(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 106161157 B
(45)授权公告日 2019.01.29

(21)申请号 201610578439.1

(22)申请日 2016.07.20

(65)同一申请的已公布的文献号
申请公布号 CN 106161157 A

(43)申请公布日 2016.11.23

(73)专利权人 珠海格力电器股份有限公司
地址 519070 广东省珠海市香洲区前山金鸡西路789号

(72)发明人 梁智将 彭敏 郭明娟 黄卫基 李昌伟 魏辉 罗宏波

(74)专利代理机构 北京煦润律师事务所 11522
代理人 何怀燕

(51)Int.Cl.
H04L 12/28(2006.01)

(56)对比文件
CN 103595717 A,2014.02.19,
CN 105116734 A,2015.12.02,
CN 105259828 A,2016.01.20,
CN 104167088 A,2014.11.26,
CN 105739460 A,2016.07.06,
US 2016132029 A1,2016.05.12,

审查员 李亢亢

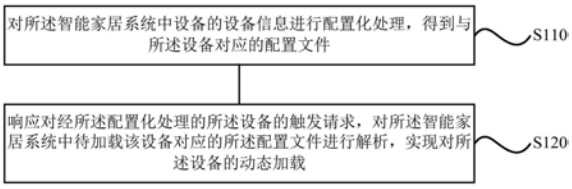
权利要求书5页 说明书15页 附图5页

(54)发明名称

智能家居系统的搭建方法、装置、智能家居系统及终端

(57)摘要

本发明公开了一种智能家居系统的搭建方法、装置、智能家居系统及终端，该方法包括：对所述智能家居系统中设备的设备信息进行配置化处理，得到与所述设备对应的配置文件；响应对经所述配置化处理的所述设备的触发请求，对该设备对应的所述配置文件进行解析，实现对所述设备的动态加载。本发明的方案，可以克服现有技术中响应速度慢、稳定性弱和可伸缩性差的缺陷，实现响应速度快、稳定性强和可伸缩性好的有益效果。



CN 106161157 B

1.一种智能家居系统的搭建方法,其特征在于,包括:

对所述智能家居系统中设备的设备信息进行配置化处理,得到与所述设备对应的配置文件;对所述智能家居系统中设备的设备信息进行配置化处理,包括:根据预设的设备类型,对所述智能家居系统中的设备进行分类;根据所述分类得到的分类结果,将每类设备的设备信息中除通用信息之外的不确定信息进行抽离;将抽离得到的所述不确定信息进行解耦合性处理;并将所述不确定信息中耦合程度大于预设值的部分,确定为所述设备的待配置信息;将所述待配置信息,通过数据库表和/或xml的形式进行配置;

响应对经所述配置化处理的所述设备的触发请求,对该设备对应的所述配置文件进行解析,实现对所述设备的动态加载。

2.根据权利要求1所述的方法,其特征在于,所述配置文件,包括:设备对应页面路径和设备对应实体类的至少之一;

相应地,对所述智能家居系统中待加载设备的所述配置文件进行解析,包括:

获取所述配置文件中该设备的设备对应页面路径;

通过预设的动态运行机制,在所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备对应实体类。

3.根据权利要求2所述的方法,其特征在于,获取所述配置文件中该设备的设备对应页面路径,包括:

根据所述配置文件中该设备的设备类型,查询所述配置文件;

根据查询得到的查询结果,获取与所述设备对应的所述设备对应页面路径。

4.根据权利要求2或3所述的方法,其特征在于,将每类设备的设备信息中除通用信息之外的不确定信息进行抽离,包括:

将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对象;

对所述设备对象进行设备标识;

设置用于访问所述设备对象的设备对应页面路径;

设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类;

设置用于查询所述设备对应实体类的设备登录方法;

相应地,所述配置文件,还包括:设备对象、设备标识、设备登录方法的至少之一。

5.根据权利要求1-3之一所述的方法,其特征在于,其中,

所述设备类型,包括:根据所述智能家居系统中设备的预设功能、和/或预设适用人群,确定得到的类型;和/或,

所述通用信息,包括:每类设备中,复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素;

相应地,所述不确定信息,包括:每类设备中,所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素;和/或,

所述待配置信息,包括:机型、功能、协议、UI、地域的至少之一。

6.根据权利要求4所述的方法,其特征在于,其中,

所述设备类型,包括:根据所述智能家居系统中设备的预设功能、和/或预设适用人群,确定得到的类型;和/或,

所述通用信息，包括：每类设备中，复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素；

相应地，所述不确定信息，包括：每类设备中，所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素；和/或，

所述待配置信息，包括：机型、功能、协议、UI、地域的至少之一。

7.根据权利要求5所述的方法，其特征在于，还包括：

对所述功能进行具体化处理，得到至少一个子功能，并使所述子功能被定制于所述设备；

其中，所述子功能，包括：能够单独运行的控制过程、和/或能够单独运行的设备界面。

8.根据权利要求6所述的方法，其特征在于，还包括：

对所述功能进行具体化处理，得到至少一个子功能，并使所述子功能被定制于所述设备；

其中，所述子功能，包括：能够单独运行的控制过程、和/或能够单独运行的设备界面。

9.根据权利要求7或8所述的方法，其特征在于，还包括：

对所述功能进行具体化处理后，在所述设备对应实体类中对所述子功能进行注册处理，和/或，对所述子功能对应的所述UI进行更改处理，和/或，对所述子功能和/或所述子功能对应的UI进行模块化处理。

10.根据权利要求7或8所述的方法，其特征在于，还包括：

对所述功能进行具体化处理后，保持所述子功能和/或所述子功能对应的UI不被更改，增加所述子功能对应设备的标志位、协议配置的至少之一，和/或，填充所述子功能对应设备的具体协议。

11.根据权利要求9所述的方法，其特征在于，还包括：

对所述功能进行具体化处理后，保持所述子功能和/或所述子功能对应的UI不被更改，增加所述子功能对应设备的标志位、协议配置的至少之一，和/或，填充所述子功能对应设备的具体协议。

12.根据权利要求10所述的方法，其特征在于，还包括：

通过所述标志位，对所述具体协议进行更改。

13.根据权利要求11所述的方法，其特征在于，还包括：

通过所述标志位，对所述具体协议进行更改。

14.根据权利要求1-3、6-8、11-13之一所述的方法，其特征在于，还包括：

当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

15.根据权利要求4所述的方法，其特征在于，还包括：

当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

16.根据权利要求5所述的方法，其特征在于，还包括：

当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

17.根据权利要求9所述的方法，其特征在于，还包括：

当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

18.根据权利要求10所述的方法，其特征在于，还包括：

当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

19.一种智能家居系统的搭建装置，其特征在于，包括：

配置化处理单元，用于对所述智能家居系统中设备的设备信息进行配置化处理，得到与所述设备对应的配置文件；配置化处理单元，包括：分类模块、抽离模块、确定模块和配置模块；其中，分类模块，用于根据预设的设备类型，对所述智能家居系统中的设备进行分类；抽离模块，用于根据所述分类得到的分类结果，将每类设备的设备信息中除通用信息之外的不确定信息进行抽离；确定模块，用于将抽离得到的所述不确定信息进行解耦合性处理；并将所述不确定信息中耦合程度大于预设值的部分，确定为所述设备的待配置信息；配置模块，用于将所述待配置信息，通过数据库表和/或xml的形式进行配置；

