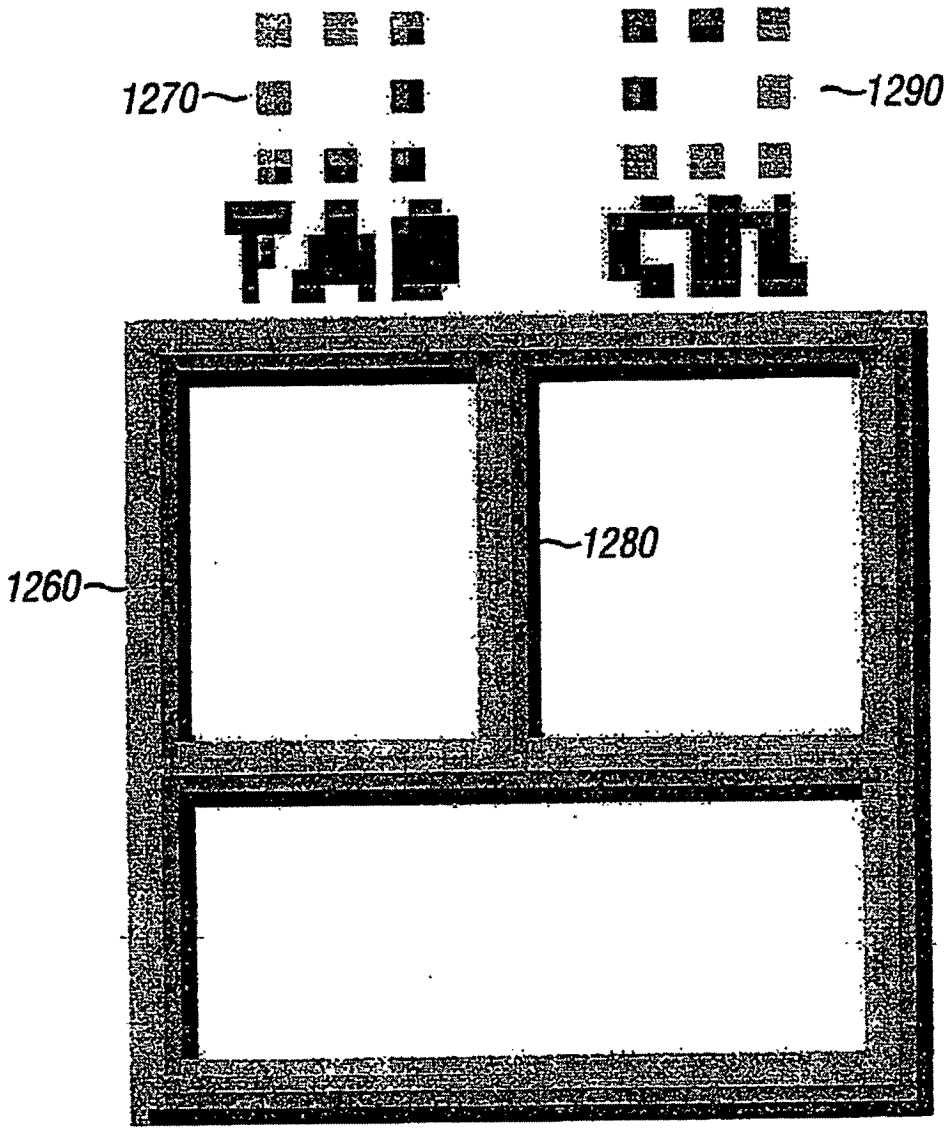


FIG. 37

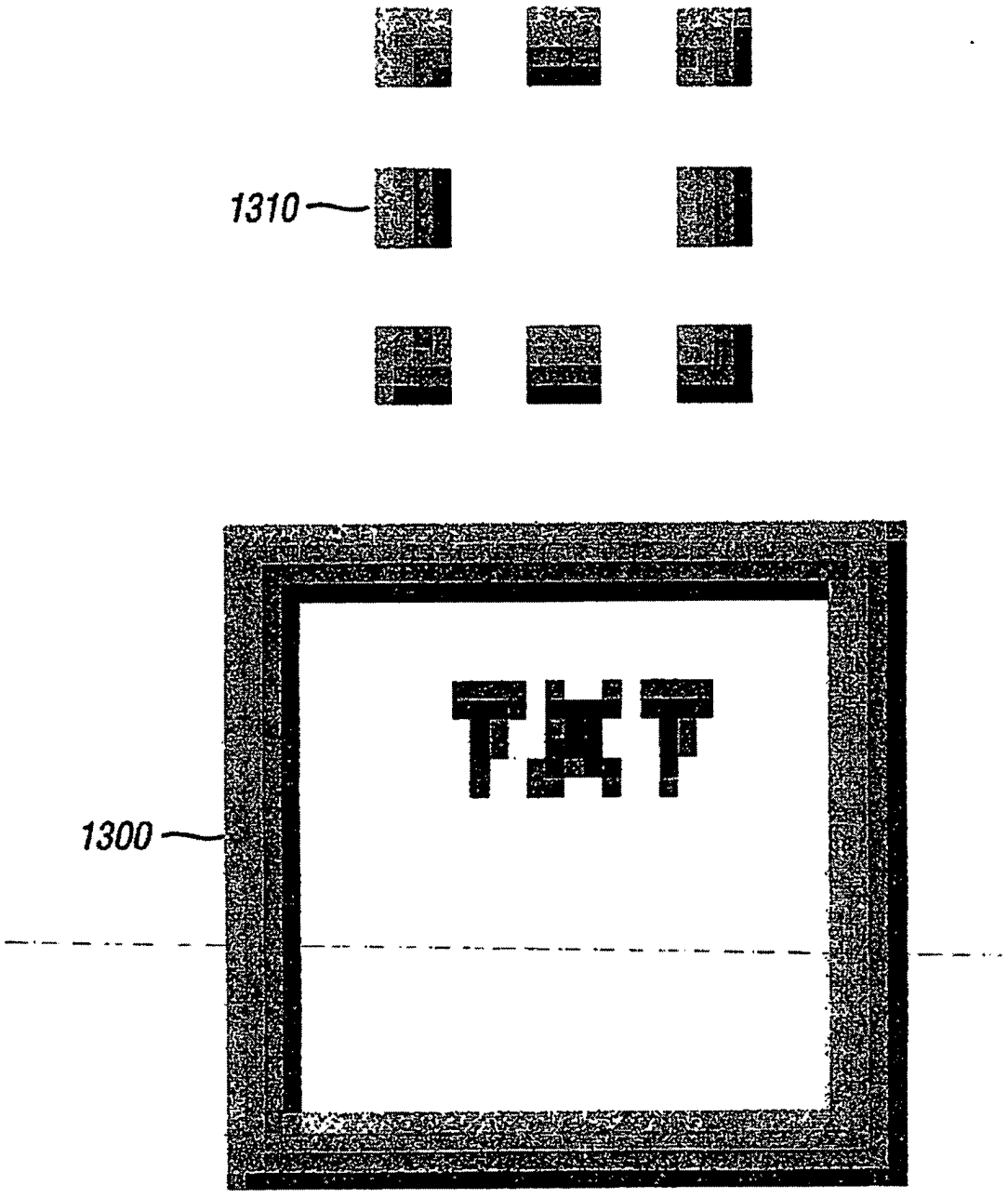


FIG. 38

