

(12) 发明专利

(10) 授权公告号 CN 1922608 B

(45) 授权公告日 2010. 05. 26

(21) 申请号 200580005392. X

G06F 9/44 (2006. 01)

(22) 申请日 2005. 02. 21

(56) 对比文件

(30) 优先权数据

0403709. 9 2004. 02. 19 GB

US 20020151327 A1, 2002. 10. 17, 说明书第 11, 22, 24, 62-68, 83, 221-227 段、附图 1, 6.

US 20030236917 A1, 2003. 12. 25, 说明书全文.

(85) PCT 申请进入国家阶段日

2006. 08. 18

审查员 李原野

(86) PCT 申请的申请数据

PCT/GB2005/000630 2005. 02. 21

(87) PCT 申请的公布数据

W02005/081132 EN 2005. 09. 01

(73) 专利权人 高通剑桥有限公司

地址 英国剑桥

(72) 发明人 S·G·布特林 N·H·克拉里

J·B·布劳克夫

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 邵亚丽

(51) Int. Cl.

G06F 17/30 (2006. 01)

H04L 29/06 (2006. 01)

H04M 1/725 (2006. 01)

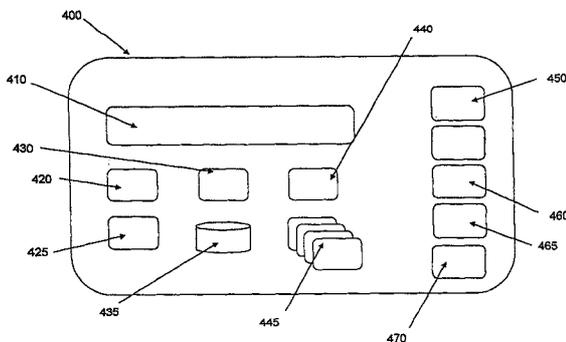
权利要求书 1 页 说明书 13 页 附图 6 页

(54) 发明名称

虚拟文件系统

(57) 摘要

本发明说明了一种既能使真实数据资源（诸如内容文件）又能使用虚拟数据资源（诸如数据库中的字段或者由标记语言元素确定的状态），通过单独的根访问的虚拟文件系统。



1. 一种移动设备,包括:
 - 用于存储多个数据资源的存储装置;
 - 显示器;
 - 用于提供对所述多个数据资源的用户访问的用户界面;
 - 用于对数据资源的位置提供来自所述用户界面的单独分级界面的文件系统;
 - 绘制器,用于在用户界面上显示内容;
 - 其中:
 - i) 所述文件系统包括:
 - a. 包括存储在所述存储装置中的直接寻址数据内容文件的一个或多个位置;
 - b. 包括数据库的一个或多个位置,并且在使用中,通过查询所述数据库来访问所述数据资源;
 - c. 包括一个或多个标记,其在使用中导致在相关数据资源中开放监听查询;并且在使用中,所述数据资源被显示在显示器上。
2. 如权利要求 1 所述的移动设备,其中所述对数据资源的引用包括标记语言元素,并且在使用中,所述标记语言元素被所述绘制器绘制。
3. 一种用于将多个数据资源存储在移动设备的文件系统内的方法,所述方法包括以下步骤:
 - 定义包括直接寻址数据内容文件的一个或多个位置;
 - 定义包括数据库的一个或多个位置;
 - 定义一个或多个标记,其在使用中导致在数据资源中开放监听查询;
 - 其中文件系统提供来自于所述用户界面的单独分级界面,以访问所有位置。
4. 如权利要求 3 所述的方法,其中所述方法还包括以下步骤:访问直接寻址数据内容文件,使得所述数据内容文件的内容被显示在所述用户界面内。
5. 如权利要求 3 所述的方法,其中所述方法还包括以下步骤:访问数据库以使数据库查询的结果被显示在所述用户界面内。
6. 如权利要求 3 所述的方法,其中所述方法还包括以下步骤:访问用于生成数据资源的代码,并且绘制所述数据资源以使所述数据资源被显示在所述用户界面内。

虚拟文件系统

技术领域

[0001] 本发明涉及一种虚拟文件系统,特别涉及一种用于与移动通信网络一起使用的移动设备的虚拟文件系统。

背景技术

[0002] 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。移动网络运营商和设备制造者处于品牌塑造的目的保持对设备用户界面的一些方面进行控制的需要,与用户定制和修改他们的设备外观以适合其自身需要的需要之间存在矛盾。提供所需灵活性和用户化需要的复杂软件也与典型移动设备的有限的处理功率和数据存储能力存在矛盾。本发明设法缓解这些问题。

发明内容

[0003] 根据本发明的第一方面,提供了一种移动设备,该移动设备包括:用于存储多个数据资源的存储装置;用于组织移动设备存储装置中存储的多个数据资源的文件系统;和用于提供用户对该多个数据资源的用户访问的用户界面,其中:i) 文件系统包括一个或多个位置,该一个或多个位置包括直接寻址数据资源和间接寻址数据资源,该间接寻址数据资源通过数据提供者是可访问的;并且 ii) 该文件系统被配置成,在使用中,对直接寻址数据资源和间接寻址数据资源,都提供来自用户界面的单独界面。

[0004] 直接寻址数据资源可包括数据内容文件,其在使用中被显示在用户界面内。间接寻址数据资源可包括数据库,并且在使用中,一个或多个查询的结果被显示在用户界面内。而且,间接寻址数据资源包括标记语言元素,并且在使用中,标记语言元素被绘制(render),并且相关结果被显示在用户界面内。

[0005] 根据本发明的第二方面,提供了一种将多个数据资源存储在移动移动设备的文件系统内的方法,该方法包括以下步骤:定义包括一种直接寻址数据资源的一个或多个位置;定义包括间接寻址数据资源的一个或多个位置,该间接寻址数据资源通过数据提供者是可访问的;其中文件系统提供来自于用户界面的单独界面,以访问直接寻址数据资源和间接寻址数据资源。

[0006] 根据本发明的第三方面,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。

附图说明

[0007] 图 1 示出了结合本发明的系统的示意图;

[0008] 图 2 比较详细地示出了服务器 100 的结构和操作;

[0009] 图 3 示出了用于移动设备 300 的软件 400 的示意图;

[0010] 图 4 示出了内容工具箱的示意图;

[0011] 图 5 示出了四个分层的平面的示意图;

[0012] 图 6 示出了根据本发明的一个实施例的设备 800 的示意图,设备 800 包括用户界面。

具体实施方式

[0013] 现将仅通过对于参考附图进行例证的方法,来说明本发明,其中图 1 示出了结合本发明的系统示意图。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 (contentfeed) 500 和用户界面 (UI) 资源 600。在使用中,服务器 100 将内容数据和 UI 数据传递给移动设备 300、301、...,其中每个移动设备都包括软件包 400。服务器 100 通过接口与 OSS 700 连接,该 OSS 是传统上被用于操作移动网络的那些 OSS,例如账单、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 资源 610、620 接收数据,并将 UI 数据打包,使得服务器能够将打包的 UI 数据发射到移动设备 300 内包含的软件包。服务器从多个内容供给 510、520、530 接收数据,并且该数据经过处理和打包,以便它能够被发送到软件包 400 或者以便移动设备 300 能够使用软件包 400 访问数据。

[0014] 可以设想将该系统分成三个分离的域:运营商域包括由移动网络运营商 (MNO) 操作的系统和仪器;用户域包括多个移动设备,而第三方域包括可以被许多不同的实体控制或操作的内容供给和 UI 供给。

