



(12) 发明专利申请

(10) 申请公布号 CN 104866701 A

(43) 申请公布日 2015. 08. 26

(21) 申请号 201410663259. 4

(22) 申请日 2014. 11. 19

(30) 优先权数据

14156001. 1 2014. 02. 20 EP

(71) 申请人 西门子公司

地址 德国慕尼黑

(72) 发明人 K. 多恩

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 谢强 熊雪梅

(51) Int. Cl.

G06F 19/00(2011. 01)

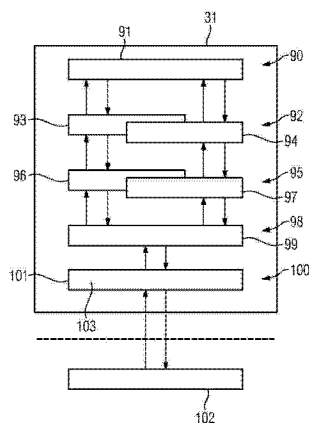
权利要求书2页 说明书50页 附图6页

(54) 发明名称

用于管理和处理医学设施的数据的系统

(57) 摘要

本发明涉及一种用于管理和处理医学设施(2, 2')的数据的系统(1)。该系统(1)包括至少一个用于在用户设备(8)上运行的浏览器应用(31-33), 该浏览器应用(31-33)被构造为, 为了查看和处理, 显示数据组列表(35)以及各个列表条目(27), 该数据组列表对于至少一个特定数据类型的多个医学数据组(P, W, C)分别包含列表条目(27)。浏览器应用(31-33)具有多层的组件架构, 具有: 框架层(90)、视图层(92)、视图模型层(95)、模型层(98)和驱动器层(100)。视图模型层(95)的组件(96, 97)在此是通用性的, 也就是独立于数据类型。



1. 一种用于管理和处理医学设施 (2, 2') 的数据的系统 (1), 具有至少一个用于在用户设备 (8) 上运行的浏览器应用 (31-33), 其中, 所述浏览器应用 (31-33) 被构造为, 用于为了查看和处理而显示

- 数据组列表 (35), 该数据组列表对于至少一个特定数据类型的多个医学数据组 (P, W, C) 分别包含列表条目 (27), 和

- 各个列表条目 (27),

其中, 所述浏览器应用 (31-33) 关于其组件架构被分为如下的层:

- 框架层 (90), 包括至少一个框架组件 (91), 其中, 所述框架组件 (91) 定义了同时待显示的分别用于数据组列表 (35) 的 GUI 页 (36) 或分别用于列表条目 (27) 的 GUI 页 (52) 的数量和布置, 并且其中, 所述框架组件 (91) 还实现了在不同的 GUI 页 (36, 52) 之间的控制转移的功能,

- 视图层 (92), 包括多个组件 (93, 94), 这些组件中的每一个特定于确定的数据类型分别定义了用于显示数据组列表 (35) 的 GUI 页 (36) 或用于显示列表条目 (27) 的 GUI 页 (52) 的图形内容,

- 视图模型层 (95), 包括多个组件 (96, 97), 这些组件中的每一个对于所述视图层 (92) 的至少一个相关的组件 (93, 94) 定义了对于在那里定义的 GUI 页 (36, 52) 的控制元件的特征和命令,

- 模型层 (98), 包括多个组件 (99), 这些组件中的每一个特定于确定的数据类型分别定义了相关的数据组 (P, W, C) 的结构, 以及

- 驱动器层 (100), 包括多个驱动器组件 (101), 这些组件中的每一个分别促成对在用户设备 (8) 内部或外部的特定的数据存储单元 (19) 的数据访问,

其中, 所述视图模型层 (95) 的组件 (96, 97) 是通用性的, 也就是独立于数据类型。

2. 根据权利要求 1 所述的系统 (1), 其中, 所述驱动器组件 (101) 是通用性的, 也就是独立于数据类型。

3. 根据权利要求 1 或 2 所述的系统 (1), 其中, 该或每个框架组件 (91) 是通用性的, 也就是独立于数据类型。

4. 根据权利要求 1 至 3 中任一项所述的系统 (1), 其中, 所述驱动器层 (100) 包括该或每个驱动器组件 (101) 所基于的应用编程接口 (103)。

5. 根据权利要求 1 至 4 中任一项所述的系统 (1), 其中, 所述视图模型层 (95) 的组件 (96, 97) 包括用于创建、读取、更新和删除数据组列表 (35) 或各个列表条目 (27) 的通用性功能。

6. 根据权利要求 1 至 5 中任一项所述的系统 (1), 其中, 所述浏览器应用 (31-33) 被构造为用于显示关于至少一个以下数据类型的数据组 (P, W, C) 的数据组列表 (35) 和列表条目 (27):

- 患者数据, 其每个数据组 (P) 分别包括关于特定患者的个人的和医学的数据,

- 任务或任务列表, 其每个数据组 (W) 包括由医学用户待完成的任务或任务序列,

- 上下文数据, 其每个数据组 (C) 包括关于医学图像数据 (B) 的上下文信息, 特别是对该图像数据组 (B) 的存储位置的参考 (29)。

7. 根据权利要求 1 至 6 中任一项所述的系统 (1), 其中, 所述浏览器应用 (31-33) 被

构造为,直接地无需求中间连接的服务器地访问在用户设备(8)外部布置的数据存储器(19)中,特别是公共云(13)的云存储器(19)中的数据组(P,W,C)和/或其相关的列表条目(27)。

8. 根据权利要求1至7中任一项所述的系统(1),具有应用模板和应用配置器,其中所述应用配置器被构造为,从所述应用模板中根据预先给定的配置信息自动地生成该或每个浏览器应用(31-33),其中所述配置信息包括以下说明:

- 关于用户设备(8)的说明,在所述用户设备上应当运行各自的浏览器应用(31-33),
- 关于数据组(P,W,C)的该或每个数据类型的说明,其相关的列表条目(27)和数据组列表(35)应当借助各自的浏览器应用(31-33)被显示和处理,和/或
- 关于各自的浏览器应用(31-33)应当进行访问的该或每个数据存储器(102)的说明。

9. 根据权利要求1至8中任一项所述的系统(1),其中,该或每个驱动器组件(101)能够多次实例化。

10. 根据权利要求1至9中任一项所述的系统(1),其中,通过该或每个驱动器组件(101)异步地且独立于另外的驱动器组件(101)地提供数据访问。

用于管理和处理医学设施的数据的系统

技术领域

[0001] 本发明涉及一种用于管理和处理医学设施的数据的系统。

背景技术

[0002] 术语“管理”在此和以下尤其被理解为将数据存档（即，在持久的数据存储器中建立数据）、从数据存储器中再现（显示）和删除数据以及根据预定标准从数据存储器中分类和找到特定的数据（浏览）。

[0003] 术语“处理”在此和以下尤其被理解为改变（编辑 / 准备）数据。

[0004] 医学设施的待管理和待处理的数据尤其包括对于医学设施的人员的患者数据、任务 (Works/Tasks) 或任务列表 (Worklists)，以及医学图像数据。

[0005] 这样的医学数据的管理越来越多地借助计算机辅助地通过服务系统、特别是所谓的信息系统进行。信息系统有规律地包括

[0006] - 一个或多个数据存储器，例如以存储区域网络 (SAN) 的形式，

[0007] - 一个或多个对应的数据服务器，

[0008] - 至少一个在数据库服务器中实现的关系数据库，

[0009] - 以及通常一个或多个另外的服务器，在该另外的服务器中执行用于数据库访问和数据处理的方法。

[0010] 在医学领域中对于不同的数据类型建立不同的信息系统。由此在医学设施、诸如诊所的环境下通常对于患者数据的管理和处理采用“医院信息系统” (Hospital Information System, HIS) 并且对于放射学检查的时间安排采用“放射学信息系统” (RIS)，其辅助医学图像数据的诊断并且记录诊断结果。此外，医院的 IT 结构通常包括所谓的“图片存档及通信系统” (PACS)，用于基于 DICOM 标准来存档和传输医学图像数据，以及“高级显示 (AV) 系统”，其提供服务器辅助的用于可视化体积数据、特别是动态体绘制的功能。

[0011] 上述服务器系统在此通常彼此并行地存在。这要求高的购置和维护开销，所述开销尤其对于小的医学设施或具有相对小的资金量的其它设施几乎不能承受。

[0012] 此外，现代医学设施的上面描述的复杂 IT 结构仅具有相对差的缩放特性。由此，这样的 IT 结构与待处理的和待存档的数据体积的和 / 或所需的计算功率的较大改变的匹配大多仅能通过极高的开销实现。

[0013] 迄今大多使用个人计算机 (PC) 作为这样的 IT 结构的用户设备或终端设备（典型地称为客户机），其中该 PC 通常被构造为所谓的“瘦客户机 (Thin Client)”，其从连接的服务器取用所需计算功率的大部分。但是最近逐渐期望也使用移动式的微型电子计算机，诸如智能电话、平板计算机或 PDA，作为用户设备。

[0014] 在医学领域中的常规的信息系统的另一个问题在于，该系统的前端软件 (Frontend-Software) 通常特定且固定地对应特定数据类型的管理和处理。这导致每个信息系统的前端必须被特殊地编程和维护。这尤其又使将新型的用户设备（诸如智能电话和平板计算机）集成到临床工作流程变得困难，因为与各个前端的前端匹配相连的、软件组

件的多样化关于制造和扩展开销仅能以大的开销实现。

[0015] 作为常规的客户机-服务器架构的替换,最近几年越来越多地建立所谓的云解决方案。在此将由独立于用户的云运营商(“云计算服务提供商,Cloud Vendor”)提供和运行的数据处理装置理解为云(“计算云,Rechnerwolke”)。“云计算服务提供商”在此向众多用户在使用协议(订阅)的框架内提供云的硬件和必要时软件作为服务。根据所提供的服务的范围,区分:

[0016] - 称为“基础架构即服务”(IaaS)的使用模式,其中仅向用户提供云的计算机硬件(计算机、网络和存储器),而用户本身完全负责在云中运行的软件,

[0017] - 称为“平台即服务”(PaaS)的使用模式,其中从云向用户提供计算机硬件以及在其上形成的编程和运行时环境,从而用户本身仅负责在该编程和运行时环境中实施的应用软件(应用),以及

[0018] - 称为“软件即服务”(SaaS)的使用模式,其中此外也从云向用户提供特定的应用软件。

[0019] 根据各个云所面向的用户范围,还区分:

[0020] - 所谓的公共云(Public Cloud),其服务可以被任何人使用,和

[0021] - 所谓的私有云(Privat Cloud),其仅被特定组织的用户、特别是特定集团的用户访问。

[0022] 对于公共云的每个用户,对云的特定硬件和软件部件的访问权利通过与用户对应的订阅来控制。公共云由此通常具有“多客户机能力”(多租客)。这是指该能力可以保持数据、用户管理和计算操作对于用户以不同的订阅而严格地分开。也就是,公共云的用户不能看到具有不同订阅的另外的用户的数据、用户管理和计算操作并且也不能影响该数据。

发明内容

[0023] 本发明要解决的技术问题是,提供一种用于管理和处理医学设施的数据的系统,其尤其可以灵活地使用,特别是可以缩放。

[0024] 上述技术问题通过按照本发明的特征来解决。本发明的有利的和部分对于本发明设置的构造和扩展在从属权利要求和下面的描述中说明。

[0025] 按照本发明的系统尤其用于显示和处理下面数据类型中的一种的医学数据组:

[0026] - 患者数据,其包括设施的关于特定患者的个人的和医学的说明,特别是关于姓名、地址、年龄、性别、以前的诊断和治疗的说明,

[0027] - 任务(Works/Tasks)和任务列表(Worklists),其可以由医学设施的一个或多个医学用户(医生、医学助手、护士等)执行,以及

[0028] - 上下文数据,也就是关于医学图像数据的上下文信息,特别参考图像数据组的存储位置以及可选地关于图像数据组的元数据,例如作为各个图像拍摄的基础的拍摄参数。

[0029] 特定数据类型的数据组在下面也被称为“资源”。

[0030] 系统包括至少一个用于在用户设备上运行的浏览器应用。该或每个浏览器应用在此一方面被构造为用于显示数据组列表,该数据组列表对于多个医学数据组分别包含一个列表条目(Listeneintrag)。该(下面称为“消息来源(Feed)”的)数据组列表特别地(但不是强制性地)是对具有特定特征的数据组进行搜索的结果,例如对关于具有特定姓氏的

患者的患者数据组进行搜索。

[0031] 该或每个浏览器应用此外被构造为,显示各个列表条目(下面称为“留言(Tweet)”)的内容,以用于视图或处理。每个留言可以完全或部分地包含相关的数据组本身的内容,在患者数据的情况下例如患者的姓氏、名字、地址等,从而数据组与相关的留言可以完全或部分地相同。但是特别是在数据组具有大的存储需求的情况下数据组或该数据组的单个分量优选与相关的留言分开地存储。在该情况下,与数据组相关的留言包括一个或多个链接(特别是以 URL 的形式)而不是数据组的信息,通过该链接可以访问数据组或该数据组的单个分量。此外,与数据组相关的留言通常还包括如下链接(仍然优选以 URL 的形式),该链接指示相关的另外的数据组的存储位置。例如与患者的患者数据组相关的留言通常包含到同样与该患者相关联的任务和/或上下文数据组的链接。

[0032] 该或每个浏览器应用优选用于选择性地和单独地显示和处理特定数据类型的数据组,从而为了一方面管理和处理患者数据以及另一方面为了显示和处理任务列表而设置不同的浏览器应用。然而在本发明的框架中也存在用于管理和处理多个数据类型的数据组的浏览器应用的实施方式。

[0033] 按照本发明,浏览器应用关于其组件架构分为五层。这些层相应于其从“上”到“下”的顺序(即,从表现层和用户交互层至数据访问层)包括框架层、视图层、视图模型层、模型层和驱动器层。

