



(12) 发明专利申请

(10) 申请公布号 CN 117806758 A

(43) 申请公布日 2024. 04. 02

(21) 申请号 202311845530.1

(22) 申请日 2023.12.28

(71) 申请人 北京百度网讯科技有限公司

地址 100085 北京市海淀区上地十街10号
百度大厦二层

(72) 发明人 王说 董睿 樊中恺 穆瑞强
周廷帅

(74) 专利代理机构 北京清亦华知识产权代理事
务所(普通合伙) 11201

专利代理师 郭美娇

(51) Int. Cl.

G06F 9/451 (2018.01)

G06F 9/445 (2018.01)

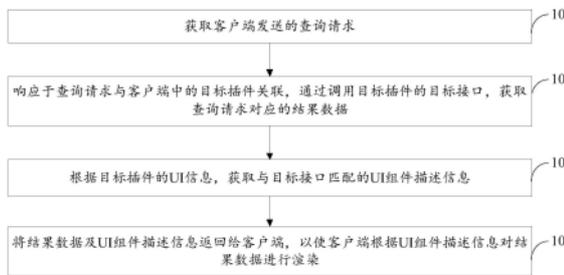
权利要求书3页 说明书11页 附图3页

(54) 发明名称

插件UI渲染方法、装置、电子设备及存储介质

(57) 摘要

本申请公开了插件UI渲染方法、装置、电子设备及存储介质,涉及计算机技术领域,尤其涉及深度学习等人工智能领域。具体实施方案为:获取客户端发送的查询请求;响应于查询请求与客户端中的目标插件关联,通过调用目标插件的目标接口,获取查询请求对应的结果数据;根据目标插件的UI信息,获取与目标接口匹配的UI组件描述信息;将结果数据及UI组件描述信息返回给客户端,以使客户端根据UI组件描述信息对结果数据进行渲染。



1. 一种插件用户界面UI渲染方法,包括:
 - 获取客户端发送的查询请求;
 - 响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据;
 - 根据所述目标插件的UI信息,获取与所述目标接口匹配的UI组件描述信息;
 - 将所述结果数据及所述UI组件描述信息返回给所述客户端,以使所述客户端根据所述UI组件描述信息对所述结果数据进行渲染。
2. 如权利要求1所述的方法,其中,所述根据所述目标插件的UI信息,获取与所述目标接口匹配的UI组件描述信息,包括:
 - 从所述UI信息中获取接口与UI描述信息之间的对应关系;
 - 从所述对应关系中确定与所述目标接口对应的UI描述信息;
 - 从所述UI描述信息中获取所述UI组件描述信息。
3. 如权利要求1所述的方法,其中,所述将所述结果数据及所述UI组件描述信息返回给所述客户端,包括:
 - 响应于获取到与所述目标接口匹配的数据映射关系,根据所述数据映射关系,对所述结果数据进行处理,以获取目标数据;其中,所述数据映射关系表示UI中的数据与接口数据之间的映射关系;
 - 将所述目标数据及所述UI组件描述信息发送给所述客户端。
4. 如权利要求1所述的方法,其中,所述将所述结果数据及所述UI组件描述信息返回给所述客户端,包括:
 - 响应于获取到所述UI组件描述信息对应的UI数据格式描述信息,将所述结果数据、所述UI组件描述信息及所述UI数据格式描述信息发送给所述客户端。
5. 如权利要求1所述的方法,其中,所述将所述结果数据及所述UI组件描述信息返回给所述客户端,包括:
 - 响应于获取到所述UI组件描述信息对应的UI数据格式描述信息,根据所述UI数据格式描述信息,对所述结果数据进行格式化,得到格式化后的数据;
 - 将所述格式化后的数据及所述UI组件描述信息发送给所述客户端。
6. 如权利要求1所述的方法,其中,所述响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据,包括:
 - 响应于所述查询请求中携带有所述目标插件的标识,确定所述查询请求与所述目标插件关联;
 - 通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据。
7. 如权利要求1所述的方法,其中,所述响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据,包括:
 - 响应于所述查询请求中未携带插件标识,对所述查询请求进行意图识别,以确定所述查询请求的查询意图;
 - 响应于所述查询意图与所述目标插件匹配,确定所述查询请求与所述目标插件关联;
 - 通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据。
8. 如权利要求6或7所述的方法,其中,所述通过调用所述目标插件的目标接口,获取所

述查询请求对应的结果数据,包括:

根据所述目标插件的多个接口的功能描述信息,确定所述多个接口中与所述查询意图匹配的接口;

将与所述查询意图匹配的接口确定为所述目标接口;

通过调用所述目标接口,获取所述结果数据。

9.如权利要求1所述的方法,还包括:

获取所述目标插件的UI描述文件;

从所述UI描述文件中获取所述UI信息。

10.一种插件UI渲染装置,包括:

第一获取模块,用于获取客户端发送的查询请求;

第二获取模块,用于响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据;

第三获取模块,用于根据所述目标插件的UI信息,获取与所述目标接口匹配的UI组件描述信息;

发送模块,用于将所述结果数据及所述UI组件描述信息返回给所述客户端,以使所述客户端根据所述UI组件描述信息对所述结果数据进行渲染。

11.如权利要求10所述的装置,其中,所述第三获取模块,用于:

从所述UI信息中获取接口与UI描述信息之间的对应关系;

从所述对应关系中确定与所述目标接口对应的UI描述信息;

从所述UI描述信息中获取所述UI组件描述信息。

12.如权利要求10所述的装置,其中,所述发送模块,用于:

响应于获取到与所述目标接口匹配的数据映射关系,根据所述数据映射关系,对所述结果数据进行处理,以获取目标数据;其中,所述数据映射关系表示UI中的数据与接口数据之间的映射关系;

将所述目标数据及所述UI组件描述信息发送给所述客户端。

13.如权利要求10所述的装置,其中,所述发送模块,用于:

响应于获取到所述UI组件描述信息对应的UI数据格式描述信息,将所述结果数据、所述UI组件描述信息及所述UI数据格式描述信息发送给所述客户端。

14.如权利要求10所述的装置,其中,所述发送模块,用于:

响应于获取到所述UI组件描述信息对应的UI数据格式描述信息,根据所述UI数据格式描述信息,对所述结果数据进行格式化,得到格式化后的数据;