[0015] 图 2 比较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 及目录 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。内容以包括一个或多个 Trig 和一个或多个 Triglet 的包 210(见下文)的形式呈现。Trig 是用于如移动电话的移动设备的用户界面,而 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,则这些 trig 之一可以是主要 trig,其它 trig 是从这个 trig 衍生出来的。

[0016] 发布组件用户界面 114 能够被用于将包输入到数据库 111 中,该处理使对每个 trig 和 triglet 的引用被加载到输入队列 114 中,该输入队列可以包括对多个包 210 的引用 (reference)。可以使用用户界面来检查包的内容,包的内容能够被传递给目录。

[0017] 图 3 示出了用于移动设备 300 的软件 400 的示意图,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,多个执行者 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。

[0018] 优选的是,软件使用 TrigML 来工作,TrigML 是 XML 应用程序,标记语言绘制器 410 绘制用于显示在移动设备 300 上的 TrigXML 代码。标记语言绘制器还是用 TrigML 分析器来分析 TrigML 资源,在设备屏幕上显示内容,并控制该浏览手持机上的内容的替换和视图。本地 UI 绘制器用于显示不使用 TrigML 就能显示的 UI 组件,并用于显示错误消息。

[0019] 软件 400 以设备特定的方式被供应和安装。例如,对于诺基亚 (Nokia) 系列 60 设备,使用 SIS 文件来安装软件,而对于 MSSmartphone 设备,使用 CAB 文件来安装软件。同样,以设备特定的方式来处理软件升级。可以以比较有限的格式来供应软件,作为仅绘制其内置内容的配套的应用程序;即,用内置 trig 来供应软件,但是之后不能添加额外的 trig。所提供的 trig 可以用无线电进行升级。

[0020] 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:持续获知使用中的 trig,改变当前 trig,选择初始 trig,选择另一个 trig 作为用于被破坏的 trig 的备用,保持安装的 trig 的集合,识别特殊 trig 被安装到资源管理器的位置,以及读取 trig 的更新信道定义和适当地配置更新管理器。

[0021] 资源管理器提供在设备上对抽象概念的持续存储,即将文件作为真实文件或者作为记录存储到数据库中。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。它负责处理文件路径逻辑,辨别真实资源文件和执行者属性,将关于 trig 的路径映射到绝对路径上,与 trig 管理器通过接口连接,并为更新管理器提供修改接口。

[0022] 资源管理器也负责确保持久存储中存储的资源的完整性,特别是面对不可预测的中断时,如,设备的断电。资源管理器不知道当前使用的 trig。它的接口是线程安全的(因为它可能被来自不同线程的更新管理器和绘制器二者使用)。

[0023] 更新管理器处理 Trig 和 Triglet 的接收和应用。更新管理器对绘制器和 trig 管理器提供接口,并负责:当受到绘制器指示时启动手动更新;当被 trig 管理器如此配置时控制和实施自动更新信道;指示手动更新进程并恢复更新下面的网络连接和/或设备供电的不期望的损失。更新分组格式可以被定义为 XML 模式的二进制序列化。

[0024] 支持管理器提供用于其它组件的界面,以报告事件或者错误的发生。根据错误的严重性,支持管理器将记录事件和/或提供错误消息弹出框。

[0025] XML 是便利的数据格式化语言,其用于定义更新分组格式,以及 TrigML 内容。出于带宽和存储效率的原因,文本 XML 被序列化为二进制表示。由相同的组件,标记语言分析器,对更新分组和 TrigML 分段进行分析。软件中对 XML 的任何其它使用都必须使用二进制 XML 编码,因此必须重新使用分析器。

[0026] 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:当内容是发送给执行者的事件时,由绘制器使用;被想要通知属性值已经改变的执行者使用,和被想要发射事件的执行者使用(见下文)。

[0027] 软件可以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。软件很可能在一个被称为主线程的线程中运行。主线程用于运行绘制器,绘制器与其它组件同步通信。执行者一直具有对绘制器的同步接口。如果执行者由于其功能性需要额外线程,则执行者负责管理内部线程通信。优选的是,使用光消息接发框架(light messaging framework)来避免在很多执行者需要内部线程通信情况下的不必要的代码重复。

[0028] 除主线程之外,更新管理器运行网络线程。网络线程被用于下载更新分组,并从主线程中被分离出来以允许在分组到达之前绘制器继续不受影响。更新管理器负责处理内部线程消息接发,使得当采用更新分组中定义的改变时,更新管理器与绘制器和资源管理器同步通信。

[0029] 软件的存储分配策略是平台特有的。在 MIDP 平台上,软件仅使用系统堆和无用单元收集程序(garbage collector)来满足其存储需求。无论何时在保持无用单元收集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,无用单元收集被强制使用。假定任何存储分配可能会失败,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。

[0030] 在以 C++ 为基础的平台, 将从系统堆进行预先分配和按需分配的混合。启动需要的所有内存是在启动期间按需分配的, 这里任何失败都会导致软件的退出 (如果可以, 伴有消息)。成功启动之后, 预先分配绘制内容文件模型所需内存。如果内容是使用少于定义的限制而创建的, 可确保能够绘制。另外利用 RAM 用于软件的快速操作所需的各种高速缓存。在低内存条件情况下, 将从软件中释放这些导致绘制性能慢的高速缓存。

[0031] 足够严重可以破坏软件正常操作的错误, 必将导致弹出对话框。对话框包含少量的国际化错误消息 (可以在编译的时候, 将这些字符串的国际化译本编辑到软件中, 要显示的错误字符串的译本是由设备上设置的相关语言确定的)。为了将消息的数量保持最小, 只涵盖几个一般问题。

[0032] 为了允许支持状况, 错误对话框还显示错误代码, 作为 4 位数 (16-bit) 十六进制字符串。每个错误代码与能够被支持人员使用而用软件来确定问题实质的说明文本相关联。发生在软件中的并且不足以严重到可以中断其操作的错误, 能够被支持管理组件记录。键入特定键序的用户能够查询支持管理器。支持管理器还通过 HTTP GET 或者 POST 方法将其错误记录提供给服务器。

[0033] 绘制器接收关于键入的信息。如果在编译时没有配置按键行为, 则它作为 TrigML 内容事件被发送到当前焦点元件。然后, 内容事件如 TrigML 的正常事件处理逻辑所定义的那样被处理。

[0034] 例如, 如果按键被按下, “按键”事件连同为相应键设置的参数被传送到绘制器。当该键被释放时, “! 按键”事件被传送到绘制器。如果保持按键按下状态预定时间段, “长按键”事件被传递到绘制器。释放的时候, “! 长按键”和“! 按键”事件二者都会被传递到绘制器。

[0035] 无论何时启动软件, 它都执行下面的操作:

[0036] • 检查并继续进行打断的更新处理;

[0037] • 检查并处理在文件系统中的更新 (或者预先提供的, 或者通过一些其它方法安装到文件系统的);

[0038] • 如果已知, 启动当前 trig (其可以是最近运行的 trig);

[0039] • 如果没有设置当前 trig, 可启动被标记为“默认” trig 的 trig。

[0040] • 在默认 trig 显示失败时, 则将选择按名称字母顺序第一个有效 trig。

[0041] 通过加载定义的资源名称启动 / 默认 (start-up/default), 启动 trig。启动 / 默认中定义的 TrigML 作为内容根节点的新内容被分析。