动态加载单元，用于响应对经所述配置化处理的所述设备的触发请求，对该设备对应的所述配置文件进行解析，实现对所述设备的动态加载。

20.根据权利要求19所述的装置，其特征在于，所述配置文件，包括：设备对应页面路径和设备对应实体类的至少之一；

相应地，动态加载单元，包括：

路径获取模块，用于获取所述配置文件中该设备的设备对应页面路径；

实体类加载模块，用于通过预设的动态运行机制，在所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备对应实体类。

21.根据权利要求20所述的装置，其特征在于，路径获取模块，包括：

查询子模块，用于根据所述配置文件中该设备的设备类型，查询所述配置文件；

获取子模块，用于根据查询得到的查询结果，获取与所述设备对应的所述设备对应页面路径。

22.根据权利要求20或21所述的装置，其特征在于，抽离模块，包括：

对象抽象子模块，用于将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对象；

标识设置子模块，用于对所述设备对象进行设备标识；

路径设置子模块，用于设置用于访问所述设备对象的设备对应页面路径；

实体类设置子模块，用于设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类；

方法设置子模块，用于设置用于查询所述设备对应实体类的设备登录方法；

相应地，所述配置文件，还包括：设备对象、设备标识、设备登录方法的至少之一。

23.根据权利要求19-21之一所述的装置，其特征在于，其中，

所述设备类型，包括：根据所述智能家居系统中设备的预设功能、和/或预设适用人群，确定得到的类型；和/或，

所述通用信息，包括：每类设备中，复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素；

相应地，所述不确定信息，包括：每类设备中，所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素；和/或，

所述待配置信息，包括：机型、功能、协议、UI、地域的至少之一。

24.根据权利要求22所述的装置，其特征在于，其中，

所述设备类型，包括：根据所述智能家居系统中设备的预设功能、和/或预设适用人群，确定得到的类型；和/或，

所述通用信息，包括：每类设备中，复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素；

相应地，所述不确定信息，包括：每类设备中，所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素；和/或，

所述待配置信息，包括：机型、功能、协议、UI、地域的至少之一。

25.根据权利要求23所述的装置，其特征在于，还包括：

功能具体化单元，用于对所述功能进行具体化处理，得到至少一个子功能，并使所述子功能被定制于所述设备；

其中，所述子功能，包括：能够单独运行的控制过程、和/或能够单独运行的设备界面。

26.根据权利要求24所述的装置，其特征在于，还包括：

功能具体化单元，用于对所述功能进行具体化处理，得到至少一个子功能，并使所述子功能被定制于所述设备；

其中，所述子功能，包括：能够单独运行的控制过程、和/或能够单独运行的设备界面。

27.根据权利要求25或26所述的装置，其特征在于，还包括：

模块化处理单元，用于对所述功能进行具体化处理后，在所述设备对应实体类中对所述子功能进行注册处理，和/或，对所述子功能对应的所述UI进行更改处理，和/或，对所述子功能和/或所述子功能对应的UI进行模块化处理。

28.根据权利要求25或26所述的装置，其特征在于，还包括：

协议兼容单元，用于对所述功能进行具体化处理后，保持所述子功能和/或所述子功能对应的UI不被更改，增加所述子功能对应设备的标志位、协议配置的至少之一，和/或，填充所述子功能对应设备的具体协议。

29.根据权利要求27所述的装置，其特征在于，还包括：

协议兼容单元，用于对所述功能进行具体化处理后，保持所述子功能和/或所述子功能对应的UI不被更改，增加所述子功能对应设备的标志位、协议配置的至少之一，和/或，填充所述子功能对应设备的具体协议。

30.根据权利要求28所述的装置，其特征在于，还包括：

所述协议兼容单元，还用于通过所述标志位，对所述具体协议进行更改。

31.根据权利要求29所述的装置，其特征在于，还包括：

所述协议兼容单元，还用于通过所述标志位，对所述具体协议进行更改。

32.根据权利要求19-21、24-26、29-31之一所述的装置，其特征在于，还包括：

设备增加单元，用于当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

33.根据权利要求22所述的装置，其特征在于，还包括：

设备增加单元，用于当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

34.根据权利要求23所述的装置，其特征在于，还包括：

设备增加单元，用于当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

35.根据权利要求27所述的装置，其特征在于，还包括：

设备增加单元，用于当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

36.根据权利要求28所述的装置，其特征在于，还包括：

设备增加单元，用于当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

37.一种智能家居系统，其特征在于，包括：如权利要求19-36任一所述的智能家居系统的搭建装置。

38.一种终端，其特征在于，包括：如权利要求37所述的智能家居系统。

智能家居系统的搭建方法、装置、智能家居系统及终端

技术领域

[0001] 本发明属于智能家居技术领域，具体涉及一种智能家居系统的搭建方法、装置、智能家居系统及终端，尤其涉及一种基于智能家居系统的移动端框架搭建方法、装置、具有该装置的智能家居系统、以及具有该智能家居系统的移动端。

背景技术

[0002] 智能家居，可以是以住宅为平台，利用综合布线技术、网络通信技术、安全防范技术、自动控制技术、音视频技术，将家居生活有关的设施集成，构建高效的住宅设施与家庭日程事务的管理系统，提升家居安全性、便利性、舒适性、艺术性，并实现环保节能的居住环境。目前智能家居市场需求很大，但是同时市场也很混乱，智能单品很多。

[0003] 作为移动端APP，在机型配置化方面，若需要兼容尽可能多的设备、而且需要响应快速的迭代开发，就需要一个能够支持快速开发迭代、可伸缩性强、稳定性强的移动端框架。

[0004] 现有技术中，存在响应速度慢、稳定性弱和可伸缩性差等缺陷。

发明内容

[0005] 本发明的目的在于，针对上述缺陷，提供一种智能家居系统的搭建方法、装置、智能家居系统及终端，以解决现有技术中无法快速响应的问题，达到提升响应速度的效果。

[0006] 本发明提供一种智能家居系统的搭建方法，包括：对所述智能家居系统中设备的设备信息进行配置化处理，得到与所述设备对应的配置文件；响应对经所述配置化处理的所述设备的触发请求，对该设备对应的所述配置文件进行解析，实现对所述设备的动态加载。

[0007] 可选地，对所述智能家居系统中设备的设备信息进行配置化处理，包括：根据预设的设备类型，对所述智能家居系统中的设备进行分类；根据所述分类得到的分类结果，将每类设备的设备信息中除通用信息之外的不确定信息进行抽离；将抽离得到的所述不确定信息进行解耦合性处理；并将所述不确定信息中耦合程度大于预设值的部分，确定为所述设备的待配置信息；将所述待配置信息，通过数据库表和/或xml的形式进行配置。