[0034] 框架层优选恰好包括一个组件。框架层的组件(下面称为“框架组件”)按照合适的实施表示用于通过视图层的组件定义的 GUI 页的通用性的运行框架(Runtime-Container,运行容器)。向外,“框架组件”在此作为自主地在用户设备的操作系统上运行的本来的应用而呈现,例如作为 IOS-App、Phone-Gap-Exe 或 Windows8-App。但是替换地,其也可以被构造为用于在 Web 浏览器中运行,并且在该方案中例如作为 HTML5 程序构造。框架组件由此优选作为在该或每个 GUI 页与应用的运行时环境(特别是用户设备的操作系统或 Web 浏览器)之间的软件技术适配器使用。

[0035] 框架组件定义了同时待显示的图形用户界面(GUI)的(GUI)页的数量和布置,其中这些 GUI 页的每一页或者用于显示消息来源(也就是留言的列表)或者用于显示留言的内容。在此,图形用户界面的封闭的且本身起作用的部分被称为 GUI 页,其例如根据视窗的类型可选地可以单独地或与另外的 GUI 页结合地运行。该或每个框架组件还包括在不同的 GUI 页之间的控制转移的功能。在此,通过用户手动地或通过系统自动地发起的、用来将 GUI 页激活(允许用户交互)或禁止(禁止用户交互),在显示屏上淡入或从显示屏淡出的动作被称为控制转移。

[0036] 该或每个框架组件在此可以从用户角度来看或者是不可见的或者本身形成浏览器应用的图形用户界面的上级部分,并且在此尤其给该或每个显示的 GUI 页加上(按字面意思的)框架。

[0037] 由于框架组件确定了显示的 GUI 页的数量和布置、控制转移的类型以及必要时浏览器应用的图形用户界面的部分,其(根据用户设备的类型)定义了与该用户设备匹配的应用式样。例如按照为了实施浏览器应用而在平板计算机上设置的框架组件,在显示屏表面的激活区域中分别并排显示分别用于消息来源或留言的三个 GUI 页,其中 GUI 页的行可以通过手动拖动平板计算机的触摸屏而在显示屏上迤拽,由此可以将边缘的 GUI 页从激

活区域移出（并且由此淡出和淡出）并且将另外的 GUI 页移入激活区域中（并且由此激活和淡入）。相反按照为了实施浏览器应用而在智能电话上设置的框架组件，对于每个时间点仅显示分别用于一个消息来源或留言的 GUI 页，其中为了在不同的 GUI 页之间变换例如又在触摸屏上手动拖拽或对控制元件进行一定的操作。

[0038] 视图层包括多个组件，即至少一个组件，但优选两个或多个组件。视图层的每个组件特定于确定的数据类型分别定义了用于显示消息来源或用于显示留言的 GUI 页的图形内容。在合适的实施中，视图层对于浏览器应用所针对的每个数据类型包括，

[0039] - 分别一个组件（示例性地称为“<Ressource>ListView”），其定义了用于显示消息来源的 GUI 页，和

[0040] - 另一个组件（示例性地称为“<Ressource>ListItemView”），其定义了用于显示留言的 GUI 页。

[0041] 仅用于管理和处理患者数据的相应的浏览器应用由此在视图层中包括两个组件（即“<Patient>ListView”和“<Patient>ListItemView”）。

[0042] 视图层的每个组件在此特别地定义了相关的 GUI 页的图形元件的类型、外观和布置，特别是控制元件的，通过其可以向用户输出信息或接收用户输入或命令。相反，通过视图层的组件没有规定与控制元件相连的逻辑（UI 逻辑），因为这保留给视图模型层的组件。

[0043] 视图模型层包括多个组件，这些组件中的每一个与视图层的至少一个组件对应。如前面说明的那样，视图模型层的每个组件实施 UI 逻辑，也就是用于在视图层的相关的组件中定义的 GUI 页的控制元件的功能性特征和命令。在合适的实施中，视图模型层包括两个组件，其中第一组件（示例性地称为“GenList-ViewModel”）实现用于显示消息来源的 GUI 页的 UI 逻辑，而第二组件（示例性地称为“GenListItem-ViewModel”）实现用于显示留言的 GUI 页的 UI 逻辑。

[0044] 模型层包括多个组件，也就是仍然包括至少一个组件。模型层的该或每个组件在此特定于确定的数据类型定义了相关的数据组的结构，特别是与数据组相关的变量。

[0045] 驱动器层同样包括多个组件，也就是还是包括至少一个组件，但优选两个或多个组件。驱动器层的每个组件（下面称为“驱动器组件”）在此被构造为，促成上级设置的层对特定的数据存储器的数据访问。浏览器应用由此仅通过针对相应的数据存储器（在所述数据存储器上存储了留言或数据组）的那个驱动器组件加载消息来源和留言和 / 或基础数据组。通过驱动器组件促使的数据访问在此是双向的并且由此能够通过浏览器应用实现（特别是异步地）下载以及（仍然特别是异步地）上传消息来源、留言和必要时的数据组。通过该或每个驱动器组件在浏览器应用的 GUI 和各自的数据存储器之间建立自动数据连接。

[0046] 数据存储器可以是内部数据存储器，例如用户设备的主存储器，或者是外部数据存储器。在后一种情况下，数据存储器特别地通过由相关的驱动器组件所直接（无要求中间连接的服务器或服务）访问的公共云的云存储器给出。概括地，数据存储器在本发明的框架中一般可以是（无线的或有线连接的）计算机网络的任意参与者，通过该计算机网络可以交换医学数据组或相关的消息来源和留言，其中通过各自的驱动器组件可以使用任意的数据传输协议（例如 E-Mail、SMS 或推特 (Twitter)）。只要驱动器层包括多个驱动器组件，则所述驱动器组件优选与不同的数据存储器（例如来自于多个云计算服务提供商的云存储器）对应。

[0047] 视图层、视图模型层和模型层相应于按照所谓的MVVM(Model View ViewModel)方案的基本原则,如在对具有图形用户界面的应用进行编程时通常地样。在此,按照本申请的组件架构与通常的MVVM方案的主要区别在于,视图模型层的组件是通用性的,也就是独立于所处理的数据组的特定的数据类型。因此可以不变地使用视图模型层的相同组件来处理不同数据类型的数据组(例如患者数据以及任务列表)或相关的消息来源和留言。视图模型层的通用性结构在此能够显著降低对于该或每个浏览器应用的制造开销。特别地由此能够实现该或每个浏览器应用的完全或至少几乎自动的生成。

[0048] 此外,根据按照本发明的教导通过两个另外的层,即框架层和驱动器层,对通常的MVVM模型进行扩展。通过层架构的最上面补充的框架层在此能够实现,在将应用与不同的用户设备类型(PC、智能电话、平板计算机等)匹配时不变地继续使用浏览器应用的大部分组件,由此显著降低了对于多样的应用方案的制造开销。通过在层架构的最下面补充的、具有其必要时的多个可更换的驱动器组件的驱动器层,又能够实现,灵活地使用具有完全不同的数据存储器和浏览器应用(例如用户设备的本地存储器与一个或多个公共云计算服务提供商的云存储器一样)。

[0049] 在系统的优选的实施中,驱动器层的组件和/或框架层的组件也是通用性的,也就是独立于所处理的数据组的数据类型。这能够实现进一步降低对于该或每个浏览器应用的制造开销并且同样有利于该或每个浏览器应用的自动生成。

[0050] 优选地,视图模型层的组件在如下程度实现了通用性的CRUD(Create-Read-Update-Delete,创建-读取-更新-删除)功能性,即,视图模型层的组件包括用于创建(Create)、读取(Read)、更新(Update)和删除>Delete)消息来源(其中包含留言)的功能,其可以被应用于关于任意数据类型的数据组的消息来源和留言。因此,该功能可以不变地应用到消息来源和留言或在系统的框架内管理的所有数据类型的数据组。合适地,关于留言定义的功能至少部分地也作用于相关的数据组。例如在删除与患者数据组对应的留言时也删除该患者数据组。

[0051] 优选地,视图层、视图模型层和模型层的组件可以多次实例化。在此合适地分别给消息来源的每个(由浏览器应用显示的或在GUI后台保持的)GUI页或留言的每个GUI页配备视图层、视图模型层和模型层的相关的组件的单独的实例。在此合适地并行实施视图层、视图模型层和模型层的不同实例。

[0052] 此外优选地,每个驱动器组件也可以多次实例化,其中也给每个GUI页配备合适的驱动器组件的单独的实例。合适地也并行地实施驱动器组件的不同实例。

[0053] 驱动器组件优选被构造为,异步地提供数据访问。由此这样构造在驱动器组件和上级层的组件之间的控制流,使得该上级层的组件在查询各自的驱动器组件时不等待驱动器组件的响应。

[0054] 为了能够将浏览器应用以特别小的开销与不同的数据存储器和相匹配,驱动器层优选包括预定的应用编程接口(下面称为“驱动器API”),驱动器层的所有驱动器组件都基于所述应用编程接口。

[0055] 在特别具有优势的实施方案中,系统包括应用模板(即,在其中可配置地预先给出浏览器应用的基本架构的应用模板)以及应用配置器。应用配置器被构造为,从应用模板中根据预先给定的配置信息自动地生成该或每个浏览器应用。配置信息尤其包括以下说

明：

[0056] -关于用户设备的说明，在所述用户设备上应当运行各自的浏览器应用。该说明例如包括设备类型的标记、操作系统的名称等，

[0057] -关于应当借助待生成的浏览器应用来显示并处理的数据组以及相关的消息来源或留言的该或每个数据类型的说明，和 / 或

[0058] -关于各自的浏览器应用应当访问的该或每个数据存储器的说明。

[0059] 应用配置器根据该说明，配置分别具有合适的组件的应用模板并且由此生成浏览器应用。

[0060] 应用配置器可以通过不取决于浏览器应用的独立的软件来实现。但优选地，应用配置器的功能集成在框架组件中。优选地，为此在数据存储器中提供例如作为 XML 文件存在的应用模板以及不同版本的框架组件，其中每个版本的框架组件针对特定的用户设备类型和必要时特定的应用式样（在此隐含地通过选择合适版本的框架组件给出关于用户设备的类型的配置信息）。在加载各自的框架组件时，框架组件然后根据应用模板通过自动地下载下级层的合适的组件来配置浏览器应用。

[0061] 在其中暂存应用模板和至少一个框架组件的数据存储器优选是公共云的云存储器（App Store），从该云存储器中可以将框架组件和应用模板下载到用户设备。但是替换地也可以将应用模板和 / 或至少一个框架组件暂存在用户设备的存储器或移动存储介质（例如 CD-Rom 或 USB 棒）中。

附图说明

[0062] 下面结合附图对本发明的实施例作进一步说明。附图中：

[0063] 图 1 按照示意性的框图示出了用于管理和处理医学设施的数据的系统，具有多个模态和用户设备或终端设备（装置）以及具有与模态和装置经由因特网相连的公共云，

[0064] 图 2 按照示意性的框图示出了系统的软件技术的结构，由此系统对于医学设施的不同类型的数据在公共云的云存储器中分别具有特殊对应的存储区（中心，Hub），也就是用于患者数据组的患者中心、用于任务的工作列表中心、用于上下文数据组的上下文中心和用于图像数据组的图像中心，并且其中系统在每个装置上对于每个中心包含用于访问在该中心中存储的数据组的对应的应用，其中在三个才提到的中心中分别提供表存储器（Table Storage），在该表存储器中对于中心的每个数据组包含对应的列表条目（留言），并且其中每个应用被构造为，查找相关的中心的表存储器（浏览）以及提取并显示所选择的留言的数据组列表（消息来源），并且其中构造与图像中心对应的应用来显示（查看）图像数据组，

[0065] 图 3 按照示意性的图示示出了为了运行而在智能电话上设置的浏览器应用的用户界面（GUI）的 GUI 页，其用于显示各个对应的中心的消息来源，

[0066] 图 4 示出了在按照图 3 的示图中浏览器应用的用户界面的另一页，其用于显示相关的中心的留言，

[0067] 图 5 示出了在按照图 3 的示图中为了运行而在个人计算机、笔记本电脑或平板计算机上设置的浏览器应用的方案，其中可以同时显示用户界面的多个 GUI 页，

[0068] 图 6 按照示意性的框图示出了按照图 3 和图 4 的浏览器应用或按照图 5 的浏览器

应用的组件架构,和

[0069] 图 7 按照示意性的框图示出了按照图 5 的浏览器应用。

[0070] 彼此相应的部分和参数在所有附图中始终具有相同的附图标记。

具体实施方式

[0071] 图 1 按照粗略示意性简化示出了用于管理和处理医学设施 2 的医学数据的系统 1, 该医学设施例如是诊所。

[0072] 系统 1 在硬件侧包括多个模态 3, 也就是设施 2 的成像医学检查设备。由此, 系统 1 在按照图 1 的示例性示图中包括计算机断层造影设备 4、C 形臂设备 5 和磁共振断层造影设备 6。每个模态 3 与一个(控制和分析)计算机 7 对应。

[0073] 此外, 系统 1 在硬件侧还包括设施 2 的多个用户设备或终端设备(下面为装置 8), 其用于显示和处理数据。在按照图 1 的简化示例中, 装置 8 包括具有所连接的显示屏 10 的个人计算机 9、平板电脑 11 以及智能电话 12。

[0074] 作为在设施 2 外部布置的部件, 系统 1 包括公共云 13。在系统 1 的范围内例如使用由微软公司以名称“Windows Azure”提供的服务作为公共云 13。但与之不同地在本发明的范围内也可以考虑另外的公共云或多个公共云的组合(必要时也来自不同的提供者)作为公共云 13。

[0075] 公共云 13 经由(数据传输)网络 14 与系统 1 的设施内部的组件、也就是模态 3 和装置 8 相连。该网络连接在设施 2 内部通过设施 2 的内网 15 构成, 其例如被构造为所谓的基于电缆连接的以太网技术的局域网(LAN)和/或无线的无线局域网(WLAN)。在设施 2 外部通过因特网 16 构成网络 14。在内网 15 和因特网 16 之间的接口处通常布置防火墙 17。

