[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/44 (2006.01)
G06F 3/033 (2006.01)
H04N 7/24 (2006.01)

[12] 发明专利说明书

专利号 ZL 00817972.7

[45] 授权公告日 2007 年 5 月 16 日

[11] 授权公告号 CN 1316356C

[22] 申请日 2000.11.1 [21] 申请号 00817972.7

[30] 优先权

[32] 1999.11.2 [33] EP [31] 99402721.7

[32] 2000.2.3 [33] EP [31] 00300832.3

[86] 国际申请 PCT/IB2000/001720 2000.11.1

[87] 国际公布 WO2001/033344 英 2001.5.10

[85] 进入国家阶段日期 2002.6.28

[73] 专利权人 卡纳尔股份有限公司

地址 法国巴黎

[72] 发明人 B·茹埃 E·恩吉芸范霍恩

J·-S·维莱尔

[56] 参考文献

US5732271A 1998.3.24

JP10-49327A 1998.2.20

EP0725337A1 1996.8.7

审查员 范玉霞

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 王 勇 张志醒

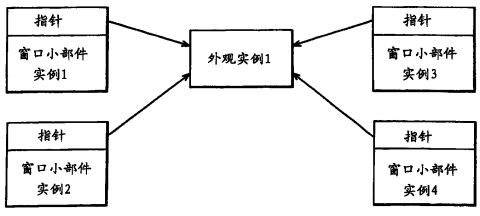
权利要求书 2 页 说明书 117 页 附图 41 页

[54] 发明名称

在一个图形用户界面中控制一个图形对象外观的方法和装置

[57] 摘要

本发明的方面涉及一种在一个图形用户界面中控制一个图形对象的外观的方法。 在本发明一个方面的一个实施例中，在一个图形用户界面中的一个诸如一个窗口小部件的对象，包含一个窗口小部件类的一个实例，在该类中定义了控制该对象操作的属性和/或方法；以及一个外观对象类的相关实例，在该类中定义了控制该对象外观的属性和/或方法。



1. 一种在一个图形用户界面中控制一个面向对象的窗口小部件的外观的方法，包含：

定义一个外观对象类；以及

将该外观对象类与该窗口小部件关联；

其中该外观对象类包含确定如何显示该窗口小部件的代码或者代码参数；以及

其中该外观对象类配置用于控制该窗口小部件的外观。

2. 如权利要求 1 所述的方法，其特征在于： 该外观对象类由面向对象的程序代码定义。

3. 如权利要求 1 所述的方法，其特征在于：该窗口小部件包含一个标识与该窗口小部件相关联的外观对象类的属性。

4. 如权利要求 1 所述的方法， 进一步包含：通过重新定义或者修改该外观对象类或者通过把该窗口小部件与一个不同的外观对象类相关联，来修改该图形对象的外观。

5. 如权利要求 1 所述的方法，其特征在于： 该外观对象类可以包含一个更新计数器，当该外观对象类被重新定义或者修改了时，就更新该计数器的值。

6. 如权利要求 1 所述的方法，还包括在一个图形用户界面中通过将外观对象类与多个窗口小部件相关联，来控制多个窗口小部件的外观。

7. 如权利要求 1 所述的方法，其特征在于：该外观对象类定义了一个图形对象的至少一个以下属性如何被显示：

背景外观；

一个背景覆盖；

该前景外观；

一个前景覆盖；

凸起或者阴影的外观；

一个对象边界的外观；

分配给该对象的任何焦点的显示；

分配给该对象的任何高亮显示的显示。

8. 一种在一个图形用户界面中控制一个窗口小部件外观的装置，

包含:

用于定义一个外观对象类的装置;

用于将该外观对象类与该窗口小部件关联的装置;以及

用于显示该窗口小部件的装置;

其中该窗口小部件是将要被显示的应用;以及

其中该外观对象类配置用于控制窗口小部件的外观。

9. 如权利要求 8 所述的装置,其特征在于:该外观对象类定义了一个图形对象的至少一个以下属性如何被显示:

背景外观;

一个背景覆盖;

该前景外观;

一个前景覆盖;

凸起或者阴影的外观;

一个对象边界的外观;

分配给该对象的任何焦点的显示;

分配给该对象的任何高亮显示的显示。

在一个图形用户界面中控制一个图形对象外观的方法和装置

本发明涉及图形用户界面。本发明的方面涉及一种用于控制在一个图形用户界面中的一个图形对象外观的方法。本发明的方面在提供一个用于诸如用于数字电视信号的接收器／解码器的设备的图形用户界面中具有一个特定的、而不是排它的应用。然而，本发明的方面还具有到通用计算机及其它设备的应用。

大多数的图形用户界面（GUI）包含一组类似的、能够由用户操作的基本部件。这些包含如按钮、滑动块、列表框、诸如此类这样的对象。这样的部件一般被称为“窗口小部件”。虽然窗口小部件的基本功能在许多 GUI 当中是共同的，但是窗口小部件的外观随着 GUI 的不同而不同。

某些图形操作系统，例如 X 窗口系统，几乎不利用在能够在 GUI 中显示的窗口小部件的外观上的约束。这允许程序员使用若干个不同窗口小部件集合开发应用程序，其中每一个都具有它自己的特殊外观。此外，大量窗口管理程序在影响由应用程序创建的窗口的整体外观的 X 下面运行。通常，在开发一个应用程序期间以及某种程度上在运行期间，有可能对显示在 GUI 上的该应用程序的外观实现某些控制。然而，在这两种情况下，该外观都由窗口小部件集合的硬编码部分或者窗口管理程序确定。对于一个用户来说，不可能显著地改变一个应用程序的外观而不用重新编码该窗口小部件集合、该窗口管理程序、或者这二者。这些选项中的任何一个都需要大量的重编码工作以及大量的新代码被安装到一个主系统上。

一种减少必须执行以更新一个外观的重编码工作的数量的方案是，从保存在一个主系统存储器中的像素映像中构造在一个 GUI 显示中一个窗口的单元（例如，边角、边界、诸如此类）。这可以作为本发明的一个独立方面来提供。然而当使用时，该像素映像能够占据相当大量的存储器，而且当该外观将被更新时，表示有大量的数据要被传送。如果 GUI 必须利用有限资源操作而且必须经由一个限制带宽的链接更新的话，则这能够是一个重要的缺点。当一个应用程序在用于数字电视信号的一个接收器／解码器上被执行时，就出现了这样一种情

况的一个实例。这样一个解码器与一台通用计算机相比较具有有限数量的存储器，而且通过从形成所接收电视信号一部分的一个频道中下载数据来更新该软件（包含 GUI 的外观）。

这个发明的一个目的是向一个应用程序开发者提供，以一种一致和可容易控制的方式、利用最少必需的重编码以及最少要求传输到一个执行环境的数据，来控制一个应用程序外观的能力。

在本发明的第一方面，提供了一种控制在一个图形用户界面中的一个图形对象外观的方法，包含:

定义一个外观对象；以及

将该外观对象与该图形对象关联。

通过明确地定义一个外观对象，而不是在一个应用程序中嵌入控制外观的代码，本发明可以允许比迄今为止控制一个图形对象外观的方法更具有灵活性。

更可取地是，该外观对象包含确定如何显示该图形对象的代码或者参数，这样的代码或者参数最好被保存在存储器中。例如，该外观对象可以由面向对象的程序代码定义。

该外观对象可以通过实例化一个外观对象类来定义。以这种方式定义的一个外观对象可以包含一个指向另一个外观对象类的指针（除了从中导出它的那个类以外）。这能够允许该外观对象访问那另一个外观类的属性和／或方法。以这种方式，一个外观对象能够从两个或更多其它外观中获取它的特征，这能够允许要被创建的外观是两个不同外观的组合，或者允许额外的特征被添加到该外观中。

为了把该图形对象与外观对象关联，该图形对象可以包含一个属性来标识与该图形对象相关的外观对象。

更可取地是，该外观对象定义了分配给特定指定颜色的实际颜色。例如，该外观对象可以定义分配给黑色、白色、和一个或者几个灰度中至少一个的实际颜色。以这种方式，例如通过向与那个外观有关的图形对象给予某个色调，该外观能够定义某个颜色方案。该外观对象还可以定义一个颜色映射，其设置在显示任何特定颜色时将被使用的实际颜色值。

为了改变该图形对象的外观，该外观对象可以被重新定义或者被修改（例如通过在编译时改变代码或者参数，或者在运行时改变参

数），或者一个不同的外观对象可以与该图形对象相关联。因此该方法可以进一步包含通过重新定义或者修改该外观对象或者通过把该图形对象与一个不同的外观对象相关联，来修改该图形对象的外观。

如果一个外观对象被重新定义或者修改了，则可能必需更新与该外观对象相关联的图形对象的外观。为了实现这个，该外观对象可以包含一个更新计数器，当该外观对象被重新定义或者修改了时，就更新该计数器的值。

更可取地是，图形对象存储它们已经影响的外观对象的更新计数器的值。每当要重新显示一个图形对象时，由该图形对象存储的值与该外观对象的更新计数器的值相比较。如果两个值不同，则该图形对象考虑在该外观对象中的修改并且存储该更新计数器的新值。

该外观对象可以包含一个验证掩码来指示能够由该外观对象调用的方法，以便使没有由该外观对象实现的方法不被调用。该图形对象能够访问该外观对象的验证掩码以最优化该图形对象的绘制。以这种方式，能够避免对没有实现的方法的调用，从而能够加速该图形对象的绘制。

在某些情况下，例如在第一次创建一个外观对象时，该外观对象可以与单个图形对象相关联。然而在最佳实施例中，该外观对象与多个图形对象相关联。通过将该外观对象与多个图形对象相关联，可以实现用于那些对象的统一外观，而且与每个图形对象都具有它独立定义的外观的情况相比，可以减少定义该外观对象所需要的数据数量。

因此，该方法可以是一种在一个图形用户界面中控制多个图形对象的外观的方法，并且可以包含将多个图形对象与该外观对象相关联。

在某些情况下，诸如没有图形对象与一个外观对象相关联时，可能期望删除该外观对象，例如以允许由该外观对象使用的存储空间被重新分配。这在诸如接收器／解码器这样、其中存储器可能有限的设备中可能是尤其重要的。为了这个目的（在其它目的当中）该外观对象可以包含一个计数器来指示与该外观对象有关联的图形对象的数目。更可取地是，每当一个图形对象与该外观对象相关联时，递增该计数器，而每当一个图形对象与该外观对象不相关，递减该计数器。如果该计数器为零值，则可以假定该外观对象能够被安全地删除。

在本发明另一个方法方面中，提供了一种在一个图形用户界面中定义一个图形对象的方法，包含：提供一个控制该图形对象外观的外观对象（例如，通过定义控制该图形对象外观的属性和／或方法），并且提供一个控制该图形对象操作的窗口小部件对象（例如，通过定义控制该图形对象操作的属性和／或方法）。上述任何特征都可以结合这个方面来提供。

如上所述的任何一种方法最好是进一步包含例如在一个诸如一个计算机屏幕或者一个电视屏幕的屏幕上显示该图形对象。

如上所述的任何一种方法可以由一个接收器／解码器、诸如一个数字或者模拟电视接收器／解码器执行。

在这里使用的术语“接收器／解码器”可以意味着一个用于接收编码或者不编码信号、例如可以由某些其它装置广播或者传送的电视和／或无线电信号的接收器。该术语还可以意味着用于解码接收信号的一个解码器。这样的接收器／解码器的实施例可以包含例如在一个“机顶盒”中与接收器集成、用于解码该接收信号的解码器，这样一个解码器与一个物理上分离的接收器一起起作用，或者这样一个解码器包含附加功能，诸如一个网络浏览器、一个录像机或者一台电视。

在本发明的一个装置方面，提供了用于控制在一个图形用户界面中的一个图形对象外观的装置，包含：

用于例如通过在存储器中创建或者修改代码或者参数、来定义一个外观对象的装置，以确定将如何显示一个图形对象；以及

例如通过设置该图形对象的一个属性来标识该外观对象、用于将该外观对象与图形对象相关联的装置。

更可取地是，该装置包含一个被适当编程的处理器用于定义该外观对象以及用于将该外观对象与该图形对象相关联，而且包含一个存储器用于存储该外观对象和图形对象。该外观对象可以由面向对象的程序代码定义。该外观对象可以通过实例化一个外观对象类来定义。该外观对象可以包含一个指向另一个外观对象类的指针。

该装置可以适于通过重新定义或者修改该外观对象、或者通过把该图形对象与一个不同的外观对象相关联，来修改该图形对象的外观。该外观对象可以包含一个更新计数器（诸如在存储器中的一个存储单元），当该外观对象被重新定义或者修改时更新该更新计数器的

值。

该外观对象可以包含一个验证掩码（例如，保存在存储器中）来指示那些能够由该外观对象调用的方法。

该装置可以是用于在一个图形用户界面中控制多个图形对象外观的装置，而且可以适于将该外观对象与多个图形对象相关联。该外观对象可以包含一个计数器（诸如在存储器中的一个存储单元）来指示与该外观对象有关联的图形对象的数目。

该装置可以进一步包含诸如一个屏幕（例如一个计算机屏幕或者一个电视屏）、用于显示图形对象的装置。

在一个实施例中，该装置是一个接收器／解码器，诸如一个数字或者模拟电视接收器／解码器。

在另外一个方面中，本发明提供了在一个图形用户界面中的一个对象，其包含控制该对象操作的一个窗口小部件类的一个实例，以及控制该对象外观的一个外观对象类的一个实例。

如上所述，本发明还提供了用于创建对象的一个窗口小部件集合，该窗口小部件集合包含多个窗口小部件类以及一个或多个外观对象类。更典型地，该窗口小部件集合包含多个外观对象类，这些类包含一个基类以及一个从该基类中导出的类。

为了修改该对象的外观，该外观对象类能够被改变。这不会改变该对象的函数，因为函数由该窗口小部件类控制。此外，有可能让该外观对象类的一个实例由该窗口小部件类的多个实例使用，由此最小化使用的存储器的数量。

该外观对象类能够被包含在一个在运行时间被链接到一个应用程序里的库中。一个用户因此能够通过选择一大堆库中的哪一个应当被链接来为该对象选择一个最佳的外观。因为该外观对象类独立于另一项被改变，所以重新编码被简化到实现该新外观所必需的最少数。

做为选择，该外观对象类能够被包含在在编译时被链接到一个应用程序里的一个库中。这允许一个应用开发者具有对该对象外观的完全控制。

该外观对象类可以导出能够被激活以在一个 GUI 显示上绘制一个组件的一部分的绘制方法。该窗口小部件类一般包含激活该外观对象类的绘制方法的代码。该外观对象类还可以导出属性，该属性提供与

一个组件的单元有关的数据。

该外观对象类可以是一个缺省类，其为在一个 GUI 显示中的窗口小部件提供一个缺省的外观。做为选择，该外观对象类可以是一个从一个缺省类中导出的类，该导出类覆盖一个或多个该缺省类的方法和／或属性。通过这种方案，一个用户或者开发者能够对在一个 GUI 的外观内仅仅有限数量的单元进行改变而不用必须重新处理任何保持不变的部分。

该外观对象类实例可以是该窗口小部件类实例的一个属性。一般地，指向该外观对象类的一个实例的指针被分配给该窗口小部件类实例的一个属性。这样一个指针可以作为一个参数传递给该窗口小部件类的一个构造器。

本发明还提供了一种在一个图形用户界面中创建一个对象的方法，其中实例化一个外观对象类，在该外观对象类中定义了控制该对象的外观对象的属性和／或方法；以及实例化一个窗口小部件类，在该窗口小部件类中定义了控制该对象操作的属性和／或方法。

在依据前面最后一段的方法中，指向该外观对象类的一个指针一般被保存在该窗口小部件类实例的实例存储器中。

通过控制操作，不是一定意味着该对象是有作用的；尽管滑动块和按钮等能够执行动作，然而某些图形对象可能仅仅是作装饰的、诸如符号或者组合对象的一部分。因此控制该对象操作的属性和／或方法只是可以确定该对象的基本功能（例如表现为一个指针），该对象的精确外观由该外观对象控制。

在另一个方面，本发明提供了一种在一个图形用户界面中创建多个对象的方法，其中实例化一个外观对象类，在该外观对象类中定义了控制该对象外观的属性和／或方法；以及一个窗口小部件类被实例化了多次，在该窗口小部件类中定义了控制该对象操作的属性和／或方法。

在另一个方面，本发明提供一种为一个应用程序执行设备（例如，一台计算机、一个用于一个数字电视信号的接收器／解码器、或者另一个设备）实现一个图形用户界面显示的方法，该方法包含加载如上所述的一个窗口小部件集合到该设备的存储器里，通过如上所述的一种方法在该图形用户界面中创建多个对象，以及在该显示上显示该对

象。

在这样一种应用于一个接收器／解码器的方法中，通过在一个数字电视信号内的一个频道中被下载，该窗口小部件集合可以被载入到该接收器／解码器的存储器中。

在另一个方面，本发明提供了一种为一个应用程序执行设备更新一个图形用户界面显示的外观的方法，其中显示包含如上所述的一个窗口小部件集合，其中至少一个外观对象类被另一个外观对象类替换。

在另一个方面，本发明提供了一种更新用于数字电视信号的接收器／解码器的一个图形用户界面显示的外观的方法，其中显示包含如上所述、被保存在该接收器／解码器的存储器中的一个窗口小部件集合，其中至少一个替换外观对象类在一个属于数字电视信号一部分的频道中被传输到该接收器／解码器以替换一个存储的外观对象类。

本发明扩展到相应的装置。特别地，本发明扩展到被安排用来执行一种任何前面方面的方法的一个接收器／解码器。

在另一个方面，本发明提供了一个用于实现一个图形用户界面的应用程序执行设备（例如，一台计算机、一个用于一个数字电视信号的接收器／解码器、或者另一个设备），该设备包含一个处理器、存储器、以及一个用于输出一个信号到一个显示器的输出端口，其中该存储器具有在其中存储的、用于执行上述任一方法、或者任一上述对象、或者任一上述窗口小部件集合的方法。

在任何上述情况中，该外观对象或者外观对象类最好定义了一个图形对象的至少一个以下属性（如果给出的话）如何被显示：

该背景外观，例如一个背景色；

一个背景覆盖图，例如一个商标；

该前景外观；

一个前景覆盖图，例如一个商标；

凸起或者阴影的外观；

一个对象边界的外观；

分配给该对象的任何焦点的显示；

分配给该对象的任何高亮显示的显示。

本发明还扩展到了一段用于执行任何上述方法的计算机程序，并

且扩展到了一个计算机可读介质，该介质中存储有用于执行任何上述方法的程序。本发明还扩展到一个应用程序，该应用程序用于一个被安排用来执行一种任何在前方面的方法的接收器／解码器。本发明还扩展到实质上和此处参考附图所描述的相同的一种方法或者装置。

显示的图形对象的设计

大多数的图形用户界面（GUI）显示各种图形对象，其中有一些图形对象能够被一个用户操作（“活动”对象）而某些不能被一个用户操作（“静态”对象）。更可取地是，如上所述，该显示的图形对象最好被设计成相辅相成以产生一个相关的设计方案。例如，在适当的地方，该显示的图形对象最好共享一种主颜色、纹理和字体。此处一个设计方案被称作该图形对象一个“外观”。

在某些情况下，只有活动图形对象的外观能够由该用户改变。依据本发明的一个方面，最好静态图形对象的外观是可改变的。依据本发明的一个方面，最好一组图形对象的外观是可改变的。因此能够获得该显示的图形对象的一个相关设计。

参考本发明的其它方面，在图形对象和链接单元被显示的地方，最好该图形对象的外观和该链接单元的外观都是可改变的。更可取地是，在由该用户选择的外观中，该图形对象的外观和该链接单元的外观是相同的。以这种方式，能够实现用于图形对象显示链的一个相关外观。

在本发明的最佳方面中，该链的外观独立于其它显示的图形对象外观是可选的。

本发明的一个方面包含一种在一个图形用户界面中控制多个图形对象的外观的方法，本方法包含将一个外观对象与多个图形对象相关联。

在本发明的一个最佳实施例中，该多个图形对象包含一个链接的链。因此更可取地是，一个特定外观或者设计能够作为一个整体应用于该链或者链表。更可取地是，该外观或者设计能够独立于任何应用于其它链、链表或者图形对象的外观或者设计，被应用于该链、链表或者图形对象。

本发明的一个方面提供了一种显示一组图形对象的方法，该方法

包含步骤：提供用于定义多个不同的可显示图形对象组以及用于从多个可显示组中选择一组图形对象的装置。

更可取的是，该方法包含：定义一组可显示的图形对象，并且使一个外观对象与该组图形对象相关联。

依据本发明，提供了一种用于控制一组图形对象的外观的方法，该方法包含定义一组可显示的图形对象。

此外提供的是一个用于控制一组图形对象的外观的装置，该装置包含用于定义一组可显示的图形对象的装置。

更可取地是，一组中图形对象的外观不同于另一组中图形对象的外观，例如在颜色、形状、尺寸、纹理或模式中的一个或多个。

更可取地是这组图形对象包括用于显示图象对象的链接链的单元。

更可取地是这组图形对象包含静态对象和活动对象。因此本装置可以包含一个图形对象组库，更可取地是具有不同的外观或者设计。

更可取地是，来自于第一组中显示的图形对象在一个或多个颜色、形状、尺寸、纹理或者模式方面不同于另一个组中显示的图形对象。在第一组中的图形对象方案可以不同于在另一组中的图形对象方案。

根据本发明的一个最佳实施例，图形对象组包含图形对象的一个链接链。更可取地是，该链接链被显示为在该 GUI 中的一个链接链。

本发明这个方面的其它最佳特征在前面在标题“显示图形对象”下面描述了。

构造图形对象

大多数的图形用户界面（GUI）包含各种图形对象，其中一些能够被一个用户操作。这样的图形对象包含窗口、对话框和按钮。该 GUI 通常包含每个图形对象的几种不同类型，例如该 GUI 可以被安排用于具有不同尺寸和／或长宽比的大量对话框的显示。

其中一个 GUI 具有，例如两种类型的对话框，一个的尺寸为另一个的一半，每一种类型的方框作为一个单独对象被订制和保存在该系统中。其中该 GUI 包含大量不同类型的对话框和／或不同类型的其它图形对象，这能够导致系统资源的低效率使用。

本发明的一个方面提供了一种在一个图形用户界面中控制一个图形对象外观的方法，该方法包含定义该图形对象的多个图形对象单元，供显示该图形对象的一个表现之用。

本发明的一个方面提供了适宜于与其它图形对象单元组合以形成图形对象的一个图形对象单元。

其中该 GUI 包含若干个不同类型的同一个图形对象，不同的类型将经常具有和该对象其它类型一样的特征。例如，不同尺寸的窗口和方框常常将具有彼此相同的边角，不同类型的方框仅仅在该方框的边的长度上有差别。通过提供用于构造该图形对象的对象单元，能够获取各种对象的公共特征的优点，以便使少量的构造块（对象单元）能够被用于创建大量不同类型的图形对象。

该图形对象可以包含，例如，一个窗口、一个对话框、一个按钮、一个滚动条、一个图标或者其它在屏幕上的对象。该图形对象可以是用户能够与它相互作用的“活动”对象，或者是“静态”的，例如背景的特征。

本发明的一个方面提供了一个包含多个图形对象单元的图形对象。

本发明的一个方面提供了一种在一个图形用户界面中显示一个图形对象的一个表现的方法，该方法包含组合多个图形对象单元以形成该图形对象。

本发明的一个方面提供了用于在一个图形用户界面中形成一个图形对象的装置，该装置包含用于组合多个图形对象单元以形成该图形对象的装置。

本发明的一个方面提供了一个用于实现一个图形用户界面的应用程序执行设备，该设备包含一个处理器、存储器、和一个用于输出一个信号到显示器的输出端口，其中该存储器中存储有一个用于实现本发明一个方面的程序。

更可取地是，该应用程序执行设备包含一个数字电视的接收器／解码器。

本发明的一个方面提供了一段用于执行本发明的一种方法的计算机程序。

一个图形对象的图形对象单元可以全部是相同的，该图形对象的

尺寸和形状由单元的数量和配置确定。

更可取地是，该图形对象包含图形对象单元的多个不同集合。更可取地是，该方法方面包含定义该图形对象的图形对象单元的多个不同集合。

一个图形对象单元集合可以包含，例如该图形对象的角、侧面或者边缘部分。例如，在该图形对象是一个对话框的地方，用于组成该方框的图形对象集合可以包含边角单元和边缘单元。例如，一个长方形方框可以包含四个边角单元和四个将这些边角单元链接在一起的边缘单元。

在每个单元集合内，可能有不同类型的单元，例如这些单元具有不同的尺寸、形状、方向和/或颜色。因此该方框的实例可以包含四种类型的边角单元和两种类型的边缘单元。

一个按钮可以包含两个侧面单元和一个取决于该图形对象将是的按钮类型而选择的中心单元。例如，一个 PLAY 按钮可以具有一个包含一个箭头符号的中心单元，一个 STOP 按钮可以具有一个包含一个正方形符号的中心单元。该 PLAY 按钮的侧面单元可以和该 STOP 按钮的侧面单元相同。

一个对象单元库还可以用来形成具有相对复杂形状的图形对象。例如，各种复杂矩阵的对话栏能够容易地从包含方框边缘、边角和交点的对象单元的选择中形成。

在该发明的一个最佳实施例中，该图形对象包含至少两个“边角”单元，其定义了该图形对象的“边角”或者“侧面”；以及多个填充单元，其中能够改变该填充单元的数量以改变该图形对象的尺寸和/或形状。因此，一个文本框最好具有四个边角单元，且多个填充单元用来填充在该边角单元之间的方框以创建期望尺寸的方框。该填充单元可以全部是相同的。在下面的实施例中，使用了不同类型的填充单元，例如中心单元以及四种类型的边缘单元，来构造方框和窗口。

更可取地是，重复该填充单元来构造所需尺寸与形状的图形对象。

更可取地是，该方法包含定义了一个固定单元集合和一个可重复单元集合。

本发明还提供了一种形成一个图形对象的方法，该方法包含提供

一个固定单元集合和一个可重复单元集合。

例如，对于一个窗口或者方框，该固定单元集合可以包含用于该方框的边角和任何装饰，那些固定单元为所有那个类别、任何尺寸的方框进行固定；该可重复单元集合用来构造所需尺寸与形状的对象。更可取地是，该图形对象的每个实例仅仅需要每种固定单元的一个。因此该可重复单元集合可以包含用于填充在该边角之间的边缘段和中心部分。

更可取地是，选择该可重复单元的尺寸以便最小化用于一个特定对象的可重复单元类型的数量。例如，更可取地是，考虑到该图形对象所需的最小尺寸，来选择该可重复单元的尺寸。因此能够减少构造每个图形对象所需要的、在该“成套设备”的存储介质中占用的存储器。

本发明还提供了一套用于一个图形对象的单元，该套包含一个固定单元集合和一个可重复单元集合。

本发明还提供了一种产生一套用于一个图形对象的单元的方法，该方法包含定义一个为该图形对象的实例所共用的固定单元集合，以及进一步定义一个可重复单元集合。

一个对象单元的尺寸可以是可调整的。因此需要更少类型的对象单元来创建大范围的对象。对于对话框的范例，边缘单元的长度可以是可调整的。以这种方式，仅仅使用四个边角单元和两种类型的边缘单元（在这种情况下是可调整长度的垂直边缘单元和可调整长度的水平边缘单元），就能够创建大量不同的方框形状。

本发明进一步提供了一种在一个图形用户界面中显示一个图形对象的一个表现的方法，该方法包含组合多个图形对象单元。

更可取地是，该方法包含显示一个固定单元集合和显示多个可重复单元。

本发明进一步提供了装置用于在一个图形用户界面中控制一个图形对象的外观，该装置包含用于定义该图形对象的多个图形对象单元、供显示该图形对象的一个表现时使用的装置。

本发明进一步提供了一套在显示一个图形对象中使用的单元，其包含一个为该图形对象的实例所共用的固定单元集合，以及一个可重复单元集合。

本发明进一步提供了一种用于在一个图形用户界面中显示一个图形对象的一个表现的装置、该装置包含用于组合多个图形对象单元的装置。

本发明在本发明到宽银幕电视的应用中具有特别的优点；和用于一个标准电视的图形对象相比，通过改变该图形对象单元的数量或者尺寸，能够容易地调整该图形对象的长宽比。

本发明还允许容易地进行在该图形对象中的变化例如修饰和装饰。一个对象单元可以包含一个要被包含在对话框上的一个装饰。使用本发明，能够创建具有不同或者相同装饰的许多不同的对象。

工具提示和菜单链

图形用户界面（GUI）通常包含各种图形对象，其中一些图形对象能够由一个用户操作（“活动”对象），而其它不能被用户操作（“静态”对象）。该图形对象，特别是活动对象，常常通过可能包含说明该图形对象是什么的图片或者图示的图标，向该用户进行标识。例如，用于执行一个 PLAY 功能的按钮可能包含一个包含一个箭头的图标。这样的图标能够提高该图形对象的视觉外观，而且有助于容易地导航该显示。

特别地，在屏幕空间非常珍贵的地方，图标是尤其有用的。一个示例是一个用于数字电视的一个接收器／解码器的 GUI，其中该图标是在屏幕上的图标，其供该电视操作之用和／或用于其它在屏幕上的功能、例如网络浏览。一个标准电视屏幕的分辨度是相对低的，而且为了易读，任何在该屏幕上显示的数据都被要求是相对大的。此外，在用户使用一台电视的时候，将比在他们使用例如一台个人计算机时坐得离屏幕更远。因此任何在该屏幕上显示的图标和文本都被要求是相对大的。

这对同时能够显示在该屏幕上的信息数量施加了限制。因此图标的使用是尤其有利的。

做出努力以选择清楚描述该图形对象做什么的图标。然而，某些用户可能不熟悉使用的图标以及可能不确定该特征表示什么。然而，让图形对象由一个文本说明在该屏幕上进行标识将是不合需要的，因为该屏幕可能变得太混乱了。

本发明的一个方面提供了一种在一个 GUI 中控制一个图形对象外观的方法，该方法包含：定义一个图形对象的第一可显示表现，以及进一步定义该图形对象的第二可显示表现，其中第二个表现可紧邻第一个表现显示。

本发明的一个方面提供了一种在一个图形用户界面上显示一个图形对象的方法，该方法包含显示一个图形对象的第一表现，以及紧邻第一个表现显示该图形对象的第二表现。

第一个表现最好包含一个图标。最好第二个表现被有选择地显示。

更可取地是，第一个表现是活动的，以便激活或者选择第一个表现，来执行该图形对象的一个动作。例如第一个表现可以是一个按钮。更可取地是，第二个表现是该图形对象的一个文本描述，例如指示该对象功能的一个词。第二个表现可以是或者可以不是活动的。

更可取地是，第二个表现仅仅是有选择可显示的。因此，更可取地，该图形对象的第二个表现不是始终在该屏幕上显示，只有当对该用户有帮助时才显示。

更可取地是，通过聚焦在第一个表现上，可显示第二个表现。例如仅仅在该用户。例如通过使用光标键移动一个焦点到第一个表现或者通过移动一个鼠标指针指向它、已经聚焦在第一个表现上时，才可显示第二个表现。

更可取地是，该图形对象的文本表示显示在一个框架内、例如在一个方框或者一个窗口内。更可取地是，该框架的设计和显示的其它图形对象的设计，尤其是同一个图形对象的其它表现，相补充或者相匹配。因此能够获得一个相关的设计方案。更可取地是，例如通过把该方框或者窗口链接到第一个表现，第二个表现被链接到第一个表现。因此该用户能够更清楚地看到该文本描述确切应用到什么。

更可取地是，该方法包含定义一个用于链接第一个表现和第二个表现的链接单元。更可取地是，该方法包含将一个外观对象与第一个表现、第二个表现、以及最好是该链接单元相关联。因此更可取地是，第一个和第二个表现以及链接单元共享例如颜色、纹理、和形状特征，以给出一个协调的外观。该外观对象可以包含多个单独的外观对象，从而能够获得一个协调的外观用于各种表现和链接单元。

本发明进一步提供了一种显示一个图形对象的方法，该方法包含显示该图形对象的一个第一表现，显示该图形对象的一个第二表现，以及显示一个把第一个表现和第二个表现链接起来的链接单元。

本发明提供了一种用于在一个 GUI 中控制一个图形对象外观的装置，该装置包含：用于定义一个图形对象的第一个可显示表现的装置，和用于定义该图形对象的第二可显示表现的装置，以及用于紧邻第一个表现显示第二个表现的装置。

更可取地是，该装置包含用于最好当检测到在第一个表现上的一个焦点时有选择地显示第二个表现的装置。

依据本发明，还提供了用于显示一个图形对象的装置，该装置包含用于显示该图形对象的第一个表现的装置，用于显示该图形对象的一个第二个表现的装置，以及用于显示一个把第一个表现和第二个表现链接起来的链接单元的装置。

在如下所述、该图形对象在一个链或者链接表中的地方，最好第二个表现是该显示的链或者链表的一部分。更可取地是，该方法包含定义图形对象的一个接链，第一个以及第二个表现可被显示为图形对象表现链的链接。

更可取地是，第二个表现在一个子链中。更可取地是该子链从第一个表现中分支出来。

本发明的一个方面提供了一个包含一个图形对象的显示，该图形对象包含一个图形对象的第一个表现，以及紧邻该图形对象的第一个表现、最好是有选择地显示的该图形对象的第二表现。

本发明的一个方面提供了用于在一个 GUI 中控制一个图形对象外观的装置，该装置包含用于定义该图形对象的第一个表现的装置以及用于定义该图形对象的第二个表现的装置，其中第二个表现紧邻第一个表现显示。

本发明的一个方面提供了一个用于实现一个图形用户界面的应用程序执行设备，该设备包含一个处理器、存储器、和一个用于输出一个信号到显示器的输出端口，其中该存储器中存储有一段用于实现本发明一个方面的程序。

更可取地是，该应用程序执行设备包含一个数字电视的接收器／解码器。

还提供了用于执行本发明的一个方法的一段计算机程序。

对于计算机及其它装置的 GUI，通常向用户提供一个鼠标或者其它指针设备用于导航该显示以及操作所显示的图形对象。然而，在某些应用中，例如在电视接收器／解码器装置以及移动电话以及个人管理器中，该用户可能不具有一个指针设备。例如，该用户可能使用一个光标小键盘来导航该显示。这样的光标小键盘通常包含上、下、左和右键。这能够困惑仅仅使用那些键来导航该显示的用户，尤其是在在该屏幕上显示的图形对象不是沿明显的线和／或列布置的地方。

本发明的一个方面提供了一种在一个图形用户界面中控制图形对象外观的方法，该方法包含定义一个图形对象链接链。

本发明的一个方面提供了一种在一个图形用户界面中控制一个图形对象外观的方法，该方法包含定义一个第一图形对象、定义一个第二图形对象以及定义一个用于链接图形对象的链接单元。

本发明的一个方面提供了用于在一个 GUI 中显示图形对象的装置，该装置包含用于显示多个图形对象单元的装置以及用于显示在图形对象之间的链接单元的装置。

本发明的一个方面提供了一个图形对象以及一个与该图形对象相关联的链接单元。

本发明的一个方面提供了一种在一个图形用户界面（GUI）中显示一个图形对象表现的方法，该方法包含显示图形对象的表现，以及显示一个链接单元的一个表现，其中该链接单元用于说明在图形对象之间的一个链接。

更可取地是，显示的链接单元从一个图形对象延伸到另一个。更可取地是，该方法包含显示图形对象表现的一个链接链。

更可取地是，该方法进一步包含定义一个图形对象子链，该子链包含来自于一个链接链的分支，更可取地是该子链是有选择可显示的。

更可取地是，该方法进一步包含在一个子链中显示图形对象，该子链最好包含该链接链的一个分支，并且更可取地是，通过聚焦在在该链接链中的一个图形对象上可显示该子链。因此可以仅仅当该用户感兴趣时，显示该子链，因此减少了显示的杂乱。

本发明的一个方面提供了一种在一个 GUI 中显示图形对象的方

法，该方法包含作为一个链接表显示多个图形对象。因此，本发明的一个方面提供了能够使用一个光标小键盘导航的图形对象。更可取地是，该显示的图形对象包含图标。更可取地是，在该图形对象之间的链接是可显示的。

该显示的链接单元可以从一个图形对象延伸到另一个以指示在它们之间的一个路径。这在用户使用光标键而不是一个鼠标来导航该显示的地方是特别有利的。然而，在某些情况下，该链接单元可能不延伸贯穿在图形对象之间的整个距离，但是仍然向该用户提供了在该图形对象之间的一个路径的指示。例如，该链接单元可以包含一个在一个图标上的箭头，以指向在该路径中的下一个图标。

本发明的一个方面提供了一个图形对象链。可以为每个图形对象提供多于一个的链接单元，以指示来自于那个对象的导航路径的一个选择。因此能够形成图形对象和链接单元的一个分支网络或树结构，该用户即使没有使用一个鼠标也可以容易地导航该网络。本发明的一个方面还提供了一个图形对象网络。

本发明的一个方面提供了一个图形对象主链以及一个从该主链中分支出的图形对象子链。更可取地是，该图形对象的表现是按钮（在这里使用的术语“按钮”更可取地是，被广泛地解释为涉及任何当被选择时执行一个功能的图形对象）。

更可取地是，多个子链是可显示的。更可取地是，通过选择该主链的一个图形对象可显示一个子链。

更可取地是该子链实质上垂直于该主链延伸出来。

根据本发明的另一个方面，在紧邻该图形对象可显示该图形对象的第二个表现的情况中，更可取地是，该图形对象的第二个表现被显示为该图形对象第一个表现的子链的一部分。因此，在该图形对象的第二个表现是一个包含该图形对象一个描述的文本框的情况下，更可取地是，该子链将在该子链中包含文本框和描述。这能够对该用户有所帮助。

依据本发明的一个方面，提供了用于显示一个图形对象的装置，该装置包含用于显示该图形对象的第一个表现的装置，用于显示该图形对象的一个第二个表现的装置，以及用于显示一个把第一个表现和第二个表现链接起来的链接单元的装置。

还提供了一个装置用于在一个图形用户界面中控制图形对象的外观，该装置包含用于定义一个图形对象链接链的装置。

还提供了用于在一个图形用户界面中控制一个图形对象外观的装置，该装置包含用于定义一个第一图形对象、定义一个第二图形对象以及定义一个用于链接图形对象的链接单元的装置。

本发明还提供了装置用于在一个图形用户界面中显示一个图形对象的外观，该装置包含用于显示该图形对象的表现的装置，以及用于显示一个链接单元的表现的装置，该链接单元用于说明在图形对象之间的一个链接。

本发明还提供了在一个图形用户界面中的一个对象，包含一个图形对象的第一个表现和一个图形对象的第二个表现，以及一个链接第一个和第二个表现的链接单元；以及在一个图形用户界面中的一个对象，包含图形对象的一个链接链；以及在一个图形用户界面中的一个对象，包含第一个图形对象、第二个图形对象以及链接第一个和第二个图形对象的一个链接单元。更可取地是，该对象进一步包含一个图形对象子链，该子链包含来自于一个链接链的分支，更可取地是该子链可有选择地显示。

本发明的一个方面包含一个用于实现一个图形用户界面的应用程序执行设备，该设备包含一个处理器、存储器、和一个用于输出一个信号到显示器的输出端口，其中该存储器中存储有一段用于执行本发明一个方面的程序。

本发明的该方面还包含一段用于执行本发明的一个方法的计算机程序。

头尾相接的触发器

大多数的图形用户界面（GUI）包含各种图形对象，其中一些能够被一个用户操作。这样的图形对象经常被显示为图标，这些图标通常包含说明该图形对象是什么的图片或者图示说明。例如，一个用于执行一个 PLAY 功能的对象可以被显示为一个包含一个箭头的图标。一个用于实施印刷的对象可以包含一个打印机的图示。这样的图标能够提高所显示图形对象的视觉外观，以及有助于容易地导航该 GUI。

做出努力以选择清楚描述该图形对象做什么的图标。使用了许多

“标准”图标，而且这些图标通常可被认出是描述“向前”、“后退”、“播放”和“停止”的图标。然而，某些用户可能不熟悉使用的图标而且将不确定该特征表示什么。

本发明的一个方面提供了一个具有显示的一个特征的图形对象，其中该特征以多种形式显示；本发明的一个方面还提供了用于改变该特征被显示的形式的装置。

更可取地是，该特征被显示的一个形式是一个图标。例如一个PLAY按钮可以是一个箭头的形式。

本发明的一个方面提供了一种在一个GUI中控制一个图形对象外观的方法，该方法包含：定义该图形对象的第一个表现以及该图形对象的第二个表现，第一个和第二个表现被有选择地显示，更可取地是可循环地显示。

本发明的一个方面提供了一种在一个GUI中显示一个图形对象的方法，该方法包含显示图形图形对象的第一表现以及随后显示该图形对象的第二表现。

更可取地是，该图形对象的第一个表现包含一个图标。更可取地是，该图形对象的第二个表现包含该图形对象的一个文本描述。更可取地是，该图形对象的显示在第一个表现和第二个表现之间循环。

本发明的一个方面提供了用于在一个GUI中控制一个图形对象外观的装置，该装置包含用于定义该图形对象的第一个表现的装置以及用于定义该图形对象的第二个表现的装置。

本发明的一个方面提供了用于实现一个图形用户界面的应用程序执行设备，该设备提供一个处理器、存储器、和一个用于输出一个信号到显示器的输出端口，其中该存储器中存储有一段用于执行本发明一个方面的程序。

更可取地是，该应用程序执行设备包含一个数字电视的接收器／解码器。

还提供了用于执行本发明的一个方法的一段计算机程序。

可以被显示的不同表现包含，例如，一个图标和该特征的一个文本描述。因此一个PLAY图标（例如一个箭头的说明）可以同时被显示为单词“播放”。因此能够获得使用该图标的视觉及其它好处，同时确保用户从该文本描述中知道该图标的意思。通过提供该图形对象

的不同表现，能够提高该 GUI 的图像外观，而且这些不同的显示不会弄乱该用户的可视区域。

在本发明的第一个实施例中，要被显示的、该图形对象的表现能够由该用户设置。例如，该用户可能熟悉使用的图标，而不需要该图标的一个文本说明。在这种情况下他可以选择仅仅显示图标。

在本发明的第二个实施例中，通过选择或者聚焦（例如高亮显示）在该显示的表现上，显示的表现能够由该用户改变。例如，如果该图形对象的表现被设置为“播放”图标，通过聚焦在第一个表现上，例如高亮显示或者选择第一个表现，用户能够把该“播放”图标改变为文本“播放”。

在本发明的第三个实施例中，不用该用户的介入，该图形对象的显示就会在不同表现之间进行切换。例如，能够设置该参数选择，以便显示的图形对象在两种表现类型之间循环。在一个最佳实施例中，该显示在图标和文字描述之间循环。循环时间可以是可调整的。循环时间的一个示例是大约 2 秒。

在本发明的一个实施例中，当显示一个表现时启动一个计时器；当该计时器达到一个预定值时，改变该表现。表现的改变可以直接从一个表现变到另一个，例如从图标变到文本描述，或者可以包含表现的一个序列。例如，表现的序列可以显示从一个表现到另一个表现的顺序改变。在一个实施例中，表现的序列给出这样的效果，即这两个表现在一个旋转硬币的两个面上。

本发明进一步提供了一种用于在一个 GUI 中控制一个图形对象外观的装置，该装置包含：用于定义该图形对象的第一个表现和该图形对象第二个表现的装置，以及用于有选择地显示第一个和第二个表现的装置，以及最好包含用于循环地显示该第一个和第二个表现的装置。

还提供了用于在一个 GUI 中显示一个图形对象的一个表现的装置，该装置包含：用于显示该图形对象的第一个表现的装置，以及用于随后显示该图形对象的第二个表现的装置。

更可取地是，该装置进一步包含用于在显示第一个和第二个表现之间显示一个表现序列的装置。

本发明进一步提供了一个在一个 GUI 中的对象，包含该图形对象

的第一个表现以及该图形对象的第二个表现，第一个和第二个表现被有选择地显示，更可取地是被循环显示。

在屏幕上的键盘

具有数字小键盘的设备被越来越多地使用来执行任务，其中这些任务涉及使用该小键盘写入包含文本的数据。例如，移动电话被用来输入信息到电话存储器里，例如输入姓名到该电话中的电话簿里，或者发送文本消息。其它的示例包含，例如当使用一个具有接收器／解码器的电视访问国际互链网时，用于输入文本到一个电视屏幕上的电视遥控器。

这样的设备通常仅仅具有对应于每一数字 0 到 9 键的数字小键盘，而且可能具有一些其它的键，例如#和*。为了能够输入文本，至少需要 26 个附加键用于字母表的字母，而且在需要大小写字母、标点与符号以及重音字母的地方需要更多的键功能。

在已知的用于文本输入的小键盘中，该小键盘的每个键都被分配了若干个功能，而且用户通过按压该键若干次循环遍历可用的功能，不同的功能通常可在一个显示屏上看见。当该用户到达该键的期望功能时，通常通过按压一个“选择”键或者通过移动该显示光标，选择该功能。将可以理解：在一全套功能可以得到的地方，小键盘的每个键必然要涉及几个不同的功能。例如，一个小键盘的“2”键可以包含以下功能：

A，B，C，a，b，c，2，Ä，ä，Á，á，Â，â，À，à，Æ，æ，Ç 和 ç

而且可以进一步包含符号功能。每当需要一个特定字母、数字或符号时，该键不得不被按压若干次。这显然能够是费时和低效率的。此外，设计限制和键的实际尺寸通常指示不是所有的、能够由一个特定键访问的字母和符号都能够向该用户显示。因此，如果该用户希望输入一个特定的字母或符号，他常常不得不反复试验使用以发现激活它的键。