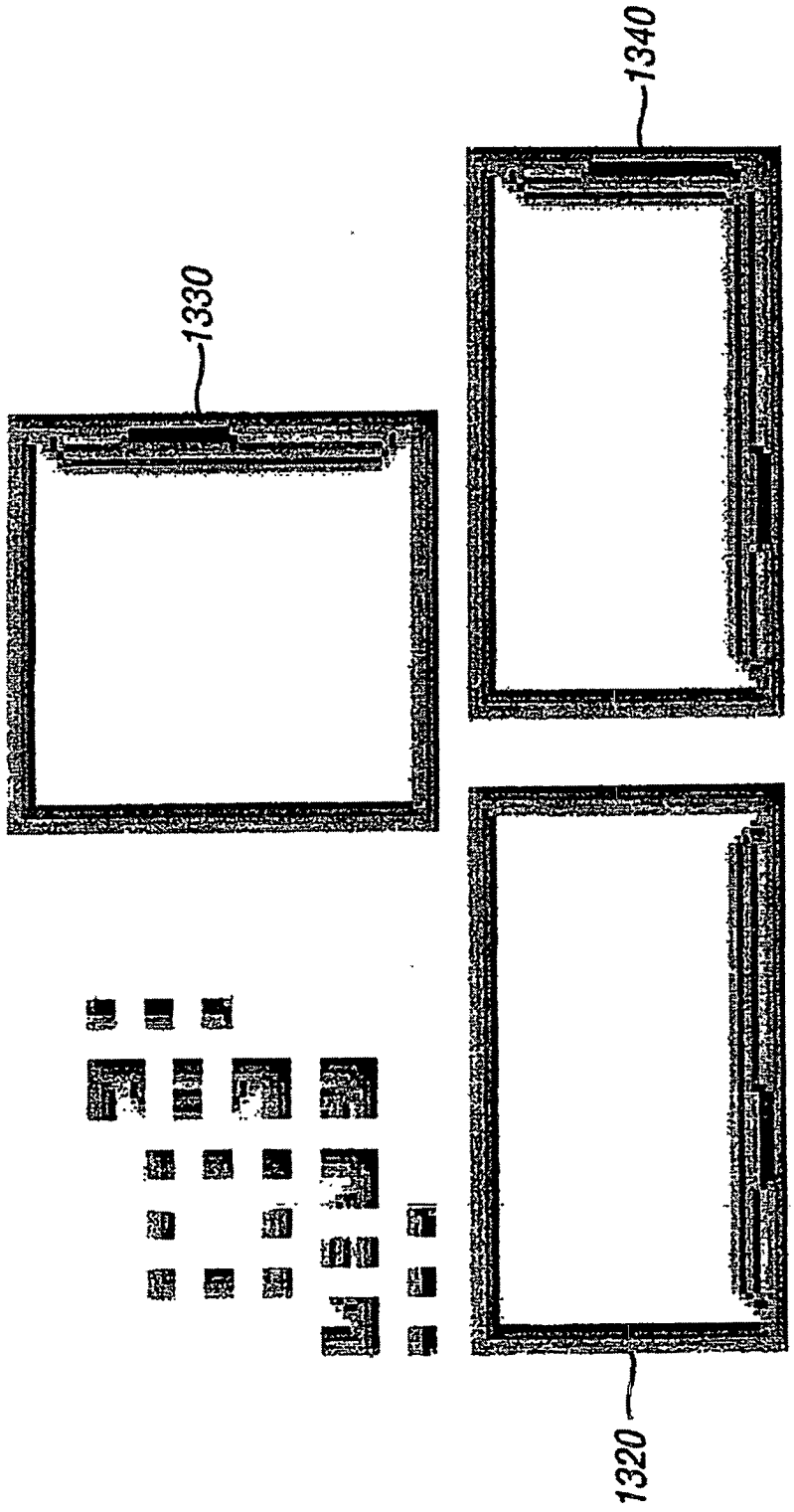


FIG. 39

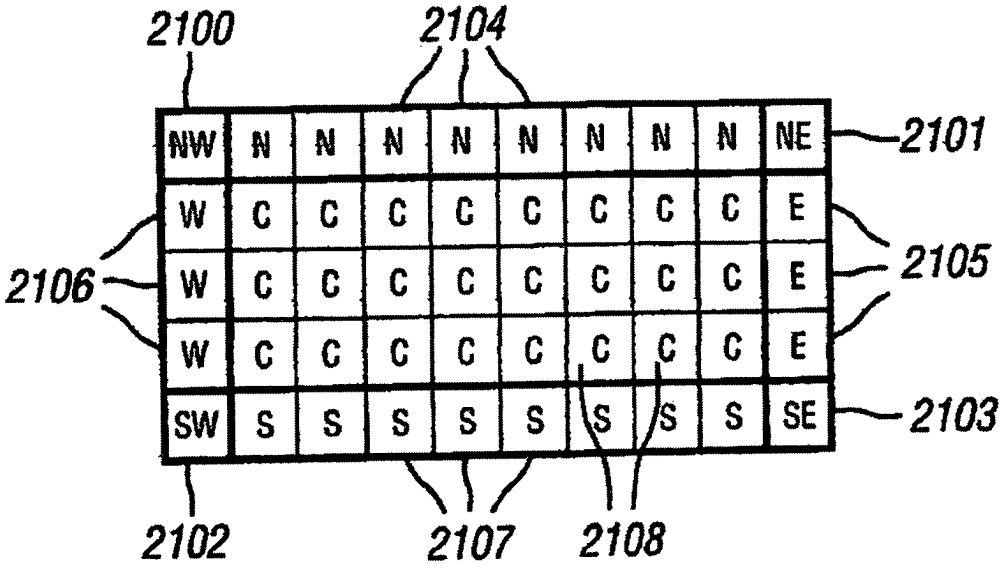


FIG. 40