[0076] 在系统 1 的范围内由公共云 13 提供的服务通过被称为订阅 18 的使用协议来规定。订阅 18 在此控制, 公共云 13 的哪些硬件和软件部件可以被系统 1 的设施内部组件访问。关于术语订阅 18, 在下面是指公共云 13 的、专门配备给在系统 1 的范围内的设施 2 的那部分。公共云 13 的另外的区域可以(如在图 1 中表示的那样)在另外的订阅 18' 的范围内分配给另外的医学设施 2'。每个设施 2、2' 在此仅能访问其按照其订阅 18、18' 所分配的数据和服务, 而不能访问另外的设施 2 或 2' 的数据和服务。公共云 13 在该意义上具有“多客户机能力”(多租客)。订阅 18、18' 的“客户机”(租客)优选是医院、医生诊疗室或其它医学设施。但基本上还可以预见的是, 订阅 18、18' 被分配给较小的单位或个人团体直至单个患者。

[0077] 按照订阅 18, 在公共云 13 内向设施 2 的系统 1 提供

[0078] - 下面称为云存储器 19 的数据存储器,

[0079] - 下面称为 App Store 20 的、用于应用(应用软件)的存储器以及

[0080] - 云计算服务 21。

[0081] 云存储器 19 用于持久地存储设施 2 的数据。App Store 20 为应用、驱动器和其它外围软件提供例如配置文件和模板(Template), 其可以由系统 1 的设施内部的组件, 也就是模态 3 和装置 8 来下载并且在系统 1 运行时在模态 3(更确切地说对应的计算机 7)或在装置 8 上运行。云计算服务 21 在系统的范围内仅可选地用于在模态 3 或装置 8 本身上没有进行的所有计算操作。所述计算操作尤其涉及为了存储(预处理)而对各个模态 3 的三

维图像数据（体积数据）进行的计算量大的准备和 / 或为了对这样的体积数据进行二维可视化（图像合成, 体绘制）而对绘制的视图 V（图像场景或场景图）进行的导出。

[0082] 在设施 2 的范围内待管理和待处理的数据按照图 2 包括患者数据组 P、任务 W、上下文数据组 C 和图像数据组 B, 其中这些数据分别具有上面详细解释的内容。图像数据组 B 在此可以包含医学图像数据或体积数据或其它图像、声音、图形或文本信息（照片、视频、文件）。

[0083] 按照图 2, 对于这些不同的数据类型（资源）的每个, 在云存储器 19 中分别提供单独的存储区, 其在下面被称为（资源）中心。由此在云存储器 19 中提供

[0084] -（患者）中心 22, 用于存储患者数据组 P,

[0085] -（工作列表）中心 23, 用于存储任务 W,

[0086] -（上下文）中心 24, 用于存储上下文数据组 C, 以及

[0087] -（图像）中心 25, 用于存储图像数据组 B。

[0088] 每个中心 22-24 分别包含表存储器（下面称为表存储器 26）。

[0089] 在系统 1 的范围内对于在分别对应的中心 22-24 的表存储器 26 中存储的每个数据组提供一个列表条目, 其在下面被称为留言 27。

[0090] 在其性质上由相对小的离散的表达组成的数据组的情况下, 在此全部数据组或至少一部分数据组作为留言 27 存储在各自的中心 22-24 的表存储器 26 中。这在按照图 2 的示例中涉及患者数据组 P、任务 W、以及上下文数据组 C。然而替代表存储器 26, 用于根据先入先出原则 (First-in-first-out-Prinzip) 存储任务 W 的工作列表中心 23 也可以具有所谓的队列存储器 (Queue-Storage)。

[0091] 为了存储通常基本上由较大且非文字数字的数据块构成的图像数据组 B, 在图像中心 25 中替代表存储器 26 设置所谓的 Blob 存储器 28, 在该 Blob 存储器中图像数据组 B 作为“二进制大对象 (Binary Large Objects, BLOB)”以不进一步结构化的形式存储。对于在图像中心 25 中存储的图像数据组 B, 在上下文中心 24 中分别存储以留言 27 形式的上下文数据组 C。该留言 27 包含至少一个 URL 29 (统一资源定位符, Uniform Resource Locator), 其标记图像数据组 B 在图像中心 25 的 Blob 存储器 28 中的存储位置。

[0092] 对于每个中心 22-25 在 App Store 20 中提供应用 31-34, 其分别选择性地用于显示和处理在分别对应的中心 22-25 中存储的数据组。每个应用 31-34 由此表示在装置 8 的层面上的对应的中心 22-25。应用 31-34 因此（依据资源中心）也称为“应用中心”。

[0093] 特别地, 在 App Store 20 中提供的应用由此包括

[0094] - 用于管理和处理在患者中心 22 中存储的患者数据组 P 的应用 31,

[0095] - 用于管理和处理任务 W 的应用 32,

[0096] - 用于管理和处理上下文数据组 C 的应用 33, 以及

[0097] - 用于管理和处理在图像中心 25 中存储的图像数据组 B 的应用 34。

[0098] 应用 31-34 可以从 App Store 20 下载到每个装置 8 并且在那里执行。只要系统 1 (如图 1 所示那样) 包括不同类型的装置 (例如个人计算机 9、平板计算机 11 和 / 或智能电话 12), 在 App Store 20 中就分别提供为了在各自的装置 8 上运行而匹配的不同版本的应用 31-34。但是优选地, 应用 31-34 及其变形不是以预制的状态暂存在 App Store 20 中。而是在 App Store 20 中至少对于功能类似的 (浏览器) 应用 31-33 存储以 XML 文件形式

的通用应用模板,在需要时通过装置 8 中的一个借助应用配置器根据预先给定的配置信息从该通用应用模板中自动地生成各自的应用。

[0099] 例如为了在平板计算机 11 和智能电话 12 上运行以与各自的设备类型匹配的 App 的形式提供应用 31-34,而为个人计算机 9 提供的、应用 31-34 的版本被设计为用于在 Web 浏览器中运行。

[0100] 与通常的用于管理医学数据的 IT 结构相比,患者中心 22 和对应的应用 31 从用户角度来看例如接管通常的医院信息系统 (HIS) 的功能。特别地,在患者中心 22 中记录所有与人有关的(并且数据保护重要的)患者数据,其由此对于各个患者建立电子患者档案。其余的中心 23-25 优选不包含与特定患者对应的信息(私人信息,Privacy Information)。工作列表中心 23 和对应的应用 32 对于用户基本上与通常的放射学信息系统 (RIS) 对应,这通过列出在设施 1 内将要执行任务来实现。上下文中心 24 和对应的应用 33 从用户角度来看例如接管通常的 PACS 的功能,这通过在那里可以存储并调查为查找和处理图像数据组 B 所需的信息来实现。最后,图像中心 25 和对应的应用 34 例如接管通常的 AV 系统的功能。

[0101] 虽然在中心 22-25 和对应的应用 31-34 中包含的功能完全彼此相关联,但是中心 22-25 连同分别对应的应用 31-34 也可以分别不取决于另外的中心 22-25 及其应用 31-34 运行。由此例如也可以在没有上下文中心 24 的相关的上下文数据组 C 的条件下访问在图像中心 25 中包含的图像数据组 B,如果与图像中心 25 对应的应用 34 以另外的方式,例如通过 E-Mail、SMS 或推特,提供待显示的图像数据组 B 的 URL 29 的话。此外,患者数据 P 也可以暂存在公共云 13 之外,因为对于其余的中心 23-25 和对应的应用 32-34 的功能来说不需要患者中心 22。

[0102] 应用 31-33 仅用于从数据存储单元中根据预定的标准找到特定的数据(浏览)。其特别地不用于显示大版式的医学图像数据(查看)。大多数时候其可以示出所谓的邮票图像,也就是强烈缩小的医学图像数据组的图像,作为在其列表内的特征性信息。现在这样设计该浏览应用,使得其列表全部由云存储器 19 的 REST 标准提供,由此可以取消为此的单独的服务器服务。

[0103] 与上下文中心 24 对应的浏览应用 33 管理例如与患者的检查(研究)一起显示的条目(留言)的列表(消息来源),并且可以将其各自的内容直接进一步传输到图像中心 25 的查看应用 34。然而特别地,应用 33(和相关的上下文中心 24)不取决于查看应用 34(和相关的图像中心 25)。应用 33 几乎等效于 PACS 浏览器,但其不是用于查看(也就是图像显示),而是可以进一步给出分别以留言的形式对待显示的图像数据组的参考。

[0104] 与工作列表中心 23 对应的应用 32 管理例如与“客户机”(租客)的命令(指令)一起显示的条目(留言)的列表(消息来源),并且可以将其各自的内容直接进一步传输到与上下文中心 23 对应的应用 33。然而特别地,应用 32(和相关的工作列表中心 23)不取决于应用 33(和相关的上下文中心 24)。应用 32 几乎等效于 RIS 浏览器,但其不是呈现上下文信息,而是可以分别以留言的形式进一步给出对此的参考。更概要地表达可以是,应用 32 管理对上下文信息的参考的列表,其具有更短期的性质,并且典型地可以在短的时间段内(例如在一天内)处理。

[0105] 与患者中心 22 对应的应用 31 管理例如与租客的患者档案一起显示的条目(留言)的列表(消息来源),并且可以将其各自的内容直接进一步传输到应用 32。然而特别

地,应用 31(和患者中心 22)不取决于应用 32(和相关的工作列表中心 23)。应用 31 在此几乎等效于 HIS/EPR 浏览器,但其不是呈现工作列表或上下文信息,而是可以分别以留言的形式进一步给出对此的参考。更概要地表达可以是,应用 31 管理对上下文信息的参考的列表,其具有更长期的永久的性质,并且典型地显示患者在其活着期间(通过不同的检查)获得的上下文,并且其被用作参考检查。

[0106] 与患者中心 22 对应的应用 31 是浏览器应用,其在患者中心 22 的表存储器 26 中查找留言 27,该留言相应于特定的由用户预先给定的搜索条件。从患者中心 22 在此根据应用 31 的搜索查询 Q(图 2)将下面称为消息来源 35 的数据组列表返回给应用 31,该数据组列表包含相应于搜索查询 Q 的留言 27。应用 31 在装置 8 上在图 3 中示例性示出的图形用户界面 37(GUI)的 GUI 页 36 上显示获得的消息来源 35。

[0107] 在按照图 3 的示例性实施中,GUI 页 36 基本上具有四个不同功能的显示范围,也就是页眉 40、搜索区 41、显示区 42 和命令条 43。在页眉 40 中为与应用 31 对应的中心 22-24 命名。在应用 31 的示例中页眉 40 由此例如以词语“患者中心”标注。此外在页眉 40 中优选为系统所配备的设施 2 命名。

[0108] 搜索区 41 包含输入行 44,经由该输入行可以输入文字数字的搜索条件,以及按键 45,通过由用户按下该按键可以基于之前在输入行 44 中输入的搜索条件开始搜索查询 Q。

[0109] 附加地或替换地,搜索区 41 包含预选区 46,通过该预选区可以利用必要时存在的指示设备通过轻敲(在触摸敏感的显示屏的情况下)或点击来选择一个或多个预设的搜索条件。

[0110] 可通过预选区 46 生成的搜索条件尤其是通常使用的搜索方案。例如预选区 46 在与患者中心 22 对应的应用 31 的情况下能够实现如下预选择,即是否应当将

[0111] - 所有留言 27(“所有”)或

[0112] - 仅关于预定要进行检查的患者的那些留言 27(“预定的”)或

[0113] - 仅关于已经结算了过往服务的患者的那些留言 27(“结帐的”)

[0114] 作为消息来源 35 返回。

[0115] 可选地设置,在通过预选区 26 生成搜索条件的情况下自动地在输入行 44 中显示该搜索条件的文字数字对应性。

[0116] 在显示区 42 中显示基于搜索查询 Q 返回的消息来源 35。显示区 42 由此包含以列表形式的基于应用 31 的搜索查询 Q 在患者中心 22 中找到的留言 27。对于每个留言 27 例如显示缩小图像 47 和 / 或关键字 48。由此在与患者中心 22 对应的应用 31 中对于每个显示的留言 27 优选显示患者的缩小图像以及患者的姓名和可选的患者识别码。搜索查询 Q,和基于其返回的消息来源 35 可以由用户在应用 31-33 的运行时间交互地改变。

[0117] 基于按照 REST 或 OData(开放数据协议标准,Open Data Protocol Standard)的语法定义搜索。REST(Representational State Transfer)在此指用于因特网应用的编程范例,其满足如下原则:

[0118] - 给每个提供的服务分配唯一的地址(统一资源定位符,Uniform Resource Locator,缩写 URL)。

[0119] - 根据地址访问的服务可以具有不同的表现形式(表达)。

[0120] - 使用无状态(stateless)协议,其能够实现始终独立于彼此地处理不同的查询。

[0121] - 在所有资源上必须应用具有定义的特征的预先给定数量的操作（即特别是四个操作 GET、POST、PUT、DELETE）。

[0122] - 对于应用信息以及对于状态变化都使用多媒体超文本（超媒体）。

[0123] 在此，表达优选按照因特网标准 XML (Extensible Markup Language)、JSON (JavaScript Object Notation) 和 / 或 ATOM 进行。ATOM 在此作为对于 ATOM 联合格式 (Atom Syndication Format, ASF) 的议题被使用，其表示用于交换消息的 XML 格式，以及 ATOM 发布协议 (ATOM Publishing Protocol, APP)，其表示用于生成和处理 Web 内容的编程接口。

[0124] 搜索功能基本上类似于根据 SQL 标准的 WHERE 条款。但其与后者不同，不取决于静态的索引重建 (Re-Indexierung)。特别地，中心的表不需要用于搜索的索引元素 (Index-Element)，因此也不需要索引重建，并且相应地也不必保持单独的数据服务器或数据库服务器。

[0125] 通过简单地轻敲或点击在显示区 42 中显示的留言 27 可以由用户选择该留言 27。

[0126] 命令条 43 包含多个按键 49 和 50，通过由用户按下按键的基本操作可以进行用户界面 37 的控制和所显示的消息来源 35 的管理和处理。

[0127] 在图 3 中示出的版本中，应用 31 对于在智能电话 12 上的显示是优化的。在此在智能电话 12 上对于每个时间点始终仅显示具有单个消息来源 35 的单个 GUI 页 36。但是可以彼此并行地生成 GUI 页 36 的多个实例，其中这些实例在智能电话 12 上在显示器后台并行地运行。由此尤其可以彼此并行地处理多个搜索查询 Q。GUI 页 36 在通用性的框架 51（或容器）中运行。该框架 51（和由此的整个应用 31）例如可以实施为单独的应用或用于 Web 浏览器的插件程序。然而替换地，应用 31 也可以被构造为在 Web 浏览器中直接运行，例如作为 HTML5 应用。

