

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200580004701.1

[51] Int. Cl.

H04M 1/725 (2006.01)

G06F 17/30 (2006.01)

G06F 3/033 (2006.01)

[43] 公开日 2007 年 6 月 27 日

[11] 公开号 CN 1989762A

[22] 申请日 2005.2.21

[21] 申请号 200580004701.1

[30] 优先权

[32] 2004.2.19 [33] GB [31] 0403709.9

[86] 国际申请 PCT/GB2005/000641 2005.2.21

[87] 国际公布 WO2005/081094 英 2005.9.1

[85] 进入国家阶段日期 2006.8.11

[71] 申请人 高通剑桥有限公司

地址 英国剑桥

[72] 发明人 S·G·布特林 J·B·布劳克夫
N·C·布鲁克

[74] 专利代理机构 北京纪凯知识产权代理有限公司

代理人 沙 捷

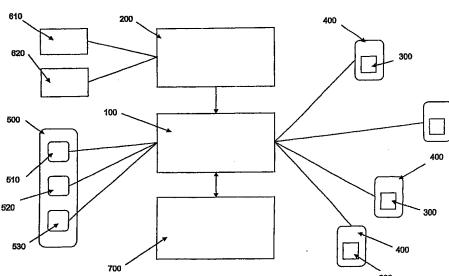
权利要求书 4 页 说明书 11 页 附图 4 页

[54] 发明名称

用户界面中菜单项的显示

[57] 摘要

提供了一种设备的用户界面，其中，该用户界面生成元素列表，元素列表过大而不能显示在 UI 区域内。确定要显示的列表元素的数目，然后从生成的元素列表中选择列表元素的适当子集。



1. 一种显示用户界面内多个用户界面元素的子集的方法，所述方法包括以下步骤：

(i) 确定可在所述用户界面内显示的所述多个 UI 元素的子集的大小；

(ii) 确定可被选择以在所述用户界面内显示的多个 UI 元素；

(iii) 从在步骤 (ii) 中确定的所述多个 UI 元素中选择所述 UI 元素的子集；和

(iv) 在所述用户界面内显示在步骤 (iii) 中所选的所述 UI 元素的子集。

2. 如权利要求 1 所述的方法，其中，响应用户输入，重复步骤 (iii) 来选择 UI 元素的另外的子集，然后重复步骤 (iv) 以在所述用户界面内显示所述 UI 元素的另外的子集。

3. 如权利要求 2 所述的方法，其中，所述用户输入包括激活用户输入装置，并且所述 UI 元素的另外的子集的选择和显示使菜单列表滚动。

4. 如权利要求 1 到 3 中任一项所述的方法，其中，所述多个 UI 元素被存储在单独的位置处，并且提供定义所述多个 UI 元素的位置的标记语言组件。

5. 如权利要求 4 所述的方法，其中，所述标记语言组件进一步定义所述被选择的 UI 元素的子集在列表中的显示。

6. 如权利要求 5 所述的方法，其中，模板与所述标记语言组件相关联，所述模板确定在所述列表中显示的所述被选择的 UI 元素的子集的外观。

7. 如权利要求 1 到 3 中任一项所述的方法，其中，所述多个 UI

元素被存储在单独的文件中，提供定义所述文件位置的标记语言组件，并且所述文件包括用于在所述用户界面内显示的一个或多个数据资源。

8. 如权利要求 7 所述的方法，其中，所述标记语言组件进一步定义所述被选择的 UI 元素的子集在列表中的显示。

9. 如权利要求 8 所述的方法，其中，模板与所述标记语言组件相关联，所述模板确定在所述列表中显示的所述被选择的 UI 元素的子集的外观。

10. 如权利要求 3 到 9 中任一项所述的方法，其中，所述被选择的 UI 元素的子集的所述列表包括一个或多个另外的列表，所述一个或多个另外的列表中的每个列表由唯一表达式来标识。

11. 一种数据载体，其包括用于执行权利要求 1 到 10 中任一项所述方法的计算机可执行代码。

12. 一种设备，其包括显示器和用户界面，所述设备被配置成，在使用中，执行以下操作：

- (i) 确定可在所述用户界面内显示的多个 UI 元素的子集的大小；
- (ii) 确定可被选择以在所述用户界面内显示的多个 UI 元素；
- (iii) 从在步骤 (ii) 中确定的所述多个 UI 元素中选择所述 UI 元素的子集；和
- (iv) 在所述用户界面内显示在步骤 (iii) 中所选的所述 UI 元素的子集。

13. 如权利要求 12 所述的设备，其中，所述设备进一步包括用户输入装置并被配置成，在使用中，响应用户输入，以重复步骤 (iii) 来选择 UI 元素的另外的子集，然后重复步骤 (iv) 以在所述用户界面内显示所述 UI 元素的另外的子集。

14. 如权利要求 12 或 13 所述的设备，其中，所述设备响应所述用户输入装置的激活，从而所述 UI 元素的另外的子集的选择和显示使菜单列表滚动。

15. 如权利要求 12 到 14 中任一项所述的设备，其中，所述设备进一步包括存储装置，并且所述多个 UI 元素被存储在单独的位置，并且提供定义所述多个 UI 元素位置的标记语言组件。

