(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102105862 B
(45)授权公告日 2014.07.23

(21)申请号 200980130468.X
(22)申请日 2009.07.16
(30)优先权数据
12/180,563 2008.07.28 US
(85)PCT国际申请进入国家阶段日
2011.01.27
(86)PCT国际申请的申请数据
PCT/US2009/050890 2009.07.16
(87)PCT国际申请的公布数据
WO2010/014429 EN 2010.02.04
(73)专利权人 微软公司
地址 美国华盛顿州
(72)发明人 R·L·F·布里德
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 胡利鸣
(51)Int.Cl.
G06F 9/44(2006.01)
G06F 3/0483(2013.01)
(56)对比文件
US 2007/0157087 A1,2007.07.05, 全文.
US 2007/0130205 A1,2007.06.07,
CN 1969312 A,2007.05.23, 全文.
US 2002/0063734 A1,2002.05.30, 全文.
审查员 史佳鹏

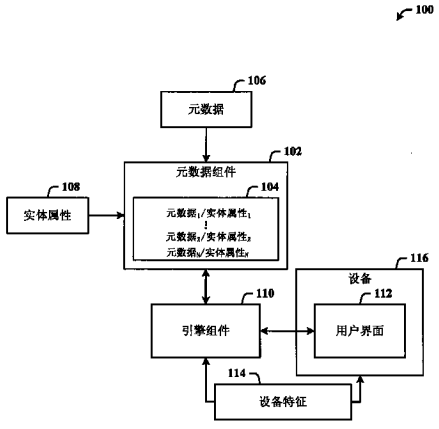
权利要求书2页 说明书11页 附图12页

(54)发明名称

实体交互的自动用户界面生成

(57)摘要

本发明公开了实体交互的自动用户界面生成。架构通过提供自动地创建应用用户界面(UI)的片断的引擎,允许开发者更快地创建应用。引擎能将实体或任何实体类型的实例作为输入,创建一允许应用用户查看并修改实体的UI作为输出。架构也促进了元数据和源实体的关联来引导引擎决定;诸如以下决定,引擎选择哪些UI控件来表示实体属性,向实体提供了多少"区域"(UI空间),以及如何放置实体属性。此外,应用允许用户与已知的实体类型进行交互,但也允许与在应用设计时所未知的类型进行交互。换句话说,应用(例如,行业)能够处理动态生成的随机实体。



CN 102105862 B

1. 一种计算机实现的界面生成系统(100),包括 :

用于创建元数据和实体关联的元数据组件 (102),所述元数据组件包括由值范围表示的重要性元数据,所述重要性元数据被用于在用户界面上对所述实体按垂直地或水平地进行排序 ;

引擎组件(110),所述引擎组件用于 :

为设备自动创建用户界面 ;

基于所述元数据在用户界面内呈现实体,其中根据使用所述重要性元数据和组元数据的布局并基于所述设备的设备查看者的可用区域来在所述用户界面中呈现所述实体 ; 以及

展示重要性阈值,所述重要性阈值被用于选择性地显示所述实体中的各个实体的更多或更少的字段,使得具有低于所述重要性阈值的重要性的属性被忽视并且不在所述用户界面中表示。

2. 如权利要求 1 所述的系统,其特征在于,所述引擎组件进一步考虑所述用户界面要在其中呈现所述用户界面的设备的设备特征。

3. 如权利要求 1 所述的系统,其特征在于,所述引擎组件展示用于生成所述用户界面的实体类型和实体实例,并使用实体实例来填充所述用户界面。

4. 如权利要求 1 所述的系统,其特征在于,所述引擎组件促进与实体的用户交互,包括可视化、编辑以及验证。

5. 如权利要求 1 所述的系统,其特征在于,所述元数据组件和所述引擎组件是行业应用的一部分,所述元数据组件和所述引擎组件促进动态生成的随机商业实体的处理。

6. 如权利要求 1 所述的系统,其特征在于,所述引擎组件展示用于所述用户界面的生成的实体类型、或用于生成所述用户界面的实体实例,并使用实体实例来填充所述用户界面。

7. 如权利要求 1 所述的系统,其特征在于,所述元数据组件将定义较不重要性的所述重要性元数据与较不重要的实体属性关联,并且所述引擎组件基于所述重要性元数据从用户界面的视图中隐藏较不重要的实体属性,并使得已隐藏的较不重要的实体属性通过可选择的链接可见。

8. 一种用于生成界面的计算机实现的方法,包括 :

将元数据与实体关联(1000);

基于所述元数据为设备自动创建用于实体的表示的用户界面(1002),其中根据使用重要性元数据和组元数据的布局并基于所述设备的可查看区域来在所述用户界面中呈现所述实体,其中所述重要性元数据由值范围表示并定义所述实体的一个或多个属性的相对重要性 ;以及

展示重要性阈值,所述重要性阈值被用于选择性地显示所述实体中的各个实体的更多或更少的字段,使得具有低于所述重要性阈值的重要性的属性被忽视并且不在所述用户界面中表示。

9. 如权利要求 8 所述的方法,其特征在于,进一步包括基于所述元数据对实体属性进行分组。

10. 如权利要求 8 所述的方法,其特征在于,进一步包括基于所述设备的所述可查看区

域选择实体的布局。

11. 如权利要求 8 所述的方法，其特征在于，进一步包括基于使用元数据重新解释传送到实体的数据。

实体交互的自动用户界面生成

技术领域

[0001] 本发明涉及实体交互的自动用户界面生成。

背景技术

[0002] 通常，软件应用，以及尤其是行业 (LOB) 应用表示各种自然的数据对象（也称为实体）。例如，在行业 (LOB) 应用中，客户、订单、产品、以及发票是需要被创建并操作的实体的示例。由于应用能用于多个不同的部署，对各个独立的应用以及应用要在其上运行的各个类型的设备设计并创建用户界面。由此，开发者必须为那些实体类型中的每一个创建特定的图形用户界面。这是消耗时间的并且是相当重复的任务。然而，如果不是必须创建实体特定的用户界面，应用可被更快地创建。

[0003] 概述

[0004] 下面提供了简化的概述，以便提供对此处所描述的一些新颖实施例的基本理解。本概述不是详尽的概述，并且它不旨在标识关键 / 重要元素或描绘本发明的范围。其唯一的目的是以简化形式呈现一些概念，作为稍后呈现的更详细描述的序言。

[0005] 揭示的架构通过提供能自动地创建应用的用户界面 (UI) 的片断的引擎，允许开发者更快地创建应用。引擎能将实体或任何实体类型的实例作为输入，并创建一允许应用的用户查看并修改实体的 UI 作为输出。架构也促进了元数据和源实体的关联来引导引擎决定 ；决定诸如 ：引擎选择哪些 UI 控件来表示实体属性，向实体提供了多少“区域 (real estate)”(UI 空间)，以及如何放置 UI 控件。

[0006] 此外，应用允许用户与已知的实体类型进行交互，但也允许与在应用设计时所未知的类型进行交互。换句话说，应用（例如，行业）能够处理动态生成的随机实体。

[0007] 为了为实现上述及相关目的，本文结合下面的描述和附图来描述某些说明性方面。这些方面指示了可以实践本文所公开的原理的各种方式，所有方面及其等效方面旨在落入所要求保护的主题的范围内。结合附图阅读下面的详细描述，其他优点和新颖特征将变得显而易见。

[0008] 附图简述

[0009] 图 1 示出了根据所公开的架构的计算机实现的界面生成系统。

[0010] 图 2 示出了由引擎组件展示的实体信息。

[0011] 图 3 示出了能用于创建 UI 的属性元数据的实例。

[0012] 图 4 示出了使用多个引擎生成不同 UI 的系统。

[0013] 图 5 示出了通过使用元数据和实体的关联能在运行时被创建的示例性用户界面。

[0014] 图 6 示出了在自动生成的 UI 内实体属性的水平流布局。

[0015] 图 7 示出了在自动生成的 UI 内实体属性的垂直流布局。

[0016] 图 8 示出了在自动生成的 UI 内实体属性的水平流布局。

[0017] 图 9 示出了在自动生成的 UI 内实体属性的垂直流布局。

[0018] 图 10 示出了生成界面的计算机实现的方法。

[0019] 图 11 示出了将布局策略应用到已生成的用户界面的方法。

[0020] 图 12 示出了当实体类型数据被设置时，由引擎进行处理的方法。

[0021] 图 13 示出可用于根据所公开的架构将元数据与实体关联并自动地生成 UI 的计算系统的框图。

[0022] 详细描述

[0023] 诸如行业 (LOB) 的应用经常操作实体（具有属性的对象，诸如客户对象）。相应地，通常在可视化实体和编辑实体中涉及应用。通常，实体存储在数据库内，以在列表中的紧凑表示显示，或单独地以扩展的表示显示。在大多数情况下，开发者需要从头创建用户界面 (UI) 来以详细视图表示具体的实体。

[0024] 所揭示的架构通过将元数据附加到实体促进了应用 UI 的自动创建，该元数据能够引导 UI 生成器（引擎）以产生更加有针对性的结果。引擎使用逻辑或算法在给定实体、实体元数据和设备特征的情况下产生有意义的 UI，诸如包括用于在 UI 内表示实体的被分配区域的硬件参数以及软件参数。

[0025] 现在将参考附图，全部附图中相同的附图标记用于表示相同的元件。在下面的描述中，为了进行说明，阐述了很多具体细节以便提供对本发明的全面理解。然而，显而易见，可以没有这些具体细节的情况下实施各新颖实施例。在其它情况下，以框图形式示出了公知的结构和设备以便于描述它们。本发明将涵盖落入所要求保护的主题的精神和范围内的所有修改、等效方案和替换方案。