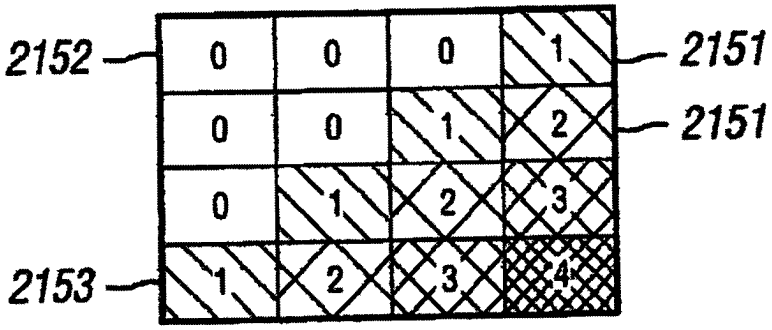


FIG. 41

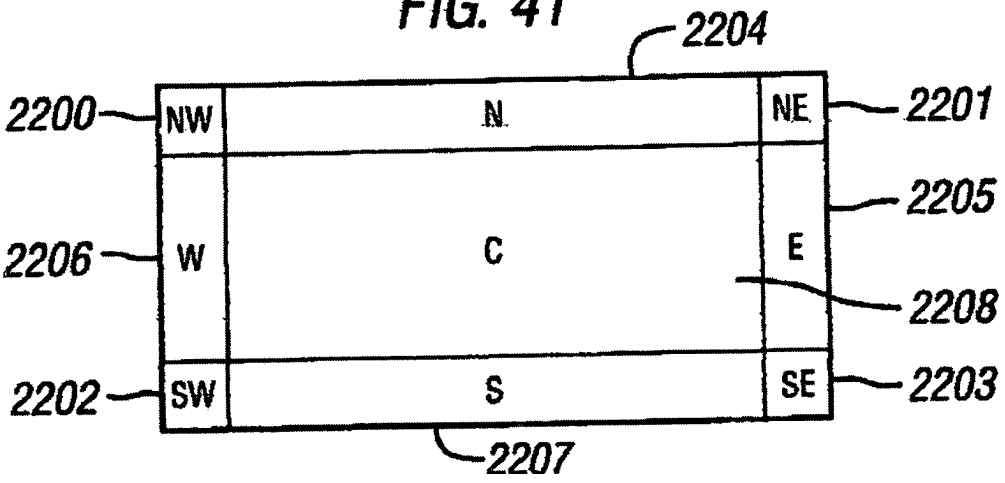


FIG. 42

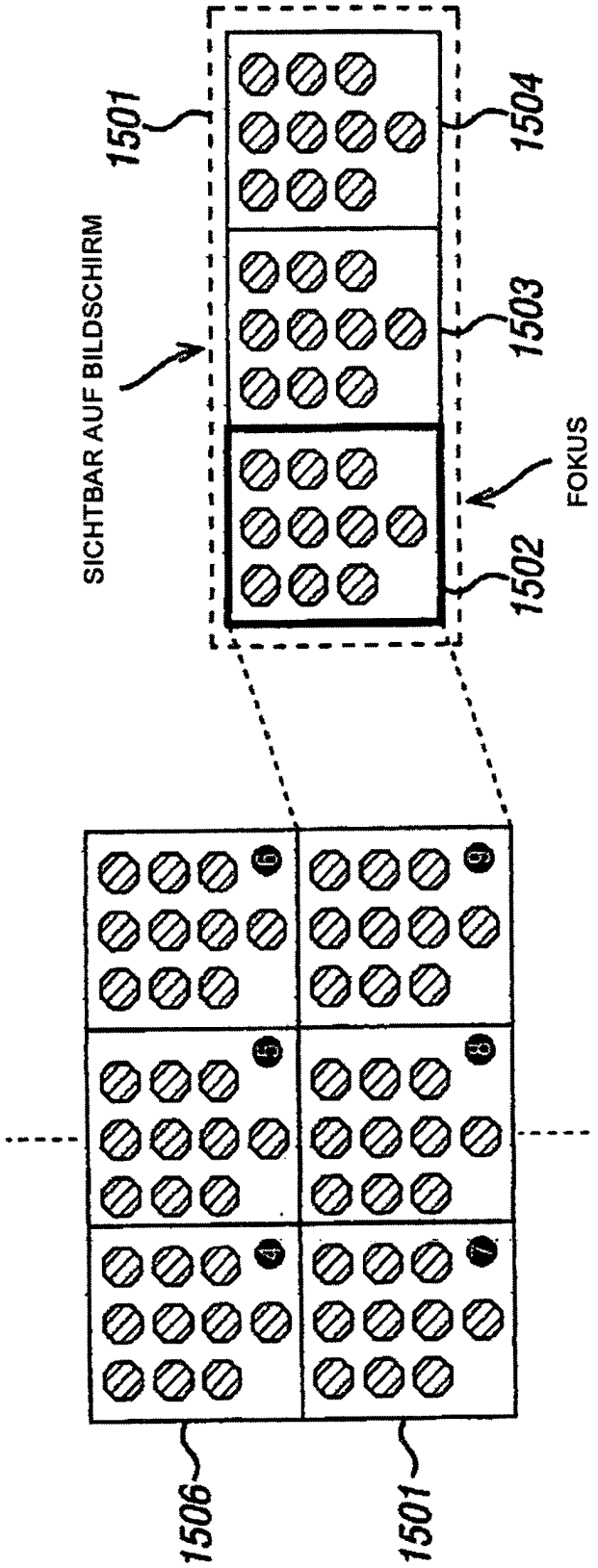