将所述格式化后的数据及所述UI组件描述信息发送给所述客户端。

15.如权利要求10所述的装置,其中,所述第二获取模块,用于:

响应于所述查询请求中携带有所述目标插件的标识,确定所述查询请求与所述目标插件关联;

通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据。

16.如权利要求10所述的装置,其中,所述第二获取模块,用于:

响应于所述查询请求中未携带插件标识,对所述查询请求进行意图识别,以确定所述查询请求的查询意图;

响应于所述查询意图与所述目标插件匹配,确定所述查询请求与所述目标插件关联;
通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据。

17. 如权利要求15或16所述的装置,其中,所述第二获取模块,用于:

根据所述目标插件的多个接口的功能描述信息,确定所述多个接口中与所述查询意图匹配的接口;

将与所述查询意图匹配的接口确定为所述目标接口;

通过调用所述目标接口,获取所述结果数据。

18. 如权利要求10所述的装置,还包括:

第四获取模块,用于获取所述目标插件的UI描述文件;

第五获取模块,用于从所述UI描述文件中获取所述UI信息。

19. 一种电子设备,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1-9中任一项所述的方法。

20. 一种存储有计算机指令的非瞬时计算机可读存储介质,其中,所述计算机指令用于使所述计算机执行根据权利要求1-9中任一项所述的方法。

21. 一种计算机程序产品,包括计算机程序,所述计算机程序在被处理器执行时实现权利要求1-9中任一项所述方法的步骤。

插件UI渲染方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及深度学习等人工智能领域,具体涉及插件UI(User Interface,用户界面)渲染方法、装置、电子设备及存储介质。

背景技术

[0002] 生成式大模型可以是指能够以端到端的形式对自然语言进行生成、理解和推理的大规模神经网络模型。生成式大模型可以支持以插件的形式接入一些外部服务,以丰富模型的能力,满足用户的多样化需求。

发明内容

[0003] 本申请提供了一种插件UI渲染方法、装置、电子设备及存储介质。

[0004] 具体方案如下:

[0005] 根据本申请的一方面,提供了一种插件UI渲染方法,包括:

[0006] 获取客户端发送的查询请求;

[0007] 响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据;

[0008] 根据所述目标插件的UI信息,获取与所述目标接口匹配的UI组件描述信息;

[0009] 将所述结果数据及所述UI组件描述信息返回给所述客户端,以使所述客户端根据所述UI组件描述信息对所述结果数据进行渲染。

[0010] 根据本申请的另一方面,提供了一种插件UI渲染装置,包括:

[0011] 第一获取模块,用于获取客户端发送的查询请求;

[0012] 第二获取模块,用于响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据;

[0013] 第三获取模块,用于根据所述目标插件的UI信息,获取与所述目标接口匹配的UI组件描述信息;

[0014] 发送模块,用于将所述结果数据及所述UI组件描述信息返回给所述客户端,以使所述客户端根据所述UI组件描述信息对所述结果数据进行渲染。

[0015] 根据本申请的另一方面,提供了一种电子设备,包括:

[0016] 至少一个处理器;以及

[0017] 与所述至少一个处理器通信连接的存储器;其中,

[0018] 所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行上述实施例所述的方法。

[0019] 根据本申请的另一方面,提供了一种存储有计算机指令的非瞬时计算机可读存储介质,其中,所述计算机指令用于使所述计算机执行根据上述实施例所述的方法。

[0020] 根据本申请的另一方面,提供了一种计算机程序产品,包括计算机程序,所述计算机程序在被处理器执行时实现上述实施例所述方法的步骤。

[0021] 应当理解,本部分所描述的内容并非旨在标识本申请的实施例的关键或重要特征,也不用于限制本申请的范围。本申请的其它特征将通过以下的说明书而变得容易理解。

附图说明

[0022] 附图用于更好地理解本方案,不构成对本申请的限定。其中:

[0023] 图1为本申请一实施例提供的插件UI渲染方法的流程示意图;

[0024] 图2为本申请另一实施例提供的插件UI渲染方法的流程示意图;

[0025] 图3为本申请另一实施例提供的插件UI渲染方法的流程示意图;

[0026] 图4为本申请实施例提供的一种插件UI渲染过程示意图;

[0027] 图5为本申请一实施例提供的插件UI渲染装置的结构示意图;

[0028] 图6是用来实现本申请实施例的插件UI渲染方法的电子设备的框图。

具体实施方式

[0029] 以下结合附图对本申请的示范性实施例做出说明,其中包括本申请实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本申请的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0030] 下面参考附图描述本申请实施例的插件UI渲染方法、装置、电子设备和存储介质。

[0031] 生成式大模型可以支持以插件的形式接入一些外部服务,以丰富模型的能力,满足用户的多样化需求。

[0032] 在一些实施例中,插件的返回结果可以以纯文本类型,或者markdown类型通知前端,再由前端通过文本框或markdown格式渲染器渲染。

[0033] 但是,纯文本类型和markdown类型能够表达的内容是只读的静态内容,不能展示更丰富的可交互的信息。

[0034] 基于此,本申请实施例提供一种插件UI渲染方法,可以满足插件展示可交互UI的需求,丰富了插件的返回结果的展示方式。

[0035] 图1为本申请一实施例提供的插件UI渲染方法的流程示意图。

[0036] 本申请实施例的插件UI渲染方法,可以由本申请实施例的插件UI渲染方法装置执行,该装置可以配置于电子设备中,以实现插件UI渲染功能。

[0037] 其中,电子设备可以为任一具有计算能力的设备,例如可以为个人电脑、移动终端、服务器等,移动终端例如可以为车载设备、手机、平板电脑、个人数字助理、穿戴式设备等具有各种操作系统、触摸屏和/或显示屏的硬件设备。

[0038] 如图1所示,该插件UI渲染方法包括:

[0039] 步骤101,获取客户端发送的查询请求。

[0040] 本申请中,客户端可以是能够运行至少一个插件的应用程序或网页。比如,客户端可以是运行生成式大模型插件的应用程序或网页,大模型插件可以是一个或多个。

[0041] 用户可以在客户端中输入查询语句,客户端可以生成相应的查询请求并发送给服务端,从而服务端可以获取客户端发送的查询请求。

[0042] 比如,用户在某生成式大模型的应用程序中输入问题“请为我推荐附近的美食”,

该应用程序可以将相应的查询请求发送给该生成式大模型的服务端。

[0043] 步骤102,响应于查询请求与客户端中的目标插件关联,通过调用目标插件的目标接口,获取查询请求对应的结果数据。

[0044] 其中,查询请求与客户端中的目标插件关联,可以理解为可以调用目标插件对查询请求进行处理。

[0045] 可选地,如果查询请求中携带有目标插件的标识,可以确定查询请求与目标插件关联,并通过调用目标插件的目标接口,获取查询请求对应的结果数据。

[0046] 示例性的,用户可以在客户端中打开某插件,在该插件中输入查询语句,由此生成的查询请求中可以携带该插件的标识。

[0047] 由此,在查询请求携带有目标插件的标识的情况下,可以确定需要通过调用目标插件对查询请求进行处理,提高了查询请求的处理效率。

[0048] 可选地,如果查询请求中未携带插件标识,可以对查询请求进行意图识别,以确定查询请求的查询意图,如果查询意图与目标插件匹配,可以确定查询请求与目标插件关联,并通过调用目标插件的目标接口,获取查询请求对应的结果数据。

[0049] 示例性的,用户可以在客户端中直接输入查询语句,由此生成的查询请求中未携带插件标识,那么服务端可以对查询请求进行识别,以确定查询请求的查询意图,基于查询意图,确定是否需要调用某个插件对查询请求进行处理。

[0050] 由此,在查询请求中未携带插件标识的情况下,可以基于查询请求的查询意图,确定是否需要调用插件对查询请求进行处理,并在查询意图与目标插件匹配时,调用目标插件对查询请求进行处理,提高了插件调用的准确性。

[0051] 本申请中,插件可以有至少一个接口,如果查询请求与客户端中的目标插件关联,可以从至少一个接口中确定出目标接口,通过调用目标插件的目标接口,获取查询请求对应的结果数据。其中,结果数据可以是指目标接口返回的数据。

[0052] 可选地,查询请求中可以包括查询语句,可以将查询语句与多个接口的功能描述信息进行匹配,可以将匹配度最高的功能描述信息所属的接口,确定为目标接口,从而通过调用目标接口获取查询请求的结果数据。其中,接口的功能描述信息可以用于描述接口的功能。

[0053] 可选地,可以将查询请求的查询意图与目标插件的多个接口的功能描述信息进行匹配,可以将匹配度最高的功能描述信息所属的接口确定为与查询意图匹配的接口,并将与查询意图匹配的接口确定为目标接口,从而通过调用目标接口获取结果数据。

[0054] 由此,通过根据查询意图与目标插件的多个接口的功能描述信息的匹配情况,从多个接口中确定出目标接口,提高了接口调用的准确性。

[0055] 示例性的,可以将查询请求通过目标接口发送给目标插件的服务端,由目标插件的服务端对查询请求进行处理,并将结果数据通过目标接口发送给客户端对应的服务端,由此可以获取查询请求对应的结果数据。

[0056] 示例性的,插件的接口可以为API(Application Programming Interface,应用程序编程接口)。

[0057] 步骤103,根据目标插件的UI信息,获取与目标接口匹配的UI组件描述信息。

[0058] 其中,目标插件的UI信息可以是指用于描述目标插件的返回结果的UI的信息,不

同的插件可以具有相应的UI信息,从而可以获取目标插件的UI信息。

[0059] 可选地,客户端中每个插件可以具有对应的UI描述文件,UI描述文件中可以包括UI信息,从而可以从目标插件的UI描述文件中获取目标插件的UI信息。

[0060] 示例性的,UI描述文件的格式可以是.json格式,.json格式更灵活,更适合用于前端描述。

[0061] 以生成式大模型为例,对生成式大模型插件协议的定义可以包含:ai-plugin.json文件:用于描述插件的基本信息;openapi.yaml文件:描述插件服务的接口信息;ui.json文件:描述插件的返回结果的UI信息。

[0062] 由此,不同的插件可以具有相应的UI描述文件,从而基于UI描述文件,针对不同的插件分发场景可以实现插件UI的能力,适用性强。

[0063] 本申请中,UI信息中可以包括但不限于接口、UI组件描述信息、UI中的数据与接口数据之间的映射关系等。

[0064] 其中,UI组件描述信息可以是指用于描述UI组件的信息,UI组件描述信息可以包括但不限于UI组件的类型、布局、样式、交互等。这里交互可以是指UI组件的交互行为,比如一个按钮点击后的行为、一个图片可以点开查看大图等。

[0065] 示例性的,UI组件可以是单组件或也可以是多个单组件拼装得到一个复杂组件,由多个单组件拼装得到的复杂组件可以称为容器组件。其中,单组件可以是指插件UI中的原子化组件,原子化组件是一个不可分割的业务单元,可以通过组件的类型区分是单组件还是容器组件。对于容器组件,容器组件内部可以是一个递归的组件树,根组件是唯一的。

[0066] 本申请中,插件的接口与UI之间具有对应关系,不同的接口可以对应不同的UI,那么插件的每个接口可以有对应的UI信息,可以从目标插件的UI信息中,获取与目标接口匹配的UI组件描述信息。

[0067] 步骤104,将结果数据及UI组件描述信息返回给客户端,以使客户端根据UI组件描述信息对结果数据进行渲染。

[0068] 本申请中,可以将结果数据和UI组件描述信息返回给客户端,那么客户端可以根据UI组件描述信息对结果数据进行渲染,得到相应的UI,从而客户端可以展示可交互UI。

[0069] 本申请实施例中,通过根据目标插件的UI信息,确定与目标接口匹配的UI组件描述信息,将查询请求对应的结果数据及UI组件描述信息返回给客户端,由客户端根据UI组件描述信息对结果数据进行渲染,从而可以展示丰富的可交互的信息,满足展示可交互UI的需求。

[0070] 图2为本申请另一实施例提供的插件UI渲染方法的流程示意图。

[0071] 如图2所示,该插件UI渲染方法包括:

[0072] 步骤201,获取客户端发送的查询请求。

[0073] 步骤202,响应于查询请求与客户端中的目标插件关联,通过调用目标插件的目标接口,获取查询请求对应的结果数据。

[0074] 本申请中,步骤201-步骤202可以采用本申请各实施例中任一种实现方式,故在此不再赘述。

[0075] 步骤203,从目标插件的UI信息中获取接口与UI描述信息之间的对应关系。

[0076] 其中,UI描述信息可以包括UI组件描述信息、UI中的数据与接口数据之间的映射

关系等。

[0077] 本申请中,一个插件可以有多个接口和多个UI,而每个接口可以对应一个UI,插件的UI信息中可以包括接口与UI描述信息之间的对应关系。

[0078] 示例性的,可以通过数组形式列举接口与UI描述信息之间的对应关系;其中,接口可以通过接口唯一标识字段(比如operationID字段)描述,operationID字段可以是字符串类型,默认method为get或者object,比如该字段的值为“/get_word”;UI描述信息中的UI组件描述信息可以通过声明组件字段(比如component字段)描述,UI描述信息中UI中的数据与接口数据之间的映射关系可以通过数据扩展字段(比如dataExtend字段)描述。

[0079] 这里数组中每个数组项可以包含operationID字段和component字段,还可以包括dataExtend字段;其中,component字段中包括组件类型字段type字段,operationID字段与openapi.yaml文件中的operationId字段对应。

[0080] 步骤204,从对应关系中确定与目标接口对应的UI描述信息。

[0081] 本申请中,可以从对应关系中查找目标接口,并确定与目标接口对应的UI描述信息。

[0082] 步骤205,从UI描述信息中获取UI组件描述信息。

[0083] 本申请中,可以基于UI描述信息中描述UI组件信息的字段,获取UI组件描述信息。

[0084] 示例性的,描述UI组件信息的字段(比如component字段)可以包括描述UI组件的类型的字段(比如type字段)、描述UI组件的布局的字段、描述UI组件的样式的字段等。

[0085] 步骤206,将结果数据及UI组件描述信息返回给客户端,以使客户端根据UI组件描述信息对结果数据进行渲染。

[0086] 本申请中,步骤206可以采用本申请各实施例中的一种实现方式,故在此不再赘述。

[0087] 举例来讲,某插件开发者在生成式大模型的客户端注册了一个插件“景区推荐助手”,其中,UI信息中operationID为‘getLocation’的接口对应一个component字段,component字段中type字段的值为‘AMap’,用户在打开客户端,并选中插件“景区推荐助手”,在插件中输入问题“请为我推荐b市的热门景点”,生成式大模型服务端调用该插件服务,获取景点数据,并根据该插件的UI描述文件得到UI信息,其中获取景点数据所用的接口对应的UI组件描述信息中component字段中type字段的值为‘AMap’,客户端根据此UI组件描述信息渲染出一个标注了b市多个景点地址的A地图组件。其中,A地图是地图应用程序名称。

[0088] 本申请实施例中,通过从UI信息中获取接口与UI描述信息之间的对应关系,从对应关系中确定与目标接口对应的UI描述信息,再从UI描述信息中获取UI组件描述信息,由此可以从UI信息中准确地找到与目标接口匹配的UI组件描述信息,以使客户端精准地展示相应的插件UI。

[0089] 图3为本申请另一实施例提供的插件UI渲染方法的流程示意图。

[0090] 如图3所示,该插件UI渲染方法包括:

[0091] 步骤301,获取客户端发送的查询请求。

[0092] 步骤302,响应于查询请求与客户端中的目标插件关联,通过调用目标插件的目标接口,获取查询请求对应的结果数据。

[0093] 步骤303,根据目标插件的UI信息,获取与目标接口匹配的UI组件描述信息。

[0094] 本申请中,步骤301-步骤303可以采用本申请各实施例中任一种实现方式,故在此不再赘述。

[0095] 步骤304,响应于获取到与目标接口匹配的数据映射关系,根据数据映射关系,对结果数据进行处理,以获取目标数据。

[0096] 在实际应用中,插件的接口返回的结果数据不能满足UI展示需求,比如,接口返回的字段是与支付相关的字段,比如返回的是一个值18,为了便于用户理解,可以在该数值后面加上货币单位。或者,插件的接口返回的是一个结构数据,比如,接口返回的是一个数据结构为data.a.b.c,访问路径比较长,UI组件又有很多地方用到这个值。

[0097] 基于此,本申请中,UI信息中一个或多个接口对应的UI描述信息中可以包括数据映射关系,该数据映射关系可以表示UI中的数据与接口数据之间的映射关系。示例性的,可以通过dataExtends字段描述该数据映射关系。

[0098] 本申请中,如果从UI信息中获取到与目标接口匹配的数据映射关系,那么可以根据该数据映射关系,对结果数据进行处理,获取目标数据。

[0099] 示例性的,UI信息中包括接口与UI描述信息之间的对应关系,如果从与目标接口匹配的UI描述信息中获取到数据映射关系,则可以根据该数据映射关系,对目标接口返回的结果数据进行处理,获取目标数据。

[0100] 比如,数据映射关系包括目标接口返回的值后面加上货币单位“元”,那么根据该数据映射关系,可以对接口数据进行复用,在目标接口返回的值后面加上货币单位“元”,得到目标数据。

[0101] 又如,数据映射关系包括接口返回的值为data.a.b.c,可以给该值赋一个名称,那么在UI组件里用到这个值的地方就无需再写data.a.b.c了,从而简化了访问路径。

[0102] 步骤305,将目标数据及UI组件描述信息发送给客户端。

[0103] 本申请中,可以将目标数据和UI组件描述信息返回给客户端,那么客户端可以根据UI组件描述信息对目标数据进行渲染,得到相应的UI,从而客户端可以展示可交互UI。

[0104] 本申请实施例中,如果获取到与目标接口匹配的数据映射关系,可以根据数据映射关系,对结果数据进行处理,得到目标数据,将目标数据和UI组件描述信息返回给客户端,由客户端根据UI组件描述信息对结果数据进行渲染,从而可以满足插件UI展示多样化需求、简化访问路径等。

[0105] 在实际应用中,不同的UI组件对数据格式的需求可能不同。其中,UI组件的数据格式可以是指UI组件如何展示和呈现数据的方式。UI组件的数据格式可以定义数据的展示方式、布局、格式化规则等。比如,格式格式化规则可以是指数据的格式化方式,例如日期时间的格式、货币的格式等。

