



(12) 发明专利申请

(10) 申请公布号 CN 111819535 A

(43) 申请公布日 2020. 10. 23

(21) 申请号 201880090401.7

(74) 专利代理机构 中国贸促会专利商标事务所
有限公司 11038

(22) 申请日 2018.12.27

代理人 李晓芳

(30) 优先权数据

15/857,026 2017.12.28 US

(51) Int.Cl.

G06F 8/35 (2006.01)

(85) PCT国际申请进入国家阶段日

G06F 8/38 (2006.01)

2020.08.27

G06F 9/54 (2006.01)

(86) PCT国际申请的申请数据

PCT/US2018/067671 2018.12.27

(87) PCT国际申请的公布数据

W02019/133709 EN 2019.07.04

(71) 申请人 阿韦瓦软件有限责任公司

地址 美国加利福尼亚

(72) 发明人 L·亚达弗 S·洛伊 C·帕特尔

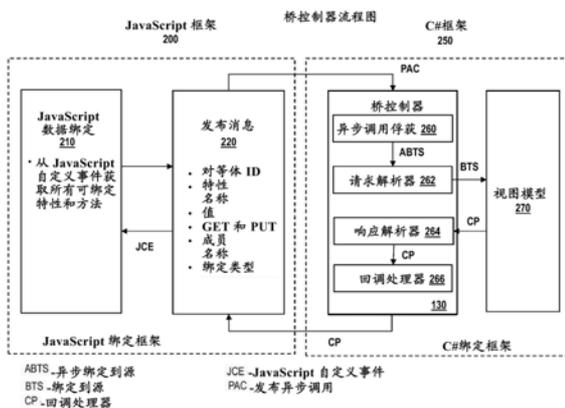
权利要求书3页 说明书10页 附图9页

(54) 发明名称

异步C#-JS数据绑定桥

(57) 摘要

一种方法和系统提供了用于在不同框架上实现的用户界面和数据模型之间的异步双向绑定。该系统包括数据控制器、数据模型控制器和桥控制器。数据控制器提供前端绑定框架，该前端绑定框架与用户设备的用户界面交互以管理用于用户设备上的视图的可绑定特性或方法。数据模型控制器提供管理数据模型的后端绑定框架，前端绑定框架和后端绑定框架是不同类型的框架。桥控制器为前端绑定框架和后端绑定框架之间的可绑定特性或方法实现异步双向绑定，以当用户界面处数据改变时更新数据模型中的可绑定特性或方法，和当数据模型处数据改变时更新用户设备上的视图。



1. 一种计算机实现的方法,包括:

提供具有数据控制器的前端绑定框架,该前端绑定框架与用户设备的用户界面交互以管理用于用户设备上的视图的可绑定特性或方法;

提供具有数据模型控制器的后端绑定框架,该后端绑定框架管理数据模型,前端绑定框架和后端绑定框架是不同类型的框架;以及

经由桥控制器为前端绑定框架的数据控制器和后端绑定框架之间的可绑定特性或方法实现异步双向绑定,以当用户界面处数据改变时更新数据模型中的可绑定特性或方法,和当数据模型处数据改变时更新用户设备上的视图。

2. 如权利要求1所述的计算机实现的方法,其中前端绑定框架是Javascript绑定框架,并且后端绑定框架是C#绑定框架。

3. 如权利要求1所述的计算机实现的方法,还包括:

在前端绑定框架处从用户界面接收具有针对绑定特性或方法的请求数据的绑定请求,其中所述实现包括:

异步地从数据控制器获得与绑定请求对应的发布消息,该发布消息至少包括与绑定请求相关联的对等体标识符、绑定类型和动词;

解析该发布消息;以及

基于经解析的发布消息将桥接请求调度到数据模型控制器,以至少根据绑定类型和动词对数据模型执行实现。

4. 如权利要求3所述的计算机实现的方法,还包括:

在前端绑定框架中针对绑定请求生成请求对象,该请求对象至少包括与绑定请求相关联的对等体标识符、绑定类型和动词;

将请求对象放在前端绑定框架中的回调队列上;以及

在前端绑定框架中为请求对象生成发布消息。

5. 如权利要求3所述的计算机实现的方法,其中所述实现还包括:

从数据模型接收作为对桥接请求的答复的数据模型响应,该数据模型响应包括与来自数据模型的绑定特性或方法相关联的值和对等体标识符;

在桥控制器处解析数据模型响应;以及

基于经解析的数据模型响应将具有来自数据模型的数据的桥接响应发送到前端绑定框架。

6. 如权利要求5所述的计算机实现的方法,还包括:

根据对等体标识符访问回调队列以识别与数据模型响应对应的请求对象;以及

响应于绑定请求,经由具有来自数据模型的数据的桥接响应为请求对象调用回调函数以更新用户界面的视图。

7. 如权利要求1所述的计算机实现的方法,其中提供操作和实现操作在移动设备上执行。

8. 一种计算机系统,用于在不同框架上实现的用户设备的用户界面和数据模型之间提供异步双向绑定,该计算机系统包括:

存储器;以及

一个或多个处理器,所述一个或多个处理器实现:

数据控制器,提供前端绑定框架,该前端绑定框架与用户设备的用户界面交互以管理用于用户设备上的视图的可绑定特性或方法;

数据模型控制器,提供后端绑定框架,该后端绑定框架管理数据模型,前端绑定框架和后端绑定框架是不同类型的框架;以及

桥控制器,为前端绑定框架的数据控制器和后端绑定框架之间的可绑定特性或方法实现异步双向绑定,以当用户界面处数据改变时更新数据模型中的可绑定特性或方法,和当数据模型处数据改变时更新用户设备上的视图。

9. 如权利要求8所述的计算机系统,其中前端绑定框架是Javascript绑定框架,并且后端绑定框架是C#绑定框架。

10. 如权利要求8所述的计算机系统,其中数据控制器被配置为在前端绑定框架处从用户界面接收具有针对绑定特性或方法的请求数据的绑定请求,以及

其中桥控制器还被配置为:

异步地从数据控制器获得与绑定请求对应的发布消息,该发布消息至少包括与绑定请求相关联的对等体标识符、绑定类型和动词;

解析该发布消息;以及

基于经解析的发布消息将桥接请求调度到数据模型控制器,以至少根据绑定类型和动词对数据模型执行实现。

11. 如权利要求10所述的计算机系统,其中桥控制器还被配置为:

在前端绑定框架中针对绑定请求生成请求对象,该请求对象至少包括与绑定请求相关联的对等体标识符、绑定类型和动词;

将请求对象放在前端绑定框架中的回调队列上;以及

在前端绑定框架中为请求对象生成发布消息。

12. 如权利要求10所述的计算机系统,其中桥控制器还被配置为:

从数据模型接收作为对桥接请求的答复的数据模型响应,该数据模型响应包括与来自数据模型的绑定特性或方法相关联的值和对等体标识符;

在桥控制器处解析数据模型响应;以及

基于经解析的数据模型响应将具有来自数据模型的数据的桥接响应发送到前端绑定框架。

13. 如权利要求12所述的计算机系统,桥控制器还被配置为:

根据对等体标识符访问回调队列以识别与数据模型响应对应的请求对象;以及

响应于绑定请求,经由具有来自数据模型的数据的桥接响应为请求对象调用回调函数以更新用户界面的视图。

14. 如权利要求8所述的计算机系统,其中数据模型控制器和桥控制器在包括存储器和处理器的移动设备上实现。

15. 一种存储可执行代码的有形存储器介质,该可执行代码在由一个或多个处理器执行时实现一种用于在不同框架上实现的用户界面和数据模型之间提供异步双向绑定的方法,该方法包括:

提供具有数据控制器的前端绑定框架,该前端绑定框架与用户设备的用户界面交互以管理用于用户设备上的视图的可绑定特性或方法;

提供具有数据模型控制器的后端绑定框架,该后端绑定框架管理数据模型,前端绑定框架和后端绑定框架是不同类型的框架;以及

经由桥控制器为前端绑定框架的数据控制器和后端绑定框架之间的可绑定特性或方法实现异步双向绑定,以当用户界面处数据改变时更新数据模型中的可绑定特性或方法,和当数据模型处数据改变时更新用户设备上的视图。

16.如权利要求15所述的有形存储器介质,其中前端绑定框架是Javascript绑定框架,并且后端绑定框架是C#绑定框架。

17.如权利要求15所述的有形存储器介质,其中所述方法还包括:

在前端绑定框架处从用户界面接收具有针对绑定特性或方法的请求数据的绑定请求,其中所述实现包括:

异步地从数据控制器获得与绑定请求对应的发布消息,该发布消息至少包括与绑定请求相关联的对等体标识符、绑定类型和动词;

解析该发布消息;以及

基于经解析的发布消息将桥接请求调度到数据模型控制器,以至少根据绑定类型和动词对数据模型执行实现。

18.如权利要求17所述的有形存储器介质,其中所述方法还包括:

在前端绑定框架中针对绑定请求生成请求对象,该请求对象至少包括与绑定请求相关联的对等体标识符、绑定类型和动词;

将请求对象放在前端绑定框架中的回调队列上;以及

在前端绑定框架中为请求对象生成发布消息。

19.如权利要求17所述的有形存储器介质,其中所述实现还包括:

从数据模型接收作为对桥接请求的答复的数据模型响应,该数据模型响应包括与来自数据模型的绑定特性或方法相关联的值和对等体标识符;

在桥控制器处解析数据模型响应;以及

基于经解析的数据模型响应将具有来自数据模型的数据的桥接响应发送到前端绑定框架。

20.如权利要求19所述的有形存储器介质,其中所述方法还包括:

根据对等体标识符访问回调队列以识别与数据模型响应对应的请求对象;以及

响应于绑定请求,经由具有来自数据模型的数据的桥接响应为请求对象调用回调函数以更新用户界面的视图。

异步C#-JS数据绑定桥

技术领域

[0001] 本公开涉及用于跨在不同平台(诸如Java Script框架和C-sharp(C#)框架)上实现的前端和后端进行数据和方法绑定的多平台桥接方法和系统。

背景技术

[0002] 应用(也称为“app”)中的数据绑定涉及模型和视图部件之间的数据自动同步。当模型中的数据改变时,视图反映这改变;并且当视图中的数据改变时,模型也会以异步方式更新。这个过程立即且自动发生,这确保模型和视图始终得以更新。

发明内容

[0003] 目前没有在多平台环境中支持双向绑定的机制,多平台环境诸如具有基于Xamarin的混合多平台app,其中前端和后端在不同的框架上实现。为了解决这些和其它问题,提供了混合多平台绑定桥,其采用桥控制器来促进与用户设备的用户界面(UI)交互的前端绑定框架和与数据模型(例如,视图模型或模型)交互的后端绑定框架之间的异步双向通信。以这种方式,桥控制器可以将具有用于可绑定特性或方法的请求数据的绑定请求从UI经由前端绑定框架传递到后端绑定框架,以在数据模型上实现,并且可以将来自数据模型的用于可绑定特性或方法的数据响应从后端绑定框架传递到前端绑定框架,以更新用户设备的视图部件和UI。

[0004] 绑定桥可以帮助快速开发或设计在HTML/JavaScript用户界面(UI)和C# ViewModel(例如,用Xamarin C#、HTML和JavaScript)之间用模型-视图-视图模型(MVVM)架构模式开发的多平台app,并且可以充分利用特定于平台的基础设施在独立或离线app中快速执行代码并改善数据绑定性能。

[0005] 在示例实施例中,计算机实现的方法和系统涉及提供具有数据控制器的前端绑定框架,该前端绑定框架与用户设备的用户界面交互以管理用于用户设备上的视图的可绑定特性或方法;提供具有数据模型控制器的后端绑定框架,该后端绑定框架管理数据模型,前端绑定框架和后端绑定框架是不同类型的框架;以及经由桥控制器为前端绑定框架的数据控制器和后端绑定框架之间的可绑定特性或方法实现异步双向绑定,以当用户界面处数据改变时更新数据模型中的可绑定特性或方法,和当数据模型处数据改变时更新用户设备上的视图。前端绑定框架可以是Javascript绑定框架,而后端绑定框架可以是C#绑定框架。

[0006] 该计算机实现的方法和系统还可以涉及在前端绑定框架处从用户界面接收具有针对绑定特性或方法的请求数据的绑定请求。该实现操作可以包括异步地从数据控制器获得与绑定请求对应的发布消息。该发布消息可以至少包括与绑定请求相关联的对等体标识符、绑定类型和动词。该实现操作还可以包括解析该发布消息;并基于经解析的发布消息将桥接请求(例如,来自桥控制器的请求)调度到数据模型控制器,以至少根据绑定类型和动词对数据模型执行实现。

[0007] 该计算机实现的方法和系统还可以涉及在前端绑定框架中针对绑定请求生成请

求对象；将请求对象放在前端绑定框架中的回调队列上；并在前端绑定框架中为请求对象生成发布消息。请求对象可以至少包括与绑定请求相关联的对等体标识符、绑定类型和动词。

[0008] 在另一个示例中，实现操作还可以涉及接收数据模型响应（例如，来自数据模型的响应）作为对桥接请求的答复；在桥控制器处解析数据模型响应；并基于经解析的数据模型响应将具有来自数据模型的数据的桥接响应发送到前端绑定框架。数据模型响应可以包括与来自数据模型的绑定特性或方法相关联的值和对等体标识符。

[0009] 此外，该计算机实现的方法和系统还可以涉及根据对等体标识符访问回调队列以识别与数据模型响应对应的请求对象；并响应于绑定请求，经由具有来自数据模型的数据的桥接响应为请求对象调用回调函数以更新用户界面的视图。