16. 如权利要求 15 所述的设备，其中，所述标记语言组件进一步定义所述被选择的 UI 元素的子集在列表中的显示。

17. 如权利要求 16 所述的设备，其中，模板与所述标记语言组件相关联，所述模板确定在所述列表中显示的所述被选择的 UI 元素的子集的外观。

18. 如权利要求 12 到 14 中任一项所述的设备，其中，所述设备进一步包括存储装置，并且所述多个 UI 元素被存储在单独的文件中，其中，提供定义所述文件位置的标记语言组件，并且所述文件包括用于在所述用户界面中显示的一个或多个数据资源。

19. 如权利要求 18 所述的设备，其中，所述标记语言组件进一步定义所述被选择的 UI 元素的子集在列表中的显示。

20. 如权利要求 19 所述的设备，其中，模板与所述标记语言组件相关联，所述模板确定在所述列表中显示的所述被选择的 UI 元素的子集的外观。

21. 如权利要求 15 到 20 中任一项所述的设备，其中，所述被选择的 UI 元素的子集的所述列表包括一个或多个另外的列表，所述一个或多个另外的列表中的每个由唯一的表达式来标识。

22. 如权利要求 11 到 22 中任一项所述的设备，其中，所述设备包

括无线通信装置。

23. 一种设备，其包括处理装置、存储装置、显示器、用户输入装置、无线通信装置和用户界面，其中，所述设备被配置成执行权利要求 1 到 10 中任一项所述的方法。

用户界面中菜单项的显示

技术领域

本发明涉及一种用于在用户界面中显示菜单项的方法，特别涉及与移动通信网络一起使用的设备所使用的这样的方法。

背景技术

许多移动设备中共有的局限性之一是显示屏非常小，在显示菜单时，不可能始终在屏幕上同时显示所有的菜单项。传统的方法趋向于将所有菜单项连同相关的图标或图形加载到存储器中，然后在用户上下滚动菜单时适当地显示它们。对于诸如移动电话的资源有限的设备，和对于使用标记语言来绘制设备显示和/或为设备提供操作软件的设备，该方法并不是有效的技术。

发明内容

根据本发明的第一方面，提供了一种显示用户界面内多个用户界面元素的子集的方法，该方法包括以下步骤：(i) 确定可在用户界面内显示的多个 UI 元素的子集的大小；(ii) 确定可被选择以在用户界面内显示的多个 UI 元素；(iii) 从在步骤 (ii) 中确定的多个 UI 元素中选择 UI 元素的子集；和 (iv) 在用户界面内显示在步骤 (iii) 中所选的 UI 元素的子集。响应用户输入，可重复步骤 (iii) 来选择 UI 元素的另外的子集，然后重复步骤 (iv) 以在用户界面内显示该 UI 元素的另外的子集。

根据本发明的第二方面，提供了一种数据载体，其包括用于执行上述方法的计算机可执行代码。

根据本发明的第三方面，提供了一种设备，其包括显示器和用户界面，该设备被配置成，在使用中，执行以下操作：(i) 确定可在用户界面内显示的多个 UI 元素的子集的大小；(ii) 确定可被选择以在用户界面内显示的多个 UI 元素；(iii) 从步骤 (ii) 中确定的多个 UI 元素

中选择 UI 元素的子集；和 (iv) 在用户界面内显示在步骤 (iii) 中所选的 UI 元素的子集。

根据本发明的第四方面，提供了一种设备，其包括处理装置、存储装置、显示器、用户输入装置、无线通信装置和用户界面，其中，该设备被配置成执行如上所述的方法。

附图说明

图 1 示出了结合本发明的系统的示意图；

图 2 更详细地示出了服务器的结构和操作；

图 3 示出了移动设备的软件的示意图；

图 4 示出了根据本发明的一个实施例的包括用户界面的设备的示意图。

具体实施方式

现将仅通过例证的方式，参照附图，对本发明进行说明，其中，图 1 示出了结合本发明的系统的示意图。该系统包括服务器 100、内容工具箱 200、移动设备 300、操作支持系统 (OSS) 700、内容供给 500 和用户界面 (UI) 源 600。在使用中，服务器 100 向移动设备 300,301,⋯, 传送内容数据和 UI 数据，每个移动设备包括软件包 400。服务器 100 与 OSS 700 通过接口连接，该 OSS 按照惯例被用于操作移动网络，例如，记帐、帐户管理等。服务器 100 还与内容工具箱 200 通过接口连接：内容工具箱从 UI 源 600,601,⋯, 接收数据，并将 UI 数据打包，使得服务器能将打包的 UI 数据传输到移动设备 300 内所包括的软件包 400。服务器从多个内容供给接收数据，并且该数据被处理并打包，使得它能够被传送到软件包 400，或者使得移动设备 300 能使用软件包 400 存取数据。

该系统可被视为分成三个独立的域：操作者域 50 包括由移动网络操作者 (MNO) 操作的系统和仪器；用户域 60 包括多个移动设备；以及，三方域 70 包括可由若干不同实体控制或者操作的内容供给和 UI 供给。

图 2 更详细地示出了服务器 100 的结构和操作。服务器 100 包括

发布组件 110 和内容服务器组件 150。发行组件包括数据库 111、导入队列 112、内容工具箱接口 113、用户界面 114 和目录 115。在操作中，发布组件在内容工具箱接口从内容工具箱接收内容。内容以包括一个或者多个 trig 和一个或多个 triglet 的包 210a,210b, \dots , (见下文) 的形式出现。trig 是用于诸如移动电话的移动设备的用户界面，并且 triglet 是能够用于扩展或改变 trig 的数据文件。如果一个包包括多于一个 trig，则其中一个 trig 可以是主 trig，其它 trig 可从主 trig 导出。