[0008] 可选地，所述配置文件，包括：设备对应页面路径和设备对应实体类的至少之一；相应地，对所述智能家居系统中待加载设备的所述配置文件进行解析，包括：获取所述配置文件中该设备的设备对应页面路径；通过预设的动态运行机制，在所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备对应实体类。

[0009] 可选地，获取所述配置文件中该设备的设备对应页面路径，包括：根据所述配置文件中该设备的设备类型，查询所述配置文件；根据查询得到的查询结果，获取与所述设备对应的所述设备对应页面路径。

[0010] 可选地，将每类设备的设备信息中除通用信息之外的不确定信息进行抽离，包括：将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对

象;对所述设备对象进行设备标识;设置用于访问所述设备对象的设备对应页面路径;设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类;设置用于查询所述设备对应实体类的设备登录方法;相应地,所述配置文件,还包括:设备对象、设备标识、设备登录方法的至少之一。

[0011] 可选地,其中,所述设备类型,包括:根据所述智能家居系统中设备的预设功能、和/或预设适用人群,确定得到的类型;和/或,所述通用信息,包括:每类设备中,复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素;相应地,所述不确定信息,包括:每类设备中,所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素;和/或,所述待配置信息,包括:机型、功能、协议、UI、地域的至少之一。

[0012] 可选地,还包括:对所述功能进行具体化处理,得到至少一个子功能,并使所述子功能被定制于所述设备;其中,所述子功能,包括:能够单独运行的控制过程、和/或能够单独运行的设备界面。

[0013] 可选地,还包括:对所述功能进行具体化处理后,在所述设备对应实体类中对所述子功能进行注册处理,和/或,对所述子功能对应的所述UI进行更改处理,和/或,对所述子功能和/或所述子功能对应的UI进行模块化处理。

[0014] 可选地,还包括:对所述功能进行具体化处理后,保持所述子功能和/或所述子功能对应的UI不被更改,增加所述子功能对应设备的标志位、协议配置的至少之一,和/或,填充所述子功能对应设备的具体协议。

[0015] 可选地,还包括:通过所述标志位,对所述具体协议进行更改。

[0016] 可选地,还包括:当所述智能家居系统中有待增加设备时,根据所述待增加设备的设备信息对所述配置文件进行更改,以将所述待增加设备增加至所述智能家居系统。

[0017] 与上述方法相匹配,本发明另一方面提供一种智能家居系统的搭建装置,包括:配置化处理单元,用于对所述智能家居系统中设备的设备信息进行配置化处理,得到与所述设备对应的配置文件;动态加载单元,用于响应对经所述配置化处理的所述设备的触发请求,对该设备对应的所述配置文件进行解析,实现对所述设备的动态加载。

[0018] 可选地,配置化处理单元,包括:分类模块,用于根据预设的设备类型,对所述智能家居系统中的设备进行分类;抽离模块,用于根据所述分类得到的分类结果,将每类设备的设备信息中除通用信息之外的不确定信息进行抽离;确定模块,用于将抽离得到的所述不确定信息进行解耦合性处理;并将所述不确定信息中耦合程度大于预设值的部分,确定为所述设备的待配置信息;配置模块,用于将所述待配置信息,通过数据库表和/或xml的形式进行配置。

[0019] 可选地,所述配置文件,包括:设备对应页面路径和设备对应实体类的至少之一;相应地,动态加载单元,包括:路径获取模块,用于获取所述配置文件中该设备的设备对应页面路径;实体类加载模块,用于通过预设的动态运行机制,在所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备对应实体类。

[0020] 可选地,路径获取模块,包括:查询子模块,用于根据所述配置文件中该设备的设备类型,查询所述配置文件;获取子模块,用于根据查询得到的查询结果,获取与所述设备对应的所述设备对应页面路径。

[0021] 可选地，抽离模块，包括：对象抽象子模块，用于将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对象；标识设置子模块，用于对所述设备对象进行设备标识；路径设置子模块，用于设置用于访问所述设备对象的设备对应页面路径；实体类设置子模块，用于设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类；方法设置子模块，用于设置用于查询所述设备对应实体类的设备登录方法；相应地，所述配置文件，还包括：设备对象、设备标识、设备登录方法的至少之一。

[0022] 可选地，其中，所述设备类型，包括：根据所述智能家居系统中设备的预设功能、和/或预设适用人群，确定得到的类型；和/或，所述通用信息，包括：每类设备中，复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素；相应地，所述不确定信息，包括：每类设备中，所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素；和/或，所述待配置信息，包括：机型、功能、协议、UI、地域的至少之一。

[0023] 可选地，还包括：功能具体化单元，用于对所述功能进行具体化处理，得到至少一个子功能，并使所述子功能被定制于所述设备；其中，所述子功能，包括：能够单独运行的控制过程、和/或能够单独运行的设备界面。

[0024] 可选地，还包括：模块化处理单元，用于对所述功能进行具体化处理后，在所述设备对应实体类中对所述子功能进行注册处理，和/或，对所述子功能对应的所述UI进行更改处理，和/或，对所述子功能和/或所述子功能对应的UI进行模块化处理。

[0025] 可选地，还包括：协议兼容单元，用于对所述功能进行具体化处理后，保持所述子功能和/或所述子功能对应的UI不被更改，增加所述子功能对应设备的标志位、协议配置的至少之一，和/或，填充所述子功能对应设备的具体协议。

[0026] 可选地，还包括：所述协议兼容单元，还用于通过所述标志位，对所述具体协议进行更改。

[0027] 可选地，还包括：设备增加单元，用于当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

[0028] 与上述装置相匹配，本发明再一方面提供一种智能家居系统，包括：以上所述的智能家居系统的搭建装置。

[0029] 与上述系统相匹配，本发明又一方面提供一种终端，包括：以上所述的智能家居系统。

[0030] 本发明的方案，通过将复杂的业务逻辑或者不确定的因素进行配置化，运行的时候通过解析配置文件实现动态的加载，可以解决无法快速响应并满足众多的机型需求的问题。

[0031] 进一步，本发明的方案，通过不断完善智能家居的功能库和功能协议，就可以兼容所有单品，解决团队合作开发困难的问题，解决APP大小急速膨胀的问题。

[0032] 进一步，本发明的方案，通过搭建足够稳定的系统，可以同时开发多个智能单品，使得单品的开发和功能开发不冲突，可以进一步提升智能家居系统的响应速度和可伸缩性。

[0033] 进一步，本发明的方案，通过将不确定的因素可配置化，在一定程度上降低了代码

的复杂性，并且让合作开发变得更简单，更多的开发人员可以专注于功能的具体实现；而APP可以通过配置的方式动态去加载需要的功能，让APP更强大。

[0034] 由此，本发明的方案，通过将智能家居中各设备间的复杂的业务逻辑或者不确定的因素进行配置化，并在运行时通过解析配置文件实现相应设备的动态加载，解决现有技术中无法快速响应的问题，从而，克服现有技术中响应速度慢、稳定性弱和可伸缩性差的缺陷，实现响应速度快、稳定性强和可伸缩性好的有益效果。

