

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 9/445 (2006.01)

G06F 9/44 (2006.01)



[12] 发明专利说明书

专利号 ZL 200580005444.3

[45] 授权公告日 2009 年 2 月 18 日

[11] 授权公告号 CN 100462917C

[22] 申请日 2005.2.18

[21] 申请号 200580005444.3

[30] 优先权

[32] 2004. 2. 19 [33] GB [31] 0403709. 9

[86] 国际申请 PCT/GB2005/000610 2005.2.18

[87] 国际公布 WO2005/081101 英 2005.9.1

[85] 进入国家阶段日期 2006.8.18

[73] 专利权人 高通剑桥有限公司

地址 英国剑桥

[72] 发明人 M · L · 图梅尔 M · 狄更斯

[56] 参考文献

US20030120667A1 2003.6.26

审查员 刘邵频

[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

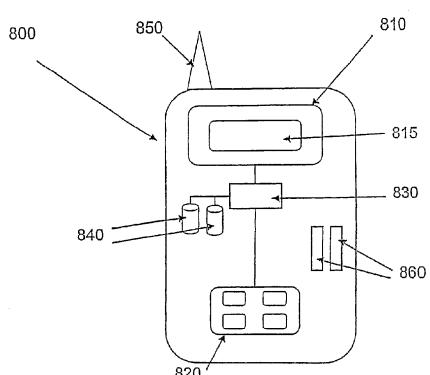
权利要求书 3 页 说明书 15 页 附图 5 页

[54] 发明名称

用户界面内容数据的数据容器

[57] 摘要

提供了一种数据容器，用于为诸如移动电话的设备的用户界面提供内容数据。该容器包括内容数据和与该内容数据有关的元数据，元数据限制对内容数据的访问并且/或者为用户界面内的内容数据的使用提供竞争。



1. 一种向移动设备提供用户界面的方法，所述方法包括以下步骤：

- a) 创建容器，所述容器包括：用于用户界面的可执行代码；所述用户界面中使用的一个或多个内容资源；以及元数据，所述元数据包括关于所述内容资源或各个内容资源的文件和指令，所述可执行代码、所述内容资源或各个内容资源和所述元数据在所述容器内被存储为序列化的对象；
- b) 将所述容器传输给一个或多个移动设备；
- c) 在所述移动设备或各个移动设备处提取所述容器的内容；以及
- d) 执行所述代码以为所述移动设备生成用户界面。

2. 如权利要求 1 所述的方法，其中，所述元数据还包括在步骤(a)期间确定对所述可执行代码和/或所述内容资源或各个内容资源的访问以防止对所述可执行代码和/或所述内容资源或各个内容资源的未授权访问的数据。

3. 如权利要求 1 或 2 所述的方法，其中，如果在步骤 a) 期间，所述可执行代码和/或内容资源被变更，则所述元数据被相应地更新。

4. 如权利要求 1 或 2 所述的方法，其中，关于所述内容资源或各个内容资源的元数据与一个或多个层次分类有关，所述层次分类与移动设备的性能有关。

5. 如权利要求 1 或 2 所述的方法，进一步包括以下步骤：

e) 将所述容器内容处理成用于传输给移动设备的格式，步骤 e) 在步骤 a) 之后并在步骤 b) 之前执行。

6. 一种用于向一个或多个移动设备提供用户界面的服务器，所述服务器包括：

接收数据容器的存储装置；

在使用所述数据容器时使所述数据容器能够被编辑的编辑装置，

其中，所述数据容器包括用于用户界面的可执行代码；在所述用户界面中使用的一个或多个内容资源；以及元数据，所述元数据包括关于所述内容资源或各个内容资源的文件和指令，所述可执行代码、所述内容资源或各个内容资源以及所述元数据在所述数据容器内被存储为序列化的对象；以及

用于将数据容器传输给一个或多个设备的传输装置。

7. 如权利要求 6 所述的服务器，其中，所述服务器进一步包括处理装置，其被配置成，在使用中，在将数据容器传输给一个或多个移动设备之前处理数据容器。

8. 一种将用户界面安装到设备中的方法，所述方法包括以下步骤：

a) 在移动设备处通过通信网络接收容器，所述容器包括：用于用户界面的可执行代码；在所述用户界面中使用的一个或多个内容资源；以及元数据，所述元数据包括关于所述内容资源或各个内容资源的文件和指令，所述可执行代码、所述内容资源或各个内容资源以及所述元数据在所述容器内被存储为序列化的对象；

b) 在所述移动设备处提取所述容器的内容；以及

c) 执行所述代码以为所述设备生成用户界面。

9. 如权利要求 8 所述的方法，其中，所述元数据包括在步骤 (b) 期间确定对所述可执行代码和/或所述内容资源或各个内容资源的访问以控制对所述可执行代码和/或所述内容资源或各个内容资源的访问的数据。

10. 如权利要求 9 所述的方法，其中，可响应接收来自所述通信网络的控制消息而更新所述访问确定元数据。

11. 一种移动设备，包括显示器、用户界面、存储装置、处理装置和通信接口，所述移动设备被配置成，在使用中：

经由所述通信接口从通信网络接收数据容器；

将所述数据容器存储在所述存储装置中；
使用所述处理装置处理所述数据容器，以提取所述数据容器的内容，所述数据容器包括：用于用户界面的可执行代码；在所述用户界面中使用的一个或多个内容资源；以及元数据，所述元数据包括关于所述内容资源或各个内容资源的文件和指令，所述可执行代码、所述内容资源或各个内容资源以及所述元数据在所述数据容器内被存储为序列化的对象；

根据所述提取出的所述数据容器的内容来形成用户界面；以及在所述设备显示器上显示所述用户界面。

12. 如权利要求 11 所述的移动设备，其中，存储在所述存储装置中的所述元数据包括确定对所述可执行代码和/或所述内容资源或各个内容资源的访问以控制对所述可执行代码和/或所述内容资源或各个内容资源的访问的数据。

13. 如权利要求 13 所述的移动设备，其中，所述设备进一步被配置成，在使用中，经由所述通信接口从所述通信网络接收控制命令，所述控制命令更新确定对所述代码和/或内容资源的访问的所述元数据。