优选的是，软件使用 TrigML 运行，TrigML 是 XML 应用程序，并且标记语言绘制器 410 绘制用于在移动设备 300 上显示的 TrigXML 代码。标记语言绘制器还使用 TrigML 分析器来分析 TrigML 资源，在设备屏幕上显示内容，并控制送受话器上内容的替换和显示。本地 UI 绘制器用于显示不需使用 TrigML 就能显示的 UI 组件，并用于显示错误消息。

以设备特有的方式来提供和安装软件 400。例如，对于诺基亚 60 系列设备，使用 SIS 文件来安装软件，而对于 MS 智能手机设备，使用 CAB 文件来安装软件。类似地，以设备特有的方式处理软件升级，并且可通过无线电提供升级。可以以更受限的格式来提供软件，例如，作为仅绘制它的内置内容的独立的应用程序，即，以后不能向与应用程序一起提供的 trig 添加另外的 trig。

trig 管理器 465 提供对资源管理器 430 和标记语言绘制器的接口。通常它负责 trig 管理。这包括：保持获知使用中的 trig，改变当前 trig，启动时选择 trig，选择另外的 trig 作为被破坏的 trig 的备用，维持已安装的 trig 的集合，识别特定 trig 被安装到资源管理器的位置，并读取 trig 的更新通道定义并适当地配置更新管理器。

资源管理器提供了在设备上永久存储的抽象概念，即，将文件作为真实文件或数据库中的记录来存储。资源管理器提供对标记语言绘

制器和更新管理器的文件系统接口。资源管理器负责处理文件路径逻辑，区分真实资源文件和执行器属性，将 trig 相关的路径映射到绝对路径上，通过接口连接 trig 管理器并提供对更新管理器的修改接口。

资源管理器还负责确保存储在永久存储中的资源的完整性，特别是在面对诸如设备断电的不可预料的中断的时候。资源管理器不知道当前使用的 trig。它的界面是线程安全的（因为它可被更新管理器和绘制器二者从不同的线程使用）。

更新管理器处理 trig 和 triglet 的接收和应用。更新管理器提供对绘制器和 trig 管理器的接口，并负责：当被绘制器指示时，初始化手动更新；当被 trig 管理器如此配置时，控制和实现自动更新通道；指示手动更新的进度，并在网络连接和/或设备电源意外丢失后恢复更新。更新分组格式可定义为 XML 模式的二进制序列化。

支持管理器提供了用于其它组件用来报告发生事件或错误的接口。根据错误的严重程度，支持管理器将记录事件，并且/或者给出错误消息弹出框。

执行器管理器 440 监督软件中出现的执行器 445 的集合。执行器管理器 440：当内容正向执行器发送事件时，被绘制器使用；被想要通知属性值已经改变的执行器使用；以及，被想要发出事件的执行器使用（见下文）。

软件可包括运行至少两个线程的多线程应用程序，比较可能取决于包括执行器的数目和种类。软件主要在被称为主线程的一个线程中运行。主线程用于运行与其它组件同步通信的绘制器。执行器通常具有绘制器的同步接口。如果执行器由于其功能需要额外的线程，则执行器负责管理线程间通信。优选地，使用轻量级的消息框架以防止多个执行器需要线程间通信情况下的不必要的代码重复。

除了主线程，更新管理器运行网络线程。网络线程被用于下载更新分组，并与主线程相分离，以允许绘制器在分组到达前保持不受影响。更新管理器负责处理线程间消息传递，使得更新管理器在应用更新分组中定义的改变时与绘制器和资源管理器同步通信。

软件的内存分配策略是针对平台的。在 MIDP 平台上，该软件仅使用系统堆和无用单元收集程序用于其所有存储需求。无论何时在保

持无用单元收集程序可预测和不遭受操作中的意外停止的尝试中发生内容代替事件，无用单元收集程序被强制打开。假设任何存储分配可能失败，在这种情况下，软件将删除其所有对对象、无用单元收集程序的引用（reference），并重新启动——假设软件已经成功启动并绘制了第一页。

在基于 C++ 的平台上，将从系统堆中进行预分配和按需分配的混合分配。在启动时按需分配启动所需的所有内存，这里出现的任何故障都会导致退出软件（如果可以，伴有消息）。在成功启动后，预先分配绘制内容文件模型所需的内存。如果内容是使用比预定限制少的内存来创建的，可确保绘制。可另外使用 RAM 用于软件快速操作所需的各种高速缓存。在低内存条件下，将从软件中释放这些导致绘制性能慢的高速缓存。

绘制器接收关于按键的信息。如果在编译时没有为一个键配置行为，则它作为 TrigML 内容事件被发送到当前焦点元素。随后遵照 TrigML 的标准事件处理逻辑所定义的那样处理该内容事件。

例如，如果键被按下，“按键”事件连同为该相关键设置的参数被传送至绘制器。当该键被释放时，“!按键”事件被传送至绘制器。如果按键持续一段延长的时间，“长按键”事件被传送至绘制器。在释放时，“! 长按键”和“!按键”两个事件均被传送至绘制器。

无论何时启动软件，它都执行以下操作：

- 检查并继续中断的更新处理；
- 检查并处理文件系统中的更新（或者预先规定的，或者通过一些其它方法安装在文件系统中）；
- 如果当前 trig 已知，则启动当前 trig（其可为最后运行的 trig）；
- 如果没有设置当前 trig，可启动被标记为“默认”的 trig；
- 在默认 trig 显示失败时，则将选择按照名称字母顺序的第一有效 trig。