[0128] 命令条 43 的按键 50 在此能够实现在运行的 GUI 页 36 之间的变换，方法是在按下按键 50 时将所显示的 GUI 页 36 从用户界面 37 淡出并且显示另外的 GUI 页 36。

[0129] 命令行 43 的另外的按键 49 尤其与以下 CRUD 功能对应：

[0130] - “创建”：按下相应的按键 49 生成用户界面 37 的新的 GUI 页 36。

[0131] - “删除”：按下相应的按键 49 结束用户界面 37 的所显示的 GUI 页 36。

[0132] - “读取”：按下相应的按键 49 打开在图 4 中示意性示出的用户界面 37 的 GUI 页 52，在该 GUI 页 52 中显示了所选择的留言 27 的数据内容，仅用于查看或用于改变。

[0133] 按照图 4，GUI 页 52 同样包括四个区域，即，页眉 60、标题区 61、显示区 62 以及命令条 63。

[0134] 页眉 60 在此类似于 GUI 页 36 的页眉 40 具有对应的中心 22-24 和设施 2。标题区 61 标出了所显示的数据组的类型并且由此在应用 31 的情况下例如包含词语“患者”（或“患者实例”）。

[0135] 在显示区 62 中显示所选择的留言 27 的内容，在应用 31 的情况下由此显示（患者）数据组的内容。在这种情况下在显示区 62 中特别地将患者数据组的字段名 67（例如“名字”、“姓氏”、“患者 ID”、“年龄”、...）连同各自的字段内容 68（也就是是例如具体的名字、姓氏、具体的患者 ID 或患者的具体年龄）一起显示。

[0136] 命令行 63 包含按键 64，特别地为其分配如下 CRUD 功能：

[0137] - “创建”：按下相应的按键 64 生成 GUI 页 52 的新的实例，用于建立新的留言 27

和由此新的（患者）数据组 P。

[0138] -“更新”:按下相应的按键 64 存储可能改变的字段内容 68,方法是应用 31 直接在患者中心 22 的表存储 26 中以改变的说明覆盖所显示的留言 27。

[0139] -“删除”:按下相应的按键 64 通过应用 31 直接在患者中心 22 的表存储器 26 中删除所显示的留言 27 ;关闭所显示的 GUI 页 52。

[0140] 此外,命令行 63 包含按键 65,按下该按键,应用 31 在不改变留言 27 的情况下关闭所显示的 GUI 页 52 并且重新显示基本 GUI 页 36。

[0141] 命令行 43 和 63 可以具有用于命令的其它按键。同样也可以给智能电话 12 的可能存在的电子机械按键中的一个分配一个或多个按照按键 49、50、64 和 65 描述的功能。在该情况下各个按键优选从命令行 43 移除。

[0142] 应用 32 和 33 同样是浏览器应用,其关于其结构和功能相应于应用 31。特别地,应用 32 和 33 的各自的用户界面 37 也分别具有按照图 3 和图 4 描述的 GUI 页 36 和 52。

[0143] 但是应用 32 和 33 访问工作列表中心 23 或上下文中心 24,并且相应地匹配。由此特别地

[0144] -在应用 32 的范畴内显示的留言 27 包含关于特定的任务 W 的说明（例如关于任务的设置日期、委托任务的部门的说明、以及关于任务本身和该任务可能包含在其中的工作流程的说明）,和

[0145] -在应用 33 的范畴内显示的留言 27 分别包含上下文数据组 C 的内容,特别是关于文件类型、存储位置、数据格式以及用于图像数据组 B 的可选拍摄参数的说明。

[0146] 在（未详细示出的）替换的实施中,将两个或甚至所有三个应用 31 至 33 综合为一个全面应用。该全面应用包括共同的框架 51,在该框架 51 内运行多个 GUI 页 36 和 / 或 52,其与不同的资源中心对应并且由此用于显示不同数据类型的消息来源 35 或留言 37。在应用 31-33 中特别地这样实施框架 51,使得其可以利用一个或多个单个的浏览应用 31-33 来配置。如在附录中的代码示例所示出的那样,框架 51 和浏览应用 31-33 优选在唯一的装置应用中作为容器 App(Container-App) 和子 App(Sub-App) 实现,其也可以在 Web 浏览器中作为 HTML5 应用运行。

[0147] 如前面描述的,这样实施在每个中心 22-25 和分别分配的应用 31-34 之间的通信,使得其与在因特网数据交换中常用的 REST(Representational State Transfer,表述性状态转移)原理兼容。特别地,在中心 22-25 和各自的应用 31-34 之间交换的数据满足标准 XML、JSON 和 ATOM。图像数据 B 尤其按照 JPEG 格式存储在图像中心 25 中,体积数据优选按照 DICOM 格式存储。必要时从公共云 13 向应用 34 提供的体积数据的视图 V(场景图)优选按照 JPEG 格式建立。但附加地或替换地,也可以在图像中心 25 中以其它数据格式,例如 webm、mp4、pdf 或 dz 格式存储图像数据。

[0148] 为此每个应用 31-33 被构造为,按照用户的指示将消息来源 35 从相关的中心 22-24 具体化到装置 8 的本地存储器,从而该消息来源 35 和在其中包含的留言 27 通过应用 31-33 在没有与公共云 13 网络连接的情况下也可以被显示和 / 或处理。此外,每个应用 31-33 被构造为,按照用户的相应命令将具体化的消息来源 35 或留言 27 传送到相同或另外的应用 31-34 的另外的实例。对于消息来源 35 或留言 27 的设备内部的传送,在装置 8 的存储器内构造应用 31-34 的所有实例共同访问的、被称为“交换板 (Exchangeboard)”的本

地存储区。为了通过交换板交换,将待交换的消息来源 35 或留言 27 在 XML/JSON 文档中串行化,其存储在该装置 8 的局部存储器中并且然后又被解串。为了在不同的装置 8 之间交换消息来源 35 或留言 27,应用 31-34 优选具有与推特或 E-Mail 程序的接口。

[0149] 图 5 示出了应用 31-33 的方案,其中用户界面 37 对于在大版式的显示屏,例如个人计算机 9 的显示屏 10 或平板电脑 11 和显示屏上的显示是优化的。按照图 5 的用户界面 37 与在图 3 和图 4 中示出的用户界面 37 基本上如下地区分,即,并排地显示并且由此可以同时看到用户界面 37 的 GUI 页 36 和 / 或 52 的多个实例。

[0150] 在按照图 5 的示意图中,用户界面 37 还具有附加的导航按键 50,通过按下该导航按键又可以在所显示的 GUI 页 36 和 / 或 52 之间翻阅。

[0151] 此外,用户界面 37 附加地具有菜单行 80,通过按下该菜单行可以在不同的应用 31-34 之间交换。菜单行 80 在此尤其是框架 81 (或容器) 的组成部分,在其中运行应用 31-34 的存在的实例。

[0152] 图 6 示意性示出了应用 31-33 中的一个的组件架构的模型。从示意图可以得出,应用 31-33 由按照五个层布置的组件构成。

[0153] 该架构的最高级的层被称为框架层 90。该框架层 90 在按照图 6 的示例中通过框架组件 91 构成,其实施为框架 51 或 81。如由图 3 至图 5 明显示出的,通过框架 51 或 81 (并且由此通过框架组件 91) 规定,同时显示多少 GUI 页 36 或 52、这些 GUI 页 36、52 怎样布置在显示屏面积上以及怎样控制在 GUI 页 36 和 52 的并行存在的实例之间的转移,也就是对于用户交互怎样激活和禁止以及在显示屏上淡入和淡出这些实例。

[0154] 框架组件 91 特殊地依据在其上应当运行应用 31-33 的装置 8 的类型来选择,并且按照各个装置 8 的形状因数 (也就是显示屏的大小和几何形状) 以及按照在装置 8 上存在的用于用户交互的部件而裁剪。例如为了在平板电脑 11 或智能电话 12 上运行而设置的框架组件 91 版本支持用户交互方法,如其对于通过触摸屏的用户交互常用的那样 (例如通过在显示屏上手动拖动而移动 GUI 页 36 和 52、通过两指操作放大缩小图像信息、在倾斜装置 8 时旋转图像信息等)。相反,对于在个人计算机 9 中运行而构造的框架组件 91 版本优选按照与用户交互的匹配借助鼠标或触摸板提供以按键 (按钮) 形式的图形控制元件。

[0155] 在框架层 90 下级是视图层 92。该视图层 92 在应用 31-33 的情况下包括两个组件 93 和 94,其定义了“视图”,即 GUI 页 37 的图形结构。组件 93 在此关于类型、外观和布置定义了 GUI 页 36 的控制元件。组件 94 相应地定义了 GUI 页 52 的控制元件。

[0156] 组件 93 和 94 特定于待显示的数据组的类型而设计,也就是在应用 31 的情况下针对患者数据组 P 设计。应用 31 的组件 93 和 94 就此而言与应用 32 和 33 的相应组件 93 和 94 不同地构造。只要应用 (与图 6 不同地) 被构造为用于显示多个数据类型,则应用在视图层 92 中对于每个待显示的数据类型分别包括两个组件,其中一个定义了用于显示相关的消息来源 35 的 GUI 页 36,而另一个定义了用于显示相关的留言 27 的 GUI 页 52。

[0157] 在视图层 92 之下应用 31-33 在视图模型层 95 中包括两个另外的组件 96 和 97。这些组件 96 和 97 的每一个对于视图层 92 的分别对应的组件 93 或 94 定义了用于在该组件 93 或 94 中定义的图形用户界面 37 的控制元件的特征和命令,特别是上面描述的 CRUD 功能。视图模型层 95 的组件 96 和 97 在此通用性地,也就是不取决于待显示的数据类型而构造。由此组件 96 和 97 在所有应用 31、32 和 33 中也是相同的。

[0158] 在视图模型层 95 之下应用 31-33 包括模型层 98。该模型层 98 包括组件 99, 该组件 99 定义了待显示的数据组的结构, 也就是在应用 31 的情况下是患者数据组 P 的结构。组件 99 在此尤其包含用于各个数据组的变量的变量定义, 在患者数据组 P 的情况下例如是“姓氏”、“名字”等。

[0159] 又在模型层 98 之下布置驱动器层 100。驱动器层 100 在此包括一个或多个驱动器组件 101, 其对于上级设置的层促成对在预先给定的数据存储器 102 中存储的数据组和对应的消息来源 35 和留言 27 的访问。

[0160] 根据驱动器组件 101 的构造, 数据存储器 102 是装置 8 的本地存储器、表存储器 26 或对应的中心 22-24 的另外的存储器或其它外部存储器。在推广的意义上, 数据存储器 102 也可以是 E-Mail 或 SMS 接收器、或社交网络, 特别是推特。应用 31-33 在此优选包括多个不同类型的驱动器组件 101, 其分别与特定的数据存储器 102 对应。

[0161] 所有驱动器组件 101 都基于共同的、规定的驱动器 API 103。驱动器组件 101 同样是通用性的, 也就是不取决于待显示的数据类型。

[0162] 在驱动器组件 101 或每个驱动器组件 101 与上级层之间的通信在此异步地进行。这意味着, 上级层的组件在向驱动器组件 101 或驱动器组件 101 中的一个查询之后不等待驱动器组件 101 的响应, 从而在访问数据组时可能的延迟不造成应用 31 的运行的负担。

[0163] 如上面描述的那样, 可以并行地运行 GUI 页 36 和 52 的多个实例。相应地如在图 7 中示出的那样, 也可以多次实例化并且并行地运行视图层 92、视图模型层 95 和模型层 98 的组件 93、94、96、97、99。也可以多次实例化驱动器层 100 的驱动器组件 101, 从而将 GUI 页 36 或 52 的每个实例与驱动器组件 101 的相关的实例对应。驱动器实例并行地运行并且可以(如上面提到的)访问相同的数据存储器 102 或访问不同的数据存储器 102。

[0164] 本发明在上面描述的实施例中是特别清楚的, 虽然如此但本发明不限于该实施例。而是可以根据说明书和权利要求导出本发明的另外的实施方式。特别地结合实施例描述的本发明的特征也可以以其它方式组合, 而不偏离本发明。

[0165] 在如下附录中实现关于应用模板、驱动器 API 103 和视图模型层 95 的组件 97 的具体实施的示例。

[0166] 附录 1 : 应用模板的示例性实施

[0167]

```
//-----  
// AppConfig.xml  
//-----  
  
<?xml version="1.0" encoding="utf-8" ?>  
<Apps>  
  <!------->  
  <!-- Application 1 -->  
  <!------->  
  <App>  
    <A_AppTitle>patient hub      SINGAPORE@W8  
kd-24-12-2011</A_AppTitle>  
  
    <A_PanoramaBackgroundImaUri>http://singapore4.blob.core.windows.net/  
mysingapore/thumbs/syngo.via_1.jpg</A_PanoramaBackgroundImaUri>  
  
    <A_NormalMainPage>NormalMainPage1.xaml</A_NormalMainPage>  
    <A_PivotMainPage>PivotMainPage1.xaml</A_PivotMainPage>  
  
    <A_PanoramaMainPage>PanoramaMainPage1</A_PanoramaMainPage>  
    <A_MainPage>MainPage1.xaml</A_MainPage>  
    <AppFeeds>  
      <!-- AppFeed 1 -->  
      <AppFeed>  
        <A_Name>Patient</A_Name>  
        <V_Name>PatientView1</V_Name>  
        <V_UserControlGuiName>all</V_UserControlGuiName>  
        <V_SearchString></V_SearchString>  
        <V_NextAppNavUri>Application 2</V_NextAppNavUri>
```