用户界面内容数据的数据容器

技术领域

本发明涉及数据容器，并特别涉及在移动通信网络中使用的移动设备的用户界面内容数据使用的数据容器。

背景技术

移动网络运营商和内容提供商的增长领域之一是供应移动电话和设备的铃声、壁纸和其它多媒体内容。在移动网络运营商和设备制造商为了品牌塑造目的而保持对设备用户界面的一些方面的控制的需求，与用户定制和修改他们的设备的外观以适合其自身需要的需求之间存在着矛盾。在提供所需的灵活度和用户化需要的复杂软件，与典型移动设备有限的处理功率和数据存储能力之间也存在着矛盾。本发明设法缓解这些问题。

发明内容

根据本发明的第一方面，提供了一种向设备提供用户界面的方法，该方法包括以下步骤：a) 创建容器，该容器包括：用于用户界面的可执行代码；用户界面中使用的一个或多个内容资源；以及关于该内容资源或各个内容资源的元数据，可执行代码、该内容资源或各个内容资源和元数据在容器内被存储为序列化的对象；b) 将容器传输给一个或多个设备；c) 在该设备或各个设备上提取该容器的内容；以及 d) 执行代码以为该设备生成用户界面。

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

根据本发明的第三方面，提供了一种用于向一个或多个设备提供用户界面的服务器，该服务器包括：接收数据容器的存储装置；在使用数据容器时使数据容器能够被编辑的编辑装置，其中，该数据容器包括用于用户界面的可执行代码；在用户界面中使用的一个或多个内

容资源；以及与该内容资源或各个内容资源有关的元数据，可执行代码、该内容资源或各个内容资源以及元数据在数据容器内被存储为序列化的对象；以及用于将数据容器传输给一个或多个设备的传输装置。

根据本发明的第四方面，提供了一种将用户界面安装到设备中的方法，该方法包括以下步骤：a) 在设备处通过通信网络接收容器，该容器包括：用于用户界面的可执行代码、在用户界面中使用的一个或多个内容资源，以及与该内容资源或各个内容资源有关的元数据，可执行代码、该内容资源或各个内容资源以及元数据在容器内被存储为序列化的对象；b) 在设备处提取容器的内容；以及 c) 执行代码以为设备生成用户界面。

根据本发明的第五方面，提供了一种包括用于执行上述方法的计算机可执行代码的数据载体。

根据本发明的第六方面，提供了一种设备，包括显示器、用户界面、存储装置、处理装置和通信接口，该设备被配置成，在使用中：经由通信接口从通信网络接收数据容器；将数据容器存储在存储装置中；使用处理装置处理数据容器，以提取数据容器的内容，数据容器包括：用于用户界面的可执行代码、在用户界面中使用的一个或多个内容资源，以及与该内容资源或各个内容资源有关的元数据，可执行代码、该内容资源或各个内容资源以及元数据在数据容器内被存储为序列化的对象；根据提取出的数据容器的内容来形成用户界面；以及在设备显示器上显示用户界面。

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 示出了用于移动设备的软件 400 的示意图；

图 4 示出了内容工具箱的示意图；和

图 5 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

现将仅通过举例的方式并参照附图来对本发明进行说明，在附图中，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（content feed）500 和用户界面（UI）源 600。在使用中，服务器 100 将内容数据和 UI 数据传送给移动设备 300、301、…，每个移动设备包括软件包 400。服务器 100 通过接口与 OSS 700 连接，其中 OSS 是传统上被用于操作移动网络的那些 OSS，例如帐单、账目管理等。服务器 100 还通过接口与内容工具箱 200 连接：内容工具箱从 UI 源 600、601、…接收数据，并封装 UI 数据以便服务器能将经封装的 UI 数据发送给包含在移动设备 300 内的软件包 400。服务器从多个内容供给接收数据，并处理和封装该数据以便能将其发送给软件包 400 或者以便移动设备 300 能使用软件包 400 来访问数据。

可设想将该系统划分为三个分离的域：运营商域 50 包括由移动网络运营商（MNO）操作的系统和装备；用户域 60 包括多个移动设备；以及第三方域 70 包括可由许多不同实体控制或操作的内容供给和 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。发布组件包括数据库 111、输入队列 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处接收来自内容工具箱的内容。该内容是以包括一个或多个 Trig 和一个或多个 Triglet 的包（参加下文）210a、210b、…的形式呈现的。trig 是用于诸如移动电话的移动设备的用户界面，triglet 是可用于扩展或变更 trig 的数据文件。如果一个包包括多个 trig，则其中一个 trig 可以是从中导出其它 trig 的主要 trig。

发布组件用户界面 114 可用于将包输入到数据库 111 中，该处理使对每个 trig 和 triglet 的引用（reference）被加载到输入队列 114 中，该输入队列可包括对多个包 210a、210b、…的引用。可使用用户界面来检查包的内容，并且包的内容可被传递给目录。

MNO 可具有若干发布域，例如，每个发布域用于许多国家或地区中的每个目标服务器。每个域与其它域隔离开来对待，并且具有自身

的描述如何在实时（live）和登台（staging）环境中将对象发布到内容服务器上的发布方案。发布组件 GUI 向每个域提供若干不同的视图，从而使运营商能完全管理内容的发布。目录包括对存储在该目录中的对 trig 以及用于将内容传输到不同域的更新信道和供给信道的引用。对于每个域，运营商使用发布组件 GUI 来设置域结构，并将来自目录的 trig 分配到每个域节点。为了帮助运营商有效率地选择 trig，在目录中设置过滤器以便仅显示相关的项目。

可将 trig 分配到域内的若干节点。在每种情况下，根据将要访问 trig 的手机，目标服务器上对 trig 的封装可能需要是不同的，例如 SIS 或 CAB 文件。该封装可使用发布组件 GUI 来控制。

更新信道可由 trig 引用以控制内容的传递。更新信道包括这样的 URL，它是到包括 triglet 更新包的相关域上的资源的链接。该 URL 可以预先定义的间隔由用于访问该包的 HTTP GET 函数轮询（很容易理解的是，可使用其它传输方案，例如 SyncML 或 SMS 或用于小更新的小区广播）。Triglet 更新包描述怎样能修改 trig，例如，替换一个或多个由 trig 使用的图像或文本文件。发布组件 GUI 使运营商能够定义和控制：为某个域而存在的更新信道，与更新信道上的每个 triglet 相关联的 URL，以及 triglet 与域的更新信道的关联性。由于每个 triglet 与更新信道相关联，所以运营商可输入应该发布更新的日期和时间，使得能够设置进度表。