[0035] 本发明的其它特征和优点将在随后的说明书中阐述，并且，部分地从说明书中变得显而易见，或者通过实施本发明而了解。

[0036] 下面通过附图和实施例，对本发明的技术方案做进一步的详细描述。

附图说明

[0037] 图1为本发明的智能家居系统的搭建方法的一实施例的流程示意图；

[0038] 图2为本发明的方法中配置化处理的一实施例的流程示意图；

[0039] 图3为本发明的方法中实体类加载处理的一实施例的流程示意图；

[0040] 图4为本发明的方法中路径获取处理的一实施例的流程示意图；

[0041] 图5为本发明的方法中抽离处理的一实施例的流程示意图；

[0042] 图6为本发明的智能家居系统的搭建装置的一实施例的结构示意图；

[0043] 图7为本发明的装置中抽离模块的一实施例的结构示意图；

[0044] 图8为本发明的装置中路径获取模块的一实施例的结构示意图；

[0045] 图9为本发明的智能家居系统的一实施例的结构示意图；

[0046] 图10为本发明的智能家居系统的另一实施例的结构示意图。

[0047] 结合附图，本发明实施例中附图标记如下：

[0048] 102-配置化处理单元；1022-分类模块；1024-抽离模块；1026-确定模块；1028-配置模块；104-动态加载单元；1042-路径获取模块；1044-实体类加载模块；106-设备增加单元；108-功能具体化单元；110-模块化处理单元；112-协议兼容单元。

具体实施方式

[0049] 为使本发明的目的、技术方案和优点更加清楚，下面将结合本发明具体实施例及相应的附图对本发明技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0050] 根据本发明的实施例，提供了一种智能家居系统的搭建方法，如图1所示本发明的方法的一实施例的流程示意图。该智能家居系统的搭建方法可以包括：

[0051] 在步骤S110处，对所述智能家居系统中设备的设备信息进行配置化处理，得到与所述设备对应的配置文件。

[0052] 例如：可以增加配置文件，用来配置设备信息。

[0053] 由此，通过对设备信息的配置化处理，可以提升动态加载的便捷性和可靠性，用户体验好。

[0054] 可选地，可以结合图2所示本发明的方法中配置化处理的一实施例的流程示意图，

进一步说明步骤S110中对所述智能家居系统中设备的设备信息进行配置化处理的具体过程。

[0055] 步骤S210,根据预设的设备类型,对所述智能家居系统中的设备进行分类。

[0056] 例如:首先通过设备类型对设备进行分类。

[0057] 可选地,所述设备类型,可以包括:根据所述智能家居系统中设备的预设功能、和/或预设适用人群,确定得到的类型。通过基于用户需求确定的设备类型,通用性强,人性化好。

[0058] 步骤S220,根据所述分类得到的分类结果,将每类设备的设备信息中除通用信息之外的不确定信息进行抽离。

[0059] 其中,抽离,可以是将同一类设备的设备信息中,确定该类设备的通用信息(例如:该类设备的通用功能对应的设备信息)后,将该类设备的设备信息中除了所述通用信息之外的其它信息(即不确定信息)进行提取。

[0060] 例如:抽离并确定需要配置化的地方。

[0061] 可选地,所述通用信息,可以包括:每类设备中,复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素。

[0062] 相应地,所述不确定信息,可以包括:每类设备中,所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素。通过基于预设的复杂程度和不确定程度区分通用信息与不确定信息,有利于提高对设备信息进行配置化处理的精准性和可靠性。

[0063] 可选地,可以结合图5所示本发明的方法中抽离处理的一实施例的流程示意图,进一步说明步骤S220中将每类设备的设备信息中除通用信息之外的不确定信息进行抽离的具体过程。

[0064] 步骤S510,将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对象。

[0065] 例如:具体的xml配置格式,可以如:

[0066] <?xml version＝”1.0”encoding＝”utf-8”>

[0067] <DeviceType id＝1name＝”空调”>

[0068] 步骤S520,对所述设备对象进行设备标识。

[0069] 例如:具体的xml配置格式,可以如:

[0070] <Mid>ab492s2sdfa123</Mid>(设备唯一标识)

[0071] 步骤S530,设置用于访问所述设备对象的设备对应页面路径。

[0072] 例如:具体的xml配置格式,可以如:

[0073] <DevicePath>/com/device/activity/DeviceAcActivity.java</DevicePath>(设备对应页面路径)

[0074] 步骤S540,设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类。

[0075] 例如:具体的xml配置格式,可以如:

[0076] <DeviceEntity>/com/device/activity/DeviceEntity.java</DeviceEntity>(设备对应实体类)

[0077] 步骤S550,设置用于查询所述设备对应实体类的设备登录方法。

[0078] 例如:<DeviceLogin>DeviceAcLogin</DeviceLogin>(设备登录方法)

[0079] </DeviceType>

[0080] 相应地,所述配置文件,还可以包括:设备对象、设备标识、设备登录方法的至少之一。

[0081] 由此,通过对设备信息中不确定信息的抽离,有利于提升配置化处理的便捷性和高效性。

[0082] 步骤S230,将抽离得到的所述不确定信息进行解耦合性处理;并将所述不确定信息中耦合程度大于预设值的部分,确定为所述设备的待配置信息。

[0083] 其中,耦合性,也叫耦合度,可以是对模块(例如:所述不确定信息中的子信息)间关联程度的度量。

[0084] 其中,解耦合性,即解除模块(例如:所述不确定信息中的子信息)间的关联,将模块变成单个的子模块(例如:将所述不确定信息变成多个子信息),使变成的每个子模块能够区别于所述模块中其它子模块而独立存在。这样,对所述不确定信息进行修改时,只要修改对应的子信息即可,一方面,对该子信息的修改不会影响到所述不确定信息中除该子信息之外的其它子信息的运行,有利于提升所述不确定信息运行的可靠性;另一方面,只对该子信息修改即可实现对所述不确定信息的修改,有利于提高修改的高效性。

[0085] 例如:可以将复杂的业务逻辑或者不确定的因素进行配置化。

[0086] 例如:配置文件,通过xml(即可扩展标记语言,是一种用于标记电子文件使其具有结构性的标记语言)的形式进行配置。

[0087] 例如:可以将复杂的逻辑抽离,解耦合性,如果实在无法解耦合的地方就是需要写到配置文件的东西。

[0088] 可选地,所述待配置信息,可以包括:机型、功能、协议、UI、地域的至少之一。

[0089] 例如:从功能的角度去考虑APP的实现,APP需要配置化的地方,包括:机型不确定性,功能不确定性,协议不确定性。

[0090] 例如:从适用人群选择的角度去考虑APP的实现,APP需要配置化的地方包括:UI不确定性(例如:不确定哪种UI适用于哪种人群),地域不确定性(例如:不确定服务器是否兼容该地区)等。