通过加载定义的资源名启动/默认，启动 trig。启动/默认中定义的 TrigML 被分析为内容根节点的新内容。

在安装软件后第一次由软件运行 trig 时，通过加载资源名启动/首次，来启动 trig。软件可将 trig 是否已经被运行记录在位于该 trig 的顶

层文件夹中的文件里。根据移动设备使用的平台，软件的自动启动可被设置为编译时的配置选项。此外，在自动启动后将软件置于后台也可以是编译时的配置选项。

显示启动器可作为应用程序图标显示给用户，并选择该图标会启动带有该启动器所指定的 trig（该 trig 可由启动器图标和/或名称指示）的该软件。当使用启动器启动 trig 时，可指定“进入点”参数。该参数为在“启动”文件夹中找到的文件的资源名。如果该 trig 在之前从未运行过，则不使用该文件，在此情况下，使用称为“首次”的文件来代替。

软件使用设备上的虚拟文件系统内存储的内容资源文件。该文件系统被称为是虚拟的，是因为其可以不被实现为标准文件系统，然而，所有对资源的引用都是文件路径，好像被存储在文件夹和文件的分级系统中一样。此外，该软件存储以下信息中的一些或全部：使用统计；活动用户计数；TrigManager 状态；TrigML 分段和更新通道定义（序列化为二进制 XML）；PNG 图像；被编码为 UTF-8 OTA 并随后在平台特有的编码过程中被存储的纯文本；其它平台特有的资源，例如，铃声文件、背景图像等。

在执行器属性值改变时，或在文件由 trigLet 替换时，能够改变文件系统中的文件。当 attrs 目录中的文件改变时，立即通知绘制器，并且内容树的相关分支被更新并刷新。当图像和文字资源被改变时，绘制器表现为使受到影响的资源被立即重载（整个内容树或仅受到影响的分支被刷新）。当 TrigML 分段改变时，绘制器表现为它没有得到通知，并继续显示其当前的可能过时的内容。这是为了避免软件需要保留<include（包括）>元素和当前内容的<load（加载）>历史。

将软件 400 以设备特有的方法提供给移动设备。可提供一个或多个 trig 作为安装的一部分，例如，存储为未被压缩的更新分组。在启动时，分组可被扩展并安装到文件系统。

执行器 445 是发布属性值并处理和发出事件的组件。执行器与绘制器同步通信。如果执行器需要异步行为，则执行器负责管理绘制器的主线程以外的线程并与之通信。

许多移动设备中共有的局限性之一是显示屏非常小，在显示菜单

时，不可能始终在屏幕上同时显示所有的菜单项。传统的方法趋向于将所有菜单项连同相关的图标或图形加载到存储器中，然后在用户上下滚动菜单时适当地显示它们。对于诸如移动电话的资源有限的设备，和对于使用标记语言来绘制设备显示和/或为设备提供操作软件的设备，该方法并不是有效的技术。

通过提供一种有效的技术，使有限数目的菜单项被加载到存储器，使得这些菜单项可被设备同时显示，来解决这一问题。当用户上下滚动菜单时，不再显示的项目被删除，而当前正显示的菜单项被加载到存储器中。

优选地，这可通过使用 TrigML 中的`<griddata>`元素以定义一些数据的查看列表来实现，其中数据被存储在文件系统的文件夹内，而对每个菜单项来说，该列表外观都具有相同的结构。`<griddata>`元素包括指定数据可被放置在其中的文件夹的‘repeat-over（重复）’属性。`<griddata>`的单一子元素是列表中每个菜单项的外观的模板。

该模板可使用例如‘\$\$’的特定符号来表示迭代程序。这是每当该模板被实例化（instantiated）时发生改变的模板变量；例如

```
<griddata repeatover="news/headlines">
    <text res="news/headlines/$$/title.txt"/>
</griddata>
```

其中文件夹 news/headlines/包括：

```
0/title.txt
1/title.txt
2/title.txt
3/title.txt
```

这可显示 4 个项目的列表，分别由指向‘news/headlines/\$\$’文件夹中‘title.txt’资源的简单的`<text>`元素描述。在源数据中具有的项目比显示器上`<griddata>`元素的空间所容许的项目更多的情况下，`<griddata>`元素仅显示能够显示的那些项目。当用户滚动列表时，`<griddata>`元素相应地改变‘data-window（数据窗口）’。该技术的优点在于，仅当前显示所需的资源被实际加载到存储器中，这减少了内存使用，并减小了绘制该项目列表所花费的时间。

一种相似策略可被用于确定列表显示的次序。如果‘repeat-over’

属性的目标为文件而不是文件夹，则该文件可被认为包括迭代程序中使用的资源名列表。例如，

```
<griddata repeatover="football/league">
    <text res="football/teams/$$/name.txt"/>
</griddata>
```

其中，文件 football/league 包括：

Manchester

Arsenal

Chelsea

文件夹 football/teams/包括：

Manchester/name.txt

Arsenal/name.txt

Chelsea/name.txt

并且每个 name.txt 是包括球队名的文本文件。这使得与球队相关联的测试文件以所定义的次序被显示在设备显示器的定义的区域内。

下面给出其另一个实例，模板使得显示出姓氏和名字数据的列表，该列表包括三个条目。

```
<gridlist id="myList" repeatover="path/to/list" rows="3">
    <group>
        <text res="$$/surname"/>
        <text res="$$/firstname"/>
    </group>
</gridlist>
```