内容供给类似于内容更新定期为其自动生成的更新信道。内容供给通过轮询 URL、检索更新分组和将它应用于 trig 而被访问。然而，由于手动构造的 triglet 更新和自动生成内容的不同性质，将更新信道和内容供给被分开管理。此外，可使用其它传输方案，诸如 SyncML 或 OMA-DM（开放移动联盟设备管理）。

内容服务器组件 150 是网络服务器的标准实现方案，因而缩比模型很好理解。服务器的性能可使用指示网络服务器在基准条件下可处理的并行会话数的“SPECweb99”数来评估。所发布的 SPECweb99 数在从 404 到 21,000 范围内，典型的商用网络服务器的 SPECweb99 数为 5,000。1M 用户每小时更新内容的典型使用情况，需要 SPECweb99 额定值仅为 1,112 的网络服务器。成功的使用将导致服务使用增加，该

服务使用增加可通过使另外的服务器能够建立既可扩展又能对故障具有高度弹性的基础结构来提供。

可建立经由 WAP 网关的从移动设备到服务器的连接。在这种情况下，网络服务器会话存在于 WAP 网关和网络服务器之间，而不是移动电话和网络服务器之间。当经由 WAP 网关做出对文件的请求时，与网络服务器的会话仅维持将文件从网络服务器传输到 WAP 网关所需的时间，即，由于连接带宽将非常高并且等待时间极短，所以该会话极短。

可替代地，可在移动电话和网络服务器之间建立直接连接。在这种情况下，网络服务器需要在将数据下载到电话所需的时间把会话保持开通状态。

有两种类型的由内容服务器组件传递的内容：trig，典型地约为 100KB，并定期更新 triglet，triglet 典型地约为 1KB。trig 下载产生的业务量与现有内容下载产生的业务量非常相似。因此，有关的问题很好理解。定期的 triglet 更新的下载是 MNO 的业务量模型中的新特征，但是由于更新的尺寸小（典型地，该更新可装入一个数据分组内），所以有可能显示出业务量仍能够由典型的网络服务器处理。

图 3 示出了用于移动设备 300 的软件 400 的示意图，该软件 400 包括标记语言绘制器 410、更新管理器 420、网络通信代理 425、资源管理器 430、虚拟文件系统 435、执行者管理器 440、多个执行者 445a、445、…、本地 UI 绘制器 450、支持管理器 460、trig 管理器 465 和标记语言分析器 470。

优选地，软件使用 TrigML 来运行，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，以及控制手机上的内容的替换和视图。本地 UI 绘制器用于显示可不使用 TrigML 来显示的 UI 组件，并用于显示错误消息。

软件 400 以设备特定方式来提供和安装。例如，对于 Nokia 系列 60 设备，软件使用 SIS 文件来安装，而对于 MS 智能手机设备，软件使用 CAB 文件来安装。类似地，软件升级也以设备特定方式来处理。软件可以更受限制的格式来提供，如仅绘制其内置内容的自含式应用

程序那样：即，该软件提供有内置 trig，但是随后不能添加另外的 trig。所提供的 trig 可无线升级。

trig 管理器 465 向资源管理器 430 和标记语言绘制器提供界面。它通常负责 trig 管理。这包括：持续获知使用中的 trig，变更当前的 trig，选择启动时的 trig，为损坏的 trig 选择另一个 trig 作为备用（fall back），维持所安装的 trig 的集合，识别特定的 trig 安装到资源管理器中的位置，以及读取 trig 的更新信道定义和适当地配置更新管理器。

资源管理器提供设备上的永久存储的抽象概念，即，将文件存储为真实文件，或存储为数据库中的记录。资源管理器向标记语言绘制器和更新管理器提供文件系统界面。它负责处理文件路径逻辑，区分真实资源文件和执行者属性，将与 trig 有关的路径映射到绝对路径上，通过接口与 trig 管理器连接，以及向更新管理器提供修改界面。

更新管理器处理 Trig 和 Triglet 的接受和应用。更新管理器向绘制器和 trig 管理器提供界面，并负责：当受到绘制器指示时，启动手动更新；当由 trig 管理器如此配置时，控制和实现自动更新信道；指示手动更新的进度，以及在网络连接和/或设备电源意外中断之后恢复更新。可将更新分组的格式定义为 XML 模式的二进制序列化。

XML 是用于定义更新分组格式以及 TrigML 内容的便利的数据格式化语言。出于带宽和存储效率的原因，将文本 XML 序列化为二进制表示。更新分组和 TrigML 分段二者是由相同的组件，MARK-UP LANGUAGE PARSER（标记语言分析器）分析器，来进行分析的。在软件中对 XML 的任何进一步的使用必须使用二进制 XML 编码，因此必须重新使用分析器。

执行者管理器 440 监督软件中出现的执行者 445 的集合。它的使用情况是：在内容向执行者发送事件时，由绘制器使用；被想要通知属性值已经变更的执行者使用；和被想要发射事件的执行者使用（参加下文）。

优选地，软件包括最少运行两个线程的多线程应用程序，取决于包括多少和什么样的执行者的可能性更大。软件主要在称为主线程的线程中运行。主线程被用于运行与其它组件同步通信的绘制器。执行者始终具有对绘制器的同步界面。如果执行者由于其功能需要额外线

程，则执行者负责管理线程间的通信。优选地，使用光消息接发框架（light messaging framework）来避免在许多执行者需要线程间通信的情况下不必要的代码重复。

除主线程之外，更新管理器运行网络线程。该网络线程用于下载更新分组，并与主线程分离以允许绘制器在分组到达之前继续不受影响。更新管理器负责处理线程间消息接发，以便更新管理器在应用更新分组中定义的变更时与绘制器和资源管理器同步通信。

软件 400 以设备特定方法被提供给移动设备。可提供一个或多个 Trig，作为安装的一部分，例如，可将其存储为未压缩的更新分组。在启动时，可扩展该分组并将其安装到文件系统中。

执行者 445 是发布属性值以及处理和发射事件的组件。执行者与绘制器同步通信。如果执行者需要异步行为，则该执行者需要负责管理和与绘制器的主线程外部的线程通信。

执行者属性可作为文件引用被读出。属性是下列四种类型之一：单独的简单值；简单值的矢量；单独的字段结构，每个字段具有简单值；或者结构矢量。可使用类似于许多面向对象的编程语言的 object.member 符号的表达式来引用属性：

```
<image res="signallevels/{protocol.signalstrength}" />
```