[0091] 由此,通过多种形式的待配置信息,可以进一步提升配置化处理的灵活性和通用性,进而提升用户的使用体验。

[0092] 步骤S240,将所述待配置信息,通过数据库表和/或xml的形式进行配置。

[0093] 例如:将复杂的业务逻辑关系和无法确定的因素通过数据库表或者xml的方式存储,代码里面并不会体现这些复杂的逻辑关系。

[0094] 例如:可以通过机型不确定性来说明配置化方案。

[0095] 例如:为了满足各种机型的需求,需要在代码增加各种机型的判断,去识别不同的机型,将这种复杂的逻辑抽离成xml配置文件。

[0096] 由此,通过对设备的不确定因素机械能数据库表、xml等形式的配置,可以使得不确定因素的调用更加方便、更加精准。

[0097] 在步骤S120处,响应对经所述配置化处理的所述设备的触发请求,对该设备对应

的所述配置文件进行解析，实现对所述设备的动态加载。

[0098] 例如：运行的时候通过解析配置文件实现动态的加载。

[0099] 例如：在加载设备信息前，不需要运行库不需要知道执行的方法；只有在用户点击触发的时候，通过路径去获取自己要跳转的机型页面。

[0100] 例如：APP通过反射机制，在程序运行时动态调用配置文件，并通过配置文件来访问设备对应页面路径。

[0101] 由此，通过对智能家居系统中设备的配置化处理，通过对配置化处理得到的配置文件进行解析实现动态加载，可以提升加载的响应速度，进而大大提升用户使用的便捷性。

[0102] 可选地，所述配置文件，可以包括：设备对应页面路径和设备对应实体类的至少之一。

[0103] 例如：将设备类型对应的设备对应页面路径、设备对应实体类存储在一个配置文件中。

[0104] 由此，通过对设备对应页面路径和设备对应实体类的配置，可以提升解析的便捷性和可靠性，进而提升用户使用的便捷性。

[0105] 相应地，可以结合图3所示本发明的方法中实体类加载处理的一实施例的流程示意图，进一步说明步骤S120中对该设备对应的所述配置文件进行解析的具体过程。

[0106] 步骤S310，获取所述配置文件中该设备的设备对应页面路径。

[0107] 例如：读取对应的设备信息，获取到设备对应的页面路径。

[0108] 可选地，可以结合图4所示本发明的方法中路径获取处理的一实施例的流程示意图，进一步说明步骤S310中获取所述配置文件中该设备的设备对应页面路径的具体过程。

[0109] 步骤S410，根据所述配置文件中该设备的设备类型，查询所述配置文件。

[0110] 例如：每次加载设备的时候，根据设备的类型，查询配置文件。

[0111] 步骤S420，根据查询得到的查询结果，获取与所述设备对应的所述设备对应页面路径。

[0112] 由此，通过查询配置文件的方式实现设备的动态加载，加载效率高，精准性好。

[0113] 步骤S320，通过预设的动态运行机制，在所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备对应实体类。

[0114] 例如：通过动态运行机制，在运行过程中加载对应路径下的页面实体类，而无需在代码里面写复杂的逻辑去实现。

[0115] 由此，通过获取设备对应页面路径的方式实现设备对应实体类的动态加载，加载方式简便、且可靠性高。

[0116] 在一个可选实施方式中，还可以包括：对所述功能进行具体化处理，得到至少一个子功能，并使所述子功能被定制于所述设备。

[0117] 其中，所述子功能，可以包括：能够单独运行的控制过程、和/或能够单独运行的设备界面。

[0118] 例如：将功能具体化，不依赖于实体，每个功能都可以单独运行(例如：该运行，可以包括设备控制、界面UI等)，而设备只需要定制自己需要的功能，而不直接具备任何功能。

[0119] 由此，通过对设备功能的具体化处理，可以实现功能定制，进而满足众多的机型需求，使用方便，用户体验好。

[0120] 在一个可选实施方式中，还可以包括：对所述功能进行具体化处理后，在所述设备对应实体类中对所述子功能进行注册处理，和/或，对所述子功能对应的所述UI进行更改处理，和/或，对所述子功能和/或所述子功能对应的UI进行模块化处理。

[0121] 例如：将功能具体化之后，设备对应实体类注册功能。当界面刷新、下发指令时，通知对应的UI(User Interface，用户界面)进行更改，将功能、UI模块化。

[0122] 例如：可以实现设备功能通用化，并实现功能模块化。

[0123] 由此，通过对功能的注册处理、模块化处理等方式，可以解决团队合作开发困难的问题，解决APP大小急速膨胀的问题，可靠性高，环保性好。

[0124] 在一个可选实施方式中，还可以包括：对所述功能进行具体化处理后，保持所述子功能和/或所述子功能对应的UI不被更改，增加所述子功能对应设备的标志位、协议配置的至少之一，和/或，填充所述子功能对应设备的具体协议。

[0125] 例如：将功能具体化之后，功能和UI界面不改动，增加厂家设备标志位，增加协议配置，并填充具体协议，APP通过标志位改变采用的具体协议。

[0126] 由此，通过增加标志位和协议配置，可以提升兼容性，进而提升使用便捷性和可靠性。

[0127] 在一个可选实施方式中，还可以包括：通过所述标志位，对所述具体协议进行更改。

[0128] 例如：可以通过增加协议转换工具，实现协议兼容。

[0129] 由此，通过对具体协议的更改，可以提高兼容的便捷性，减小存储容量，扩大适用范围。

[0130] 在一个可选实施方式中，还可以包括：当所述智能家居系统中有待增加设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

[0131] 例如：例如：增加设备配置文件，每次添加设备只需要修改配置文件，而不需要去兼容设备，适配设备。

[0132] 由此，通过修改配置文件的方式实现新设备的增加，增加方式简便，控制可靠性高。

[0133] 经大量的试验验证，采用本实施例的技术方案，通过将复杂的业务逻辑或者不确定的因素进行配置化，运行的时候通过解析配置文件实现动态的加载，可以解决无法快速响应并满足众多的机型需求的问题。

[0134] 根据本发明的实施例，还提供了对应于智能家居系统的搭建方法的一种智能家居系统的搭建装置。参见图6所示本发明的装置的一实施例的结构示意图。该智能家居系统的搭建装置可以包括：配置化处理单元102和动态加载单元104。

[0135] 在一个例子中，配置化处理单元102，可以用于对所述智能家居系统中设备的设备信息进行配置化处理，得到与所述设备对应的配置文件。该配置化处理单元102的具体功能及处理参见步骤S110。

[0136] 例如：可以增加配置文件，用来配置设备信息。

[0137] 由此，通过对设备信息的配置化处理，可以提升动态加载的便捷性和可靠性，用户体验好。

[0138] 可选地，配置化处理单元102，可以包括：分类模块1022、抽离模块1024、确定模块1026和配置模块1028。

[0139] 在一个可选例子中，分类模块1022，可以用于根据预设的设备类型，对所述智能家居系统中的设备进行分类。该分类模块1022的具体功能及处理参见步骤S210。

[0140] 例如：首先通过设备类型对设备进行分类。

[0141] 可选地，所述设备类型，可以包括：根据所述智能家居系统中设备的预设功能、和/或预设适用人群，确定得到的类型。通过基于用户需求确定的设备类型，通用性强，人性化好。