[0026] 图 1 示出了根据所公开的架构的计算机实现的界面生成系统 100。系统 100 包括元数据组件 102，用于创建元数据 106 和实体属性 108 的关联 104（表示为 $Metadata_1$/$EntityProperty_1$（元数据 1/ 实体属性 1），，$Metadata_N$/$EntityProperty_N$（元数据 $_N$/ 实体属性 $_N$）），以及用于基于元数据 106 自动地创建用户界面 112 并在用户界面 112 中呈现实体属性 108 的引擎组件 110。

[0027] 引擎组件 110 还能考虑在其中呈现用户界面 112 的设备 116 的设备特征 114。设备特征 114 可以包括设备 116 的硬件能力和 / 或软件能力，诸如查看在设备 116 上可用于查看用户界面 112 的区域。引擎 110 促进与实体属性 108 的用户交互，交互包括，例如，可视化、编辑和 / 或验证。

[0028] 元数据组件 102 和引擎组件 110 可以是 LOB 应用的一部分。在这样的实施中，元数据组件 102 和引擎组件 110 促进已动态生成的随机实体的处理，随机实体可以是商业实体。

[0029] 如将要在以下更加详细描述的，可以使用重要性元数据和组元数据、并基于用户界面 112 的设备查看者的可用区域，在用户界面 112 中呈现实体 108。元数据组件 102 包括重要性元数据。重要性元数据定义与实体属性关联的重要性等级（例如，较不重要）。例如，较不重要的实体属性将与指示较低等级重要性的重要性元数据关联。引擎组件 110 基于重要性元数据从用户界面 112 中的视图隐藏较不重要的实体属性。然而，已隐藏的较不重要的实体属性能通过可选择的链接变得可查看。当用户选择该链接时，已隐藏的较不重要的实体属性能被查看。

[0030] 根据使用，例如，重要性元数据和组元数据的布局、并基于设备查看者的可用区域，在用户界面 112 中呈现实体属性 108 的部分或全部。元数据 106 涉及，仅举数例，可见

性、大小、呈现提示、分组、重要性、以及使用。

[0031] 图 2 示出了由引擎组件 110 展示的实体信息。引擎组件 110 能展示实体类型 200 以及实体实例 202，用于生成用户界面并用实体实例填充用户界面。换言之，引擎组件 110 展示用于用户界面的生成的实体类型 200，和 / 或用于生成用户界面并用实体实例 202 填充用户界面的实体实例 202。以下是用于展示实体信息的实例代码。

[0032] public Type EntityViewer.EntityType

[0033] public object EntityViewer.Entity

[0034] 设置 EntityType 允许 UI 的生成。设置 Entity 不仅仅允许 UI 的生成，也用所提供的实体填充 UI。

[0035] 引擎组件 110 确定与各种实体属性关联的潜在的属性，并基于那元数据和内建映射（例如，映射到文本属性编辑器的串属性）在运行时构建 UI。

[0036] 图 3 示出了能被用于创建 UI 的属性元数据 106 的元数据示例 300。示例 300 包括但不限于，例如，可见性、显示名称、通常大小（例如，长度、宽度、高度、以及其变型）、呈现提示、组、重要性、以及使用。

[0037] 示例 300 能被代码表示为如下：UIDescriptionVisible(UI 描述可见性)(boolvisible)，UIDescriptionDisplayName(UI 描述显示名称)(string displayName)，UIDescriptionTypicalSize(UI 描述通常大小)(uint length，uint variation)，UIDescriptionTypicalSize(UI 描述通常大小)(uint width，uint height，uintwidthVariation，uint heightVariation)，UIDescriptionRenderHint(UI 描述呈现提示)(string assemblyQualifiedTypeName)，UIDescriptionRenderHintv(UI 描述呈现提示)(Type type)，UIDescriptionGroup(UI 描述组)(string groupName)，UIDescriptionImportanceAttribute(UI 描述重要性属性)(uint importance)，andUIDescriptionUsageAttribute(UI 描述使用属性)(UIDescriptionPropertyUsagepropertyUsage)。

[0038] 公开的架构的其它方面包括以下：适合于在 UI 内表示的可读公共属性；基于 UIDescriptionGroup 属性进行分组的实体属性；基于 UIDescriptionImportance（基于多个可能的排序算法）对不属于一个组（一单实体属性）以及属性的多个组的属性进行排序；组内的属性也基于 UIDescriptionImportance 进行排序；基于属性类型、可写性、UIDescriptionRenderHint 属性挑选 UI 控件；基于 UIDescriptionTypicalSize 属性对 UI 控件的大小缩放；UI 控件基于 UIDescriptionUsage 属性重新解释接收到的数据（例如，串能被重新解释为图像 URI（统一资源标识符））；基于布局策略以及可用的区域大小放置 UI 控件。

[0039] 引擎组件（也称为实体查看者控件）可以展示被称为重要性阈值的属性。该阈值可以被称为虚拟调节器 (virtual knob)，当调节时显示实体的较多或较少的字段。例如，假设名字被给定重要性 17。如果实体控件的阈值值是例如 50，那么具有重要性大于 50 的属性将在 UI 内呈现。具有低于 50 的重要性的属性将被忽视并且不再 UI 内表示。

[0040] 图 4 示出了使用多个引擎 402 来生成不同 UI404 的系统 400。如上所述，元数据组件 102 基于元数据 106 和实体属性 108 创建元数据 / 实体属性的关联 104。在此，引擎组件 110 包括多个引擎 402（表示为引擎 $_1$，，引擎 $_S$）用于基于设备特征创建 UI（表示为 UI_1，，UI_T），设备特征诸如在其上将呈现实体（例如，商业对象）的设备的可查看区域。例如，如果

正在使用的应用在台式计算机上运行，那么相比于如果该应用在移动设备上运行的情况，生成不同类型的UI。由此，不同类型控件和不同类型交互模型的使用，以及元数据对于实体的附加使得用户在运行时创建以不同类型的设备能力和设备供应商为目标的应用。

[0041] 图5示出了通过使用元数据和实体的关联能在运行时被创建的示例性用户界面500。UI500显示了实体属性的布局以及分组。姓名分组502将名字和姓氏进行组合。类似地，地址分组504将房屋号(#)实体、街道实体、城市实体、邮政编码实体以及州实体进行组合。照片实体506也被呈现。当处理元数据时，UI500基于元数据在运行时被创建，该元数据指示姓名分组502、地址分组504、照片实体506、以及其它实体属性（例如，生日、年龄、个人介绍等）将基于显示的布局策略（例如，自顶向下的顺序和/或从左往右的顺序）被呈现。

[0042] UI500，可以例如，根据设备上的可用区域来改变布局和大小，其中应用（例如，商业）正在为该设备生成视图。例如，由于在视图内有更多的区域，大的文本框能被选择并在台式机器上呈现，相反，较小的文本框将被选择以在给定小了许多的查看区域的PDA或手机上呈现。然而，诸如姓氏的某些信息能被固定在最小和/或最大尺寸。例如，姓氏的通常大小是10个字符，并且关联的实体能被限制为10个字符。

[0043] 元数据能作为给定应用的默认的一组常用实体来提供，然后能使用由用户创建的一组自定义的元数据来补充。例如，金融应用能具有关于姓名和地址的一组相同的基本元数据，但是也具有与具有相同的默认组以及涉及产品、物流等的其它元数据的商业应用不同的关于账户、利息等的其它元数据。

[0044] 关于组信息，一个人具有名字和姓氏。当生成一个人的UI500时，本能地将姓氏和名字邻近地放置（例如，并排）。这意味着名字和姓氏能共享相同的组502。由此，元数据与指示与名字和姓氏关联的组Name（姓名）的这两个属性相关联。此人具有由称为地址(Address)的组504定义的地址，该地址可以包括街道号、街道名称、邮政编码、城市名称、国家等。

[0045] 重要性元数据能够是值范围（例如，一在2-100范围内的值）。特定组的重要性元数据以及在该组上的放置的属性（例如，该组放置在用户的空间内，将组放在底下、隐藏该组等）能指示该组的查看位置，或该组到底是否要被呈现。当为一个人创建UI500时，本能地，名字和姓氏将是突出的；由此，名字和姓氏属性的重要性元数据将是高的。另一方面，眼睛颜色可以是一个人的无意义的特征；因此，相对于名字而言，重要性元数据是低的。

[0046] 相应地，当生成UI500时，实体属性能被自上向下放置，其中被分配重要性元数据的高重要性属性在上而被分配重要性元数据的较低属性在降序范围内。

[0047] 可以对较不重要的属性创建链接使得用户实际上必须选择链接来引起对话框的呈现，例如，显示较不重要属性的对话框。当实体要在与例如PDA或手机关联的区域上呈现时，这是非常有用的。因此，如果UI是针对较大的手机界面设计的，但是然后在较小的手机界面上使用，链接能自动地在运行时实现以适应较小的UI。然后，用户可以选择链接来访问隐藏的属性。

[0048] 例如，考虑引擎所接收指示查看区域的设备特征是宽200像素、高300像素。引擎随后创建用于该设备的合适的UI。然而，当在PDA上使用时，引擎可以接收指示查看区域的设备特征是宽50像素、高60像素，并接着创建合适的UI。相应地，将具有诸如与UI的向导

类型关联的链接的 UI 呈现给 PDA 用户，其中导航是从一页面到另一页面，以可视化此人的不同特征。相反，台式机用户可被给予足够查看此人的所有特征的单个表格。

[0049] 在更稳健的实施例中，除了查看区域，可以考虑其它设备特征，诸如 CPU、存储器、软件（例如，操作系统）、输入设备（例如，键盘、麦克风、鼠标等）、语音输入能力等。此外，或可替换地，可以基于用户偏好（例如，用户偏好在上部放置图像，接下来是姓名信息，并且没有地址信息）和 / 或数据类型（例如，金融、商业）创建 UI。