1501
1504
SICHTBAR AUF BILDSCHIRM
1503
FOKUS
1502
1506
1501

FIG. 43

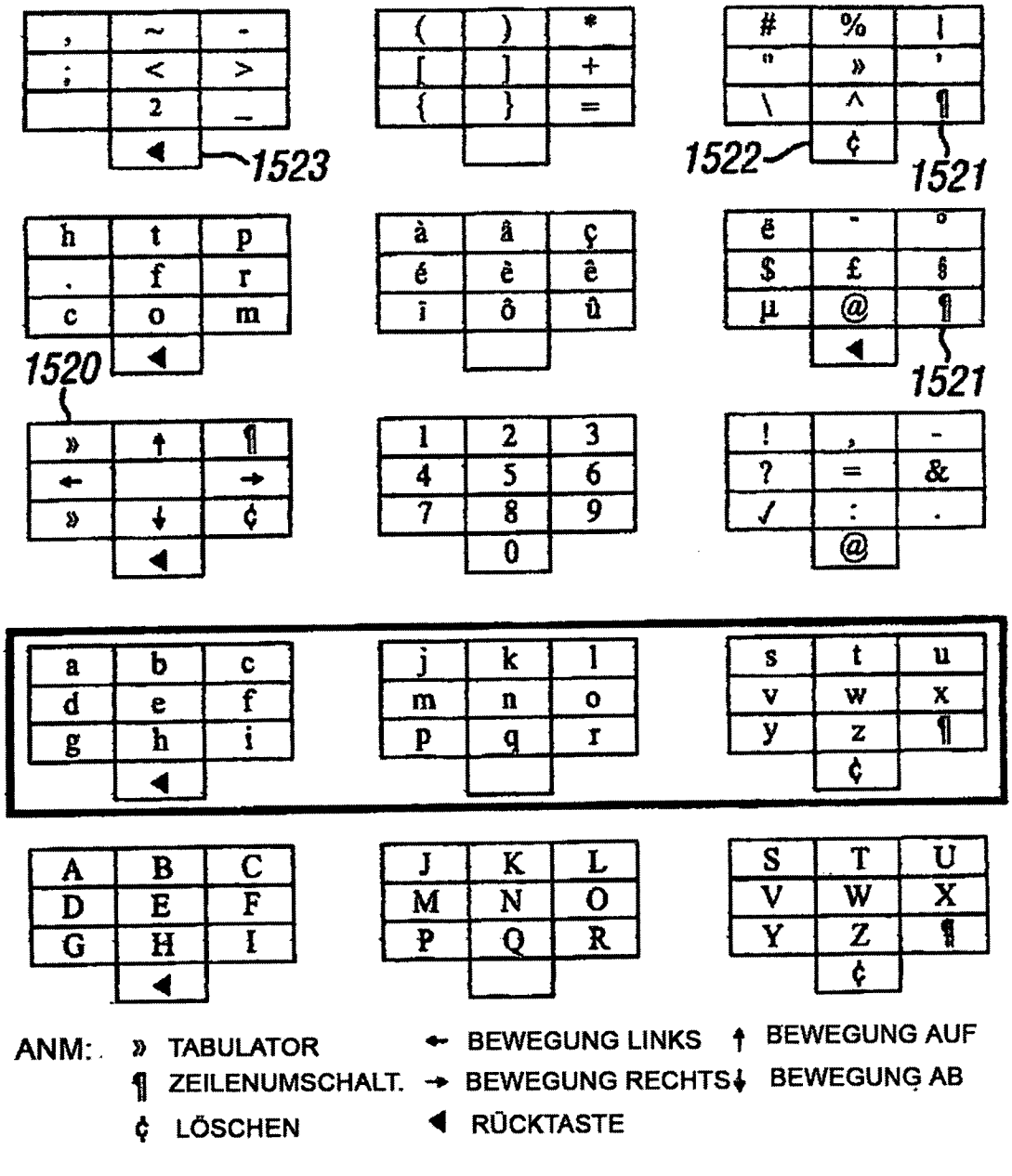