当需要作为文件时，属性经由 attrs 文件夹而被访问。

```
<text res="/attr/network/name">
```

可通过用<throw>元素向执行者发送事件来通知该执行者。可将执行者发射的事件传递给内容树作为内容事件：这些可被送往元素 Id 或 'top'（顶端）处。对执行者的界面由执行者界面定义文件定义。这是定义属性、类型、字段名、输入事件和参数以及输出事件的 XML 文档。执行者的集合在软件的构建时刻是可配置的。附录 A 给出了可使用的一些执行者以及相关联的函数或变量的示例列表。

更新包括新 trig（新的或替换 UI）或 triglet（对现有 trig 的修改），并可将其看作是对软件文件系统的修改。更新管理器通过读取分组来确定文件系统中什么需要改变。更新分组可由软件 400 使用 HTTP 或其它适合的传输机构无线地下载，以设备特定包的格式来打包，或者可通过安装软件本身来预先提供。

存在着需要考虑的其它故障模式：如果不能启动 HTTP-GET，或者遇到 HTTP 错误响应代码，则取消该次更新尝试，并使用重试策略在稍后的时间开始新的更新尝试。在 HTTP 响应由于网络信号的丢失而中断的情况下，删除任何临时文件，并使用重试策略以在稍后的时候重新开始更新尝试。如果更新头指示更新有效负荷大小可能太大而不能安装在设备上，如果更新需要软件的不兼容版本，或者如果更新已经在设备上，则删除头数据文件并取消更新尝试和任何随后的重试。

内容格式在实现该软件的所有平台上是共同的。内容编译器是将收集的裸资源（文本 TrigML、PNG 图像、文本串定义）转换为可写入设备的文件系统的无线电更新分组的内容创作工具。

TrigML 分段是包含文本 TrigML 的文件，在这些分段内的资源引用是虚拟文件路径。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 trig 的其它特性。当被用于编译 triglet 时，该文件还定义怎样将输入的 TrigML/PNG/Text 资源映射到 trig 的虚拟文件系统的修改上。

对于 PNG 和 Text 资源，trig 定义文件指向主机文件系统上的真实文件列表，并且该资源被复制到输出。

TrigML 可使用常变量来代替属性值。常变量是用与<include>参数相同的语法来访问的，例如，\$background_colour。常量在 trig 中被作为全局变量对待，并在保留的文件夹 constants/ 中定义。包含在 constants/ 文件夹内的文件中的变量定义可在编译时通过直接替换它们的值来解析。在一个可选实施例中，constants/ 中的变量定义被作为全局变量来编译，并在内容分析时由软件解析。这允许 trig 通过简单替换它的 constants 文件中的一个或全部而被更新。

为了成功地绘制移动设备的用户界面，标记语言必须具有下列性质：简明的页定义，一致的布局规则，可在小型绘制器中实现，提供多分层和任意重叠的内容、事件模型，仅要求重新绘制 UI 各页之间必须改变的显示器区域，包括向用于读取接收事件和发送事件的特性值的平台的 hooks，可扩展，并且在图形上有灵活性。TrigML 提供这些特征，并且我们的提交于 2004 年 2 月 19 日的，第 GB0403709.9 号同时待审的申请，给出了提供所需功能的元素和属性的综述。

希望使重新标记 (re-branding) UI 和产生连续更新流的成本最小。这是通过提供从创建处理直到向用户传输数据的有效的信息流来实现的。

被称为包的容器，被用于 UI、UI 更新和用于包括第三方的模板。这些包包含由第三方产生、测试和传递标记的 (branded) UI 和更新所必需的所有信息。

图 4 示出了内容工具箱 200 的示意图，内容工具箱 200 包括脚本环境 220、测试和仿真环境 230 和维护环境 240。

包处理包括五个处理阶段：

- 1) 脚本环境 220 提供用来设计一个或多个 UI 的模板和基于该模板更新 UI 策略的手段。
- 2) 维护环境 240 在能够被内容提供者从外部获得的具有良好控制和引导的环境中提供快速的 UI 和更新产品。
- 3) 维护环境 240 'pre-flight (预检)' 功能允许部署管理员检查和调整它们从第三方接收到的 UI 和更新。
- 4) 发布组件 110 提供对 UI 管理，并在部署点处更新，包括新版本的升级 (staging)。
- 5) 发布组件 110 使得来自即时内容供给的更新能够自动生成。

在一个典型项目中，在脚本环境 220 内创建包以用于：使内容提供商从模板创建重新标记的 UI，结合相同的'感觉'但是不同的'外观'；使内容提供商从模板创建更新，该更新向 UI 内容提供周期性的或者用户选择的变更；或者使广告代理商在周期基础上从模板创建促进新服务的更新。

对于所有这些用途情况，维护环境 240 被用于输入包、重新标记和重新配置内容，以及创建提交给发布组件 110 的新的包。在 UI 模板的设计中，应考虑下列问题：UI 的哪部分能被重新标记；UI 的哪些特征需要在重新标记时重新配置或在远程重新配置；UI 内容的哪部分可以被更新；以及，如果对 UI 进行重新标记，则用户是否能选择使用中的内容供给。脚本环境 220 允许定义这些策略，并使维护环境 240 能作为每个策略的每个实例的实施者。

脚本环境 220 的主要功能可包括：

- 定义菜单结构和页映射。
- 定义标记内容被放入其中的框架。
- 定义 UI 中可更新的部分。
- 定义更新中重新标记时可替换的部分。
- 提供交互式预览以辅助编辑者。
- 提供每个 UI 层的图形代码视图。
- 允许将资源拖放到交互式预览和代码视图中。
- 输出用于特定重新标记或更新构建任务的模板。
- 在手机仿真器上仿真 UI 和更新。
- 构建用于在真实设备上测试的 UI 和更新。
- 提供扩展的调试工具以辅助开发。

此外，维护环境 240 的目的是提供设计者和管理员的 UI，用于外壳和更新的重新标记和维护，主要功能包括：

- 重新标记 UI 模板
- 用新内容提供更新
- 通过更新来管理 UI 菜单条目
- 将 UI 和更新翻译成另外的语言
- 用于另外的设备的目的字符串和内容
- 在手机仿真器上仿真所使用的 UI
- 构建用于在真实设备上测试重新标记的 UI 的 UI 和更新