[0050] 在另一示例中，用户能与 UI 交互，以按响应于用户交互来生成元数据的方式更改 UI。接着可以将该元数据附加或合并到现有的来自实体的元数据中，并且引擎不仅仅将由实体提供的元数据作为输入，也将由终端用户生成的元数据作为输入。

[0051] 基于生成的 UI，属性可与访问等级关联。例如，用户的体重信息能被作为只读属性。可替换地，所提供的诸如街道号属性的地址可以是可写的。

[0052] 引擎组件能选择并初始化 UI 内实体方向的各种布局以填充区域。接下来是四个示例，但是可以理解的是也能使用其它布局。图 6 示出了水平流布局 600。在该流布局中，实体呈现是从上至下以及从左到右。图 7 示出了垂直流布局 700。图 8 示出水平布局 800。图 9 示出垂直布局 900。

[0053] 根据正在被创建的表格表示的目的，可以添加特定的实体元数据从而以一种用于正在被创建的视图的特定的方式引导 UI 的创建。使用电子邮件实体作为示例，电子邮件具有发送方、接收方、电子邮件发送的日期和时间、电子邮件在某个日期和时间被接收、以及正文。电子邮件的正文通常可以是一个或多个段落形式的一大段信息（例如，文本、图像、链接等）。由开发者给予电子邮件正文的通常大小是，例如，10,000 字符。当以更紧凑的方式呈现电子邮件时，对正文分配 10,000 字符是不理想的。可以创建一个以更紧凑表示的方式来表示电子邮件的表格。例如，对于特定的视图，可以减少分配给电子邮件的区域，使得对于电子邮件的特定视图，所分配的区域现在是 200 字符。对于特定视图，用户能覆盖附加到特定字段的默认元数据。

[0054] 以下是能被使用的示例性简单类分层。顶级 EntityViewer（实体查看者）类可以为如下：

[0055] pubLIc class EntityViewer :Control, IEntityEditor

[0056] 引擎组件能实现 public interface IEntityEditor，使得实体能以标准方式被编辑。单个属性编辑器能实现 public interfaceIEntityPropertyEditor 以标准化在 UI 片断和顶级实体查看者之间的协议。TextPropertyEditor（文本属性编辑器）是该界面的特定实现：

[0057] public class TextPropertyEditor :Control, IEntityPropertyEditor

[0058] EntityViewer（实体查看者）能够使用各个属性的说明的预定标记：

[0059] public class EntityPropertyLabel :Control

[0060] 界面详细描述的一个示例可包括以下：

[0061]

```
public interface IEntityEditor
{
    bool AllowEdit
    {
        get;
        set;
    }

    object Entity
    {
        get;
        set;
    }

    Type EntityType
    {
        get;
        set;
    }
```

[0062]

```
        event EventHandler CancelEdit;
        event         EventHandler<MemberValueChangedEventArgs>
ValueChanged;
        event         EventHandler<MemberValueChangedEventArgs>
ValueChangeCanceled;
    }

    public interface IEntityPropertyEditor
    {
        bool AllowEdit
        {
            get;
            set;
        }

        object Value
        {
            get;
            set;
        }

        Size PreferredSize(uint width, uint height,
                           uint        widthVariation,        uint
heightVariation);

        event         EventHandler<MemberValueChangedEventArgs>
ValueChanged;
        event         EventHandler<MemberValueChangedEventArgs>
ValueChangeCanceled;
      }
```

[0063] 如在上述代码中使用的，EntityViewer 是引擎组件，并且其负责创建和填充 UI。

EntityViewer 不初始化、提交、或取消关联的实体的编辑。是例如协作数据导航器控件的责任来执行这三个任务。EntityViewer 将由单个属性控件引起的属性更改通知通过 IEntityEditor's ValueChanged 以及 ValueChangedCanceled 事件转发。此外，当用户试图取消实体编辑时，EntityViewer 引起其 CancelEdit（取消编辑）事件。

[0064] 实体属性编辑器负责表示单个属性，并且处理编辑经验（如果有的话）。属性编辑器实现 IEntityPropertyEditor，使得 :EntityViewer 能告知属性应该是只读的或可读写的 ;EntityViewer 以及属性编辑器具有交换属性值的标准方式 ;属性编辑器具有通知值更改（以及取消）的标准方式 ;并且，EntityViewer 能指出用于查看属性所需的合理的区域。

[0065] 以下是公开的架构的过程的功能流程的示例。当 EntityViewer 的 EntityType（实体类型）属性被设置时（例如，在 Entity（实体）属性被设置时直接地或内部地设置），控件能执行以下 :类型属性被枚举并且关联的自定义属性被访问 ;基于那些属性，引擎创建一组 EntityProperty（实体属性）对象和 EntityPropertyGroup（实体属性组）对象，并填充对象 ;引擎对具有建议的可视性的每个属性发起 RequestVisibility（请求可视性）事件，并允许开发者覆盖默认行为（注意的是，可视性取决于属性的重要性是否低于或高于重要性阈值）;然后，基于属性类型、潜在的呈现提示、以及 RequestPropertyRenderControlType（请求属性呈现控件类型）事件选择属性控件（注意的是，虽然属性不需要与 IEntityPropertyEditor 实现关联，但是实体查看者也能处理几个存储控件 :TextBlock（文本块），TextBox（文本框），DateTimePicker（日期时间提取器），and PictureBox（图片框））;如果没有提供呈现提示，当属性使用 UIDescriptionPropertyUsage.ImageSource 和 / 或当属性类型是 System.DateTime 时使用 DateTimePicker 时，实体查看者选取 PictureBox ;确定各个属性控件和关联的标记的偏好大小 ;从那里，组的偏好大小被计算 ;并且，最后，创建呈现组的控件 (GroupBox（组框）控件）。

[0066] 当引擎要呈现自己时，引擎首先根据属性重要性和 GroupImportanceDefinition（组重要性定义）属性对组进行排序。然后，各种组、标记和属性控件根据当前的布局策略被放置。当设置 Entity 属性时，如果类型被更改以及属性控件通过 IEntityPropertyEditor.Value 成员被填充，则 UI 被重新生成。

[0067] 以下是表示用于执行所公开的体系结构的各新颖方面的示例性方法的一系列流程图。尽管出于解释简明的目的，此处例如以流图或流程图形式示出的一个或多个方法被示出并描述为一系列动作，但是可以理解和明白，各方法不受动作的次序的限制，因为根据本发明，某些动作可以按与此处所示并描述的不同的次序和 / 或与其他动作同时发生。例如，本领域技术人员将会明白并理解，方法可被替换地表示为一系列相互关联的状态或事件，诸如以状态图的形式。此外，并非在一方法中示出的所有动作都是新颖实现所必需的。

[0068] 图 10 示出了生成界面的计算机实现的方法。在 1000，元数据与实体关联。在 1002，基于元数据和设备的可查看区域，为该设备自动创建用于实体的表示的用户界面。

[0069] 图 11 示出了将布局策略应用到已生成的用户界面的方法。在 1100，引擎组件检测实体类型属性被设置。在 1102，引擎组件访问并处理实体元数据。在 1104，初始化呈现。在 1106，基于重要性属性和组重要性定义属性，对组进行排序。在 1108，访问布局策略信息。在 1110，根据策略呈现组并标记属性控件。

[0070] 图 12 示出了当实体类型数据被设置时，由引擎进行处理的方法。在 1200，设置引擎检测实体类型属性。在 1202，枚举类型属性并且访问关联的元数据。在 1204，创建实体属性对象并创建实体属性组对象，并且填充各个对象。在 1206，展示并建议可视性数据，并且允许默认覆盖。在 1208，基于类型、呈现提示以及呈现控件类型事件选择属性控件。在 1210，计算各个属性以及关联的标记的大小。在 1212，计算组大小。在 1214，创建表示组的控件。

[0071] 尽管参考如屏幕截图的各个附图示出并描述了向用户显示信息的一些方式，但相关领域的技术人员可以认识到，可采用各种其它替换方案。术语“屏幕”、“屏幕截图”、“网页”、“文档”和“页面”在本文中一般可互换使用。页面或屏幕作为显示描述、作为图形用户界面或通过描绘屏幕（例如，无论是个人计算机、PDA、移动电话还是其它合适的设备）上的信息的其它方法被存储和 / 或传输，其中要显示在页面上的布局和信息或内容被存储在存储器、数据库或另一存储设施中。

[0072] 如在本申请中所使用的，术语“组件”和“系统”旨在表示计算机相关的实体，其可以是硬件、硬件和软件的组合、软件、或者执行中的软件。例如，组件可以是但不限于，在处理器上运行的进程、处理器、硬盘驱动器、多个（光和 / 或磁存储介质的）存储驱动器、对象、可执行代码、执行的线程、程序、和 / 或计算机。作为说明，在服务器上运行的应用程序和服务器两者都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程内，且组件可以位于一台计算机上和 / 或分布在两台或更多的计算机之间。词语“示例性”此处可用于表示用作示例、实例或说明。在此被描述为“示例性”的任何方面或设计并不一定要被解释为相比其他方面或设计更优选或有利。

[0073] 现在参考图 13，图 13 示出可用于根据所公开的架构将元数据与实体关联并自动地生成 UI 的计算系统 1300 的框图。为了提供用于其各方面的附加上下文，图 13 及以下讨论旨在提供对其中可实现该各方面的合适的计算系统 1300 的简要概括描述。尽管以上描述是在可在一个或多个计算机上运行的计算机可执行指令的一般上下文中进行的，但是本领域的技术人员将认识到，新颖实施例也可结合其他程序模块和 / 或作为硬件和软件的组合来实现。