[0168]

```
<V_PushNotificationChanName>App1-Patient-Notif Channel
    </V_PushNotificationChanName>
<V_ListView>PatientView.xaml</V_ListView>
<V_ListItemView>PatientItemView.xaml</V_ListItemView>
<M_Model>Patient</M_Model>
<D_Account>singapore4</D_Account>
<D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
<D_Table>patients</D_Table>
<D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 2 -->
<AppFeed>
    <A_Name>Patient</A_Name>
    <V_Name>PatientView2</V_Name>
    <V_UserControlGuiName>scheduled</V_UserControlGuiName>
    <V_SearchString></V_SearchString>
    <V_NextAppNavUri>Application 2</V_NextAppNavUri>
    <V_PushNotificationChanName>App1-Patient-Notif Channel
        </V_PushNotificationChanName>
    <V_ListView>PatientView.xaml</V_ListView>
    <V_ListItemView>PatientItemView.xaml</V_ListItemView>
    <M_Model>Patient</M_Model>
    <D_Account>singapore4</D_Account>
    <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
    <D_Table>patients</D_Table>
    <D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 3 -->
<AppFeed>
    <A_Name>Patient</A_Name>
    <V_Name>PatientView3</V_Name>
```

[0169]

```

    <V_UserControlGuiName>billed</V_UserControlGuiName>
    <V_SearchString></V_SearchString>
    <V_NextAppNavUri>Application 2</V_NextAppNavUri>
    <V_PushNotificationChanName>App1-Patient-Notif Channel
        </V_PushNotificationChanName>
    <V_ListView>PatientView.xaml</V_ListView>
    <V_ListItemView>PatientItemView.xaml</V_ListItemView>
    <M_Model>Patient</M_Model>
    <D_Account>singapore4</D_Account>
    <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
    <D_Table>patients</D_Table>
    <D_Type>AzureTableDriver</D_Type>
</AppFeed>
</AppFeeds>
</App>
<!------->
<!-- Application 2 -->
<!------->
<App>
    <A_AppTitle>worklist hub        GLOBAL SHARED WORKLIST
</A_AppTitle>

    <A_PanoramaBackgroundImaUri>http://singapore4.blob.core.windows.net/
        mysingapore/thumbs/syngo.via_1.jpg</A_PanoramaBackgroundImaUri>

    <A_NormalMainPage>NormalMainPage2.xaml</A_NormalMainPage>
        <A_PivotMainPage>PivotMainPage2.xaml</A_PivotMainPage>

    <A_PanoramaMainPage>PanoramaMainPage2</A_PanoramaMainPage>
        <A_MainPage>MainPage2.xaml</A_MainPage>
    <AppFeeds>

```

[0170]

```
<!-- AppFeed 2 -->
<AppFeed>
  <A_Name>Worklist</A_Name>
  <V_Name>WorklistView1</V_Name>
  <V_UserControlGuiName>all</V_UserControlGuiName>
  <V_SearchString></V_SearchString>
  <V_NextAppNavUri>Application 3</V_NextAppNavUri>
  <V_PushNotificationChanName>App2-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>WorklistView.xaml</V_ListView>
  <V_ListItemView>WorklistItemView.xaml</V_ListItemView>
  <M_Model>Worklist</M_Model>
  <D_Account>singapore4</D_Account>
  <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
  <D_Table>workitems</D_Table>
  <D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 2 -->
<AppFeed>
  <A_Name>Worklist</A_Name>
  <V_Name>WorklistView2</V_Name>
  <V_UserControlGuiName>unread</V_UserControlGuiName>
  <V_SearchString></V_SearchString>
  <V_NextAppNavUri>Application 3</V_NextAppNavUri>
  <V_PushNotificationChanName>App2-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>WorklistView.xaml</V_ListView>
  <V_ListItemView>WorklistItemView.xaml</V_ListItemView>
  <M_Model>Worklist</M_Model>
  <D_Account>singapore4</D_Account>
  <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
```

[0171]

```

        <D_Table>workitems</D_Table>
        <D_Type>AzureTableDriver</D_Type>
    </AppFeed>
    <!-- AppFeed 3 -->
    <AppFeed>
        <A_Name>Worklist</A_Name>
        <V_Name>WorklistView3</V_Name>
        <V_UserControlGuiName>read</V_UserControlGuiName>
        <V_SearchString></V_SearchString>
        <V_NextAppNavUri>Application 3</V_NextAppNavUri>
        <V_PushNotificationChanName>App2-Patient-Notif Channel
            </V_PushNotificationChanName>
        <V_ListView>WorklistView.xaml</V_ListView>
        <V_ListItemView>WorklistItemView.xaml</V_ListItemView>
        <M_Model>Worklist</M_Model>
        <D_Account>singapore4</D_Account>
        <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
        <D_Table>workitems</D_Table>
        <D_Type>AzureTableDriver</D_Type>
    </AppFeed>
</AppFeeds>
</App>
<!------->
<!-- Application 3 -->
<!------->
<App>
    <A_AppTitle>image hub
CONTEXT
FOLDER </A_AppTitle>

    <A_PanoramaBackgroundImaUri>http://singapore4.blob.core.windows.net/
    mysingapore/thumbs/syngo.via_1.jpg</A_PanoramaBackgroundImaUri>

```

[0172]

```
<A_NormalMainPage>NormalMainPage3.xaml</A_NormalMainPage>
  <A_PivotMainPage>PivotMainPage3.xaml</A_PivotMainPage>

<A_PanoramaMainPage>PanoramaMainPage3</A_PanoramaMainPage>
  <A_MainPage>MainPage3.xaml</A_MainPage>
  <AppFeeds>
    <!-- AppFeed 1 -->
    <AppFeed>
      <A_Name>Context</A_Name>
      <V_Name>ContextView1</V_Name>
      <V_UserControlGuiName>head</V_UserControlGuiName>
      <V_SearchString>ARegion eq 'Head'</V_SearchString>
      <V_NextAppNavUri>Application 4</V_NextAppNavUri>
      <V_PushNotificationChanName>App3-Patient-Notif Channel
        </V_PushNotificationChanName>
      <V_ListView>ContextView.xaml</V_ListView>
      <V_ListItemView>ContextItemView.xaml</V_ListItemView>
      <M_Model>Context</M_Model>
      <D_Account>singapore4</D_Account>
      <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
      <D_Table>contextfolder</D_Table>
      <D_Type>AzureTableDriver</D_Type>
    </AppFeed>
    <!-- AppFeed 2 -->
    <AppFeed>
      <A_Name>Context</A_Name>
      <V_Name>ContextView2</V_Name>
      <V_UserControlGuiName>thorax</V_UserControlGuiName>
      <V_SearchString>ARegion eq 'Thorax'</V_SearchString>
      <V_NextAppNavUri>Application 4</V_NextAppNavUri>
```

[0173]

```
<V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
<V_ListView>ContextView.xaml</V_ListView>
<V_ListItemView>ContextItemView.xaml</V_ListItemView>
<M_Model>Context</M_Model>
<D_Account>singapore4</D_Account>
<D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
<D_Table>contextfolder</D_Table>
<D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 3 -->
<AppFeed>
    <A_Name>Context</A_Name>
    <V_Name>ContextView3</V_Name>
    <V_UserControlGuiName>body</V_UserControlGuiName>
    <V_SearchString>ARegion eq 'WholeBody'</V_SearchString>
    <V_NextAppNavUri>Application 4</V_NextAppNavUri>
    <V_PushNotificationChanName>App3-Patient-Notif Channel
        </V_PushNotificationChanName>
    <V_ListView>ContextView.xaml</V_ListView>
    <V_ListItemView>ContextItemView.xaml</V_ListItemView>
    <M_Model>Context</M_Model>
    <D_Account>singapore4</D_Account>
    <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
    <D_Table>contextfolder</D_Table>
    <D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 4 -->
<AppFeed>
    <A_Name>Context</A_Name>
    <V_Name>ContextView4</V_Name>
```

[0174]

```
<V_UserControlGuiName>key ima</V_UserControlGuiName>
<V_SearchString>ARegion eq 'KeyImage'</V_SearchString>
<V_NextAppNavUri>Application 4</V_NextAppNavUri>
<V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
<V_ListView>ContextView.xaml</V_ListView>
<V_ListItemView>ContextItemView.xaml</V_ListItemView>
<M_Model>Context</M_Model>
<D_Account>singapore4</D_Account>
<D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
<D_Table>contextfolder</D_Table>
<D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 5 -->
<AppFeed>
    <A_Name>Context</A_Name>
    <V_Name>ContextView5</V_Name>
    <V_UserControlGuiName>photo</V_UserControlGuiName>
    <V_SearchString>ARegion eq 'Photo'</V_SearchString>
    <V_NextAppNavUri>Application 4</V_NextAppNavUri>
    <V_PushNotificationChanName>App3-Patient-Notif Channel
        </V_PushNotificationChanName>
    <V_ListView>ContextView.xaml</V_ListView>
    <V_ListItemView>ContextItemView.xaml</V_ListItemView>
    <M_Model>Context</M_Model>
    <D_Account>singapore4</D_Account>
    <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
    <D_Table>contextfolder</D_Table>
    <D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 6 -->
```

[0175]

```

<AppFeed>
  <A_Name>Context</A_Name>
  <V_Name>ContextView6</V_Name>
  <V_UserControlGuiName>video</V_UserControlGuiName>
  <V_SearchString>ARegion eq 'Media'</V_SearchString>
  <V_NextAppNavUri>Application 4</V_NextAppNavUri>
  <V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>ContextView.xaml</V_ListView>
  <V_ListItemView>ContextItemView.xaml</V_ListItemView>
  <M_Model>Context</M_Model>
  <D_Account>singapore4</D_Account>
  <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
  <D_Table>contextfolder</D_Table>
  <D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 7 -->
<AppFeed>
  <A_Name>Context</A_Name>
  <V_Name>ContextView7</V_Name>

```

```

<V_UserControlGuiName>documents</V_UserControlGuiName>
  <V_SearchString>ARegion eq 'Document'</V_SearchString>
  <V_NextAppNavUri>Application 4</V_NextAppNavUri>
  <V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>ContextView.xaml</V_ListView>
  <V_ListItemView>ContextItemView.xaml</V_ListItemView>
  <M_Model>Context</M_Model>
  <D_Account>singapore4</D_Account>
  <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>

```

[0176]

```
<D_Table>contextfolder</D_Table>
<D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 8 -->
<AppFeed>
  <A_Name>Context</A_Name>
  <V_Name>ContextView8</V_Name>
  <V_UserControlGuiName>all</V_UserControlGuiName>
  <V_SearchString></V_SearchString>
  <V_NextAppNavUri>Application 4</V_NextAppNavUri>
  <V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>ContextView.xaml</V_ListView>
  <V_ListItemView>ContextItemView.xaml</V_ListItemView>
  <M_Model>Context</M_Model>
  <D_Account>singapore4</D_Account>
  <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
  <D_Table>contextfolder</D_Table>
  <D_Type>AzureTableDriver</D_Type>
</AppFeed>
<!-- AppFeed 9 -->
<AppFeed>
  <A_Name>Context</A_Name>
  <V_Name>ContextView9</V_Name>
  <V_UserControlGuiName>files</V_UserControlGuiName>
  <V_SearchString>File eq "</V_SearchString>
  <V_NextAppNavUri>Application 4</V_NextAppNavUri>
  <V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>FileView.xaml</V_ListView>
  <V_ListItemView>FileItemView.xaml</V_ListItemView>
```

[0177]

```
<M_Model>File</M_Model>
<D_Account>singapore4</D_Account>
<D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
<D_Table></D_Table>
<D_Type>IsoDeviceDriver</D_Type>
</AppFeed>
<!-- AppFeed 10 -->
<AppFeed>
  <A_Name>Context</A_Name>
  <V_Name>ContextView10</V_Name>
  <V_UserControlGuiName>blobs</V_UserControlGuiName>
  <V_SearchString>blob eq 'config'</V_SearchString>
  <V_NextAppNavUri>Application 4</V_NextAppNavUri>
  <V_PushNotificationChanName>App3-Patient-Notif Channel
    </V_PushNotificationChanName>
  <V_ListView>BlobView.xaml</V_ListView>
  <V_ListItemView>BlobItemView.xaml</V_ListItemView>
  <M_Model>Blob</M_Model>
  <D_Account>singapore4</D_Account>
  <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
  <D_Table>config</D_Table>
  <D_Type>AzureBlobDriver</D_Type>
</AppFeed>
<!-- AppFeed 11 -->
<AppFeed>
  <A_Name>Context</A_Name>
  <V_Name>ContextView11</V_Name>
  <V_UserControlGuiName>queues</V_UserControlGuiName>
  <V_SearchString>Queue eq 'modworklist'</V_SearchString>
  <V_NextAppNavUri>Application 4</V_NextAppNavUri>
  <V_PushNotificationChanName>App3-Patient-Notif Channel
```

[0178]

```
        </V_PushNotificationChanName>
    <V_ListView>QueueView.xaml</V_ListView>
    <V_ListItemView>QueueItemView.xaml</V_ListItemView>
    <M_Model>QueueMessage</M_Model>
    <D_Account>singapore4</D_Account>
    <D_Key>s0Zh73EV6WqkQp9CjATic7vEYC4k+ .... </D_Key>
    <D_Table>modworklist</D_Table>
    <D_Type>AzureQueueDriver</D_Type>
</AppFeed>
</AppFeeds>
</App>
</Apps>
```

[0179] 附录 2 :根据图 6 的驱动器 API 的驱动器 API103 的示例性实施

[0180]

```
//-----  
// R-MVVM-D – Model and the Driver-API  
//-----  
  
public class DriverBase<T> : DependencyObject,  
INotifyPropertyChanged  
{  
    public event PropertyChangedEventHandler PropertyChanged;  
    protected void OnNotifyPropertyChanged(string p)  
    {  
        if (PropertyChanged != null)  
        {  
            PropertyChanged(this, new  
PropertyChangedEventArgs(p));  
        }  
    }  
    public delegate void DriverNotificationEventHandler(object  
sender, EvtArg e);  
    public virtual string Name { get; set; }  
    public virtual string Account { get; set; }  
    public virtual string Key { get; set; }  
    public virtual string Table { get; set; }  
    public virtual IStorageCredentials Creds { get; set; }  
    public virtual CloudTableClient Client { get; set; }  
    public virtual TableServiceContext Ctx { get; set; }  
    public virtual T SelectedItem { get; set; }  
    public virtual int SelectedIndex { get; set; }  
    public virtual string State { get; set; }  
    public virtual string SearchText { get; set; }
```