包由包括模板 UI 或更新的脚本环境 220 生成以进行编辑。完成编辑之后，将包保存在准备好发送到维护环境 240 的'outbox (待发箱)'中，以发布给内容服务器。提供了下面的 '包' 功能。维护环境 240 可用于编辑/替换包内容纳的资源。可将包输出到仿真环境以测试移动设备上的 UI 或 UI 更新的性能。

浏览器被提供给用户，用以访问这些类别，同时用户能够改变：被标为可视的任何 UI 或更新，或者 UI 或更新内被标为 '可替换' 的资源。当被保存时，'可视的' 对象的缩略图被保存在包中，以用于维护环境中的识别用途和用于其它服务。

可双击包条目以启动适当的编辑器。(例如，图像资源将启动图像编辑器)。所有资源可具有插入维护环境中、并显示在维护环境中的适

当的上下文中的文本描述/注释。将菜单条目的列表作为特殊资源类型来处理，每个条目代表其自身资源（例如——标题、帮助字符串、图像、滚动图像、URL 和铃声预览）的子目录。

许多不同的 UI 可从共同的基础导出。典型地，该共同的基础将实现其自身界面的大部分，并且从其中导出的各 Trig 将在其上实现小变更，诸如标记。Triglet 可从 Trig 中导出，并且它可覆盖它所选择的来自父 Trig 的任何资源（可选地，它可引入它自身的资源）。应注意，此处的“资源”也指 TrigML，所以 Trig 的行为和布局可由 Triglet 修改，就像它替换单个图像或单条文本那样简单。

包可包括一个或多个基 Trig(即，不是从任何其它 trig 导出的 Trig)，一个或多个从基 Trig 导出的 Trig，多个从任何 trig 导出的 triglet，以及多个从其它 triglet 导出的 triglet。

包格式是将所有该信息存储为序列化对象的不透明的二进制格式。包可包括许多资源，诸如图像、文本、URL、更新信道、铃声文件、壁纸、本地应用程序等。每项资源包含关于如何察看、编辑或删除资源的许可信息。每项资源还包含元信息，诸如与资源有关的文件和指令。每个包工具或者承担有关的角色，或者要求用户注册为特定角色。

trig 开发具有这样的性质：许多人和/或许多组人能够参与到对 Trig 的最终设计和实现的贡献中。此外，这些专业人员的集合需要使用非常简化和受控的方案来使对 trig 造成无意损坏的风险达到最小。用于相当复杂的 Trig 的典型开发工作流程可包括：

- UI 设计者创建初始 UI 结构。该设计可使用维护环境来创建该 UI 的第一个版本来构建。

- 图形设计者创建最终图形，并将它们添加到设计中。

- 在初始设计中识别出的专门用于动态内容的 UI 的区域需要进行充实。

- 用于动态更新的图形需要由图形设计者最终确定下来。

- 然后设计和实现 UI 的个人化区域。这可由许多第三方内容提供商来处理。

包在上述工作流程中起辅助作用，因为它们包含单个文件中的整

个项目（project），这使得它很容易从队伍中的一个成员被传递给另一个。可通过添加关于需要修改的资源的评论和指示，甚至通过设置其它资源的可编辑性以限制可改变的内容，来为开发的下一阶段重新确定作为目标的包。通过允许包被派生，以及允许在包的每个派生中进行分别开发，可支持更复杂的工作流程。合并工具允许将个别改变组合回单个包中。包可使用 Python 编程语言的 pickle 模块来实现。

包可用于开发用于具有不同性能（诸如显示尺寸、RAM 容量）的移动设备的 trig 和/或 triglet。为了将其简化，可定义许多分层，并在这些分层内将数据资源或 TrigML 元素分类。当从包编译 trig 或 triglet 时，可为特定设备选择和编译最适当的资源或 TrigML 元素。

图 5 示出了包括根据本发明的一个实施例的用户界面的设备 800 的示意图。该设备包括显示用户界面 815 的显示器 810，以及使用户能与用户界面 815 交互的用户界面装置 820。处理器 830 执行存储在一个或多个存储装置 840 内的软件，并且可设置有一个或多个无线通信接口 850，以使得能够与其它设备和/或通信网络通信。可容纳一个或多个电池 860 来为设备供电，该设备还可包括接口来接收电功率和/或通信电缆。

这些组件和界面的性质将取决于设备的性质。可以理解的是，这种用户界面可在移动或蜂窝电话手机内实现，但是也可将其应用到其它便携式设备（诸如数码相机、个人数字助理、数字音乐播放器、GPS 导航设备、便携式游戏机等）中。此外，还可将其应用到包括用户界面的其它设备，诸如，膝上型或台式计算机。

用户界面装置可包括多个按钮，诸如数字或字母数字混合键盘，或者触摸屏或类似的。一个或多个存储设备可包括某种形式的非易失性存储器，诸如存储器卡，以便所存储的数据在断电的情况下不会丢失。可提供 ROM 存储装置以存储不需要更新或改变的数据。可为暂时存储器提供一些 RAM，因为更快的响应时间支持频繁访问数据的高速缓存。该设备还可容纳用户可移动存储器卡，并且可选地，可将硬盘驱动器用作存储装置。所使用的存储装置将通过权衡设备尺寸、功率消耗、所需要的存储容量等方面的不同需求来确定。

这种设备可实际上结合任何无线通信网络，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，以及这些和类似网络的改进和派生来实现。在建筑物和校园内，还可使用其它技术，诸如蓝牙、IrDa或无线LAN（不管是基于无线电还是光学系统）。可提供用于与其它设备进行数据同步和/或电池充电的USB和/或火线连接。用于实现上述方法和/或用于配置上述设备的计算机软件可设置在数据载体上，诸如，软盘、CD-ROM、DVD、非易失性存储器卡等。

本申请要求2004年2月19日提交的英国专利申请第0403709.9号的优先权，其内容结合在本文中作为参考。

附录A

Trigplayer Actor (Trig播放器执行者)	Attributes (属性)	UpdateState (更新状态)	
	Messages (消息)	exit (退出)	
		predial_mode	on/off
	Events (事件)	idle (空闲)	