用于某些型号移动电话的一个解决方案是包含一个 QWERTY 键盘，但是显然这在期望一个小键盘的地方不是理想的；该键必须保持足够大以用于使用。

本发明的一个方面提供了一个具有用于执行功能的键的小键盘，每个键都包含一个功能选项，其中该键的功能选项是可改变的。

在本发明的一个实施例中，每个键每次仅仅具有一个功能，这样该键仅仅需要被按压一次以执行那个功能。通过提供键的功能选项是可改变的，仍然有可能使用该小键盘执行比有多个键更多的功能，但是能够提高使用效率。

更可取地是，一组选项被布置为用于该小键盘的一个模板，该模板是可改变的。例如，如果该小键盘将在一个仅仅键入数字的任务中使用，如果对于每个要被键入的数字，用户不得不滚动 3 个或更多字母，这将是低效率的。通过改变该小键盘所有键的选项到“数字”，一个键仅仅需要被该用户按压一次以键入所需的数字。类似地，如果该用户希望输入大写，则“大写”模板能够被应用于该小键盘。因此只有大写字母将被包含在用于该键功能的选项中，从而减少了该用户必须滚动以找到期望的一个所需要通过的功能数目。

更可取地是，作为一个整体为该小键盘改变该键选项，但是做为选择可以仅仅改变用于某些键的选项。

本发明的一个方面提供了一种使用一个小键盘帮助输入数据到一个图形用户界面（GUI）里的方法，该方法包含定义多个模板，其中每个模板都包含用于该小键盘的一组功能。

本发明的一个方面提供了用于使用一个小键盘输入数据到一个图形用户界面里的装置，该装置包含多个模板，其中每个模板都包含用于该小键盘的一组功能。

更可取地是，这些模板在一个显示屏上显示。

更可取地是，提供了多个不同的模板，而且该方法包含选择一个模板用于使用的步骤。更可取地是，该装置进一步包含用于选择一个模板的装置。更可取地是不同的模板提供了用于该小键盘的不同功能。例如，一个模板可以为该小键盘的数字键提供大写字母 A 到 J。不同的模板可以提供大写字母 K 到 T，等等。

在这里描述的该发明的实施例中，每个模板都仅仅包含 10 个功能，供该小键盘的十个键使用。在一个不同的实施例中，该小键盘可以包含更多的键，而且有可能在该模板上表示整个字母表。例如，该小键盘可以包含一个键盘。对应于不同类型的键盘，例如不同的语言

变体、不同的数字和键排列、以及符号及其它字符，可以提供不同的模板。所有需要的功能可以被包含在每个模板上，而且提供了对应于不同类型的小键盘或者键盘的不同键布局的不同模板。

在这里使用的术语小键盘更可取地是被广泛地解释以包含所有具有一系列键或者按钮等等的设备。因此更可取地是该术语包含遥控器、电话和计算机键盘以及其它设备。

更可取地是，该方法包含，例如在一台计算机或者电视的屏幕上，显示该模板的步骤。更可取地是显示多个可用的模板，而且更可取地是用户能够选择显示模板中的一个用于使用。

在一个最佳实施例，可用的模板被显示为一个网格，而且该用户使用该小键盘的光标键来选择期望的模板。做为选择，该用户可以使用一个不同的设备，例如一个指向设备，如一个鼠标。

更可取地是，该显示器在一个对应于该小键盘相关键的排列的配置中显示该模板的功能。更可取地是，该方法包含显示一个小键盘的图像。更可取地是，选择模板的步骤包含相对于在该屏幕上的模板移动该小键盘的图像。

更可取地是，该装置进一步包含一个可显示的小键盘图像，该装置最好提供了用于相对于模板的一个显示移动该小键盘图像的装置。

更可取地是，在该屏幕上显示该小键盘图像，包含选择要被显示的小键盘图像的步骤。在一个小键盘上的键的布局能够随着小键盘的类型不同而不同，因此使在所显示模板上的键布局对应于该小键盘布局是有利的。

更可取地是，用户使用在该小键盘上的光标键相对于该模板移动小键盘的图像。

模板的功能可以被设置为 QWERTY 键盘，该键盘被分成用于该小键盘图像的各种区域。例如，一个位置中，小键盘可以用来输入字母 q、w、e、a、s、d、z、x、c 和空格。

在其它实施例中，模板功能以其它方式设置。更可取地是，布置经常由用户使用的功能以便使它们在多于一个的模板上出现。更可取地是，该模板可以由用户定制。

由该模板键盘一个键提供的功能可以包含多于一个的字母、数字或符号。例如，在该小键盘用于一个网络浏览器的情况中，一个键可

以用来输入一个文本串，例如“http:\”或者“www”。

更可取地是，在用户输入数据的同时，在该屏幕上显示该小键盘模板。

本发明的一个方面提供了一个用于小键盘的模板。更可取地是，该模板在一个 GUI 中显示。

本发明的一个方面提供了一种使用一个小键盘输入数据的方法，该方法包含选择一个包含有一组用于该小键盘的功能的模板。

本发明的该方面进一步提供了用于使用一个小键盘输入数据的装置，该装置包含用于显示一个小键盘模板的装置。更可取地是，该装置包含用于显示多个小键盘模板的装置以及用于选择一个模板的装置。

更可取地是，用于数据输入的小键盘模板的显示与任何其它该装置可以运行的应用分开。例如，如果该装置正运行一个网络浏览器应用程序并且需要在一个对话框中输入数据，则该用户调用并行于该浏览器应用程序运行的小键盘模板。更可取地是，该小键盘模板应用程序能够从在该装置中的多个应用程序中调用。

本发明的该方面还提供了一种用于实现一个图形用户界面的应用执行设备，其包含一个处理器、存储器以及一个用于输出一个信号到一个显示器的输出端口，其中该存储器中存储有一段用于执行本发明一个方面的程序。更可取地是，该存储器进一步已经在其中存储了一个小键盘模板库，而且最好是已经在其中进一步存储了一个小键盘图像库。

本发明的该方面进一步提供了一个小键盘模板和／或小键盘图像库。

本发明的一个方面提供了一段用于执行本发明的一种方法的计算机程序。

概括

在这里对图形及其它对象进行引用的地方，这种术语将被广义地解释。特别地，它将不会被解释为暗示该对象被面向对象的程序代码定义。

在这里描述的方法，更可取地是使用一个具有一个被适当编程的

处理器的应用程序执行设备来实现。

更可取地是，本发明涉及广播和接收传输。

如上所述的任何一种方法可以由一个接收器／解码器，诸如一个数字或者模拟电视接收器／解码器执行。做为选择，该方法可以由一个移动电话、一台计算机、或者另一个设备执行。

在这里描述的发明还扩展到一段用于执行任何描述的方法的计算机程序，以及扩展到一个计算机可读介质，该介质中存储有用于执行任何上述方法的程序。本发明还扩展到一个应用程序，该应用程序用于一个被安排用来执行本发明任何方面的一种方法的接收器／解码器。

本发明还扩展到计算机程序产品，例如包含计算机软件和／或计算机软件升级。这样的计算机程序产品可以，例如，通过例如电缆或者卫星广播，从一个广播中心传输到一个用户。该计算机程序产品可以在一张磁盘、例如一张 CD ROM 上提供。该计算机程序产品可以被有形地体现为，例如存储在一个存储介质诸如硬盘或易失性或非易失性存储器上的软件，或一个信号。

因此本发明的方面进一步提供了：一个计算机程序产品，包含用于执行下列步骤的代码：

定义一个外观对象；以及

将该外观对象与该图形对象关联；

一个用于在一个图形用户界面中定义一个图形对象而使用的计算机程序产品，包含：

用于提供一个控制该图形对象外观的外观对象的代码；以及

用于提供一个控制该图形对象操作的窗口小部件对象的代码；

一个用于在一种控制一组图形对象外观的方法中使用的计算机程序产品，包含用于定义多组不同的可显示图形对象、以及用于从该多组可显示对象中选择一组图形对象的代码；一个用于在一种控制一组图形对象外观的方法中使用的计算机程序产品，包含用于定义一组可显示图像对象的代码；用于在一种在一个图形用户界面中控制一个图形对象外观的方法中使用的一个计算机程序产品，包含用于定义该图形对象的多个图形对象单元供显示该图形对象的一个表现使用的代码；在一种产生一套用于一个图形对象的单元的方法中使用的一个计

算机程序产品，包含用于定义一组为该图形对象的实例所共用的固定单元、以及用于定义一组可重复单元的代码；在一种在一个图形用户界面中显示一个图形对象表现的方法中使用的一个计算机程序产品，包含用于组合多个图形对象单元的代码；在一种在一个 GUI 中控制一个图形对象外观的方法中使用的一个计算机程序产品，包含用于定义一个图形对象第一个可显示表现以及用于定义该图形对象第二个可显示表现的代码，以及用于紧邻第一个表现显示第二个表现的代码；在一种显示一个图形对象的方法中使用的一个计算机程序产品，包含用于显示该图形对象的第一个表现、显示该图形对象的第二个表现、以及显示链接该第一表现和第二个表现的一个链接单元的代码；在一种在一个图形用户界面中控制图形对象外观的方法中使用的一个计算机程序产品，包含用于定义一个图形对象链接链的代码；用于在一种在一个图形用户界面中控制一个图形对象外观的方法中使用的一个计算机程序产品，包含用于定义第一个图形对象、定义第二个图形对象以及定义用于链接图形对象的一个链接单元的代码；在一种在一个图形用户界面中显示图形对象表现的方法中使用的一个计算机程序产品，包含用于显示图形对象的表现以及用于显示一个用于说明在图形对象之间的一个链接的链接单元的表现的代码；在一种在一个 GUI 中控制一个图形对象外观的方法中使用的一个计算机程序产品，包含用于定义该图形对象第一个表现以及该图形对象第二个表现的代码，以及用于有选择地显示第一个表现和第二个表现、更可取地是循环显示这些表现的代码；在一种在一个 GUI 中显示一个图形对象的表现的方法中使用的一个计算机程序产品，包含用于显示该图形对象的第一个表现、以及用于随后显示该图形对象的第二个表现的代码；以及在这里描述的和一个接收器／解码器一起使用的一个计算机程序产品。

还提供了一种包含一个存储器和处理器的计算机程序产品，该存储器中已经存储了一个应用程序，而且该处理器（在该应用程序的控制下）适合于执行在这里描述的任何方法；一个计算机程序产品，包含一段用于执行在这里描述的任何方法的程序；以及一段用于执行在这里描述的任何方法的计算机程序。

本发明还提供了一个在其中已经存储了一段用于执行在这里描述的任何方法的程序的计算机可读介质，以及一个在其中已经存储了在

这里描述的一个计算机程序产品的计算机可读介质。

还提供了一个可有形地体现为在这里描述的一个计算机程序产品的信号。

本发明进一步提供了一种实质上在这里通过参考附图加以描述并且在附图中说明的方法，以及实质上在这里通过参考附图加以描述并且在附图中说明的装置。

在本发明涉及图形对象方面的地方，本发明提供了一种用于定义该图形对象、控制该图形对象的外观以及在一个适当的显示器、例如一个电视屏幕或者一个移动电话的 LCD 上显示该图形对象的表现的方法。本发明还提供了用于更可取地使用一个被适当编程的处理器来实现本发明以及本发明的方法的装置。

更可取地是，本发明找到了用于数字电视的应用，以及更可取地是用于一个电视的网络浏览器。本发明可以在一个例如用于一个数字电视的接收器／解码器中体现。如上所述本发明的方面的特征更可取地是由一个处理器和／或存储器、例如该接收器／解码器中的一个处理器和存储器来提供的。

在这里描述的任何方法可以由该接收器／解码器、诸如一个数字或者模拟电视接收器／解码器实现。

在这里使用的术语“接收器／解码器”可以意味着一个用于接收编码或者不编码信号、例如可以由某些其它装置广播或者传送的电视和／或无线电信号的接收器。该术语还可以意味着用于解码所接收信号的一个解码器。这样的接收器／解码器的实施例可以包含例如在一个“机顶盒”中与接收器集成、用于解码该接收信号的解码器，这样一个解码器与一个物理上分隔的接收器一起起作用，或者这样一个解码器包含附加功能、诸如一个网络浏览器、一个录像机或者一台电视。

应当理解：在这里仅仅通过举例对本发明进行了描述，而且能够在本发明的范围内对细节做出修改。

在该说明书以及（在适当处的）权利要求和附图中公开的每个特征，可独立地提供或者以任何适当组合的形式提供。

在这里该装置的特征被描述为用于一个特定功能的“装置”，意图使那些术语被广泛地解释，而且更可取地是不局限于在这里描述的本发明的任何特定实施例。在最佳实施例中，该装置的特征由一个被

适当编程的一台和多台计算机提供，而且因此该装置的特征更可取地是由包含一段计算机程序的一台计算机或者一件产品的相关特征提供。例如，该装置的特征可以由一个处理器，或者一台计算机的其它部分、例如一个存储器或者数据存储器提供。

一个方面的特征可以应用于任何其它方面；方法特征可以应用于装置方面，反之亦然。

下面将仅仅通过举例，并结合附图对本发明的最佳特征进行描述，其中：

图 1a 显示了一个典型数字电视系统的一个概述；

图 1b 显示了该交互式电视系统的总体结构；

图 2a 是一个接收器／解码器的框图；

图 2b 显示了一个接收器／解码器的结构；

图 2c 进一步说明了该接收器／解码器的结构；

图 3 是在一个窗口小部件集合内的窗口小部件的部分分级结构的一个框图；

图 4 是在一个 GUI 显示上显现的一个窗口小部件的一个简图；

图 5 说明了几个窗口小部件在存储器中的位置；

图 6a 说明了该网络浏览器的一个屏幕显示；

图 6b 显示了用于导航该网络浏览器的一个遥控器；

图 7 说明了该网络浏览器的一个进一步的屏幕显示；

图 8 说明了该网络浏览器的一个进一步的屏幕显示；

图 9 说明了该网络浏览器的一个进一步的屏幕显示；

图 10 说明了该网络浏览器的一个进一步的屏幕显示；

图 11 说明了该网络浏览器的一个进一步的屏幕显示；

图 12 说明了该网络浏览器的一个进一步的屏幕显示；

图 13 说明了该网络浏览器的一个进一步的屏幕显示；

图 14 说明了该网络浏览器的一个进一步的屏幕显示；

图 15 说明了该网络浏览器的一个进一步的屏幕显示；

图 16 说明了该网络浏览器的一个进一步的屏幕显示；

图 17 说明了该网络浏览器的一个进一步的屏幕显示；

图 18 说明了该网络浏览器的一个进一步的屏幕显示；

图 19 说明了该网络浏览器的一个进一步的屏幕显示；

图 20 说明了该网络浏览器的一个进一步的屏幕显示；
图 21 说明了该网络浏览器的一个进一步的屏幕显示；
图 22 说明了该网络浏览器的一个进一步的屏幕显示；
图 23 说明了该网络浏览器的一个进一步的屏幕显示；
图 24 说明了该网络浏览器的一个进一步的屏幕显示；
图 25 说明了该网络浏览器的一个进一步的屏幕显示；
图 26 说明了该网络浏览器的一个进一步的屏幕显示；
图 27 说明了该网络浏览器的一个进一步的屏幕显示；
图 28 说明了该网络浏览器的一个进一步的屏幕显示；
图 29 说明了该网络浏览器的一个进一步的屏幕显示；
图 30 说明了该网络浏览器的一个进一步的屏幕显示；
图 31 是由并列显示形成的一个图形对象的一个示例；
图 32 是由并列显示形成的一个图形对象的一个进一步的示例；
图 33 是由并列显示形成的一个图形对象的一个进一步的示例；
图 34 是由并列显示形成的一个图形对象的一个进一步的示例；
图 35 是由并列显示形成的一个图形对象的一个进一步的示例；
图 36 是由并列显示形成的一个图形对象的一个进一步的示例；
图 37 是由并列显示形成的一个图形对象的一个进一步的示例；
图 38 是由并列显示形成的一个图形对象的一个进一步的示例；
图 39 说明了并列显示图形对象的方法；
图 40 显示了一个典型的图像缓冲区；
图 41 显示了一个并列显示的图形对象的布局；
图 42 示意地说明了一个虚拟键盘的操作；
图 43 显示了在一个虚拟键盘上一个典型的字符到键的映射；
图 44 显示了一个虚拟键盘的示例；以及
图 45 显示了一个虚拟键盘的典型尺寸。

一个数字电视系统的概述

一个数字电视系统 1 的概况如图 1a 所示。本发明包含一个最传统的数字电视系统 2，它使用已知的 MPEG - 2 压缩系统传送压缩的数字信号。更详细地，在一个广播中心中 MPEG - 2 压缩器 3 接收一个数字信号流（通常是一个视频信号流）。压缩器 3 通过链接 5 与一

个多路复用器和扰频器 4 相链。

多路复用器 4 接收多个进一步的输入信号，组合传输流，并且经由链接 7 将压缩的数字信号传送到广播中心的一个发送器 6，当然链接 7 能够采取各式各样的形式，包括电信链路。发送器 6 经由上行链路 8 向一个卫星应答器 9 传送电磁信号，它们在卫星应答器中被电子地处理，并且通常以终端用户拥有或租用的一个反射器的形式，经由概念上的下行链路 10 传播到地面接收器 12。其它用于数据传输的传输通道当然是可能的，诸如地面广播、电缆传输、组合的卫星／电缆链路、电话网等。

由接收器 12 接收的信号被传送到终端用户拥有或租用的、并且链接到该终端用户的电视机 14 的一个集成的接收器／解码器 13。接收器／解码器 13 解码所压缩的 MPEG - 2 信号成为一个用于电视机 14 的电视信号。尽管一个单独的接收器／解码器如图 1a 所示，但是该接收器／解码器也可以是一个集成数字电视的一部分。在这里使用的术语“接收器／解码器”包含一个单独的接收器／解码器，诸如一个机顶盒，以及一台具有一个与此集成的接收器／解码器的电视。

在一个多通道系统中，多路复用器 4 处理从多个平行源接收的音频与视频信息，并且与发送器 6 相互作用以沿着相应数目的通道传播该信息。除了视听信息之外，消息或应用程序或其它任何类别的数字数据，可以被引入到某些或所有、与传送的数字音频与视频信息交错的这些通道中。

一个条件访问系统 15 与多路复用器 4 和接收器／解码器 13 相链，并且部分地位于广播中心中，部分地位于接收器／解码器中。它允许终端用户访问来自于一个或多个广播供应商的数字电视广播。一个能够解密与商业供应（即，由广播供应商出售的一个或几个电视节目）有关的消息的智能卡能够被插入到接收器／解码器 13 中。使用接收器／解码器 13 和智能卡，终端用户可以以一种认购方式或者以一种按次计费的方式购买商业供应。在这里使用的术语“智能卡”包含，但不局限于任何基于芯片的卡设备，或者具有类似功能和性能、具有例如微处理器和／或存储器的对象。在这个术语中包含的设备具有与一个卡不同的替换物理形式，例如，诸如经常在 TV 解码器系统中使用的钥匙形状的设备。

如上所述，由系统传送的节目在多路复用器 4 处被扰频，应用于一个给定传输的条件和加密密钥由访问控制系统 15 确定。以这种方式传输被扰频的数据，在计费电视系统领域中是非常公知的。通常，被扰频数据与一个用于解扰频该数据的控制字一起被传送，该控制字本身由一个所谓的使用密钥加密，并且以加密形式传送。

然后该被扰频数据和加密的控制字由解码器 13 接收，该解码器 13 能访问存储在一个插入到该解码器中的智能卡上的使用密钥的一个等效码，以解密该加密控制字，尔后解扰频发送的数据。一个缴清费用的认购者将，例如在一个每月广播一次的 ECM（Entitlement Control Message，权利控制消息）中，接收解密该加密控制字所必需的使用密钥，以便允许查看该传输。

交互系统

一个交互式系统 16，也链接到多路复用器 4 和接收器／解码器 13，而且同样部分地位于该广播中心中，部分地位于接收器／解码器中，以允许终端用户与各个应用程序经由一个调制解调的返回通道 17 相互作用。调制解调的返回通道也可能被用于在条件访问系统 15 中使用的通信。

图 1b 显示了数字电视系统 1 的交互式电视系统 16 的总体结构。

例如，该交互系统 16 允许一个终端用户通过他们的电视机从在屏幕上的目录中购买物品、根据需要查阅本地新闻和天气图片以及玩游戏。

交互系统 16 主要包含四个主要部件：

- 在广播中心或者在别处的一个创造工具 4004，用于允许一个广播供应者创建、开发、调试和测试应用程序；
- 一个在该广播中心处的应用程序和数据服务器 4006，链接到创造工具 4004 用于允许一个广播供应商筹备、验证和格式化应用程序和数据，以发送到该多路复用器和扰频器 4 用于插入到 MPEG - 2 传送流中（一般它的专用段），以广播到终端用户。
- 一个包含一个运行时间引擎（RTE）4008 的虚拟机，其是安装在由终端用户拥有或者租用的接收器／解码器 13 中的一段可执行代码，用于允许一个终端用户接收、验证、解压缩、和加载应用程序

到该解码器 13 的工作存储器中用于执行。引擎 4008 还运行驻留的、通用的应用程序。该引擎 4008 独立于硬件和操作系统；以及

• 在接收器／解码器 13 和应用程序和数据服务器 4006 之间的调制解调的返回通道 17，以在终端用户的请求下启动，指示该服务器 4006 插入数据和应用程序到 MPEG - 2 传输流中的信号。

该交互式电视系统使用控制该接收器／解码器以及在其中包含的各种设备的功能的“应用程序”来进行操作。应用程序在引擎 4008 中被表示为“资源文件”。一个“模块”是一组资源文件和数据。该接收器／解码器的一个“存储器容量”是一个用于模块的存储空间。模块可以从该 MPEG - 2 传输流中下载到接收器／解码器 13 中。

接收器／解码器

参见图 2a，下面将以功能块的方式描述接收器／解码器 13 的各个部件。

接收器／解码器 13，可以是例如，一个数字机顶盒（DSTB），包含：一个中央处理器 220，它包括相关的存储单元，而且适于从一个串行接口 221、一个并行接口 222、一个调制解调器 223（链接到图 1a 中的调制解调返回通道 17）接收输入数据；以及在该解码器的前面板上的切换触点 224。

另外，该接收器／解码器还适于经由一个控制单元 226 从一个红外线远程控制 225 接收输入，而且还拥有两个分别适于读取银行卡以及预定智能卡 242、240 的智能卡阅读器 227、228。该预定智能卡阅读器 228 与一个插入的预定卡 240 以及一个条件访问单元 229 相接合，以提供必要的控制字到一个多路分用器／解扰器 230，以便允许加密的广播信号被解扰频。解码器还包括一个传统的调谐器 231 和解调制器 232，以在被单元 230 滤波和多路分解之前接收和解调制该卫星传输。

在该接收器／解码器内的数据处理通常由该中央处理机 220 处理。图 2b 说明了该接收器／解码器的中央处理器 220 的软件体系结构。参考图 2b，该软件体系结构包含一个运行时间引擎（Run - Time - Engine）4008、一个设备管理程序（Device Manager）4068、以及多个用于运行一个或多个应用程序 4056 的设备 4062（Device）和设备

驱动程序（Device Driver）4066。

如在这个说明书中使用的那样，一个应用程序最好是一段用于控制该接收器／解码器 13 的高级功能的计算机代码。例如，当终端用户把遥控器 225 的焦点定位在在电视机 14 的屏幕上看到的一个按钮对象上、并且按下一个确认键时，运行与该按钮有关的指令序列。

一个交互式应用程序应终端用户的请求提出菜单且执行指令，并且提供与该应用程序目的有关的数据。应用程序可以是驻留的应用程序，即存储在接收器／解码器 13 的 ROM（或 FLASH（闪速存储器），或其它非易失性存储器）中，或是被广播和下载到接收器／解码器 13 的 RAM 或者 FLASH 存储器里。

应用程序被存储在接收器／解码器 13 中的存储单元内，而且被表示为资源文件。如在上述专利说明书中更详细描述的那样，该资源文件包含图形对象描述单元文件，变量块单元文件、指令序列文件、应用程序文件和数据文件。

接收器／解码器包含被分成一个 RAM 部分、一个 FLASH 部分、和一个 ROM 部分的存储器，但是这个物理结构不同于逻辑结构。该存储器可以被进一步分成与各个接口有关的存储器部分。从一个观点来说，该存储器能够被认为是硬件的一部分；从另一个观点来说，该存储器能够被认为是支持或包含除硬件之外所显示的整个系统。

软件体系结构

中央处理器 220 能够被认为是集中在一个形成一个虚拟机 4007 一部分的运行时间引擎 4008 上。这在一端（“高级”端）与应用程序相链，而在另一端（“低级”端），经由以下讨论的各种中间逻辑单元，连接到该接收器／解码器硬件 4061，其包含如上讨论的各种端口（这就是说，例如，串行接口 221、并行接口 222、调制解调器 223、以及控制单元 226）。

具体参考图 2b，各种应用程序 4057 与虚拟机 4007 相连；一些更一般使用的应用程序可以或多或少地永久驻留在该系统中，如在 4057 处指示，而其它将按要求，例如从 MPEG 数据流或者从其它端口中下载到该系统中。

除了运行时间引擎 4008 之外，虚拟机 4007 还包含，某些包括一

个工具箱 4058 在内的驻留库函数 4006。该库包含由引擎 4008 使用的以 C 语言编写的各种函数。这些包括诸如压缩、扩展或者比较数据结构等的数据操作、画线，等。库 4006 还包括有关在该接收器／解码器 13 中的固件的信息，诸如硬件和软件版本号以及可用的 RAM 空间，以及当下载一个新设备 4062 时使用的一个函数。函数能够被下载到该库中，并被保存在 FLASH 或者 RAM 存储器中。

运行时间引擎 4008 与一个设备管理器 4068 相连，该设备管理器 4068 与一组设备 4062 相链，该设备和设备驱动程序 4060 相链，这些设备驱动程序 4060 反过来与端口或者接口相链。在广义的术语中，一个设备驱动程序能够被认为是定义了一个逻辑接口，所以两个不同的设备驱动程序可以与一个公共的物理端口相链。一个设备通常将与多于一个的设备驱动程序相连；如果一个设备与单个设备驱动程序相链的话，则该设备通常将被设计为包括通信需要的所有功能，所以排除了需要一个单独的设备驱动程序的必要。某些设备可以在它们自己之中进行通信。

该接收器／解码器 13 的每个功能都被表示为在该接收器／解码器 13 的软件体系结构中的一个 4062。设备能够是本地或者远程的。本地设备 4064 包括智能卡、SCART 链接器信号、调制解调器、串行和并行接口、一个 MPEG 视频和音频播放器、以及一个 MPEG 部分和表格提取器。在一个远处执行的远程设备 4066，不同于本地设备之处在于：必须由该系统管理机构或者设计者定义一个端口和过程，而不是由一个由接收器／解码器厂商提供和设计的一个设备和设备驱动程序定义一个端口和过程。

运行时间引擎 4008 在一个微处理器以及一个公共应用程序编程接口（API）的控制下运行。它们被安装在每个接收器／解码器 13 中，以便从该应用程序角度来看，所有的接收器／解码器 13 是一样的。

引擎 4008 在接收器／解码器 13 上运行应用程序 4056。它执行交互式应用程序 4056，并且从接收器／解码器 13 的外面接收事件，显示图形和文本，调用用于服务的设备，以及使用链接到引擎 4008 的库 4006 中的函数用于具体的计算。

运行时间引擎 4008 是安装在每个接收器／解码器 13 中的一段可执行代码，并且包括一个用于解释和运行应用程序的解释器。引擎 4008

适合于任何操作系统，包含单任务操作系统（诸如 MS DOS）。该引擎 4008 基于进程定序器单元（其使用诸如一个键按压的各种事件，来执行各种动作），并且包含它自己的调度表来管理来自于不同硬件接口的事件队列。它还处理图形和文字的显示。一个进程定序器单元包含一组动作组。每个事件，取决于该事件的特性，导致该进程定序器单元从它当前的动作组移动到另一个动作组，并且执行该新动作组的动作。

引擎 4008 包含一个代码加载器来加载和下载应用程序 4056 到该接收器／解码器的存储器中。为了确保最优使用，只有必要的代码被载入到 RAM 或者 FLASH 存储器中。所下载的数据由一个验证装置进行验证以防止对一个应用程序 4056 的任何修改或者任何未被授权的应用程序的执行。引擎 4008 进一步包含一个解压缩器。因为为了节省空间和从该 MPEG 流或者经由一个嵌入接收器／解码器模式中快速下载，而压缩该应用程序代码（一种中间代码形式），所以在代码被载入到 RAM 中之前必须被解压缩。引擎 4008 还包含一个解释该应用程序代码以更新各种变量值和确定状态改变的解释器，以及一个错误检测器。

接收器／解码器结构

接收器／解码器包含五个这样组织的软件层，以便使该软件能够在任何接收器／解码器中以及利用任何操作系统实现。参见图 2c，各个软件层是应用层 250、应用程序编程接口（API）层 252、虚拟机层 254、设备层 256 和系统软件／硬件层 258。

应用层 250 包含驻留在接收器／解码器中或是被下载到该接收器／解码器中的应用程序。它们可以是由客户使用的交互式应用程序，例如用 Java、HTML、MHEG - 5 或其它语言编写，或者它们可以是由接收器／解码器使用以运行这种应用程序的应用程序。这个层基于由虚拟机层提供的一组开放的应用程序编程接口（APIs）。这个系统允许应用程序在运行时或被要求时被下载到接收器／解码器的闪存或 RAM 存储器中。能够使用诸如数据存储媒体指令与控制（DSMCC）、网络文件服务程序（NFS）或其它协议的协议，以压缩的或未压缩的格式传送该应用程序代码。

交互式应用程序是用户与其相互作用、例如以获得诸如电子程序指南、远程银行事务应用程序和游戏等的产品和服务、的应用程序。

提供了各种安全特征用于这些下载的应用程序和数据，如下所述：

- 如果没有被首先为指定网络进行验证，则不能下载任何内容到接收器／解码器，这防止了任何未登记的软件在该接收器／解码器中运行。这意味着在该接收器／解码器中运行的任何软件被识别了，而且已经被完全地检测了。
- 一个安全管理器限制应用程序对各个存储器区域的访问，因此保证了数据完整性。
- 该系统能够与任何利用安全处理器（例如，插入在接收器／解码器中的智能卡）的条件访问系统相接口。

下列驻留的应用程序被用来管理交互式应用程序：

- Boot（启动）。Boot 应用程序 260 是当接收器／解码器通电时启动的第一个应用程序。Boot 应用程序起动在虚拟机中不同的“管理程序”，其中第一个是应用程序管理程序。
- 应用程序管理程序。应用程序管理程序 262 管理在接收器／解码器中运行的交互式应用程序，即它开始、停止、中止、恢复、处理事件和处理在应用程序之间的通信。它允许多个应用程序同时运行，因此涉及它们当中的资源分配。这个应用程序对用户是完全透明的。
- SetUp（设置）。设置（SetUp）应用程序的目的是配置接收器／解码器，主要是在它被第一次使用时。它执行诸如搜索电视频道、设置日期与时间、建立用户偏爱项等动作。然而，设置（SetUp）应用程序能被用户随时使用以改变接收器／解码器的配置。
- Zapping（频道转换）。频道转换应用程序 268 被用来使用 Program-up（上一个频道）、Program-down（下一个频道）和数字键变换频道。当另一个形式的频道转换被使用、例如通过一个标题（导向）应用程序使用时，该频道转换应用程序被停止。
- 回调（Callback）。该回调应用程序用来提取保存在该接收器／解码器存储器中的各种参数的值，并且经由调制解调的返回通道

17，或者通过其它方式返回这些值到商业操作员。

API 层 252 提供了用于交互式应用程序开发的高级工具。它包括几个构成这个高级 API 的程序包。程序包提供了运行交互式应用程序所必需的全部功能。该程序包可由应用程序访问。

在一个最佳实施例中，该 API 适于运行用 Java 编程语言编写的应用程序。此外，它能够解释 HTML 及其它格式、诸如 MHEG-5。除这些解释程序之外，它还包含按照规定要求可分离和可扩展的其它程序包和服务模块。

虚拟机层 254 由语言解释器和各种模块和系统组成。它包含了在接收器／解码器中接收和执行交互式应用程序所必需的所有事物，包括下列内容:

- Language Interpreters（语言解释器）。能够安装不同的解释器以便符合将被读取的应用程序类型。这些包括 Java、HTML、MHEG－5 和其它。
- Service Information（SI）Engine（服务信息引擎（SI））。SI Engine 加载并监视公用的数字视频广播（Digital Video Broadcasting，DVB）或程序系统信息协议（Program System Information Protocol，PSIP）表，并把它们放置到一个高速缓存中。它允许需要在它们中包含的数据的应用程序访问这些表。
- Scheduler（调度器）。这个模块允许预先为空的、多线程的调度，其中每个线程都具有它自己的事件队列。
- Memory Manager（存储器管理程序）。这个模块管理对存储器的访问。当必要时它还自动地压缩在存储器中的数据，并且执行自动的垃圾收集。
- Event Manager（事件管理程序）。这个模块允许依据优先级触发事件。它管理定时器和事件抓取，并且允许应用程序彼此发送事件。
- Dynamic Linker(动态链接器)。这个模块允许解决从本地 Java 函数中产生的地址，从下载到 RAM 里的一个 Java 类中加载本地方法，并且解决从下载的本地代码到 ROM 的调用。
- Downloader（下载程序）。这个模块使用从一个远程 DSMCC

圆盘传送带或通过 NFS 协议加载的自动数据，使下载的文件以与驻留文件同样的方式被访问。还提供了存储器清除、压缩和验证。

- Class Manager（类管理程序）。这个模块加载类，并且解决任何类引用问题。
- File System（文件系统）。这个模块是紧凑的，而且被优化用以管理一个具有多个 ROM、闪速存储器、RAM 和 DSMCC 部分的分级文件系统。保证闪速存储器的完整性以不受任何事故的影响。
- Security Manager（安全管理程序）。这个模块验证应用程序，并且控制应用程序对敏感的存储器及其它机顶盒区域的访问。
- Graphics System（图形系统）。这个系统是面向对象的，而且被优化了。它包含图形窗口和对象管理，以及一个具有多种语言支持的矢量字体引擎。

此外，支持 DAVIC 资源通知模型以便有效地管理客户资源。

设备接口层 256 包含一个设备管理程序和设备。设备是包含管理外部事件和物理接口所必需的逻辑资源的软件模块。设备层管理在驱动程序和应用程序之间的通信通道，并且提供增强的错误异常检查。某些被管理的设备的示例是：卡阅读器、调制解调器、网络、PCMCIA（Personal Computer Memory Card International Association，个人计算机存储器卡国际联合会）、LED（发光二极管）显示器等等。由于 API 层从上面控制这些设备，所以程序设计员不必直接处理这个层。

系统软件／硬件层 258 由接收器／解码器的制造商提供。由于该系统的模块化，以及由于由 OS 提供的服务（诸如事件调度和存储器管理）是虚拟机的一部分，所以较高层不依赖于一个特定的实时操作系统（RTOS）或一个特定的处理器。

窗口小部件集合

在一个最佳实施例中，提供了用于在一个图形用户界面（GUI）中运行的应用程序中使用的一个窗口小部件集合。这样一个窗口小部件集合的一个特定应用是在一个数字电视接收器／解码器的一个 GUI 显示中提供窗口小部件。每个窗口小部件都被实现为一个面向对象的

模块，从而使得对于每个窗口小部件来说都有一个相应的窗口小部件类。因此，任何窗口小部件都可以通过继承或者聚集其它窗口小部件类，从更简单的组件窗口小部件中构成。

图 3 是在一个窗口小部件集合内该窗口小部件分级结构的一个简图。在这个实施例中，该窗口小部件集合包含一组基本的窗口小部件类 410，在其它当中包含窗口和对话框框架、一个滑动块控制、一个按钮、一个复选框、一个文本域、以及一个文本编辑框。在下一个复杂等级中，有组合几个基本的窗口小部件类或者修改一个基本窗口小部件的行为的类 420。例如，诸如一个列表框的窗口小部件可以从一个文本编辑框类中创建可编辑的列表项，并且允许一个用户使用从一个滑动块控制类中导出的一个滚动条来滚动该列表。在还有一个较高复杂等级中，该窗口小部件集合包含诸如一个文件选择对话框的聚集窗口小部件 430，该文件选择对话框包含按钮、可滚动列表、文本区域以及文本编辑框，所有这些在该窗口小部件集合的其它类中定义。

每一窗口小部件类都实现了方法和事件处理程序，以控制该窗口小部件的操作。该窗口小部件类也可以包含用于绘制该窗口小部件某些部分的方法。然而，为了为该窗口小部件提供一个特定外貌或者"外观"，该窗口小部件类调用与该窗口小部件类相关的一个外观对象类的绘制方法。这将在下面进行进一步的详细描述。

外观类的公共方法以及 API

为了让该外观对象类和窗口小部件类能够相互作用，有必要让该外观对象类具有一个一致的公共方法集合，该方法集合保证可以由该窗口小部件类使用。特别地，该外观对象类必须提供一个标准的 API，其包含该窗口小部件类能够调用以便在一个 GUI 显示上绘制它本身的方法。

由窗口小部件使用的 API 在一个基类中定义，从该基类中导出所有的外观类。该 API 包含以下单元:

1.一般的显示方法

2.特定的显示方法

3.创建和破坏实例的控制

4.边界控制

5.修改控制

一般的显示方法是那些对于所有窗口小部件都可以用的方法，而特定的显示方法是某些类型的窗口小部件所特有的方法。

使用一个分层体系结构构造外观。通过从导出它的类中继承属性、方法和缺省值，然后增加新的属性、方法和缺省值，或者覆盖那些继承的东西中的某些或者全部，来创建一个新的外观类。

一个外观类被组织为一个包含指向公共方法的指针表格。从另一个外观类中导出的一个外观类可以因此通过改变相关的指针以便指向另一个方法而重新定义一个方法。一般地，一个外观类仅仅实现一些可用的公共方法。

在一个实现中，提供了称为伪方法的公共方法，其仅仅返回这个调用。如果一个没有被该外观类实现的方法被该窗口小部件调用了的话，则调用一个伪方法。这样做是为了确保即使当该方法事实上没有被实现时，方法调用的无错误功能。调用伪方法的一个缺点是在调用一个无所作为方法中的时间浪费。

在另一个实现中，外观类具有验证掩码。一个验证掩码定义哪个方法能够被该外观类调用，而使得没有被实现的方法不被调用。一个窗口小部件能够访问一个外观类的验证掩码以优化该窗口小部件的绘制。在这种情况下，该窗口小部件知道没有被实现的方法，因此该窗口小部件将避免产生到这些方法的调用。以这种方式，有可能防止在调用伪方法中浪费的时间。

一个外观类可以从两个或更多其他类中导出（多种继承）。这能够允许创建一个是两个或更多其他外观类的结合的外观类。如上所述，当创建一个外观类时，它采纳了在从中导出它的外观类中的属性、方法和缺省值。为了实现多种继承，该外观还包含一个或多个指向从中导出它的属性、方法和缺省值的附加类的指针。该外观能因此访问那些属性、方法和缺省值，而不用不得不复制或者创建它们本身。

在另一个实施例中，当创建一个外观类时，它采用了在从中导出它的所有外观类中的属性、方法和缺省值。

多种继承的原理在要设计不标准窗口小部件的情况下也是有用

的，其可能要求该外观实现非标准的方法。在该外观中的一个指针能够指向包含显示该窗口小部件所需要的非标准方法的一个第二外观类。

确保从中导出一个外观的各种外观类不相互冲突是重要的。这能够通过保证该附加外观类仅仅包含不在从中导出该外观的主外观类中的方法，或者向各个类给定一个优先级次序，来完成。

一个外观对象类的公共方法的示例如下所示。

```
/* 实例初始化 */

MhwWgtLookInitDefault (MhwWgtLookclass* MhwWgtLookAtts*);
MhwWgtLookInitClass (Void);
MhwWgtLookResetDefault (MhwWgtLookclass*);

MhwWgtLookAttsGetDefault (MhwWgtLookClass*, MhwWgtLookAtts*);
MhwWgtLookAttslnit (MhwWgtLookAtts *);

/* 检索和设置边界尺寸 */

MhwWgtLookAttsGetBorderwidthBottotm (MhwWgtLookAtts *, Card8 *);
MhwWgtLookAttsGetBorderWidthLeft (MhwWgtLookAtts *, Card8*);
MhwWgtLookAttsGetBorderWidthRight (MhwWgtLookAtts *, Card8*);
MhwWgtLookAttsGetBorderWidthTop (MhwWgtLookAtts *, Card8 *);

MhwWgtLookAttsSetBorderwidthBottom (MhwWgtLookAtts *, Card8);

MhwWgtLookAttsSetBorderWidthLeft (MhwWgtLookAtts *, Card8);
MhwWgtLookAttsSetBorderwidthRight (MhwWgtLookAtts *, Card8);
MhwWgtLookAttsSetBorderWidthTop (MhwWgtLookAtts *, Card8);


/* 检索和设置颜色*/
```

```
MhwWgtLookAttsGetColorBackground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorBlack (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorDarkGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorForeground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorHighlight (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorLightGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorMapAndVisual     (MhwWgtLookAtts*,
MhwWgtColorMapId*, MhwWgtVisual*);
MhwWgtLookAttsGetColorMiddleGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsGetColorTransparent     (MhwWgtLookAtts*,
MhwWgtColor*);
MhwWgtLookAttsGetColorVeryLightGray     (MhwWgtLookAtts*,
MhwWgtColor*);
MhwWgtLookAttsGetColorWhite (MhwWgtLookAtts*, MhwWgtColor*);

MhwWgtLookAttsSetColorBackground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorBlack (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorDarkGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorForeground (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorHighlight (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorLightGray (MhwWgtLookAtts*, MhwWgtColor*);
MhwWgtLookAttsSetColorMapAndVisual     (MhwWgtLookAtts*,
MhwWgtColorMapId, MhwWgtVisual);
MhwWgtLookAttsSetColorMiddleGray (MhwWgtLookAtts*, MhwWgtColor);
MhwWgtLookAttsSetColorTransparent (MhwWgtLookAtts*, MhwWgtColor);
MhwWgtLookAttsSetColorVeryLightGray     (MhwWgtLookAtts*,
MhwWgtColor);
MhwWgtLookAttsSetColorWhite (MhwWgtLookAtts*, MhwWgtColor);
```

/* 检索和设置继承数据 */

```
MhwWgtLookAttsGetHeritageData1 (MhwWgtLookAtts*, Void**);
MhwWgtLookAttsSetHeritageData1 (MhwWgtLookAtts*, Void*);
```

/* 构造器 */

```
MhwWgtLookNew (MhwWgtLookAtts*);
```

/* 析构器 */

```
MhwWgtLookDelete (anObject)
```

/* 标准 API */

```
MhwWgtLookDrawAnchor (anObject, aWidget, aX, aY, aW, aH, aText,
aLength, anAscent, aState)
MhwWgtLookDrawBackground (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookDrawCheckSymbol (anObject, aWidget, aX, aY, aW, aH,
aState, aSymbol)
MhwWgtLookDrawChoiceSymbol (anObject, awidget, aX, aY, aW, aH)
MhwWgtLookDrawCross (anObject, awidget, aX, aY, aW, aH)
MhwWgtLookDrawCursor (anObject, aWidget, aX, aY, 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,awidget1 aX, aY, aW, aH)
MhwWgtLookDrawIterm (anObject, aWid9et, aX, aY, aW, aH, aText,
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)
MhwWgtLookGetClassId (anObject)
MhwWgtLookGetClassName (anObject)
MhwWgtLookGetItemBorderWidth (anObject)
MhwWgtLookGetMethodMask (anObject)
MhwWgtLookGetPreferredSizeArrow (anObject)
MhwWgtLookGetPreferredSizeCheck (anObject)
MhwWgtLookGetPreferredSizeChoice (anObject)
MhwWgtLookGetPreferredSizeCross (anObject)
MhwwgtLookGetUpdateCounter (anObject)
MhwWgtLookIsInstanceOf (anObject, aClassId)
MhwWgtLookReDrawItem (anObject, awidget,aX1 aY, aW, aH, aText,
aLength, anAscent, aState)
MhwWgtLookRef (anObject)
MhwWqtLookSetBorderWidth (anObject, aBorder, aWidth)
MhwWgtLockUnDrawCross (anObject, aWidget, aX, aY, aW, aH)
MhwWgtLookUnDrawCursor (anObject, awidget, aX, aY anAscent, aH)
```

```
MhwWgtLookunDrawFocus (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)
MhwWgtLookGetColorLightGray(anObject)
MhwWgtLookGetColorMap(anObject)
MhwWgtLookGetColorMiddleGray(anObject)
MhwWgtLookGetColorTransparent(anObject)
MhwWgtLookGetColorVeryLightGray(anObject)
MhwWgtLookGetColorWhite(anObject)
MhwWgtLookGetColorForeground(anObject)
MhwWgtLookGetColorHeritageData1(anObject)
MhwWgtLookGetColorHeritageLink1(anObject)
```

创建和显示一个窗口小部件

当一个应用程序要求在一个 GUI 的显示上显现一个窗口小部件时，它必须执行的第一个任务是构造该窗口小部件类的一个实例。在创建该窗口小部件实例期间，一个外观类实例变得与该窗口小部件类实例相关联。如下所述选择该特定外观:

1.如果一个外观类实例由该应用程序传递到该构造器，则使用它。

2.否则，如果有一个缺省外观的话，则使用该为正创建的窗口小部件类指定的缺省外观。

3.否则，如果有一个缺省外观的话，则使用为该窗口小部件环境指定的缺省外观。

4.否则，使用用于该窗口小部件集合的缺省外观。

一旦已经实例化了该窗口小部件类，则该应用程序能够激活它的公共方法中的适当一个以显示它。

该窗口小部件类更可取地是还向能够被调用的一个公共方法提供一个指向一个外观类实例的指针，而且因此外观类实例变得与该窗口小部件类实例相关联。这导致该窗口小部件的外观根据该新关联的外观类实例而改变。应当理解:“关联”，实际上只不过是设置在该窗口