该方法可被扩展以使得该列表能够嵌套到其它列表内。遵照以上方法，如果存在许多不同列表，则在 ‘\$\$’ 操作符引用哪个列表上存在不明确性。该问题可通过将 ‘\$\$’ 替换为唯一识别其引用的列表的表达式来解决，例如：

{/elem/myList/index}

除了可唯一引用列表，该方法还可使项目引用该列表，但并不实际位于该列表中。例如，窗口的标题栏可包括与列表中所选项目相关的文字，该标题栏中的文字可随着列表滚动而改变。

在数据被文件系统以外的装置访问的情况下，例如，它被存储在数据库中，或其由其它软件组件即时生成，如果使用虚拟文件系统 435，

则仍可使用该方案，该方案可将文件系统接口映射到底层数据提供程序。这意味着该内容仍可被如上所述布置，但也可以有效数据存储和检索的方法提供该数据。

可使用软件激励的 HTTP-GET 请求来进行 OTA 发送的更新。GET 请求被送往与更新相关联的 URL。HTTP 响应的主体为载有更新分组格式的数据的二进制文件。在与绘制器线程相独立的线程中处理数据接收。对于后台更新（自动启动的），这允许用户连续操纵（navigating）UI。对于前台更新（手动启动的），这允许绘制器线程显示进度条并监听取消指示。

用于解包和安装更新的算法是设备特有的。然而，重要的是，该算法对于不可预料的中断（例如，电源故障）是安全的，从而文件系统不会出现变坏或不可恢复的状态。这可通过使用两个线程（网络线程和绘制器线程），由网络线程执行尽可能多的更新处理以便于绘制器线程中断最短可能时间的目的来达到。

还需要考虑其它故障模式：如果 HTTP-GET 不能启动，或遇到 HTTP 错误响应码，则此种更新的尝试被放弃，而使用重试策略以在后续阶段开始新的更新尝试。在网络信号的损失使 HTTP 响应中断的情况下，删除任何临时文件，并使用重试策略以在后续阶段重启更新尝试。如果更新头指示更新有效载荷大小可能太大而不能安装到该设备上，如果更新需要不兼容版本的软件，或如果更新已经存在于该设备上，则删除该头数据文件，并取消更新尝试和任何后续重试。

TrigML 分段为包括文本 TrigML 的文件，这些段内的资源引用为虚拟文件路径。这些虚拟文件路径到真实文件路径的映射是由 TrigDefinition 文件定义的。该文件还定义了该 trig 的其它属性。当用于编译 trigLet 时，该文件还定义如何将输入的 TrigML/PNG/Text 资源映射到 trig 的虚拟文件系统的修改上。

为了成功绘制移动设备的用户界面，标记语言必须具有以下性质：简洁的页面定义，一致的布局规则，可在小型绘制器中实现，提供多层和任意重叠的内容、事件模型，仅需要重绘在 UI 的页面间必须改变的显示区域，包括读取接收事件和发送事件的属性值的平台钩子（hook），可扩展性，和图形表示上的灵活性。TrigML 提供这些特性，

并且，提供所需功能的属性和元素的综述可在 2004 年 2 月 19 日提交的我们的同时待审的申请 GB0403709.9 中找到。

希望使重新标记（re-branding）UI 和生产更新的连续流的成本最小。这是通过从生成过程通过数据传输向用户提供有效的信息流来实现。被称为包的容器被用于 UI、UI 更新和第三方参与模板。容纳第三方生产、测试和传递标记的 UI 和更新的所有必要信息。

许多不同 UI 可从公共基础获得。典型地，公共基础本身可实现大部分界面，并且从其获得的 trig 可实现其上小的更改，诸如标记。TrigLet 可从 Trig 获得，并且该 TrigLet 可覆盖任何它所选的来自父 Trig 的资源（可选地，它可引入它自身的资源）。应注意，此处“资源”也称为 TrigML，从而，Trig 的行为和布置可由 TrigLet 进行更改，正如其替换单个图像或单条文本一样简单。

包可包括一个或多个基 trig(即，不是从任何其它 trig 获得的 Trig)，从基 Trig 获得的一个或多个多重(multiple)trig，从任何 trig 获得的多个 triglet 以及多个从其它 triglet 获得的多个 triglet。

包格式为将所有该信息存储为序列化目标的不透明二进制格式。该包可包括许多资源，诸如图像、文字、URL、更新通道、振铃文件、墙纸、本地应用程序等。每个资源包括关于如何浏览、编辑或删除资源的许可信息。该包可被用于开发具有诸如显示尺寸、RAM 容量的不同能力的移动设备的 trig 和/或 triglet。为了对此进行简化，可定义若干分层结构，并且数据资源或 TrigML 元素在分层结构中被分类。当 trig 或 triglet 从包中被编译时，可为特定设备选择并编译最适当的资源或 TrigML 元素。

图 4 示出了根据本发明的一个实施例的包括用户界面的设备 800 的示意图。该设备包括显示用户界面 815 的显示器 810，和使用户与用户界面 815 能够进行交互的用户接口装置 820。处理器 830 执行存储在一个或多个存储装置 840 中的软件，并且可提供一个或多个无线通信接口 850，使得能够与其它设备和/或通信网络进行通信。可容纳一个或多个电池 860 用于为设备提供电源，设备也可包括容纳电源和/或通信电缆的接口。