[0042] 在安装软件后第一次由软件运行 trig 时, 通过加载资源名称启动 / 首次 (startup/firsttime) 来启动 trig。软件可以为 trig 记录该 trig 是否在位于顶层文件夹中的文件中运行。根据移动设备使用的平台, 软件的自动启动可以被设置成编译时间配置选项。此外, 将软件置于自动启动之后的背景中也可以是编译时间配置选项。

[0043] 启动装置可以出现在用户面前, 作为应用程序图标, 并选择它来启动带有该启动装置指定的 trig 的软件 (该 trig 可以由启动装置图标和 / 或名称指示)。当使用启动装置来启动 trig 时, 可以指定“入口点”参数。该参数是在“启动”文件夹中发现的文件的资源名称。如果该 trig 先前从未被运行过, 不使用该文件, 在这种情况下使用被称为“首次” (“firsttime”) 的文件来代替。

[0044] 软件使用设备上的虚拟文件系统中存储的内容资源文件。文件系统被描述为虚拟的是因为它可以不作为传统的文件系统来执行,而是,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。

[0045] 下面在附录 A 中给出关于文件系统布置的细节。此外,软件存储下面的一些或者全部信息:使用统计;有效用户计数;Trig 管理器状态;TrigML 分段和更新信道定义(序列化为二进制 XML);PNG 图像;纯文本,被编码为 UTF-80TA,然后存储在特定编码平台中;其它特定资源平台,例如,铃声文件、背景图片等。

[0046] 文件系统中的文件可以改变,或者是在执行者属性值改变的时候,或者是在用 triglet 替换文件的时候。当 /attrs 目录中的文件改变时,绘制器立刻得到通知,并且内容树的相关分支被更新并刷新。当图像和文本资源改变时,绘制器就像受影响的资源被立刻重新下载那样工作(或者整个内容树被更新,或者只有受影响的分支可以被更新)。当 TrigML 分段被改变时,绘制器就像它没有得到通知那样工作,并继续显示其当前的可能过期的内容。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。

[0047] 软件 400 以设备专用的方式被提供给移动设备。一个或者多个 Trig 可作为安装的一部分来提供,例如,存储为未压缩的更新分组。在启动时,分组可被扩展并安装到文件系统。

[0048] 执行者 445 是公布属性值并处理和发出事件的组件。执行者与绘制器同步通信。如果执行者需要异步工作,则执行者负责管理和与绘制器的主线程外部的线程通信。

[0049] 执行者属性可以作为文件引用被读出。属性是以下四种类型之一:单独的简单值;简单值的矢量;字段的单独结构,每个字段具有简单值;或者结构矢量。可以使用类似于许多面向对象编程语言的 object.member 表示法,用表达式来引用属性:

[0050] <image res = "signallevels/{protocol.signalstrenght}" />

[0051] 当需要作为文件时,属性是经由 /attrs 文件夹访问的。

[0052] <text res = "/attr/network/name">

[0053] 可通过向执行者发送一个用 <throw> 元素的事件来消息通知执行者。执行者发出的事件可作为内容事件被传递到内容树:这些可以被送到一个元素 Id 或者‘顶端(top)’. 对执行者的接口是由执行者接口定义文件定义的。这是定义属性、类型、文件名、事件输入(events-in)和参数,以及事件输出(event out)的 XML 文档。可在编译时为软件配置执行者的集合。附录 B 给出了可以使用的一些执行者的示例性列表,以及相关联的函数或者变量。

[0054] 多数移动设备共有的限制之一在于,显示屏幕相当小,并且当菜单显示出来时,屏幕上不能一直同时显示所有菜单项目。传统方法往往把所有菜单项目以及相关按钮或图形下载到存储器中,然后当用户向上或者向下滚动菜单的时候将它们适当地显示出来。

[0055] 提供了一种将要被下载到存储器中的菜单项目数目限制到能够同时显示在屏幕上的项目数的技术。当用户滚动菜单时,不再显示出来的项目被删除,而当前正在显示的项目被下载到存储器中。

[0056] 优选的是,这可以通过使用 TrigML 中的 <griddata> 元素以定义一些数据的浏览列表来实现,其中数据被存储在文件系统的文件夹中,并且对于每个项目,列表外观具有相

同的结构。<griddata> 元素包括 ‘repeat-over’ 属性, 该属性指定数据可能位于的文件夹。<griddata> 的单独的子元素是列表中的每个项目外观的模板。

[0057] 该模板使用特定符号, 例如, ‘\$\$’ 来表示迭代程序。这是每当该模板被实例化 (instantiated) 时就发生改变的模板变量; 例如

[0058] <griddata repeatover = "news/headlines">

[0059] <text res = "news/headlines/\$\$/title.txt" />

[0060] </griddata>

[0061] 其中文件夹 news/headlines/ 包含:

[0062] 0/title.txt

[0063] 1/title.txt

[0064] 2/title.txt

[0065] 3/title.txt

[0066] 这可以显示 4 个项目的列表, 分别用指向 ‘news/headlines/\$\$’ 文件夹中的 ‘title.txt’ 资源的简单 <text> 元素描述。在资源数据中具有的项目比显示器上 <griddata> 元素的空间所容许的更多的情况下, <griddata> 元素只显示那些能够显示的项目。当用户滚动列表时, <griddata> 元素相应地转换 ‘data-window(数据窗口)’。该技术的优点在于, 实际上只有当前显示需要的资源被加载到存储器中, 这减小了存储器利用, 并减小了绘制项目列表所花费的时间量。

[0067] 可使用类似的方案来定义列表显示的顺序。如果 ‘repeat-over’ 属性的目标是文件而不是文件夹, 则可认为文件包含迭代中使用的资源名称列表。例如,

[0068] <griddata repeatover = "football/league">

[0069] <text res = "football/teams/\$\$/name.txt" />

[0070] </griddata>

[0071] 其中文件 football/league 包含:

[0072] Manchester

[0073] Arsenal

[0074] Chelsea

[0075] 文件夹 football/teams/ 包含:

[0076] Manchester/name.txt

[0077] Arsenal/name.txt

[0078] Chelsea/name.txt

[0079] 并且每个 name.txt 是包含队名的文本文件。其结果是与该队相关联的测试文件会以所定义的顺序被显示在设备显示器的定义的区域。

[0080] 软件利用虚拟文件系统 435, 该虚拟文件系统 435 允许传统文件系统与其它形式的数据相结合, 使得存储在虚拟文件系统的任何部分中的数据都能以通常的方式被访问。能够被结合到这种虚拟文件系统内的非传统数据存储器的实例包括存储在数据库内的数据, 或者由另一软件组件根据命令生成的数据。

[0081] 例如, 在传统的移动设备中, 关于电池强度、信号强度、新文本消息等的信息, 被显示给用户, 并且该信息典型地是通过操作系统向相关硬件或者软件实体发送呼叫并且 UI

对接收到的答复进行译码和显示的来获得的。

[0082] 可以使用 TrigML 标记将该信息显示在 UI 中（见下文），诸如 <phonestatus>（或者 <signalstrength>）。绘制该标记会导致监听查询向相关硬件或者软件实体开放。如果发生状态改变，则 UI 绘制器得到通知，并且 UI 绘制器加载相关按钮或者图形，以将状态改变传送给用户。如果用户改变 UI 内的视图，标记会被撤回，并且监听查询被终止。因为只有当使用标记时监听查询才有效，所以该方法的资源效率更高。