[0074] 一般而言，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、组件、数据结构等等。此外，本领域的技术人员可以理解，本发明的方法可用其他计算机系统配置来实施，包括单处理器或多处理器计算机系统、小型计算机、大型计算机、以及个人计算机、手持式计算设备、基于微处理器的或可编程消费电子产品等，其每一个都可操作上耦合到一个或多个相关联的设备。

[0075] 所示各方面也可以在其中某些任务由通过通信网络链接的远程处理设备来执行的分布式计算环境中实施。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备中。

[0076] 计算机通常包括各种计算机可读介质。计算机可读介质可以是可由计算机访问的任何可用介质，且包括易失性和非易失性介质、可移动和不可移动介质。作为示例而非限制，计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以存储如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术来实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括但不限于 RAM、ROM、EEPROM、闪

存或者其他存储器技术、CD-ROM、数字视频盘 (DVD) 或其他光盘存储、磁带盒、磁带、磁盘存储或其他磁存储设备、或可以用于存储所需信息并且可以由计算机访问的任何其他介质。

[0077] 再次参考图 13，用于实现各方面的示例性计算系统 1300 包括具有处理单元 1304、系统存储器 1306 和系统总线 1308 的计算机 1302。系统总线 1308 向包括但不限于系统存储器 1306 的各系统组件提供到处理单元 1304 的接口。处理单元 1304 可以是市场上可购买到的各种处理器中的任意一种。双微处理器和其他多处理器体系结构也可用作处理单元 1304。

[0078] 系统总线 1308 可以是若干种总线结构中的任一种，这些总线结构还可互连到存储器总线（带有或没有存储器控制器）、外围总线、以及使用各类市场上可购买到的总线体系结构中的任一种的局部总线。系统存储器 1306 可包括非易失性存储器 (NOV-VOL)1310 和 / 或易失性存储器 1312（例如，随机存取存储器 (RAM)）。基本输入 / 输出系统 (BIOS) 可被存储在非易失性存储器 1310（例如，ROM、EPROM、EEPROM 等）中，其中 BIOS 是帮助诸如在启动期间在计算机 1302 内的元件之间传输信息的基本例程。易失性存储器 1312 还可包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0079] 计算机 1302 还包括内置硬盘驱动器 (HDD)1314（例如，EIDE、SATA），该内置 HDD 1314 还可被配置成在合适的机壳中外部使用；磁软盘驱动器 (FDD)1316（例如，从可移动磁盘 1318 中读取或向其写入）；以及光盘驱动器 1320（例如，从 CD-ROM 盘 1322 中读取，或从诸如 DVD 等其他高容量光学介质中读取或向其写入）。HDD 1314、FDD 1316、以及光盘驱动器 1320 可分别由 HDD 接口 1324、FDD 接口 1326 和光盘驱动器接口 1328 连接到系统总线 1308。用于外置驱动器实现的 HDD 接口 1324 可包括通用串行总线 (USB) 和 IEEE 1394 接口技术中的至少一种或两者。

[0080] 驱动器及相关联的计算机可读介质提供了对数据、数据结构、计算机可执行指令等的非易失性存储。对于计算机 1302，驱动器和介质容纳适当的数字格式的任何数据的存储。尽管以上对计算机可读介质的描述涉及 HDD、可移动磁盘（例如 FDD）以及诸如 CD 或 DVD 等可移动光学介质，但是本领域的技术人员应当理解，示例性操作环境中也可使用可由计算机读取的任何其他类型的介质，诸如 zip 驱动器、磁带盒、闪存卡、盒式磁带等等，并且任何这样的介质可包含用于执行所公开的体系结构的新颖方法的计算机可执行指令。

[0081] 多个程序模块可被存储在驱动器和易失性存储器 1312 中，包括操作系统 1330、一个或多个应用程序 1332、其他程序模块 1334 和程序数据 1336。一个或多个应用程序 1332、其他程序模块 1334、以及程序数据 1336 能包括例如，元数据组件 102、关联 104、元数据 106、实体属性 108、实体组件 110、UI112、设备特征 114、实体类型 200、实体实例 202、元数据示例 300、引擎 402、UI404、UI500、水平以及垂直流 (600 和 700)，水平以及垂直布局 (800 和 900)，附图 10-12 的方法。设备 116 能够是计算机 1302、手机、PDA、或其他表示信息的设备。

[0082] 操作系统、应用程序、模块和 / 或数据的全部或部分也可被高速缓存在易失性存储器 1312 中。应该明白，所公开的体系结构可以用市场上可购得的各种操作系统或操作系统的组合来实现。

[0083] 用户可以通过一个或多个有线 / 无线输入设备，例如键盘 1338 和诸如鼠标 1340 等定点设备将命令和信息输入到计算机 1302 中。其他输入设备（未示出）可包括话筒、IR

遥控器、操纵杆、游戏手柄、指示笔、触摸屏等等。这些和其他输入设备通常通过耦合到系统总线 1304 的输入设备接口 1342 连接到处理单元 1308，但也可通过诸如并行端口、IEEE 1394 串行端口、游戏端口、USB 端口、IR 接口等其他接口连接。

[0084] 监视器 1344 或其他类型的显示设备也经由诸如视频适配器 1346 等接口连接到系统总线 1308。除了监视器 1344 之外，计算机通常包括诸如扬声器、打印机等其他外围输出设备（未示出）。

[0085] 计算机 1302 可使用经由有线和 / 或无线通信至诸如远程计算机 1348 等的一个或多个远程计算机的逻辑连接在网络化环境中操作。远程计算机 1348 可以是工作站、服务器计算机、路由器、个人计算机、便携式计算机、基于微处理器的娱乐设备、对等设备或其他常见的网络节点，并且通常包括相对于计算机 1302 描述的许多或所有元件，尽管为简明起见仅示出了存储器 / 存储设备 1350。所描绘的逻辑连接包括到局域网 (LAN)1352 和 / 或例如广域网 (WAN)1354 等更大的网络的有线 / 无线连接。这一 LAN 和 WAN 连网环境常见于办公室和公司，并且方便了诸如内联网等企业范围计算机网络，所有这些都可连接到例如因特网等全球通信网络。

[0086] 当在 LAN 连网环境中使用时，计算机 1302 通过有线和 / 或无线通信网络接口或适配器 1352 连接到 LAN 1356。适配器 1356 可以方便到 LAN 1352 的有线和 / 或无线通信，并且还可包括其上设置的用于使用适配器 1356 的无线功能进行通信的无线接入点。

[0087] 当在 WAN 连网环境中使用时，计算机 1302 可包括调制解调器 1358，或连接到 WAN 1354 上的通信服务器，或具有用于诸如通过因特网等通过 WAN1354 建立通信的其他装置。或为内置或为外置以及有线和 / 或无线设备的调制解调器 1358 经由输入设备接口 1342 连接到系统总线 1308。在联网环境中，相对于计算机 1302 所描绘的程序模块或其部分可以存储在远程存储器 / 存储设备 1350 中。应该理解，所示网络连接是示例性的，并且可以使用在计算机之间建立通信链路的其他手段。

[0088] 计算机 1302 可操作来使用 IEEE 802 标准家族来与有线和无线设备或实体进行通信，这些实体例如是在操作上安置成与例如打印机、扫描仪、台式和 / 或便携式计算机、个人数字助理 (PDA)、通信卫星、任何一件与无线可检测标签相关联的设备或位置（例如，电话亭、报亭、休息室）以及电话进行无线通信（例如，IEEE 802.11 空中调制技术）的无线设备。这至少包括 Wi-Fi（即无线保真）、WiMax 和蓝牙 ™ 无线技术。由此，通信可以如对于常规网络那样是预定义结构，或者仅仅是至少两个设备之间的自组织 (ad hoc) 通信。Wi-Fi 网络使用称为 IEEE 802.11x(a、b、g 等等）的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi 网络可用于将计算机彼此连接、连接到因特网以及连接到有线网络（使用 IEEE 802.3 相关介质和功能）。

[0089] 上面描述的包括所公开的体系结构的各示例。当然，描述每一个可以想到的组件和 / 或方法的组合是不可能的，但本领域内的普通技术人员应该认识到，许多其他组合和排列都是可能的。因此，该新颖体系结构旨在涵盖所有这些落入所附权利要求书的精神和范围内的更改、修改和变化。此外，就在说明书或权利要求书中使用术语“包括”而言，这一术语旨在以与术语“包含”在被用作权利要求书中的过渡词时所解释的相似的方式为包含性的。