附图说明

[0010] 结合附图解释各种示例实施例的描述，其中：

[0011] 图1图示了根据本公开的示例实施例的示例计算环境的框图，该示例计算环境通过桥控制器来实现绑定桥以促进跨多平台环境的数据绑定。

[0012] 图2图示了根据本公开的示例实施例的示例桥控制器的功能框图，该示例桥控制器促进跨多平台环境的数据绑定。

[0013] 图3图示了根据本公开的示例实施例的示例桥控制器的更详细的功能框图，该示例桥控制器促进跨多平台环境的数据绑定。

[0014] 图4图示了根据本公开的示例实施例的跨多平台环境实现的示例数据绑定处理的流程图。

[0015] 图5图示了根据本公开的示例实施例的示例请求协议定义。

[0016] 图6图示了根据本公开的示例实施例的示例响应协议定义。

[0017] 图7图示了根据本公开的示例实施例的示例请求格式。

[0018] 图8图示了根据本公开的示例实施例的示例响应格式。

[0019] 图9图示了根据本公开的示例实施例的示例支持的可绑定数据类型。

[0020] 图10至图12图示了用户界面的示例页面或屏幕，该示例页面或屏幕允许用户在计算机设备上设置或配置用于双向绑定的app。

具体实施方式

[0021] 提供了一种系统和方法，以在多平台环境中使用绑定桥来促进诸如数据和/或方法绑定之类的绑定，该多平台环境针对前端和后端采用不同的框架。在示例中，可以使用 Javascript (JS) 来实现前端框架，并且可以使用 C# (也称为 C-sharp) 来实现后端框架，反之亦然。前端框架包括具有数据控制器的前端数据绑定框架，该前端数据绑定框架例如经由 HTML 通过前端应用与用户设备的用户界面 (UI) 交互，以在 UI 上提供 (一个或多个) 视图及视图的更新。后端框架包括具有数据模型控制器的后端数据绑定框架，以管理和更新数据模型 (例如，视图模型或模型)。经由桥控制器实现的绑定桥以异步方式在前端 (例如，JS/HTML 前端) 和后端 (例如，C# 后端) 之间提供双向通信和绑定，诸如数据和/或方法绑定。

[0022] 桥控制器与前端绑定框架和后端绑定框架通信。以这种方式，绑定桥可以例如在

两个不同的绑定框架(诸如C#和JS)之间提供异步通信层,并且例如在HTML/JS前端和C#后端之间创建可移植的绑定层。因此,桥控制器允许HTML/JS前端和C#后端之间进行直接通信,以实现异步双向数据绑定。例如,绑定桥可以被配置为在Android中跨C#-JS层注入多种数据类型(例如,JS(Java.lang)对象)并对消息进行排队;针对iOS以IWKScriptMessage句柄从JS俘获消息;并在Windows 10中执行WinRT部件排队。下面将参考附图更详细地描述绑定桥及其一般操作的示例。

[0023] 绑定桥可以帮助快速开发或设计在HTML/JavaScript(UI)用户界面与C#ViewModel(例如,用Xamarin C#、HTML和JavaScript)之间用模型-视图-视图模型(MVVM)架构模式或其它视图-模型架构模式开发的多平台应用,并且可以充分利用特定于平台的基础设施来在独立或离线app中快速执行代码并改善数据绑定性能。此外,双向绑定基础设施的异步性质使UI释放出来以在绑定请求挂起的同时执行其它任务。

[0024] I. 架构

[0025] 图1图示了根据本公开的示例实施例的示例计算环境100的框图,该示例计算环境通过桥控制器来实现绑定桥以促进跨多平台环境的数据绑定。如图1中所示,计算环境100包括用户界面(UI)110(例如,用户界面或web视图)、前端框架的前端绑定框架的数据控制器120、桥控制器130、以及后端框架的后端绑定框架的数据模型控制器140。如前面所讨论的,前端框架可以是JS框架,并且后端框架可以是C#框架。在这种环境中,前端和后端应用(或app)可以在不同的框架上被开发和被实现,诸如对于其中UI上的视图和数据模型(该数据模型管理应用的数据、规则和逻辑)在不同平台上被管理的架构模式。一种这样的模式是MVVM,它进一步将数据模型分为视图模型(例如,用于视图的逻辑和规则)和包含用于应用的数据的模型。视图模型是视图和模型之间的中介。

[0026] 数据控制器120与UI 110交互以管理和更新UI 110上的视图,以当UI 110上的视图中的数据已经改变时接收更新并促进数据模型(例如,取决于架构模式的视图模型和模型)的更新。在这个示例中,数据控制器120是JS前端框架的JS控制器,并且管理用于应用(包括前端app)的JS数据部件。

[0027] 数据模型控制器140管理和更新数据模型中的数据。例如,数据模型控制器140被配置为当UI 110处与一个或多个可绑定特性和/或方法相关联的数据已改变时更新数据模型中的数据。数据模型控制器140还被配置为促进视图的更新作为对请求(例如,对GET动作的请求)的答复或者当数据模型中与可绑定特性或方法相关联的数据改变时促进视图的更新。在这个示例中,数据模型控制器是C#后端框架的C#控制器。

[0028] 桥控制器130是前端框架和后端框架之间的绑定桥的一部分,并且被配置为在前端框架和后端框架的部件(诸如数据控制器120和数据模型控制器140)之间提供双向异步通信。如将在下面更详细地描述的,桥控制器130经由异步调用从前端绑定框架获得与每个绑定请求相关联的发布消息,解析发布消息并与数据模型控制器140通信以在目标对象(例如,数据模型)处实现绑定请求的特性绑定和/或方法绑定的动作(例如,GET或PUT)。在数据模型处采取动作之后,数据模型控制器140将具有相关数据的数据模型响应发送给桥控制器130。桥控制器130解析该响应,并通过前端绑定框架为回调队列中的对应请求实现回调。回调调用回调函数以将响应数据(例如,来自数据模型的针对可绑定特性或方法的数据)发送到数据控制器120作为对请求的答复。然后,数据控制器120相应地更新UI上的视图。

[0029] 在这个示例中,计算机环境可以是用户设备,诸如移动设备。可以在用户设备上实现具有数据控制器120的前端绑定框架、桥控制器130和具有数据模型控制器140的后端绑定框架。用户设备可以至少包括(一个或多个)处理器(例如,CPU)、存储器和(一个或多个)网络接口设备或NID以经由一个或多个网络与其它计算机交互、输入设备(例如,触摸屏/显示屏、键盘、按钮、音频输入设备和/或其它输入设备)和输出设备(例如,显示器、音频输出设备和/或其它输出设备)。存储器可以存储数据和计算机可执行应用,包括操作系统、双向绑定桥和其它应用或程序。用户设备可以是移动设备(例如,Android或iOS智能电话、平板电脑或设备、膝上型计算机等)、台式计算机或其它计算机实现的系统。

[0030] 当用户设备经由其(一个或多个)处理器执行存储在存储器中的应用时,(一个或多个)计算机的处理器和存储器提供框架、绑定和通信桥以及本文所述的功能和特征,包括不同框架之间的异步双向通信(例如,用于在不同框架之间实现异步通信的通信层)以及用于跨不同框架(例如,不同的前端和后端框架)实现的多平台应用的异步双向绑定(例如,数据和/或方法绑定)。

[0031] 以上描述了计算机环境100的示例。用户设备可以经由(一个或多个)网络与其它计算机(例如,(一个或多个)服务器等)交互,以实现应用并更新在远程位置处维护的数据,包括数据模型中的数据等等。双向绑定处理和功能,包括各种控制器操作,也可以跨不同的计算机实现,例如,用户设备可以与(一个或多个)其它计算机交互。其它计算机可以跨(一个或多个)网络被连接,并且至少包括(一个或多个)处理器(例如,CPU)、存储器和NID,以促进跨有线网络、无线网络、互联网或其组合的网络通信。存储器可以存储数据和包括操作系统的计算机可执行应用。

[0032] 图2图示了示例桥控制器及其功能部件的高级功能框图,该示例桥控制器及其功能部件促进跨多平台环境的数据绑定。如图2中所示,多平台环境包括前端框架200和后端框架250。在这个示例中,前端框架200包括具有JS数据控制器等的JS绑定框架,而后端框架250包括具有数据模型控制器的C#绑定框架。桥控制器130在前端绑定框架和后端绑定框架之间提供双向通信层,以启用它们之间的异步双向通信和绑定。