[0083] 虚拟文件系统包括单独的根，其既包含可直接寻址的数据资源，诸如，内容文件（例如，文本、图形、视频、音频等），又包括间接寻址的数据资源，诸如，TrigML 标记或者存储在数据库内的数据。这为用户提供了通过用户界面的统一的浏览，用户界面使用户能够访问保留在虚拟文件系统内的不同类型的数据。虽然不同数据类型以这种方式被提交给用户，但是数据还可以用使得能够进行高效数据存储和检索的方法来存储。

[0084] 由于商业上的原因，需要 MNO 和 / 或内容提供者能够对显示在移动设备的屏幕上的用户界面进行一些控制。一定程度的灵活性也很重要，即允许用户下载 triglet 或者新的 trig 以修改它们的设备外观并且还允许用户对通过使用中的 trig 或者 triglet 确定的显示图像进行进一步改变。

[0085] 这个问题可通过考虑用许多分层平面来形成 UI 来解决，每个等级包括一个或者多个 UI 实体。通过分配层次给 MNO、设备制造商、trig 提供者和设备用户，可以提供所需的允许等级。

[0086] 图 5 示出了四个分层平面 405a-d 的示意图：平面 405a 包括由 MNO 定义的 UI 元素；平面 405b 包括由设备制造商定义的 UI 元素；平面 405c 包括由 trig 定义的 UI 元素；以及，平面 405d 包括由用户定义的 UI 元素。平面 405a 具有分层中的最高位置，并且 405d 具有分层中的最低位置。例如，平面 405a 中的 mno_logo 元素定义了使用的图形元素，和它在设备的显示屏幕上的位置。由于它位于分层中的最高平面中，所以它将一直表现出并将采取比试图使用 mno_logo 所使用的像素的较低分层元素中的任何其它 UI 元素更高的优先权。平面 405d 包括 backgroundcolour（背景颜色）元素，其不是在任何其它平面中定义的，这样，backgroundcolour 中定义的颜色将在 UI 中使用。

[0087] 平面 405c 包括 windowtitle.txt 元素，该元素定义了窗口标题中使用的文本的属性。这可以通过把 windowtitle.txt 元素添加到平面 405a 或者 405b 以定义文本属性，或者通过把 windowtitle.txt_deleted 元素添加到平面 405a 或者 405b 以指示 UI 绘制器忽略任何随后的 windowtitle.txt 元素，来进行改写。

[0088] 用户可设置软件 300 内的优先权，以控制显示在 UI 内的内容。例如，关于许多足球队的内容可以存储在服务器上，使用具有类似于 /demoUI/football/team_xxxx/team_menu.png 的路径，其中 team_xxxx 变量是由用户从队伍列表（manu, chel, leed, manc 等）中选择并插入到路径中，使得 UI 显示关于所选择队伍的内容。team_xxxx 变量的改变将导致显示内容相应地变化。应注意，优先权的选择控制从远程服务器上存储的内容中选择的内容的显示，这与从存储在本地存储器上的内容中选择相反。

[0089] 在服务器需要执行数据库查询，以便于识别要显示的内容的这种情况下，该方法优选地发送这种形式的请求：<http://t1.trigenix.com/triglets/football/triglet&pn=0776655443322>”，这将会极大地增加服务器提供请求内容所需的资源。

[0090] 另一种能够得到相同结果的已知技术是发送这种形式的请求：`http://t1.trigenix.com/triglets/football/triglet&fc='ManU'`，但是该方法的缺点在于，每次添加新的队伍，则服务器逻辑必须被更新以包括新的队伍。相反，该方法需要内容在新的位置被添加到服务器，该处理比较简单，并且实现它所需要的资源较少。

[0091] 更新包括新的 trig(新的或者替换 UI) 或者 triglet(对现有 trig 的修改)，并且可以被看作对软件文件系统的修改。更新管理器用于通过读取分组来确定文件系统的什么需要改变。更新分组可以被软件 400 使用 HTTP 或者其它适当的传输机制无线地下载，或者被用设备专用分组格式打包，或者与软件本身的安装一起被预先提供。

[0092] 更新可以用许多方式来触发，包括：

- [0093] • 启动时检查中断的更新处理的软件
- [0094] • 启动时检查预先安装的更新分组的软件
- [0095] • 如更新信道配置的那样自动地
- [0096] • 用户启动
- [0097] • 接收专用 SMS 的设备

[0098] 用于解包和安装更新的算法是设备专用的。然而，算法免受非预期的中断（例如，断电）是很重要的，这样，文件系统中不会达到恶化或者不可恢复的状态。这可以通过使用两个线程（一个网络线程和一个绘制器线程）来实现，目标是使网络线程执行尽可能多的更新处理，以使得绘制器线程的中断时间尽可能最短。

[0099] TrigML 分段是包含文本 TrigML 的文件，并且这些分段内部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。该文件还定义了 trig 的其它性质。当用于编译 triglet 时，该文件还定义了输入 TrigML/PNG/Text 资源怎样映射到 trig 的虚拟文件系统的修改上。对于 PNG 和文本资源，trigDefinition 文件指向主机文件系统上的真实文件列表，并且资源被复制到输出。

[0100] TrigML 可使用常量代替属性值。常量是用与 `<include>` 参数相同的语法访问的，例如，`$background colour`。常数被当作 trig 中的全局变量，并且被定义为在保留的文件夹 `constants/` 中。可以在编译的时候通过直接替换它们的值来解析 (resovle) `constants/` 文件夹中的文件包含的变量定义。可替代地，`constants/` 中的变量定义作为全局变量被编译，并且在内容分析的时候被软件解析。这允许通过它的 `constants` 文件中的一个或者全部的简单替换来更新 trig。

[0101] System String Dictionary(系统字符串字典)定义了使用所有已知字符串的整数值，即，保留字。这些具有几种类型，包括：TrigML 元素和属性名称（‘group’、‘res’、‘layer’、‘image’、‘x’），TrigML 属性值（例如：‘left’、‘activate’、‘focus’）和公共资源路径（例如：‘attr’、‘start-up’、‘default’）。作为输入，String Dictionary 是可选的。首次编译 trig 的时候，它将不具有 String Dictionary。该首次编译增加了 StringDictionary，然后将其用于该 trig 的所有即将到来的编译。Triglet 编译必须具有定义了它正在修改的 trig 所使用的所有字符串映射的 StringDictionary。

[0102] 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、一致的层规则、可在压缩绘制器中实现、提供多层和任意重叠内容、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属性

值的平台的 hooks、可扩展,并且图形上灵活。TrigML 提供这些特征和我们的提交于 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。

[0103] 我们希望重新标记 UI 和产生持续更新流的消耗达到最小。这可以通过提供来自创建处理经由数据传输到达用户的高效信息流来实现。

[0104] 被称作包的容器,被用于 UI、UI 更新,以及用于包括的第三方的模板。这些包包含第三方产生、测试和传递标记的 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。

[0105] 包处理包括五个处理阶段:

[0106] 1) 脚本环境 220 提供为一个或者多个 UI 设计模板和基于该模板更新 UI 策略的手段。

[0107] 2) 维护环境 240 提供快速 UI 和良好控制和引导的环境下产生更新,其能够被内容提供者从外部获得。

[0108] 3) 维护环境 240 ‘预检’(‘pre-flight’)功能允许配置管理员检查和调整它们从第三方所接收到的 UI 和更新。

[0109] 4) 公布组件 110 提供对 UI 的管理,并在配置点处更新,包括新版本的分级。

[0110] 5) 公布组件 110 使得能够从即时内容供给自动生成更新。

[0111] 在典型项目中,在脚本环境 220 内创建包以用于:使内容提供者从模板创建重新标记的 UI,结合相同的‘感觉(feel)’不同的‘外观(look)’;使内容提供者从模板创建更新,其提供周期性的,或者用户选择的 UI 内容的变化;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。

[0112] 对于所有这些用途情况,维护环境 240 被用于引入包、重新标记和重新配置内容,并创建用于提交给公布组件 110 新的包。在 UI 模板的设计中,应考虑下面的问题:UI 的哪部分能够被重新标记;UI 的哪个特征需要在重新标记的时候或者远程地被重新配置;UI 内容的哪部分可以被更新;以及,如果 UI 被重新标记,则用户能够选择使用的内容供给。脚本环境 220 允许定义这些策略,并使得维护环境 240 作为每个策略的每个例子的实现者。

[0113] 包是由脚本环境 220 生成的,脚本环境 220 包括模板 UI 或者用于编辑的更新。编辑完成后,该包被保存在‘待发箱(outbox)’中,准备好用于分配给维护环境 240,以公布给内容服务器。提供了下面的‘打包(parcel)’功能。维护环境 240 能够被用于编辑/替换包中保留的资源。包能够被输出到仿真环境,以测试移动设备上的 UI 或者 UI 更新的性能。

[0114] 很多不同的 UI 能够从公共基础上派生出来。典型地,公共基础会实现界面本身的大部分,从它派生出的 Trig 会实现关于它的小改变,诸如标记。Triglet 能够从 Trig 中派生出来,并且它能够覆盖任何来自它选择的父 Trig 的资源(可选地,它可以引入它自身的资源)。注意“资源”这里也指 TrigML,所以 Trig 的行为和布置能够被 Triglet 修改,和它替换单独的图像或者文本一样容易。

[0115] 包可以包括一个或者多个基本 Trig(即,不是从任何其它 trig 中派生出的 Trig),从基本 Trig 派生出的一个或者多个 Trig,从任何 trig 派生的多个 triglet 和从其它 triglet 派生的多个 triglet。

[0116] 包格式是将所有这种信息存储为序列化目标的不透明的二值格式。包可以包括许

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。每个资源包括关于如何浏览、编辑、或删除资源的许可信息。而且,每个资源包括元信息,诸如,关于该资源的文件编制和指令。每个包工具或者假定相关的角色,或者要求用户作为特定的角色登陆。

[0117] 图6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。设备包括显示用户界面815的显示器810,和用户界面装置820,用户界面装置820使得用户能够与用户界面815交互。处理器830执行存储在一个或者多个存储装置840内的软件,并且可以设置一个或多个无线通信接口850,以能够与其它设备和/或通信网络进行通信。可以容纳一个或者多个电池860来为设备供电,其也可以包括接口,用来接收电功率和/或通信电缆。

[0118] 这些组件和接口的性质将取决于设备的性质。应理解,这种用户界面能够实现在移动或者蜂窝电话手持机内,但是也可以应用到其它便携设备,如数码相机、个人数字管理器、数字音乐播放器、GPS导航设备、便携游戏控制台等。而且,也可应用于包括如膝上型电脑或者台式电脑的用户界面的其它设备。

[0119] 用户界面装置可以包括多个按钮,诸如,数字键盘或者字母数字混合键盘,或者触摸屏等。一个或多个存储设备可以包括非易失性存储器的形式,诸如存储卡,使得断电的时候存储的数据不丢失。ROM存储器装置可以设置用来存储不需要更新或者改变的数据。一些RAM可以被设置用于暂时存储器,作为被频繁访问的数据的超高速缓存的较快速反应时间支持。设备可以接受用户移动存储卡,可选地,硬盘驱动器可以用作存储装置。使用的存储装置将通过平衡设备大小、功率消耗、所需存储器容量等的不同需求来确定。

[0120] 这种设备可以与任何无线通信网络虚拟结合来实现,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或者CDMA-2000网络,和对这些以及类似网络的改进和衍生物。在建筑物和校园内,也可以使用其它技术,如蓝牙、IrDa或者无线LAN(不管基于无线还是光学系统)。USB和/或固件连接可以被提供用于与其它设备的数据同步和/或用于电池充电。

[0121] 如上所述的用于实现方法和/或用于配置设备的计算机软件可以设置在数据载体上,诸如,软盘、CD-ROM、DVD、非易失性存储卡等。

[0122] 该申请要求2004年2月19日提交的UK专利申请第0403709.9号的优先权,其内容作为参考结合在本文中。

[0123] 附录A

[0124] 对于‘/’开始的文件路径:

[0125]

/attrs	和unix/proc目录一样,当需要属性作为文件引用时,存储用于内容引用的执行者属性值。
<actor>(执行者)	/attrs的每个子目录都是执行者名称。

/attrs	和 unix/proc 目录一样,当需要属性作为文件引用时,存储用于内容引用的执行者属性值。
<attribute>(属性)	每个属性都作为执行者子目录中的节点被访问。
<field>(字段)	如果属性是一种结构,则字段名称指定要访问的结构成员。
<index>(索引)	如果属性是矢量属性,则索引号指定到所需属性的矢量中的索引。
<field>(字段)	如果矢量属性是结构的集合(collection),则字段名称再次指定结构成员。

[0126] 不以 '/' 开头的文件路径被认为是关于当前 trig 的,即,每个 trig 都存储在来源于单独文件夹的其自己的文件夹层次中。

[0127]

config	每个 trig 中的公共文件夹,用于存储 trig 元数据。
channels(信道)	公共文件夹,用于存储更新信道定义。
<channeldefs>	定义 trig 的更新信道收集(collection)的文件集合。每个文件能够定义一个或者多个更新信道。
start-up(启动)	公共文件夹,用来存储 trig 的实体点。
default(默认)	公共 TrigML 文件,用于存储 trig 的默认实体点。
firsttime(首次)	公共 TrigML 文件,用于存储该 trig 首次运行时使用的 TrigML。
<tirgmlfiles>	如果在启动文件夹中发现其它名称的 TrigML 文件,其它名称的 TrigML 文件可用作实体点。

config	每个 trig 中的公共文件夹,用于存储 trig 元数据。
constants	该文件夹不是通过的 OTA,相反是完全在内容编译的时候被解析的。
<restofcontent>	trig 内容被组织成 Trigs 文件夹下的 trig-defined 格式。

[0128] 附录 B

[0129]

Trigplayer Actor (Trigplayer 执行者)	Attributes (属性)	UpdateState	
	Message (消息)	exit	
		predial_mode	on/off
	Event (事件)	idle	
Launch Actor (启动执行者)	Attributes (属性)		
	Message (消息)	browser	url
		SMS	Number
			message
		Camera	
		Inbox	
		profiles	
		missed_calls	number
		dialer	
		...	
	native_app	app_id	
		url	
	Event (事件)		

[0130]

[0131]

[0132]