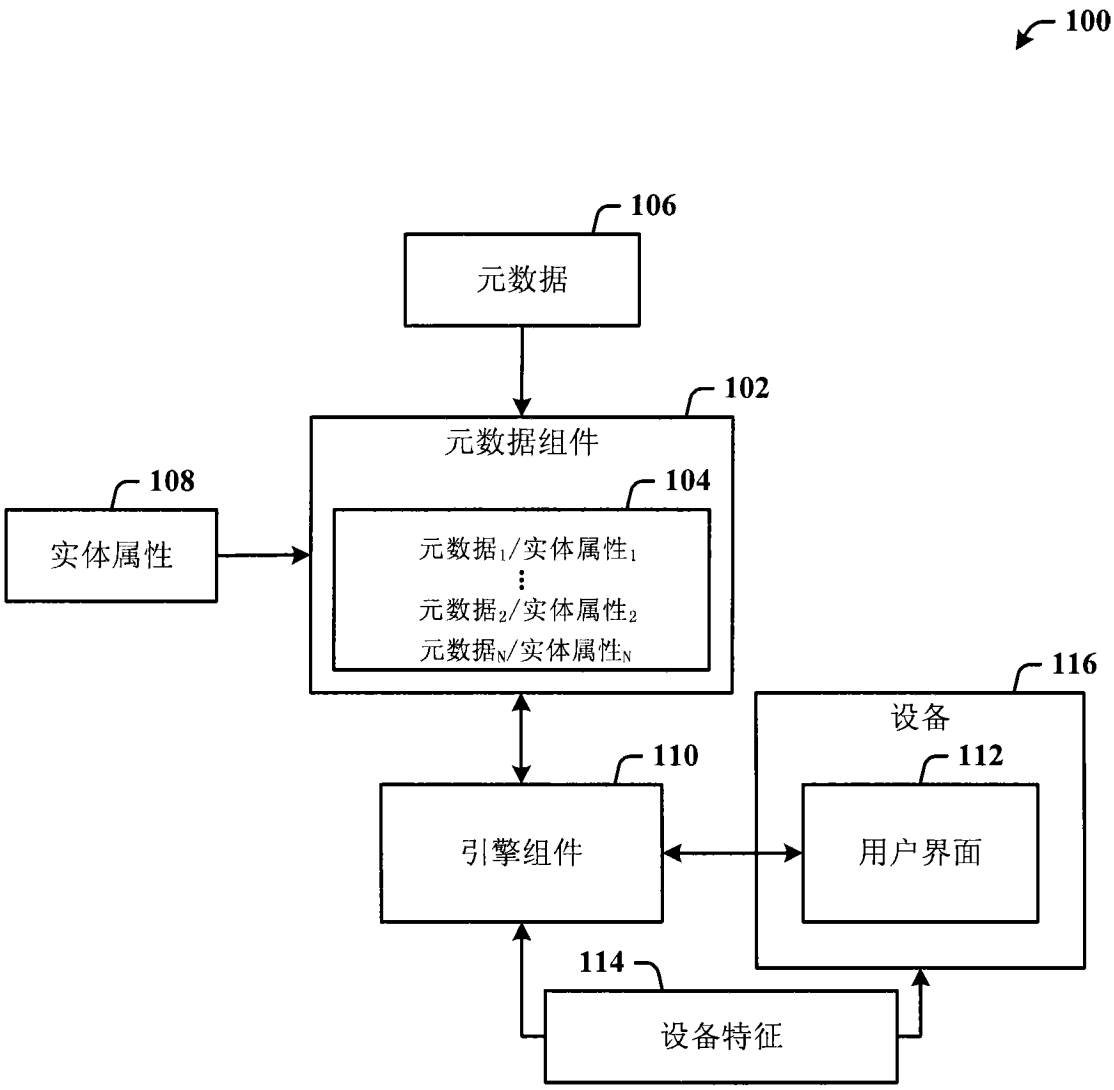

100
106
元数据
102
元数据组件
108
实体属性
104
元数据1/实体属性1
元数据2/实体属性2
元数据N/实体属性N
116
设备
110
112
引擎组件
用户界面
114
设备特征

图 1

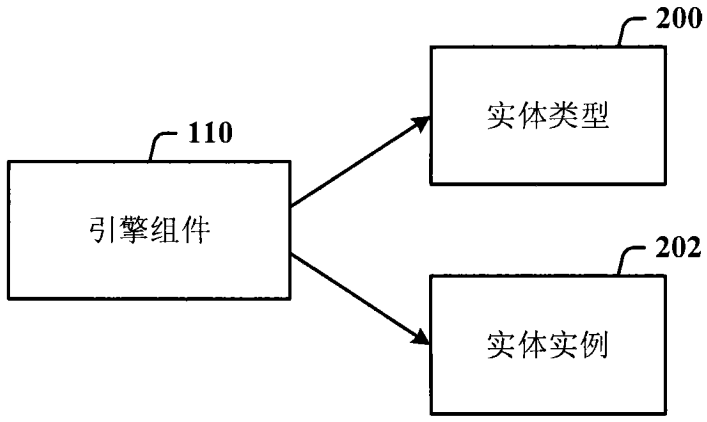

200
实体类型
110
引擎组件
202
实体实例

图 2

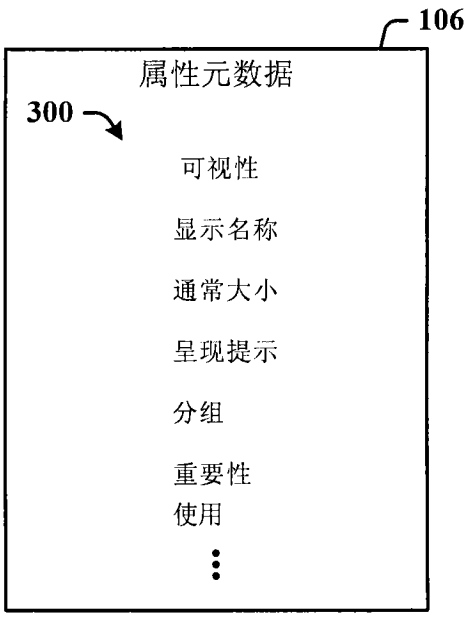


图 3

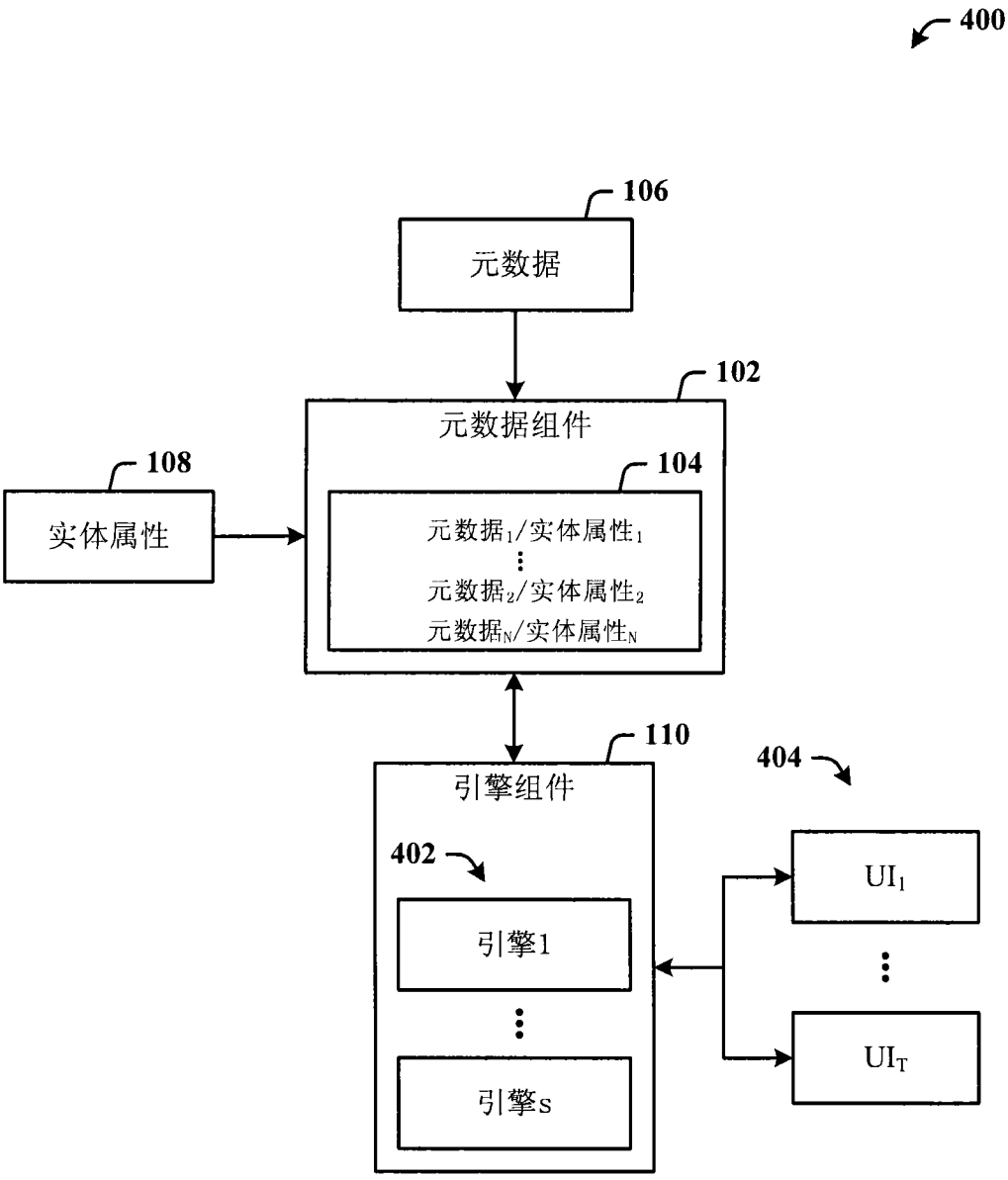

400
106
元数据
102
元数据组件
108
实体属性
104
元数据1/实体属性1
元数据2/实体属性2
元数据N/实体属性N
110
引擎组件
404
402
引擎1
引擎s
UI1
UIT