[0106] 基于此,在本申请的一个实施例中,响应于获取到UI组件描述信息对应的UI数据格式描述信息,可以根据UI数据格式描述信息,对结果数据进行格式化,再将格式化后的数据、UI组件描述信息及UI数据格式描述信息发送给客户端。

[0107] 其中,UI数据格式描述信息可以用于对UI组件需要的数据格式进行描述,UI数据格式描述信息可以用于告知服务端或客户端UI组件需要的返回格式是什么,由服务端或客户端根据UI数据格式描述信息进行格式化。

[0108] 示例性的,可以通过acceptDataSchema字段描述UI组件需要的数据格式。比如,某UI组件的acceptDataSchema字段如下所示:

```

"acceptDataSchema": [
  {
    "key": "userInfo",
    "description": "用户信息",
    "type": "object",
    "fields": [
      {
        "key": "name",
        "description": "姓名"
      },
      {
        "key": "age",
        "description": "年龄"
      }
    ]
  }
]

```

[0110] 本申请中,如果从UI信息中获取到UI组件描述信息对应的UI数据格式描述信息,那么可以根据UI数据格式描述信息,对结果数据进行格式化。

[0111] 示例性的,UI信息中包括接口与UI描述信息之间的对应关系,UI描述信息包括UI组件描述信息,可以根据需要在UI描述信息中定义UI组件需要的数据格式,如果从与目标接口匹配的UI描述信息中获取到UI数据格式描述信息,则可以根据UI数据格式描述信息,对结果数据进行格式化。

[0112] 比如,结果数据中日期的格式为“年/月/日”,可以根据数据格式描述信息中对日期格式的描述为“年-月-日”,则可以将结果数据中的日期格式化为“年-月-日”形式。

[0113] 可选地,如果未获取到UI数据格式描述信息,可以使用目标接口的接口信息中定义的目标接口返回的数据的数据格式。

[0114] 本申请实施例中,如果获取到UI组件描述信息对应的UI数据格式描述信息,可以由服务端基于UI数据格式描述信息,对结果数据进行格式化,再将格式化的数据、UI组件描述信息及UI数据格式描述信息发送给客户端,从而客户端可以基于UI组件描述信息、UI数据格式描述信息等对格式化后的数据进行渲染,可以满足插件UI对不同数据格式的需求。

[0115] 在本申请的一个实施例中,如果获取到UI组件描述信息对应的UI数据格式描述信

息,可以将结果数据、UI组件描述信息及UI数据格式描述信息发送给客户端,由客户端根据UI数据格式描述信息对结果数据进行格式化,再根据UI组件描述信息对格式化后的数据进行渲染,从而可以满足插件UI对不同数据格式的需求。

[0116] 为了便于理解本申请的方案,下面结合图4以生成式大模型插件为例进行说明,图4为本申请实施例提供的一种插件UI渲染过程示意图。

[0117] 如图4所示,插件分发宿主在插件中接收用户提问并触发模型请求,将模型请求发送给生成式大模型服务端,生成式大模型服务端调用相应插件,将模型请求发送给插件服务端,插件服务端接收模型请求,并返回结果数据,生成式大模型接收插件服务端返回的结果数据,并根据operationID(也即目标接口的标识)匹配UI组件描述信息,返回UI组件描述信息,从而插件分发宿主可以根据UI组件描述信息,通过渲染组件,对结果数据进行渲染,得到相应的插件UI。

[0118] 其中,插件分发宿主可以是运行生成式大模型插件的网页或应用程序,插件服务端可以是指提供插件服务的服务端。

[0119] 可见,该方案能够让生成式大模型交互与GUI(Graphical User Interface,图形用户界面)交互结合,在一些GUI操作更高效的场景下,通过GUI的方式丰富用户体验,可以弥补纯文字或markdown格式的表达的单一性。并且,该方案具有通用性,声明式的.json格式,将ui.json作为插件的标准协议的一部分,任意插件分发场景均可以以此ui.json格式作为标准,支持此标准从而实现这种插件UI能力。

[0120] 为了实现上述实施例,本申请实施例还提出一种插件UI渲染装置。图5为本申请一实施例提供的插件UI渲染装置的结构示意图。

[0121] 如图5所示,该插件UI渲染装置500包括:

[0122] 第一获取模块510,用于获取客户端发送的查询请求;

[0123] 第二获取模块520,用于响应于所述查询请求与所述客户端中的目标插件关联,通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据;

[0124] 第三获取模块530,用于根据所述目标插件的UI信息,获取与所述目标接口匹配的UI组件描述信息;

[0125] 发送模块540,用于将所述结果数据及所述UI组件描述信息返回给所述客户端,以使所述客户端根据所述UI组件描述信息对所述结果数据进行渲染。

[0126] 可选地,所述第三获取模块530,用于:

[0127] 从所述UI信息中获取接口与UI描述信息之间的对应关系;

[0128] 从所述对应关系中确定与所述目标接口对应的UI描述信息;

[0129] 从所述UI描述信息中获取所述UI组件描述信息。

[0130] 可选地,所述发送模块540,用于:

[0131] 响应于获取到与所述目标接口匹配的数据映射关系,根据所述数据映射关系,对所述结果数据进行处理,以获取目标数据;其中,所述数据映射关系表示UI中的数据与接口数据之间的映射关系;

[0132] 将所述目标数据及所述UI组件描述信息发送给所述客户端。

[0133] 可选地,所述发送模块540,用于:

[0134] 响应于获取到所述UI组件描述信息对应的UI数据格式描述信息,将所述结果数

据、所述UI组件描述信息及所述UI数据格式描述信息发送给所述客户端。

[0135] 可选地,所述发送模块540,用于:

[0136] 响应于获取到所述UI组件描述信息对应的UI数据格式描述信息,根据所述UI数据格式描述信息,对所述结果数据进行格式化,得到格式化后的数据;

[0137] 将所述格式化后的数据及所述UI组件描述信息发送给所述客户端。

[0138] 可选地,所述第二获取模块520,用于:

[0139] 响应于所述查询请求中携带有所述目标插件的标识,确定所述查询请求与所述目标插件关联;