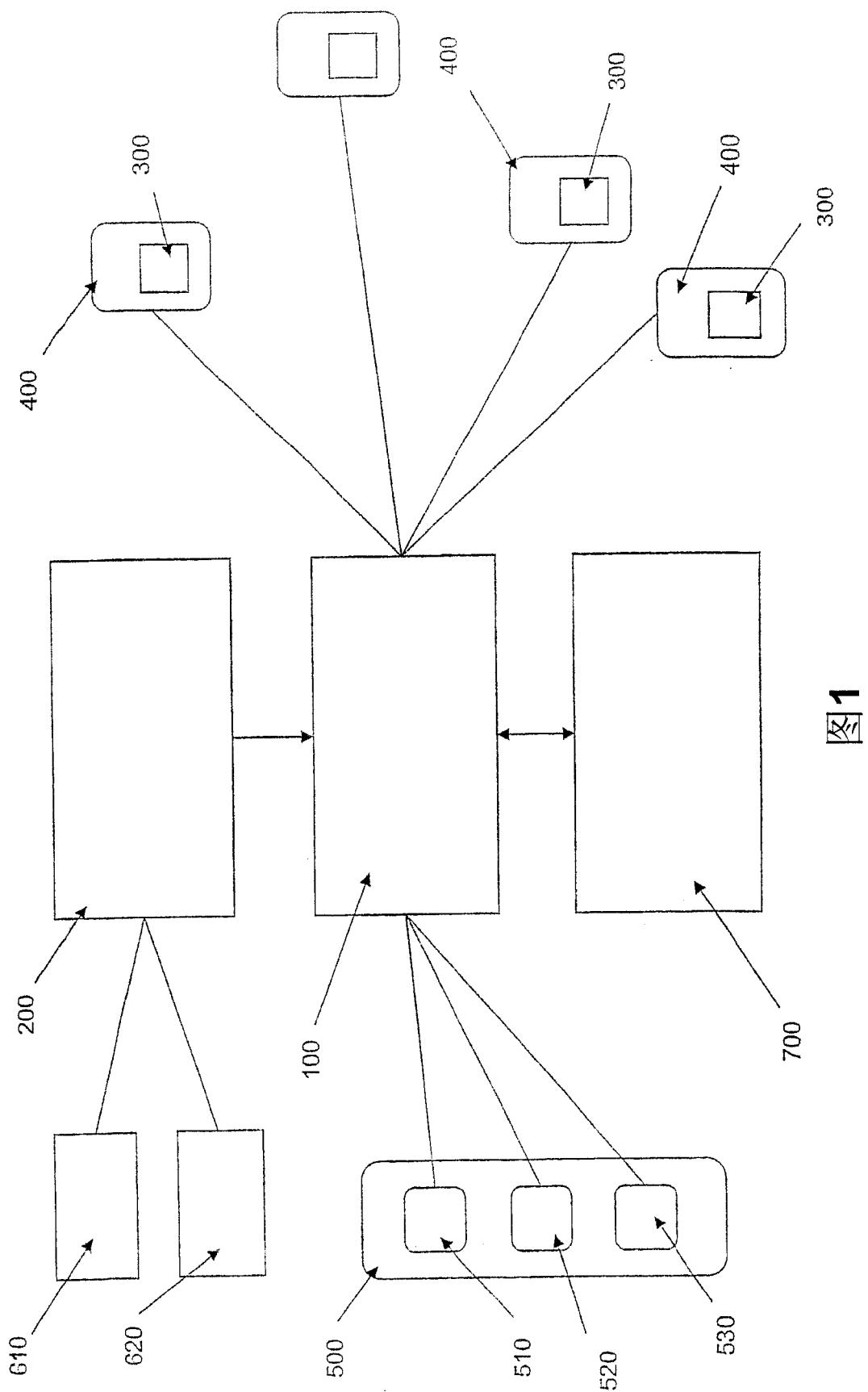
Launch Actor (启动执行者)	Attributes (属性)		
	Messages (消息)	browser (浏览器)	url
		SMS	Number
			Message (消息)
		Camera(照相机)	
		Inbox (收件箱)	
		profiles	
		missed_calls (未接呼叫)	
		Dialer (拨号器)	number (号码)
		...	

		native_app	app_id
			url
Events (事件)			

Install Actor (安装执行者)	Attributes (属性)		
	Messages (消息)	ringtone (铃声)	resource_path
		wallpaper (壁纸)	resource_path
	Events (事件)		

Phone Actor (电话执行者)	Attributes (属性)	Bluetooth (蓝牙)	
		IrDA	
		Call (呼叫)	
		GPRS	
		UnreadSMS (未读 SMS)	
		UnreadVoiceMail (未读语音邮件)	
		UnreadMsgs (未读消息)	
		BatteryLevel (电池电量)	
		SignalStrength (信号强度)	
	Messages (消息)		
Events		missed_call	

	(事件)	(未接呼叫)	
		message_arrived (到 达的消息)	
		voice_mail_arrived (到达的语音邮件)	