小部件类实例中的一个字段的值。在最简单的实施例中，仅仅通过做出对该字段的一个赋值就能够进行该窗口小部件与一个不同外观类的关联。（然而，参见在下面有关存储器管理和方法 MhwWgtXxxSetLook 的注释）此外，许多窗口小部件类实例可以与一个外观类实例相关联。这在图 5 中图解说明了。

当调用一个窗口小部件类方法，以在一个 GUI 显示上绘制该窗口小部件时，它按以下顺序构造该窗口小部件的图像:

1.该窗口小部件的背景（例如，一个背景色或者一幅图像）。

2.背景覆盖（例如，一个商标）

3.该窗口小部件的前景

4.前景覆盖（例如，一个商标）

5.该窗口小部件边界。

6.突出显示

7.输入焦点。

对于一个给定的窗口小部件，某些部分可能是不存在的。该窗口小部件一部分的存在与否取决于以下条件。

1.硬编码。一些没有为窗口小部件的某些类定义的部分。

2.可选部分。例如，借助于该窗口小部件的一个公共属性，可以随意禁止焦点、凸起和突出显示。

3.外观定义。一个外观能够省略一个或多个部分。

在一个典型实例中，执行以下步骤。

首先由该窗口小部件类方法本身绘制该窗口小部件的背景，例如，通过涂抹一个背景色、一个背景模式或者一幅图像来完成。然后通过激活关联的外观实例的公共方法 MhwWgtLookDrawBackground、指定该窗口小部件在该显示上的高度、宽度和位置的适当参数，来绘制该背景覆盖。因此由该外观，例如通过覆盖一个商标来修改该背景外观。

该窗口小部件类方法然后必须构造该窗口小部件的前景；这就是说，它创建实际上由一个用户操作或者当使用该窗口小部件时显示信息的图像对象。例如，该窗口小部件可以实现一个复选框，而在这样情况下它激活该外观类方法 MhwWgtLookDrawCheckSymbol。该外观然后可以例如通过覆盖一个商标，来修改前景。

然后绘制该窗口小部件的边界区域，如将在下面描述的那样。

如果该窗口小部件类确定在该窗口小部件中的一个对象具有输入焦点，则它激活该外观类方法 MhwWgtLookDrawFocus 以在该显示的窗口小部件中指示这个。类似地，如果部分该窗口小部件被突出显示，则该窗口小部件类调用外观类方法 MhwWgtLookDrawHighlight。

窗口小部件边界管理

该外观在一个 GUI 显示上控制该窗口小部件外观的方法的一个具体示例是边界的管理。在图 4 中显示了在一个 GUI 显示上，具有边界处于它最一般形式的一个窗口小部件的外观。窗口小部件 500 在一个 GUI 显示中占据一个矩形区域。由该窗口小部件占据的区域包含两个区域：一个由一个边界区域围绕的内部操作区域 510。

该边界区域一般不对该窗口小部件的功能有所帮助（虽然它可以，有时候由一个用户使用来移动和／或重新调整该窗口小部件大小）。因此，根据一个用户的希望，该边界区域的外观变化有相当大的范围。例如，颜色，宽度、背景模式全部能够被选择以吸引一个用户或者提供一个一致的外观。因此，绘制该边界的职责被给予该外观类。

该外观维护四个尺寸以指定该边界宽度。这些指定从该操作区域 510 的左边、右边、顶端和底端到该窗口小部件边缘的距离。这些尺寸分别在图 4 中以 L、R、T、和 B 指示。这些尺寸值在缺省的外观中指定。一个应用程序可以定义一个从该缺省外观类中导出的类，在该类中覆盖这些值以创建一个外观，该外观创建具有一个不标准宽度边界的窗口小部件。一个应用程序（例如，一个外观管理器）还能够在运行时间通过调用外观方法
MhwWgtLookAttsSetBorderwidthBottom，
MhwWgtLookAttsSetBorderWidthLeft，
MhwWgtLookAttsSetBorderwidthRight，或者
MhwWgtLookAttaSetBorderWidthTop 来改变该些值。

在该外观类中，具有依据由该窗口小部件类传递到该外观类的值，控制一个边界的详细布局的代码。

窗口小部件颜色管理

一个外观还包含一个颜色定义，以便使与一个特定外观实例有关的任何窗口小部件实例将使用在那个外观实例中定义的颜色。在一个实施例中，一个外观定义了以下颜色：

- 黑色
- 暗灰色
- 中灰色
- 浅灰色
- 非常浅灰色
- 白色
- 透明
- 突出显示颜色

当绘制一个窗口小部件时使用在该外观中的颜色定义。例如，如果当显示一个窗口小部件时将绘制一条黑线，则将使用已经被定义为“黑色”的颜色。例如，如果该外观将具有一个红色调，则“黑色”可能定义为暗红，而“白色”可能被定义为亮粉红色，并且两者之间定义红色的各种色调。在这个示例中，通常将绘制一条黑线的一个绘制操作将被代之以绘制一条暗红线，等等。

此外，该外观定义了一个颜色映射，其设置了当在一个 GUI 的显示上显示任何特定颜色时要被使用的实际色值。

创建一个修改的窗口小部件

假如该缺省外观的方法 MhwWgtLookDrawCheckSymbol 绘制了一个矩形方框，该方框取决于它的状态是空的或者包含一个小的点符号，且这个在一个 GUI 中定义了一个正常复选框的外观。现在假定要求在其中显示一个点或者一个交叉的另一个窗口小部件。由于该外观由该外观类完全控制，所以仅仅需要改变该外观类。此外，实现这个行为的一个新外观类能够从存在的外观类中导出，并且提供仅仅一个方法 MhwWgtLookDrawCheckSymbol 来覆盖在该基外观类中同名的方法。此外，当用参数 aState 设置为真来调用方法 MhwWgtLookDrawCheckSymbol 时，能够调用基类的方法 MhwWgtLookDrawCheckSymbol 来绘制一个点。需要编写的新代码

仅仅需要处理其中将要绘制一个交叉的情况。因此能够用最少的编程努力来创建一个新的窗口小部件。

应当理解：这个过程不改变原有复选框窗口小部件的外观；这个窗口小部件使用没有被修改的基本外观类。为了在一个应用程序中对原有复选框窗口小部件的外观实施一个改变，必须修改在该基本外观类中的方法 MhwWgtLookDrawCheckSymbol。在该外观类和该应用程序链接的下一个时机（在编译时、或者运行时、看情况而定）之后，从那个类导出的全部复选框窗口小部件然后将改变它们的外观。

该窗口小部件类原理上将协作以创建一个窗口小部件，该窗口小部件具有任何具有一个适当公共方法和属性组的类作为一个外观类。然而，从尽可能小数量的公用基类，以及最理想是只有一个基类，中导出全部外观类是有利的。那些熟悉面向对象编程的人将理解这将最小化由该外观类使用的存储器及其他资源。一个导出的类具有一个指向它的基类的指针，以便它能够访问该基类的方法代码和静态资料而不用在存储器中复制这样的代码或者数据。

版本控制

可能某些窗口小部件实例具有一个非常长的寿命。例如，一个窗口管理程序的根窗口、在一个工作站显示屏中的任务条窗口小部件、等等。在这样的情况下，该外观类在该窗口小部件的生命期间被更新具有一个强烈的可能性。当这发生时，必须导致该窗口小部件类被本身重新绘制。

一种实现这个的方法是向每个外观类给定一个更新计数器，其作为一个公共属性导出或者可通过一个公共方法进行访问。当实例化一个窗口小部件类时，该窗口小部件类实例查询关联外观的更新计数器，并且在该窗口小部件类实例的本地存储器中存储该更新计数器的值。如果该外观类的实例被随后更新了，通过比较保存在它的本机存储器中的值和在该外观类实例中更新计数器的值，该窗口小部件类实例能够检测到这个改变。如果该外观类实例已经被更新了，则该窗口小部件能因此使用该外观类的方法重新绘制它本身

构造和析构外观类实例

通常，每个外观类的实例要比每个窗口小部件类的实例少。在某些情况下，在一个应用程序中，可能仅仅只有一个外观基类实例被全部窗口小部件类的实例所引用。还有可能有导出的外观类的实例被一个应用程序的某些窗口小部件类实例所引用。然而，一个窗口小部件类不能假定在该窗口小部件类正在实例化的时候总是存在一个外观类实例；该窗口小部件实例可能是第一个要求与一个特定外观类关联的实例。

因此，建议在实例化每个窗口小部件类期间，该窗口小部件类构造器激活关联外观类的构造器 MhwWgtLookNew。如果没有该外观类的实例存在，则创建一个新的实例。然后值 1 被保存在一个引用计数器中，该计数器保持在该外观类实例的局部存储器中。如果已经存在了该外观类的一个实例，则该外观类构造器返回一个指向它的指针，并且递增该引用计数器。

在析构每个窗口小部件类实例期间，该窗口小部件类的析构器调用用于关联外观类实例的析构器 MhwWgtLookDelete。该析构器 MhwWgtLookDelete 递减该引用计数器。如果该计数器保持大于零，则该析构器简单地返回。然而，如果该析构器达到零，则没有窗口小部件类实例（除了那个进行析构的以外）与那个外观类实例相关联，而在这样情况下该外观类析构器继续从存储器中删除该外观类实例。

能够调用该窗口小部件类方法 MhwWgtXxxSetLook 来改变与一个特定窗口小部件类实例关联的外观。在这个方法中，首先进行对不要的外观类实例的析构器的调用，并且然后进行一个对该新外观类的引用函数的调用以获得一个指向一个类实例的指针。这个确保该外观类的引用计数器被正确地更新。

还必须规定，即使已经存在一个实例还要创建一个外观类的一个新实例。这允许一个应用程序具有任何给定外观类的多于一个实例，并且在不同的实例中设置不同的属性。例如可能有同一个外观类的两个实例，它们在所有方面都是相同的，除了一个具有根据它们的缺省值保持全部属性，而另一个具有分配给它的一个或多个属性的不同值（边界宽度、颜色、等等）。

外观管理程序

应当理解，外观类和实例的系统允许非常详细地控制一个应用程序的整体外观。例如，任何外观的仅仅一个属性能够被改变，以对一个应用程序的外观实施一个较小的改变。因此，可以提供一个外观管理器应用程序以允许一个用户按要求改变这些属性。这样一个应用程序一般包含一个 GUI 显示，该显示包含体现本发明的窗口小部件，以便该用户能够立即看见改变在该窗口小部件的外观上的外观属性的效果。

网络浏览器

现在将参考附图描述该国际互链网导航界面。

图 6a 显示了一个国际互链网浏览器的主要屏幕导航显示屏的一个屏幕截图。该主要屏幕显示了一个包含该主菜单的垂直链 1100，其中该主菜单包含作为一个链接表的各个按钮。按钮通过该链的链接单元进行链接。在图 6a 的链 1100 中显示的按钮包含：RELOAD STOP 按钮 1110、PREVIOUS 按钮 1120，NEXT 按钮 1130、HISTORY 按钮 1140、BOOKMARK 按钮 1150、SETTINGS 按钮 1160 以及 QUIT 按钮 1170。

该主菜单链 1100 被安排以附加在要在屏幕 1101 上显示的超文本标志语言文件（HTML）上。在图 6a 中，没有 HTML 文件被显示，而且除了该主菜单链 1100 之外，屏幕 1101 为空。

该网络浏览器包含几个能够由该用户设置的首选项。该浏览器包含一个设备用于设置几个用户简档表。

该用户具有一个控制器，利用该控制器他能够在在该屏幕 1101 上的对象之间定位，突出显示对象并且选定对象。在本示例中，使用的控制器是一个电视遥控装置 1180。数字键 1181 用来输入数据；光标小键盘 1182 用来沿着该屏幕导航。光标小键盘 1182 包含一个 UP 键 1183、一个 DOWN 键 1184、一个 LEFT 键 1185 和一个 RIGHT 键 1187。该光标小键盘 1182 还包含一个 SELECT 键 1186 用来在该屏幕上选择对象。

UP 键 1183 和 DOWN 键 1184 用来移动一个焦点，在这个示例中是在链 1100 中上下移动的突出显示，以有选择地突出显示按钮 1110、1120、1130、1140、1150、1160 和 1170。当一个按钮被突出显示时，

它能够使用选择键 1186 被选择。

当显示一个 HTML 页时，遥控器的任何按钮调用链 1100（工具条）。链 1100 还能够由该用户打开和关闭。在一个设置选项中，当显示一个新 HTML 页时自动地隐藏链 1100，当用户想要移动到另一个 HTML 页时，他调用链 1100。

图 7 显示了具有一个 HTML 文件打开的图 6a 的屏幕，有关该打开文档的信息在一个在该链中、链接到该 RELOAD / STOP 按钮 1110 的文本框 1112 中给出。将看到在按钮之间的链链接 1114 向该用户可视地指示：他能够沿链接的方向在按钮之间移动。

图 6a 显示了突出显示的 RELOAD / STOP 按钮（该突出显示的 RELOAD / STOP 图标是在一个黑暗背景上的白色，而不是如在图 7 中它没有被突出显示时所示，在一个白色背景上的暗色）。当该 RELOAD / STOP 按钮被突出显示时，能够通过按下选择键 1186 来重新加载该 HTML 文档。

该用户使用 DOWN 键 1184 沿着链 1100 向下移动该突出显示。在图 7 中，现在 PREVIOUS 按钮 1120 被突出显示。图 8 显示了，当突出显示在 PREVIOUS 按钮 1120 上时，包含一个文本框 1122 的"工具提示"如何能在该屏幕出现。在本示例中，一旦相关的图标被突出显示时，就显示该工具提示。能够设置这个首选项以便当该按钮被突出显示时，在一个延迟之后显示该工具提示。文本框 1122 包含词"前面"，以指示 PREVIOUS 按钮 1120 的功能。通过激活该 PREVIOUS 按钮，以及通过按下 SELECT 键 1186，浏览器转移到查看过的前一页。

在图 9 中，该突出显示向下移动到 NEXT 按钮 1130，并且在一个短时间之后，显现一个包含一个文本框 1132 的工具提示以帮助该用户，该文本框 1132 包含单词“下一个”。

在图 10 中，HISTORY 按钮 1140 被突出显示并且显现具有单词“历史记录”的相关工具提示 1142。HISTORY 按钮 1140 具有多于一个的功能，因此通过按下在该控制小键盘上的 SELECT 键激活该按钮，以导致显现一个子链 1144，其给定与该历史记录功能有关的进一步选项。在图 11 中显示了子链 1144。子链 1144 包含附加的按钮，这些按钮包括一个 VIEW 按钮 1146 和一个 ADD 按钮 1148。该用户使

用 RIGHT 和 LEFT 键 1187、1185 沿着子链 1144 移动。在图 11 的屏幕显示中，突出显示 VIEW 按钮，并且显现一个工具提示 1147 以告诉该用户突出显示的按钮是 VIEW。应当注意到，附着于主链 1100 的工具提示显示在链 1100 的右手侧；用于子链的工具提示显示在该子链的上面。

用于该工具提示的方框尺寸与要被显示的一个或者多个单词的长度相配。能够设置不同的语言首选项，用于该工具提示的方框尺寸更可取地是与在选择语言中的单词长度相匹配。

图 12 显示了当选择 VIEW 按钮时获得的显示。一个 VIEW HISTORY 窗口 1141 能够在该屏幕上显示，其具有一个标题 1143（在这里已经选择了用于该标题的法语语言选项）以及由该用户查看过的前一页的详细数据 1145。该用户可以使用该 SELECT 键 1186 上下滚动该文本并且突出显示详细资料 1145 的一个。该 LEFT 和 RIGHT 键用来突出显示 OK CANCEL 键 1149、1149 。

图 13 显示了突出显示的 ADD 按钮 1148 以及它相关的工具提示。ADD 按钮 1148 用来把当前显示页增加要历史列表 1145 中。

图 14 显示了突出显示的 BOOKMARK 按钮 1150 以及它相关的工具提示 1151。在图 15 中，已经选择了 BOOKMARK 按钮 1150 而且显示了包含工具提示 1151、VIEW、ADD、DELETE 以及 EDIT 按钮 1153、1154、1155、1156 的书签子链 1152。在图 15 中，突出显示了 VIEW 按钮 1153 以及它显示的工具提示。VIEW 按钮 1153 被选择而且显示了 VIEW 窗口 1157（参见图 16）。（将可以看到，为了清楚起见，当该查看窗口 1157 在该屏幕上时不显示书签子链 1152。）VIEW 窗口 1157 包含一个标题，以及一个可滚动文本框列表书签。对于 VIEW HISTORY 窗口，该窗口还包含 OK 和 CANCEL 键。该光标小键盘 1182 用来导航该窗口以及如果期望的话，选择一个书签。

在图 17 中，突出显示了 ADD 按钮 1154 而且显示了它的工具提示。如果选择了 ADD 按钮，则显示 ADD 窗口（参见图 18）。该 ADD 窗口 1158 包含两个用于文本输入的方框，以输入该书签的 URL 以及它的标题。使用数字键 1181（例如使用在这里描述的在屏幕上键盘）把数据输入到该窗口中。该窗口还包含如上所述的 OK 和 CANCEL 键。该用户使用光标 1182 在文本输入框和 OK 以及 CANCEL 键之间

导航。

图 19 显示了突出显示的 DELETE 按钮 1155 以及它的工具提示。通过选择 DELETE 按钮 1155，可以删除书签。

图 20 显示了突出显示的 EDIT 按钮 1156 以及它的工具提示。通过选择 EDIT 按钮 1156，可以编辑书签。

图 21 显示了书签子链 1152 的一种替换形式，其中不显示书签工具提示 1151。这个能够保存在该屏幕上的空间，特别地如果该子链是长的话。该工具提示 的外观是一个能够由该用户选择的选项。

图 22 显示了突出显示的 SETTINGS 按钮以及它的工具提示 1161。当选择 SETTINGS 时，显示该验证窗口 1165（参见图 22）以提示该用户在能够改变设置之前标识他自己并且给出该用户口令。该验证窗口 1165 包含两个文本输入框用于使用数字键 1181 以及 OK 和 CANCEL 按钮输入该用户名和口令。一旦正确的用户名和口令已经输入到该验证窗口 1165 中并且选择了 OK 按钮，就显示了设置子链 1162，参见图 24。

该设置子链 1162 包含 MODEM 按钮 1163 以及 BROWSER 按钮 1164。图 24 显示了突出显示的 MODEM 按钮 1163 以及它相关的工具提示。通过选择 MODEM 按钮 1163，可以改变调制解调器的设置。图 25 显示了突出显示的 BROWSER 按钮 1164 以及它相关的工具提示。当选择 BROWSER 按钮 1164 时，显示浏览器窗口 1166，参见图 26。再次，用户使用光标键 1182 沿着在该窗口中的对象进行导航。在该浏览器窗口中的对象包含一个 COLOUR 下拉列表 1167。通过使用该光标键 1182 突出显示该列表标题以及选择它，显现该列表的条目而且用户能够使用该光标上下移动该列表以及选择一个新的浏览器颜色。类似地，使用该下拉列表 1168 能够改变该浏览器的文本语言。通过移动该突出显示到工具提示选择 1169 并且按下该选择按钮 1186，该工具提示能够被打开和关闭。该窗口如前所述，包含 OK 和 CANCEL 按钮。

图 27 显示了突出显示的 QUIT 按钮 1170 以及它的相关工具提示 1171。如果选择了 QUIT 按钮 1170，则显现该退出子链 1172（图 28）。在图 28 中，突出显示了 CONFIRM 按钮 1173 并且显示了它的工具提示。如果该用户希望退出该网络浏览器，则他选择 CONFIRM 按钮

1173。如果该用户想要取消该退出选择，则他选择 CANCEL 按钮 1174，在图 29 中突出显示了按该钮并且显示了它的工具提示。

可以使用该主菜单链 1100 的替换设计，借此更可取地是能够单独改变按钮的形状以及纹理而不改变该链的整体形状。颜色也可以被改变。这些改变能够作为在该设置菜单中的选项而变得可用，该链的形状和纹理作为一个整体被改变，从而给出用于该浏览器的一个相关设计（外壳）。

例如，该按钮可以具有正方形、圆形、菱形或者其他形状，一个纹理更可取地是被设计为给出一个三维外观。在按钮之间的链接的外观可以被设计为匹配该按钮的外观，或者做为选择补充它。还可以选择该链接的外观以便给出结构上完整的外观，以增加用户认为该按钮是有意义互链的感觉。

在图 6 - 30 中显示的链的弧形状是由开发者选择的，而且更可取地是不可由该用户改变。该按钮链和子链的其他配置，诸如一个直链或者一个半圆形链，是可能的。在其他实施例中，该按钮链和子链的按钮次序和配置可以由该用户修改。

该接收器／解码器提供了国际互链网导航和读取电子邮件功能。

现在讨论用于为一个导航器建模的图形工作室。

用于为一个导航器建模的图形是一个基本图形对象的组合或者集合。每个图形对象都是该导航器的一个功能在一个电视屏幕上的图示表现。该导航器的每个功能都能够由一个图形对象表示，或者由一个图形对象的一序列图像（一个动画），或者一个图形对象集合（例如，一个在该屏幕背景中的图像或者在一个对话框背景中的图像，其它图像能够附着于该图像上）来表示。有两种用于图像的内部格式：MPEG2 和 PIXMAP - GRL。

PNG 格式用于表示该“导航系统”功能的基本图形对象：加载、链接、前一个文档、下一个文档、退出、等。

为了在该图形平面上加印一个非矩形图像，必须使用一个定义该可见（有效）区域的剪辑掩码。这个掩码必须由该设计者以一个位图的形式提供：由于性能限制，这个掩码不由该程序计算。

用于增加一个剪辑掩码的两个阶段是首先显示一个图像，然后，在该图形平面中，用透明颜色填充与该图像尺寸和位置相同的一个矩

形，同时应用该剪辑掩码以使该图像的有用部分变得可见。

PIXMAP - GRL 图像格式用于表示该导航器资源或者用户界面的图形对象：垂直卷动条，表格，单选，多重选择，等。

该 PIXMAP 类型的对象具有可变的尺寸（每个图形对象或者模型被分解成简单的基本对象）以及是可变色的（改变颜色）。

该 PIXMAP - GRL 图像格式可以通过使用众所周知的方法转换任何其他图形格式（诸如 BMP、JPEG、GIF、PNG、等）来获得。依据一个矩阵（例如 3x3、4x4 或者 1x4）实施把一个图形对象分解成图形单元，该矩阵取决于被显示的对象类型。

调色板包含 256 种颜色。这个调色板应用于 PIXMAP 和 PNG 类型的图形对象。在一个调色板中有两个部分。第一部分包含 26 种保留便于该应用程序显示和设计的颜色。第二部分包含 230 种可以由该应用程序使用的不可修改的颜色。

该屏幕的最大尺寸是 720 像素宽 576 像素高。为了保证该屏幕在每个低档电视机上可见，必须限制该尺寸到 592 像素宽和 480 个像素高。为了利用高档电视机，用户将具有调整该屏幕尺寸的选项。对于一个国际互链网导航器，站点通常把它们的网页设计成 600 像素宽、400 像素高。

现在详细地讨论一个图形对象的一般属性。

一个图形对象（依据它表示什么）具有一个精确的尺寸。该确切尺寸由设计者定义并且起用于该显示的一个指导作用。

每个图形对象都是可调整大小的。依据图形对象的类型，有可能调整宽度和／或高度的大小。调整一个 PIXMAP - GRL 图形对象大小的方法遵循分解一个图形对象的建议。稍后进一步讨论绘制可变尺寸图形对象的方法。

该图形对象在该屏幕上的图像由一个多色彩形状表示；如果可能的话，该背景图像将具有一个颜色范围（不固定，粘土模型的效果）。

该图形对象的图像不必依据它的位置（一致），或者依据在该屏幕上下面或者前面对象的次序进行绘制：该对象一个浮动位置的概念。每个对象都与另一个对象无关进行定义（除一个背景图像之外）。

依据选择的语言由该程序打印该文本。这意味着将没有图像包含文字。一个对象的设计掌握在设计者的手中。该一般外观能够遵循一

个特定的主题（例如 Startrek、007、Simpsons）。

焦点方面能够由几种方法表示：一个在该图形对象上的矩形焦点；一个突出显示（用另一种颜色）该图形对象背景的焦点、或者一个以该图形对象的形式着色的焦点。

该标准状态（没有焦点、有效、没有按下）是一个基本的图形对象。

一个图形对象的禁止状态能够由几种方法表示：在灰色中的对象形状（或者虚光）；在议论中的图形对象上覆盖一个特殊的禁止符号；设置该对象的背景到单色、或者使该对象看不见。

一个图形对象的按下状态是一种具有焦点的对象在被点击之后但是在该按钮被释放之前的图形表示。这个表现能够是该对象的一个反相显示或者它能够与聚焦状态相同。

关于该翻滚（头尾相接）显示效果，一个图像或者图标可以包含两个视觉方面：一个表示文字（背面，例如，或者一个 Euro 硬币的“尾”侧），另一个表示一个符号（面部，或者“标题”侧）。这个视觉效应由一个运行在一个计时器上的程序动画制作；在显示第一个图标处启动一个计时器并且一旦该计时器达到一个预定值，发生图标的改变：要么显示第二个图标、要么显示一序列显示一个顺序改变的图标。

现在参考图 31 到 38 描述一个 PIXMAP - GRL 类型的图形对象的分解。这些图显示了在矩阵分解法中使用的基本图形对象的示例（1201 - 1209，1211 - 1219，等），以及当该基本对象以适当方式组合时形成的相应图形对象（1210、1220，等）。这些图已经被放大了大约 4 倍。

为了能够调整一个图形对象大小（增或者减），每个图形对象（由该图形画家创建的一个设计）被分解成图形单元的一个矩阵形式中，其类似于一个拼板的碎片。每个单元然后依据四个主要边角和中心被索引（北、南、东、西、中心、东北、西北、东南、西南、中心的中心）。该矩阵的宽度和高度取决于对象的类型。

某些图形单元（玩具碎片）仅仅被打印一次（边角）。为了使该对象更宽或者更高，某些单元以一种重复的方式被打印（分别为该单元的 n 倍宽度或者高度）。

现在列出由该矩阵分解（或者并列显示）方法形成或者形成一部分的图形对象。这些图形对象在该 HTML 文档区域中绘制。

- 在活动状态有／没有文本的按钮（1210）：3x3 矩阵（1201 - 1209）；单元尺寸：4 像素宽和高；调整宽度大小的单元：北中（1202）、正中（1205）、南中（1208）；调整高度大小的单元：西中（1204）、正中（1205）、东中（1206）。
- 在活动中有／没有文本的按钮，按下状态（1220）：3x3 矩阵（1211 - 1219）；单元尺寸：4 像素宽和高；调整宽度大小的单元：北中（1212）、正中（1215）、南中（1218）；调整高度大小的单元：西中（1214）、正中（1215）、东中（1216）。
- 在不活动中有／没有文本的按钮，灰色状态（1230）：3x3 矩阵（1221 - 1229）；单元尺寸：4 像素宽和高；调整宽度大小的单元：北中（1222）、正中（1225）、南中（1228）；调整高度大小的单元：西中（1224）、正中（1225）、东中（1226）。
- “复选框”，有／没有焦点，交叉／不交叉（图 34）：1x1 矩阵；单元尺寸：16 像素宽和高；要调整宽度大小的单元：没有；要调整高度大小的单元：没有。
- 选项列表，用于单一或者多重选择（1252、1253、1254）：3x3 矩阵（1241 - 1249）；单元尺寸：4 像素宽和高；调整宽度大小的单元：北中（1242）、正中（1245）、南中（1248）；调整高度大小的单元：西中（1244）、正中（1245）、东中（1246）；能够包含一个向上（1250）和／或向下指示符（1251）；位置：起始 x，y +该向上指示符的宽度和高度。
- 向上指示符（1250）：1x1 矩阵；单元尺寸：16 像素宽和 8 像素高；要调整宽度大小的单元：没有；调整高度大小的单元：没有；位置：起始 y，宽度的中心。
- 向下指示符（1251）：1x1 矩阵；单元尺寸：16 像素宽和 8 像素高；要调整宽度大小的单元：没有；调整高度大小的单元：没有；位置：起始 y +向上指示符的高度+列表高度，宽度中心。
- 表格（用于绘制窗体）（1260）：3x3 矩阵（1270）；单元尺寸：2 像素宽和高；要调整宽度大小的单元：北中、正中、南中；要调整高度大小的单元：西中、正中、东中；能够包含一个单元。

- 单元（用于绘制在一个窗体中的一个条目）（1280）：3x3 矩阵（1290）；单元尺寸：2 像素宽和高；要调整宽度大小的单元：北中、正中、南中；要调整高度大小的单元：西中、正中、东中；能够包含文本或者一个图像；位置：起始 x，y +该表格边界的厚度。
- 有框架的文本（文本区域）（1300）：3x3 矩阵（1310）；单元尺寸：2 像素宽和高；要调整宽度大小的单元：北中、正中、南中；要调整高度大小的单元：西中、正中、东中；能够包含文本。
- 框架（1320）：3x3 矩阵（325）；单元尺寸：4 像素宽和高；要调整宽度大小的单元：北中、正中、南中；要调整高度大小的单元：西中、正中、东中。
- 垂直卷动条（1330）：1x3 矩阵；单元尺寸：8 像素宽和高；要调整宽度大小的单元：没有；要调整高度大小的单元：正中；附着于该框架上（取决于该框架对象的位置）；包含相对于高度的索引的指示符图形对象。
- 水平卷动条（1340）：3x1 矩阵；单元尺寸：8 像素宽和高；要调整宽度大小的单元：正中；要调整高度大小的单元：没有；附着于该框架上（取决于该框架对象的位置）；包含相对于宽度的索引的指示符图形对象。
- 水平线：1x1 矩阵；单元尺寸：4 像素宽和高。
- 垂直线：1x1 矩阵；单元尺寸：4 像素宽和高。

现在给出在该网络浏览器界面中的全部图形对象以及它们相关功能的摘要。

下列所述是图形对象的一个不详尽清单，这些为在该解码器中构造一个导航器模型所必需。在这里给出的表格逐个单元列出这些对象，以及由几个图形单元组成的对象列表。

基本图形对象	功能	类型	注释
BTN_Connecter	链接／断开调制解调器线	可点击的 PNG 图像	用定义的简档表信息启动该调制解调器链接
BTN_Deconnecter	链接／断开调制解调器线	可点击的 PNG 图像	断开该调制解调器
BTN_Information_Profil	访问用于该链接简档表的验证信息	可点击的 PNG 图像	显示用于配置该订户简档表的对话框：登录名、口令以及电话号码
BTN_Information_Serveur	快速访问该代理服务器信息	可点击的 PNG 图像	显示一个用于配置该代理服务器参数的对话框
BTN_Configuration_navigateur	快速访问该导航器选项	可点击的 PNG 图像	显示一个用于配置该导航器选项的对话框
BTN_Acces_Email	快速访问该电子邮件应用程序	可点击的 PNG 图像	启动该电子邮件客户程序
BTN_Document_Precedent	前一页	可点击的 PNG 图像	显示就在前面的 HTML 页
BTN_Document_Suivant	下一页	可点击的 PNG 图像	显示紧跟在后面的 HTML 页
BTN_Stop_Chargement	停止当前文档加载	可点击的 PNG 图像	停止当前文档的加载
BTN_Annuaire	快速访问书签	可点击的 PNG 图像	显示一个包含书签列表的对话框
BTN_Quitter	离开该应用程序并且返回到 TV	可点击的 PNG 图像	离开该应用程序

基本图形对象	功能	类型	注释
BTN_Saisie_Adresse	快速访问一个链接的选择	可点击的 PNG 图像	打开一个从中能够访问一个站点地址（URL）或者从中选择一个建立的 URL 的对话框
IMG_Logo_Navigateur	商标	可点击的 PNG 图像	显示当前加载动作的图像循环序列
IMG_Logo_Operateur	商标	可点击的 PNG 图像	
IMG_Diode	状态或者活动指示	不可点击的 PNG 图像	颜色二极管（LED）：红色、绿色、蓝色、黑色、黄色
BTN_Fleche_haut	向上移动一行	可点击和不可点击的 PNG 图像	可点击的图像指示：按下状态、没有按下状态以及禁止状态。不可点击图像指示：禁止和正常状态
BTN_Fleche_bas	往下移动一行	可点击和不可点击的 PNG 图像	可点击的图像指示：按下状态、没有按下状态以及禁止状态。不可点击图像指示：禁止和正常状态
BTN_Fleche_droite	向前移动光标	可点击和不可点击的 PNG 图像	可点击的图像指示：按下状态、没有按下状态以及禁止状态。不可点击图像指示：禁止和正常状态
BTN_Fleche_gauche	向后移动光标	可点击和不可点击的 PNG 图像	3 个可点击图像：按下状态、不按下状态和禁止状态 2 个不可点击图像：禁止和正常状态

基本图形对象	功能	类型	注释
BTN_Page_haut	向上移动几行光标(半页或者前一页)	可点击和不可点击的 PNG 图像	3 个可点击图像：按下状态、不按下状态和禁止状态 2 个不可点击图像：禁止和正常状态
BTN_Page_bas	向下移动几行光标(半页或者下一页)	可点击和不可点击的 PNG 图像	3 个可点击图像：按下状态、不按下状态和禁止状态 2 个不可点击图像：禁止和正常状态
BTN_Frame_Suivante	移动到下一个框架	可点击的 PNG 图像	3 个可点击图像：按下状态、不按下状态和禁止状态 如果有接着的一个框架则这个图像是可见的；否则不显示该图像以避免覆盖该屏幕
BTN_Frame_Precedente	移动到前一个框架	可点击的 PNG 图像	3 个可点击图像：按下状态、不按下状态和禁止状态 如果有接着的一个框架则这个图像是可见的；否则不显示该图像以避免覆盖该屏幕
BTN_Choix_Croix	在一个列表内的一个选项被选中与否的指示符	可点击和不可编辑的 PNG 图像	类似于用于插入一个交叉的方框。多种选择是可能的。因此有 2 个图像：选中和未选中的
BTN_Choix_Simple	在一个列表内的一个选项被选中与否的指示符	可点击和不可编辑的 PNG 图像	类似于用于插入一个交叉的方框。只有一个来自列表的选择。因此有 2 个图像：选中和未选中的
BTN_Ok	验证一个问题或者信息	可点击和不可编辑的 PNG 图像	一般的验证图像（没有文本）。在一个对话框...中使用。

基本图形对象	功能	类型	注释
BTN_Annuler	删除一个问题或者信息	可点击和不可编辑的 PNG 图像	一般的删除图像（没有文本）。在一个对话框...中使用。
BTN_Oui	对一个问题的正面回答	可点击和不可编辑的 PNG 图像	一般的验证图像（没有文本）。在一个对话框...中使用。
BTN_Non	对一个问题的否定回答	可点击和不可编辑的 PNG 图像	一般的否定图像（没有文本）。在一个对话框...中使用。
IMG_Curseur_Normal_Souris	指针／图形光标	标准的图形光标指示符图像	类似于具有一个热点的箭头
IMG_Curseur_Attente_Sours	在繁忙模式中的指针	繁忙指示符	类似于一个炸弹计时器或者精确时计
IMG_Curseur_Texte	在访问期间在一个文本中的图形光标位置	指示符	
IMG_Statusline	显示在一个具有该屏幕尺寸的 4／5 的行中的文字	用于显示不可编辑文本的背景图像	没有文本的图像。该文本将以截取格式显示。该文本不超过 40 个字符
IMG_Indefinie	一个还没有加载的图像的一般显示	可点击和不可点击的图像	表示一个在一个 HTML 文档中、还没有被载入到存储器中，或者它的加载被中断了的图像的图像
BDG_Login_Password	用于显示或者输入该登录名和口令以访问一个安全站点的一般对话框	具有图形边界的屏幕背景图像	界定一个信息文本、登录名字段或者口令字段区域的背景图像。这个对话框是访问一个安全站点所要求的。该程序掩码该口令

基本图形对象	功能	类型	注释
BDG_Information_Profil	用于显示或者输入一个远程访问服务器的登录名、口令和电话号码的一般对话框。	具有图形边界的屏幕背景图像	界定一个信息文本、登录名字段、口令字段或者电话号码字段区域的背景图像。在拨号该访问服务器之前要求这个对话框。该程序掩码该口令，并且将要求确认该口令。
BDG_Confirmation_Password	用于确认一个口令输入。	具有一个带有图形边界的背景图像的对话框	界定一个信息文本和一个口令字段区域的背景图像。当该用户修改或者输入一个新口令时，需要这个对话框。该程序掩码该口令
IMG_Telecommande_filigrane	用于使用该虚拟键盘高级输入的遥控器的装饰(键的外形)	外形	覆盖在该虚拟键盘的图像上的遥控器外形，以使该遥控器的键相应于该虚拟键盘的键布局是可视的。由图像存储器快速访问而不由文档存储器访问。遥控器键的外形不包含一个字母或符号。

基本图形对象	功能	类型	注释
BKG_Clavier_virtuel	用于使用一个遥控器高级输入的虚拟键盘	一个键盘的背景图像	该虚拟键盘如同一个 PC 的键盘那样表现在屏幕上，但是没有在该键上显示字母或者图。把字母放在每个键上由该程序处理：这允许定义一个一般的国际语言键盘（azerty、qwerty 或者其它）。遥控器的外形图像覆盖在该键盘上并且跟随该焦点：这实现了可视显示的目标，即按下一个遥控装置按钮对应于按压在该键盘上的一个键，而不用不得不事先记住该转换。该 ESC 键取消该虚拟键盘。某些键起功能键：“http://www”、“.fr”、“.com”、“.org” 等等作用，而且其它键具有某些功能：“回车”、“退格”、“删除”，等。
BKG_Toolbar_Navigation	在该工具条背景中显示	背景图像	
BKG_Toolbar_Systeme_ Configuration	在该工具条背景中显示	背景图像	
BKG_Annuaire	在该书签对话框的背景中显示	背景图像	
BKG_Information_Profil	在该订户简档表配置对话框的背景中显示	背景图像	

基本图形对象	功能	类型	注释
BKG_Information_Serveur	在该代理服务器配置对话框的背景中显示		

以下表格描述了形成一个能够在该图形表示中具有一个可变尺寸的图形对象的、不同的基本图形对象。并置基本对象以形成一个合成对象受到该程序的影响（重新组成拼板）。这些对象具有 PIXMAP 格式。

基本图形对象	功能	类型	注释
TBL_Vertical_Gauche	表现一个可变尺寸表格的边界	图像	在一个可变尺寸的表格上绘制边界。这个对象表示该边界左侧的一个垂向拉伸
TBL_Vertical_Droite	表现一个表格的边界	图像	在一个可变尺寸的表格上绘制边界。这个对象表示该边界右侧的一个垂直延伸
TBL_Horizontal_Haut	表示一个表格的边界	图像	在一个可变尺寸的表格上绘制边界。这个对象表示该边界顶部的一个水平延伸
TBL_Horizontal_Bas	表示一个表格的边界	图像	在一个可变尺寸的表格上绘制边界。这个对象表示该边界底部的一个水平延伸
TBL_Angle_Haut_Gauche	表示一个表格边界的边角	图像	绘制一个表格边界的左上角。
TBL_Angle_Haut_Droite	表示一个表格边界的边角	图像	绘制一个表格边界的右上角。

基本图形对象	功能	类型	注释
TBL_Angle_Bas_Gauche	表示一个表格边界的边角	图像	绘制一个表格边界的左下角。
TBL_Angle_Bas_Droite	表示一个表格边界的边角	图像	绘制一个表格边界的右下角。
CEL_Vertical_Gauche	表示一个单元的边界	图像	绘制一个单元的边界。这个对象表示该边界左侧的一个垂直延伸。一个单元是在一个表格...中的隔箱。一个单元边界的厚度将小于表格的边界厚度
CEL_Vertical_Droite	表示一个单元的边界	图像	绘制一个单元的边界。这个对象表示该边界右侧的一个垂直延伸。
CEL_Horizontal_Haut	表示一个单元的边界	图像	绘制一个单元的边界。这个对象表示该边界顶部的一个水平延伸。
CEL_Horizontal_Bas	表示一个单元的边界	图像	绘制一个单元的边界。这个对象表示该边界底部的一个水平延伸。
CEL_Image	表示在一个单元中一个未加载图像的边界	一般的图像	表示在一个单元中一个未加载图像的位置的一般图像
BDG_Vertical_Gauche	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界。这个对象表示该边界左侧的一个垂直延伸。

基本图形对象	功能	类型	注释
BDG_Vertical_Droite	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界。这个对象表示该边界右侧的一个垂直延伸。
BDG_Horizontal_Haut	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界。这个对象表示该边界顶部的一个水平延伸。
BDG_Horizontal_Bas	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界。这个对象表示该边界底部的一个水平延伸。
BDG_Angle_Haut_Gauche	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界的左上角。
BDG_Angle_Haut_Droite	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界的右上角。
BDG_Angle_Bas_Gauche	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界的左下角。
BDG_Angle_Bas_Droite	表示一个对话框的边界	一般的图像	绘制一个可变尺寸对话框的边界的右下角。
LST_Vertical_Gauche	表示可选选项的一个垂直列表的边界	图像	绘制一个可变尺寸垂直列表的边界。这个对象表示该边界左侧的一个垂直延伸。
LST_Vertical_Droite	表示可选选项的一个垂直列表的边界	图像	绘制一个可变尺寸垂直列表的边界。这个对象表示该边界右侧的一个垂直延伸。
LST_Bas_Gauche	表示可选选项的一个垂直列表的边界	图像	绘制一个可变尺寸垂直列表的边界。这个对象表示该边界顶部的一个水平延伸。

基本图形对象	功能	类型	注释
LST_Bas_Droite	表示可选选项的一个垂直列表的边界	图像	绘制一个可变尺寸垂直列表的边界。这个对象表示该边界底部的一个水平延伸。
LST_Angle_Haut_Gauche	表示一个可变尺寸的可选选项垂直列表的边界的边角	图像	绘制一个可变尺寸垂直列表边界的左上角。
LST_Angle_Haut_Droite	表示一个可变尺寸的可选选项垂直列表边界的边角	图像	绘制一个可变尺寸垂直列表边界的右上角。
LST_Angle_Bas_Gauche	表示一个可变尺寸的可选选项垂直列表边界的边角	图像	绘制一个可变尺寸垂直列表边界的左下角。
LST_Angle_Bas_Droite	表示一个可变尺寸的可选选项垂直列表边界的边角	图像	绘制一个可变尺寸垂直列表的边界的右下角。
ASC_Haut	表示一个垂直滚动条（“电梯”）	可点击和不可点击的图像	绘制一个垂直滚动条（“电梯”）的顶端
ASC_Bas	表示一个垂直滚动条	可点击和不可点击的图像	绘制一个垂直滚动条的底端
ASC_Cage	表示一个垂直滚动条	可点击和不可点击的图像	绘制一个垂直滚动条的一段（“楼层”）
ASC_Ascenseur	表示一个垂直滚动条	可点击和不可点击的图像	指示该垂直滚动条的位置

基本图形对象	功能	类型	注释
SCR_Gauche	表示一个水平滚动条	可点击和不可点击的图像	绘制一个窗口、文本区域、框架等的水平滚动条的左端。
SCR_Droite	表示一个水平滚动条	可点击和不可点击的图像	绘制一个窗口、文本区域、框架等的水平滚动条的右端。
SCR_Cage	表示一个水平滚动条	可点击和不可点击的图像	绘制一个窗口、文本区域、框架等的一个水平滚动条的一个象限（分级）。
SCR_Position	表示一个水平滚动条	可点击和不可点击的图像	指示在一个窗口、文本区域、框架等的水平滚动条中光标的位置
IMG_Texte_Gauche	表示要显示的一行文字	不可点击的图像	这是在该不可编辑显示文字左侧的图像。选择的尺寸必须匹配选择字符集的尺寸
IMG_Texte_Droite	表示要显示的一行文字	不可点击的图像	这是在该不可编辑显示文字右侧的图像。选择的尺寸必须匹配选择字符集的尺寸
IMG_Texte_Caractere	表示要显示的一行文字	不可点击的图像	这是要在上面打印不可编辑字符的背景图像。选择的尺寸必须匹配选择字符集的尺寸
IMG_Edition_Gauche	表示要编辑的一行文字	可编辑和可点击的图像	这是在一个可编辑或者可选文字区域左侧的图像。选择的尺寸必须匹配选择字符集的尺寸。
IMG_Edition_Droite	表示要编辑的一行文字	可编辑和可点击的图像	这是在一个可编辑或者可选文字区域右侧的图像。选择的尺寸必须匹配选择字符集的尺寸。

基本图形对象	功能	类型	注释
IMG_Edition_Caractere	表示要编辑的一行文字	可编辑和可点击的图像	这是在其上面打印一个可编辑或者可选文字区域的一个字符的背景图像。选择的尺寸必须匹配选择字符集的尺寸。
IMG_Multi_Edition_Gauche	表示多行文字的编辑	可编辑和可点击的图像	这是在一个可编辑或者可选文字区域左侧的图像。选择的尺寸必须匹配选择字符集的尺寸。
IMG_Multi_Edition_Droite	表示多行文字的编辑	可编辑和可点击的图像	这是在一个可编辑或者可选文字区域右侧的图像。选择的尺寸必须匹配选择字符集的尺寸。
IMG_Multi_Edition_Caractere	表示多行文字的编辑	可编辑和可点击的图像	这是在其上面打印一个可编辑或者可选文字区域的一个字符的背景图像。选择的尺寸必须匹配选择字符集的尺寸。

导航器功能

功能	描述	由什么激活	图形对象	注释
链接／断开	用定义的简档表启动该调制解调器链接／断开该调制解调器链接	BTN_Connecter BTN_Deconnecter		如果是这样离线的、如果该链接简档表信息被填写了、如果点击了一个URL，则启动该调制解调器链接。