图 4

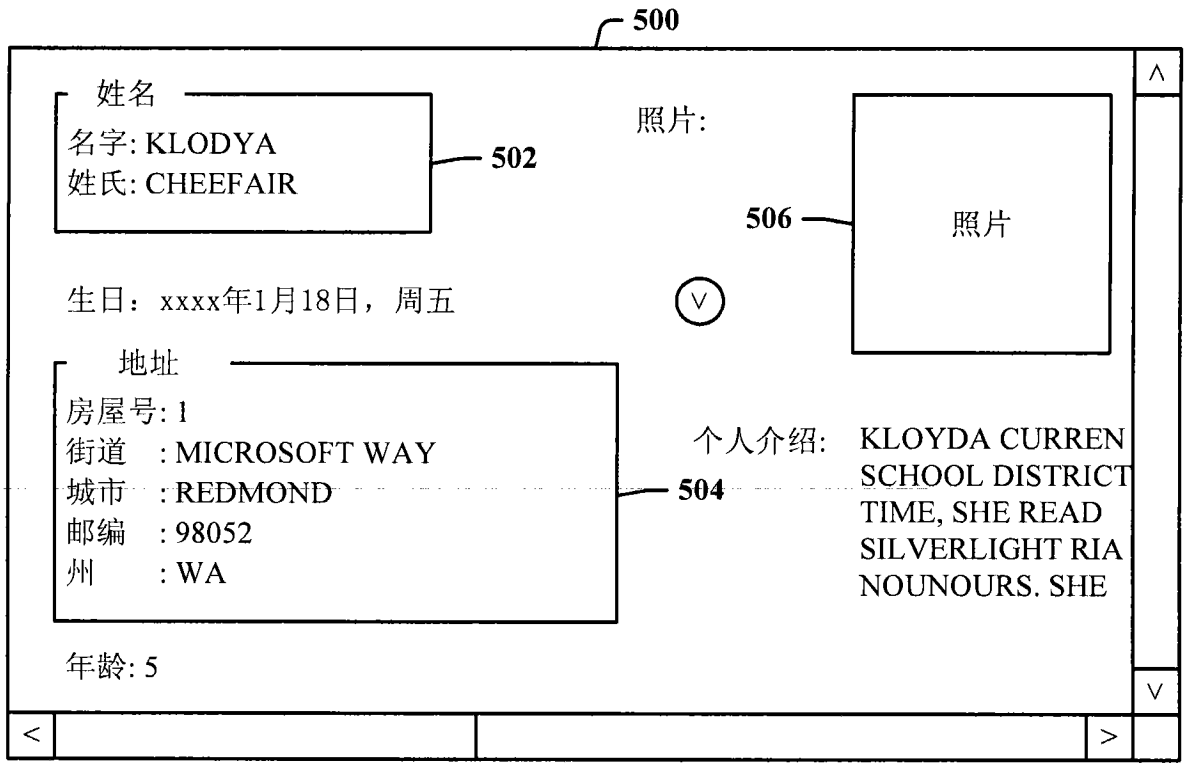

500
姓名
名字: KLODYA
姓氏: CHEEFAIR
502
照片:
506
照片
生日：xxxx年1月18日，周五
地址
房屋号: 1
街道 : MICROSOFT WAY
城市 : REDMOND
邮编 : 98052
州 : WA
504
个人介绍:
KLOYDA CURREN
SCHOOL DISTRICT
TIME, SHE READ
SILVERLIGHT RIA
NOUNOURS. SHE
年龄: 5
^
V
<
>