[0140] 通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据。

[0141] 可选地,所述第二获取模块520,用于:

[0142] 响应于所述查询请求中未携带插件标识,对所述查询请求进行意图识别,以确定所述查询请求的查询意图;

[0143] 响应于所述查询意图与所述目标插件匹配,确定所述查询请求与所述目标插件关联;

[0144] 通过调用所述目标插件的目标接口,获取所述查询请求对应的结果数据。

[0145] 可选地,所述第二获取模块520,用于:

[0146] 根据所述目标插件的多个接口的功能描述信息,确定所述多个接口中与所述查询意图匹配的接口;

[0147] 将与所述查询意图匹配的接口确定为所述目标接口;

[0148] 通过调用所述目标接口,获取所述结果数据。

[0149] 可选地,该装置还可以包括:

[0150] 第四获取模块,用于获取所述目标插件的UI描述文件;

[0151] 第五获取模块,用于从所述UI描述文件中获取所述UI信息。

[0152] 需要说明的是,前述插件UI渲染方法实施例的解释说明,也适用于该实施例的插件UI渲染装置,故在此不再赘述。

[0153] 本申请实施例中,通过根据目标插件的UI信息,确定与目标接口匹配的UI组件描述信息,将查询请求对应的结果数据及UI组件描述信息返回给客户端,由客户端根据UI组件描述信息对结果数据进行渲染,从而可以展示丰富的可交互的信息,满足展示可交互UI的需求。

[0154] 根据本申请的实施例,本申请还提供了一种电子设备、一种可读存储介质和一种计算机程序产品。

[0155] 图6示出了可以用来实施本申请的实施例的示例电子设备600的示意性框图。电子设备旨在表示各种形式的数字计算机,诸如,膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置,诸如,个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例,并且不意在限制本文中描述的和/或者要求的本申请的实现。

[0156] 如图6所示,设备600包括计算单元601,其可以根据存储在ROM(Read-Only Memory,只读存储器)602中的计算机程序或者从存储单元608加载到RAM(Random Access

Memory,随机访问/存取存储器)603中的计算机程序,来执行各种适当的动作和处理。在RAM 603中,还可存储设备600操作所需的各种程序和数据。计算单元601、ROM 602以及RAM 603通过总线604彼此相连。I/O(Input/Output,输入/输出)接口605也连接至总线604。

[0157] 设备600中的多个部件连接至I/O接口605,包括:输入单元606,例如键盘、鼠标等;输出单元607,例如各种类型的显示器、扬声器等;存储单元608,例如磁盘、光盘等;以及通信单元609,例如网卡、调制解调器、无线通信收发机等。通信单元609允许设备600通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0158] 计算单元601可以是各种具有处理和计算能力的通用和/或专用处理组件。计算单元601的一些示例包括但不限于CPU(Central Processing Unit,中央处理单元)、GPU(Graphic Processing Units,图形处理单元)、各种专用的AI(Artificial Intelligence,人工智能)计算芯片、各种运行机器学习模型算法的计算单元、DSP(Digital Signal Processor,数字信号处理器)、以及任何适当的处理器、控制器、微控制器等。计算单元601执行上文所描述的各个方法和处理,例如插件UI渲染方法。例如,在一些实施例中,插件UI渲染方法可被实现为计算机软件程序,其被有形地包含于机器可读介质,例如存储单元608。在一些实施例中,计算机程序的部分或者全部可以经由ROM 602和/或通信单元609而被载入和/或安装到设备600上。当计算机程序加载到RAM 603并由计算单元601执行时,可以执行上文描述的插件UI渲染方法的一个或多个步骤。备选地,在其他实施例中,计算单元601可以通过其他任何适当的方式(例如,借助于固件)而被配置为执行插件UI渲染方法。

[0159] 本文中以上描述的系统和技术各种实施方式可以在数字电子电路系统、集成电路系统、FPGA(Field Programmable Gate Array,现场可编程门阵列)、ASIC(Application-Specific Integrated Circuit,专用集成电路)、ASSP(Application Specific Standard Product,专用标准产品)、SOC(System On Chip,芯片上系统的系统)、CPLD(Complex Programmable Logic Device,复杂可编程逻辑设备)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括:实施在一个或者多个计算机程序中,该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释,该可编程处理器可以是专用或者通用可编程处理器,可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令,并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0160] 用于实施本申请的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器,使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行,作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

[0161] 在本申请的上下文中,机器可读介质可以是有形的介质,其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备,或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、RAM、ROM、EPROM(Electrically Programmable Read-Only-Memory,可擦除可

编程只读存储器)或快闪存储器、光纤、CD-ROM(Compact Disc Read-Only Memory,便捷式紧凑盘只读存储器)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0162] 为了提供与用户的交互,可以在计算机上实施此处描述的系统和技术,该计算机具有:用于向用户显示信息的显示装置(例如,CRT(Cathode-Ray Tube,阴极射线管)或者LCD(Liquid Crystal Display,液晶显示器)监视器);以及键盘和指向装置(例如,鼠标或者轨迹球),用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何形式的传感反馈(例如,视觉反馈、听觉反馈、或者触觉反馈);并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0163] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如,作为数据服务器)、或者包括中间件部件的计算系统(例如,应用服务器)、或者包括前端部件的计算系统(例如,具有图形用户界面或者网络浏览器的用户计算机,用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信(例如,通信网络)来将系统的部件相互连接。通信网络的示例包括:LAN(Local Area Network,局域网)、WAN(Wide Area Network,广域网)、互联网和区块链网络。

[0164] 计算机系统可以包括客户端和服务端。客户端和服务端一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务器关系的计算机程序来产生客户端和服务端的关系。服务器可以是云服务器,又称为云计算服务器或云主机,是云计算服务体系中的一项主机产品,以解决了传统物理主机与VPS服务(Virtual Private Server,虚拟专用服务器)中,存在的管理难度大,业务扩展性弱的缺陷。服务器也可以为分布式系统的服务器,或者是结合了区块链的服务器。

[0165] 根据本申请的实施例,本申请还提供了一种计算机程序产品,当计算机程序产品中的指令处理器执行时,执行本申请上述实施例提出的插件UI渲染方法。

[0166] 应该理解,可以使用上面所示的各种形式的流程,重新排序、增加或删除步骤。例如,本申请中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行,只要能够实现本申请公开的技术方案所期望的结果,本文在此不进行限制。