功能	描述	由什么激活	图形对象	注释
配置-修改该链接简档表的验证信息	显示要配置该订户简档表的对话框：登录名、口令、电话号码	BTN_Information_Profil	BDG_Information_Profil	• 在闪速存储器中输入、修改、验证、存储该访问服务器的登录名、口令和电话号码 • 取消该访问
确认一个口令的输入	请求再次输入该口令	在一个口令已经被改变了之后	BDG_Confirmation_Password	• 输入、验证、取消 • 当打印时，每个输入字符由一个掩码字符（星号）所替换 • 将该输入与修改的口令相比较
打印或者输入该登录名和口令以访问一个安全站点	请求从该远程站点进行验证	在试图访问一个安全站点之后	BDG_Login_Password	• 输入、验证、取消、发送信息到该站点 • 当显示时，该口令的每个字符都由一个掩码字符替代

功能	描述	由什么激活	图形对象	注释
管理一个虚拟键盘	用于使用一个遥控器高级输入的虚拟键盘	当该焦点在一个可编辑对象上时	BKG_Clavier_Virtuel IMG_Telecommande_filigrane	该虚拟键盘如同一个 PC 的键盘那样表现在屏幕上，但是没有在键上显示字母或者图。把字母放在每个键上由程序处理：这允许定义一个一般的国际语言键盘（azerty、qwerty 或者其它）。遥控器的外形图像覆盖在该键盘上并且跟随该焦点：这实现了可视地显示的目标，即按下一个遥控装置按钮对应于按压在该键盘上的一个键，而不用必须事先记住该转换。该 ESC 键解散该虚拟键盘。某些键起功能键：“http://www”、“.fr”、“.com”、“.org” 等等作用，而且其它键具有某些功能：“回车”、“退格”、“删除”，等。

现在描述用于网络浏览器的 Java API。

下面列出在该解码器中的导航器应用程序级别处使用的 JAVA 包列表。这个列表被分成两个部分：JDK 1.1 的 AWT（抽象窗口工具包）类，以及以本地 C 语言代码编写的不同服务的 JAVA 接口类。

类	方法
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) getLocation() getSize() setBackground(Color) setBackground(ImageMask) setBackground(int) setLocation(int) setSize(int,int)
java.awt.Graphics	drawImage getSize setLocation
java.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() getSelectedItem() remove(String) replace() setMultipleMode(boolean)
java.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	

类	方法 ods
java.awt.event.FocusListener(interface)	void focusGained(FocusEvent) void focusLost(FocusEvent)
java.awt.event.KeyEvent	
java.awt.event.KeyListener(interface)	void keyPressed(KeyEvent) void keyReleased(KeyEvent) void keyTyped(KeyEvent)
java.awt.event.MouseEvent	
java.awt.event.MouseListener(interface)	void mouseClicked(MouseEvent) void mouseEntered(MouseEvent) void mouseExited(MouseEvent) void mousePressed(MouseEvent) void mouseReleased(MouseEvent)

导航器包，称为浏览器程序包，它把不同的包组合在一起，这些包为：browser.drawer 包，提供允许取出一个 HTML 文档、以及在该文档浏览器内导航的服务；以及 mediawebtv 包，允许与该用户验证机构建立一个 Internet 联接。

现在描述该 browser.drawer.MhwBookmark 类的结构。一个书签列表与一个用户相关联。在该书签列表内没有分级结构。

构造器： MhwBookmark(subscriberId): 打开一个存在的书签列表

构造器: MhwBookmark(subscriberId): 创建一个新的书签列表

deleteBookmark(): 破坏一个书签列表

add (URL, name): 添加一个条目

remove (itemNumber): 删除一个条目

modify (itemNumber, URL, name): 修改一个存在的条目

getList (): 返回一个在该书签列表中的条目列表（不分级）

getItemCount (): 返回在该书签列表中的条目数目

isFull (): 布尔值-列表是满的?

isEmpty (): 布尔值-列表是空的?

setHomePage (itemNumber)

getHomePage (): itemNumber

goToURL (sessionNumber): 加载对应于选择的条目的文档

如果发生失败，由方法 add()、remove()、modify()、getList()、

setHomePage()返回一个错误信息（或者发生是异步的话则返回一个事件）。

该 browser.drawer.MhwHistory 类允许在一个先前显示的文档列表中，从一个文档导航到另一个文档。在该历史列表中没有分级结构。现在给出该类的详细资料。

构造器: MhwHistory (sessionNumber)

getList (): 返回该历史列表（没有分级结构）

getCurrent (): 获得当前的 URL

setCurrent(indexNumber): 改变当前的 URL

getNext (): 获取下一个条目的 URL

getPrevious (): 获取前一个条目的 URL

getItemCount (): 返回在该历史中的条目数目

addEventsRegister (): 订阅该历史列表的错误事件

removeEventsRegister ()

如果发生失败，则由方法 getList()、getNext()、getPrevious()、setCurrent 返回一个错误信息（或者发生是异步的话则返回一个事件）。这些事件是: addEventsRegister(sessionNumber): [订阅该历史列表错误事件]; 以及 removeEventsRegister(sessionNumber) [取消]。

browser.drawer.MhwDocument 类允许在该解码器中加载和显示一个 HTML 文档。现在给出该类的详细信息。

构造器: MhwDocument (sessionNumber)

freeze ():停止当前文档的显示（继续加载该文档）

unfreeze (): 重新开始当前文档的显示

isPending (): 该文档当前是否正在加载

stop ():停止当前文档的加载

reload ():重新加载该文档

getDocumentInfo (): 返回该文档的标题和 URL

addStatusRegister (): 订阅有关该状态的信息以及结束加载

goToURL (url): 加载一个网页

submit (login, password, URL): 提交用于加载一个网页的验证

getStatisticsDocument (): 返回进行中的请求数目以及当前正加载文档的 URL

browser.mediawebtv.MhwConnection 类把用户链接和验证功能集合在一起。现在给出该类的详细信息。

构造器: MhwConnection (subscriberId)

start ():请求链接

stop (): 请求断开

cancel (): 取消链接

setAuthentificationType (type): 设置验证模式到 CANAL+模式（最高有效位／口令）或者通过登录名／口令

getAttributes ():返回该链接属性

setAttributes (attributes): 修改该链接属性

setPassword (password): 修改口令

getPassword (password): 获取口令

setAutoCheckPassword (bAutoCheck): 设置该口令是否自动地用确认口令验证

getAutoCheckPassword (bAutoCheck): 读取该口令是否自动地用确认口令验证

getIPClient (ipaddress, netmask): 读取选择的 IP 和网络掩码对

setIPClient (ipaddress, netmask): 修改该 IP 和网络掩码对

getDNS (dns1, dns2): 读取主要和辅助 DNS 地址

setDNS (dns1, dns2): 修改该主要和辅助 DNS 地址

getURLConfigServer (url):读取该配置服务器的地址

setURLConfigServer (url): 修改该配置服务器的地址

getQueryCommand (queryCmd, typeOfQuery): 按类型读取要被发送给该配置服务器的请求

setQueryCommand (queryCmd, typeOfQuery): 按类型修改该请求

queryAcquisitions (tableAcquisitions , typeOfAcquisition , NumberOfAcquisitions): 读取获取的列表

startAcquisition (acquisitionId): 开始一次获取（数据／视频）

stopAcquisition (acquisitionId): 停止一次获取（数据／视频）

addStatusRegister (): 链接状态的被通知的订阅

这些事件是: 链接失败; 当前链接; 建立的链接; 链接确认请求;

链接错误；调制解调器状态：打开／关闭；在进行初始化；在进行拨号；错误但是调制解调器是打开的；服务器状态：无效端口、无效的URL；登录错误：不能识别的登录；以及无效口令。

removeStatusRegister ()：取消订阅网络链接状态

isConnected subscriberId()：返回给出调制解调器链接／断开状态的布尔值

isPending ()：返回用于调制解调器当前建立链接的布尔值

getExtendedProviderUrl (providerUrl)：读取当前预订的提供者

setExtendedProviderUrl (providerUrl)：修改当前预订的提供者

browser.mediawebtv.mhwconfiguration 类管理每个用户的简档表以及他或者她的参数选择。现在给出该类的详细信息。

构造器：MhwConfiguration()

readProfile(subscriberId)：读取简档表

writeProfile(subscriberId，profile)：写简档表

readDefaultProfile()：读取缺省的简档表

writeDefaultProfile(profile)：修改缺省的简档表

getUserCount()：用户数量

newUser(profile)：标识一个新用户

getLastConnect()：最近的链接用户

简档表的最大数当前被固定为 5，但是这不表示一个严格的限制；如果必要的话可以存储更多数量的简档表。

在失败的情况下，由 WriteProfile()和 writeDefaultProfile()方法返回一个错误信息（如果该事故是异步的则返回一个事件）。

browser.mediawebtv.MhwMultiSession 类选择一个导航器会话。一个会话是由另一个实例自动启动的一个导航器实例。当启动该导航器时，紧接着一个验证链接的建立创建一个会话。现在给出该类的详细信息。

构造器：MhwMultiSession

getCurrentSessionNumber()

setCurrentSessionNumber(int number)

getPreviousSession()

addSession()：返回创建的会话数目

removeSession(int numSession)

现在参考图 6a 和 7 - 30，更详细地描述用于该解码器的导航器模型。

在这里给出的模型是一个导航器的一个简单示例，并且给出该主要功能的大意。它在图形表示级别处给出完全的自由度。唯一重要的特征是其中同样类型的功能被组合在一起的区域或者屏幕，以及通用的用户界面。

该导航器使用该图形工作室全部可用的图形（MPEG、PIXMAP）。这个模型的屏幕以树形结构组织起来，并且每个都把一个必需功能的集合组合在一起。通过借助于箭头按钮（在遥控器上）移动一个焦点或者通过使用一个带有光标／指针的小键盘，可以访问在一个屏幕中的每个功能或者选项。一个动作的选择由一个控制器点击或者通过一个预定义的按钮（例如“OK”）来实现。

在没有一个物理键盘的情况下，要用遥控器输入文字，就必须提供一个虚拟键盘。这可以通过用一种快速轻敲可能性来移动焦点、通过把遥控器的按钮映射到该虚拟键盘上来实现；换句话说，在遥控器上的按钮图像，以其轮廓的形式或者稍微不透明地，在该虚拟键盘的图片上是可见的（以跟踪的形式）。该虚拟键盘随后在这个文档中进行更深入的讨论。

图 30 显示了导航器功能按钮（1410）的最高级链，并且看得见在它下面的一部分网页（1411）。

现在具体参考图 30，描述主要的导航器画面。

该导航器功能被组合到几个层中。该主屏幕显示一个垂直条（主菜单），其由一系列按钮组成：Reload（重载）／Stop（停止）（1401）、前一页（1402）、下一页（1403）、最近访问站点的历史记录、书签、链接／断开、配置、退出导航器（1408）。

当按压一个功能键（要么在遥控器上要么在该键盘上）时，显示该导航器 GUI（主要的屏幕）。当该 GUI 在该电视屏幕上是可见时，HTML 文档（1411）（该 GUI 覆盖在其上面）继续被载入到存储器中，但是该文档的刷新被中止了以便不影响该 GUI 的显示性能。当该导航器 GUI 被取消了时重新显示该 HTML 文档。如果最终能够满足性能的话，则将取消该限制，由此允许同时显示该 HTML 文档和 GUI。

通过按下一个功能键（在遥控器或者键盘上），或者通过在 GUI 按钮区域外面点击，可以从该电视屏幕中取消该 GUI。然后开始或者重新开始显示当前正被加载或者保存在该高速缓冲存储器中的 HTML 文档。

一个按钮实际上是一个长方形或者正方形区域（例如，32×32 像素）。当该图形光标进入一个区域时，那个区域（窗口）接收该焦点（参看 EnterNotify(WindowId)函数）。

例如，如果该按钮图形是一个轮胎，则必须检测该图形光标的实际位置实际上是否覆叠该轮胎的像素。为此，必须找到在该鼠标指针热点处的像素在该按钮的剪辑掩码中的值（通过计算相对位置，在剪辑掩码中 getpixel()，然后测试该像素值）。这个检测方法允许改善检测在启动该按钮的功能之前是否实施了一次点击。

当该鼠标指针离开该按钮的矩形或者正方形区域时，该区域失去该焦点（参看 LeaveNotify(WindowId) ）。

现在更详细地描述描述性的按钮总结。

当该鼠标指针或者矩形焦点与表示一个功能的按钮重合时，横向或者纵向显示一个短的词语（工具提示 ），来描述那个按钮的功能。当该按钮通过一次鼠标点击或者通过一个功能键被选择时，显现包含子菜单选项的一个按钮列表。短的描述词语（工具提示 ）系统还用于该子菜单按钮。

利用在遥控器上或者在键盘上的箭头键可以实现菜单选项的导航。记录在该主要屏幕上具有焦点的最后一个按钮，用于下一次显示该主屏幕。

现在参考图 42 - 45 更详细地讨论虚拟键盘。

图 42 示意地显示了当前可见的虚拟键盘（1501）如何映射到在下面的虚拟键盘“网格”上（1501，1506）。

图 43 显示了一个典型的字符到在一个虚拟键盘上的键的映射。

图 44 显示了用于虚拟键盘的图像，其中第一（1545）和第二（1546）号块分别具有焦点（还显示了两个不同类型的遥控装置 1542 和 1543）。

图 45 显示了具有典型尺寸标注的一个典型虚拟键盘布局。

首先，该虚拟键盘被认为是一个与使用它的应用程序无关的工具。这就是说，它能够在“WebBrowser（网络浏览器）”应用程序中

使用以及同样地在“Mail（邮件）”应用程序中使用。此外，它的“外观”完全独立于涉及的应用程序的“外观”。

从不具有一个物理键盘或者不具有一个带有键盘的遥控装置的用户选择该屏幕的一个可编辑区域那一刻开始就显示该虚拟键盘。该焦点定位于在该可编辑区域中文字的结尾处。按压“OK”（在遥控器上或者在虚拟键盘上）或者“Cancel”（在该虚拟键盘上）就会取消它。

在屏幕 1501）上可见的虚拟键盘由三块并排（1502，1503，1504）的十个按键（表示遥控器的三个数字块）组成。用户能够使用在遥控器上的箭头从一个块传递焦点（1505）到另一块。在选择一块之后，按下在遥控器数字块上的一个按钮键入显示在虚拟键盘上的相应字符。

用户还可以使用向上和向下箭头键。这产生在该屏幕上相同的虚拟键盘但是具有不同的字符在该键上（1506）。因此，在一组 5 个虚拟键盘之间进行切换，一个人就能够显示在一个西方计算机键盘上的所有字符。还有可能随着需要的增加而增加其他键盘。

参考图 44，为了允许在遥控器数字块和在该虚拟键盘上的焦点之间进行一个即时关联，遥控器的一个重叠图像指示该焦点（1542，1543）。因此该用户能够容易地目测仅仅具有焦点的键盘的一个部分，而且其余的字符能够通过用箭头键移动焦点而到达。该键盘被设计为着眼于具有随后几点。

该虚拟键盘解决方案占据少量的银幕高度，并且允许容易地扩展可用的字符数目。缺省时，它是具有显示的小型字母表的虚拟键盘。

某些按钮具有重要的特定功能：-

- 在遥控器上的“OK”，用于确认当前选择（如果该字段仅仅具有一行，字符“¶”还用于验证，或者做为选择可能被选定为根本没有效果；否则，它仅仅相应于一个回车）。
- 在虚拟键盘上的“Cancel”（1522），其不确认就退出该工具（在打开该键盘之后进行的修改将丢失）。
- 在该虚拟键盘上的“Back Space”（1523），其擦除最后键入的字符。
- 该上、下、左和右箭头符号键，用于在该编辑区域内部移动。
- 在该虚拟键盘上的“Tab”（1520），其一次插入一个可配置数

目的空格符（缺省时为四个空格符）。

该键盘总是处于“插入”模式中。

图中给出了一个键盘的示例，其将在下面被进一步讨论。

利用在图 43 中显示的 5 个键盘（5x3），以及为 WebBrowser 应用程序安装的两种字体（Arialweb 和 Courier），就能够覆盖一个传统键盘的所有字符。该在屏幕上键盘的尺寸是 272 像素宽 184 像素高。

该虚拟键盘以及关于它在各种应用程序中使用的功能链接包含在包“canalplus.virtualkbd”中。

包含在该包内的类包含“MhwVirtualKbd”（虚拟键盘图形描述和行为类）、“MhwVkTextField”（从‘java.awt.TextField’中导出的类，其允许在全局应用程序内定义一个虚拟键盘，以共享 TextField 来控制事件）以及“MhwTextArea”（从‘java.awt.TextArea’继承的一个类，允许在全局应用程序内定义一个虚拟键盘，以使用 TextArea 类来控制事件）。

现在更详细地描述类 MhwVirtualKbd。

类“MhwVirtualKbd”的构造器定义为‘私有’。因此，当启动可能需要使用虚拟键盘的主应用程序（例如，在没有一个物理键盘的情况下）时，仅仅能够构造一个唯一的虚拟键盘。因此其目的是给出一个特别为当前应用程序配置的键盘，而且当该用户进入一个文本域（单行或者多行）时显现它。

当创建该键盘时，将已经设置了能够被配置的四个主（静态）变量：

- 母体：容器，该虚拟键盘的‘母体’，当创建该键盘的时候，该‘母体’本身必须存在。它使用方法“setParent”设置，如果在参数中传递的‘母体’是‘空’，则该方法返回一个‘NullPointerException’。
- 描述文件：描述该键盘的 ASCII 码文件，关于在该图像之后的图形、其涉及当使用该虚拟键盘时获得的各种‘键盘’，以及关于打印在该键上的标签。字符用它们的双字节字符集代码表示。该描述文件的名称能够使用方法“setScreensFile”来设置。
- 屏幕的数目：由该虚拟键盘‘初始化’和使用的‘数字块’数目。这个数目，使用“setScreensNumber”设置，相应于键盘的数

目，键盘的特征从上面详细描述的描述文件中读取。

• 初始坐标：这些是该键盘的背景图像的左上角在母体容器（如上所述）的坐标。这使用方法“setCoordInit”设置。

一旦创建了该键盘，通过使用方法“getInstance”能够确定是否来使用它，如果存在的话，该方法查找当前应用程序的键盘（如果该应用程序的虚拟键盘还没有存在，而且如果该应用程序使用它，方法“getInstance”使用已经为它设置的变量[先前描述的四个]来创建一个）。

现在描述事件管理。

依据前面的描述，一旦显示了该虚拟键盘，其仅仅通过解释发送给它的事件起作用，这些事件通过下列来发送：数字块、“OK”按钮、以及在遥控器上的四个箭头方向键。这些按钮对于使用中的键盘将具有具体的作用。

“OK”按钮具有一个重要的作用，因为它允许用户做两件事情：‘返回’到该文本域以输入信息，然后显示并且启动该虚拟键盘的操作；以及‘离开’该正文区域，保存该改变。

箭头键

‘右’和‘左’箭头允许遥控器的图像（指示具有‘焦点’的数字块）在在该虚拟键盘上表示的三个数字块上移动。取决于具有焦点的‘键盘’，在遥控器上的数字块的键因此被‘绑定’到各种字符的显示。

在大多数公共情况中，当该虚拟键盘是有效时，在这些按钮上‘轻敲’导致自动地在那个键展示的字符插入到在当前文本域中、由光标指示的位置处。

对于字符，六个能够被认为是‘特定’的字符，并且没有直接导致在该键上显示的字符被显示在文本域中：

退格：‘<’（1520）：当在遥控器上对应于这个字符的按钮被按下时，在当前文本域中紧贴着光标位置左侧的字符被擦除了。

Tab：‘>>’：当在遥控器上对应于这个符号的按钮被按下时，一个可配置数目的空格符（‘ ’），缺省时为 4 个，被插入在当前光标位

置处。

回车：‘¶’（1521）：当在遥控器上对应于这个字符的按钮被按下时，在该光标位置处插入一个‘换行’。事实上，如果当前文本域是类‘MhwVkTextField’的一个实例，即仅仅具有一个可编辑行时，则轻敲这个按钮将要么没有效果要么导致验证该域。相反地，如果这个文本域是类‘MhwVkTextArea’的一个实例，它包含几个可编辑行，则这个字符导致一个‘换行’（如果该光标定位在最后可编辑行上，轻敲这个按钮将没有效果）。

Cancel：‘¢’（1522）：当在遥控器上对应于这个字符的按钮被按下时，所有在打开该虚拟键盘之后对当前文本域进行的修改被取消了。换句话说，它的内容返回到它在进行该修改之前具有的值，而且‘退出’该虚拟键盘。

左箭头键：当在遥控器上对应于这个字符的按钮被按下时，在当前文本域中的光标向左边移动一个位置。如果该光标已经在‘零’位置（不可能进一步向左移动），这个按钮没有效果。

右箭头键：当在遥控器上对应于这个字符的按钮被按下时，在当前文本域中的光标向右边移动一个位置。如果该光标已经位于该文本域中最后一个字符之后（而且不能更进一步向右移动），这个按钮没有效果。

向上箭头键：当在遥控器上对应于这个字符的按钮被按下时，在当前文本域中的光标向上边移动一个位置。如果该光标已经在该文本域的第一行中（或者如果当前文本域仅仅具有一行：MhwVkTextField），这个按钮没有效果。

向下箭头键：当在遥控器上对应于这个字符的按钮被按下时，在当前文本域中的光标向下边移动一个位置。如果该光标已经在该文本域的最后一行中（或者如果当前文本域仅仅具有一行：MhwVkTextField），这个按钮没有效果。

方法“findLocation”确定该虚拟键盘在该屏幕上的位置，以设法最小化被“裁切”的表面。

类 MhwVkTextField 仅仅是在包“java.awt”中的类“TextField”的一个特例化。它另外管理一个指定该虚拟键盘使用（与否）的布尔值。

构造器与那些在包“java.awt”中的类“TextField”完全相同，除了一个简单附加的参数：一个指定该虚拟键盘使用的布尔值之外。

如果该布尔值是“真”，则创建类“TextField”的一个‘基本’实例，并且同时使用方法“addKeyListener”添加一个在当前应用程序中可用的一个虚拟键盘收听器。如果不是的话，创建一个‘标准’的 TextField。

当该 TextField 具有焦点时，如果该用户按压‘OK’并且该布尔值指定虚拟键盘的使用时，显示该虚拟键盘并且获得该焦点。它管理所有该事件并且能够填充该文本域。如果该用户再次按压‘OK’，确认它的文字，并且该键盘返回焦点。如果不设想虚拟键盘的使用（布尔值=错误）的话，则该“TextField”具有和在“java.awt”中的一个标准 TextField 相同的‘行为’。

MhwVkTextArea 类仅仅是在包“java.awt”中的类“TextArea”的一个特例化。它另外管理一个指定该虚拟键盘使用（与否）的布尔值。

构造器与那些在包“java.awt”中的类“TextArea”完全相同，除了一个简单附加的参数：一个指定该虚拟键盘使用的布尔值之外。

如果该布尔值是“真”，则创建类“TextArea”的一个‘基本’实例，并且同时使用方法“addKeyListener”添加一个在当前应用程序中可用的一个虚拟键盘收听器。如果不是的话，创建一个‘标准’的 TextArea。

当该 TextArea 具有焦点时，如果该用户按压‘OK’并且该布尔值指定虚拟键盘的使用时，显示该虚拟键盘并且该虚拟键盘获得该焦点。它管理所有的事件并且能够填充该文本域。如果该用户再次按压‘OK’，就确认它的文字，并且该键盘返回焦点。如果不设想虚拟键盘的使用（布尔值-错误）的话，则该“TextArea”具有和在“java.awt”中的一个标准 TextArea 相同的‘行为’。

以下部分进一步描述如上所述的特征的实现，尤其是从一组图形单元中重构图形对象（例如文本域、按钮、滑动块、列表检查框、选择等）。

图 39 显示了一个例如由 9 个单元组成的按钮：四个角（NW 2100、NE 2101、SW、SE 2103），四个边（N 2104、E 2105、W 2106、S 2107）

以及中心（C 2108）。

一般地、但不是必需的，这个单元都是正方形的（4x4 像素、8x8 像素、16x16 像素）。块 N、E、W、S 和 C 单元在要求的区域（由该组件的尺寸定义，其边界由外观定义）内并列显示。这些区域不必是该单元尺寸的一个倍数。当前，该并列显示从左至右或者从上到下进行，所以任何不完全绘制的单元出现在底端和右手边。如果感觉"居中"和"向右"对齐并列显示规则是有用的话，能够设想添加这些。通常，NW、NE、SW 和 SE 单元和边界一样大小所以并列显示不是必要的。然而如果它们小于所要求的，则自动地执行并列显示。

接着简要地描述如何实现这些。该工作被分成三个方面：构造在本地 WGT 模块中的机制，以允许创建和分配不同的外观到不同的 WGT 窗口小部件；本地“LookPixmap”模块的开发，在该模块中包含了用于绘制组件的不同图形特征的函数；以及用于与本机代码接口的 mhw.awt Java 包的开发。

在以后几节里描述这三个方面的开发。

现在将参考图 40 和 41 描述 WGT 模块外观机制。

本节的目的是描述如何对 WGT 模块进行修改以便能够控制被管理的各种图形对象的‘外观’。能够在新类中定义的新外观能够在必要时被添加，而让 WGT 模块不受改变。缺省时向窗口小部件提供、实例化和归属一个标准的 WGT 外观（LookWgt），以便使得不希望使用这个功能的应用程序不需要使用它。

以下特征是必需的：能够为每个对象类型定义许多不同的外观；一个不同的外观能够独立地给予每个对象；以及缺省时，由当前 WGT 定义的外观施加到对象上。

两种用于绘制一个对象外观的不同技术必定是可能的：位图和矢量。

在位图技术中，通过组合许多预定义的位图单元来创建单个随后绘制的位图，来绘制该对象。为了创建可变尺寸的对象，例如单元能够沿着侧边重复，以便构造一个所需尺寸的对象。

这些单元中间每一个的尺寸和精确排列都没有被定义。该想法是让这个尽可能地自由以便使得新外观的创造不过度地受 WGT 的限

制。该外观类必须定义位图单元以及用于把它们装配在一起的规则作为该对象所需尺寸的一个函数。

稍后描述的类 LookPixmap（本地)和 PixmapLook（Java）专门使用这个方法。这并不阻止开发者，无论是否从 LookPixmap 中导出，使用矢量方法来创建他们自己的外观。

在矢量技术中，对象使用一系列基本绘制操作诸如 DrawRectancle ()、DrawArc ()来绘制。该外观类必须定义用于绘制该对象的规则作为该对象所需尺寸的一个函数。

完全有可能设想一个外观类组合了 Bitmap（位图）和 Vectorial（矢量）技术。例如，位图方法能被用于一个按钮的基本形式，而矢量方法能够提供突出显示。

现在更详细地讨论 WGT 模块外观机制

已经包括了以下机制:

- 把当前用于每个对象的绘制方法划分到七个函数中，其中的某些包含在归属于该对象的外观类中：DrawBackground ()；MhwWgtLookDrawBackground () ； DrawForeground () ；MhwWgtLookDrawForeground ()； MhwWgtLookDrawRelief ()；MhwWgtLookDrawFocus ()； MhwWgtLookHighlight ()
- 创建一个抽象类 MhwWgtLook。
- 创建一个从 MhwWgtLook 中导出的类 MhwWgtLookWgt，并且当初始化 WGT 时实例化之。
- 添加一个全局变量 g_TheDefaultLook，该变量用于：当创建一个对象时如果没有一个具体的外观归属于 MhwWgtXXXAttsSetLook，则设置将归属于每个对象的外观。
- 添加一个用于改变对象缺省外观的公共方法 MhwWgtSetDefaultLook (context)。
- 添加两个公共方法到该对象类中：MhwWgtSetXXXAttrLook (* object，* look)以及 MhwWgtGetXXXAttsLook (* object)。

这些方面，从用于绘制的方法开始，在以后几节里给出。

每个绘制函数当它被调用时调用以下方法: DrawBackground ()；MhwWgtLookDrawBackground () ； DrawForeground () ；

MhwWgtLookDrawForeground ()； MhwWgtLookDrawRelief ()； MhwWgtLookDrawFocus ()；以及 MhwWgtLookHighlight ()。

这两个方法 DrawBackground ()和 DrawForeground ()属于 WGT，其与该外观无关而被调用。其它实际上是，指向与计论中的窗口小部件相关联的外观类中的相应函数的指针。以这种方式，该外观类实现了用于这些部分的绘制函数。

Background 这允许该外观在整个窗口小部件后面绘制。

Foreground 这可用于在该窗口小部件中心部分（除边界以外）之上绘制其他图形的一个图像。

Relief 如果该窗口小部件的凸起标记被设置了则调用这个，以及用来绘制用于该窗口小部件的边界或者凸起。

Focus 如果该窗口小部件具有焦点则调用这个。它可用于图形化地指示这个。

Highlight 如果该窗口小部件被突出显示了则调用这个。它可用于图形化地指示这个。

定义了抽象类 MhwWgtLook 并且其包含以下内容：WgtCoreLookClassMethod ； WgtCoreLookClassField ； WgtCoreLookClass； WgtCoreLookPart；以及 WgtCoreLookObject。

这些被描述如下。

```
FIELDS:
MhwWgtLookClass    *extends;
Int32              classId;
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);
MhwWgtErrox        (*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, Card16, 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     (*drawSelectedBG)    (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.
                                         Card32, Int8, MhwWgtLookItemFlag);
    MhwWgtError     (*reDrawItem)        (MhwWgtLook, MhwWgtWidget, Int16,
                                         Int16, Card16, Card16, String,
                                         Card32, 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);
```

（Card8、Card16 等数据类型是用于具有指示位数的数字类型的别名，例如 Card8 相当于一个 'char'，Card16 相当于一个 'short'，等）。

对于每个对象类型，每种绘制方法都是一样的。对于需要一种不同的方法用于每种方法、或者至少用于某些方法的外观，在该外观类中的方法必须标识该窗口小部件类型并且据此作用。

注意方法 DrawNothing ()的目的。如果调用它的话，就简单地返回 OK。某些特征在一个给定的外观中是不必要被实现的。因此 WGT 不必要必须检测一个给定函数的存在，任何未实现的函数都将指向这个方法。

还要注意掩码。这是一个私有、只读的布尔数组，其中的每个元素对应于上述方法中的一个。如果一个元素设置为 1，则重新定义该相应的方法。否则，该方法没有被重新定义。以这种方式，如果 WGT 想要的话，它就能够发现在一次操作中它必须调用哪些方法。

该外观类用来定义在任何外观定义和 WGT 之间的接口。WGT 仅仅使用这些方法来显示需要的外观。如果一个外观需要附加的功能，则它能够被合并到一个扩展的外观结构中，但是由该应用程序而非 WGT 来考虑这些方法／参数。以这种方式，能够添加附加的属性和方法。

一个导出的外观结构必须包含所有这些方法和属性，而且它还可以添加它自己的方法和属性。然而 WGT 将仅仅考虑在 MhwWgtLook 结构中定义的那些方法。

现在讨论类 MhwWgtLookWgt。

所以现有的应用程序不必进行修改就可以保持与修改版本的 WGT 兼容，一个定义 WGT 对象当前所具有的外观的基本外观类由

WGT 创建和实例化。

它是 MhwWgtLook 的一个子类并且被称作 MhwWgtLookWgt。当这类被初始化时，在该结构中所有指针的值被设置为指向 WGT 定义的方法。

这些基本类不包含别的-它仅仅定义 WGT 当前提供的外观。

```
FIELDS:
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         verylightGray;
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      (*setAnchorColorBGCurrent)(MhwWgtLkWgt, MhwWgtColor);
```

对于 WGT 初始化，当启动该 WGT 时必须创建类 MhwWgtLookwgt 的一个实例。如果该应用程序没有指定一个不同的外观的话，WGT 将因此能访问这些方法。g_TheDefaultLook 全局变量（如下所述）最初必须被设置为指向这个外观。

现在描述定义新外观和设置缺省外观的过程。

WGT 不负责定义或者实例化新的外观对象。应用程序必须自己执行这个。所有创建的外观对象在一个 MhwWgtLook 结构中必须是可塑的，参看在下面的外观管理。

对于缺省的外观，字段:

MhwWgtLook * DefaultLook

必须被添加到 MhwWgtContext 对象中，以指向要施加到从这个环境中创建的任何新对象的 MhwWgtLook 实例。当创建一个新的 WGT 环境时，这个字段必须被设置为指向 WgtBasicLook。

对于设置用于一个环境的缺省外观，提供了一个公共方法:

MgwWgtSetDefaultLook (MhwWgtContext , aContext , MhwWqtLook aLook)

以设置在 aContext 中的 DefaultLook 字段以指向 aLook。

为了将一个外观与一个对象关联，以下的属性被添加到在该核心类中的 coreAtts 结构中:

MhwWgtLook * Look

这个属性因此为每个创建的对象而创建。每当一个对象被实例化时，外观被设置为指向 DefaultLook 全局变量。

两个新的公共方法 MhwWgtSetXXXAttsLook (MhwWgtWidget anObject , MhWWgtLook aLook) 和 MhwWgtGetXXXAttsLook (MhwWgtWidget anObject, MhwWgtLook * aLook)将被添加到该核心类中，以允许改变与该对象相联系的外观实例。

现在讨论外观的管理。

WGT 不提供任何外观管理。为了使用一个不同于该缺省的外观，一个应用程序首先必须确保一个或多个外观类被实例化和初始化了，然后每当它创建一个新的 WGT 对象时，使用 MhwWgtSetCoreAttsLook ()方法来把期望的外观与对象相关联。如果它希望使用一个给定的外观用于所有将来的窗口小部件，它能够使用如上所述的 MhwWgtSetDefaultLook ()方法。

一个希望使用任何不同于缺省时定义的一个外观的应用程序负责创建和实例化该外观。在本发明的另一个实施例中，一个应用程序可以远程下载外观。然而，在这种情况下，应用程序本身必须提供从 MhwWgtLook 导出的、需要的类。

一个外观不能由创建它的应用程序破坏直到所有使用它的窗口小部件都已经被破坏了为止。这需要添加一个 refCounter 字段来计数“客户”的数目。

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

这替换了 free (look)。当它的 refCounter 字段等于 0 时实际上将破坏该外观。

```
SetXxxLook(widget, look) {
if(widget->core.look)
   MhwLookUnref(widget->core.look);
widget->core.look=look;
if(look)
   MhwLookRef(look);
}
```

用于实现外观的 WGT 模块 API 列表在这里给出：

```
extern MhwWgtError MhwWgtLookInitDefault        (MhwWgtLookClass*,
                                                 MhwWgtLookAtts*);
extern MhwWgtError MhwWgtLookinitClass          (Void);
```

```
extern MhwWgtError MhwWgtLookResetDefault      (MhwWgtLookClass*);
extern MhwWgtError MhwWgtLookAttsGetBorderWidthBTLR  (MhwWgtLookAtts*,
                                        Card8*, Card8*, Card8*, Card8*);
extern MhwWgtError MhwWgtLookAttsGetBorderWidthBottom
                                        (MhwWgtLookAtts*, Card8*);
extern MhwWgtError MhwWgtLookAttsGetBorderWidthLeft  (MhwWgtLookAtts*,
                                        Card8*);
extern MhwWgtError MhwWgtLookAttsGetaorderWidthRight (MhwWgtLookAtts*,
                                        Card8*);
extern MhwWgtError MhwWgtLookAttsGetBorderWidthTop   (MhwWgtLookAtts*,
                                        Card8*);
extern MhwWgtError MhwWgtLookAttsGetDefault          (MhwWgtLookClass*,
                                        MhwWgtLookAtts*)
extern MhwWgtError MhwWgtLookAttsInit                (MhwWgtLookAtts*);

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

现在将更详细地描述 Look / LookPixmap 模块。

类 MhwWgtbookPixmap 从如上所述的类 MhwWgtLook 中导出。如上所述，实质上它通过重构每个组件的各种单元来创建一个所需尺寸的图形图像来进行工作。

这些图像在下列中使用：按钮背景；凸起（例如，就是沿着文本区域活动区的边界）；一个用于（选择）组件的符号；复选框；滑动块；和滑动块的移动按钮。

参考图 40，为了减少初始化时间，图像不被压缩，但是以一种设计成尽可能最小化所占据空间的特定格式进行保存。每一个像素（2152）的颜色在单个字节中描述，其为在当前颜色映射中颜色的索引号（2151）。图 40 显示了一个示例缓冲区，image，其包含一个 4x4 图像（2153）。

在图 40 中的图像（2153）将如下所示存储：

```
Card8 slidLiftSeVrImage4 [4] [4] =
{
{0, 0, 0, 1},
{0, 0, 1, 2},
{0, 1, 2, 3},
```

{1，2，3，4}

};

现在将描述 LookPixMap 图像结构。

为了识别一个图像缓冲区的尺寸，结构 LookPixmapImage 被定义为包含如上所述的图像缓冲区，链同该图像的宽度和高度。这个在下面定义的结构用来包含用于每个图形元素的数据。

typedef struct

{Card8 *imageData;	包含该图像的缓冲区。它是 Card8 的二维数组，其中的每个单元都包含要在该数组中该位置处显示的颜色在该颜色调色板中的索引号。
Card8 *maskData;	包含该掩码的缓冲区。它是一维 Card8 数组，每个像素一位（和由该图像模块返回的格式相同）。
Card16 width;	图像的宽度。
Card16 height;	图像的高度。
Card8 is0paque;	如果图像包含任何透明元素则为 0，如果完全不透明为 1。

}

LookPixmapImage;

图像能够具有不同的尺寸，虽然对于一个给定类型的单元，它们通常将是相同的。然而中心单元（xxxxxC）经常为尺寸 1 x 1。MhwWgtLookPixmapAllImages 结构把所有图像单元集合起来，其如下所示：

```
typedef struct
{
   LookPixmapImage  *relnoNW;          NorthWest corner of relicf - state
                                       Normal
   LookPixmapImage  *relnoSW;          SouthWest corner of relicf - state
                                       Normal
   LookPixmapImage  *relnoNE;          NorthEast corner of relief - statc
                                       Normal
```

```
        LookPixmapImage  *relnoSE;        SouthEast corner of relief - state
                                          Normal
        LookPixmapImage  *relnoN;         North edge of relief - state Normal
        LookPixmapImage  *relnoW;         West edge of relief - state Normal
        LookPixmapImage  *relnoE;         East edge of relief - state Normal
        LookPixmapImage  *relnoS:         South edge of relief - state Normal
        LookPixmapImage  *relnoC;         Central area of relief - state Normal
        LookPixmapImage  *relfoNW;        State - Focus only
        LookPixmapImage  *relfoSW;
        LookPixmapImage  *relfcNW;
        LookPixmapImage  *relfoSE;
        LookPixmapImage  *relfoN;
        LookPixmapImage  *relfoW;
        LookPixmapImage  *relfoE;
        LookPixmapImage  *relfoS;
        LookPixmapImage  *relfoC;
        LookPixmapImage  *relhiNW;        State - Highlight only
        LookPixmapImage  *relhiSW;
        LookPixmapImage  *relhiNE;
        LookPixmapImage  *reihiSE;
        LookPixmapImage  *relhiN;
        LookPixmapImage  *relhiW;
        LookPixmapImage  *relhiE;
        LookPixmapImage  *relhiS;
        LookPixmapImage  *relhiC;
        LookPixmapImage  *relfhNW;        State - Focus and Highlight
        LookPixmapImage  *relfhSW;
        LookPixmapImage  *rolfhNE;
        LookPixmapImage  *relfhSE;
        LookPixmapImage  *relfhN;
        LookPixmapImage  *relfhW;
        LookPixmapImage  *relfhE;
        LookPixmapImage  *relfhS:
        LookPixmapImage  *relfhC;
        LookPixmapImage  *butnoNW;        Button elements
        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  *butfhE;
        LookPixmapImage  *butfhS;
        LookPixmapImage  *butfhC;
        LookPixmapImage  *choice;          Choice Symbol
        LookPixmapImage  *chck1na,         Checkbox symbol - Type 1 not selected
                                           no focus
        LookPixmapImage  *chck1a;          Checkbox symbol - Typc 1 selected no
                                           focus
        LookPixmapImage  *chckf1na;        Checkbox symbol - Type 1 not selected
                                           focus
        LookPixmapImage  *chckf1a:         Checkbox symbol - Type 1 selected
                                           focus
        LookPixmapImage  *chck2na;         Checkbox symbol - Type 2 not selected
                                           no focus
        LookPixmapImage  *chck2a;          Checkbox symbol - Type 2 selected no
                                           focus
        LookPixmapImage  *chckf2na;        Checkbox symbol - Type 2 not selected
                                           focus
        LookPixmapImage  *chckf2a;         Checkbox symbol - Type 2 selected
                                           focus
        LookPixmapImage  *slidNeVr;        Slider background elements
        LookPixmapImage  *slidEVr;
        LookPixmapImage  *slidSeVr:
        LookPixmapImage  *slidSwHr:
        LookPixmapImage  *slidSHr;
        LookPixmapImage  *slidSeHr;
        LookPixmapImage  *slidSeVrHr;
        LookPixmapImage  *slidLiftNeVr;    Slider lift elements
        LookPixmapImage  *slidLiftEVr;
        LookPixmapImage  *slidLiftSeVr,
        LookPixmapImage  *slidliftSwHr;
        LookPixmapImage  *slidLiftSHr;
        LookPixmapImage  *slidLiftSeHr;
}
MhwWgtLookPixmapAllImages;
```

本节描述了 LookPixmap 模块，其包含类 LookPixmap，已经创建了该类以允许一组不同的外观应用到网络浏览器上。

这个模块包含下列源文件：MhwWgtLookPixmap.h; MhwWgtLookPixmapStruct.h; WgtLookPixmapClass.c; MhwWgtLookPixmapImages.h; MhwWgtLookPixmapImages2.h; MhwWgtLookPixmapImages3.h; MhwWgtLookPixmapImages4.h; MhwWgtLookPixmapImages5.h; 以及 MhwWgtLookPixmapImages6.h。

现在将描述 LookPixmap 模块，包含创建和使用 LookPixmap 对象的最佳方法的详情。

任何使用 WGT 用于创建和管理窗口小部件的软件能够使用该 LookPixmap 模块以提供可替换的外观到该 WGT 窗口小部件。对于一个使用该 LookPixmap 外观的应用程序，必须创建一个 LookPixmap 对象。这能够使用以下代码来执行：

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

WgtErr = MhwWgtLkWebInitClass();
WgtErr = MhwWgtLkWebAttsInit(&LookPixmapValues);
WgtErr = MhwWgtLkWebInitDefault(&PixmapLook, &LookPixmapValues);
PixmapLookObject = MhwWgtLkWebNew(&LookPixMapValues);
```

现在描述一种用于设置该缺省外观的方法。

一个应用程序缺省时能够使用一个给定的外观对象。缺省时，该缺省外观是由 WGT 创建的 LookWgt 对象。为了设置另一个缺省外观，在它已经如上所述被创建了的情况下，能够使用下列函数：

MhwWgtSetDefaultLook((MhwWgtLook)PixmapLookObject);

所有后续创建的 WGT 窗口小部件将与该 LookPixmap 外观类相关联，而不是与 WGT 缺省的 LookWgt 相关联。

一个应用程序能够选择或者设置用于一个给定类型的窗口小部件、或者一个给定窗口小部件的外观，如现在将描述的那样。

当创建该窗口小部件时，一个应用程序能够设置用于一个给定窗

口小部件的外观，这是通过就在创建该窗口小部件之前调用以下函数来完成:

MhwWgtXXXAttsSetLook (MhwWgtXXXAtts* , MhwWgtLook);

它还可以在创建一个对象之后设置一个对象的外观，这使用以下函数:

MhwWgtXXXSetLook (MhwWgtXXXWidget *, MhwWgtLook);

（其中 xxx 是窗口小部件的类型-例如 LIST）。

现在将描述使用 LookPixMap 图像的方法。

单个 LookPixmap 对象使用单个图像组。你能够简单地通过改变该图像来显著明显地改变该外观。

你能够通过调用以下函数来改变用于一个给定 LookPixmap 的图像:

MhwWgtLookPixmapSetImages(MhwWgtLookPixmap * , MhwWgtLookPixmapAllImages*);

设置在所有使用该指定 LookPixmap 对象的窗口小部件中使用的图像到指定的图像组。

MhwWgtLookPixmapSetDefaultImages (MhwWgtLookPixmap*);

设置在所有使用该指定 LookPixmap 对象的窗口小部件中使用的图像到缺省的图像组。

如果你希望使用不同的图像用于不同的窗口小部件，你必须为每个需要的图像组创建一个 LookPixmap 对象。你然后分配每个图像组到适当的外观，然后将每个外观关联到适当的窗口小部件。

现在将参考图 39 和 41，描述 LookPixmap 模块的 API。

以下公共的 API 是可用的:

MhwWptLookPixmapSetImages()

原型:

MhwWgtError MhwWgtLookPixmapSetImages (MhwWgtLkWebaLook,

MhwWgtLookpixmapAllImages*

someImages);

描述:

把由 aLook 使用的图像组设置为由 someImages 指向的图像组。

参数:

aLook MhwWgtLkWeb 对象，关联图像 someImages 到该对象

someImages 与 aLook 关联的图像组。

返回:

MHW_WGT_SUCCESS

MhwWgtLookPixmapSetImagesID()

原型:

MhwWgtError MhwWgtLookPixmapSetImagesID (MhwWgtLkWeb aLook，Card8 anImageID);

描述:

把由 aLook 使用的图像组设置为在 MhwWgtLookPixMap 中硬编程而且由 anImageID 标识的图像组。

参数:

aLook MhwWgtLkWeb 对象，关联由 anImageID 标识的图像到该对象。

anImageID 在 MhwWgtLookPixmap 中硬编程的与 aLook 关联的图像组标识符。

返回:

MHW_WGT_SUCCESS

MhwWgtLookPixmapSetDefaultImages()

原型:

MhwWgtError MhwWgtLookPixmapSetDefaultImages (MhwWgtLkWeb aLook);

描述:

把由 aLook 使用的图像组设置为在 MhwWgtLookPixMap 中硬编程而且由标识符 1 标识的图像组。

参数:

aLook MhwWgtLkWeb 对象，关联由 anImageID 标识的图像到该对象。

返回:

MHW_WGT_SUCCESS

MhwWgtLookPixmapLoadImage()

原型:

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

描述:

用于把由当前 MhwWgtLookPixmapAllImages 结构指向的单个图像改变到指向指定的图像。创建一个 LookPixmapImage 结构，而且把由 aLook 指向的当前 MhwWgtLookPixmapAllImages 设置为指向由 anElementID 指定单元的 LookPixmapImage。

参数:

aLook MhwWgtLkWeb 对象，关联由 anImageID 标识的图像到该对象。

anElementID 要改变单元的标识符。

aWidth 新图像的宽度，以像素为单位。

aHeight 新图像的高度，以像素为单位。

anImageBuffer 包含该新的图像数据的缓冲区。

返回:

MHW_WGT_SUCCESS

LookPixmapMakeImageFromElements()

原型:

```
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);
```

描述:

使用九个 elemX 图像并且在指定的 MHW 窗口中绘制它们。用于构造该图像的规则由 aDrawMode 指定(当前仅仅存在 MHW_WGT_LIKWEB_DRAW_NORMAL)。在该窗口中绘制的最终图像的左上角位于(anX, aY)处并且具有尺寸 aWidth × aHeight。

如果一个或多个单元具有零尺寸(要么 elemXX.width 要么 elemXX.height 为零),则不绘制这些单元。

参数:

elemN, elemE, elemW, elemS, elemNW, elemNE, elemSW, elemSE, elemC: 分别在上面、右边、左边、底端、左上角、右上角、左下角、右下角和中间处绘制的图像。

anX 在窗口 aWindow 中,绘制该最终图像的 x 位置。

aY 在窗口 aWindow 中,绘制该最终图像的 y 位置。

aWidth 新图像的宽度,以像素为单位。

aHeight 新图像的高度,以像素为单位。

aWindow 在其中绘制该构造图像的窗口。

aDrawMode 构造该图像的模式。

MHW_WGT_LKWEB_DRAW_NORMAL: 把 NW(2100)、NE(2101)、SW(2102)和 SE(2103)单元放置在四角(2200、2201、2202、2203)而不用平铺。横向平铺 N 和 S 单元。纵向平铺 W 和 E 单元。横向和纵向平铺 C 单元。尽管它对任何的图像尺寸都可以用,但是它仅仅保证如果该中心区域(2208)是矩形时,能够正确地平铺

该区域。

返回:

MHW_WGT_SUCCESS

现在将描述 Mhw.awt Java 界面包。

已经开发了三个 Java 类以便在 WGT 中定义的外观机制能够由 java 应用程序利用。这些是: mhw.awt.Look; mhw.awt.WgtLook; 和 mhw.awt.PixmapLook。

该外观类是对应于如上所述 MhwWgtLook 类的抽象类。

类 WgtLook 用来创建和处理 WGT 类 MhwWgtLookWgt 的实例。

PixmapLook 类用来存储由类 WgtLook 使用的图像。

现在将从构造器的详情开始描述 Mhw.awt.PixmapLook API。

PixmapLook

public PixmapLook()

创建一个 PixmapLook 对象的一个新实例，让该图像被初始化为缺省的图像（ID = 1）

PixmapLook

public PixmapLook(int imageID)

创建 PixmapLook 对象的一个新实例，让该图像被初始化为由 imageID 指定的图像。

现在将描述方法。

SetImages

public void SetImages()

把用于该 PixmapLook 对象的当前图像设置为缺省（ID = I）

SetImages

public void SetImages (int imageID)

把用于该 PixmapLook 对象的当前图像设置为由 imageID 指定的图像

LoadImage

public void LoadImage(int elementID,

int width,

int height,

byte [] buffer)

加载一个指定的图像单元。每个 PixmapLook 对象具有一组（94）与它相关联的图像。这些图像表示如下的图形组件单元：0 relnoNW; 1 relnoSW; 2 relnoNE; 3 relnoSE; 4 relnoN; 5 relnoW; 6 relnoE; 7 relnoS; 8 relnoC; 9 relfoNW; 10 relfoSW; 11 relfoNE; 12 relfoSE, 13 relfoN; 14 relfoW; 15 relfoE; 16 relfoS; 17 relfoC; 18 relhiNW; 19 relhiSW; 20 relhiNE; 21 relhiSE; 22 relhiN; 23 relhiW; 24 relhiE; 25 relhiS; 26 relhiC; 27 relfhNW; 28 relfhSW; 29 relfhNE; 30 relfhSE; 31 relfhN; 32 relfhW; 33 relfhE; 34 relfhS; 35 relfh C; 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 slidLiftSwHr; 92 slidLiftSHr; 93 slidLiftSeHr.

width 指定传递的图像宽度。

height 指定传递的图像高度。

buffer 包含该图像数据。这以一个字节数组形式，每个字节为每一个像素给定使用的颜色映射。用于像素（x，y）的索引是缓冲区[（y * width）+x]。

MakeImageFromElements

```
public void       MakeImageFromElements (int[] widths,
                                         int[] heights,
                                         byte[][] buffers,
                                         int anX,
                                         int aY,
                                         int aWidth,
                                         int aHeight,
                                         java.awt.Component aComponent)
```

基于 9 个单元（N、E、W、S、NW、NE、SW、SE、C）构造一个图像并且在与组件 aComponent 相关联的窗口上绘制它，让它的左上角在（anX，aY）处而且具有尺寸 aWidth x aHeight。该图像缓冲区作为一个二维数组被传递，一维给定图像数字（对应于 N、E、W、S、NW、NE、SW、SE、C 的 0 - 8）而且另一维包含数据。以数组宽度和高度给出每个缓冲区的宽度和高度。

DownloadLookDir

public java.lang.String DownloadLookDir()

从 MPEG 流中下载一个“外观”目录。返回一个包含有关每个能够被下载的外观图像组的信息的字符串，该字符串由换行符分隔。一个标题的行号（0 到 n - 1）对应于函数 DownLoadLookImages (int ImageSet)使用的标识符（ImageSet）。

目录实际上是包含被返回的字符串的一个简单文本文件。该文件路径在源文件中被硬编码 - 当前是 /home/users/mstoddar/mhplus/util/looks/images.dir。这能够被适当地改变。这因此能在解码器中使用，以自动地从 MPEG 流中下载。

该文件的格式为:

<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>

如果不成功的话则返回“”。

DownloadLookImages

public void DownloadLookImages (int ImageSet)

从 MPEG 流下载一个新的图像组，由在 DownLoadLookDir ()中返回的一个条目中的行号标识，并且把它们归属于这个外观。

该文件包含以以下格式的数据:

WWWWHHHHWWWWHHHH ... 一系列四字节字符串（有引

导空格）包含所有 94 个图像的宽度和高度的十进制数值（和在方法 LoadImage ()中的次序相同）。用于每个图像的数据缓冲区再次与 LoadImage ()的格式相同。在图像之间不进行调准，下一个图像的开始从接着前一个图像的字节开始。

该文件路径在源文件中被硬编码-当前是/home/users/mstoddar/mhplus/util/looks/ Images.<ImageSet>。 这能够被适当地改变。这因此能在解码器中使用，以自动地从 MPEG 流中下载。

public void DownloadLookImages (string ImageURL)

从该 MPEG 流中下载一个新的图像组，其有指定的 URL 标识，而且把它们归属于这个外观。

该文件以上面给定的格式包含数据。

为了清楚起见，在下面给出的文件结构以 C 语言的语法格式:

```
Card8 relnoNWwidth[4]   String representation of decimal value (leading
                                        spaces)
Card8 relnoNWheight[4]
Card8 relnoSWwidth[4]
Card8 relnoSWheight(4]
Card8 relnoNEwidth[4]
Card8 relnoNEheight[4]
Card8 reinoSEwidth[4]
```

```
Card8 relnoSEheight[4]
Card8 relnoNwidth[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 relfoSWheight[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 relhiSWheight[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 relfhSWheight[4]
```

```
    Card8 relfhNEwidth[d]
    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 butnoSWheight[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 butfoSWheight[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 butfoCwidth[4]
    Card8 butfoCheight[4]
    Card8 buthiNWwidth[4]
```

```
Card8 buthiNWheight[4]
Card8 buthiSWwidth[4]
Card8 buthiSWheight[4]
Card8 buthiNEwidth[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 burfhSWheight[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 chcklsewidth[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 chck2faheight[4]
Card8 slidNeVrwidth[4]
Card8 slidNeVrheight[4]
Card8 slidEVrwidth[4]
Card8 slidEVrheight[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 slidLiftEVrheight[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 x height]
Card8 RelnoSWbuffer[width x height]
Card8 RelnoNEbuffer[width x height]
Card8 RelnoSEbuffer[width x height]
Card8 RelnoNbuffer[width x height]
Card8 RelnoWbuffer[width x height]
Card8 RelnoEbuffer[width x height]
Card8 RelnoSbuffer[width x height]
Card8 RelnoCbuffer[width x height]
Card8 RelfoNWbuffer[width x height]
Card8 RelfoSWbuffer[width x height]
Card8 RelfoNEbuffer[width x height]
Card8 ReltoSEbuffer[width x height]
Card8 RelfoNbuffer[width x height]
Card8 RelfoWbuffer[width x height]
Card8 RelfoEbuffer[width x height]
Card8 RelfoSbuffer[width x height]
Card8 RelfoCbuffer[width x height]
Card8 RelhiNWbuffer[width x height]
Card8 RelhiSWbuffer[width x height]
Card8 RelhiNEbuffer[width x height]
Card8 RelhiSEbuffer[width x height]
```

```
Card8 RelhiNbuffer[width x height]
Card8 RelhiWbuffer[width x height]
Card8 RelhiEbuffer[width x height]
Card8 RelhiSbuffer[width x height]
Card8 RelhiCbuffer[width x height]
Card8 RelfhNWbuffer[width x height]
Card8 RelfhSWbuffer[width x height]
Card8 RelfhNEbuffer[width x height]
Card8 RelfhSEbuffer[width x height]
Card8 RelfhNbuffer[width x height]
Card8 RelfhWbuffer[width x height]
Card8 RelfhEbuffer[width x height]
Card8 RelfhSbuffer[width x height]
Card8 RelfhCbuffer[width x height]
Card8 ButnoNWbuffer[width x height]
Card8 ButnoSWbuffer[width x height]
Card8 ButnoNEbuffer[width x height]
Card8 ButnoSEbuffer[width x height]
Card8 ButnoNbuffer[width x height]
Card8 ButnoWbuffer[width x height])
Card8 ButnoEbuffer[width x height]
Card8 ButnoSbuffer[width x height]
Card8 ButnoCbuffer[width x height]
Card8 ButfoNWbuffer[width x height]
Card8 ButfoSWbuffer[width x height]
Card8 ButfoNEbuffer[width x height]
Card8 ButfoSEbuffer[width x height]
Card8 ButfoNbuffer[width x height]
Card8 ButfoWbuffer[width x height]
Card8 ButfoEbuffer[width x height]
Card8 ButfoSbuffer[width x height]
Card8 ButfoCbuffer[width x height]
Card8 ButhiNWbuffer[width x height]
Card8 ButhiSWbuffer[width x height]
Card8 ButhiNEbuffer[width x height]
Card8 ButhiSEbuffer[width x height]
Card8 ButhiNbuffer[width x height]
Card8 ButhiWbuffer[width x height]
Card8 ButhiEbuffer[width x height]
Card8 ButhiSbuffer[width x height]
Card8 ButhiCbuffer[width x height]
Card8 ButfhNWbuffer[width x height]
Card8 ButfhSWbuffer[width x height]
Card8 ButfhNEbuffer[width x height]
Card8 ButfhSEbuffer[width x height]
Card8 ButfhNbuffer[width x height]
Card8 ButfhWbuffer[width x height]
Card8 ButfhEbuffer[width x height]
Card8 ButfhSbuffer[width x height]
Card8 ButfhCbuffer[width x height]
Card8 Choicebuffer[width x height]
```

```
Card8 chcklnobuffer[width x height]
Card8 chcklsebuffer[width x height]
Card8 chcklfobuffer[width x height]
Card8 chcklfsbuffer[width x height]
Card8 chck2nobuffer[width x height]
Card8 chck2sebuffer[width x height]
Card8 chck2fobuffer[width x height]
Card8 chck2fsbuffer[width x height]
Card8 slidNeVrbuffer[width x height]
Card8 slidEVrbuffer[width x height]
Card8 slidSeVrbuffer[width x height]
Card8 slidSwHrbuffer[widch x height]
Card8 slidSHrbuffer[width x height]
Card8 slidSeHrbuffer[width x height]
Card8 slidSeVrHrbuffer[width x height]
Card8 slidLiftNeVrbuffer[width x height]
Card8 slidLiftEVrbuffer[width x height]
Card8 slidLiftSeVrbuffer[width x height]
Card8 slidLiftSwHrbuffer[width x height]
Card8 slidLiEtSHrbuffer[width x height]
Card8 slidLiftSeHrbuffer[width x height]
```

例如:

	宽	高	宽	高
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	8
0000200	8	8	1	1
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
```

缓冲区 1 数据的开始(relnoNW) (8x8 字节)

```
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  001 001 001 001 001 006
```

缓冲区 2 数据的开始（relnoSW）(8x8 字节）

```
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  001 001 001 001 001 270 \L  265
0001540 001 001 001 001 001 270 \t  \a  006 001 001 001 001 270 \C  006
```

缓冲区 3 数据的开始（relnoNE）(8x8 字节）

```
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 \O  \0  \0  006 022 265 265 \t  \0  \0  \0  \0  \0  \a  006 006
```

缓冲区 4 数据的开始（relnoSE）（8x8 字节）