[0033] 在前端框架200的前端绑定框架中,JS数据绑定210可以从JS自定义事件获取所有可绑定特性和/或方法。例如,当UI发起绑定请求时,诸如当UI处的视图中的数据改变或用于视图的数据被请求时,该处理可以开始。最终为请求的可绑定特性和/或方法生成发布消息200,并且发布消息200可以包括诸如对等体ID、特性名称、值、GET和PUT、成员名称和绑定类型之类的参数。然后,发布消息200作为发布异步调用被异步地发送到数据模型,在这种情况下,经由桥控制器260被发送到后端框架250的视图模型270,以从数据模型更新和/或请求数据。

[0034] 具体而言,桥控制器130实现异步调用俘获260以获得发布消息220,并通过发起请求解析器262解析该发布消息来实现异步绑定到源。然后,桥控制器130基于所解析的发布消息中的请求数据向后端框架250的后端绑定框架的数据模型控制器发送请求(也称为“桥接请求”)。然后,数据模型控制器可以相应地采取动作,以在数据模型(例如,视图模型270)上实现可绑定特性和/或方法的所请求的动作。这些动作可以包括针对可绑定特性或方法从视图模型270更新和/或获得数据。一旦在视图模型270上更新、访问和/或检索与特性和/或方法相关联的数据,就实现回调处理,该回调处理中具有数据或更新后的数据的响应(也

称为“数据模型响应”)从视图模型270发送到了桥控制器130。桥控制器130获得并用响应解析器264解析来自数据模型响应的响应数据。然后,当针对请求(例如,具有同一对等体ID的请求)返回响应时,回调处理器266获得回调,并使用例如JS自定义事件自动地将来自视图模型270的相关的响应数据向数据控制器发送。JS数据绑定210获得JS自定义事件,然后可以根据数据模型(例如,视图模型或模型)处的任何改变来更新UI处的视图。因此,通过使用桥控制器130及其部件实现的绑定桥,在UI与数据模型(例如,视图模型270)之间提供可绑定特性和/或方法的双向异步绑定。因而,当UI处视图中数据发生改变时,可以更新数据模型;当数据模型中数据发生改变时,可以在UI处更新视图。

[0035] 图3图示了具有桥控制器130及其功能部件的示例绑定桥的更详细的功能框图,该桥控制器130及其功能部件促进跨多平台环境(诸如,不同的前端框架和后端框架)的双向异步数据绑定。在这个示例中,用于前端框架的前端绑定框架包括数据部件,诸如JS部件310,该部件由数据控制器管理,以在与用户设备的UI交互的前端上获取和管理所有可绑定特性和方法。后端框架的后端绑定框架也包括部件,这些部件由数据模型控制器管理。后端框架的部件包括数据映射和执行360以及数据模型,诸如具有视图模型实现382的视图模型380。桥控制器130促进前端绑定框架的JS部件310与后端绑定框架的数据映射和执行360之间的异步双向通信,从而促进了绑定。

[0036] 作为JS绑定框架的数据部件的JS部件310包括所有可绑定特性和方法,诸如特性绑定312和方法绑定314。JS部件310通过桥控制器130从例如视图模型380(例如,C#视图模型)的数据模型中获取特性和方法,并创建与该视图的绑定。JS部件310可以包括两个部分:特性绑定和方法绑定。特性绑定将值绑定到元素的特性以修改其外观或行为;创建与视图和模型的双向数据绑定;包括诸如Nullable、Bool、Number、String、String Array、如JSON(JavaScript对象表示法)String的Complex Type、Item List Array和Item List Array之类的特性;将特性值与JS表达式以及输入元素/属性绑定;并支持数据绑定的异步模式。方法绑定将方法绑定到视图模型方法;调用带有或不带有参数的方法;并从该方法返回值;并支持方法绑定的异步模式。

[0037] 桥控制器130使用回调事件从数据模型(例如,C#视图模型)获取所有可绑定特性和方法,并将绑定特性和方法传递给数据控制器。桥控制器130包括请求创建器320,以针对从UI接收的请求(诸如绑定请求)创建对象;回调队列322保持请求数据;响应解析器264解析来自视图模型380的数据模型响应;并且获取回调324获取与回调队列322中的请求数据相关联的回调,以便调用回调函数。

[0038] 请求创建器320生成保持协议定义的JavaScript对象,诸如例如:

[0039] //bindingtype:通用方法、视图模型的方法和视图模型的特性//verb:GET和PUT

[0040] //membername:方法名

[0041] //value:用于视图模型的特性的函数名称或方法参数

[0042] //memberparameter:设置器的方法参数的值

[0043] //peerid:对等体对象标识符

[0044] 回调队列322保持所有请求数据,这将异步地将发布消息与回调函数值一起发送到数据模型控制器(例如,C#控制器)。当从来自对等体C#对象的视图模型380接收到响应数据时,获取回调324获取该回调。

[0045] 桥控制器130还包括特性设置器和获取器330、特性信息绑定332、方法信息绑定334、数据类型转换336、视图模型状态338、特性改变均衡器340和调度自定义事件344。桥控制器130可以响应于诸如绑定请求或与数据模型相关的其它请求之类的请求而实现这些功能。

[0046] 数据映射和执行360保持所有脚本消息实例，数据映射和执行360被用于通过C#反射生成特性和方法绑定。数据映射和执行360将包含特性和方法绑定实例的相同集合。数据映射和执行360包括：绑定类型362、方法标识符364、参数值366、对等体标识符368、特性回调370、方法回调374和引发特性改变事件374。

[0047] 视图模型380保持所有C#特性和方法实现，这些实现通过直接来自视图而挂钩。

[0048] 对象和对象参数的其它描述在以下针对JS绑定框架被描述，并且可以被用于促进本文描述的绑定桥系统和方法的实现。除了上述功能之外，绑定框架还可以实现其它功能：订阅特性改变、退订特性改变、状态管理器、当前视图模型、状态改变、关于特性改变、创建绑定对象、数据类型转换等等。这些功能对象和一些参数的描述如下。

[0049] • 对等体对象标识符-保持任何C#对象对等体的唯一标识符

[0050] • DataTypes-在特性绑定中保持所支持的数据类型

```
dataTypes: {
```

```
    nullable: 0,
```

```
    bool: 1,
```

```
    string: 3,
```

```
    stringArray: 4,
```

[0051]

```
    complexType: 5,
```

```
    complexTypeAsString: 6,
```

```
    complexTypeObjectArray: 7,
```

```
    itemListArray: 8
```

```
}
```

[0052] • 绑定类型-支持特性和方法绑定类型两者

[0053] • 订阅特性改变-将JavaScript对象设置为特性改变事件的订阅者

[0054] • 退订特性改变-从特性改变订阅列表中移除JavaScript对象 • 状态管理器-保持视图模型状态对象

- [0055] • 当前视图模型-从C#模型获取当前视图模型标识符
- [0056] • 状态改变-调用视图路由器以进入不同状态
- [0057] • 关于特性改变-当C#视图模型中特性改变时触发,并在JavaScript VM上调用特性改变回调
- [0058] • 创建绑定对象-生成所有绑定特性和方法、保持特性和方法信息、使用getter和setter函数从C#视图模型获取和设置特性值,并将方法信息绑定到C#方法绑定
- [0059] • 数据类型转换-将C#的基本数据类型转换成JavaScript数据类型,诸如String、Number和Boolean