这些组件和接口的性质将取决于设备的性质。应该理解，这种用

户界面可在移动式或蜂窝电话送受话器中实现，但其也适用于诸如数码相机、个人数字助理、数字音乐播放器、GPS 导航设备、便携式游戏控制台等的其它移动设备。此外，它还适用于诸如笔记本电脑或台式机的包括用户界面的其它设备。

用户界面装置可包括多个按钮，诸如数字键盘或字母数字键盘或者触摸屏或其它类似物。一个或多个存储设备可包括诸如存储卡的一种形式的非易失性存储器，使得存储的数据在掉电的情况下也不丢失。可提供 ROM 存储装置，用于存储无需更新或改变的数据。当较快的响应时间支持频繁存取数据的超高速缓存时，可提供某种 RAM 用于临时存储。设备还可接受用户可移动存储卡，并且可选地，硬盘驱动器可被用作存储装置。所使用的存储装置将通过权衡设备大小、功率消耗、所需存储容量等不同需求来确定。

这样的设备可实际上与任何无线通信网络相结合来实现，例如，第二代数字移动电话网络（即，GSM、D-AMPS），所谓的 2.5G 网络（即，GPRS、HSCSD、EDGE）、第三代 WCDMA 或 CDMA-2000 网络和对这些或者类似网络的改进和派生。在建筑物中或校园内，也可使用其它技术，诸如蓝牙、IrDa 或无线 LAN（不管基于无线电还是光纤系统）。USB 和/或火线（FireWire）连接也可被提供用于与其它设备进行数据同步，和/或用于电池充电。