```
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
```

等。

reDrawAll

```
public void        reDrawAll()
```

查找具有焦点的窗口小部件，然后查找其母体直到没有为止。顶端窗口然后设置为看不见并且再次设置为可见。然后整个窗口将被重新绘制。

上面描述的用于一个或多个图形对象、用于在多个这样的对象之间导航，或者用于从用户接收输入的各种方法可以同样地被应用于其他区域，这些区域主要是，然而并非仅仅是，在从一个广播供应商接收广播的环境中。通常，一个涉及与一个用户可视地相互作用的机顶盒的任何功能都可以使用这样的方法。

例如，一个可能具有附加子链的、可导航的图标链能够在一个家庭购物应用程序中使用，以允许该用户显示物品、查看价格、订货及其他，与该应用程序相互作用。用于定货的图形对象可以，当突出显示时，以上面描述的方式在购买符号（例如，一个美元符号，$）和表示迄今花费的数量的文本、或者以该订户语言的单词“购买”之间自动地‘翻转’。做为选择或者另外，每当选择该‘购买’图标时，能够显现一个包含以该订户语言的单词‘购买’的图形对象，并且提供一个用于让任何子链‘悬挂’的转移。

在上述示例中的‘购买’图标可以紧接着一个当点击时查看迄今为止进行的购买列表的图标放置，或者紧接着另一个当点击时设置用于刚刚购买的产品的交货选择的图标放置，以便在用户能够导航的链中提供一个逻辑次序的图标。当选择了‘购买’图标时，能够显现一个具有各种辅助选项的子链，其在更昂贵物品的情况下可以包含不同的信贷计划。任何需要来自该用户的文本信息，诸如交货的街道地址，能够利用该虚拟键盘输入。

在一个电子节目指南中，能够使用类似的方法，用于交互式地浏览并且显示不同的频道、主题以及时间和日期。依据该用户的首选项重新布置在该链中的图形选项的进一步定制化是可能的；在一个频道链表的情况下，用户的偏爱频道能够被组合在该链的头部。这样的一个参数选择能够由该用户指示，或者由该节目推导出。

用于如上所述方法的其他应用程序包含在线目录表、按请求提供的新闻和气象服务、游戏、以及机顶盒的整体管理（管理它的配置，等）。在游戏的情况下，能够使用头尾翻滚效果以提供游戏中的活动性而不必需要编写附加的方法，而且该虚拟键盘能够被用作用于更高级类型游戏的控制器的一种代替方式。

同时将要理解：所有如在这里描述的那样，使用一个遥控装置交互作用的方法，可以通过利用一个鼠标（或者其他指向控制器、诸如一个滚球或者操纵杆）和／或键盘（或者其他具有多个键的设备）来模拟一个遥控装置的按钮（例如使用在键盘上的数字 0 - 9；箭头键和返回键）或者直接地进行（例如使用该鼠标点击按钮，以及键盘来直接输入文本而不是使用该虚拟键盘）来代替或者补充。

如上所述的虚拟键盘，例如，可以在任何具有多个键的设备，诸如一个游戏机或者一个移动电话上实现。在后者的情况下，该虚拟键盘能够如描述的那样充分地显示在电话的屏幕上（在具有一个足够大显示屏的电话上），或者以一个压缩形式（在具有较小显示屏的电话上）显示。该虚拟键盘的这种压缩可能要求一次仅仅显示字符的一个数字块，其更可取地是具有一个可以通过按压向左、向右、向上和／或向下键（或者它们的等效体，例如在滚动类型的指向控制器情况下）来访问的字符建议或者字符类型。该压缩的虚拟键盘可以在其他应用程序中使用，尤其是在只有少量的可用空间来显示该键盘的地方。

术语“校验框”可以涉及一个任何形状的图形对象，例如圆盘，其能够显示不同的状态，更可取地是对应于‘选中’和‘未选中’的两个状态，但是有可能多于两种状态，而且当由该用户点击或者选择时，其可以以一种一致的方式改变它的状态。‘选中’状态可以由在该方框上的点、交叉或者其他装饰来指示。

便于参考，下面列出的、在这里使用的术语具有以下的最佳意思：

HTML :超文本标志语言，一种描述在国际互链网上互换的文档

的语言。该文档可以包含到站点的引用，格式化信息，声音和图片，等。

HTTP 超文本传输协议，一种用于在保持 HTML 文档和一个显示该 HTML 文档的导航应用程序的 Internet 服务器之间进行通信的协议。

MPEG - 2: 动画专家小组，一种实时编码动画图像和声音的方法。

PPP: 点到点协议，一种允许两个计算机经由一个调制解调器联网的远程访问通信协议。

PROXY SERVER: 一个位于该服务器上、允许安全的 Internet 联接的应用程序，而且它还缓存 HTTP 和 FTP 请求。

SESSION: 在时间中一个给定点处，在存储器中的一种类型链接或者一个应用程序的一个实例。

URL :统一资源定位，一个用于在 Internet 上定位一个文件或者资源的地址。到一个站点的链接，指定包含在该网页中的资源的地址。

WWW: 万维网，使用本地或者远程文档的 Internet 网络。一个网络文档是一个网页，而且在该页中的链接允许在不同的页之间和在不同的主题之间导航，而不管它是否位于一个本地或者远程网络上。

GUI: 图形用户界面。

WGT: 窗口小部件工具包。

应当理解: 在上面已经仅仅通过示例对本发明进行了描述，能够在本发明的范围内对细节做出修改。

在该描述中公开的每个特征，以及（在适当处）的权利要求和附图可独立地提供或者以任何适当组合的形式提供。

在任何或者所有的上述中，本发明的某些特征已经使用计算机软件实现了。然而，本领域的技术人员当然理解任何这些特征可以使用硬件或者硬件和软件的组合来实现。此外，容易理解: 由硬件、计算机软件、等等执行的函数在电和类似信号上或者使用电和类似信号执行。

涉及信息存储的特征可以由适当的存储单元或者存储器实现。涉及信息处理的特征可以由一个适当的处理器或者控制装置，以软件或者以硬件或者以两者的组合方式实现。

在任何或者所有上述中，本发明可以在任何、某些或者所有以下

形式中体现：它可以在一种操作一种计算机系统的方法中体现；它可以在计算机系统本身中体现；它可以在当一个计算机系统被编程或者修改或者安排来执行操作那个系统的方法时，在该计算机系统中体现；和／或它可以在一个其中记录了一个程序的计算机可读存储介质中体现，其中该程序适于依据操作该系统的方法进行操作。

如在这里一直使用的那样，术语“计算机系统”可以与“计算机”、“系统”、“设备”、“装置”、“机械”和类似术语互换使用。

在该权利要求中出现的参考数字仅仅用于说明，而且应当不具有在该权利要求范围上的限制作用。

申请者因此声明，为了避免引起争议，他申请在附图中的版权。

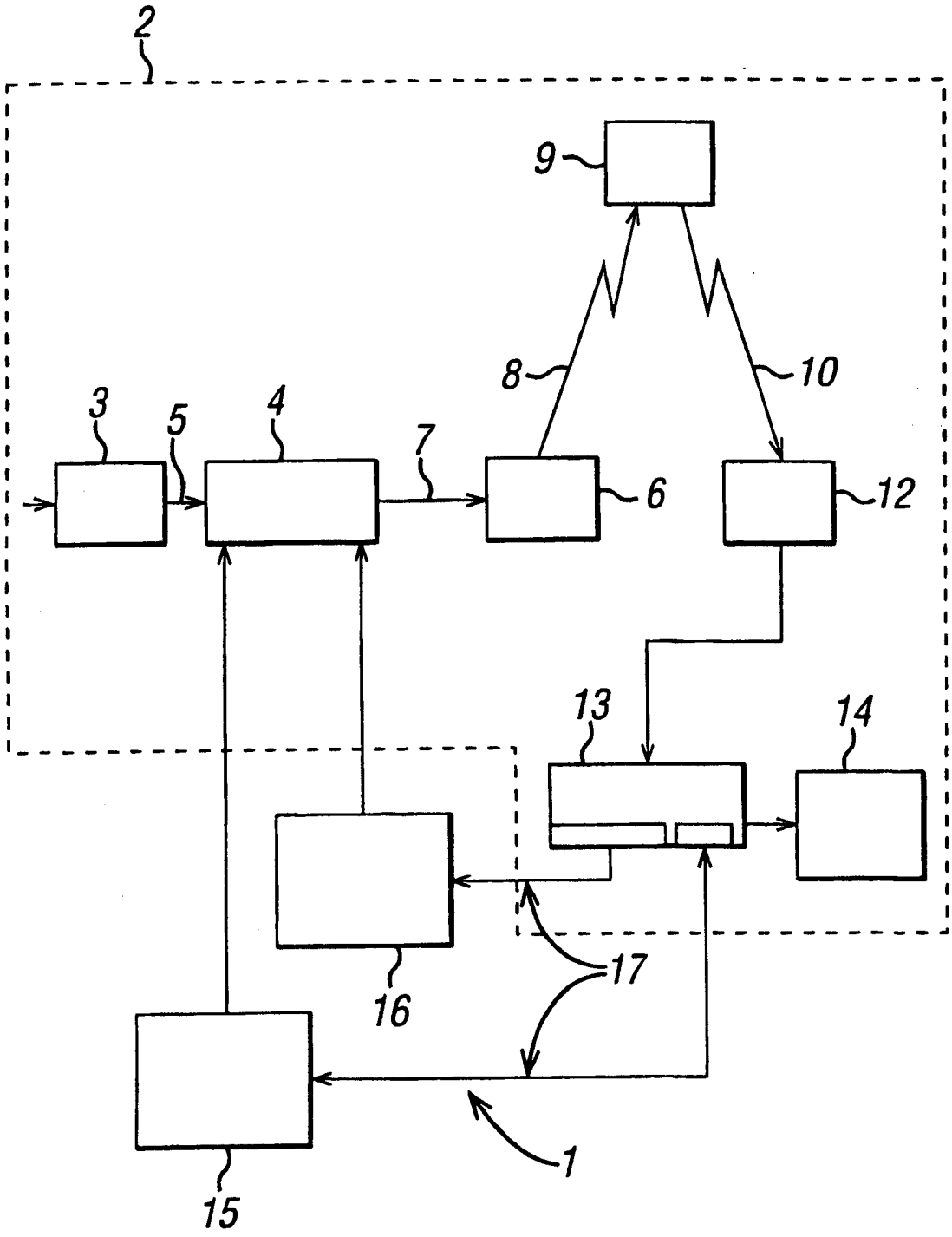


图 1a

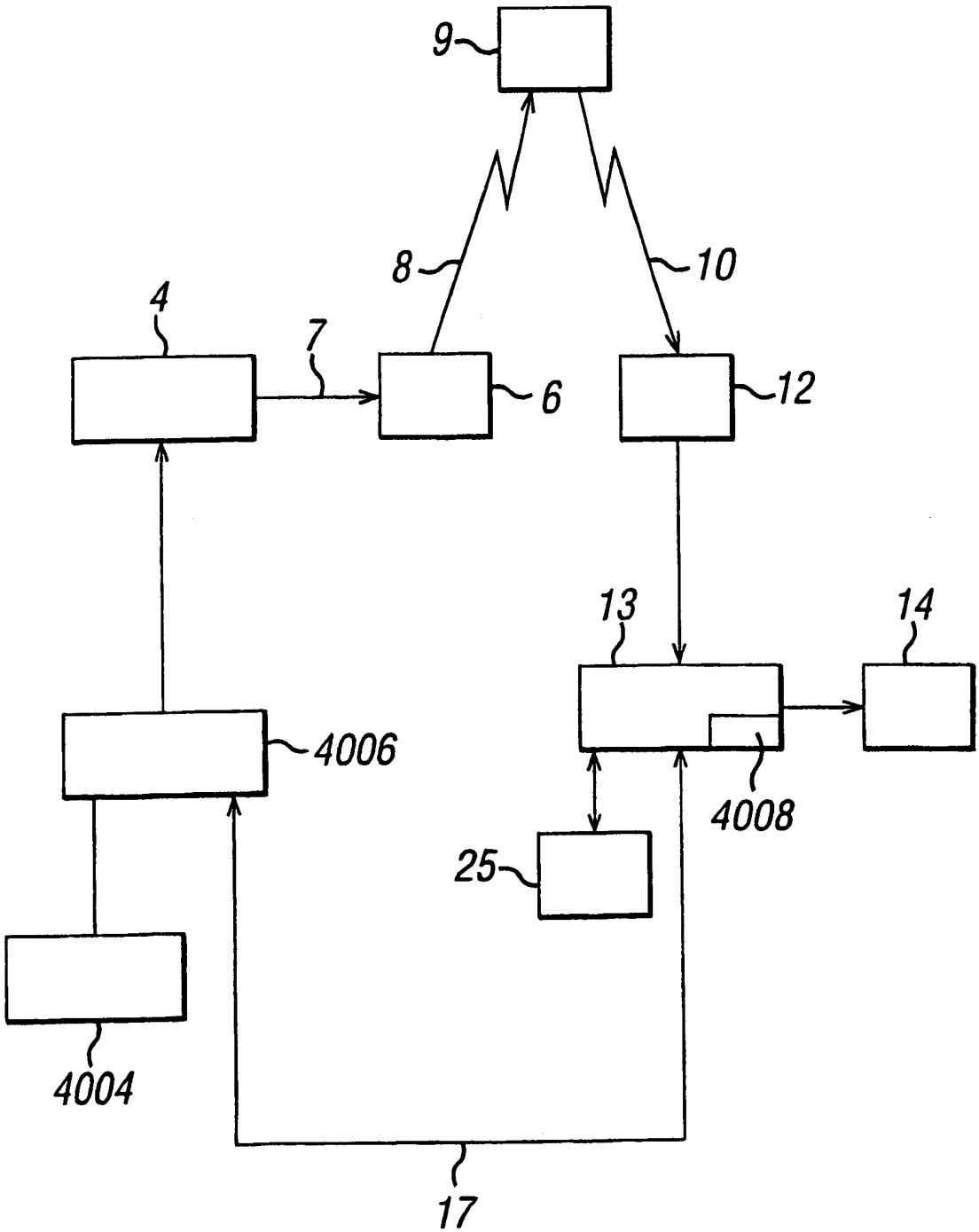


图 1b

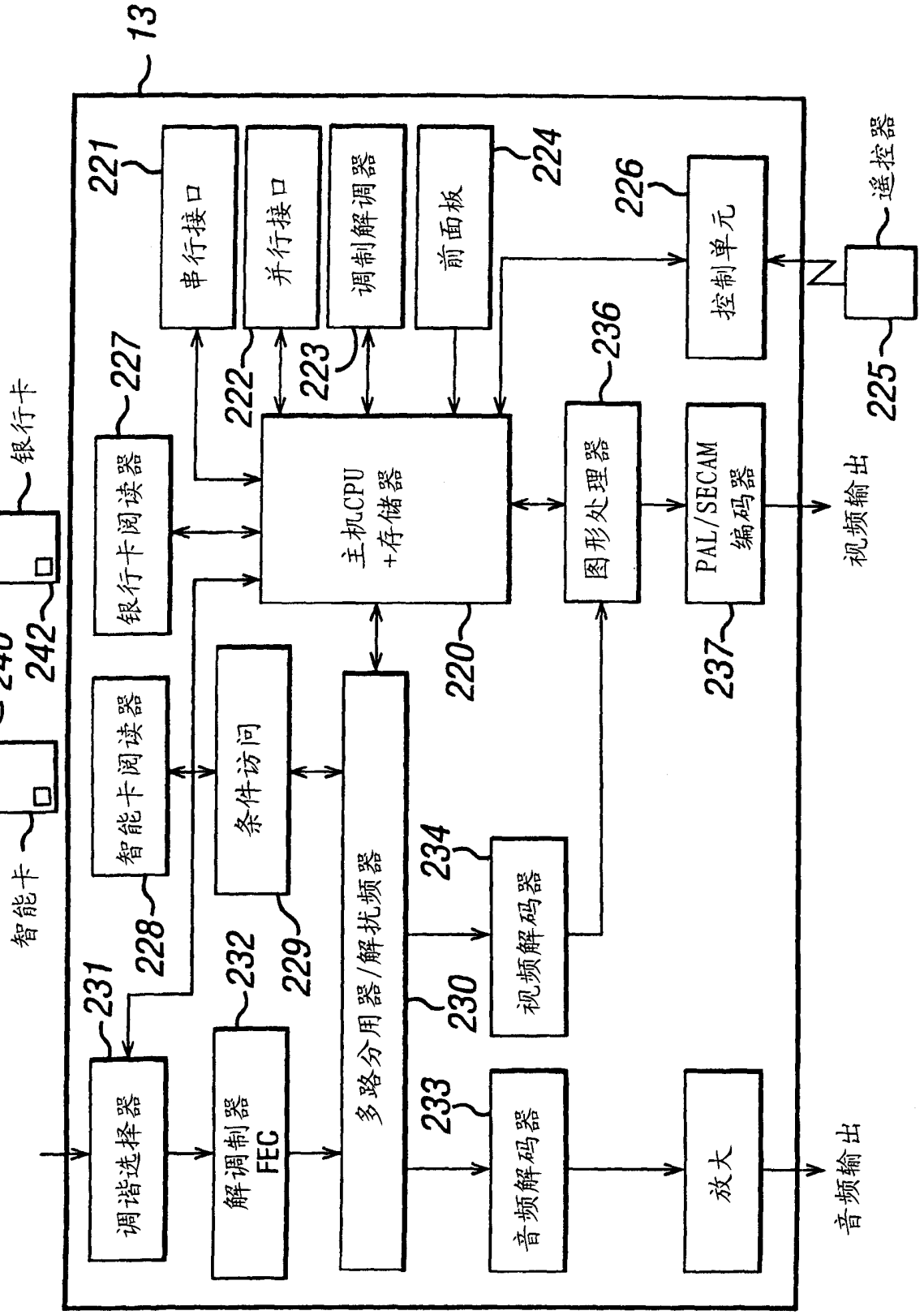

13
智能卡
240
242
银行卡
调谐选择器
231
解调制器 FEC
232
多路分用器/解扰频器
智能卡阅读器
228
条件访问
229
银行卡阅读器
227
主机CPU +存储器
220
串行接口
221
并行接口
222
调制解调器
223
前面板
224
控制单元
226
遥控器
225
图形处理器
236
PAL/SECAM 编码器
237
视频输出
视频解码器
234
音频解码器
233
放大
230
音频输出

图 2a

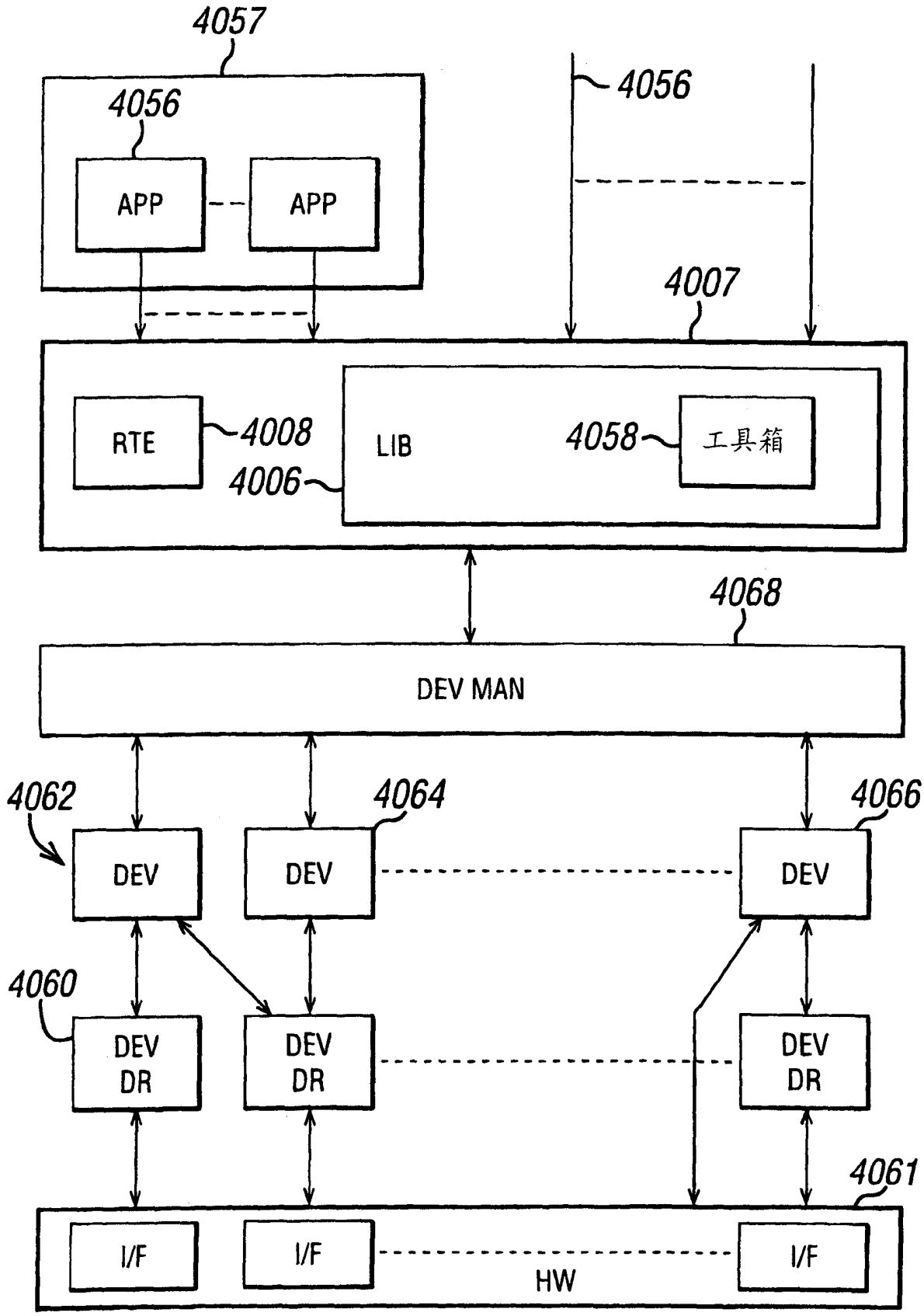

4057
4056
APP
APP
4056
4007
RTE
4008
4006
LIB
4058
工具箱
4068
DEV MAN
4062
4064
4066
DEV
DEV
DEV
4060
DEV DR
DEV DR
DEV DR
4061
I/F
I/F
I/F
HW

图 2b

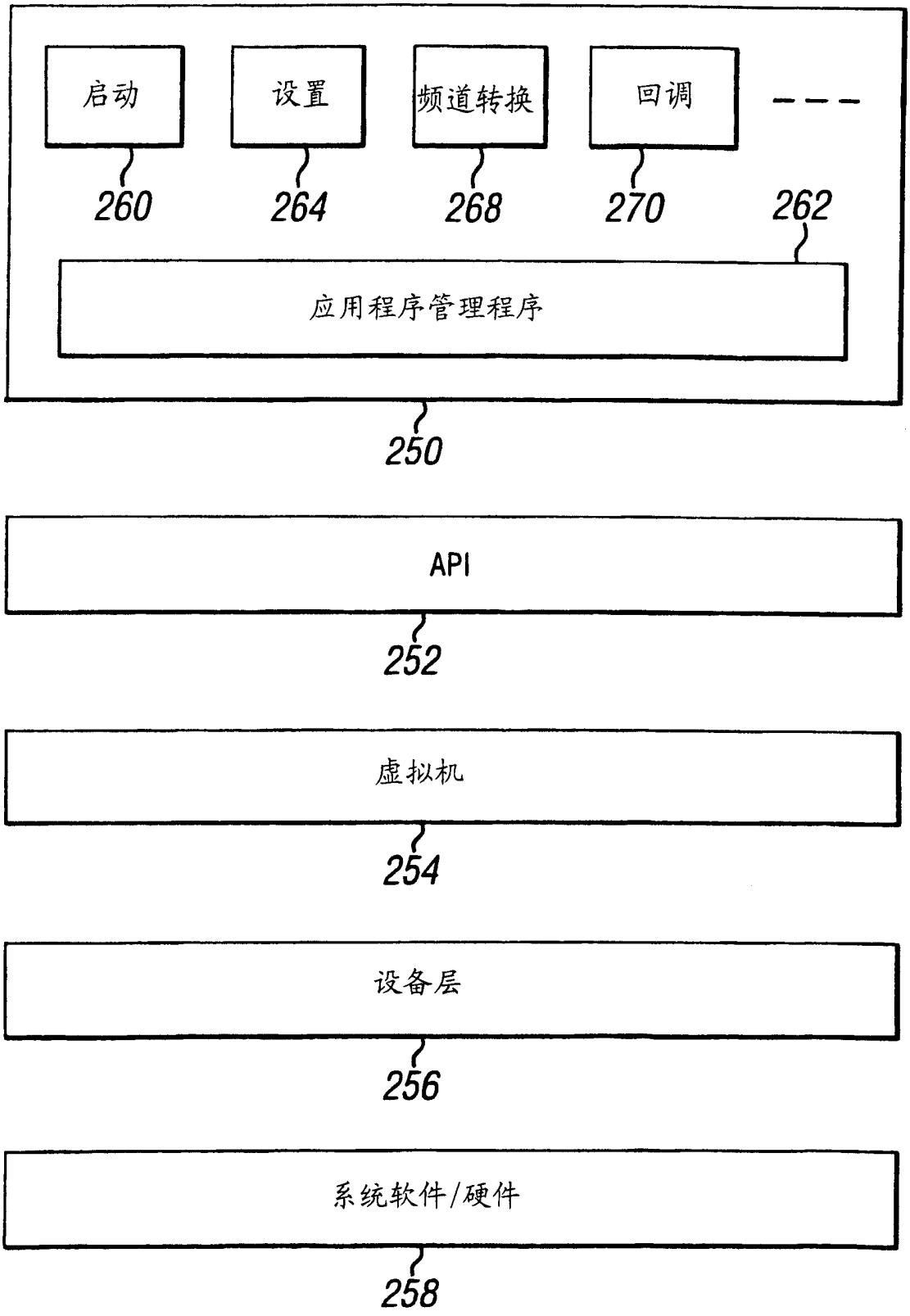

启动
设置
频道转换
回调
260
264
268
270
262
应用程序管理程序
250
API
252
虚拟机
254
设备层
256
系统软件/硬件
258

图 2c

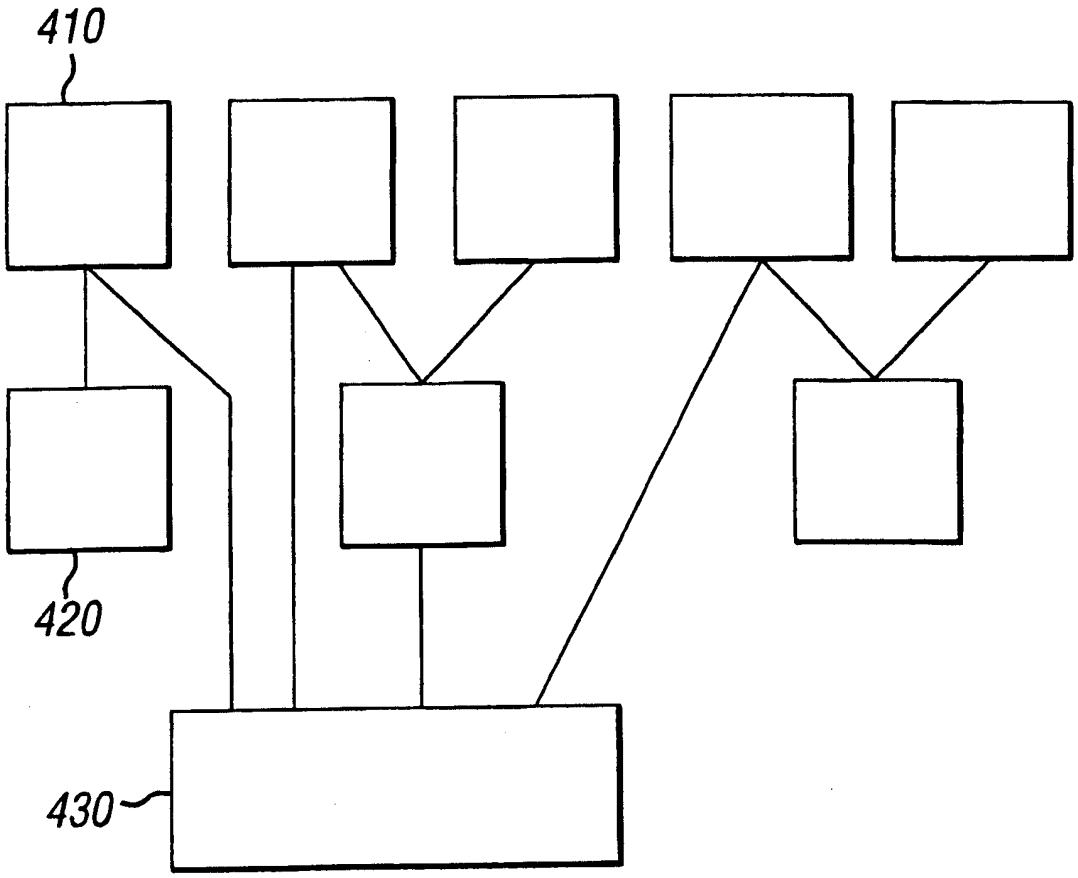


图 3

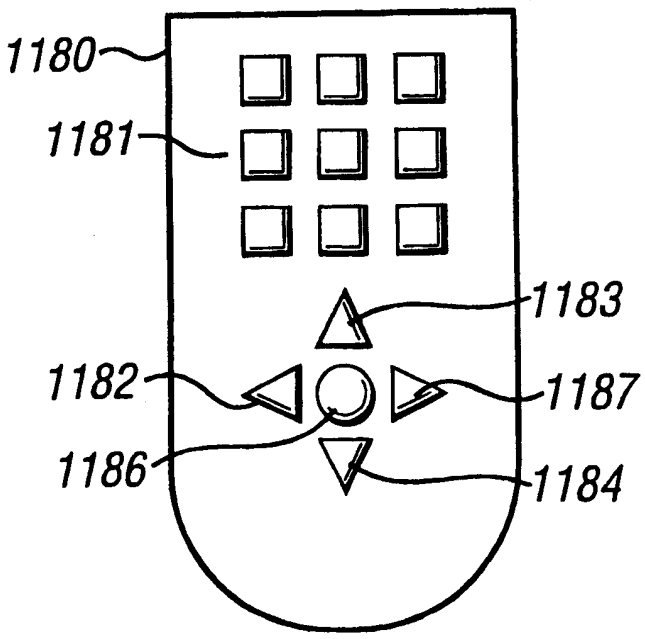


图 6b

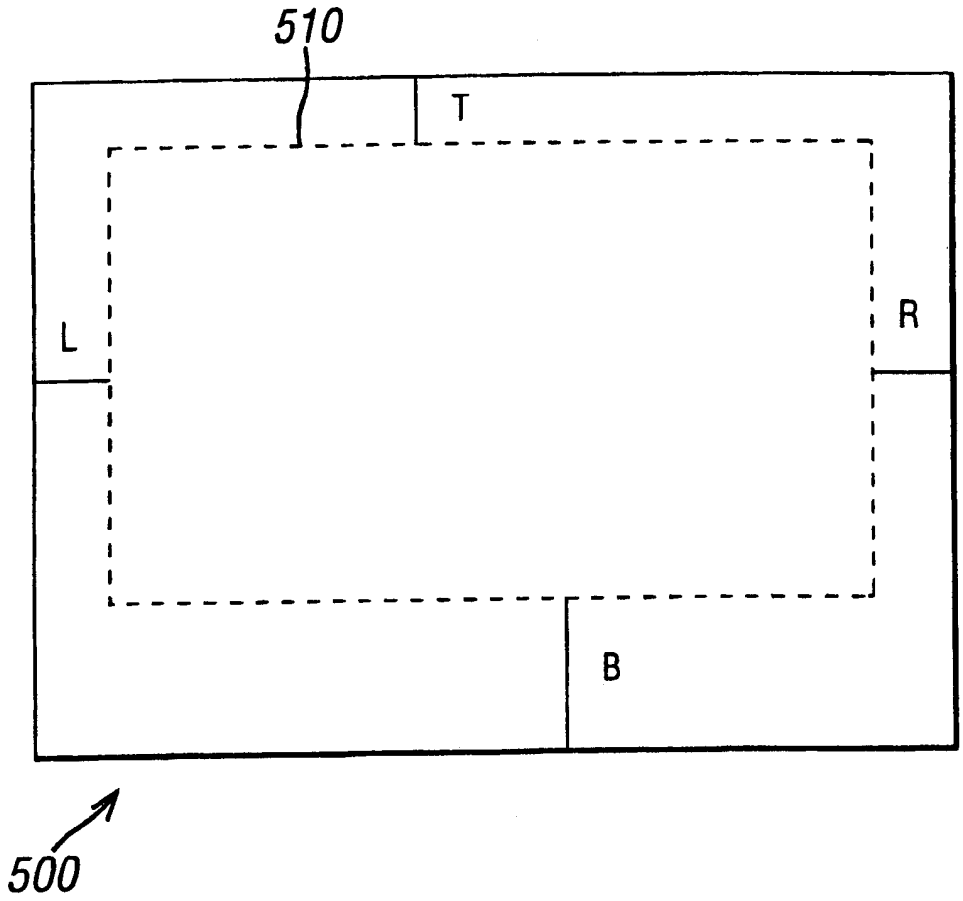

510
T
L
R
B
500

图 4

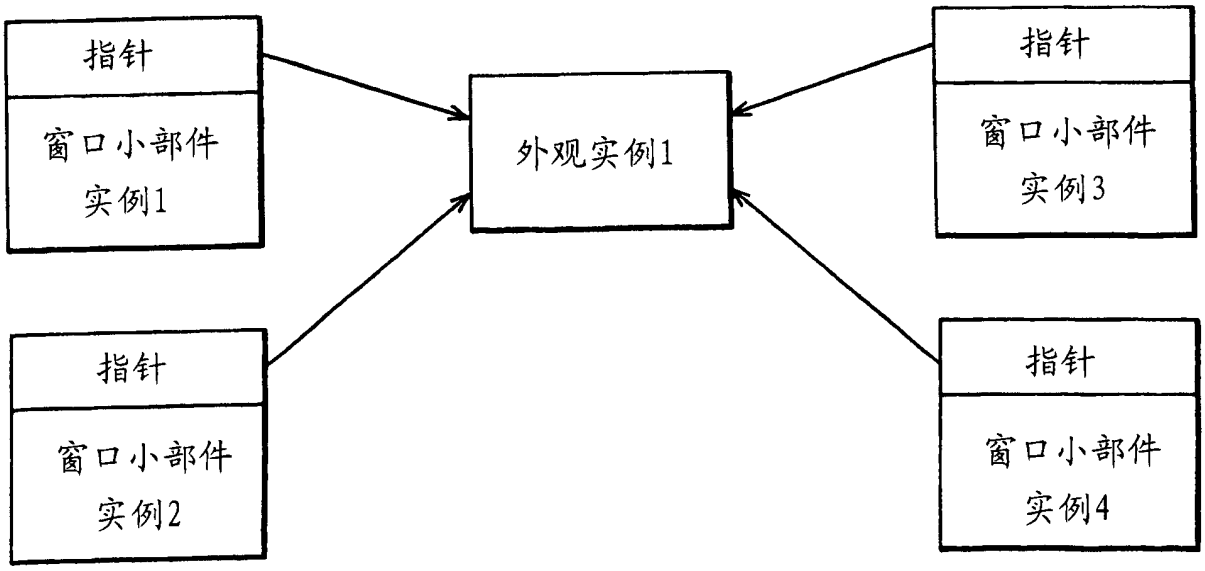

指针
窗口小部件
实例1
指针
窗口小部件
实例2
外观实例1
指针
窗口小部件
实例3
指针
窗口小部件