[0181]

```

        public virtual string SelectText { get; set; }
        public virtual string BindingItemSource { get; set; }
        public virtual string ItemType { get;
set; }/"Object","File","Blob","Msg"
        public virtual int AppIndex { get; set; } // normal/pivot/panorama
app index
        public virtual DataServiceCollection<T> ListItems { get; set; }
        public virtual ObservableCollection<ListItemViewModel<T>>
ListItemsNonOdata
        {
        get; set;
        }
        public virtual void GeneralModelNotifier(string reason, string
param) { }
        public virtual void init(DriverNotificationEventHandler han) { }
        public virtual void exit(DriverNotificationEventHandler han) { }
        public virtual void initTable() { }
        public virtual void exitTable() { }
        public virtual void BeginCreateTable() { }
        public virtual void BeginDeleteTable() { }
        public virtual void BeginReadTable() { }
        public virtual void BeginReadEntity() { }
        public virtual void BeginAddEntity(T list) { }
        public virtual void BeginUpdateEntity(T list, T oldList) { }
        public virtual void BeginRemoveEntity(T list) { }
    }

```

[0182] 附录 3 :通用性的列表项视图模型（根据图 6 的视图模型层 95 的组件 97 的示例性实施）

[0183]

```
#define GoBackModeInsteadOfNavigate
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Input;
using System.Windows.Controls;
using Microsoft.Samples.WindowsPhoneCloud.StorageClient;
using Microsoft.Samples.Data.Services.Common;
using System.Windows.Navigation;
using Microsoft.Phone.Shell;
using System.Globalization;
using System.Windows.Documents;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.IO;

namespace AzureTableStore_CRUD_Sample
{
    public class ListItemViewModel<T> : ViewModelBase
    {
        private string _bindingItemSource;
        private string _name;
        private Object _model;
```

[0184]

```
private Object _oldmodel;
private T _dataModel;
private string _appTitle;
private string _account;
private string _table;
private int _selectedIndex;
private string _lastExportedIsoEntityUriString;
private string _lastExportedAzureCloudEntityUriString;
private string _lastExportedWebEntityUriString;
UploadFile pUploader;
```

```
public ListViewItemModel()
{
}
```

```
#if DataModelInheritanceUsed
```

```
public ListViewItemModel(long p) : base(p)
{
}
```

```
public ListViewItemModel(bool set) : base(set)
{
}
```

```
public ListViewItemModel(ICloudBlobContainer o)
    : base(o)
{
}
```

```
#else
#endif
```

[0185]

```
public string BindingItemSource
{
    get { return _bindingItemSource; }
    set { _bindingItemSource = value; }
}
```

```
public string Name
{
    get { return _name; }
    set { _name = value; }
}
```

```
public string Account
{
    get { return _account; }
    set { _account = value; }
}
```

```
public string Table
{
    get { return _table; }
    set { _table = value; }
}
```

```
public int SelectedIndex
{
    get { return _selectedIndex; }
    set { _selectedIndex = value; }
}
```

[0186]

```
public Object Model
{
    get { return _model; }
    set { _model = value; }
}

public Object oldModel
{
    get { return _oldmodel; }
    set { _oldmodel = value; }
}

public T DataModel
{
    get { return _dataModel; }
    set { _dataModel = value; }
}

public string AppTitle
{
    get { return _appTitle; }
    set { _appTitle = value; }
}

public event EventHandler<NavigationEventArgs> Navigate;

public void NavigateTo(Uri navigationUri)
{
    this.RaiseNavigate(navigationUri);
}
```

[0187]

```
private void RaiseNavigate(Uri uri)
{
    var navigate = this.Navigate;
    if (navigate != null)
    {
        navigate(this, new NavigationEventArgs(null, uri));
    }
}

private void
ApplicationBarIconButton_CANCEL_INPUT_Click()
{
    SetApplicationState("CurrentResponse", "Cancel");
    SetApplicationState("CurrentViewModel", this.Model);

    #if GoBackModeInsteadOfNavigate1
        (Application.Current.RootVisual as
PhoneApplicationFrame).GoBack();
    #else
        NavigateBackToCaller();
    #endif
}

private void ApplicationBarIconButton_SAVE_ENTITY_Click()
{
    SetApplicationState("CurrentResponse", "Save");

    object focusObj = FocusManager.GetFocusedElement();
    if (focusObj != null && focusObj is TextBox)
    {
        var binding = (focusObj as
```

[0188]

```
        TextBox).GetBindingExpression(TextBox.TextProperty);
            binding.UpdateSource();
        }

        SetApplicationState("CurrentViewModel", this.Model);
        SetApplicationState("OldViewModel", this.oldModel);

        #if GoBackModeInsteadOfNavigate1
            (Application.Current.RootVisual as
PhoneApplicationFrame).GoBack();
        #else
            NavigateBackToCaller();
        #endif
    }

    private void
ApplicationBarIconButton_REMOVE_ENTITY_Click()
    {
        SetApplicationState("CurrentResponse", "Delete");
        SetApplicationState("CurrentViewModel", this.Model);

        #if GoBackModeInsteadOfNavigate1
            (Application.Current.RootVisual as
PhoneApplicationFrame).GoBack();
        #else
            NavigateBackToCaller();
        #endif
    }

    public void old_ApplicationBarMenueButton_EXPORT_Click()
```

[0189]

```
{
    MessageBox.Show("Serializing data model to string ...");
    string data = "";

    data =
DataContractSerializerHelpers.SerializeToString(this.Model);

    MessageBox.Show("Serializing done! Data=" + data);
    switch (App.DstModel)
    {
        case "IsolatedStorage":
            string StorageFile = Account + "_" + Table + "_" +
SelectedIndex + ".tweet";
            MessageBox.Show("Exporting data model to
app-dir in iso storage, filename=" + StorageFile + " ...");
            IsoImportExport.WriteStringToIsoFile(data,
StorageFile);

            break;
        case "FileSystemStorage":
            break;
        case "AzureBlobStorage":
            break;
        case "WebUriStorage":
            break;
        default:
            break;
    }
    MessageBox.Show("Exporting done!");
}

public void old_ApplicationBarMenueButton_IMPORT_Click()
```

[0190]

```
{
    string data = "";
    switch (App.SrcModel)
    {
        case "IsolatedStorage":
            string StorageFile = Account + "_" + Table + "_" +
SelectedIndex + ".tweet";
            MessageBox.Show("Importing data model to string
from app-dir in iso storage, filename=" + StorageFile + " ...");
            data =
IsoImportExport.ReadStringFromIsoFile(StorageFile);
            break;
        case "FileSystemStorage":
            break;
        case "AzureBlobStorage":
            break;
        case "WebUriStorage":
            break;
        default:
            break;
    }
    MessageBox.Show("Importing done!");
    MessageBox.Show("Deserializing data model from
string ...");

    var tweet =
DataContractSerializerHelpers.DeserializeFromString<T>(data);

    MessageBox.Show("De-Serializing done! Data=" + data);
    MessageBox.Show("Copy data  from tweet to  this.Model
as <T> ...");
```

[0191]

```
        this.Model = tweet;

        MessageBox.Show("Copy done, this.Model as <T> has been  
filled!");

        MessageBox.Show("Deserializing done!");
    }
}
```

```
//-----  
// Import-Tweet from external storage  
  
//-----
```

```
public void ApplicationBarMenueButton_IMPORT_Click()
{
    switch (App.SrcModel)
    {
        case "IsolatedStorage":
            ImportFromIsoEntityUpdate();
            break;
        case "AzureBlobStorage":
            ImportFromAzureCloudEntityUpdate();
            break;
        case "WebUriStorage":
            ImportFromWebEntityUpdate();
            break;
        default:
            break;
    }
}
```

[0192]

```
public void ImportFromIsoEntityUpdate()
{
    int index = SelectedIndex;
    string StorageType = "IsolatedStorage";
    string StorageHostName = "";
    string StorageContainerName = "";
    string StoragePathName = "";
    string StorageFileName = Account + "_" + Table + ".tweet";

    ImportEntityUpdate(index, StorageType, StorageHostName,
StorageContainerName, StoragePathName, StorageFileName);
}

public void ImportFromAzureCloudEntityUpdate()
{
    int index = SelectedIndex;
    string StorageType = "AzureBlobStorage";
    string StorageHostName = Account;
    string StorageContainerName = App.SrcContTabQueOrDir;
    string StoragePathName = "";
    string StorageFileName = Account + "_" + Table + ".tweet";

    ImportEntityUpdate(index, StorageType, StorageHostName,
StorageContainerName, StoragePathName, StorageFileName);
}

public void ImportFromWebEntityUpdate()
{
    int index = SelectedIndex;
    string StorageType = "WebUriStorage";
```

[0193]

```
string StorageHostName = App.SrcAccountOrHost;
string StorageContainerName = App.SrcContTabQueOrDir;
string StoragePathName = "";
string StorageFileName = Account + "_" + Table + ".tweet";

ImportEntityUpdate(index, StorageType, StorageHostName,
StorageContainerName, StoragePathName, StorageFileName);
}

public void ImportEntityUpdate(int index, string StorageType,
string StorageHostName, string StorageContainerName, string StoragePathName,
string StorageFileName)
{
    string data = "";
    string fnam = StorageHostName + StorageContainerName +
StoragePathName + StorageFileName;
    switch (StorageType)
    {
        case "IsolatedStorage":
            data =
IsoImportExport.ReadStringFromIsoFile(fnam);
            break;
        case "FileSystemStorage":
            MessageBox.Show("FileSystemStorage API is not
implemented, maybe you should use IsolatedStorage anyway!");
            break;
        case "AzureBlobStorage":
            string url = "http://" + Account +
".blob.core.windows.net" + "/" + StorageContainerName + "/" +
StoragePathName + StorageFileName;
            WebClient client = new WebClient();
```

[0194]

```
        client.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(client_EntityUpdateDownloadStringCo
mpleted);

        client.DownloadStringAsync(new Uri(url,
UriKind.Absolute), fnam);

        return;

        case "WebUriStorage":
            string url1 = "http://" + StorageHostName + "/" +
StorageContainerName + "/" + StorageFileName;
            WebClient client1 = new WebClient();
            client1.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(client_EntityUpdateDownloadStringCo
mpleted);

            client1.DownloadStringAsync(new Uri(url1,
UriKind.Absolute), fnam);

            return;

        default:
            break;
    }

    var tweet =
DataContractSerializerHelpers.DeserializeFromString<T>(data);
    this.Model = tweet;
}

void client_EntityUpdateDownloadStringCompleted(object sender,
DownloadStringCompletedEventArgs e)
{
    if (e.Error == null)
    {
        var tweet =
```

[0195]

```
DataContractSerializerHelpers.DeserializeFromString<T>(e.Result);
    this.Model = tweet;
    IsoImportExport.WriteStringToIsoFile(tweet as string,
e.UserState as string);
    }
    else
    {
        MessageBox.Show(e.Error.ToString());
    }
}

//-----
// Export-Tweet to external storage
//-----

public void ApplicationBarMenueButton_EXPORT_Click()
{
    switch (App.DstModel)
    {
        case "IsolatedStorage":
            _lastExportedIsoEntityUriString =
ExportEntityToIso();
            break;
        case "AzureBlobStorage":
            _lastExportedAzureCloudEntityUriString =
ExportEntityToAzureCloud();
            break;
        case "WebUriStorage":
            _lastExportedWebEntityUriString =
ExportEntityToWeb();
            break;
```

[0196]

```
        default:
            break;
    }
}

public string ExportEntityToIso()
{
    int index = SelectedIndex;
    string StorageType = "IsolatedStorage";
    string StorageHostName = "";
    string StorageContainerName = "";
    string StoragePathName = "";
    string StorageFileName = Account + "_" + Table + ".tweet";
    return ExportEntity(index, StorageType, StorageHostName,
StorageContainerName, StoragePathName, StorageFileName);
}

public string ExportEntityToAzureCloud()
{
    int index = SelectedIndex;
    string StorageType = "AzureBlobStorage";
    string StorageHostName = "";
    string StorageContainerName = App.DstContTabQueOrDir;
    string StoragePathName = "";
    string StorageFileName = Account + "_" + Table + ".tweet";
    return ExportEntity(index, StorageType, StorageHostName,
StorageContainerName, StoragePathName, StorageFileName);
}

public string ExportEntityToWeb()
{
```

[0197]

```
        int index = SelectedIndex;
        string StorageType = "WebUriStorage";
        string StorageHostName = App.SrcAccountOrHost;
        string StorageContainerName = App.DstContTabQueOrDir;
        string StoragePathName = "";
        string StorageFileName = Account + "_" + Table + ".tweet";

        return ExportEntity(index, StorageType, StorageHostName,
StorageContainerName, StoragePathName, StorageFileName);
    }

    public string ExportEntity(int index, string StorageType, string
StorageHostName, string StorageContainerName, string StoragePathName, string
StorageFileName)
    {
        string uriAsString = "";
        string data = "";

        data =
DataContractSerializerHelpers.SerializeToString(this.Model);

        switch (StorageType)
        {
            case "IsolatedStorage":
                IsoImportExport.WriteStringToIsoFile(data,
StorageFileName);

                uriAsString = StorageFileName;
                break;
            case "FileSystemStorage":
                MessageBox.Show("FileSystemStorage API is not
implemented, maybe you should use IsolatedStorage anyway!");
```

[0198]

```
uriAsString = "";
break;
case "AzureBlobStorage":
    IsoImportExport.WriteStringToIsoFile(data,
StorageFileName);

    byte[] b = Convert.FromBase64String(data);
    Stream memStream = new MemoryStream(b);
    if (pUploader == null) pUploader = new
UploadFile();

    string ret = pUploader.DoFileUpload(memStream,
StoragePathName, StorageFileName, App.File_TimeOutSeconds,
App.File_SasExpired, App.File_SasUrl);
    if (ret == "Blocked")
    {
        MessageBox.Show("Upload is currently
blocked, try again!");

        uriAsString = "";
        return uriAsString;
    }
    uriAsString = "http://" + Account +
".blob.core.windows.net" + "/" + StorageContainerName + "/" +
StoragePathName + StorageFileName;
    break;
case "WebUriStorage":
    string url = "http://" + StorageHostName + "/" +
StorageContainerName + "/" + StorageFileName;
    Uri uri = new Uri(url, UriKind.Absolute);
    WebImportExport.WriteStringToWebFile(uri,
data);

    uriAsString = url;
    break;
```