[0142] 在一个可选例子中，抽离模块1024，可以用于根据所述分类得到的分类结果，将每类设备的设备信息中除通用信息之外的不确定信息进行抽离。该抽离模块1024的具体功能及处理参见步骤S220。

[0143] 其中，抽离，可以是将同一类设备的设备信息中，确定该类设备的通用信息(例如：该类设备的通用功能对应的设备信息)后，将该类设备的设备信息中除了所述通用信息之外的其它信息(即不确定信息)进行提取。

[0144] 例如：抽离并确定需要配置化的地方。

[0145] 可选地，所述通用信息，可以包括：每类设备中，复杂程度不高于预设复杂程度的业务逻辑、和/或不确定程度不高于预设不确定程度的因素。

[0146] 相应地，所述不确定信息，可以包括：每类设备中，所述复杂程度高于所述预设复杂程度的业务逻辑、和/或所述不确定程度高于所述预设不确定程度的因素。通过基于预设的复杂程度和不确定程度区分通用信息与不确定信息，有利于提高对设备信息进行配置化处理的精准性和可靠性。

[0147] 可选地，可以结合图7所示本发明的装置中抽离模块的一实施例的结构示意图，进一步说明抽离模块1024的具体结构。抽离模块1024，可以包括：对象抽象子模块10242、标识设置子模块10244、路径设置子模块10246和实体类设置子模块10248。

[0148] 在一个可选具体例子中，对象抽象子模块10242，可以用于将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对象。该对象抽象子模块10242的具体功能及处理参见步骤S510。

[0149] 例如：具体的xml配置格式，可以如：

[0150] <?xml version＝”1.0”encoding＝”utf-8”>

[0151] <DeviceType id＝1name＝”空调”>

[0152] 在一个可选具体例子中，标识设置子模块10244，可以用于对所述设备对象进行设备标识。该标识设置子模块10244的具体功能及处理参见步骤S520。

[0153] 例如：具体的xml配置格式，可以如：

[0154] <Mid>ab492s2sdfa123</Mid>(设备唯一标识)

[0155] 在一个可选具体例子中，路径设置子模块10246，可以用于设置用于访问所述设备对象的设备对应页面路径。该对象抽象子模块10242的具体功能及处理参见步骤S530。

[0156] 例如：具体的xml配置格式，可以如：

[0157] <DevicePath>/com/device/activity/DeviceAcActivity.java</DevicePath>(设备对应页面路径)

[0158] 在一个可选具体例子中,实体类设置子模块10248,可以用于设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类。该实体类设置子模块10248的具体功能及处理参见步骤S540。

[0159] 例如:具体的xml配置格式,可以如:

[0160] <DeviceEntity>/com/device/activity/DeviceEntity.java</DeviceEntity>(设备对应实体类)

[0161] 在一个可选具体例子中,方法设置子模块10250,可以用于设置用于查询所述设备对应实体类的设备登录方法。该方法设置子模块10250的具体功能及处理参见步骤S550。

[0162] 例如:<DeviceLogin>DeviceAcLogin</DeviceLogin>(设备登录方法)

[0163] </DeviceType>

[0164] 相应地,所述配置文件,还可以包括:设备对象、设备标识、设备登录方法的至少之一。

[0165] 由此,通过对设备信息中不确定信息的抽离,有利于提升配置化处理的便捷性和高效性。

[0166] 在一个可选例子中,确定模块1026,可以用于将抽离得到的所述不确定信息进行解耦合性处理;并将所述不确定信息中耦合程度大于预设值的部分,确定为所述设备的待配置信息。该确定模块1026的具体功能及处理参见步骤S230。

[0167] 其中,耦合性,也叫耦合度,可以是对模块(例如:所述不确定信息中的子信息)间关联程度的度量。

[0168] 其中,解耦合性,即解除模块(例如:所述不确定信息中的子信息)间的关联,将模块变成单个的子模块(例如:将所述不确定信息变成多个子信息),使变成的每个子模块能够区别于所述模块中其它子模块而独立存在。这样,对所述不确定信息进行修改时,只要修改对应的子信息即可,一方面,对该子信息的修改不会影响到所述不确定信息中除该子信息之外的其它子信息的运行,有利于提升所述不确定信息运行的可靠性;另一方面,只对该子信息修改即可实现对所述不确定信息的修改,有利于提高修改的高效性。

[0169] 例如:可以将复杂的业务逻辑或者不确定的因素进行配置化。

[0170] 例如:配置文件,通过xml(即可扩展标记语言,是一种用于标记电子文件使其具有结构性的标记语言)的形式进行配置。

[0171] 例如:可以将复杂的逻辑抽离,解耦合性,如果实在无法解耦合的地方就是需要写到配置文件的东西。

[0172] 可选地,所述待配置信息,可以包括:机型、功能、协议、UI、地域的至少之一。

[0173] 例如:从功能的角度去考虑APP的实现,APP需要配置化的地方,包括:机型不确定性,功能不确定性,协议不确定性。

[0174] 例如:从适用人群选择的角度去考虑APP的实现,APP需要配置化的地方包括:UI不确定性(例如:不确定哪种UI适用于哪种人群),地域不确定性(例如:不确定服务器是否兼容该地区)等。

[0175] 由此,通过多种形式的待配置信息,可以进一步提升配置化处理的灵活性和通用性,进而提升用户的使用体验。

[0176] 在一个可选例子中,配置模块1028,可以用于将所述待配置信息,通过数据库表

和/或xml的形式进行配置。该配置模块1028的具体功能及处理参见步骤S240。

[0177] 例如:将复杂的业务逻辑关系和无法确定的因素通过数据库表或者xml的方式存储,代码里面并不会体现这些复杂的逻辑关系。

[0178] 例如:可以通过机型不确定性来说明配置化方案。

[0179] 例如:为了满足各种机型的需求,需要在代码增加各种机型的判断,去识别不同的机型,将这种复杂的逻辑抽离成xml配置文件。

[0180] 由此,通过对设备的不确定因素机械能数据库表、xml等形式的配置,可以使得不确定因素的调用更加方便、更加精准。

[0181] 在一个例子中,动态加载单元104,可以用于响应对经所述配置化处理的所述设备的触发请求,对该设备对应的所述配置文件进行解析,实现对所述设备的动态加载。该动态加载单元104的具体功能及处理参见步骤S120。

[0182] 例如:运行的时候通过解析配置文件实现动态的加载。

[0183] 例如:在加载设备信息前,不需要运行库不需要知道执行的方法;只有在用户点击触发的时候,通过路径去获取自己要跳转的机型页面。

[0184] 例如:APP通过反射机制,在程序运行时动态调用配置文件,并通过配置文件来访问设备对应页面路径。

[0185] 由此,通过对智能家居系统中设备的配置化处理,通过对配置化处理得到的配置文件进行解析实现动态加载,可以提升加载的响应速度,进而大大提升用户使用的便捷性。