实例4

图 5

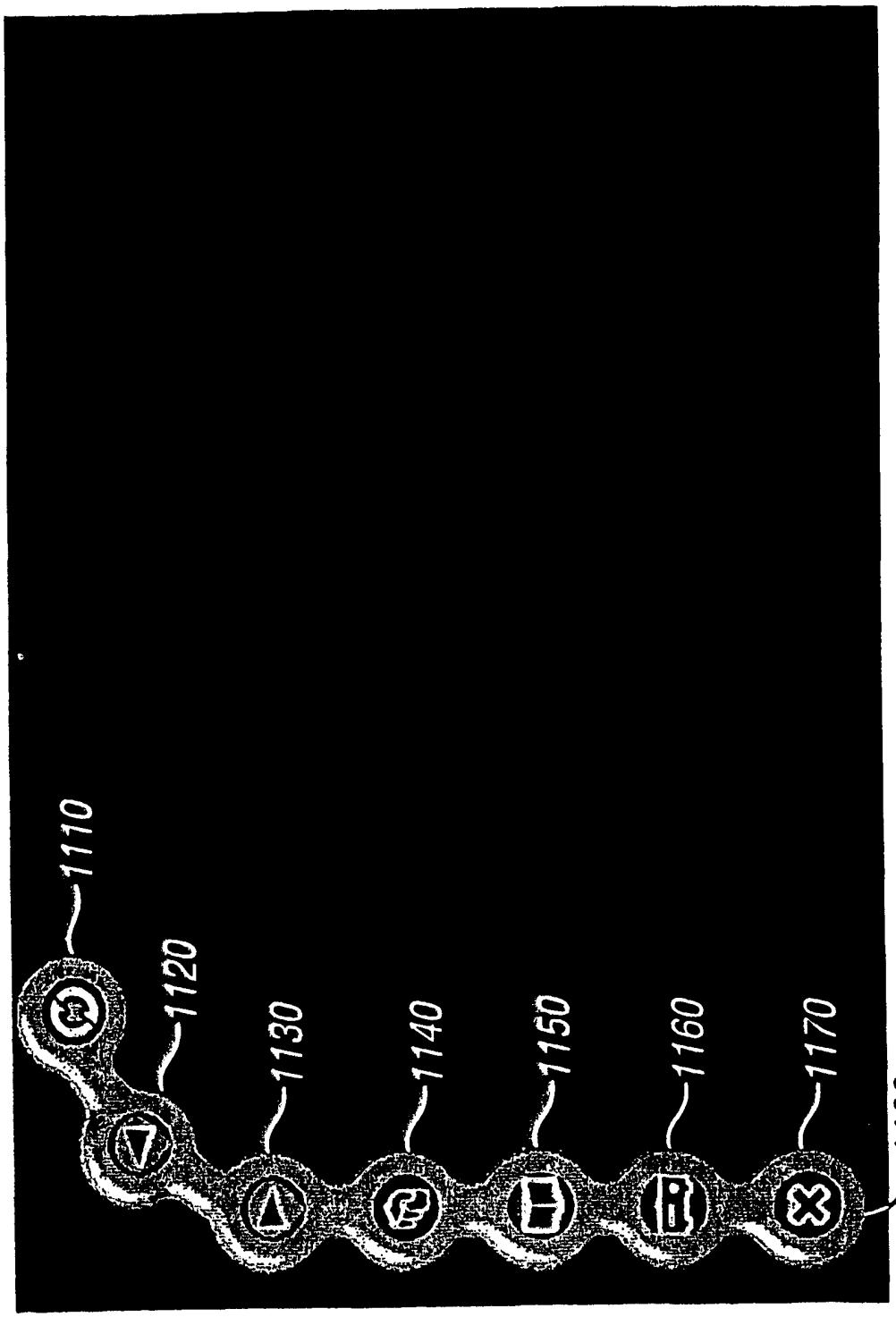


图 6a

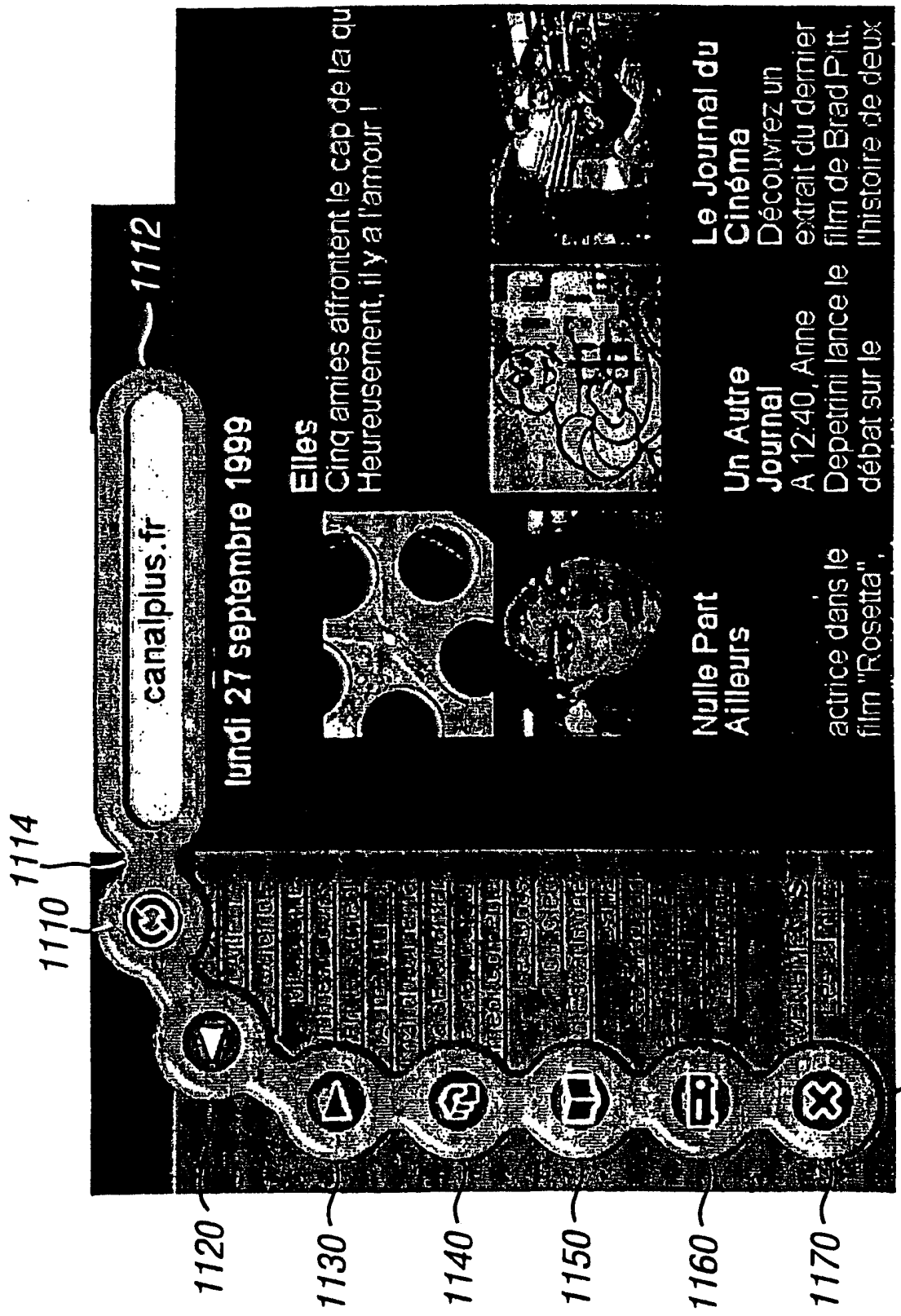

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire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

图 7

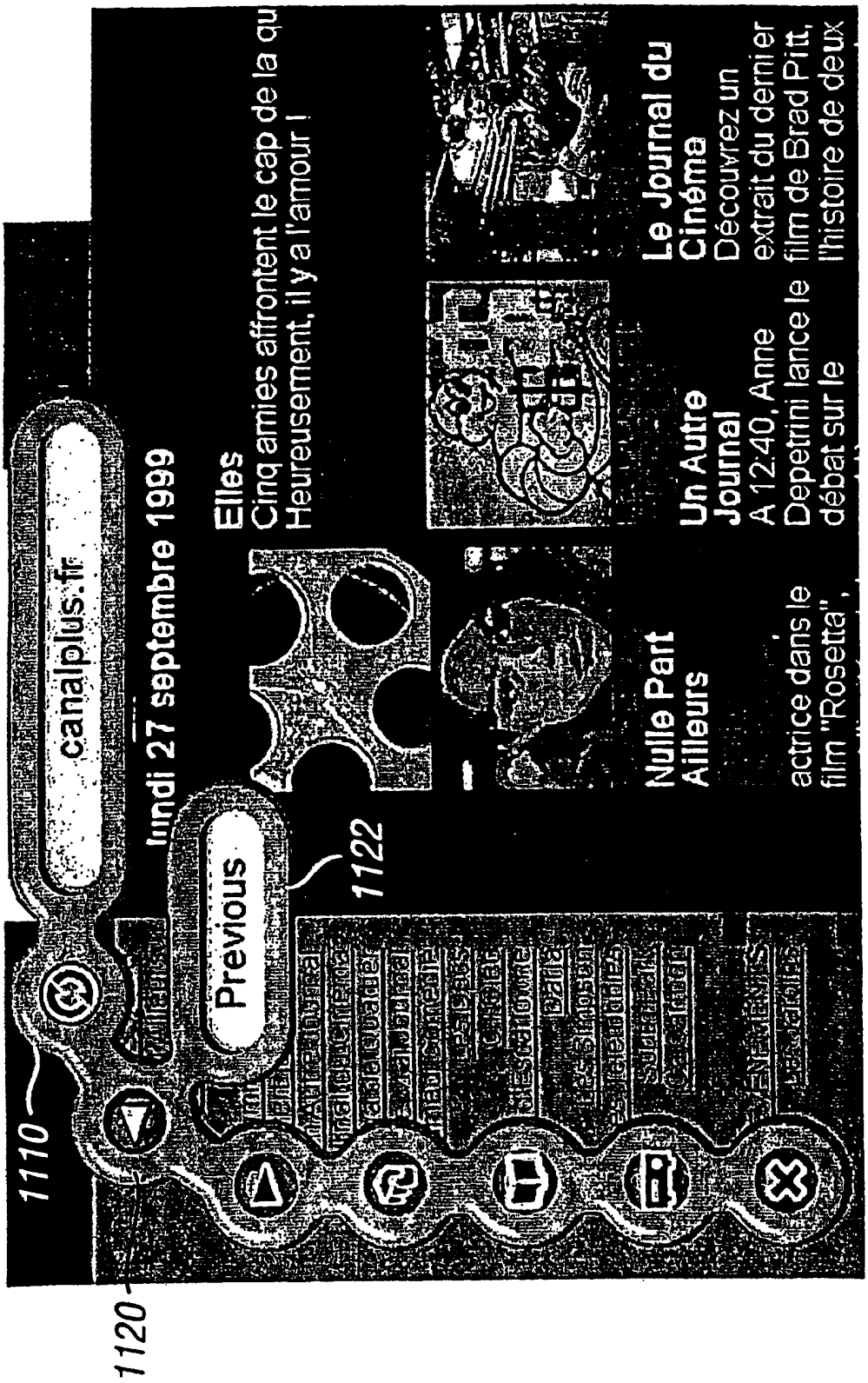

1120
1110
canalplus.fr
lundi 27 septembre 1999
Previous
1122
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part
Ailleurs
actrice dans le
film "Rosetta",
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du
Cinéma
Découvrez un
extrait du dernier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

图 8

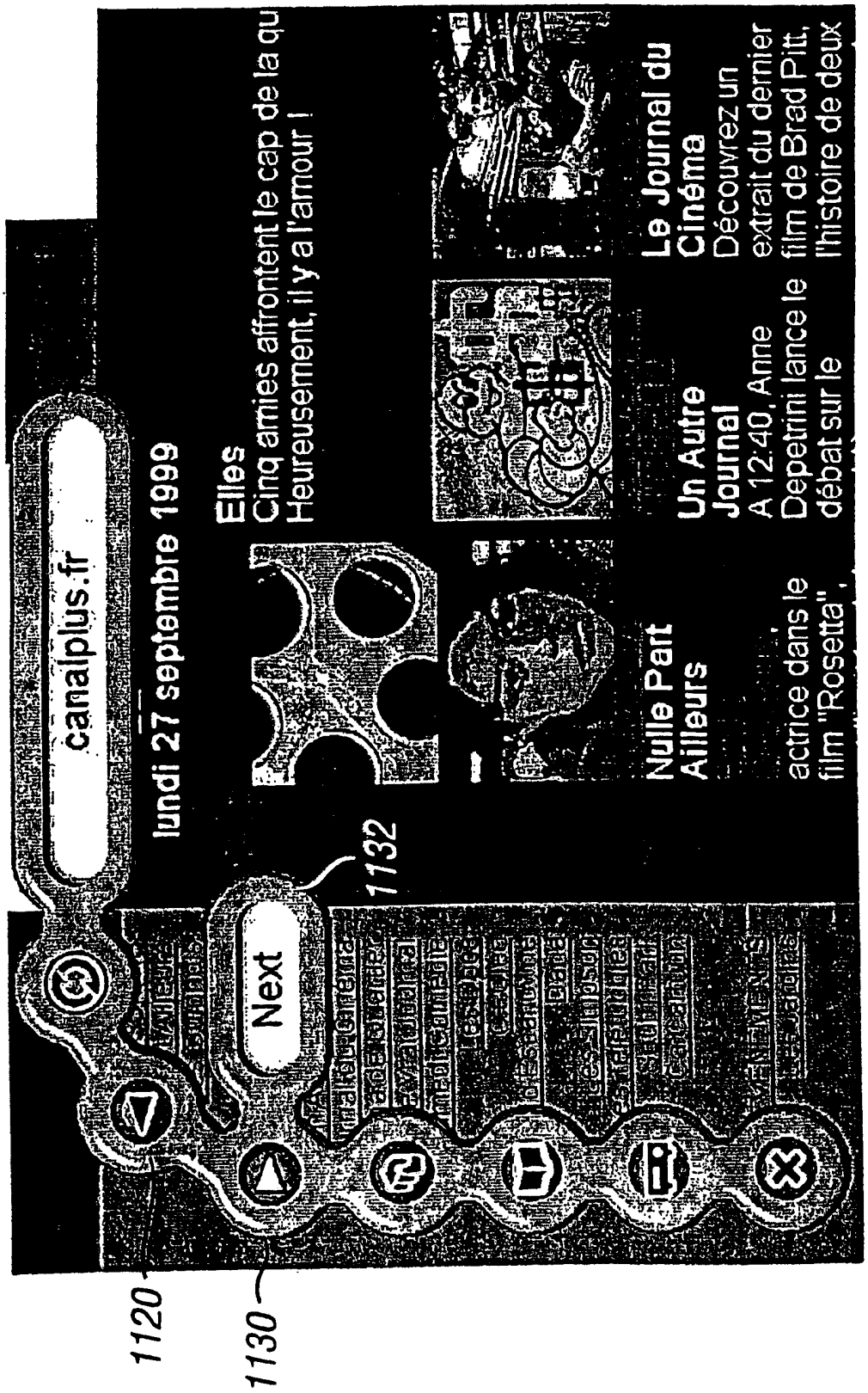

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film "Rosetta",
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire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

图 9

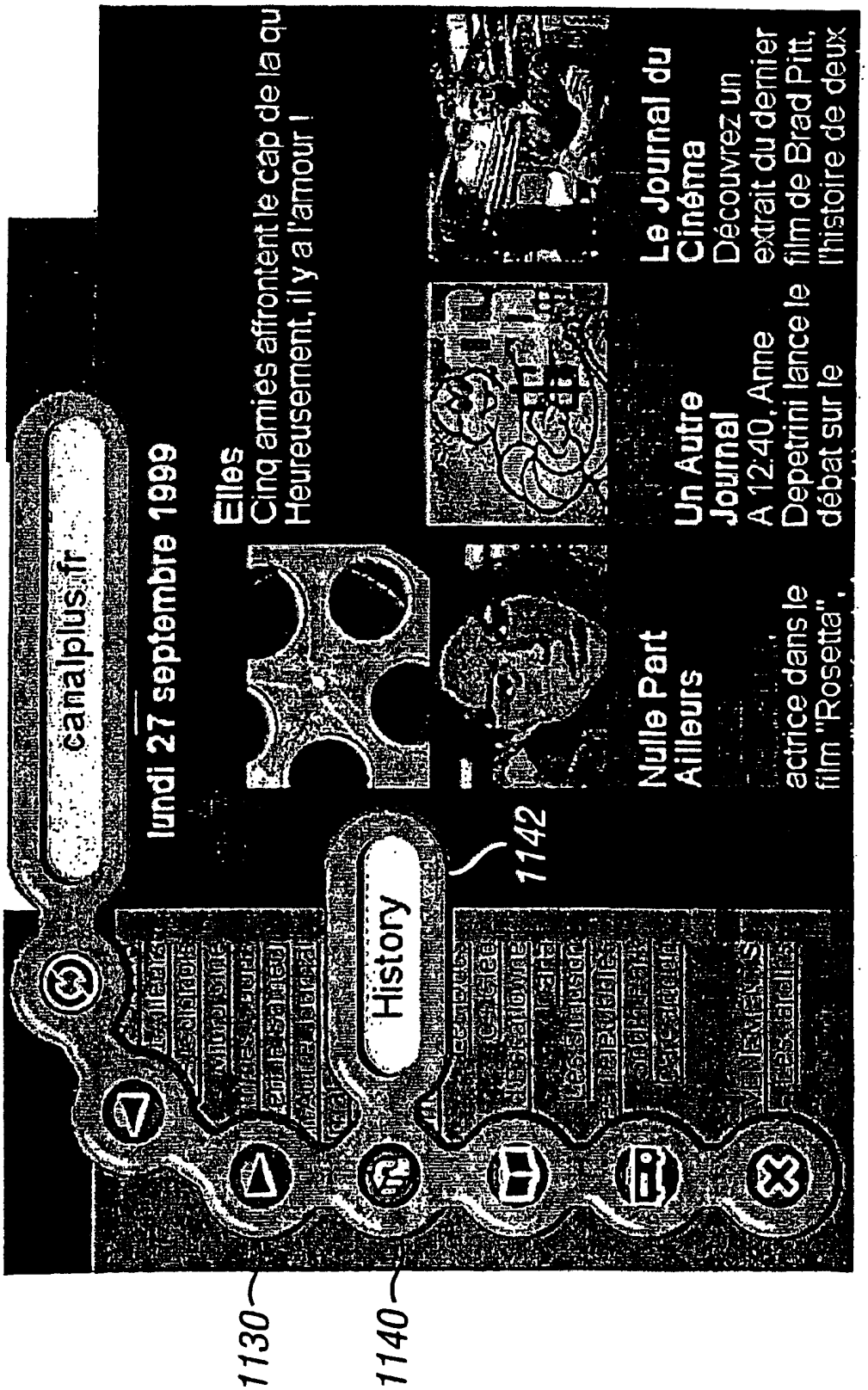

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le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

图 10

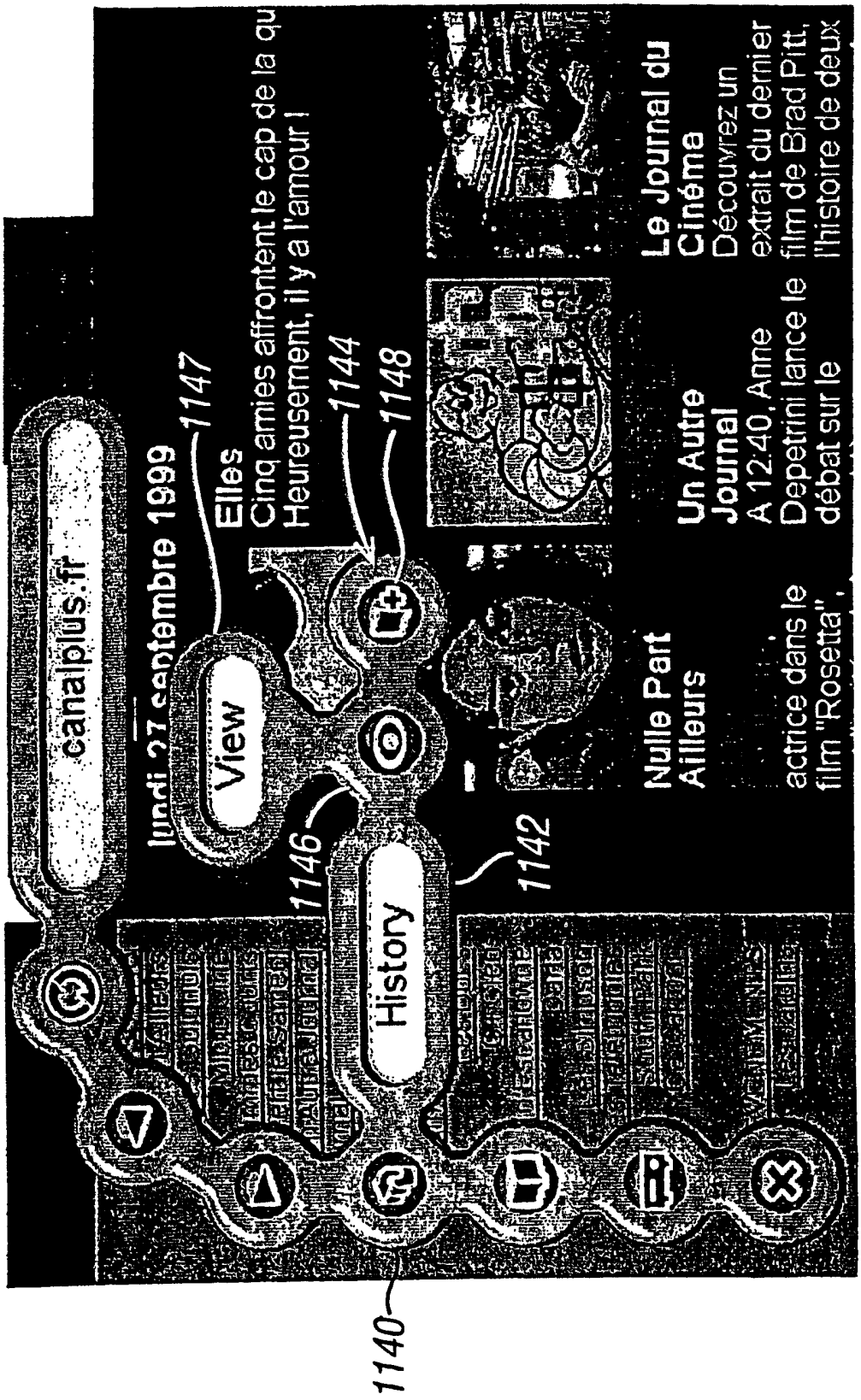

canalplus.fr
lundi 27 septembre 1999
1147
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
1144
1148
View
History
1146
1142
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait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

图 11

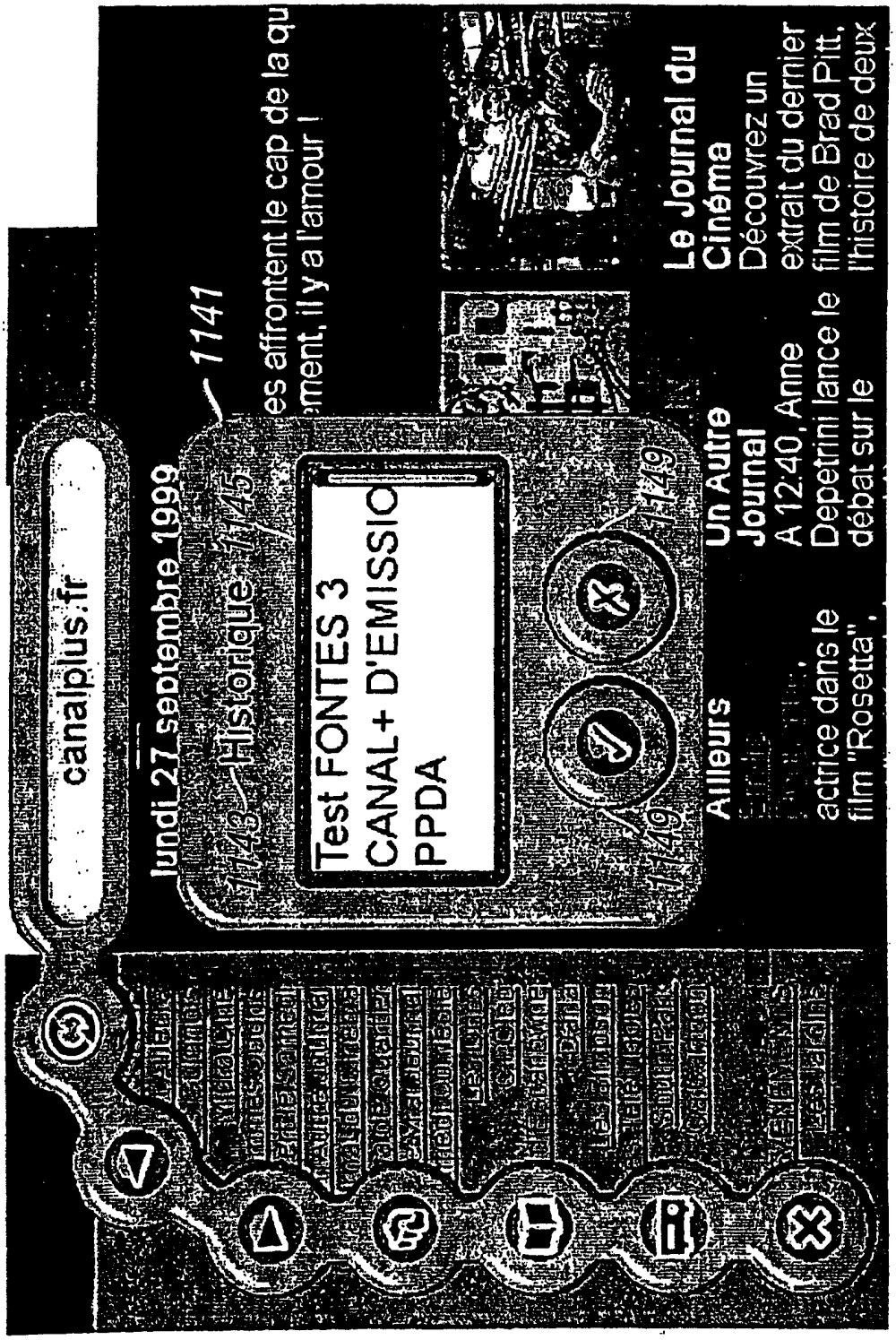

canalplus.fr
lundi 27 septembre 1999
1141
es affrontent le cap de la qu
ment, il y a l'amour !
1145 Historique 1145
Test FONTES 3
CANAL+ D'EMISSIO
PPDA
1149
1149
Ailleurs
actrice dans le
film "Rosetta",
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux

图 12

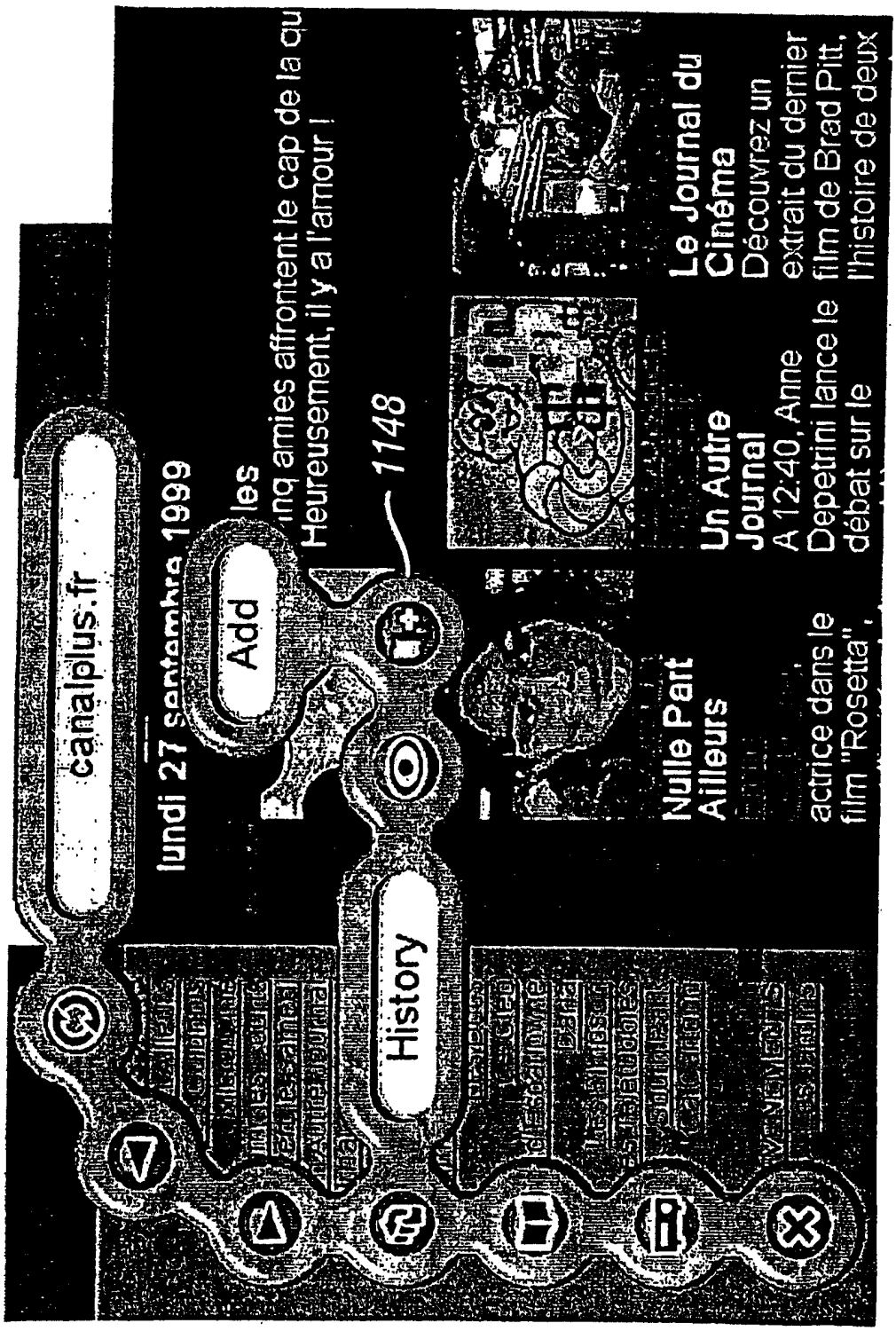

canalplus.fr
lundi 27 septembre 1999
Add
les
mq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
1148
History
Nulle Part Ailleurs
actrice dans le film "Rosetta",
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de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

图 13

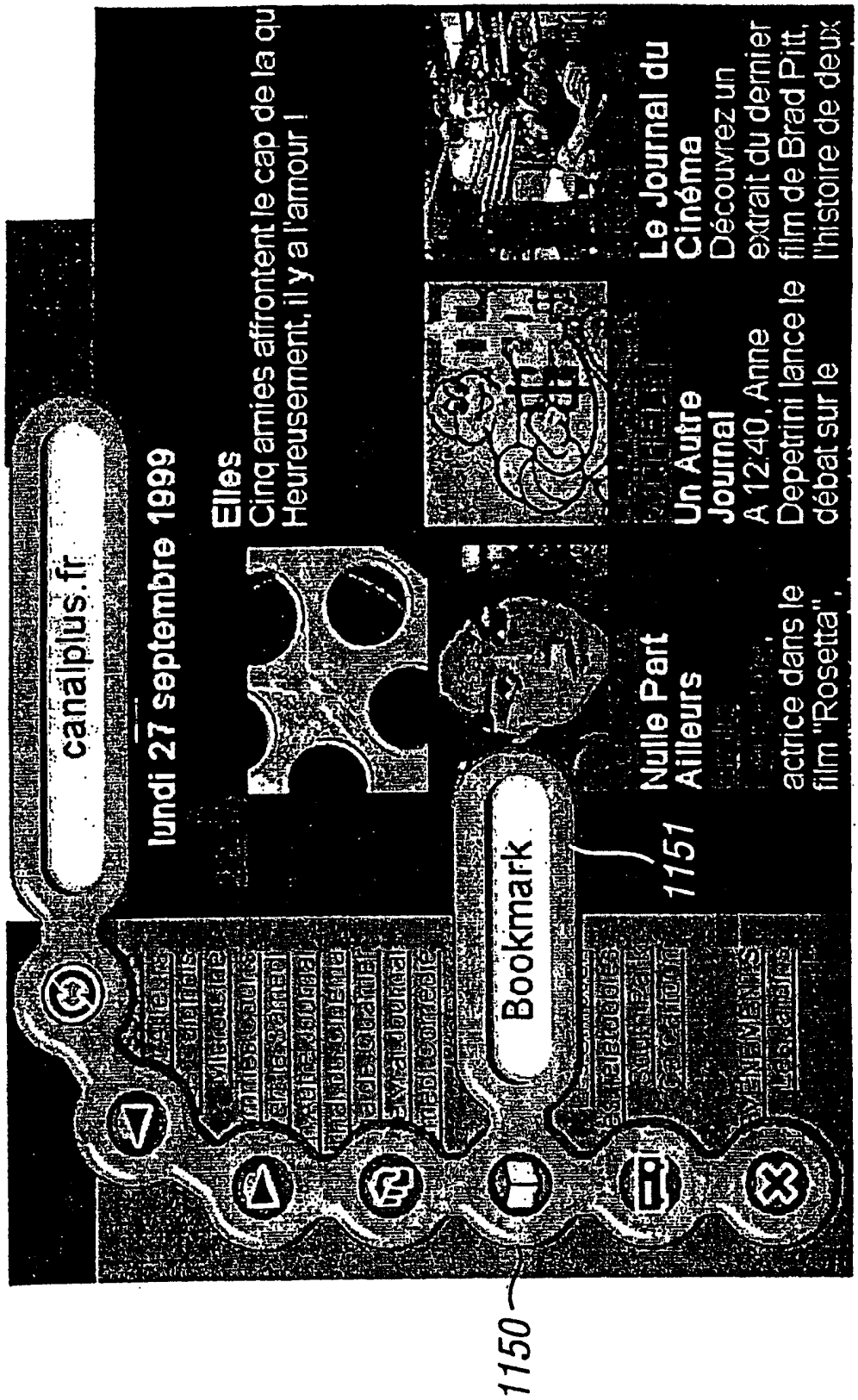

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le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

图 14

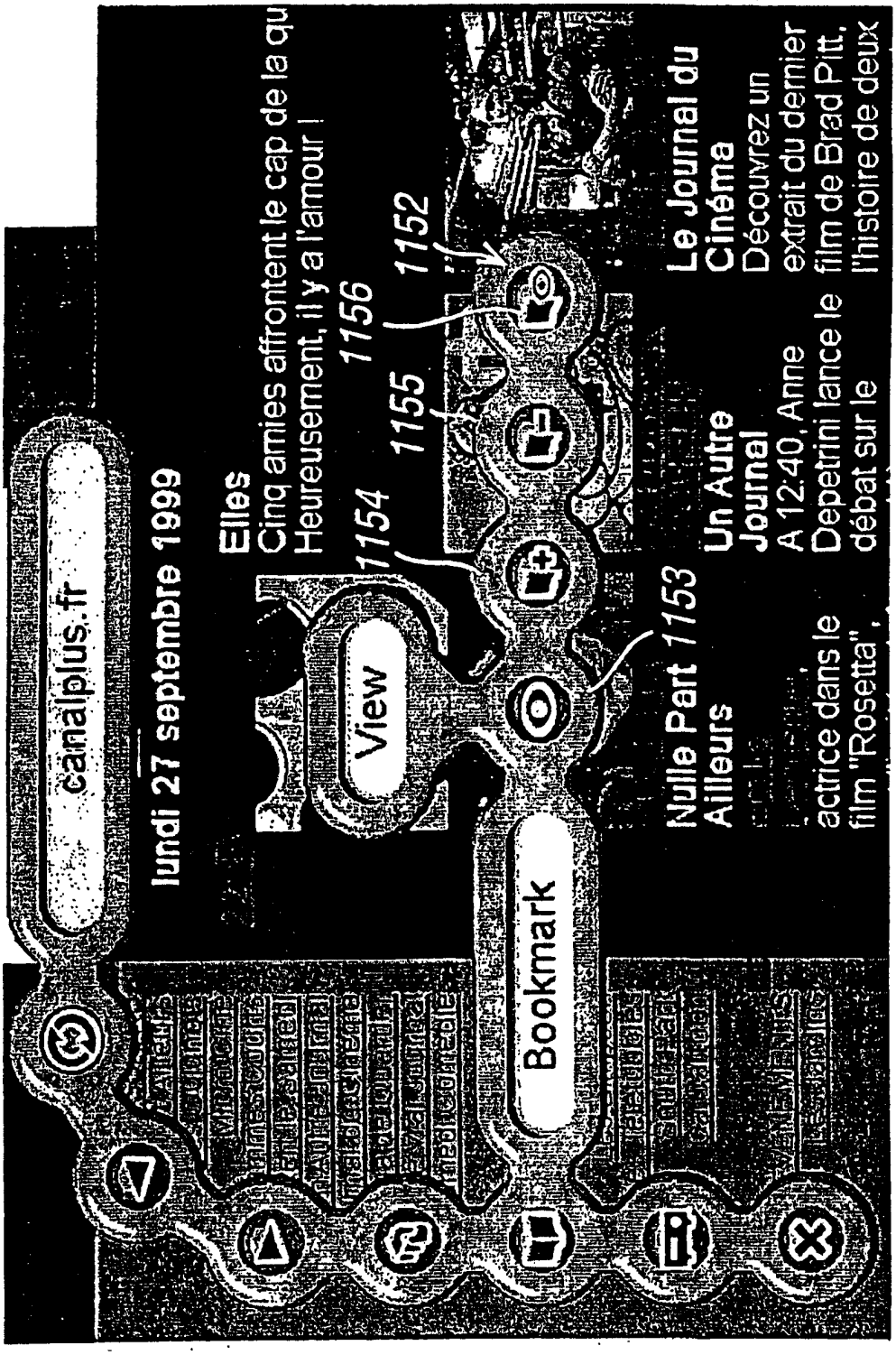

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
1156
1154
1155
1152
View
Bookmark
Nulle Part 1153
Ailleurs
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,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

图 15

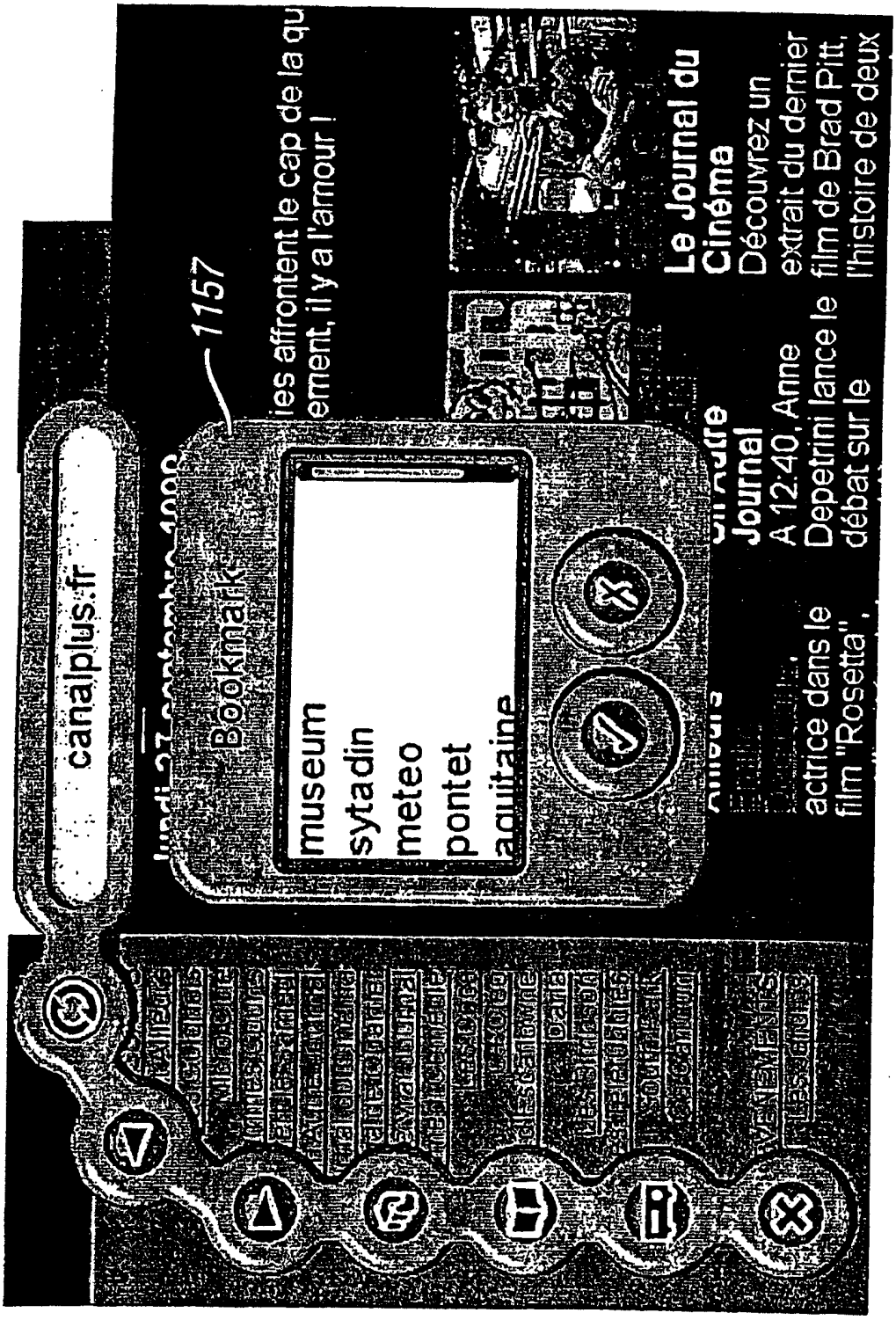

canalplus.fr
1157
Bookmark
museum
sytadin
meteo
pontet
aquitaine
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Journal
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film "Rosetta",

图 16

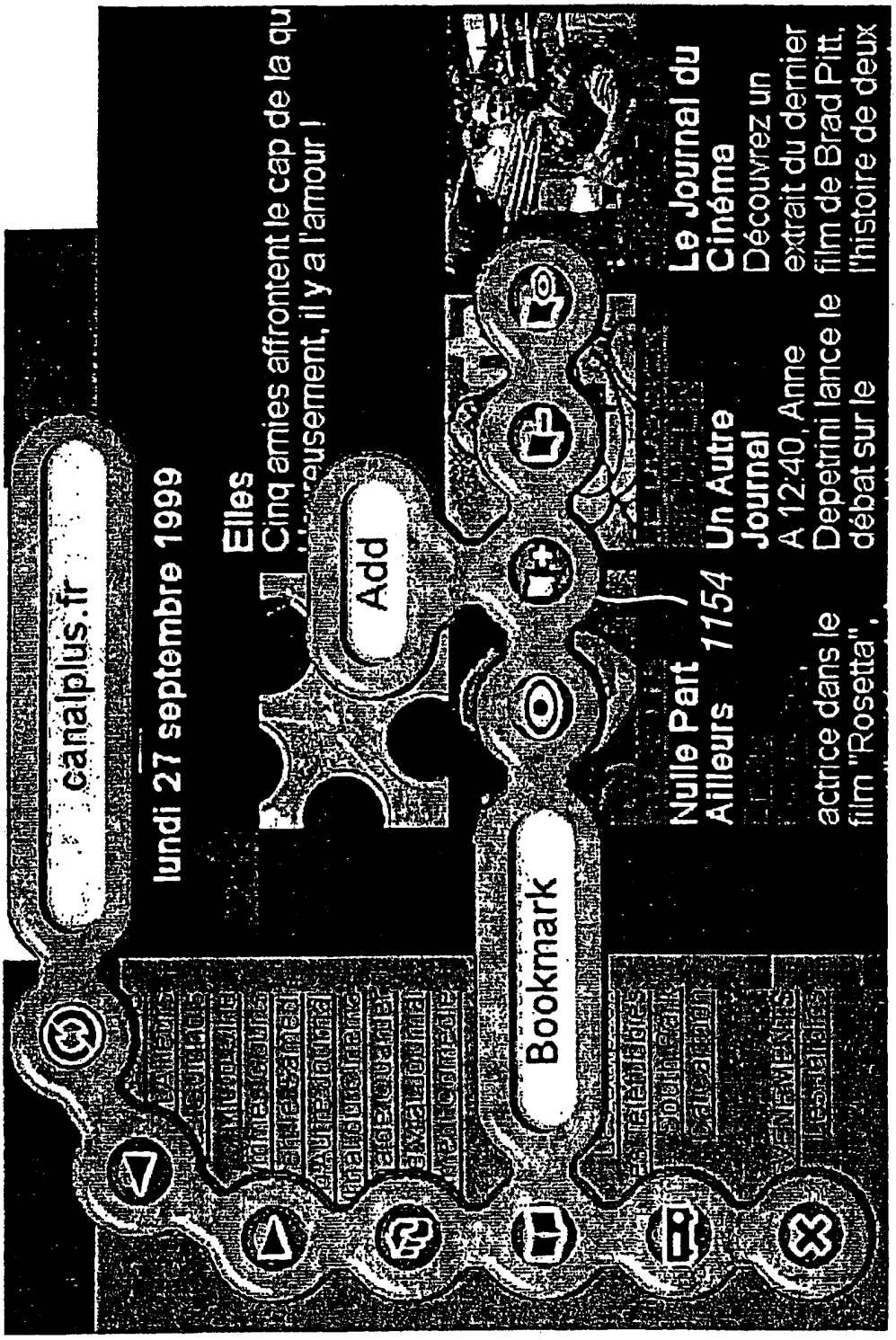

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Add
Bookmark
Nulle Part Ailleurs 1154 Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film "Rosetta",
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux

图 17

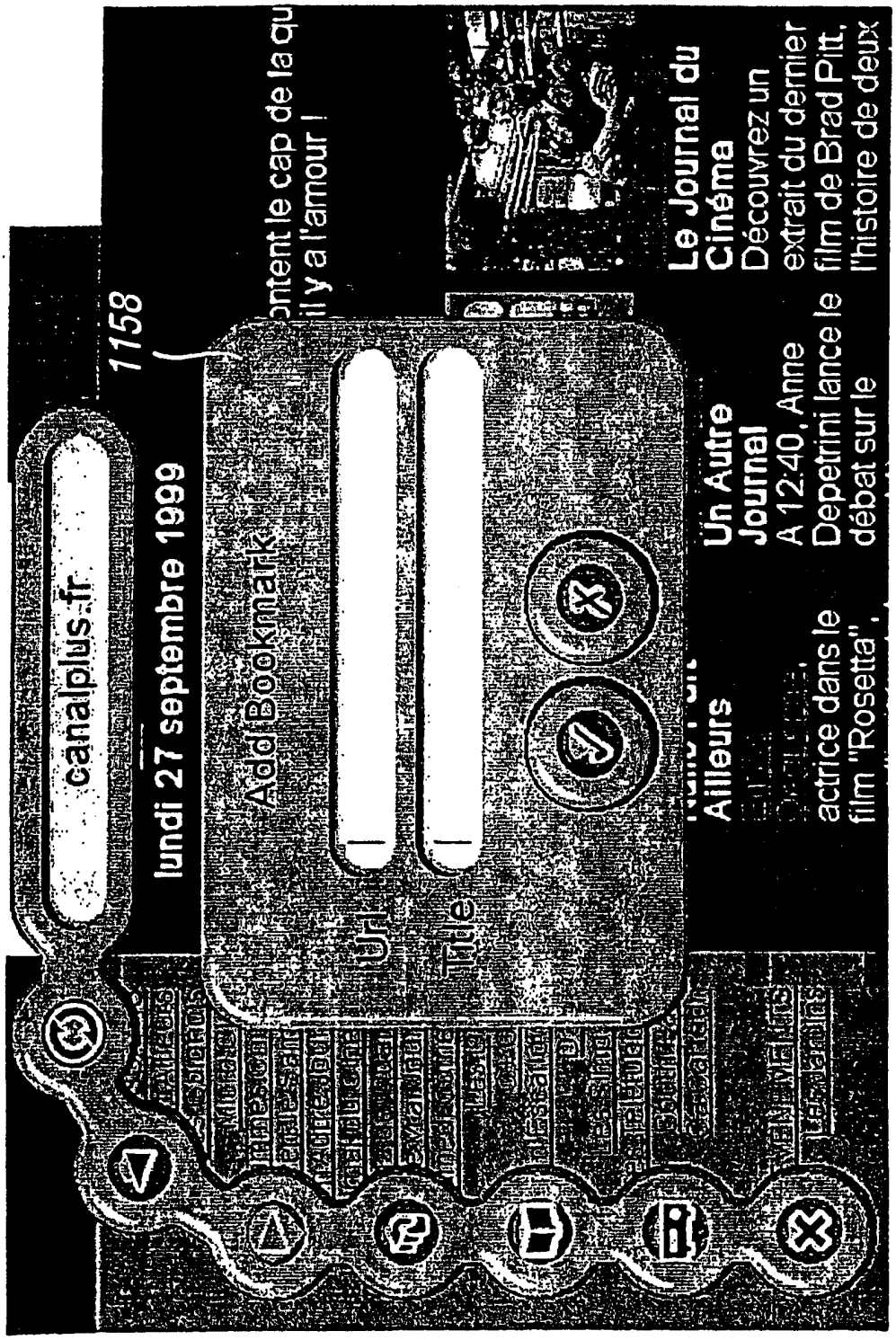

canalplus.fr
lundi 27 septembre 1999
1158
Add Bookmark
Url
Title
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Ailleurs
actrice dans le film "Rosetta",

图 18

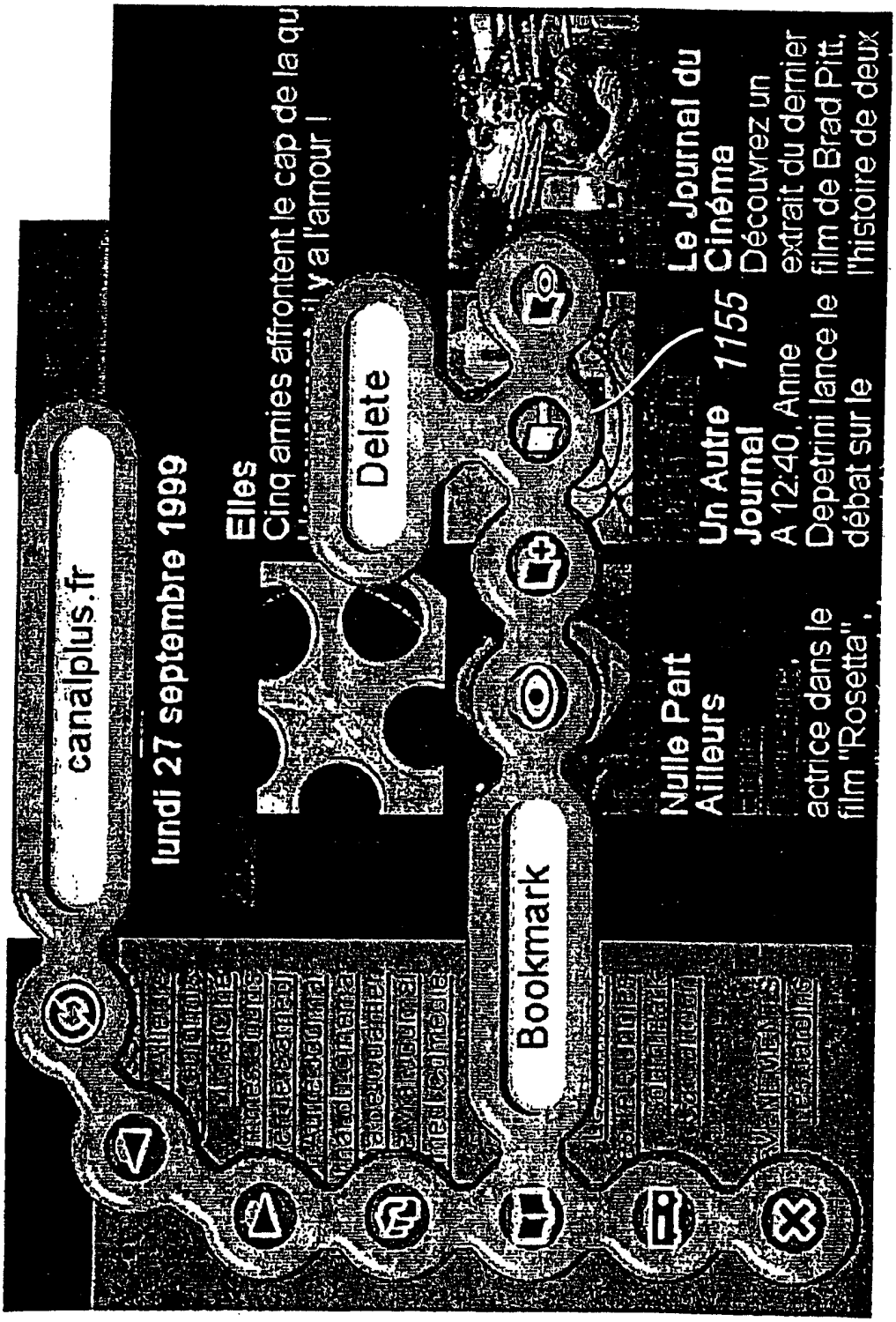

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
il y a l'amour !
Delete
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
1155
A 12:40, Anne Depetrini lance le débat sur le
Nulle Part Ailleurs
actrice dans le film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

图 19

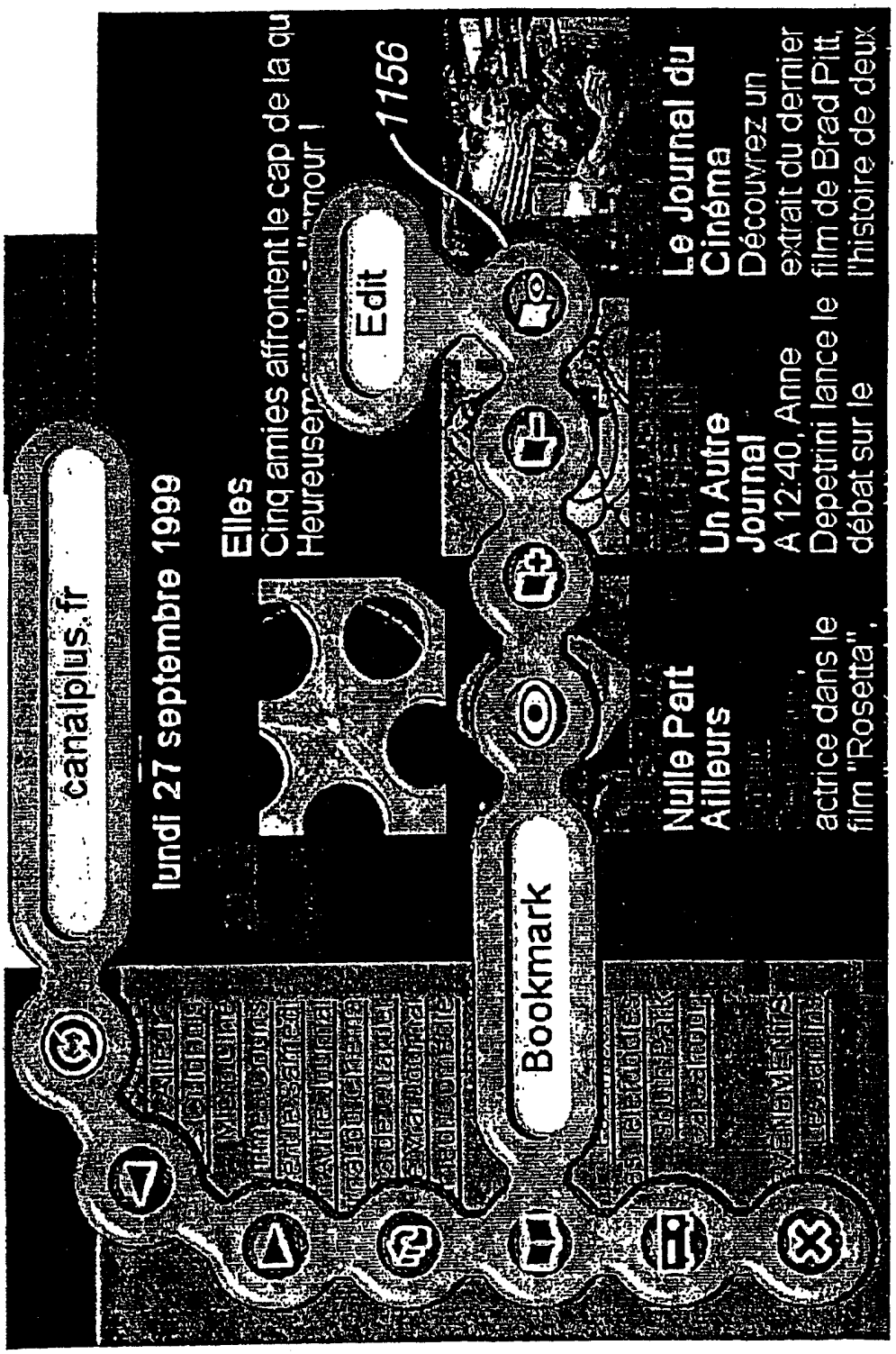

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusem
mour !
Edit
1156
Bookmark
Nulle Part
Ailleurs
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,
l'histoire de deux
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
actrice dans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

图 20

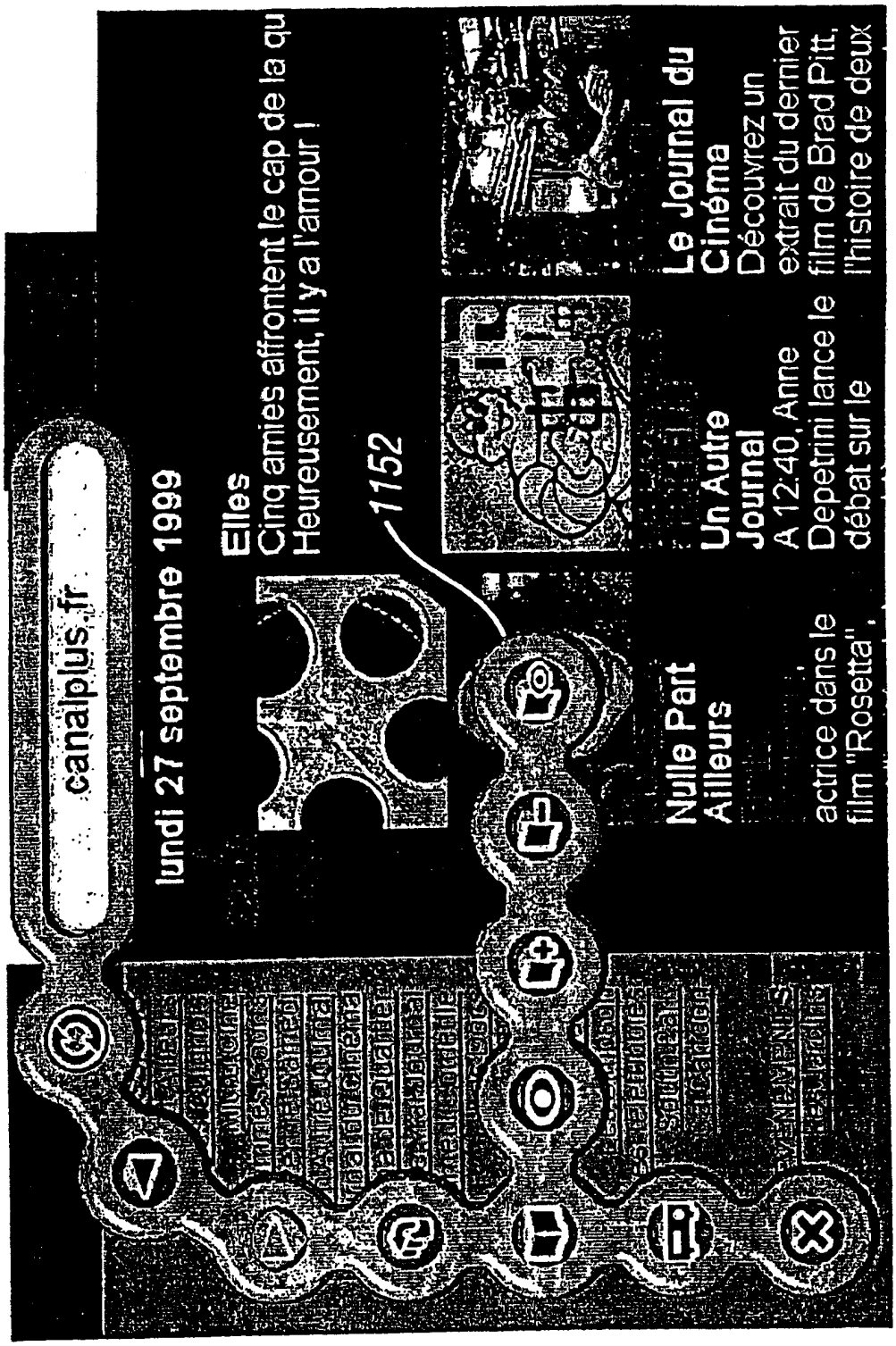