[0199]

```
        default:
            uriAsString = "";
            break;
        }
        return uriAsString;
    }

    public void
    ItemViewPopulateApplicationBarButtons(IApplicationBar applicationBar)
    {
        applicationBar.Buttons.Clear();
        applicationBar.MenuItems.Clear();

        var saveButton = new ApplicationBarIconButton(new
        Uri(string.Format(CultureInfo.InvariantCulture, "{0}{1}",
            "/Toolkit.Content/", "ApplicationBar.Save.png"),
        UriKind.Relative)) { Text = "SAVE" };
        saveButton.Click += (s, e) =>
        this.ApplicationBarIconButton_SAVE_ENTITY_Click();
        applicationBar.Buttons.Add(saveButton);

        var cancelButton = new ApplicationBarIconButton(new
        Uri(string.Format(CultureInfo.InvariantCulture, "{0}{1}",
            "/Toolkit.Content/", "ApplicationBar.Cancel.png"),
        UriKind.Relative)) { Text = "CANCEL" };
        cancelButton.Click += (s, e) =>
        this.ApplicationBarIconButton_CANCEL_INPUT_Click();
        applicationBar.Buttons.Add(cancelButton);

        var delButton = new ApplicationBarIconButton(new
        Uri(string.Format(CultureInfo.InvariantCulture, "{0}{1}",
```

[0200]

```
        "/Toolkit.Content/", "ApplicationBar.Delete.png"),
UriKind.Relative)) { Text = "DEL ENTITY" };
        delButton.Click += (s, e) =>
this.ApplicationBarIconButton_REMOVE_ENTITY_Click();
        applicationBar.Buttons.Add(delButton);

        var exportMenue = new ApplicationBarItem("export
model data to file ...");
        exportMenue.Click += (s, e) =>
this.ApplicationBarMenueButton_EXPORT_Click();
        applicationBar.MenuItems.Add(exportMenue);

        var importMenue = new ApplicationBarItem("import
model data from file into this entity ...");
        importMenue.Click += (s, e) =>
this.ApplicationBarMenueButton_IMPORT_Click();
        applicationBar.MenuItems.Add(importMenue);
    }

    private void NavigateBackToCaller()
    {
        int SelectedIndex = -1;
        var destination = (Uri)null;
        if (App.DesignerMode.ToUpper() == "NORMAL" ||
App.DesignerMode == null || App.DesignerMode.ToUpper() == "")
            destination = new
Uri("/Views/NormalMainPage.xaml?selectedItem=" + SelectedIndex,
UriKind.Relative);
        if (App.DesignerMode.ToUpper() == "PIVOT")
            destination = new
Uri("/Views/PivotMainPage.xaml?selectedItem=" + SelectedIndex,
```

[0201]

```
UriKind.Relative);
        if (App.DesignerMode.ToUpper() == "PANORAMA")
            destination = new
Uri("/Views/PanoramaMainPage.xaml?selectedItem=" + SelectedIndex,
UriKind.Relative);
        this.RaiseNavigate(destination);
    }

    private void SetApplicationState(string key, object value)
    {
        if
(PHONEAPPLICATIONSERVICE.Current.State.ContainsKey(key))
        {
            PHONEAPPLICATIONSERVICE.Current.State.Remove(key);
        }
        PHONEAPPLICATIONSERVICE.Current.State.Add(key, value);
    }

    private F GetApplicationState<F>(string key)
    {
        if
(!PHONEAPPLICATIONSERVICE.Current.State.ContainsKey(key))
        {
            return default(F);
        }

        return (F)PHONEAPPLICATIONSERVICE.Current.State[key];
    }

    private void RemoveApplicationState(string key)
    {

```

[0202]

```
        if
        (PhoneApplicationService.Current.State.ContainsKey(key))
        {
            PhoneApplicationService.Current.State.Remove(key);
        }
    }
}
```

[0203] 附图标记列表

[0204]	1	系统
[0205]	2, 2'	(医学)设施
[0206]	3	模态
[0207]	4	计算机断层造影设备
[0208]	5	C形臂设备
[0209]	6	磁共振断层造影设备
[0210]	7	(控制和分析)计算机
[0211]	8	装置
[0212]	9	个人计算机
[0213]	10	显示屏
[0214]	11	平板计算机
[0215]	12	智能电话
[0216]	13	公共云
[0217]	14	(数据传输)网络
[0218]	15	内网
[0219]	16	因特网
[0220]	17	防火墙
[0221]	18, 18'	订阅
[0222]	19	云存储器
[0223]	20	App Store
[0224]	21	云计算服务
[0225]	22	(患者)中心
[0226]	23	(工作列表)中心
[0227]	24	(上下文)中心
[0228]	25	(图像)中心
[0229]	26	表存储器
[0230]	27	留言
[0231]	28	Blob 存储器

[0232]	29	URI
[0233]	31	应用
[0234]	32	应用
[0235]	33	应用
[0236]	34	应用
[0237]	35	消息来源
[0238]	36	GUI 页
[0239]	37	(图形) 用户界面
[0240]	40	页眉
[0241]	41	搜索区
[0242]	42	显示区
[0243]	43	命令条
[0244]	44	输入行
[0245]	45	按钮
[0246]	46	预选区
[0247]	47	图像
[0248]	48	关键字
[0249]	49	按钮
[0250]	50	按钮
[0251]	51	框架
[0252]	52	GUI 页
[0253]	60	页眉
[0254]	61	标题
[0255]	62	显示区
[0256]	63	命令行
[0257]	64	按钮
[0258]	65	按钮
[0259]	67	字段名
[0260]	68	字段内容
[0261]	69	页
[0262]	70	页眉
[0263]	71	标题
[0264]	72	显示区
[0265]	73	命令行
[0266]	74	按钮
[0267]	80	菜单行
[0268]	81	框架
[0269]	90	框架层
[0270]	91	框架组件

[0271]	92	视图层
[0272]	93	组件
[0273]	94	组件
[0274]	95	视图模型层
[0275]	96	组件
[0276]	97	组件
[0277]	98	模型层
[0278]	99	组件
[0279]	100	驱动器层
[0280]	101	驱动器组件
[0281]	102	数据存储器
[0282]	103	驱动器 API
[0283]	P	患者数据组
[0284]	W	任务
[0285]	C	上下文数据组
[0286]	B	图像数据组
[0287]	V	视图
[0288]	Q	搜索查询
[0289]	R	查询
[0290]	K	关键图像