可在数据载体上设置用于实现这些方法和/或用于如上所述地配置设备的计算机软件，数据载体诸如软盘、CD-ROM、DVD、非易失性存储卡等。

本申请要求 2004 年 2 月 19 日提交的第 0403709.9 号 UK 专利申请的优先权，其内容结合在本文中作为参考。

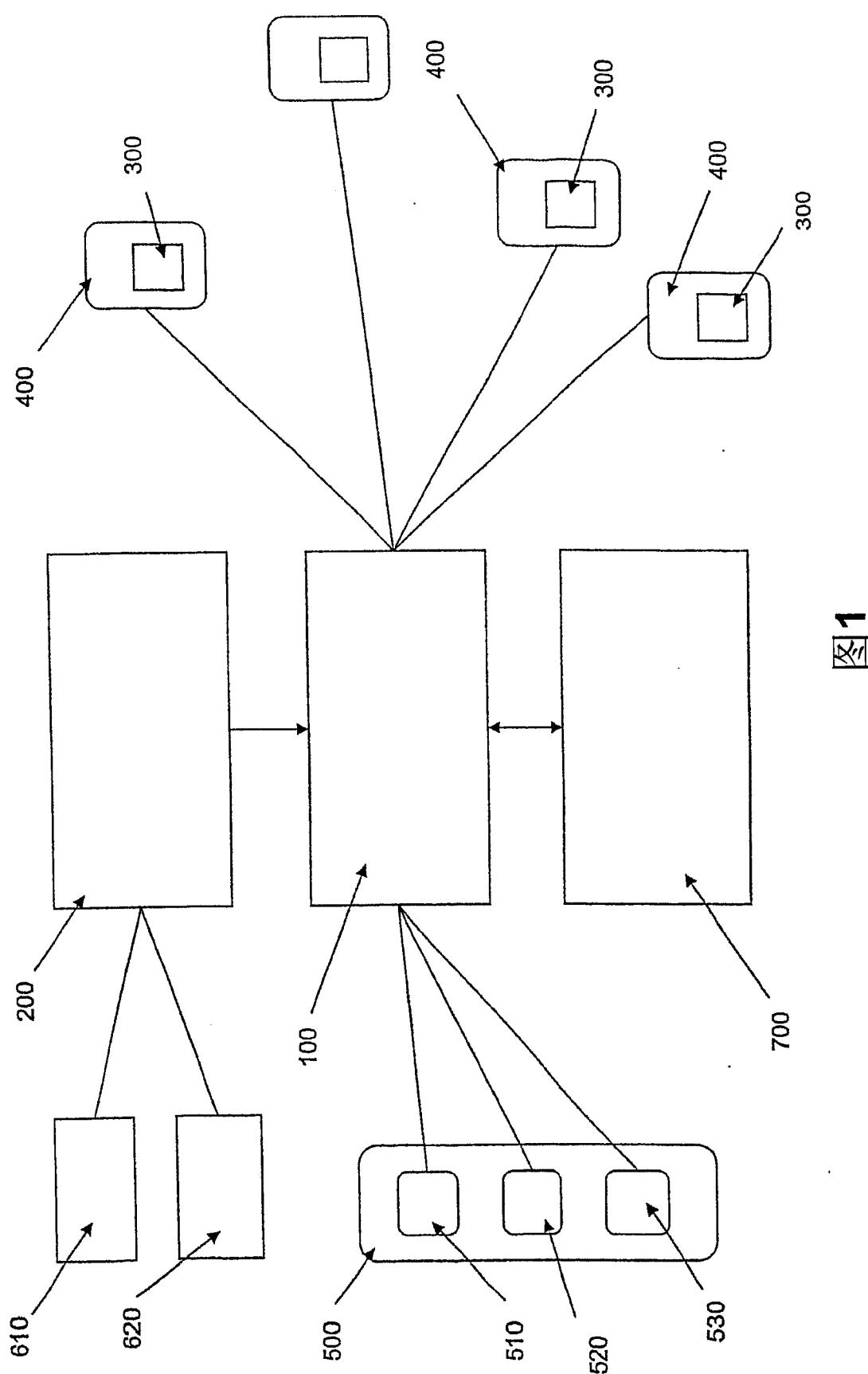