图 5

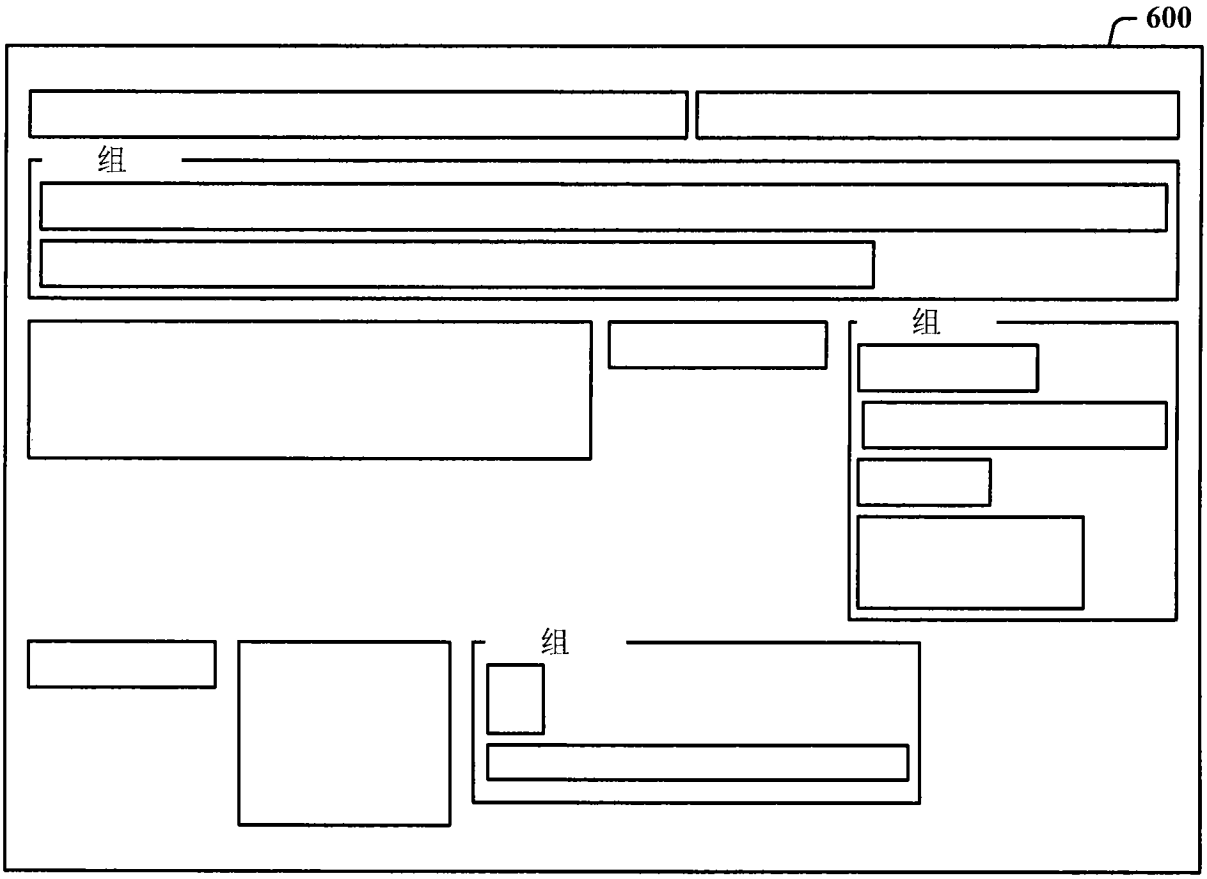


图 6

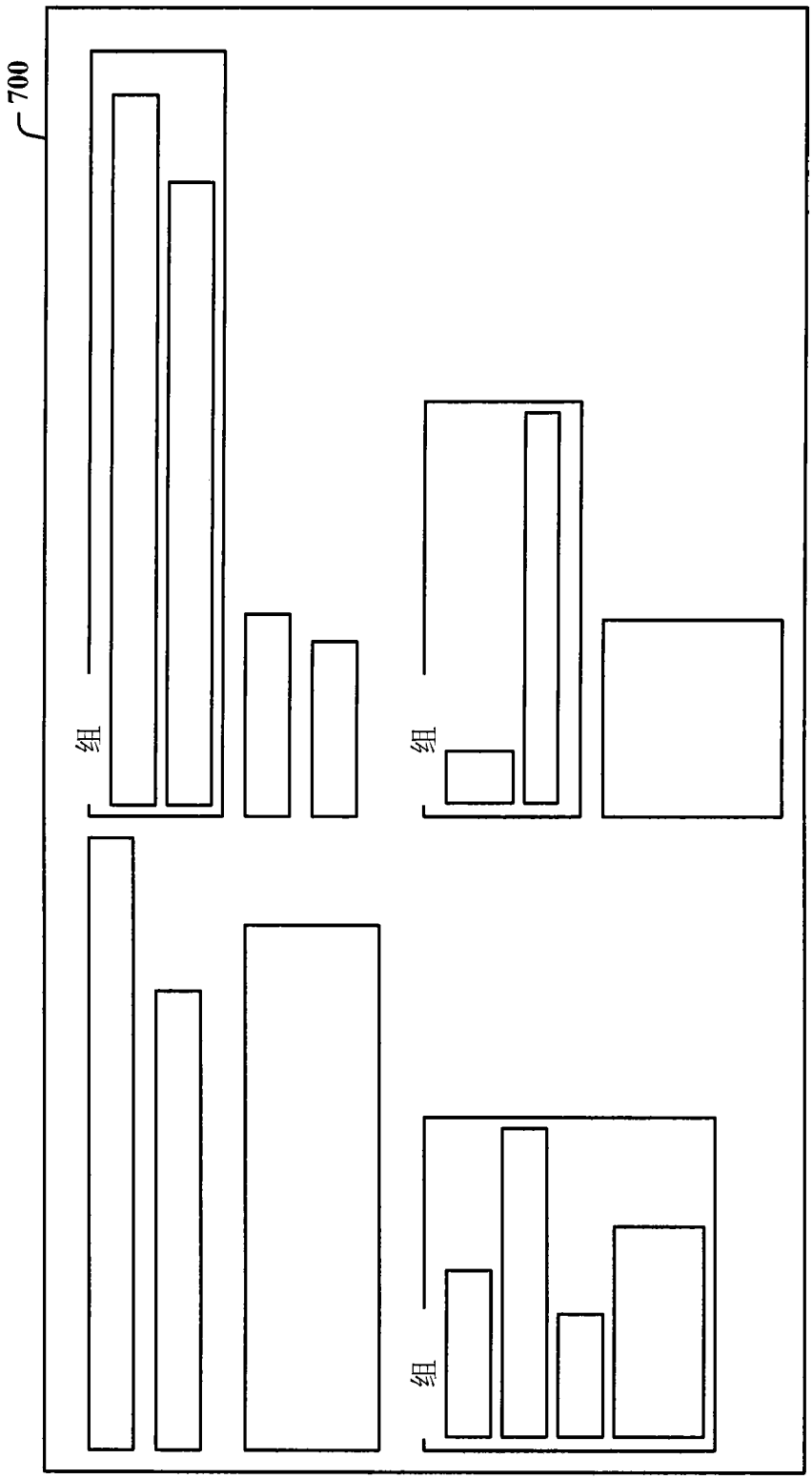


图 7

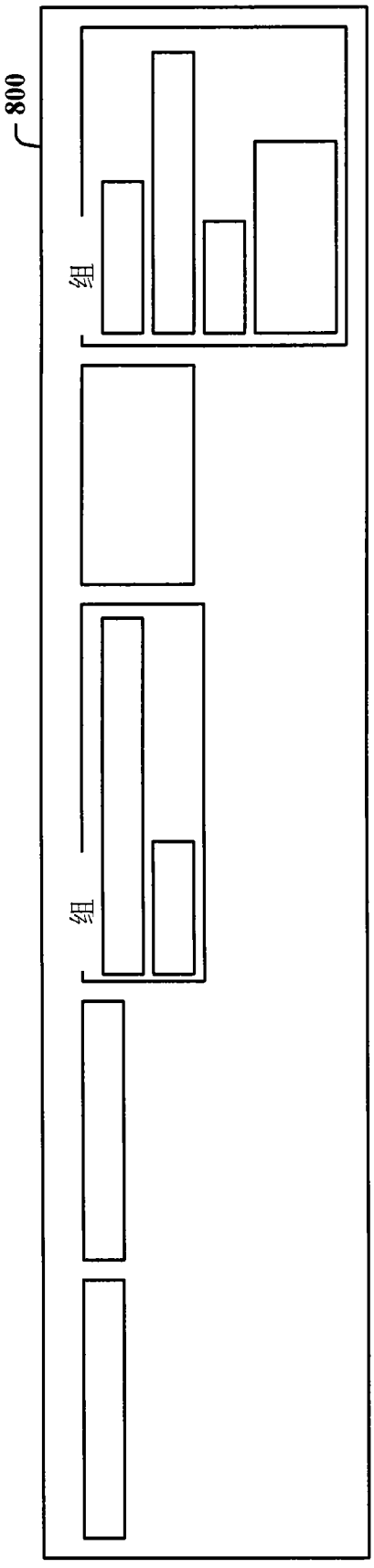


图 8

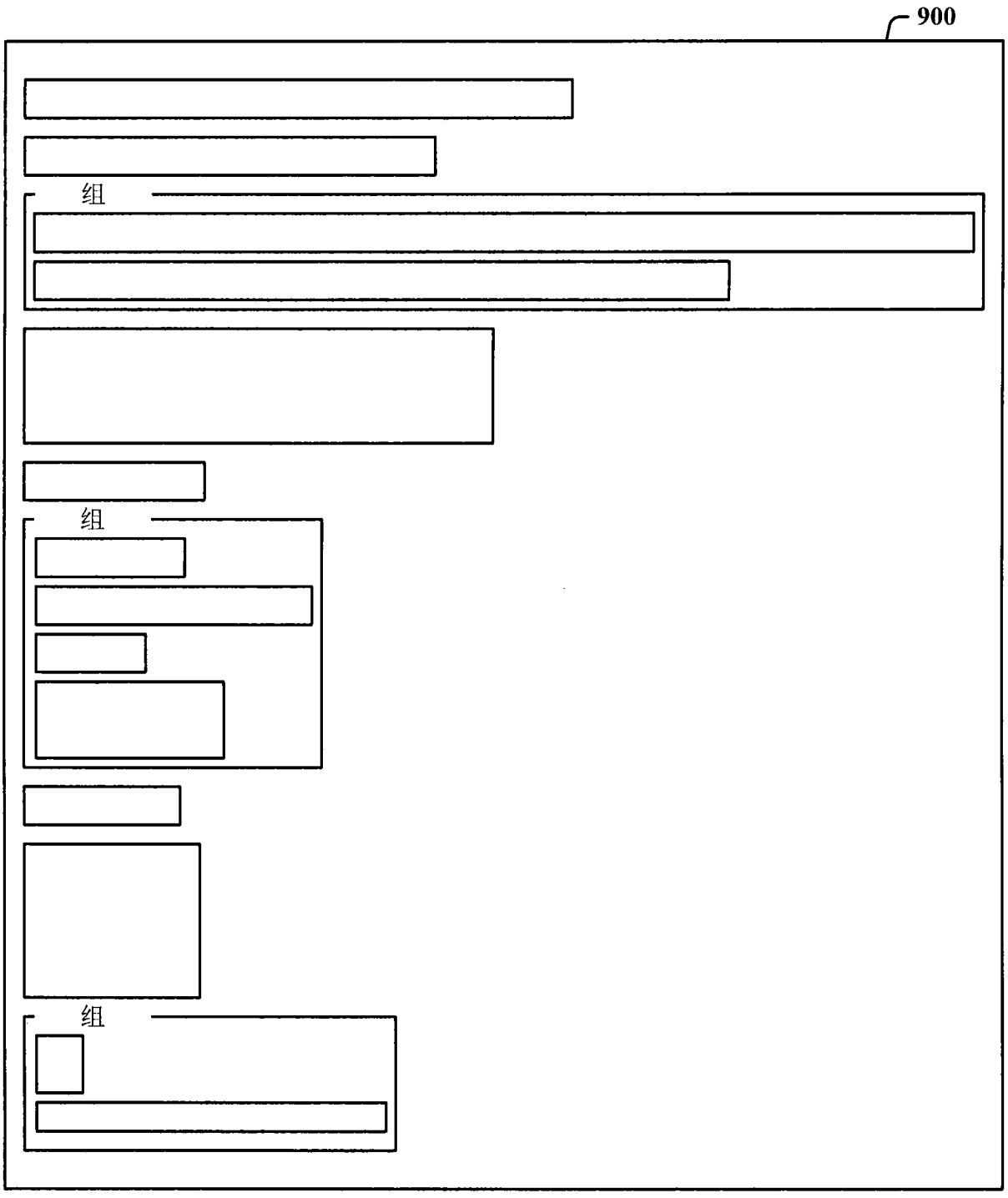


图 9

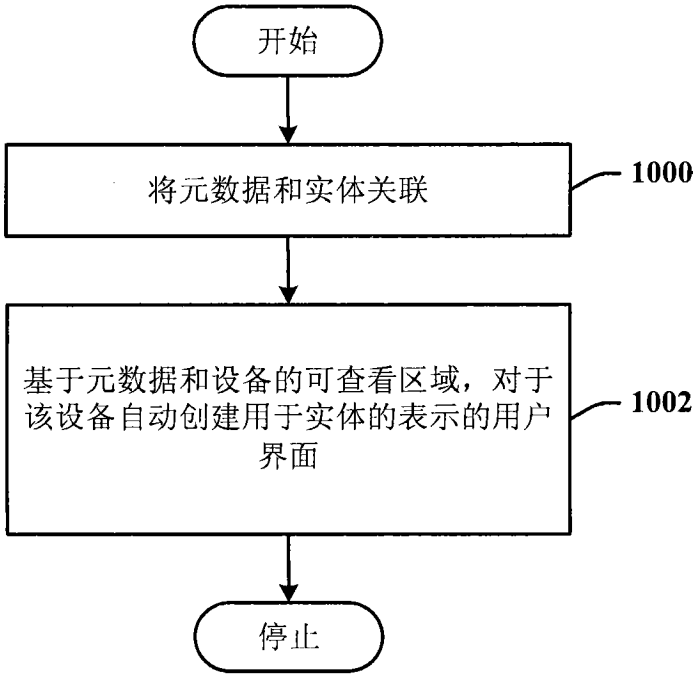

开始
将元数据和实体关联
1000
基于元数据和设备的可查看区域，对于该设备自动创建用于实体的表示的用户界面
1002
停止

图 10

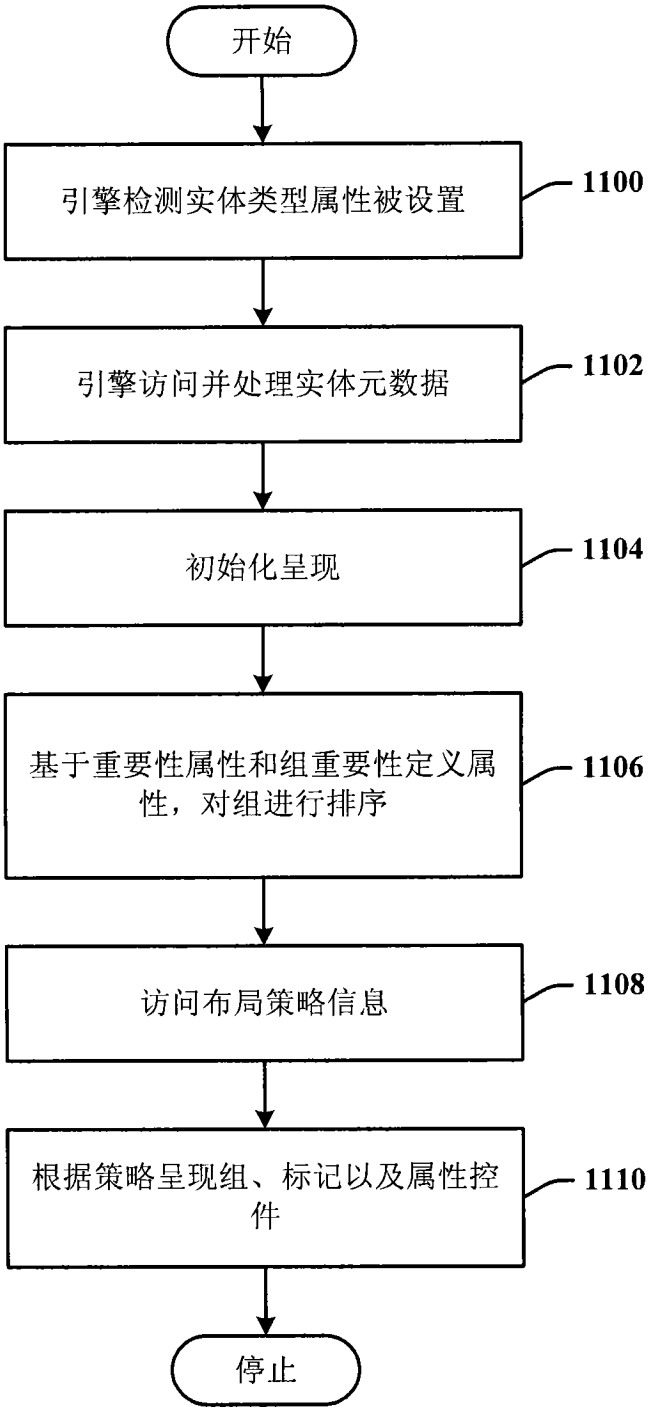

开始
引擎检测实体类型属性被设置
1100
引擎访问并处理实体元数据
1102
初始化呈现
1104
基于重要性属性和组重要性定义属性，对组进行排序
1106
访问布局策略信息
1108
根据策略呈现组、标记以及属性控件
1110
停止