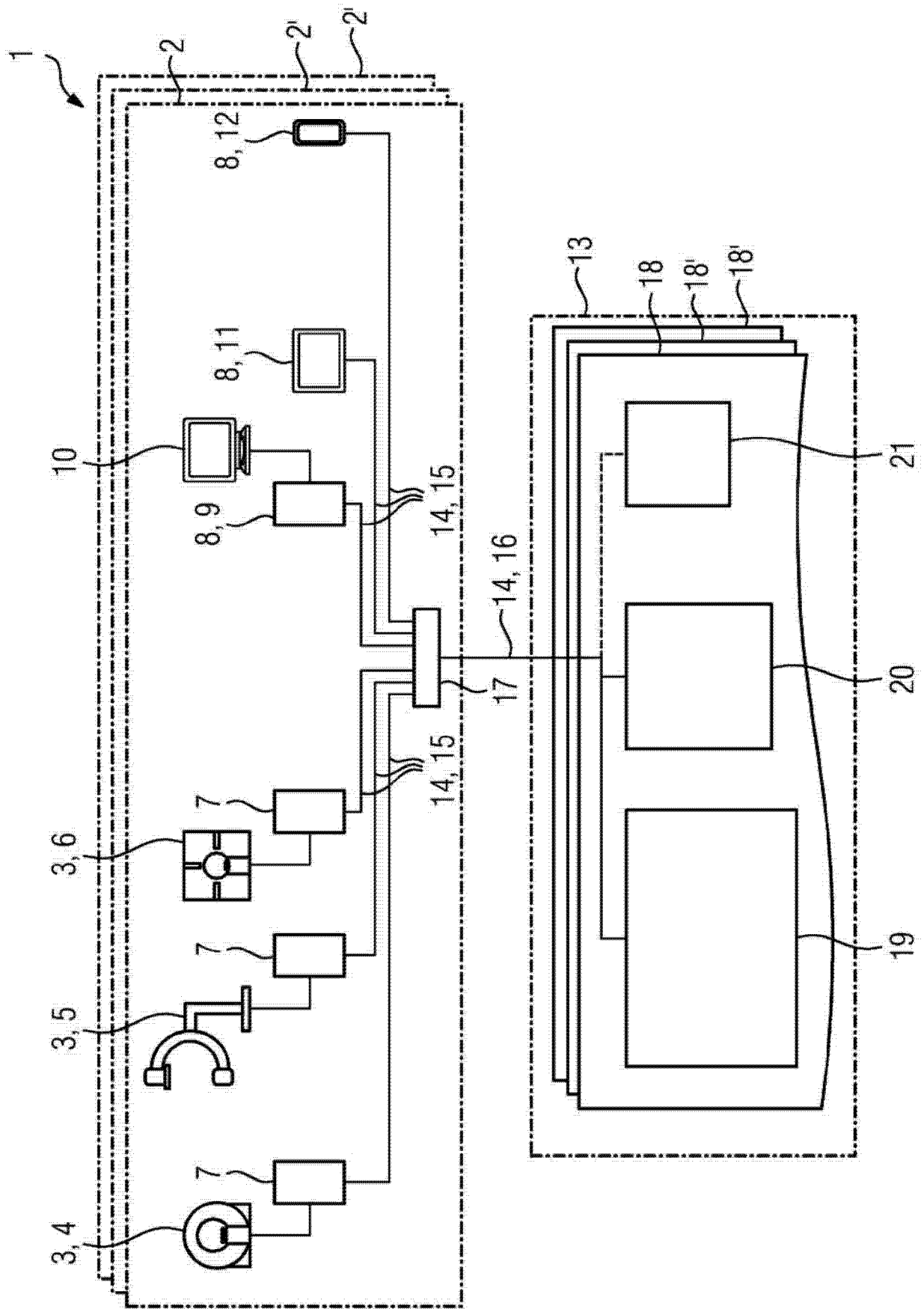


图 1

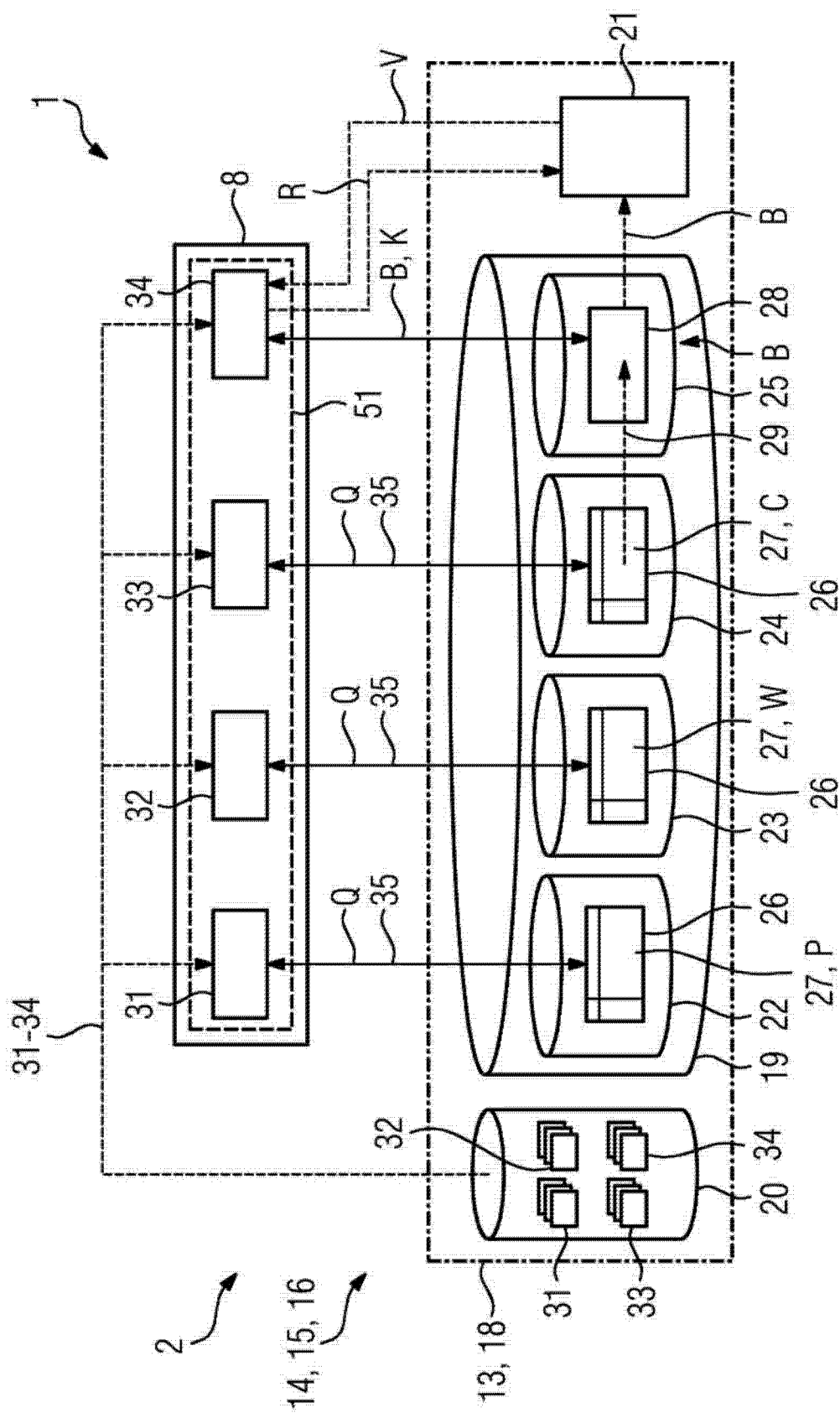


图 2

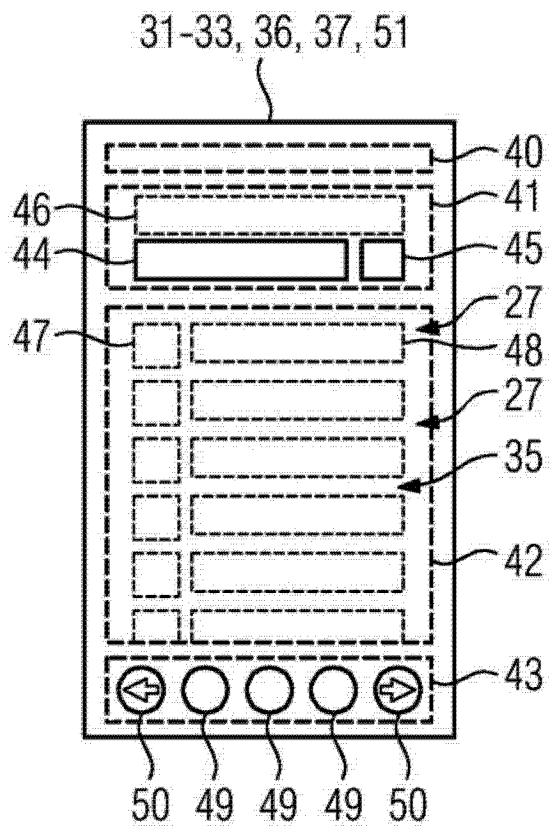


图 3

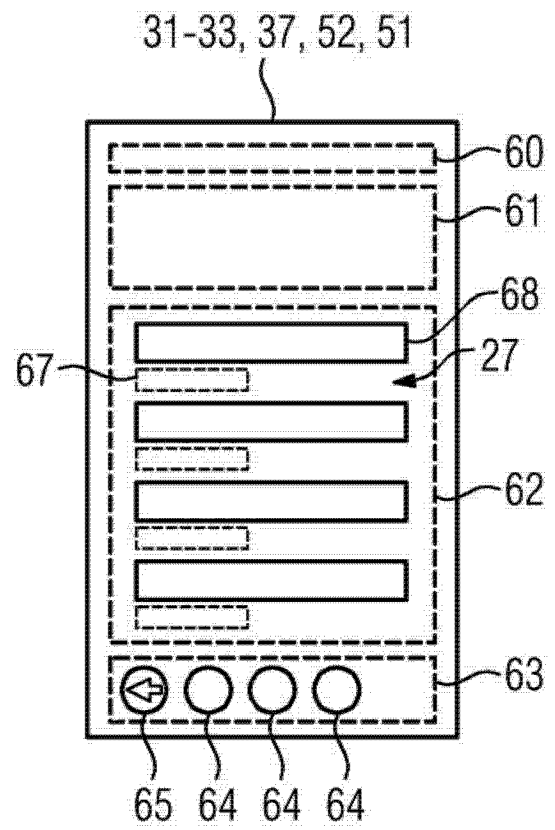


图 4

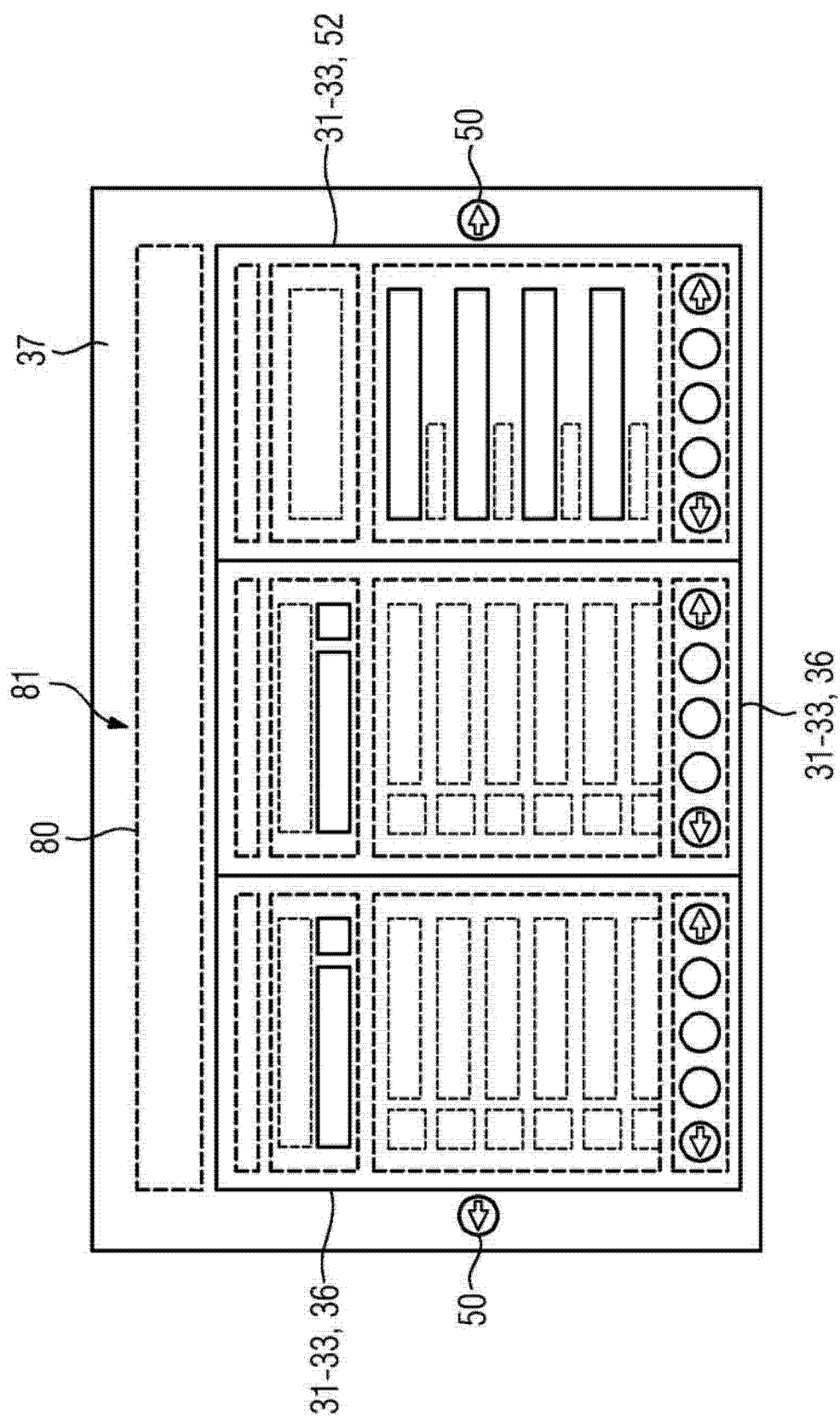


图 5

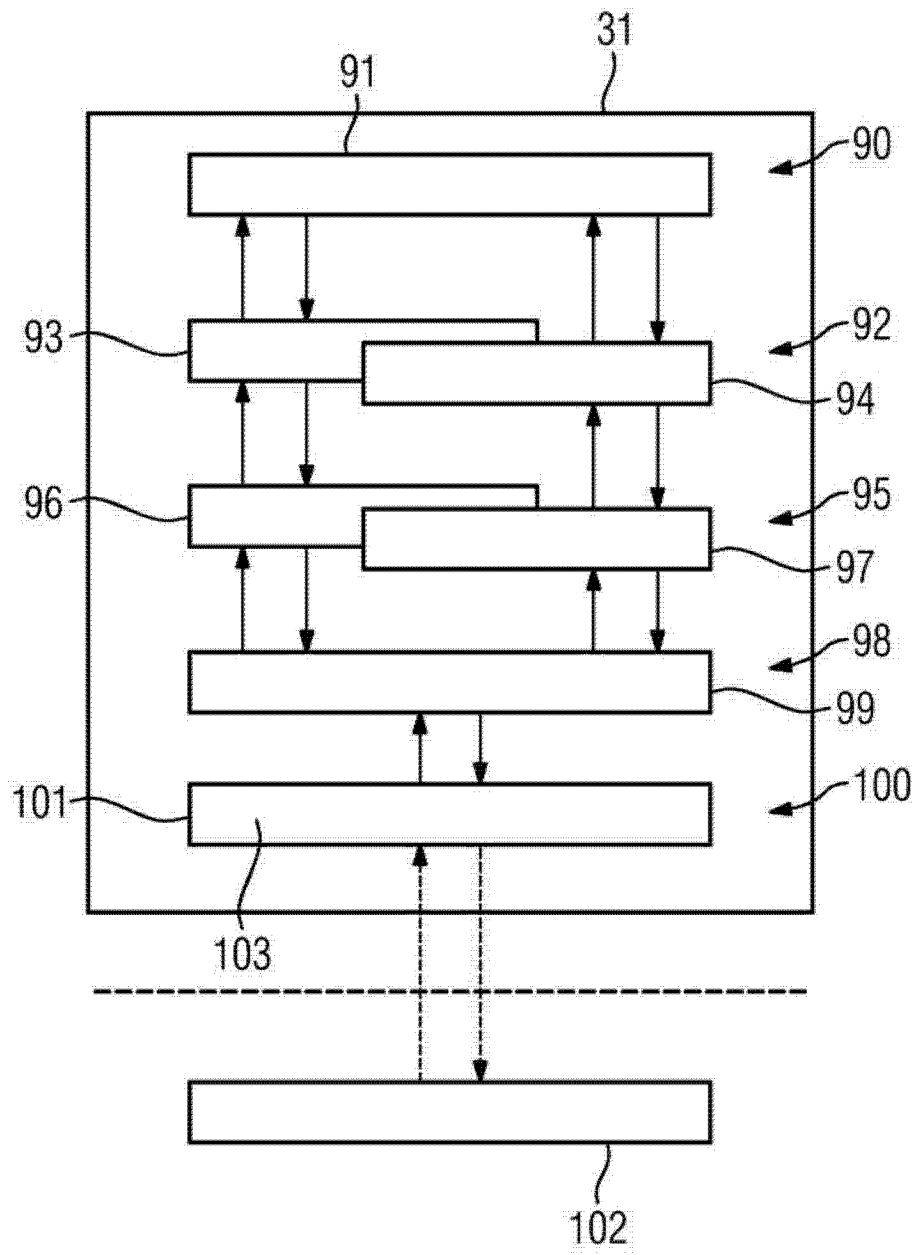


图 6

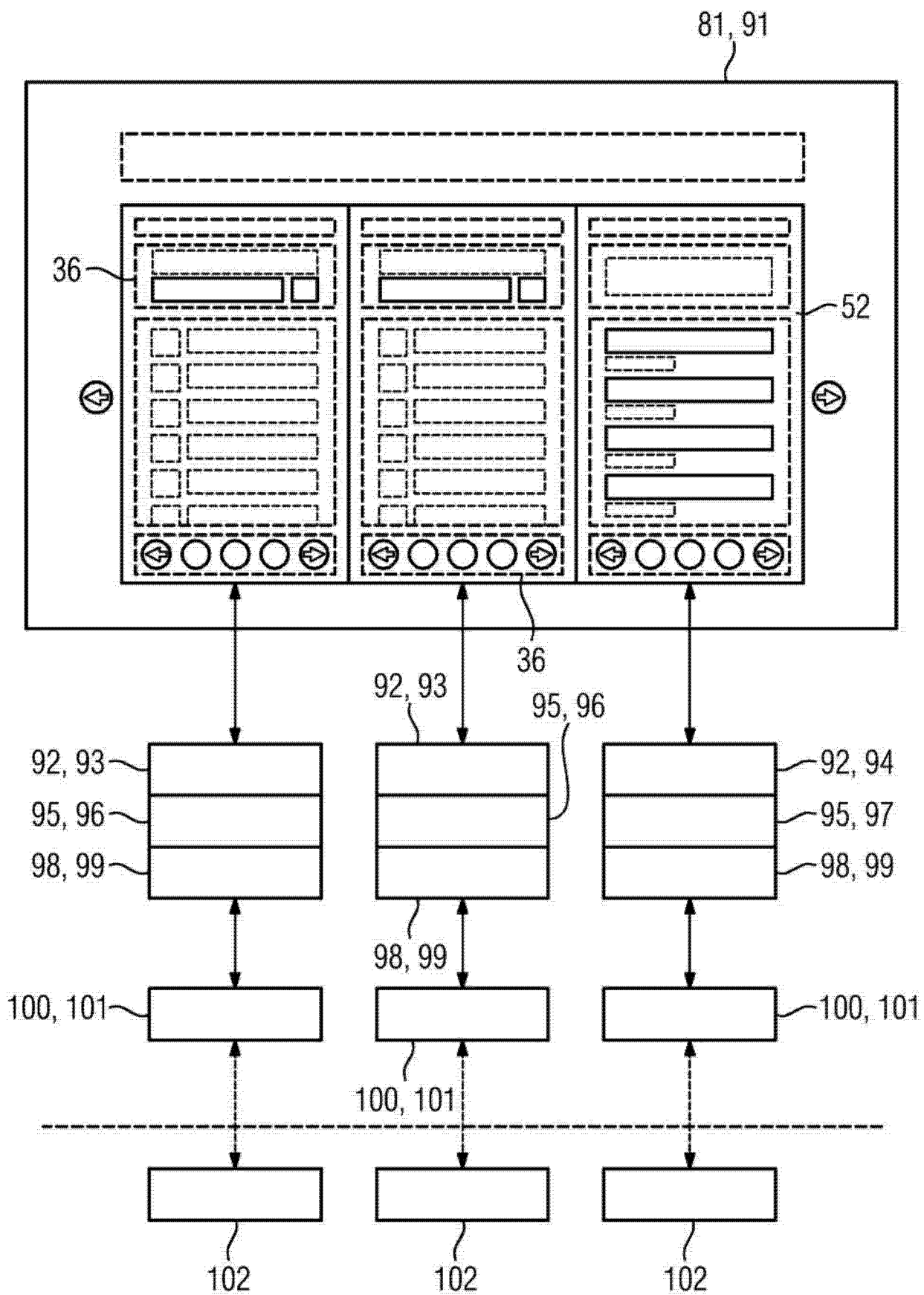


图 7