[0060] 提供了JS前端框架和C#后端框架的这些功能和参数作为示例。取决于用于实现多平台应用的框架的类型,可以使用其它标识符、数据类型、绑定类型和对象功能以及其它命名约定来实现桥绑定系统和方法。

[0061] II. 示例处理

[0062] 图4示出了根据本公开的示例实施例的跨多平台环境实现的示例数据绑定处理400的流程图。如图4中所示,在步骤402处,由用户设备的用户界面(UI)发起特性和/或方法绑定请求(例如,PropertyAndMethodBindingRequest)。在这个示例中,请求是在前端HTML/JS框架中通过UI的HTML部件实现的HTML请求。可以例如基于用户通过UI进行的改变、选择或操作来发起请求。绑定请求可以包括请求的源(例如,源对象)的标识符、(一个或多个)绑定的类型、其名称和与之相关联的(一个或多个)值、以及要采取的任何动作,包括在请求的目标(例如,目标对象,诸如数据模型)处更新数据或从请求的目标获取数据。在图5和图7中示出了用于请求的实例的协议定义和格式的示例,下面将对其进行进一步描述。

[0063] 在参考404处,数据控制器(例如,JS控制器)从UI接收绑定请求,并且作为响应,以异步方式生成发布消息(例如,GeneratePostMessage)。例如,数据控制器可以为绑定请求中的请求数据创建对象,例如JS请求对象,将该对象放在诸如回调队列之类的队列中以维护请求数据,随后针对队列中的请求数据异步地生成发布消息。在参考406处,绑定桥的桥控制器使用例如调用俘获来俘获该发布消息,并解析发布消息中的请求数据以经由桥接请求传送到数据模型控制器(例如,Web视图控制器)。在参考408处,数据模型控制器向数据模型(例如,ViewModel)发送RequestDispatch,以关于该数据模型实现可绑定特性和/或方法的所请求的动作(例如,GET或PUT等)。在参考410处,数据模型将响应数据(例如,ResponseData)发送回数据模型控制器,数据模型控制器进而在参考412处在桥控制器处引发回调事件并返回具有来自数据模型的响应数据的数据模型响应。在参考414处,桥控制器实现ResponseParser操作,该操作解析数据模型响应并获取响应数据,并从队列中获取对对应请求(例如,队列中的对等请求对象)的回调。该回调通过使用例如JavaScript自定义事件将响应数据发送到数据控制器的数据部件(例如,视图部件)来自动调用返回操作。在参考416处,数据控制器用来自数据模型的响应数据更新UI处的视图以及被管理的数据部件,例如特性和方法绑定。

[0064] 虽然用JS-C#框架描述了示例绑定桥处理,但是应当理解的是,绑定桥可以是用户与其它类型的前端和后端框架以及多平台app。

[0065] III. 示例协议和格式

[0066] 参考图5至图9示出并描述可以在具有双向绑定桥的图1-图4的计算环境中使用的

请求和响应协议的参数和格式的示例。

[0067] 图5图示了请求协议500参数及其定义的示例。在这个示例中,参数可以包括bindingType、verb、Member Name、value、memberparameter和peerid.bindingType定义绑定的性质,诸如特性绑定和/或方法绑定。verb定义要采取的动作,诸如GET和PUT。membername是方法名。value是成员值。memberparameter是用于设置器的方法参数的值。peerid是对等体对象标识符。

[0068] 图6图示了响应协议600参数的示例。在这个示例中,参数可以包括id、membername和value.id是对等体对象标识符,并允许将响应与请求(例如,对等体对象)相关联。membername是方法名。value是成员值。

[0069] 图7图示了示例请求格式。如图7中所示,请求的示例格式700可以包括:

```
[0070] {  
[0071]   "bindingType":bindingType,  
[0072]   "verb":verb,"membername":membername,  
[0073]   "value":value,  
[0074]   "memberparameter":"memberparameter",  
[0075]   "peerid":peerid  
[0076] }
```

[0077] 图8图示了示例响应格式。如图8中所示,响应的示例格式800可以包括:

```
[0078] {  
[0079]   "id":id,  
[0080]   "membername":membername,  
[0081]   "value":value  
[0082] }
```

[0083] 图9图示了示例支持的可绑定数据类型。如图9中所示,可绑定数据类型900可以包括例如Nullable、Bool、Number、String、String Array、如JSON String的Complex Type和Item List Array。

[0084] 如本文所述,绑定桥可以被配置为与用户设备上的诸如移动app或其它app之类的应用一起使用。在这个示例中,绑定桥可以提供双向数据绑定和关于特性改变通知。通过绑定桥,在多平台app中,影响模型的任何与数据相关的改变都可以立即传播到(一个或多个)匹配的视图,并且在(一个或多个)视图中(诸如例如由用户)进行的任何改变可以立即反映在底层模型中。当app数据改变时,UI相应地改变,反之亦然。参考图10-图12示出提供页面(或屏幕)以允许用户在计算机设备上设置或配置双向绑定的应用的用户界面的示例。

[0085] 如图10中所示,被称为“设备”的示例页面1000提供有输入,例如文本框和按钮控件,以在设备上设置特性和方法绑定。默认情况下,可以从数据模型中设置设备名称,并在文本框中反映出来。类似地,如果在文本框中改变数据,它将立即反映在数据模型中。

[0086] 在这个示例中,可以使用具有不同数据类型的特性绑定将文本与控件绑定。下面使用AngularJS框架描述双向绑定的示例:

```
[0087] <div ng-app="app"ng-controller"MainCtrl">
```

```
[0088] <input ng-model="mymodel,data">
```

[0089] </div>

[0090] 虽然该示例使用Angular JS框架,但数据绑定桥与任何JavaScript框架兼容。

[0091] 方法绑定可以订阅HTML按钮控件,HTML按钮控件可以感测事件并向模型传播触发的操作。下面提供了示例:

```
<div ng-app="app" ng-controller="MainCtrl">
```

```
<button ng-click="mymodel.Continue()">
```

[0092] <h2 ng-bind = "mymodel.data"></h2>

```
button>
```

```
</div>
```

[0093] 在这个示例中,ng-app、ng-controller和ng-bind是angular指令;mymodel.data是双向绑定数据,它与数据绑定桥和C#模型同步;而mymodel.Continue()是事件绑定,它直接订阅C#方法绑定。

[0094] 图11图示了可以被提供以配置与一个控件绑定的称为“同步服务器”的示例页面1100,这也反映在其它控件中。页面1100可以具有文本框或其它图形输入,以及“继续”按钮等。最初,文本框可以没有数据,并且“继续”按钮被禁用。但是,一旦在文本框中输入数据,就可以启用“继续”按钮,并且还可以在后端更新数据模型。在这个示例中,绑定可以基于设置的条件与多个控件一起使用。

[0095] 图12图示了示出也可以被提供的多个列表控件的示例页面1200。列出的数据可以基于从属选择的数据和特性改变事件。多列表页面1200可以包括图形下拉框或其它输入以输入(例如,选择选项、输入文本等)相关信息,诸如例如基本(Base)、用户、角色和密码。当任何数据被选择时,基于所选择的数据启用其它控件。

[0096] 在这个示例中,移动app可以被用于为工厂运营、维护管理、生产跟踪和合规性应用来启用工作流、数据收集和通用任务管理。

[0097] 以上仅作为示例协议和格式提供。响应和请求协议和格式可以包括其它参数,或者使用其它命名约定或语法,这具体取决于包括(一个或多个)框架的类型的多平台环境。