图 11

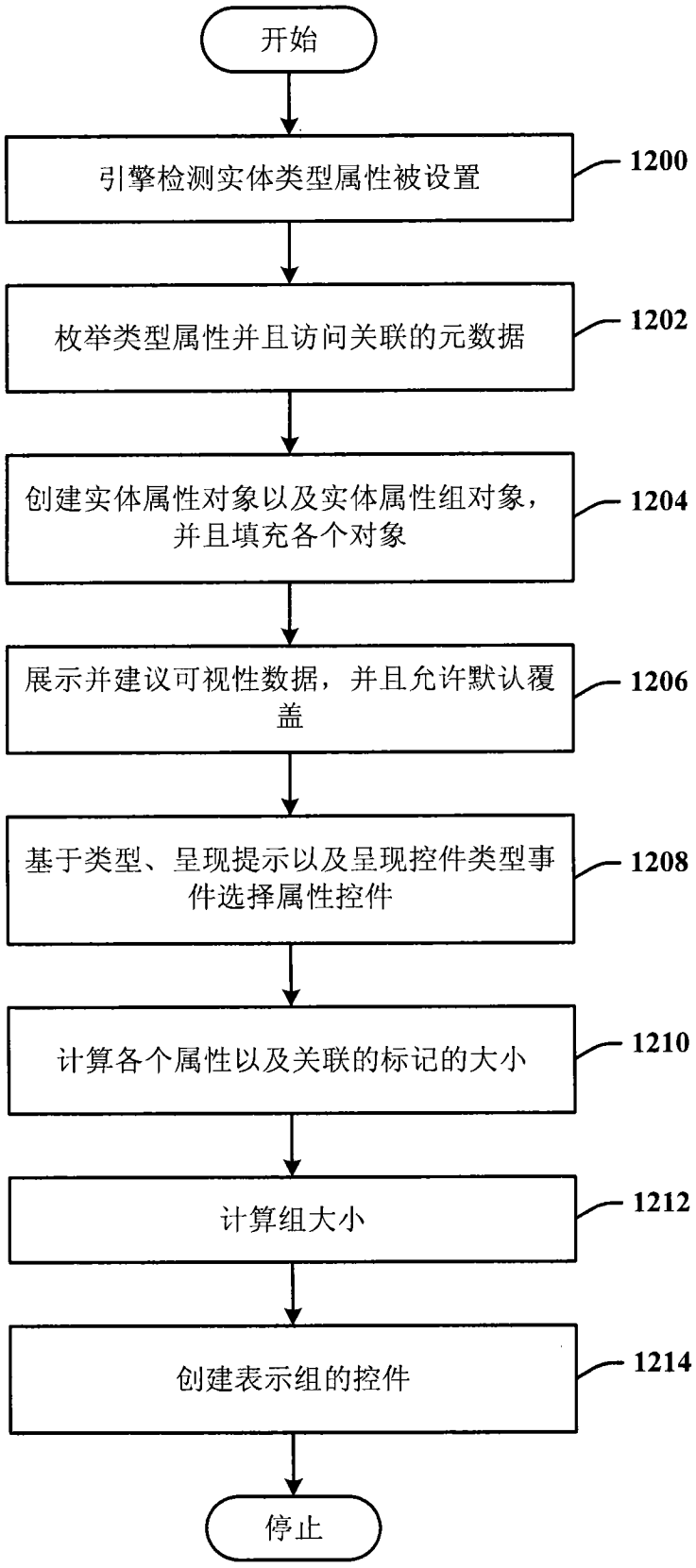

开始
引擎检测实体类型属性被设置
1200
枚举类型属性并且访问关联的元数据
1202
创建实体属性对象以及实体属性组对象，并且填充各个对象
1204
展示并建议可视性数据，并且允许默认覆盖
1206
基于类型、呈现提示以及呈现控件类型事件选择属性控件
1208
计算各个属性以及关联的标记的大小
1210
计算组大小
1212
创建表示组的控件
1214
停止

图 12

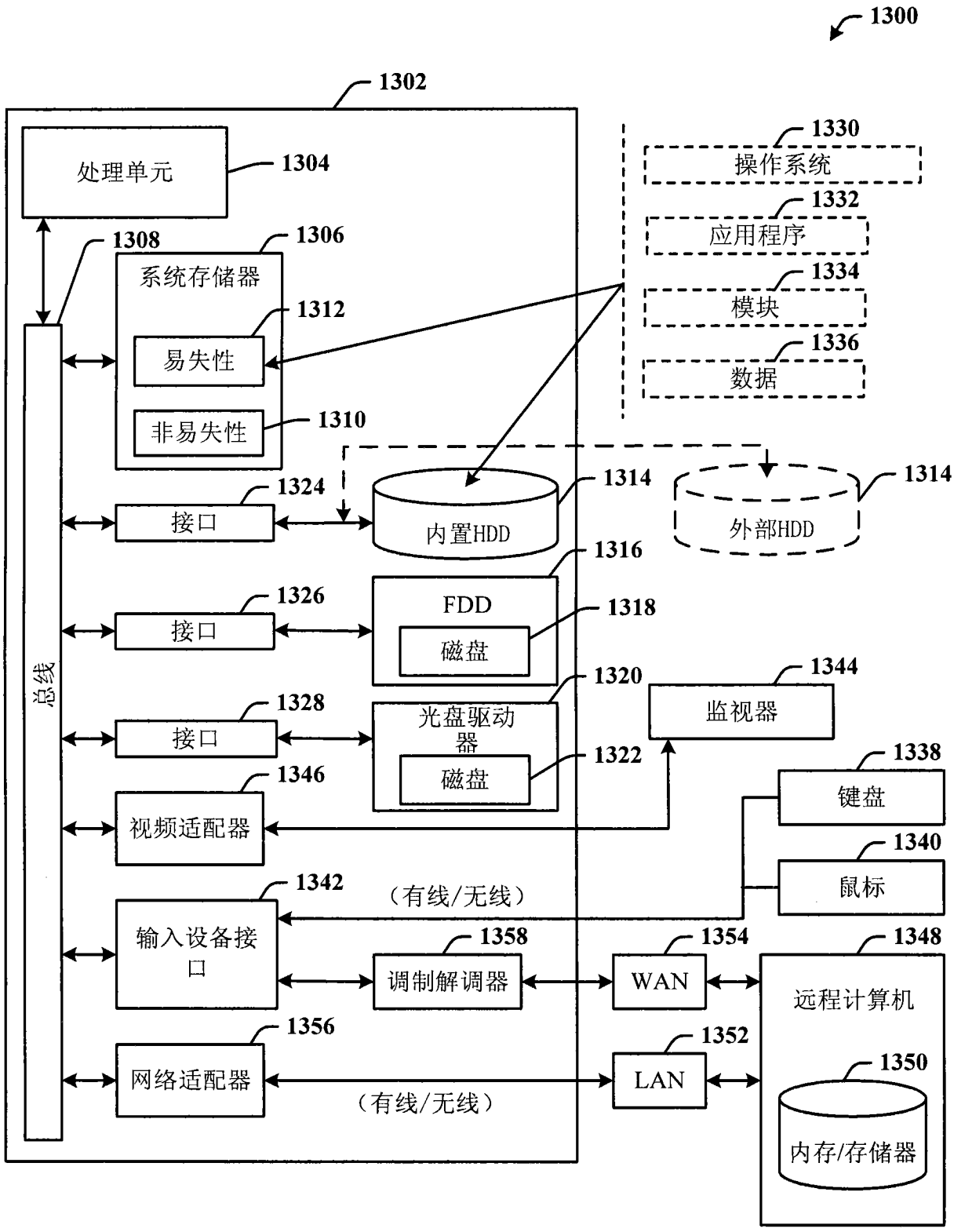

1300
1302
处理单元
1304
1308
1306
系统存储器
1312
易失性
非易失性
1310
总线
1330
操作系统
1332
应用程序
1334
模块
1336
数据
1324
接口
内置HDD
1314
外部HDD
1314
1316
1326
接口
FDD
磁盘
1318
1328
接口
光盘驱动器
磁盘
1320
1322
1344
监视器
1346
视频适配器
1338
键盘
1340
鼠标
1342
输入设备接口
（有线/无线）
1358
调制解调器
1354
WAN
1348
远程计算机
1356
网络适配器
（有线/无线）
1352
LAN
1350
内存/存储器

图 13