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
1152
Nulle Part
Ailleurs
actrice dans le
film "Rosetta",
Un Autre
Journal
A 12:40, Anne
Depetrini lance le
débat sur le
Le Journal du
Cinéma
Découvrez un
extrait du dernier
film de Brad Pitt,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

图 21

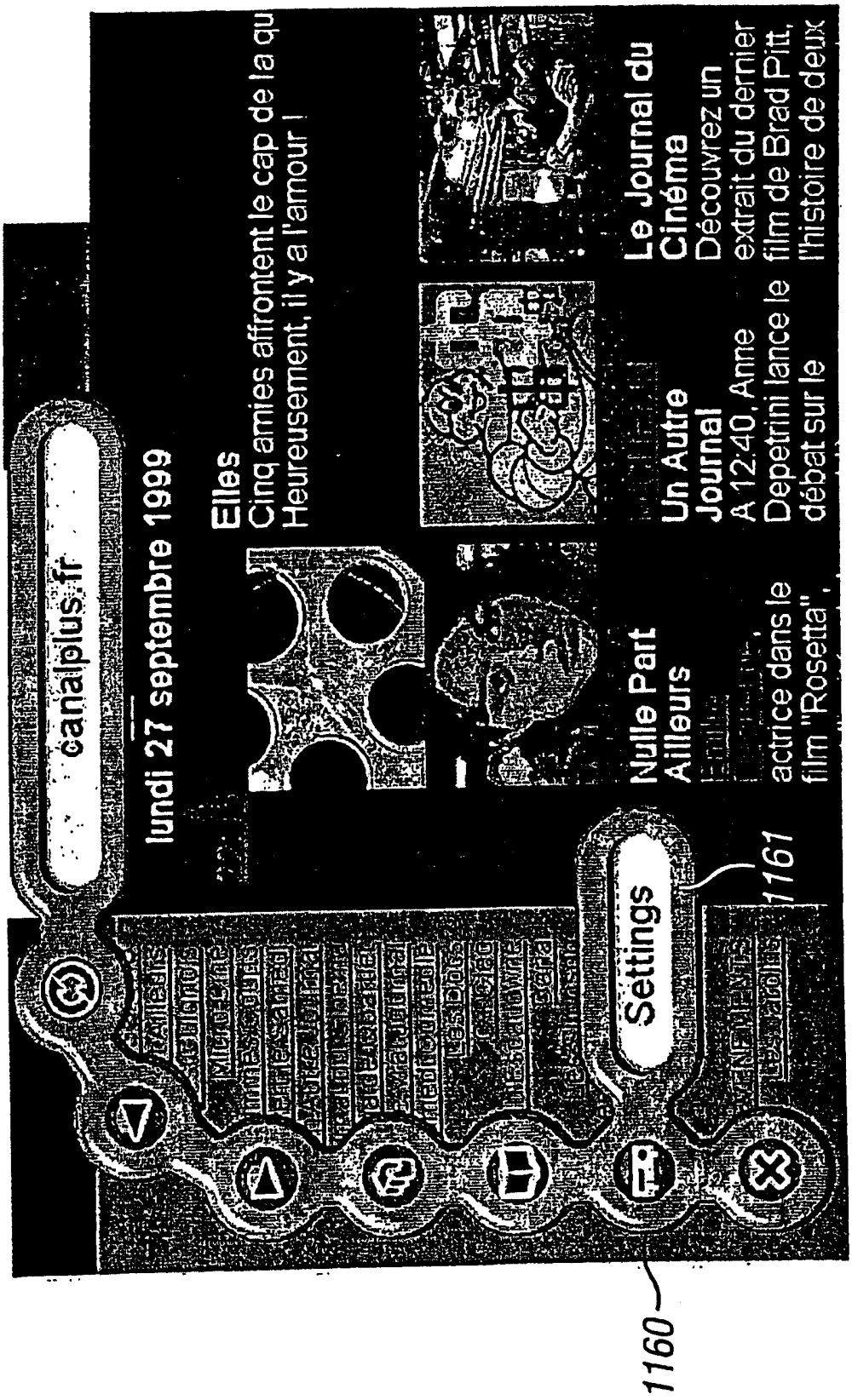

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le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

图 22

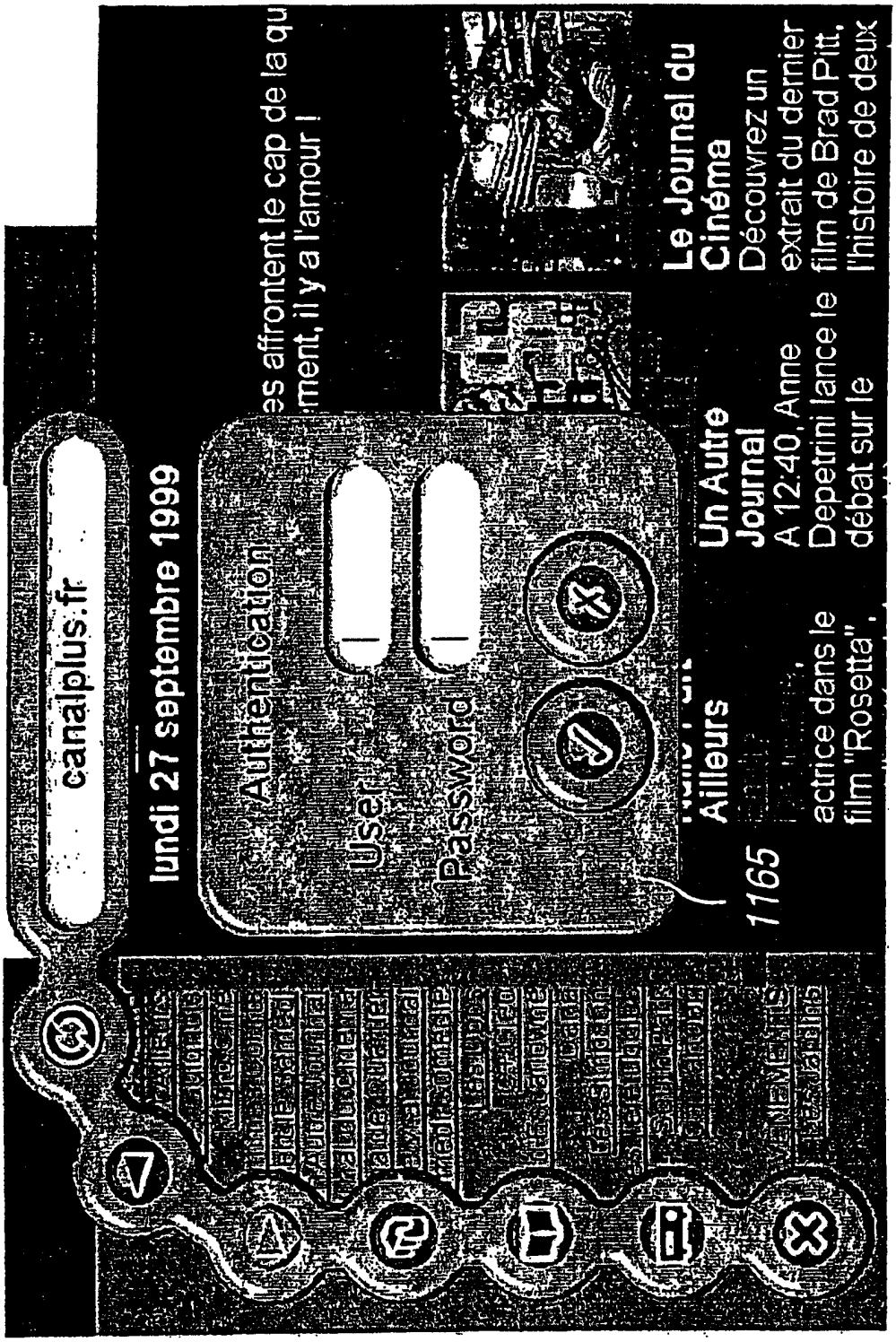

canalplus.fr
lundi 27 septembre 1999
Authentication
es affrontent le cap de la qu
ment, il y a l'amour !
User
Password
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Ailleurs
1165
actrice dans le film "Rosetta",

图 23

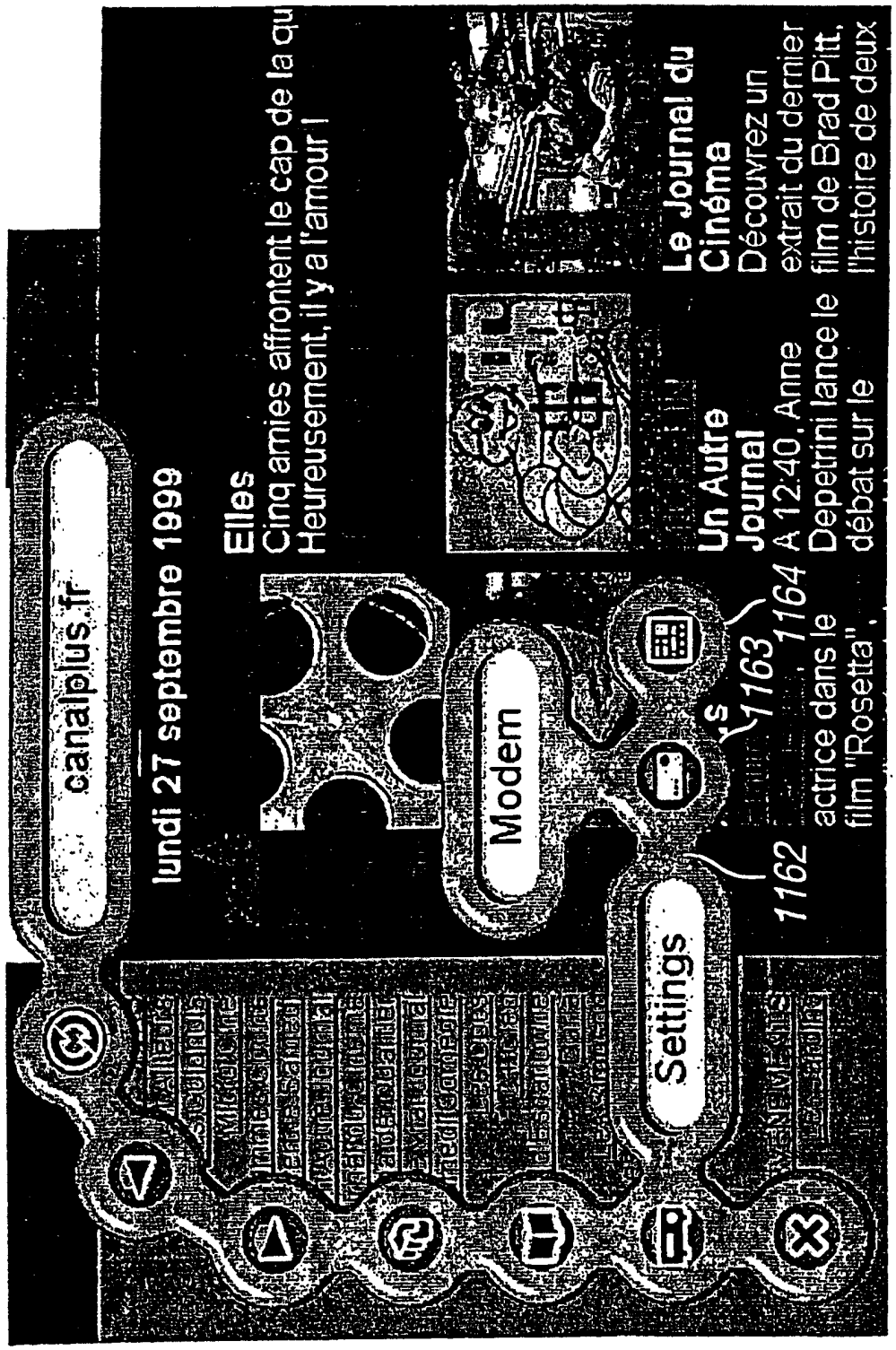

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Modem
Settings
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
1162
1163
1164

图 24

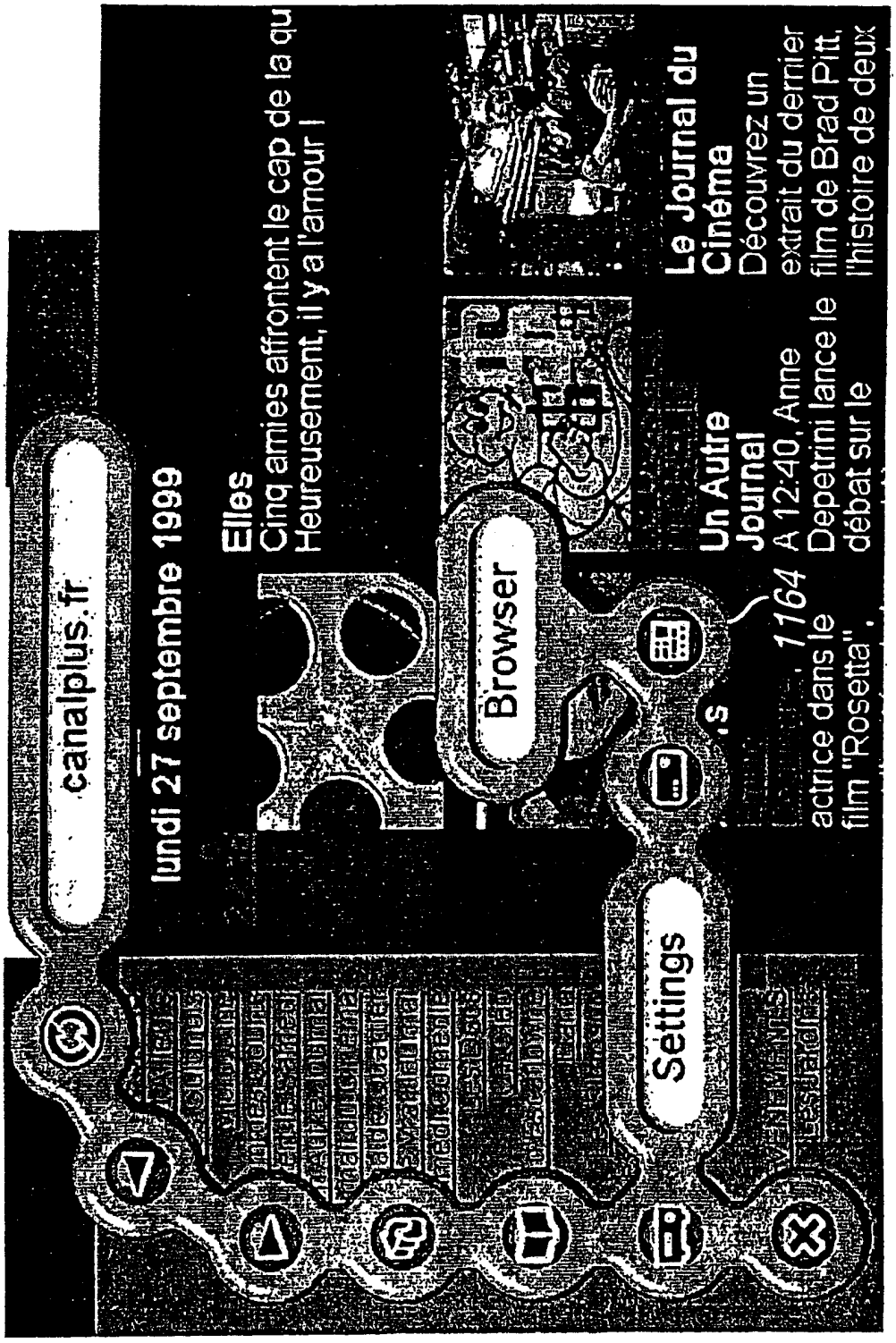

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Browser
Settings
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film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

图 25

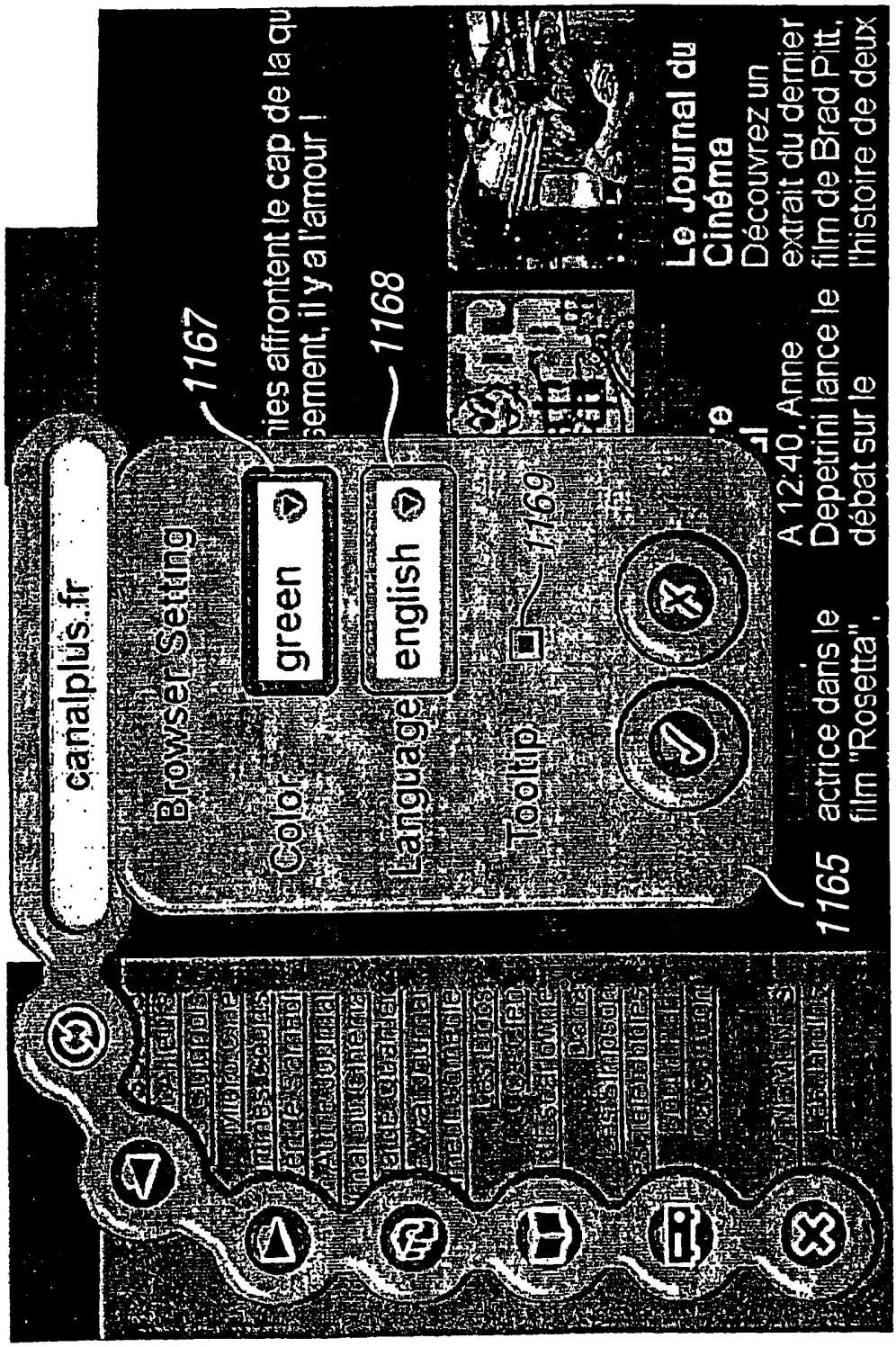

canalplus.fr
Browser Setting
Color
green
Language
english
Tooltip
1167
1168
1169
1165
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
A 12:40, Anne Depetrini lance le débat sur le
actrice dans le film "Rosetta",

图 26

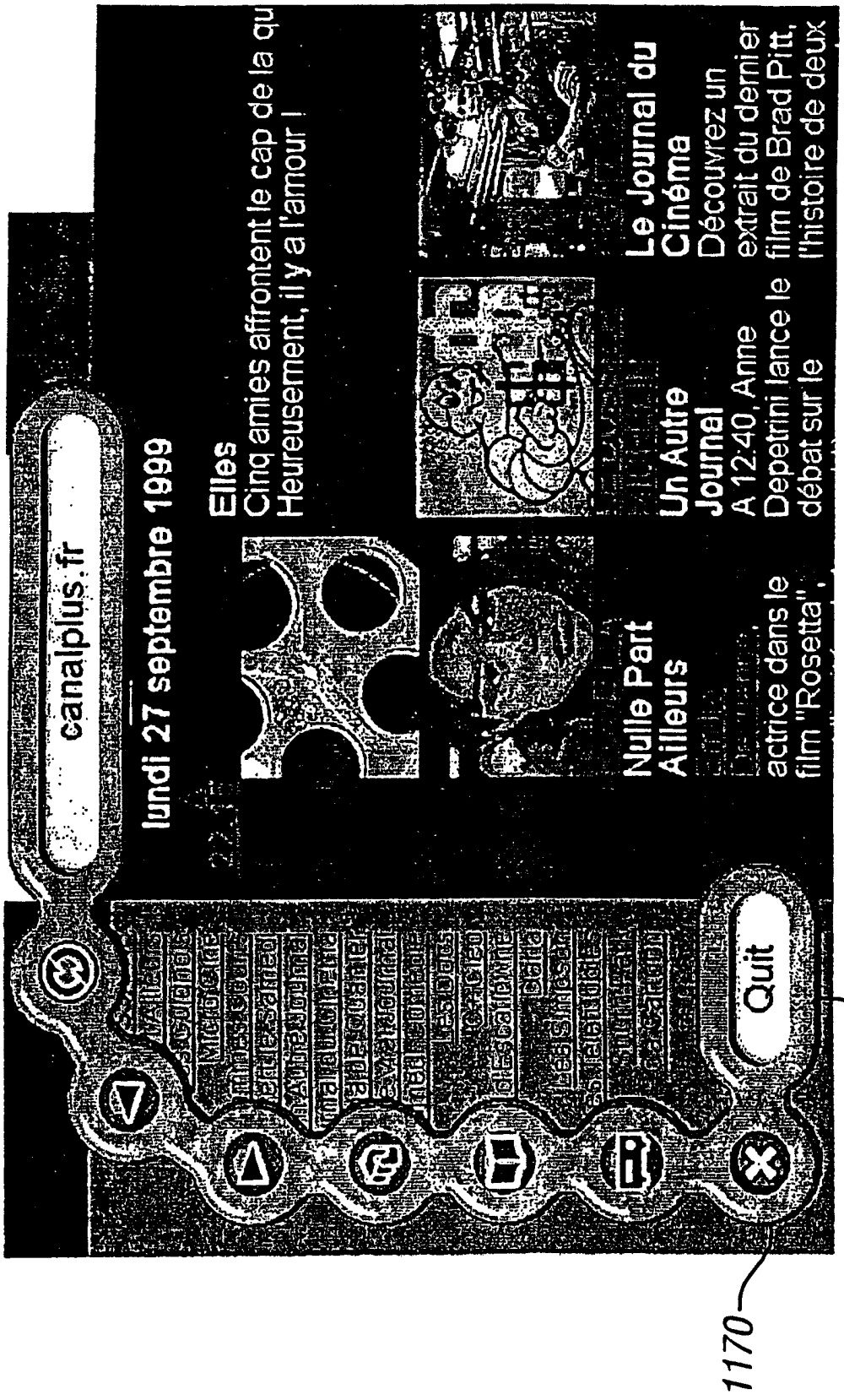

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Nulle Part Ailleurs
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Quit
1170
1171

图 27

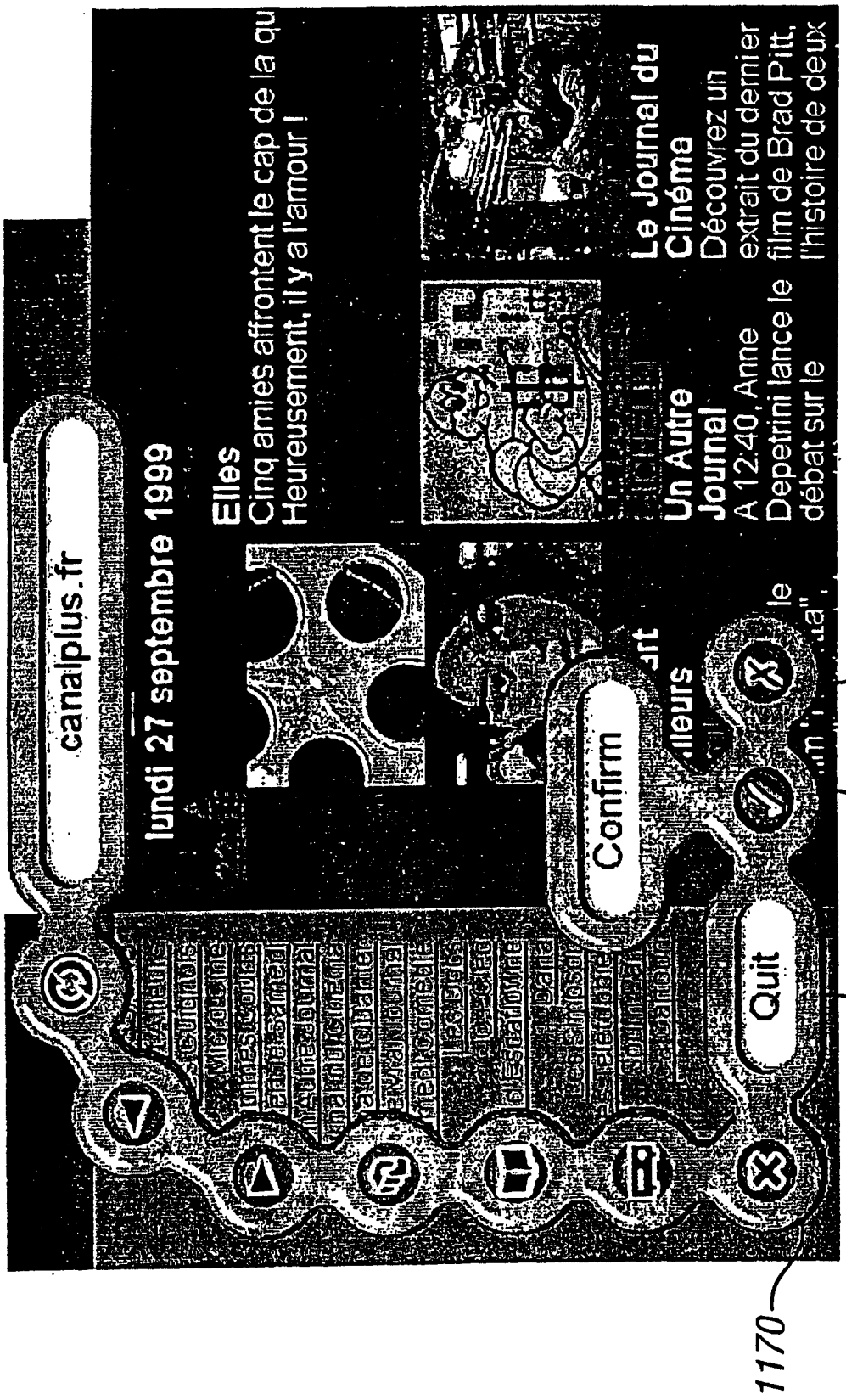

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Confirm
Quit
1170
1171
1173
1172

图 28

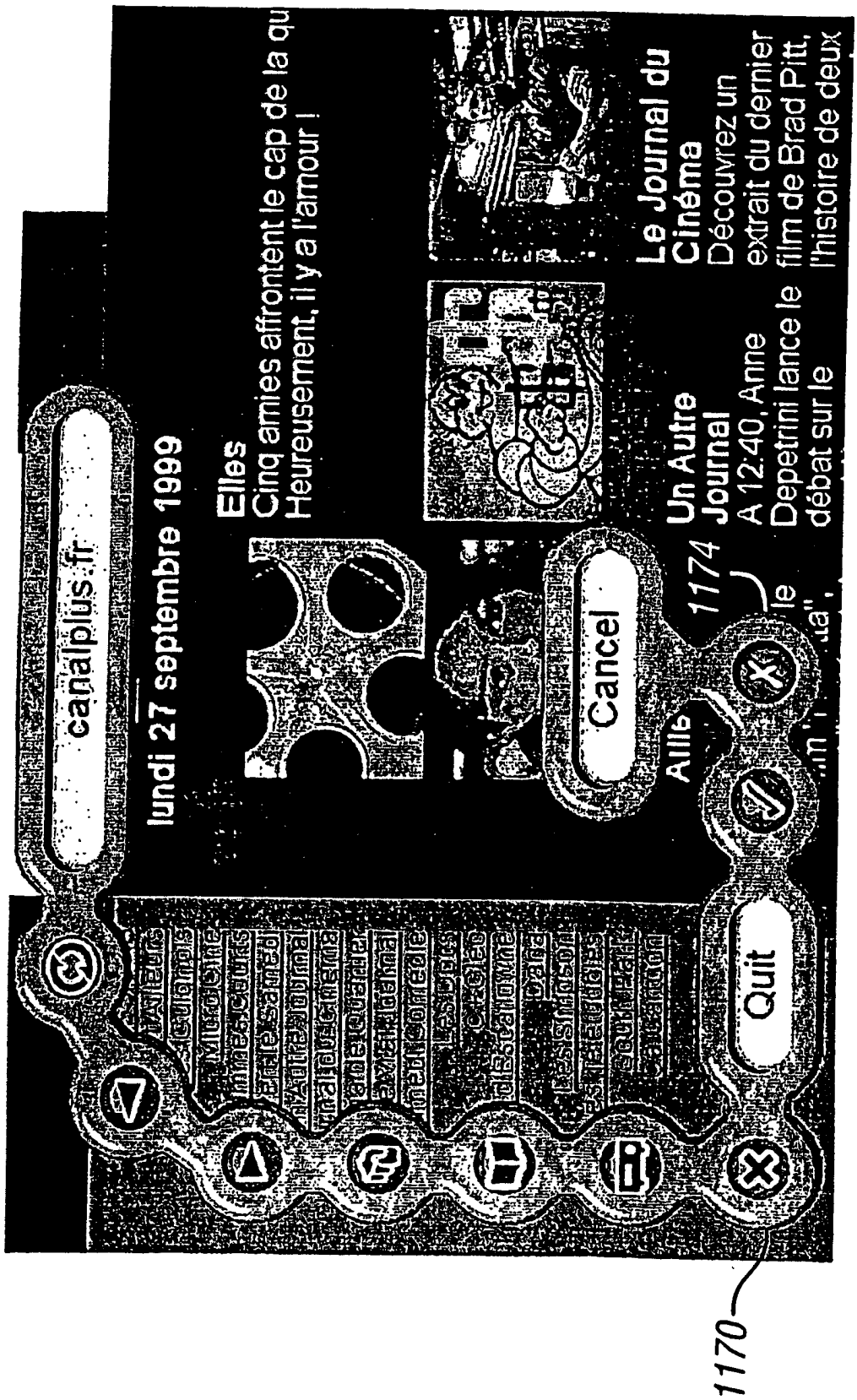

canalplus.fr
lundi 27 septembre 1999
Elles
Cinq amies affrontent le cap de la qu
Heureusement, il y a l'amour !
Le Journal du Cinéma
Découvrez un extrait du dernier film de Brad Pitt, l'histoire de deux
Un Autre Journal
A 12:40, Anne Depetrini lance le débat sur le
Cancel
Quit
1174
1170

图 29

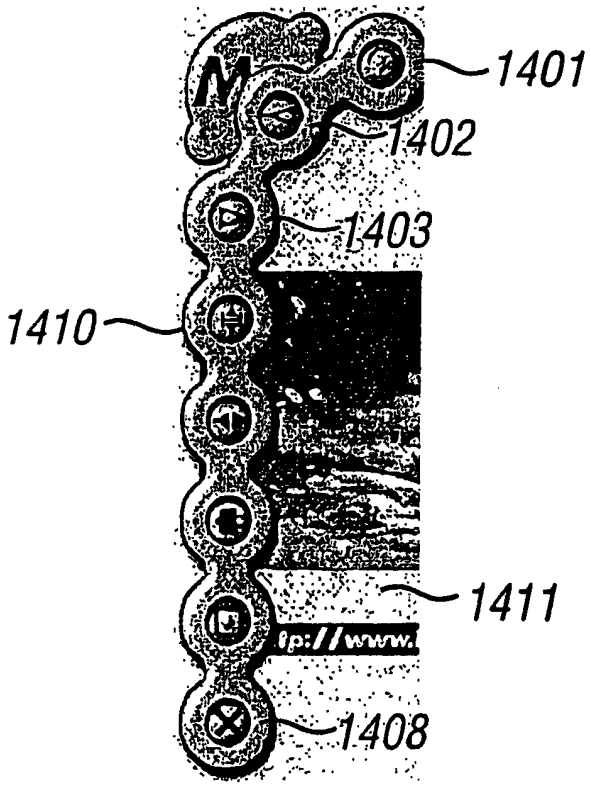

1401
1402
1403
1410
1411
1408

图 30

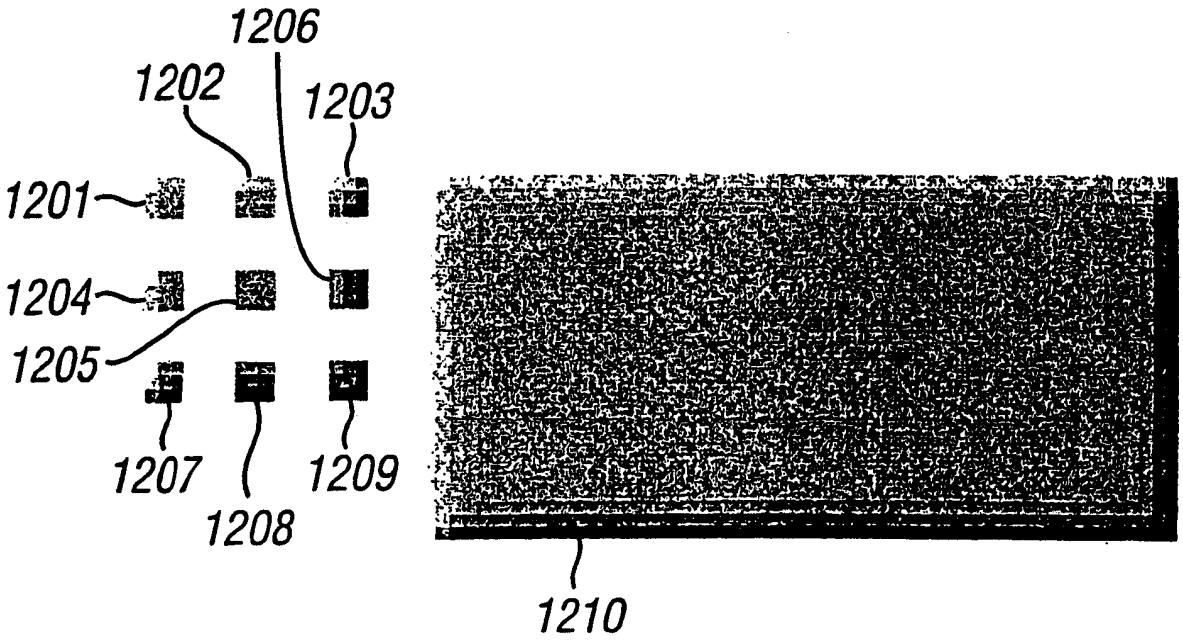

1206
1202
1203
1201
1204
1205
1207
1208
1209
1210

图 31

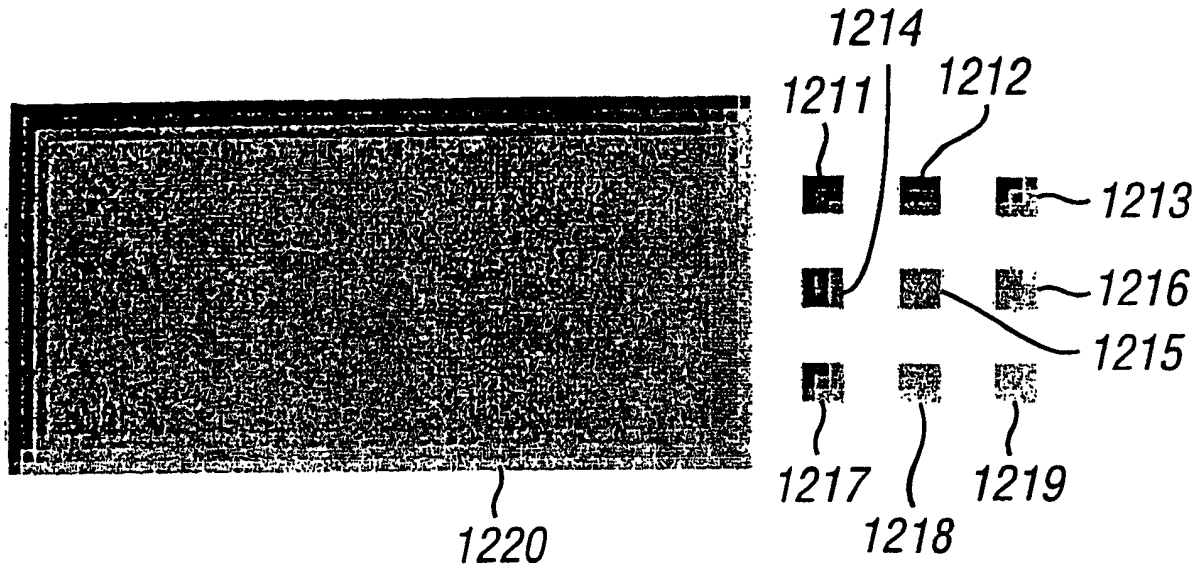

1214
1211
1212
1213
1216
1215
1217
1218
1219
1220

图 32

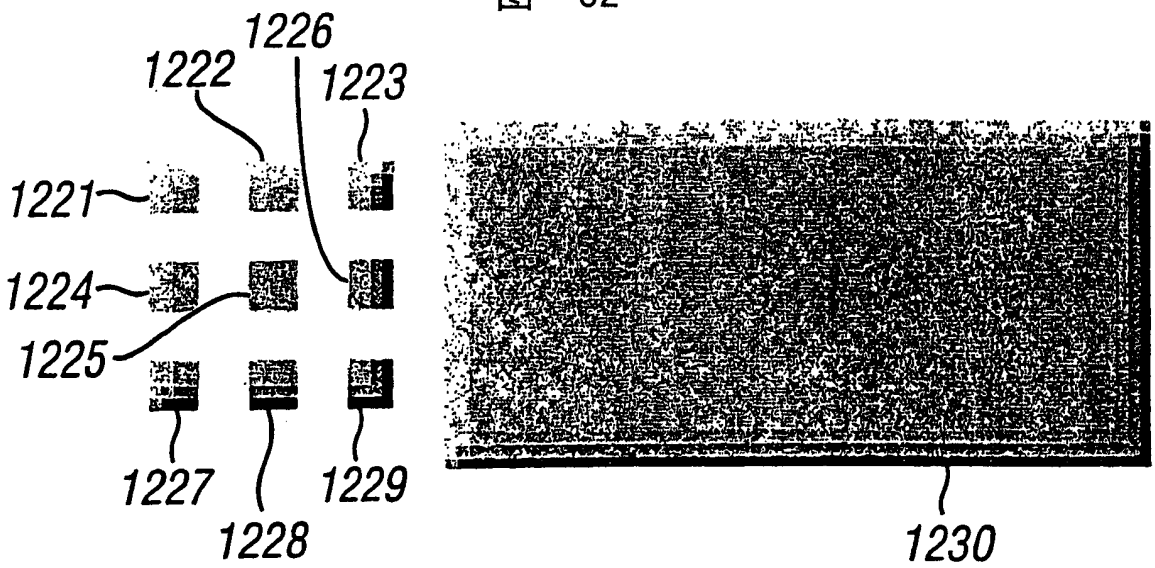

1226
1222
1223
1221
1224
1225
1227
1228
1229
1230

图 33

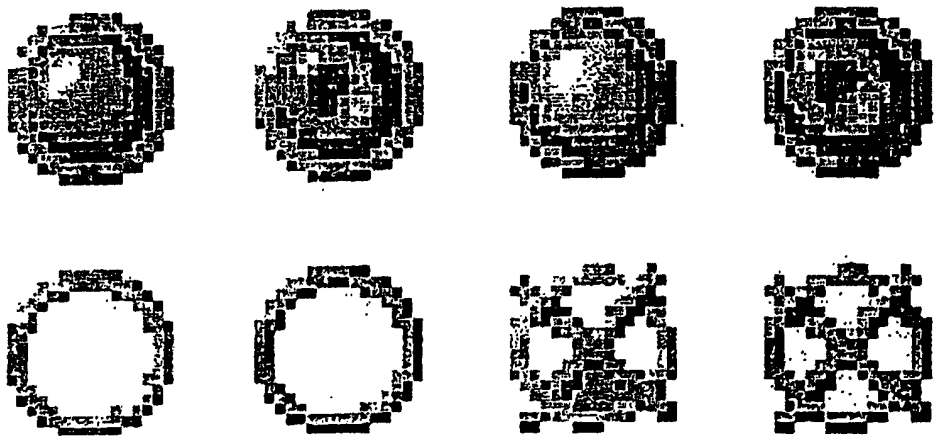


图 34

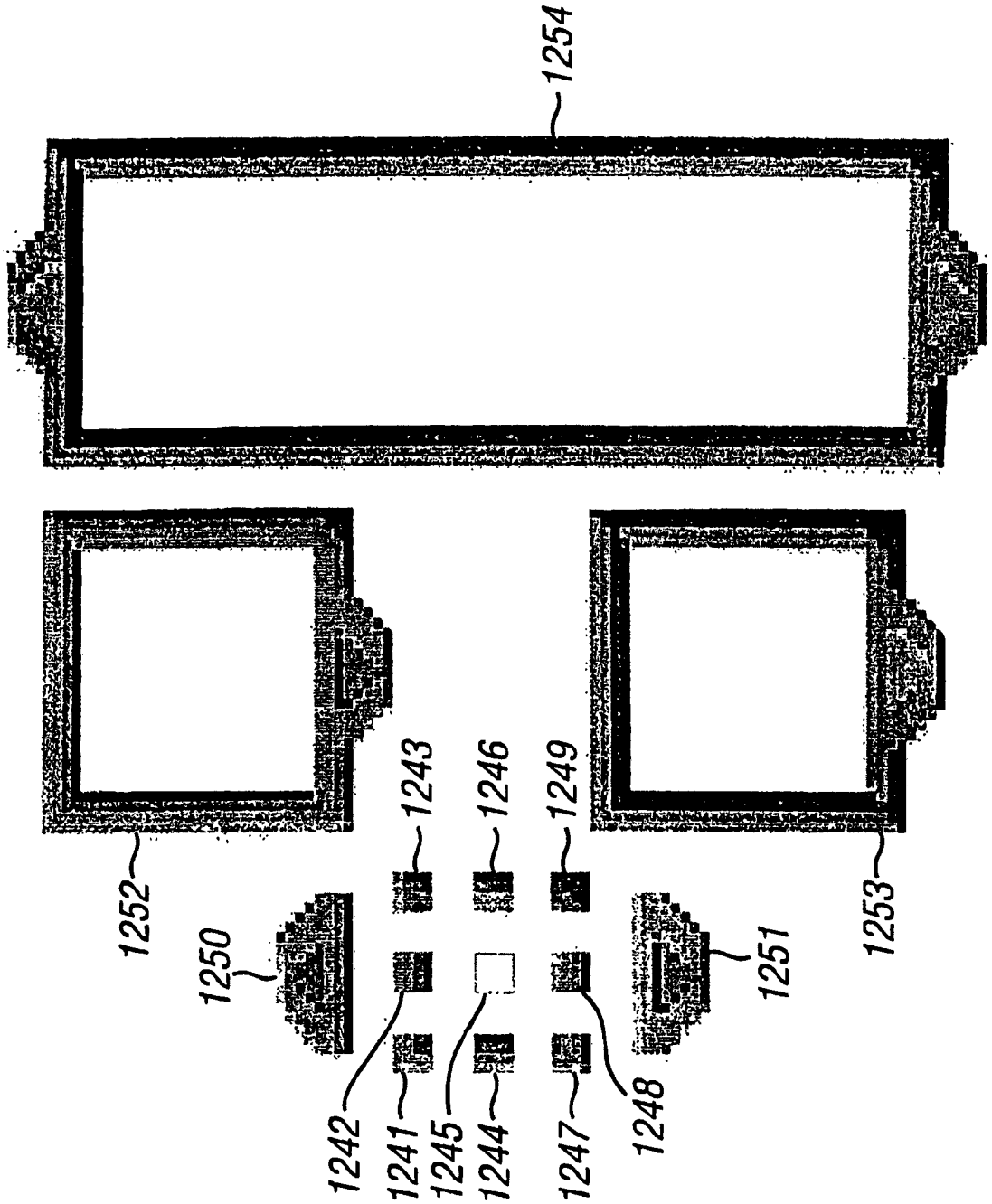

1254
1252
1250
1242
1241
1243
1245
1244
1246
1247
1249
1248
1251
1253

图 35

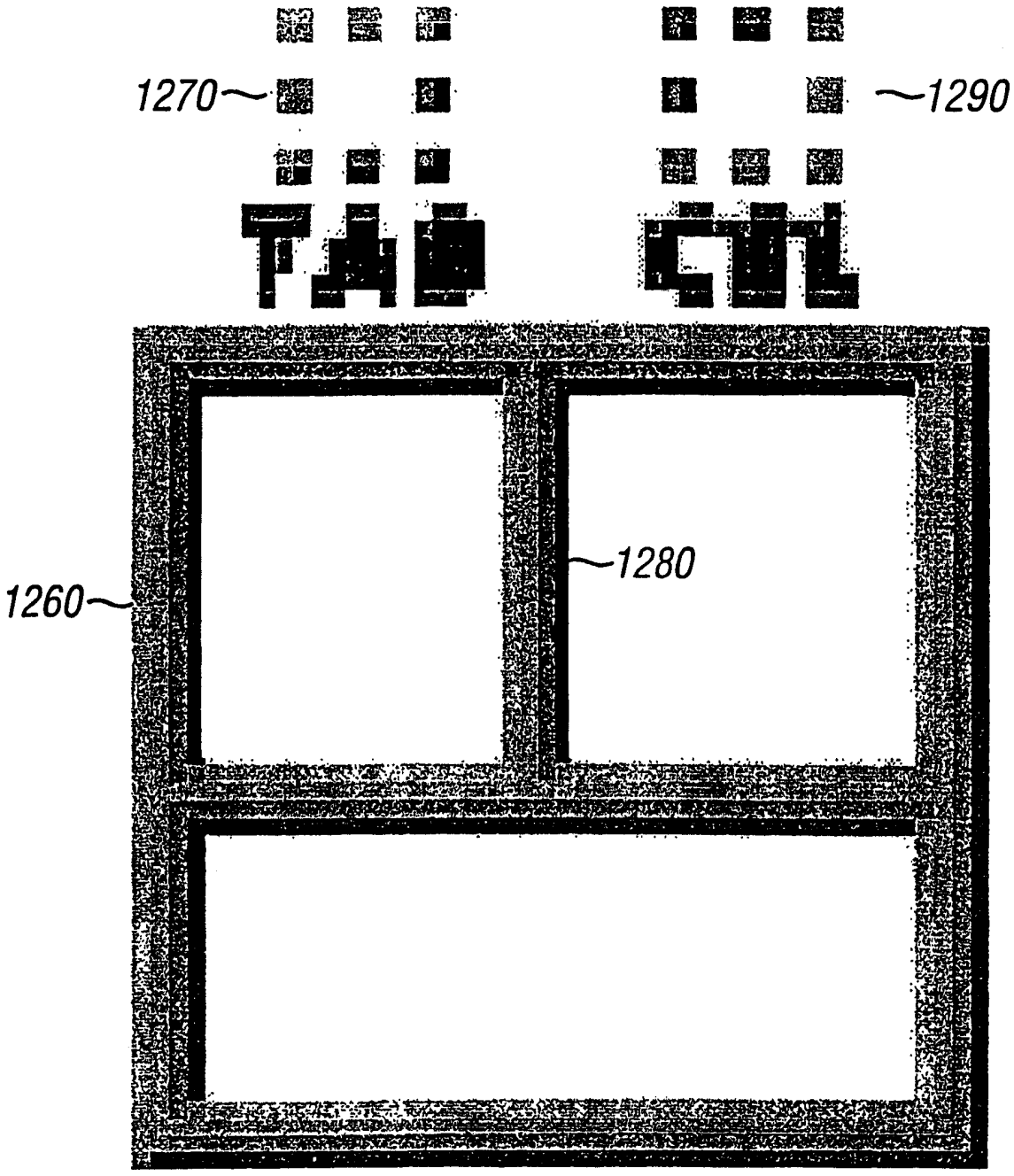


图 36

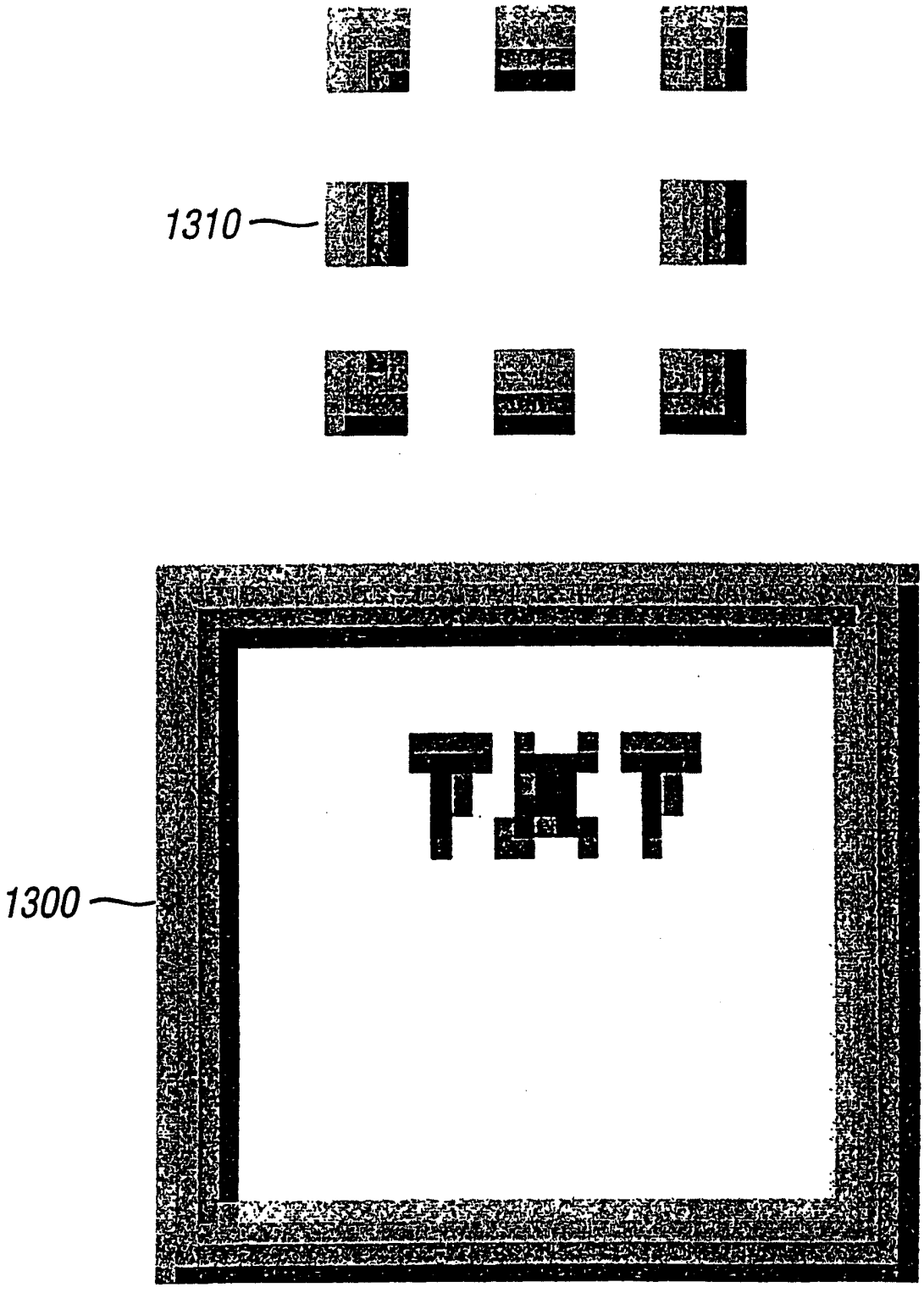


图 37

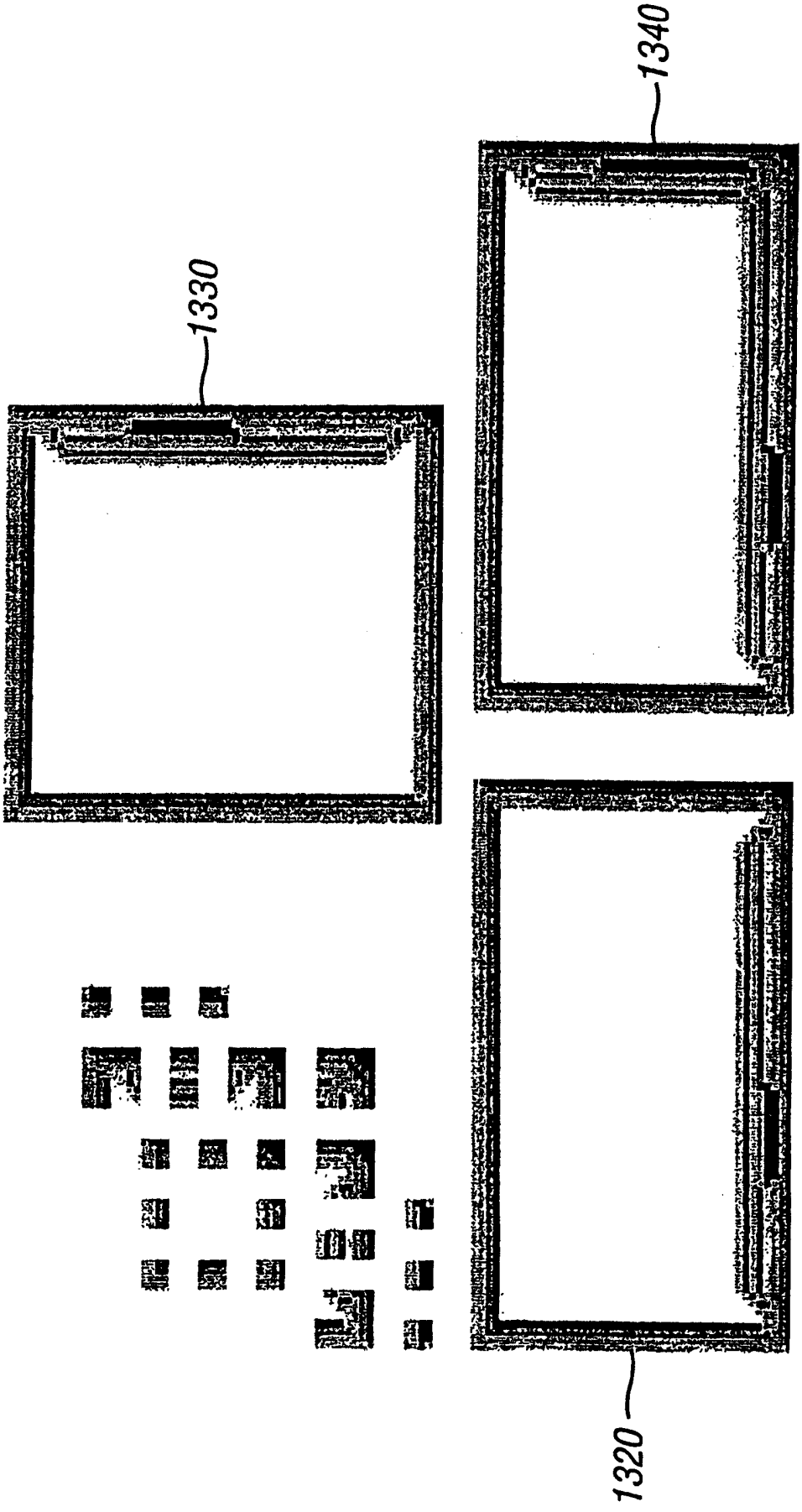


图 38

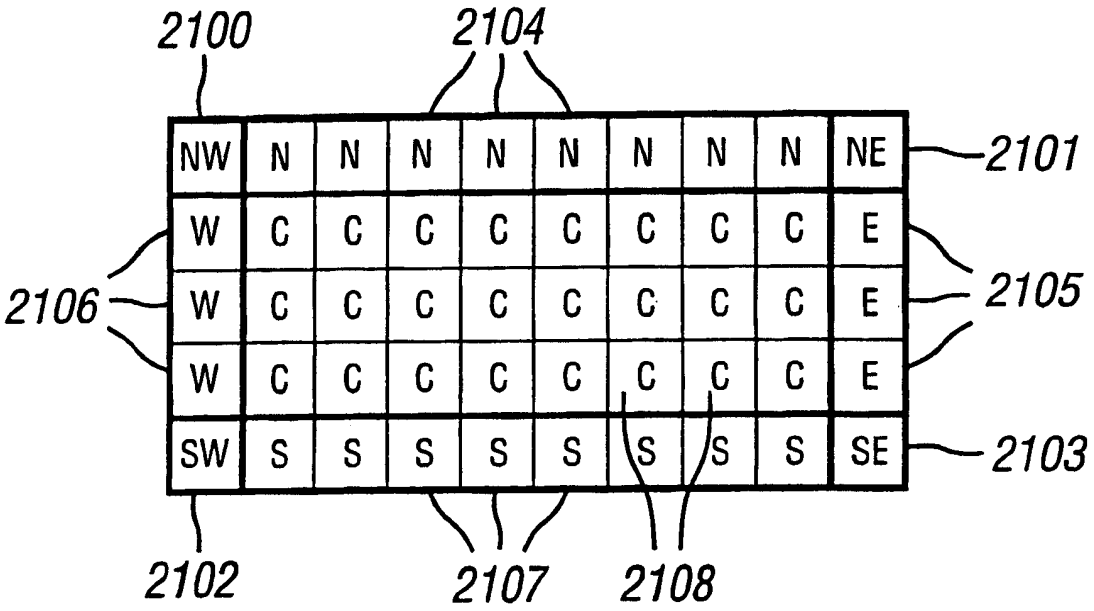

2100
2104
NW N N N N N N N N NE
W C C C C C C C C E
W C C C C C C C C E
W C C C C C C C C E
SW S S S S S S S S SE
2101
2105
2103
2106
2102
2107
2108

图 39

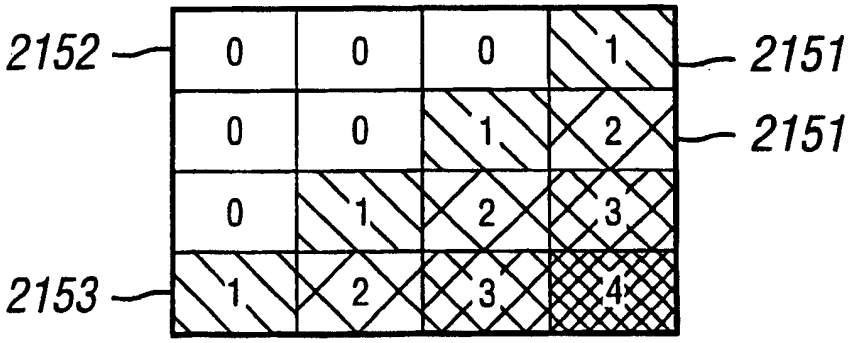

2152
0 0 0 1
0 0 1 2
0 1 2 3
1 2 3 4
2151
2151
2153

图 40

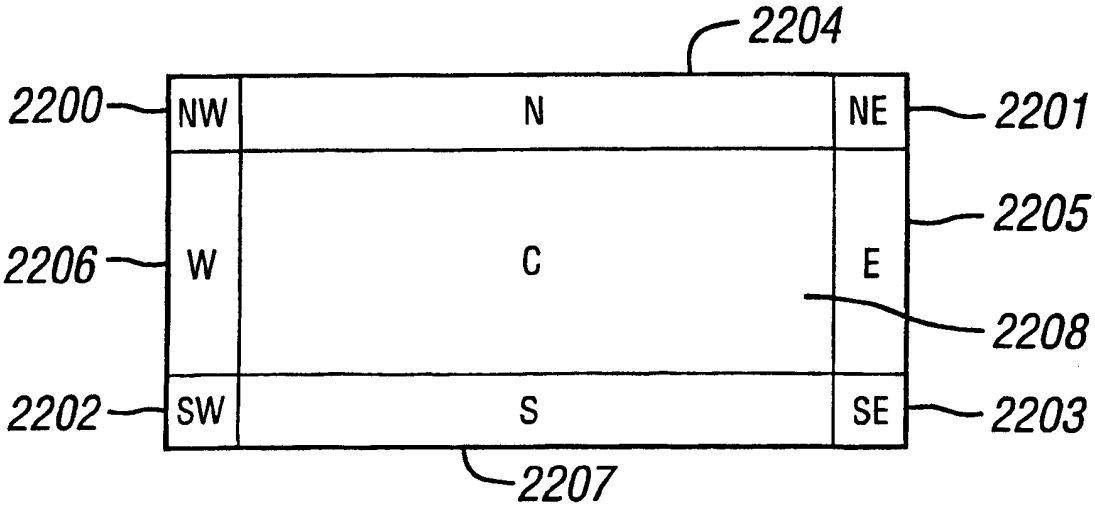

2204
2200
NW N NE
W C E
SW S SE
2201
2205
2206
2208
2202
2203
2207

图 41

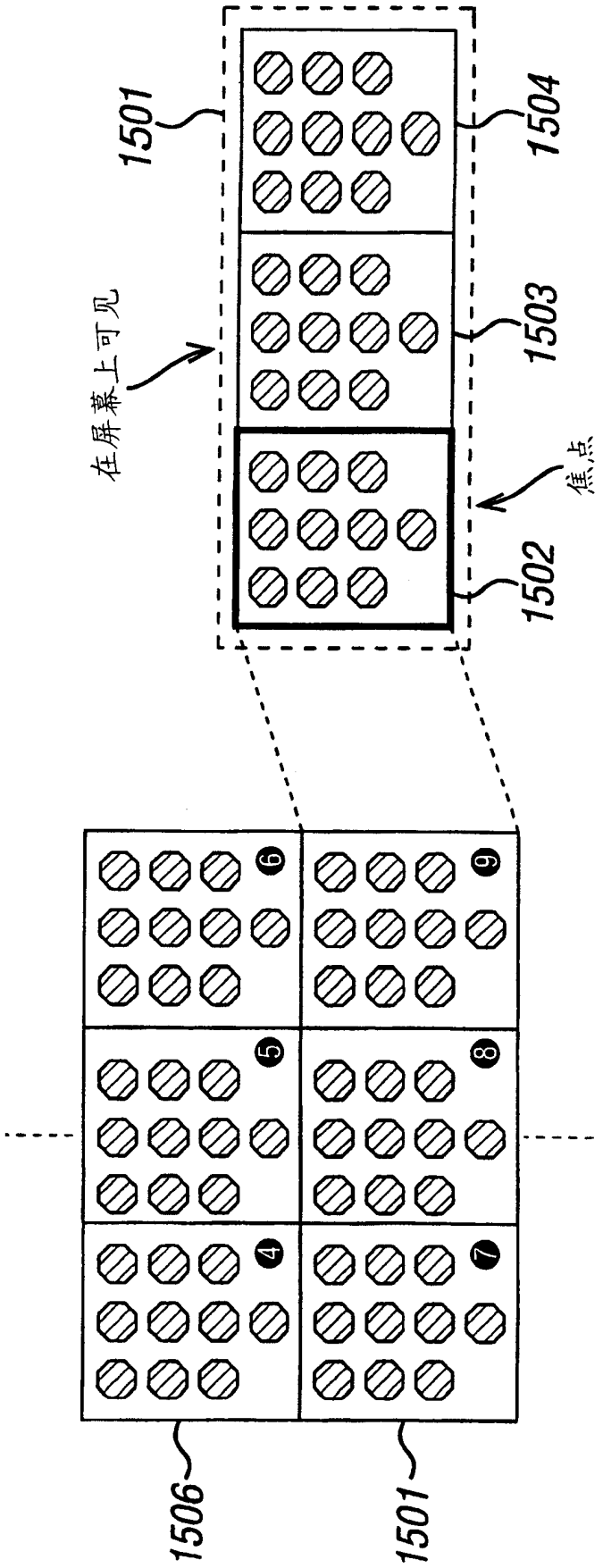


图 42

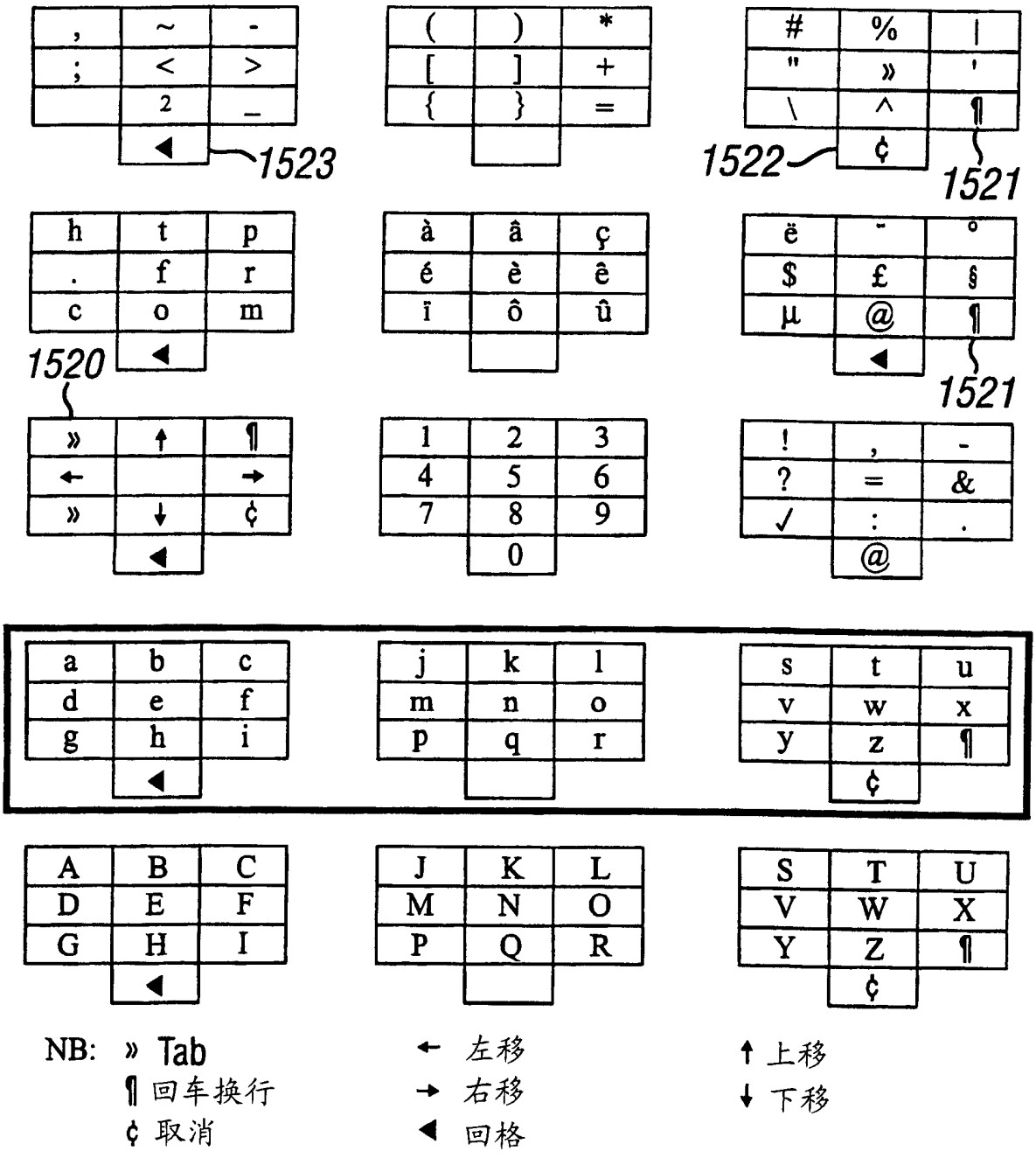

, ~ -
; < >
2 _
1523
( ) *
[ ] +
{ } =
% |
" » '
\ ^ ¶
¢
1522
1521
h t p
. f r
c o m
1520
à â ç
é è ê
ï ô û
ë - °
$ £ §
µ @ ¶
1521
» ↑ ¶
← →
» ↓ ¢
1 2 3
4 5 6
7 8 9
0
! , -
? = &
✓ : .
@
a b c
d e f
g h i
j k l
m n o
p q r
s t u
v w x
y z ¶
¢
A B C
D E F
G H I
J K L
M N O
P Q R
S T U
V W X
Y Z ¶
¢
NB: » Tab
¶ 回车换行
¢ 取消
← 左移
→ 右移
◀ 回格
↑ 上移
↓ 下移

图 43

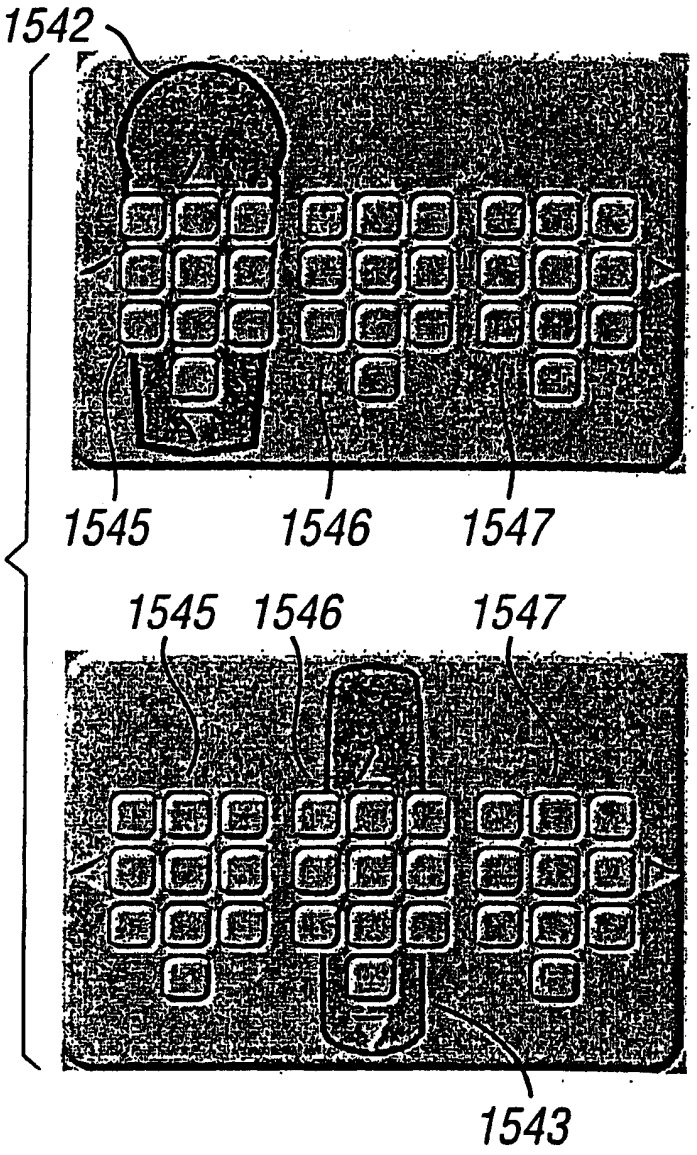


图 44

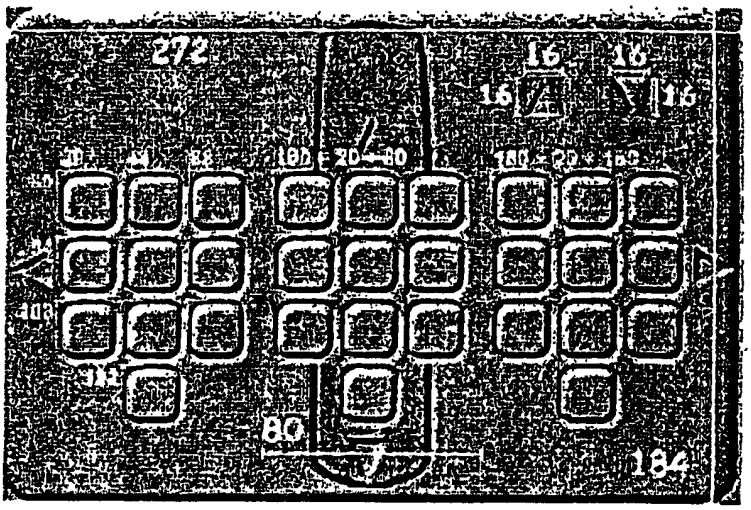


图 45