[0098] 应当理解的是,本文描述的设备、系统和方法仅作为示例提供。各种设备、控制器或系统(诸如在附图中)可以在一个或多个计算机系统上实现,该计算机系统包括(一个或多个)处理器、存储器、包括网络接口设备(NID)的通信设备以及其它部件以提供或控制服务和操作,包括不同框架之间的双向通信和绑定。

[0099] 还应当理解的是,本文公开和教导的示例实施例易于进行多种各种各样的修改和替代形式。因此,单数术语(诸如但不限于“一个”等)的使用并不意在限制物品的数量。此外,用于针对测试场景、指令、命令行开关、表达式、对象、属性、线程名、文件名、目录名的脚本文件或脚本的各个部分的脚本语法的命名约定以及本文使用的其它命名约定是作为示例提供的,并且可以赋予不同的名称或标签。

[0100] 将认识到的是,结合所公开实施例的各方面的实际、真实商业应用的开发将要求

许多特定于实现的决定,以实现开发人员针对商业实施例的最终目标。这种特定于实现的决定可以包括并且有可能不限于遵守与系统相关、与业务相关、与政府相关的以及其它约束,这些约束可以因具体实现、位置和随时间而变。虽然从绝对意义上说开发人员的工作可能是复杂且耗时的,但是对于受益于本公开的本领域技术人员来说,这样的工作仍然是例行的工作。

[0101] 使用本文提供的描述,示例实施例可以通过使用标准编程和/或工程技术被实现为机器、处理或制品,以产生编程软件、固件、硬件或其任意组合。

[0102] 具有计算机可读程序代码的(一个或多个)任何结果程序可以实现在一个或多个计算机可用介质(诸如驻留存储器设备、智能卡或其它可移动存储器设备,或传输设备)上,从而制造根据实施例的计算机程序产品或制品。照此,如本文所使用的术语“制品”和“计算机程序产品”旨在涵盖永久地或临时地存在于任何计算机可用介质上或存在于传输这种程序的任何传输介质中的计算机程序。

[0103] 如上面所指示的,存储器/存储设备可以包括但不限于盘、固态驱动器、光盘、诸如智能卡之类的可移动存储器设备、SIM、WIM、诸如RAM、ROM、PROM之类的半导体存储器等等。传输介质包括但不限于经由无线通信网络(例如,射频(RF)通信、互联网、内联网、基于电话/调制解调器的网络通信、硬连线/有线通信网络,卫星通信以及其它固定或移动网络系统/通信链路进行的传输。

[0104] 虽然已经图示和描述了本公开的特定实施例和应用,但是应当理解的是,本公开不限于本文公开的精确构造和组成,并且在不脱离如所附权利要求定义的本发明的前提下,各种修改、改变和变化可以从前面的描述显而易见。