[0186] 可选地,所述配置文件,可以包括:设备对应页面路径和设备对应实体类的至少之一。

[0187] 例如:将设备类型对应的设备对应页面路径、设备对应实体类存储在一个配置文件中。

[0188] 由此,通过对设备对应页面路径和设备对应实体类的配置,可以提升解析的便捷性和可靠性,进而提升用户使用的便捷性。

[0189] 相应地,动态加载单元104,可以包括:路径获取模块1042和实体类加载模块1044。

[0190] 在一个可选例子中,路径获取模块1042,可以用于获取所述配置文件中该设备的设备对应页面路径。该路径获取模块1042的具体功能及处理参见步骤S310。

[0191] 例如:读取对应的设备信息,获取到设备对应的页面路径。

[0192] 可选地,可以结合图8所示本发明的装置中路径获取模块的一实施例的结构示意图,进一步说明路径获取模块1042的具体结构。路径获取模块1042,可以包括:查询子模块10422和获取子模块10424。

[0193] 在一个可选具体例子中,查询子模块10422,可以用于根据所述配置文件中该设备的设备类型,查询所述配置文件。该查询子模块10422的具体功能及处理参见步骤S410。

[0194] 例如:每次加载设备的时候,根据设备的类型,查询配置文件。

[0195] 在一个可选具体例子中,获取子模块10424,可以用于根据查询得到的查询结果,获取与所述设备对应的所述设备对应页面路径。该获取子模块10424的具体功能及处理参见步骤S420。

[0196] 由此,通过查询配置文件的方式实现设备的动态加载,加载效率高,精准性好。

[0197] 在一个可选例子中,实体类加载模块1044,可以用于通过预设的动态运行机制,在

所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备对应实体类。该实体类加载模块1044的具体功能及处理参见步骤S320。

[0198] 例如:通过动态运行机制,在运行过程中加载对应路径下的页面实体类,而无需在代码里面写复杂的逻辑去实现。

[0199] 由此,通过获取设备对应页面路径的方式实现设备对应实体类的动态加载,加载方式简便、且可靠性高。

[0200] 在一个可选实施方式中,还可以包括:功能具体化单元108。

[0201] 在一个可选例子中,功能具体化单元108,可以用于对所述功能进行具体化处理,得到至少一个子功能,并使所述子功能被定制于所述设备。

[0202] 其中,所述子功能,可以包括:能够单独运行的控制过程、和/或能够单独运行的设备界面。

[0203] 例如:将功能具体化,不依赖于实体,每个功能都可以单独运行(例如:该运行,可以包括设备控制、界面UI等),而设备只需要定制自己需要的功能,而不直接具备任何功能。

[0204] 由此,通过对设备功能的具体化处理,可以实现功能定制,进而满足众多的机型需求,使用方便,用户体验好。

[0205] 在一个可选实施方式中,还可以包括:模块化处理单元110。

[0206] 在一个可选例子中,模块化处理单元110,可以用于对所述功能进行具体化处理后,在所述设备对应实体类中对所述子功能进行注册处理,和/或,对所述子功能对应的所述UI进行更改处理,和/或,对所述子功能和/或所述子功能对应的UI进行模块化处理。

[0207] 例如:将功能具体化之后,设备对应实体类注册功能。当界面刷新、下发指令时,通知对应的UI(User Interface,用户界面)进行更改,将功能、UI模块化。

[0208] 例如:可以实现设备功能通用化,并实现功能模块化。

[0209] 由此,通过对功能的注册处理、模块化处理等方式,可以解决团队合作开发困难的问题,解决APP大小急速膨胀的问题,可靠性高,环保性好。

[0210] 在一个可选实施方式中,还可以包括:协议兼容单元112。

[0211] 在一个可选例子中,协议兼容单元112,可以用于对所述功能进行具体化处理后,保持所述子功能和/或所述子功能对应的UI不被更改,增加所述子功能对应设备的标志位、协议配置的至少之一,和/或,填充所述子功能对应设备的具体协议。

[0212] 例如:将功能具体化之后,功能和UI界面不改动,增加厂家设备标志位,增加协议配置,并填充具体协议,APP通过标志位改变采用的具体协议。

[0213] 由此,通过增加标志位和协议配置,可以提升兼容性,进而提升使用便捷性和可靠性。

[0214] 在一个可选实施方式中,所述协议兼容单元112,还可以用于通过所述标志位,对所述具体协议进行更改。

[0215] 例如:可以通过增加协议转换工具,实现协议兼容。

[0216] 由此,通过对具体协议的更改,可以提高兼容的便捷性,减小存储容量,扩大适用范围。

[0217] 在一个可选实施方式中,还可以包括:设备增加单元106。

[0218] 在一个可选例子中,设备增加单元106,可以用于当所述智能家居系统中有待增加

设备时，根据所述待增加设备的设备信息对所述配置文件进行更改，以将所述待增加设备增加至所述智能家居系统。

[0219] 例如：例如：增加设备配置文件，每次添加设备只需要修改配置文件，而不需要去兼容设备，适配设备。

[0220] 由此，通过修改配置文件的方式实现新设备的增加，增加方式简便，控制可靠性高。

[0221] 由于本实施例的装置所实现的处理及功能基本相应于前述图1至图6所示的方法的实施例、原理和实例，故本实施例的描述中未详尽之处，可以参见前述实施例中的相关说明，在此不做赘述。

[0222] 经大量的试验验证，采用本发明的技术方案，通过不断完善智能家居的功能库和功能协议，就可以兼容所有单品，解决团队合作开发困难的问题，解决APP大小急速膨胀的问题。

[0223] 根据本发明的实施例，还提供了对应于智能家居系统的搭建装置的一种智能家居系统。该智能家居系统可以包括：以上所述的智能家居系统的搭建装置。

[0224] 在一个实施方式中，该智能家居系统，可以将复杂的业务逻辑或者不确定的因素进行配置化，目前移动端都是通过动态加载的机制运行的，将复杂的业务逻辑关系和无法确定的因素通过数据库表或者xml的方式存储，代码里面并不会体现这些复杂的逻辑关系，运行的时候通过解析配置文件实现动态的加载。

[0225] 可选地，该智能家居系统，可以通过以下几个方面，对上述配置化的过程进行具体说明。

[0226] 1、机型映射

[0227] 首先通过设备类型对设备进行分类，将设备类型对应的设备对应页面路径、设备对应实体类存储在一个配置文件中。在加载设备信息前，不需要运行库不需要知道执行的方法；只有在用户点击触发的时候，通过路径去获取自己要跳转的机型页面。

[0228] 在一个例子中，可以增加配置文件，用来配置设备信息。

[0229] 例如：增加设备配置文件，每次添加设备只需要修改配置文件，而不需要去兼容设备，适配设备；APP通过反射机制，在程序运行时动态调用配置文件，并通过配置文件来访问设备对应页面路径。

[0230] 2、功能定制

[0231] 将功能具体化，不依赖于实体，每个功能都可以单独运行(例如：该运行，可以包括设备控制、界面UI等)，而设备只需要定制自己需要的功能，而不直接具备任何功能。