图1

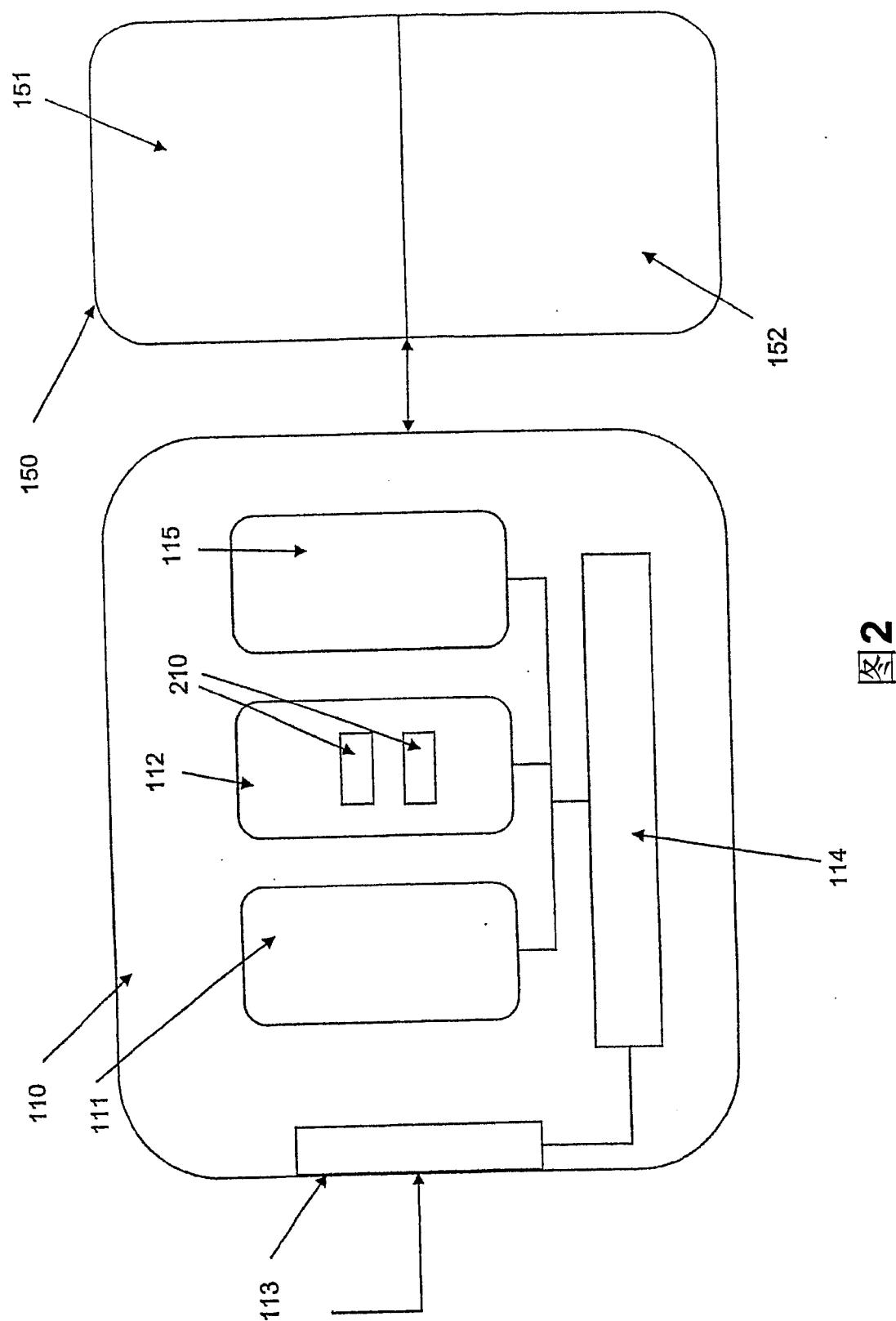


图2

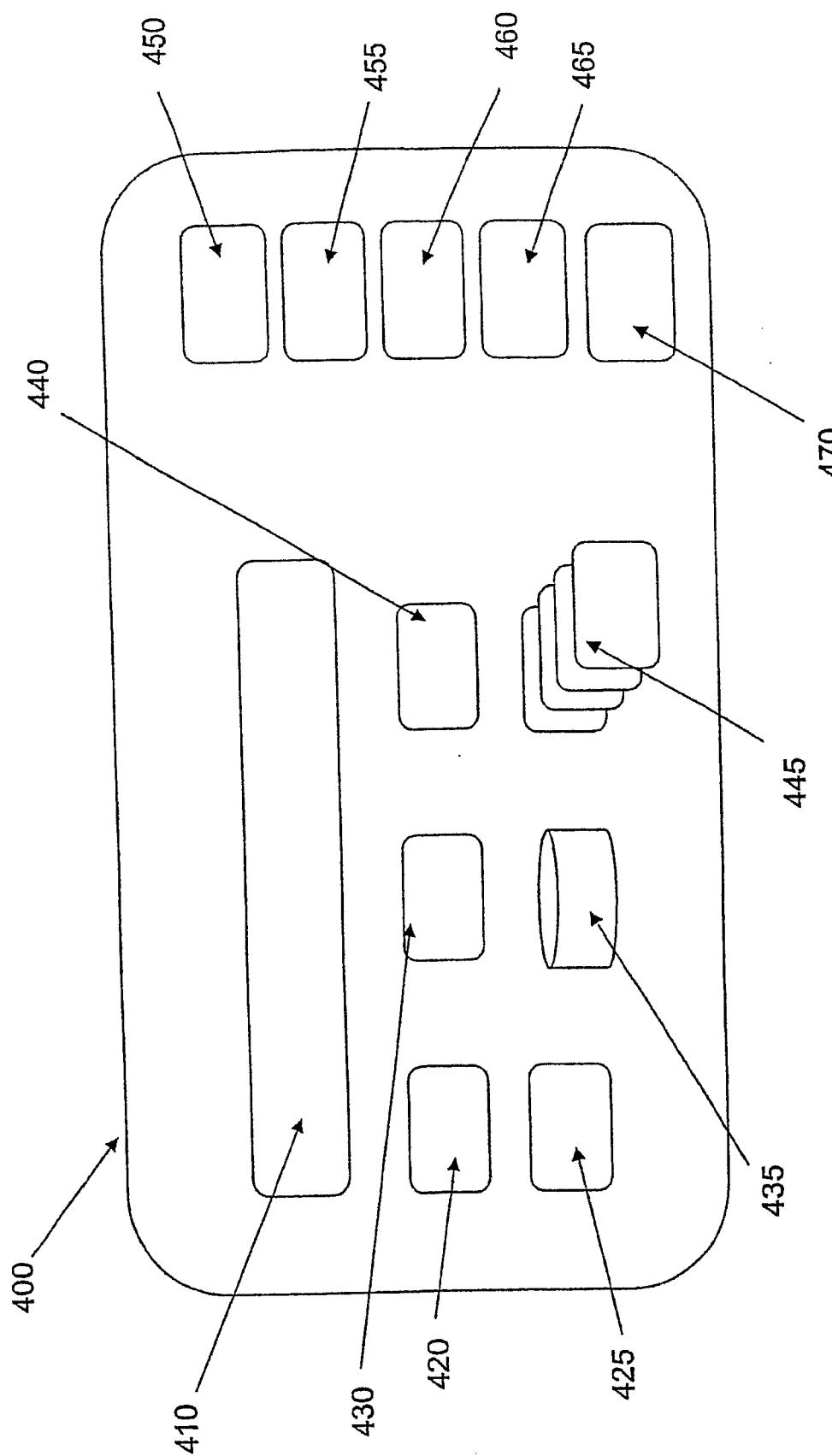


图3

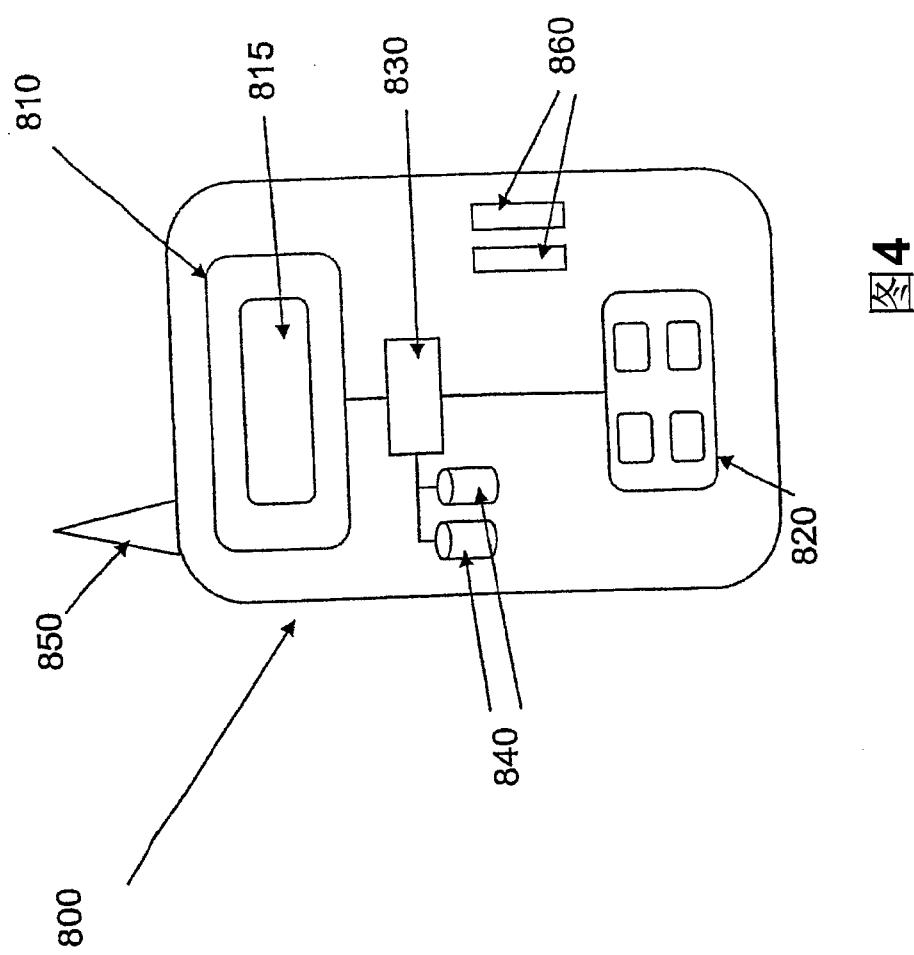


图4