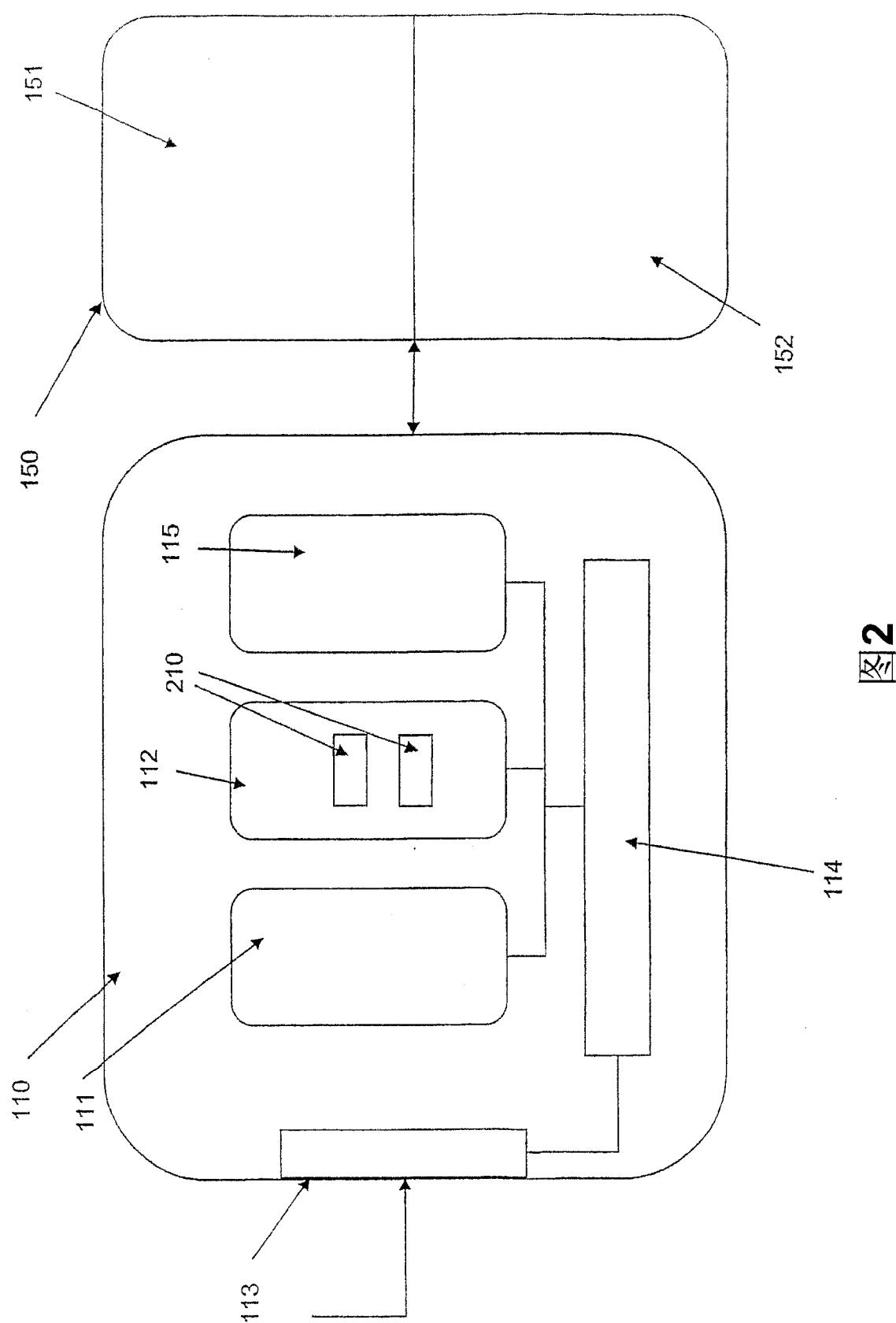


图2

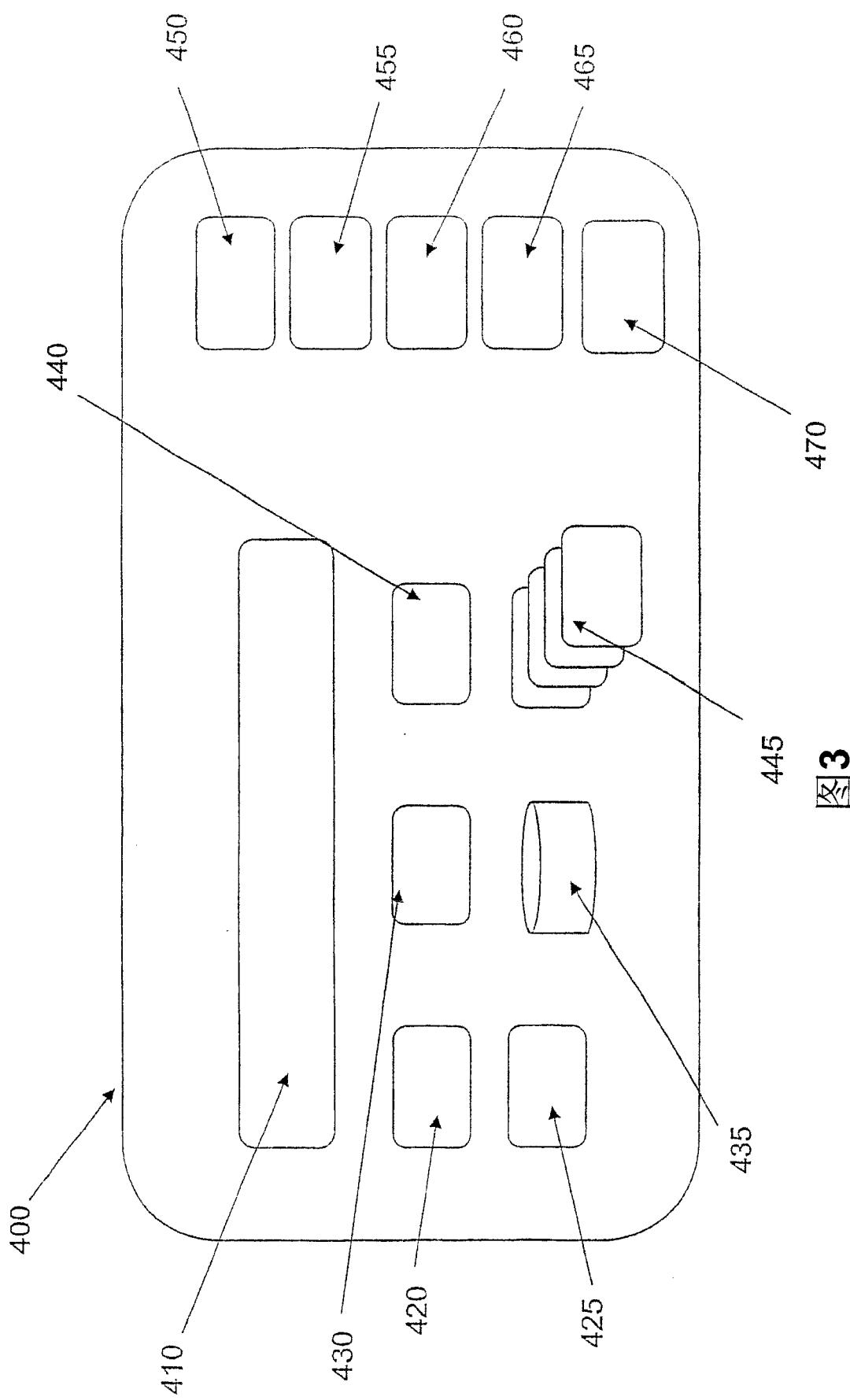


图3