[0167] 上述具体实施方式,并不构成对本申请保护范围的限制。本领域技术人员应该明白的是,根据设计要求和因素,可以进行各种修改、组合、子组合和替代。任何在本申请的精神和原则之内所作的修改、等同替换和改进等,均应包含在本申请保护范围之内。

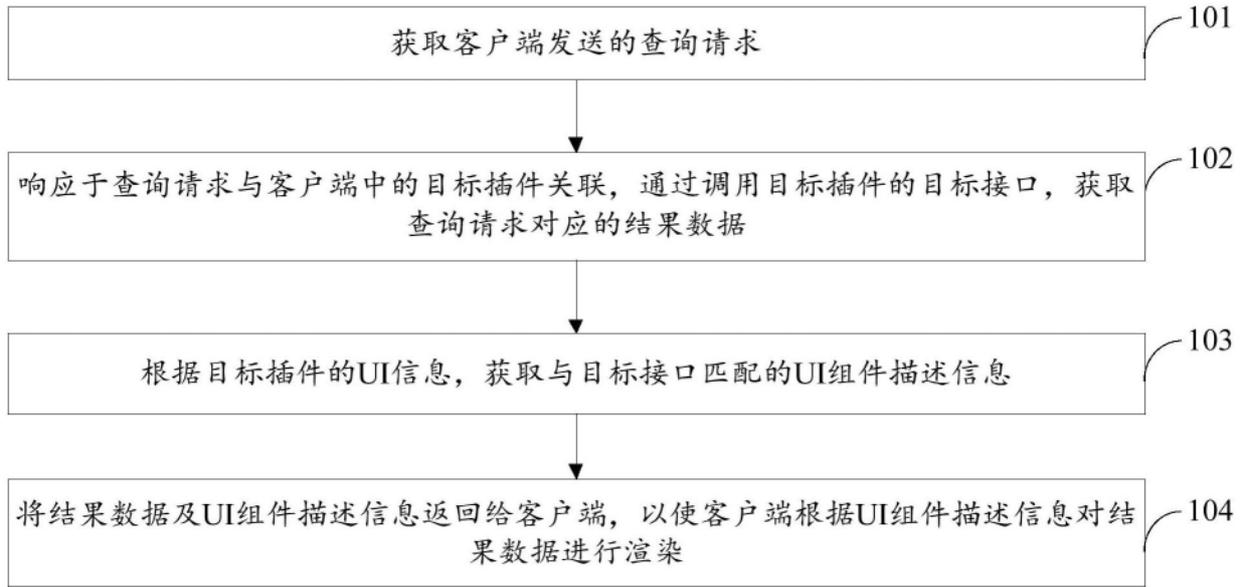


图1