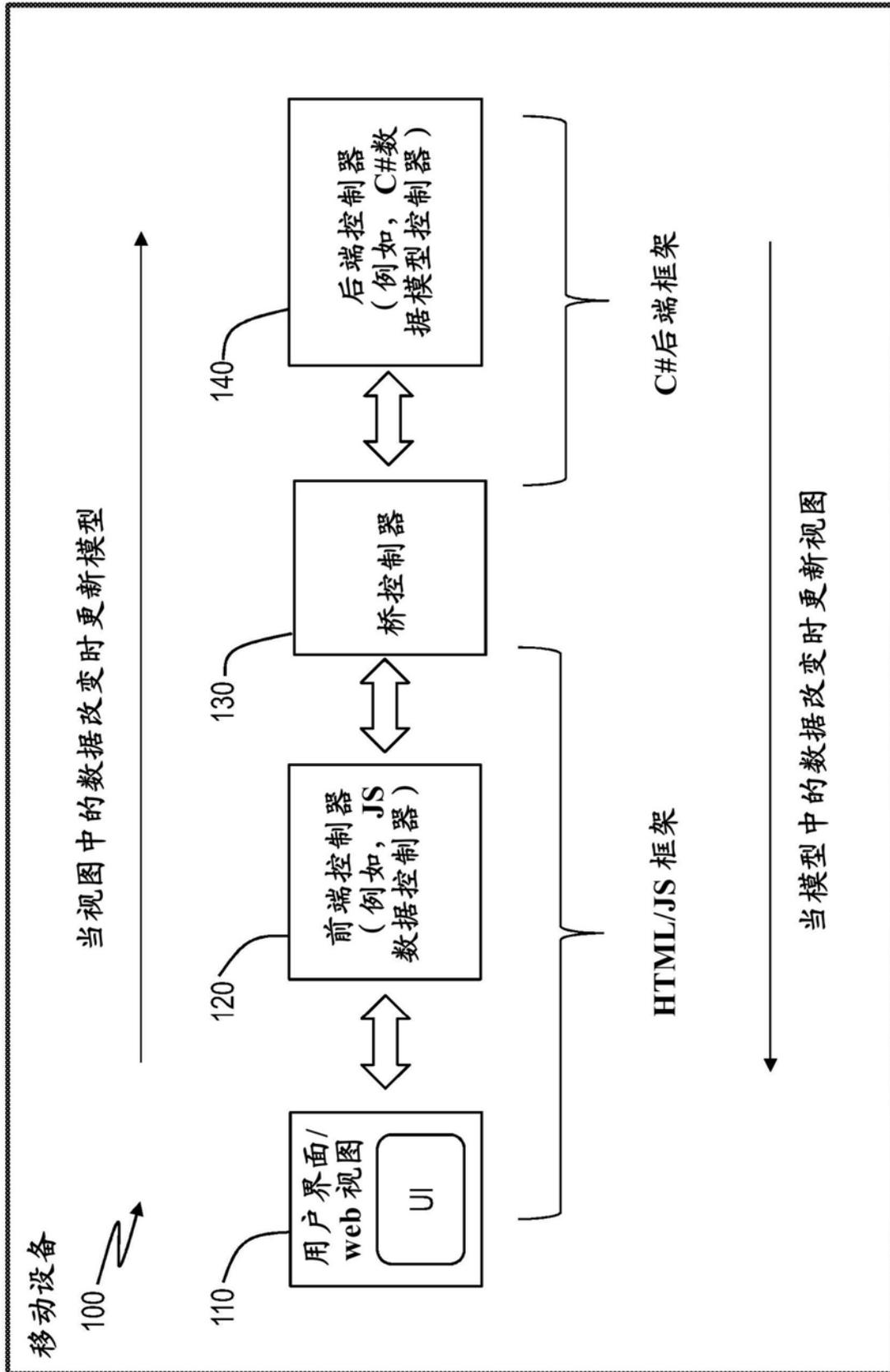


图1

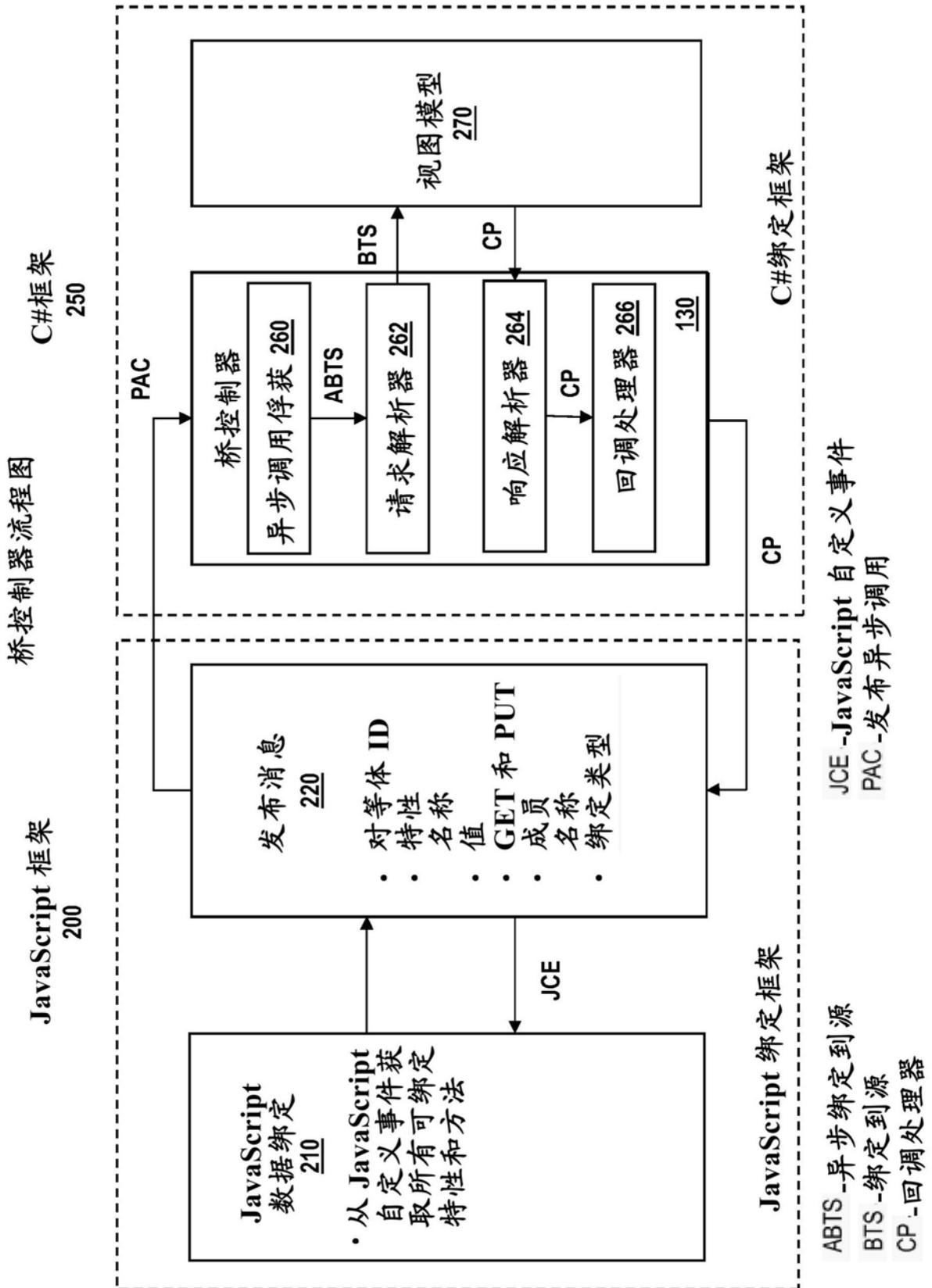


图2

桥控制器流程图

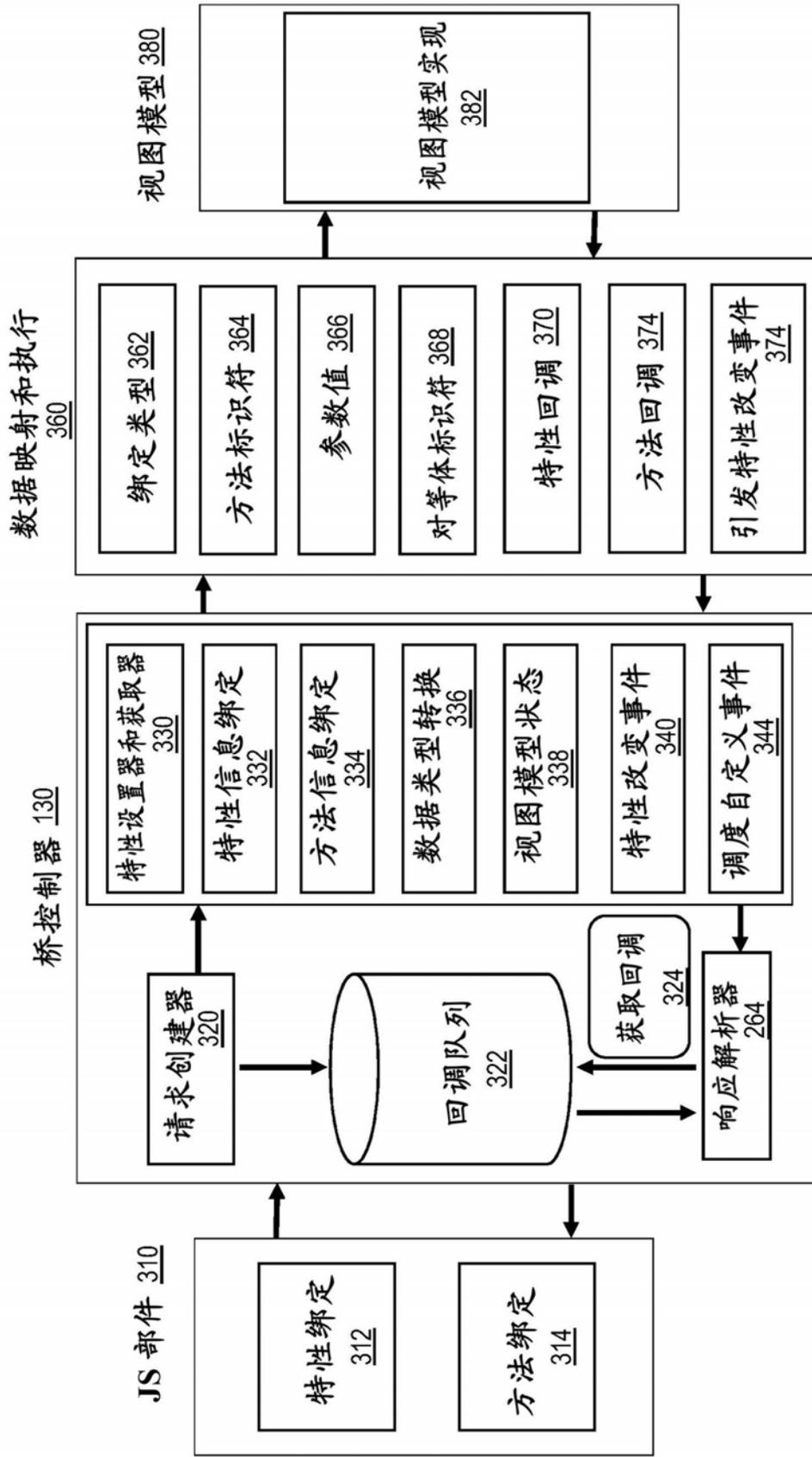
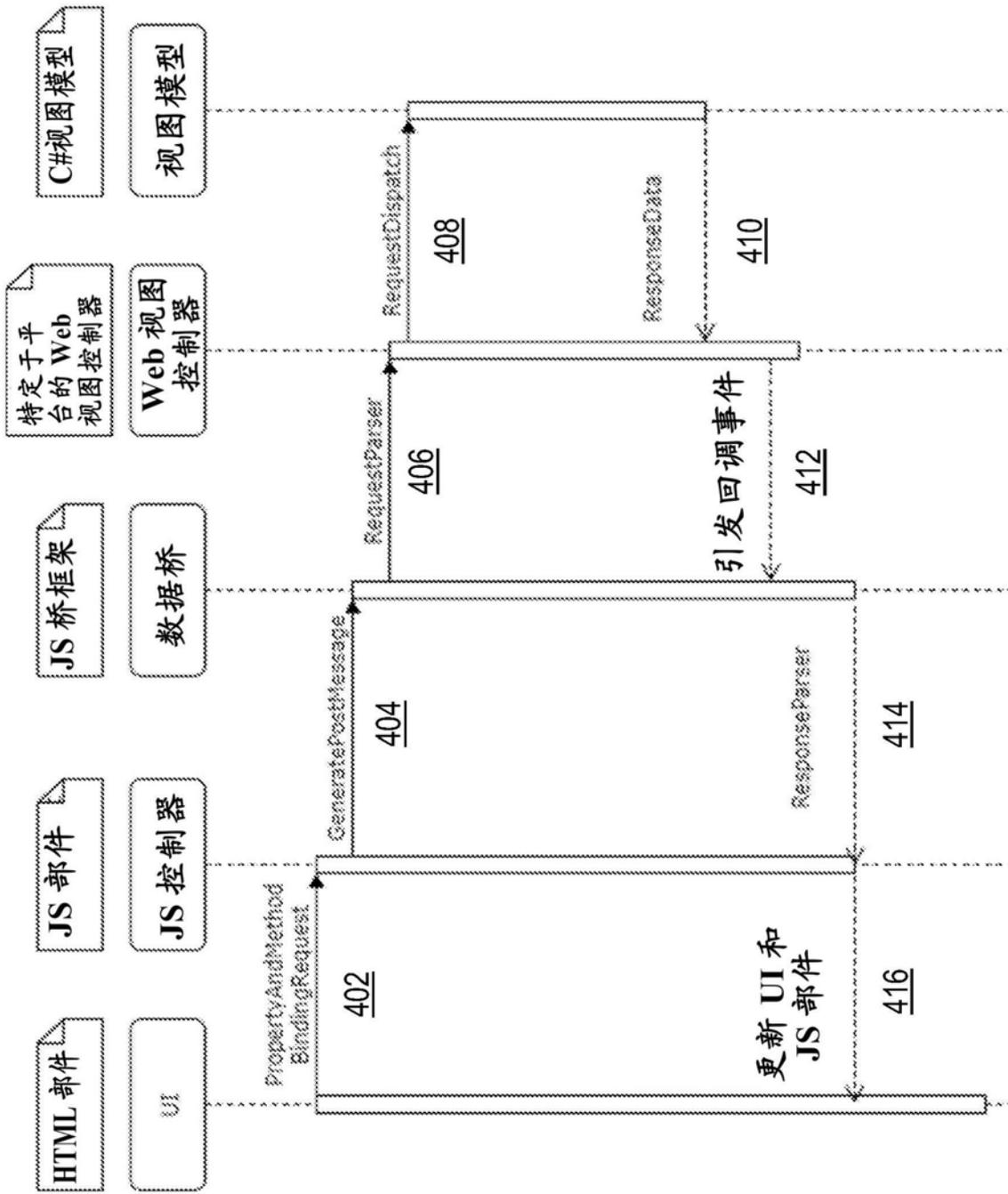