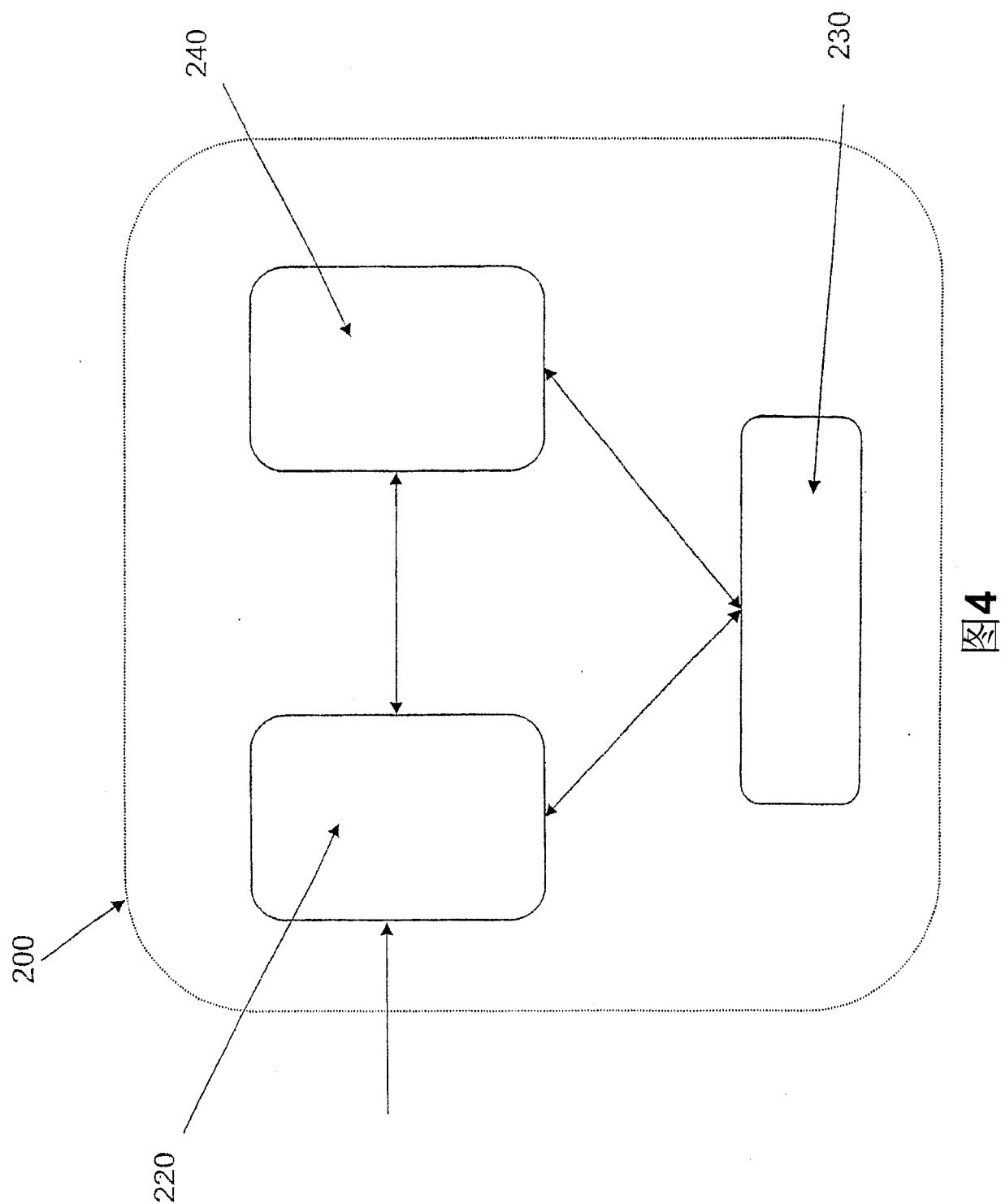


图4

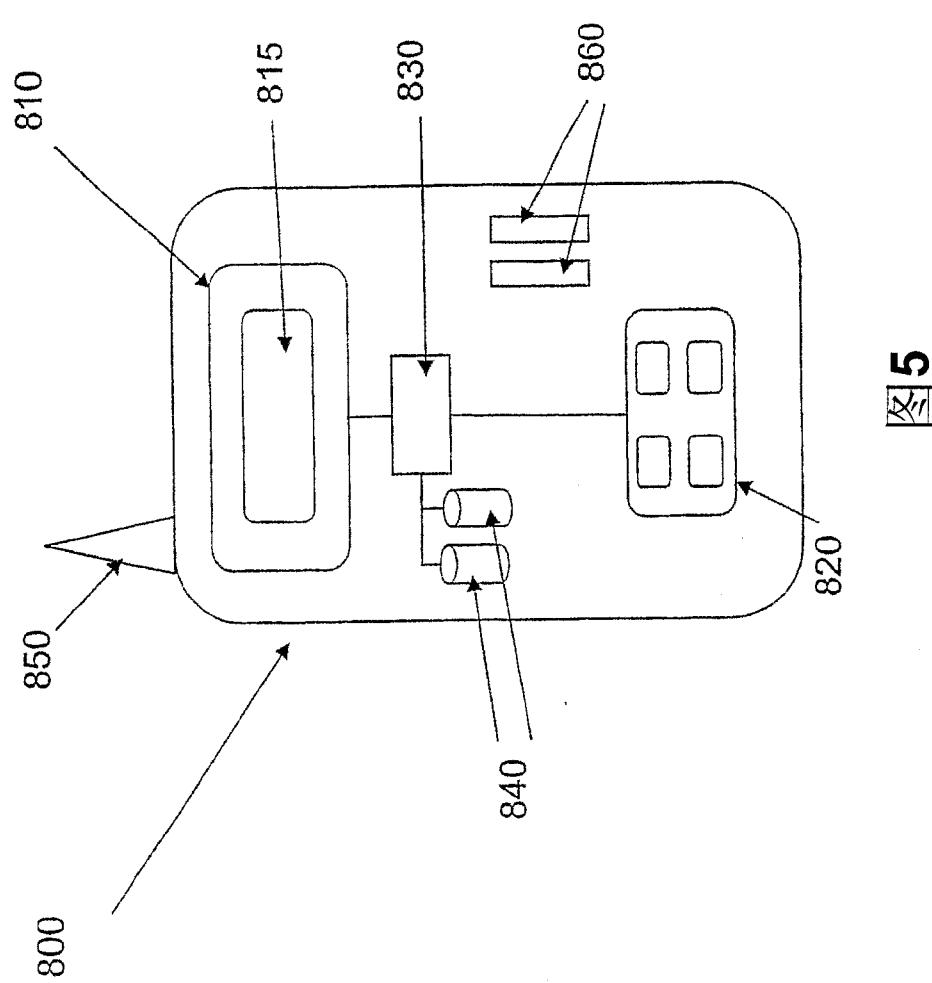


图5