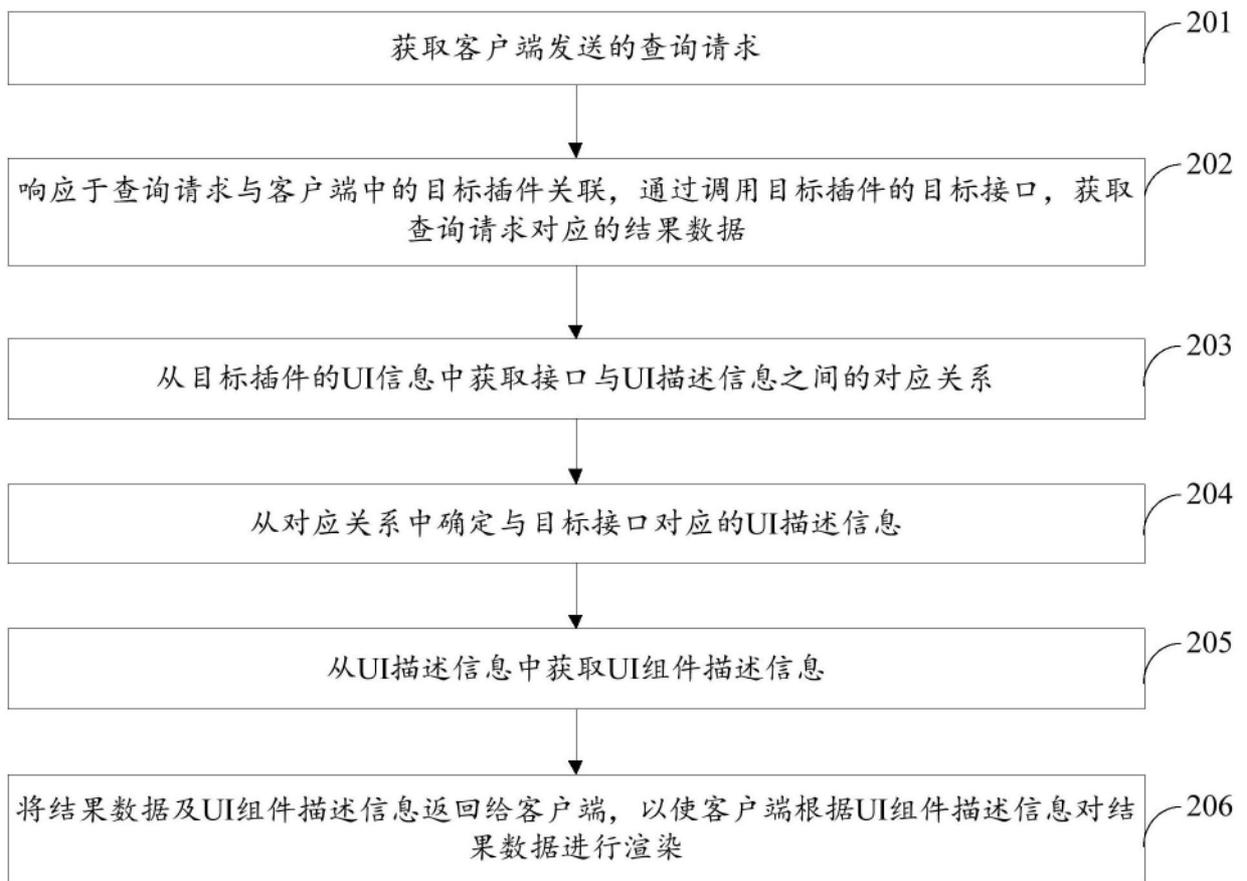


图2

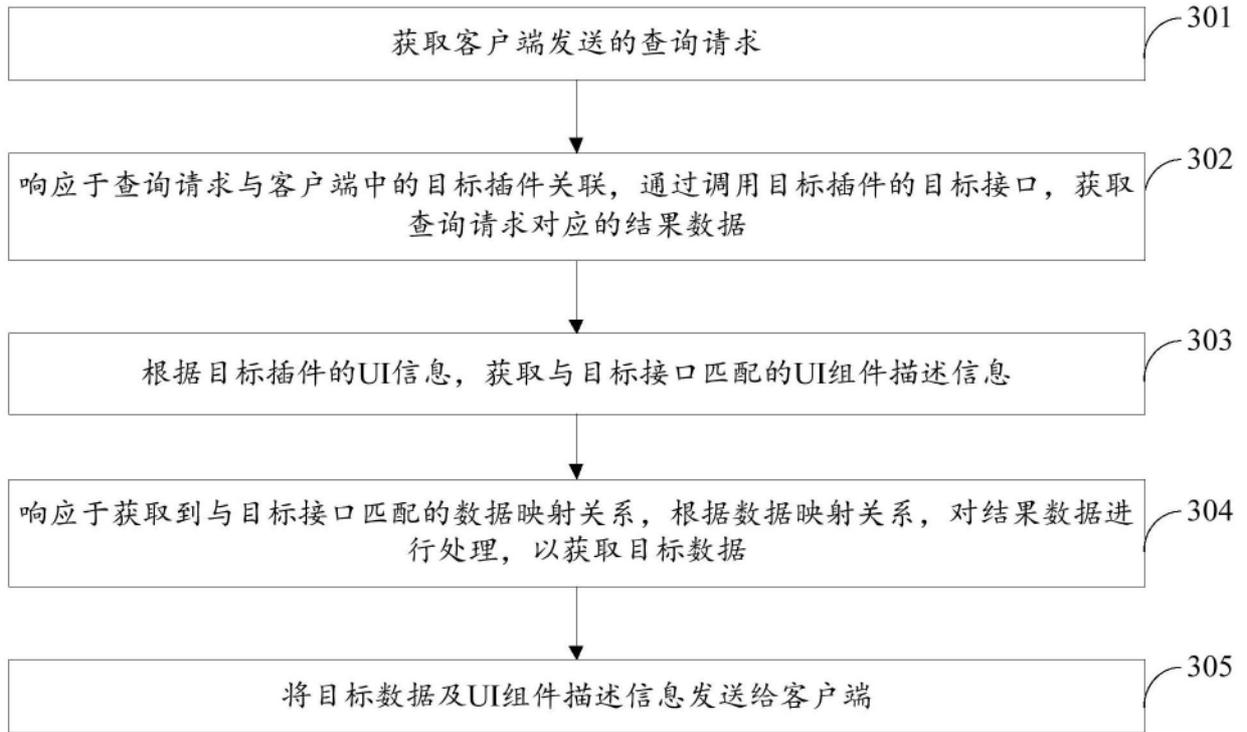


图3



图4

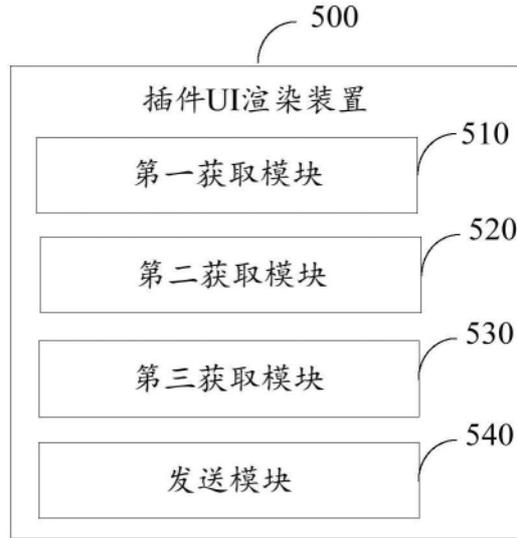


图5

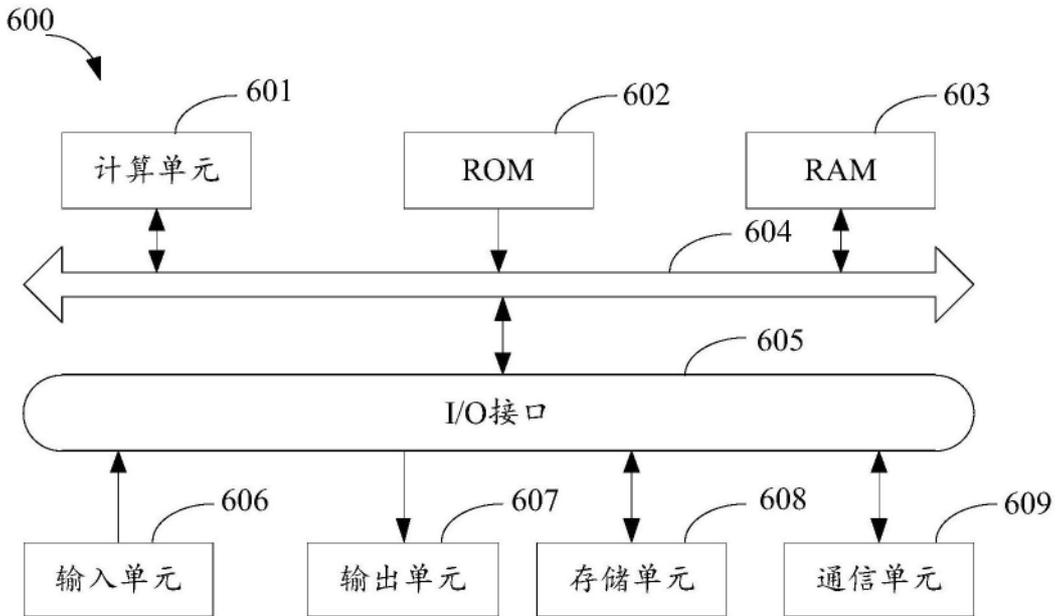


图6