图3



400 ↗

流程图

图4

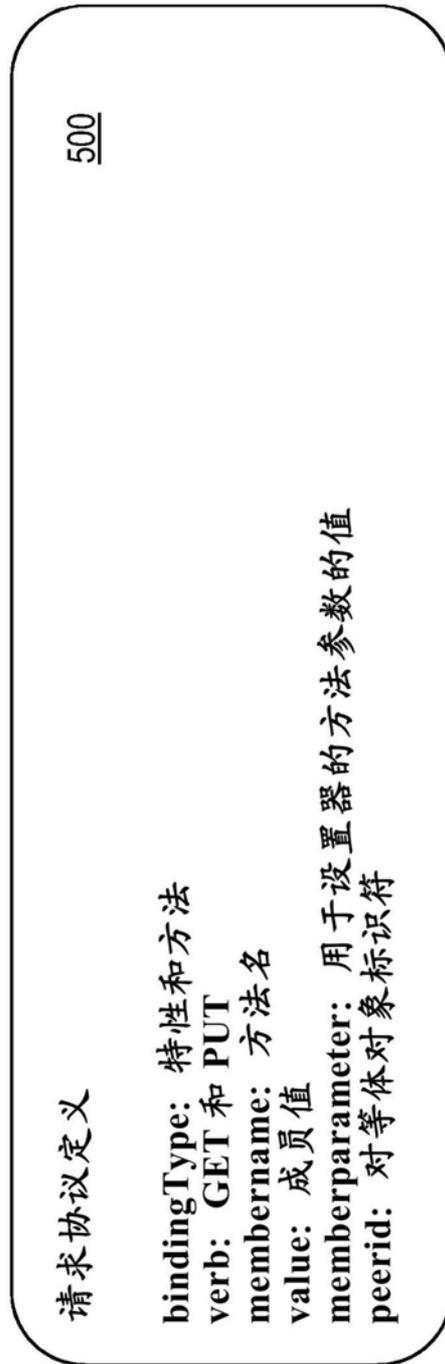


图5



图6

请求格式

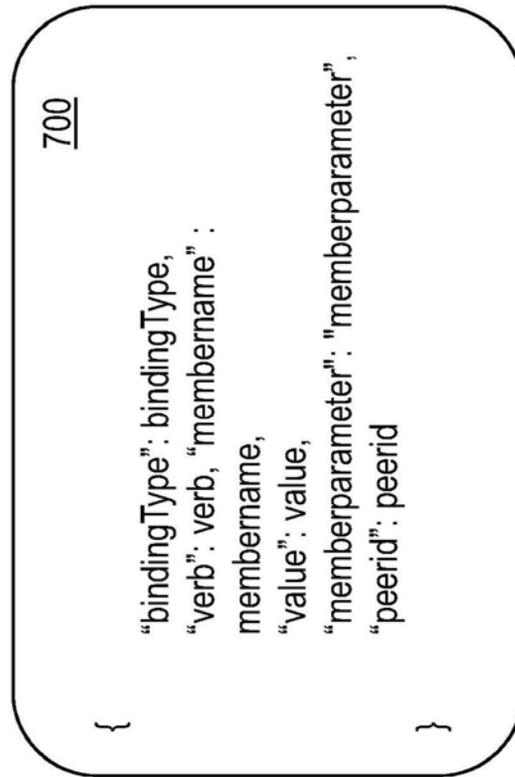


图7

响应格式



图8

支持的可绑定数据类型

- Nullable
- Bool
- Number
- String
- String Array
- 如 JSON String 的 Complex Type
- Item List Array

900

图9

1000

- □ X

← 设备

设备名

XXXXXXXX-XXXXXX-XXXX

继续

图10

1100

- □ X

← 同步服务器

同步服务器

localhost

端口号

XXXX

跳过 继续

图11

1200

- □ X
App 名
基本
库存
用户
管理员
角色
每个人
密码
XXXX
继续

图12