Install Actor (安装执行者)	Attributes (属性)		
	Message (消息)	ringtone	resource_path
		wallpaper	resource_path
	Event (事件)		
Phone Actor (电话执行者)	Attributes (属性)	Bluetooth	
		IrDA	
		Call	
		GPRS	
		UnreadSMS	
		UnreadVoiceMail	
		UnreadMsgs	
		BatteryLevel	
		SignalStrength	
	Message (消息)		
	Event (事件)	missed_call	
		message_arrived	
voice_mail_arrived			

[0133]

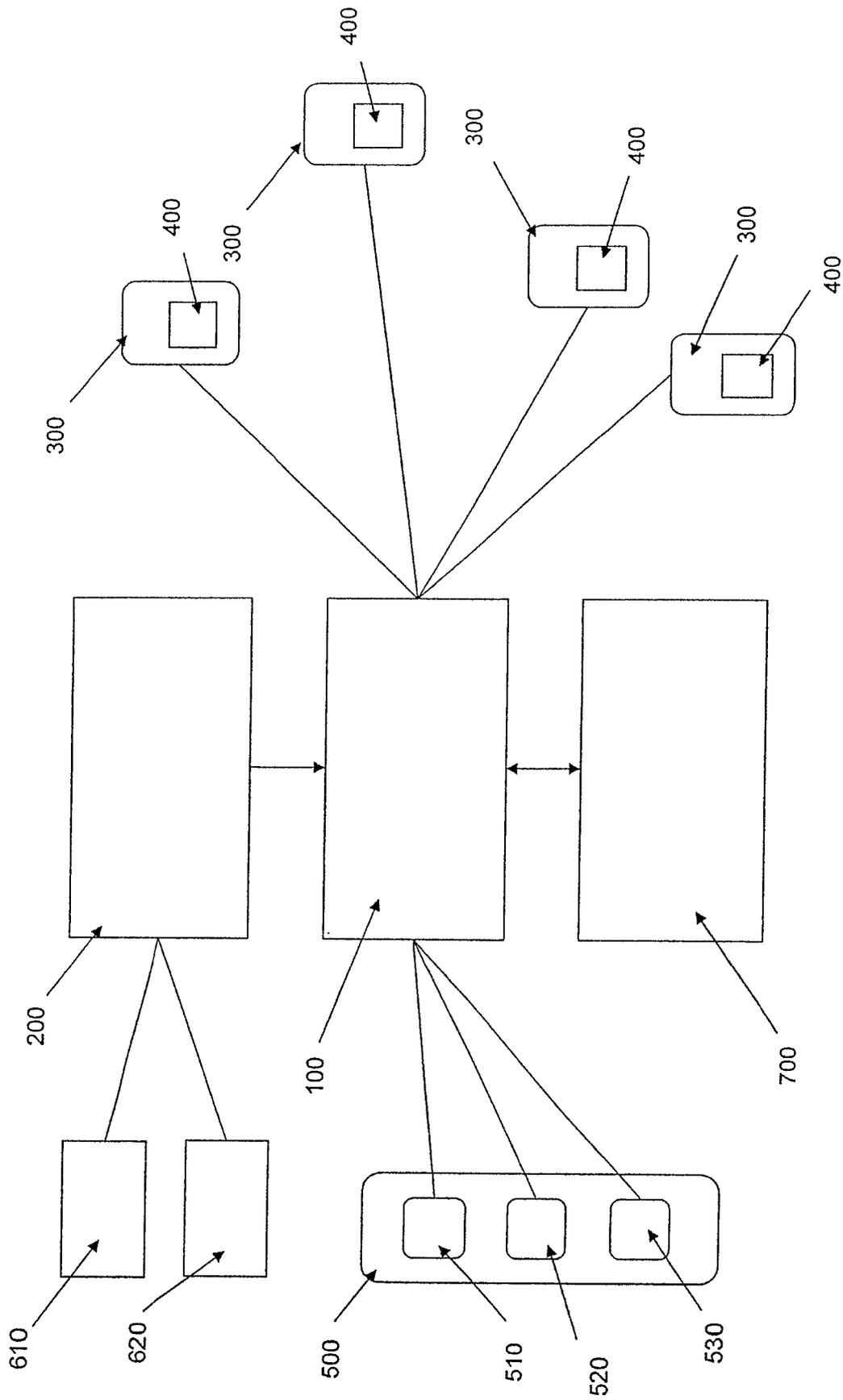


图 1

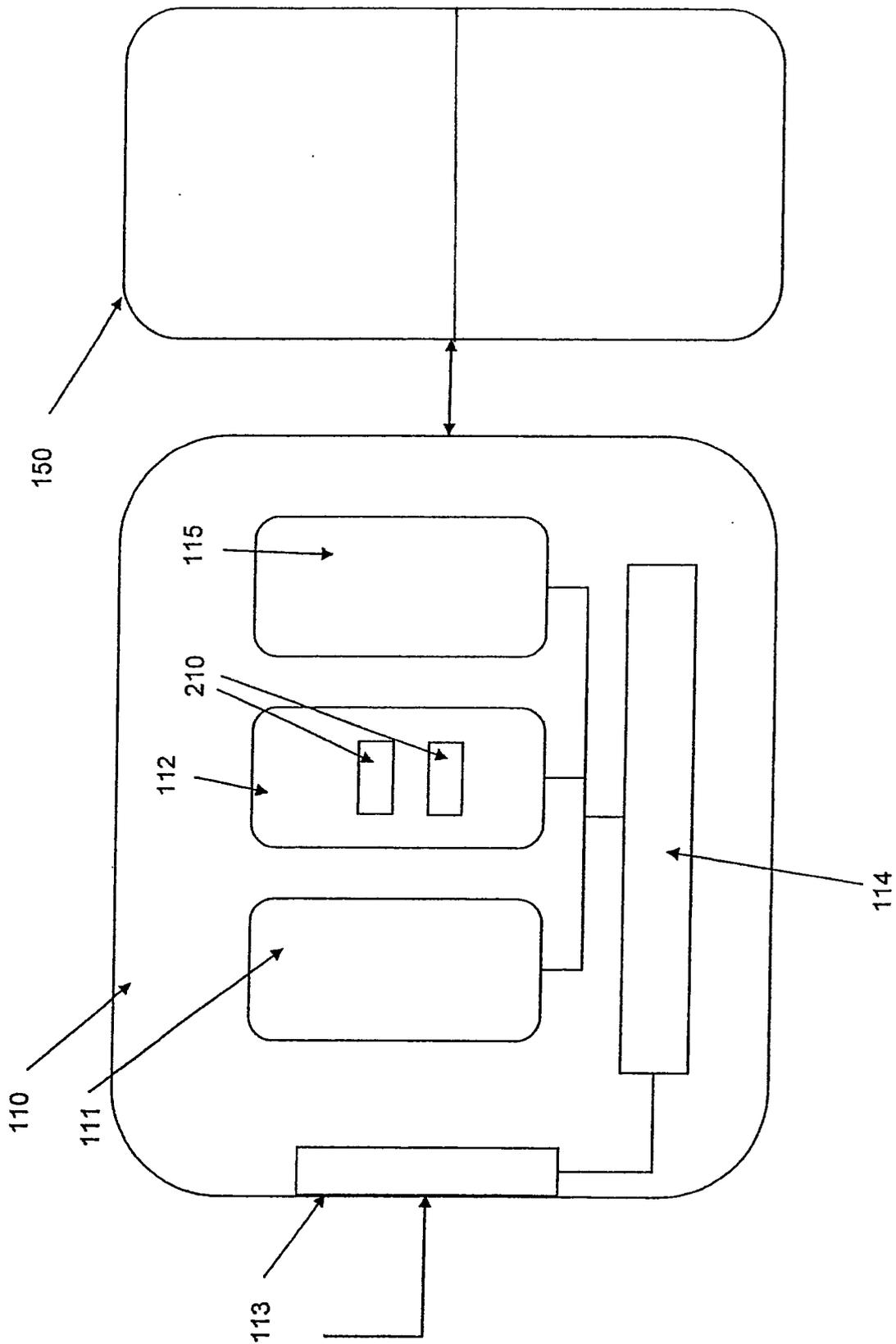