1523
1522
1521
1520
1521
ANM:
» TABULATOR
¶ ZEILENUMSCHALT.
¢ LÖSCHEN
← BEWEGUNG LINKS
→ BEWEGUNG RECHTS
↑ BEWEGUNG AUF
↓ BEWEGUNG AB
◄ RÜCKTASTE

FIG. 44

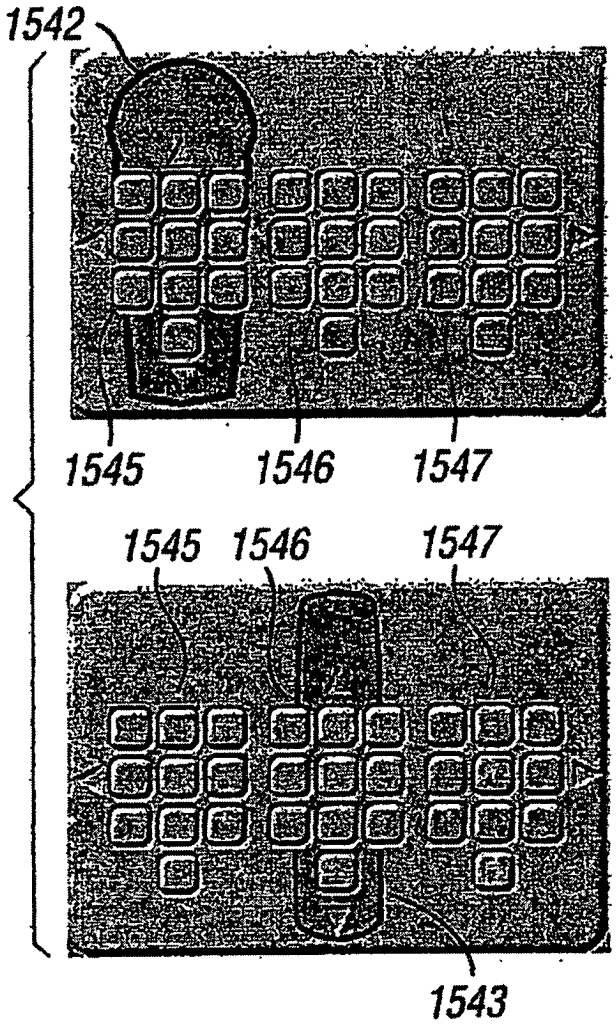


FIG. 45