图 2

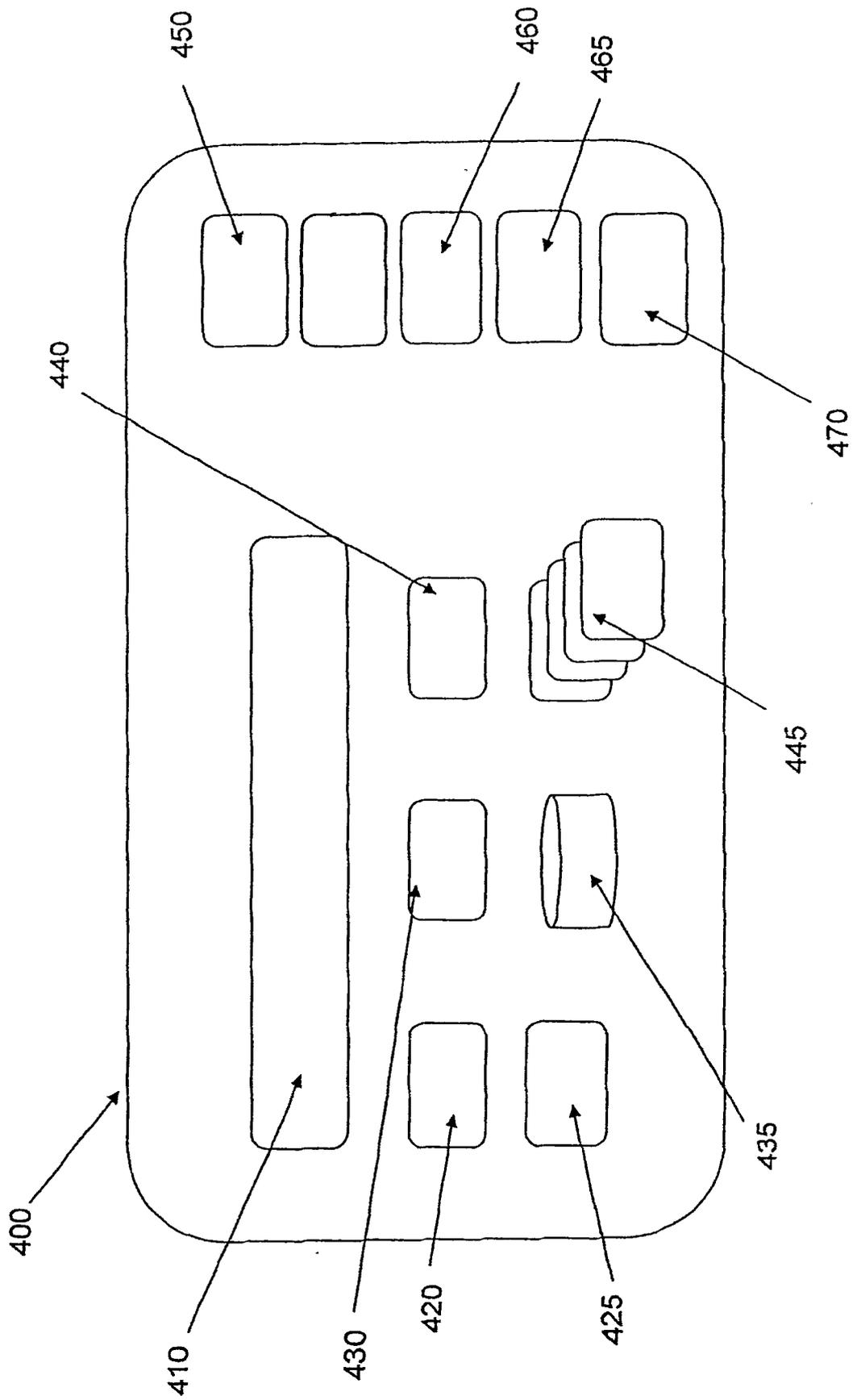


图 3

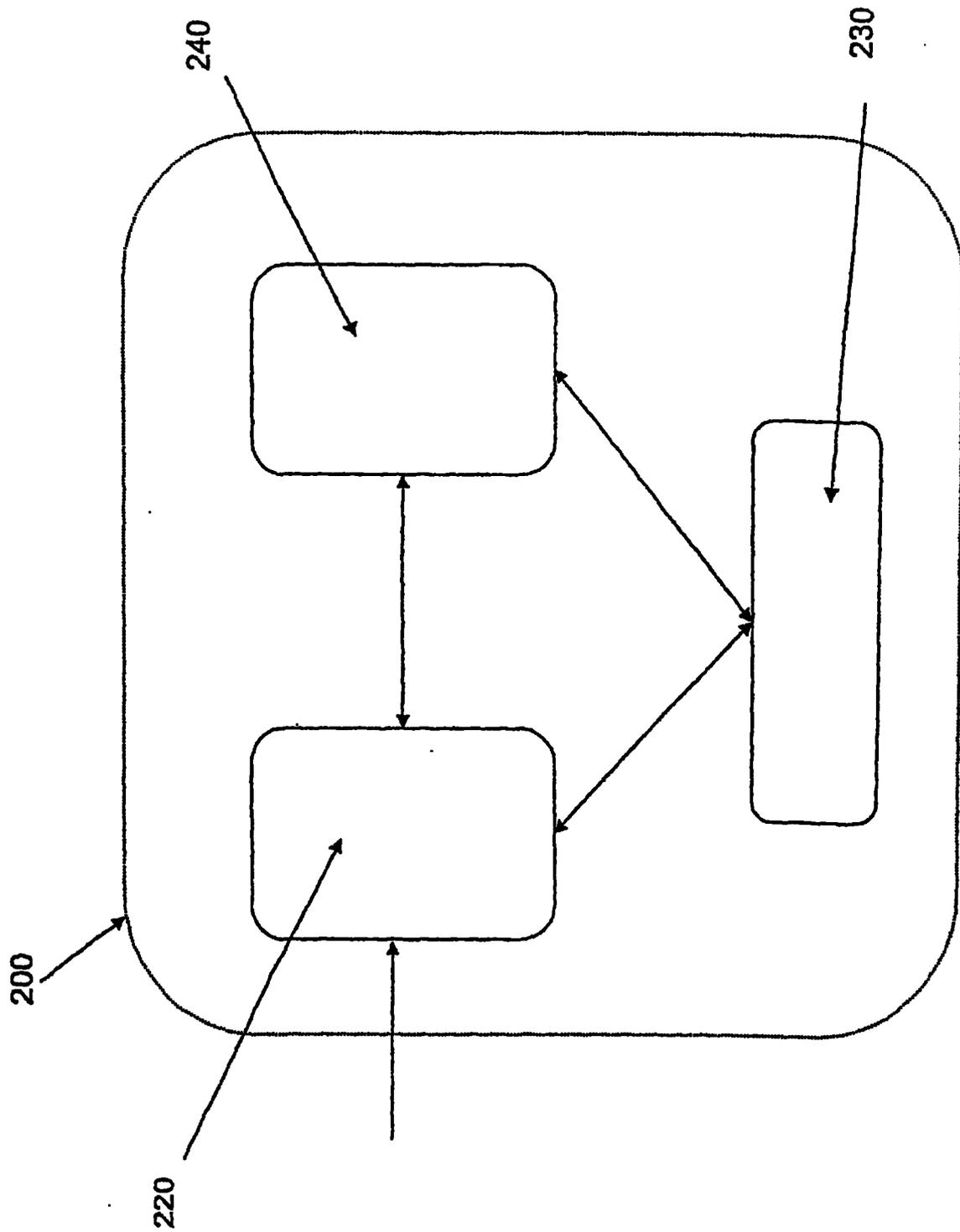


图 4

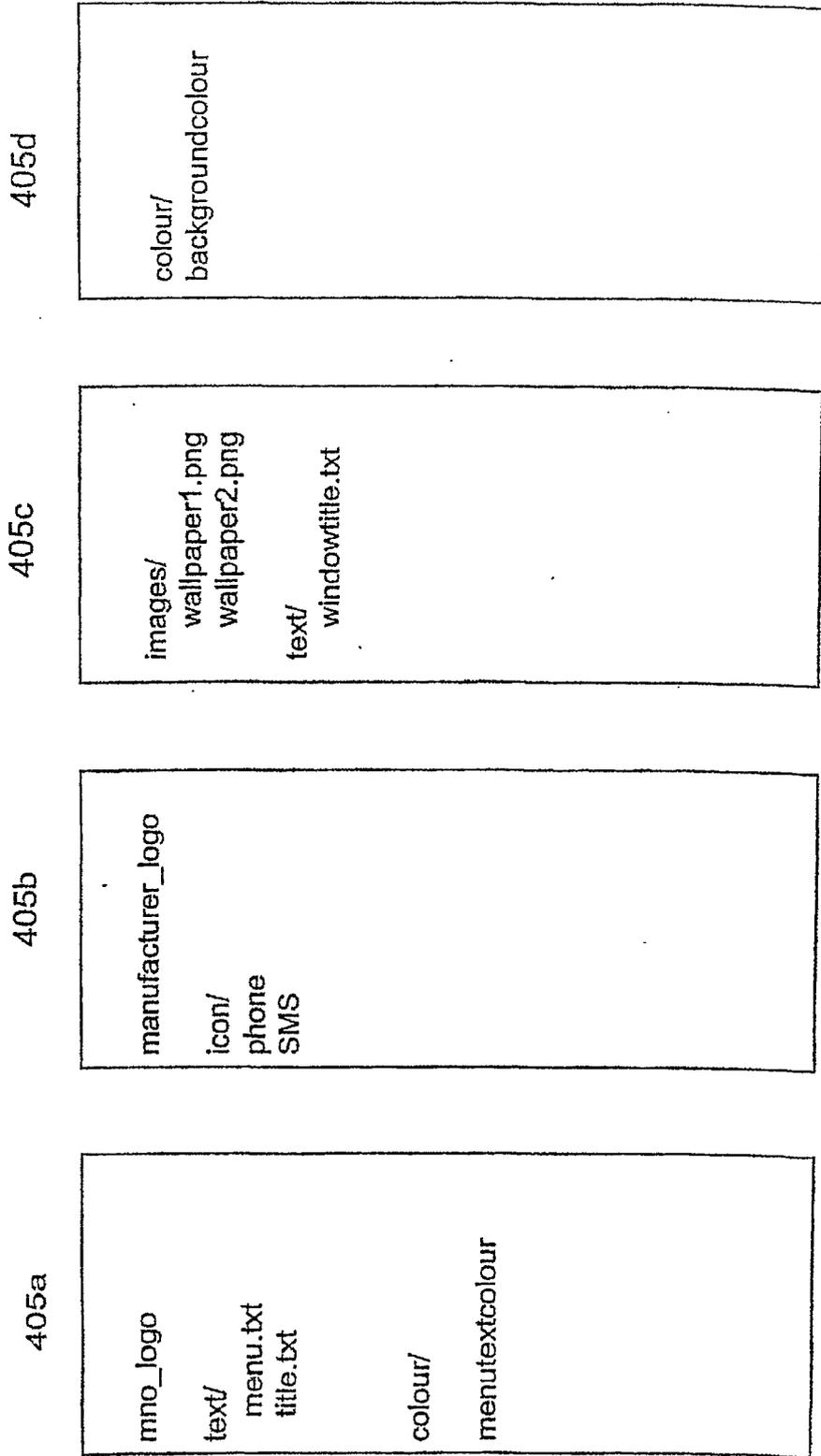


图 5

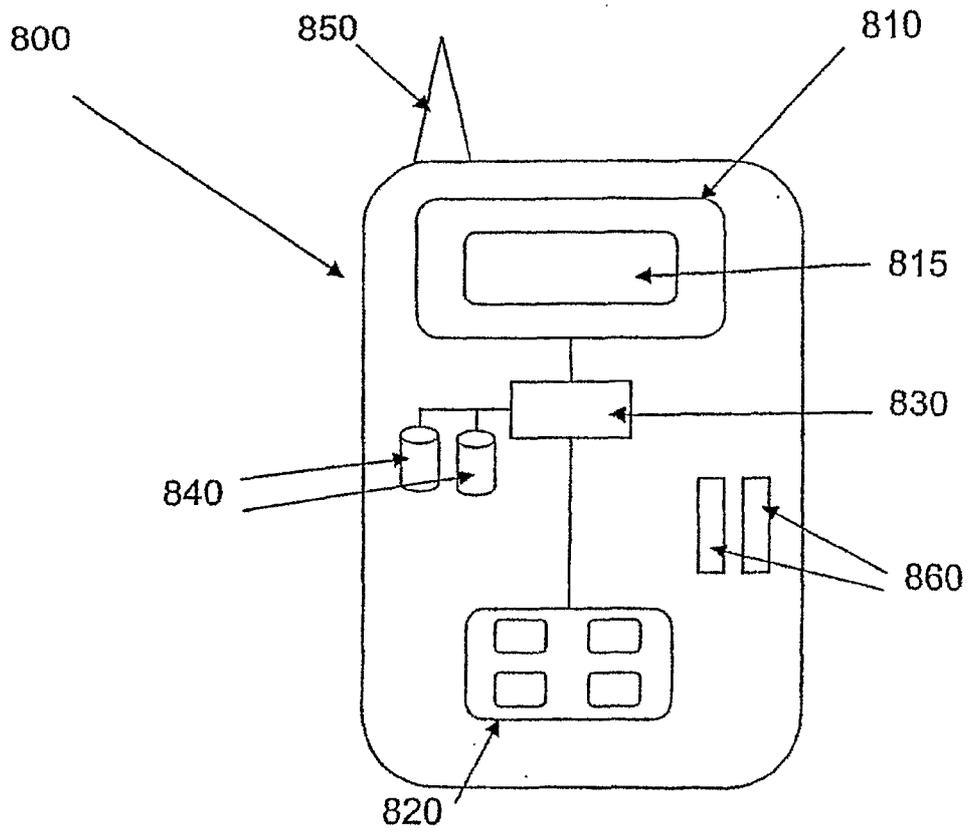


图 6