[0232] 3、通知响应

[0233] 将功能具体化之后，设备对应实体类注册功能。当界面刷新、下发指令时，通知对应的UI(User Interface，用户界面)进行更改，将功能、UI模块化。

[0234] 在一个例子中，可以实现设备功能通用化，并实现功能模块化。

[0235] 例如：可以针对具体的功能去实现界面的显示和命令查询和下发，而不依赖于具体的机型，机型页面通过注册的方式获取功能。例如：可以采用MVP三层架构的思想，Model层负责数据获取，View层负责界面显示，Presenter层负责控制逻辑的实现，具体设备对应页面路径注册功能模块，动态组装和加载设备需要的功能模块，避免重复的开发。

[0236] 4、兼容性(兼容其它厂家设备)

[0237] 将功能具体化之后,功能和UI界面不改动,增加厂家设备标志位,增加协议配置,并填充具体协议,APP通过标志位改变采用的具体协议。

[0238] 在一个例子中,可以通过增加协议转换工具,实现协议兼容。

[0239] 例如:为了兼容各种不同厂家的协议,将协议层和通讯层进行封装,通讯层负责数据的发送和接受,协议层负责数据的处理,上层获取数据通过具体的接口去调用,将上下层解耦和,通过协议标志位去调用选择不同的协议或者协议版本。

[0240] 在一个可选实施方式中,参见图9和图10所示的例子,该智能家居系统,可以将不确定的因素或者业务逻辑抽离,进行可配置化操作。

[0241] 步骤1、抽离并确定需要配置化的地方。

[0242] 从功能的角度去考虑APP的实现,APP需要配置化的地方,包括:机型不确定性,功能不确定性,协议不确定性。

[0243] 从适用人群选择的角度去考虑APP的实现,APP需要配置化的地方包括:UI不确定性(例如:不确定哪种UI适用于哪种人群),地域不确定性(例如:不确定服务器是否兼容该地区)等。

[0244] 步骤2、配置文件,通过xml(即可扩展标记语言,是一种用于标记电子文件使其具有结构性的标记语言)的形式进行配置。

[0245] 例如:可以将复杂的逻辑抽离,解耦合性,如果实在无法解耦合的地方就是需要写到配置文件的东西。其中,耦合性,也叫耦合度,是对模块间关联程度的度量。

[0246] 可选地,可以通过机型不确定性来说明配置化方案。

[0247] 例如:为了满足各种机型的需求,需要在代码增加各种机型的判断,去识别不同的机型,将这种复杂的逻辑抽离成xml配置文件,格式可以如:

[0248]

```
<?xml version=” 1.0” encoding=” utf-8” >
<DeviceType id = 1 name =” 空调” >
    <Mid>ab492s2sdfa123</Mid>（设备唯一标识）
    <DevicePath>/com/device/activity/DeviceAcActivity.java</DevicePath>
（设备对应页面路径）
    <DeviceEntity>/com/device/activity/DeviceEntity.java</DeviceEntity>
（设备对应实体类）
    <DeviceLogin>DeviceAcLogin</DeviceLogin>（设备登录方法）
</DeviceType>
```

[0249] 在一个例子中,每次加载设备的时候,根据设备的类型,查询配置文件,读取对应的设备信息,获取到设备对应的页面路径;通过动态运行机制,在运行过程中加载对应路径下的页面实体类,而无需在代码里面写复杂的逻辑去实现。

[0250] 可见,APP展示给用户的是具体的功能,但是正是因为不确定性,让APP具有可玩

性。因此，为了让APP更人性化、更完善，将不确定的因素可配置化，在一定程度上降低了代码的复杂性，并且让合作开发变得更简单，更多的开发人员可以专注于功能的具体实现，而APP可以通过配置的方式动态去加载需要的功能，让APP更强大。

[0251] 由此，从智能家居系统的整体概念不难理解，是为了实现家庭设备智能化，而目前家庭设备系统已经很完善，这些设备按照功能基本可以分为多类。例如：空气调节、照明、安防、美食、净水热水、健康生活等。既然功能是已知的，不管智能单品如何变化，只要不断完善智能家居的功能库和功能协议，就可以兼容所有单品；而且单品的开发和功能开发不冲突，可以同时开发多个智能单品，只需要搭建足够稳定的系统。

[0252] 由于本实施例的系统所实现的处理及功能基本相应于前述图6所示的装置的实施例、原理和实例，故本实施例的描述中未详尽之处，可以参见前述实施例中的相关说明，在此不做赘述。

[0253] 经大量的试验验证，采用本发明的技术方案，通过搭建足够稳定的系统，可以同时开发多个智能单品，使得单品的开发和功能开发不冲突，可以进一步提升智能家居系统的响应速度和可伸缩性。

[0254] 根据本发明的实施例，还提供了对应于智能家居系统的一种终端。该终端至少包括：以上所述的所述的智能家居系统。

[0255] 由于本实施例的终端所实现的处理及功能基本相应于前述图9和图10所示的系统的实施例、原理和实例，故本实施例的描述中未详尽之处，可以参见前述实施例中的相关说明，在此不做赘述。

[0256] 经大量的试验验证，采用本发明的技术方案，通过将不确定的因素可配置化，在一定程度上降低了代码的复杂性，并且让合作开发变得更简单，更多的开发人员可以专注于功能的具体实现；而APP可以通过配置的方式动态去加载需要的功能，让APP更强大。

[0257] 综上，本领域技术人员容易理解的是，在不冲突的前提下，上述各有利方式可以自由地组合、叠加。

[0258] 以上所述仅为本发明的实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的权利要求范围之内。

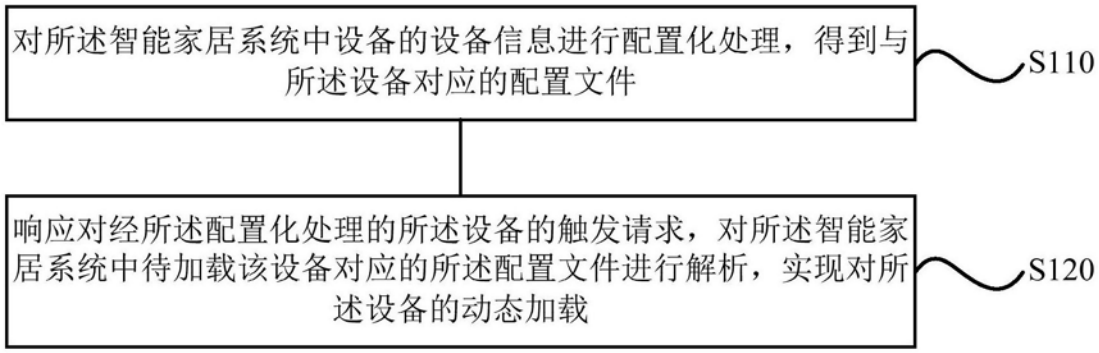

对所述智能家居系统中设备的设备信息进行配置化处理，得到与所述设备对应的配置文件
S110
响应对经所述配置化处理的所述设备的触发请求，对所述智能家居系统中待加载该设备对应的所述配置文件进行解析，实现对所述设备的动态加载
S120

图1

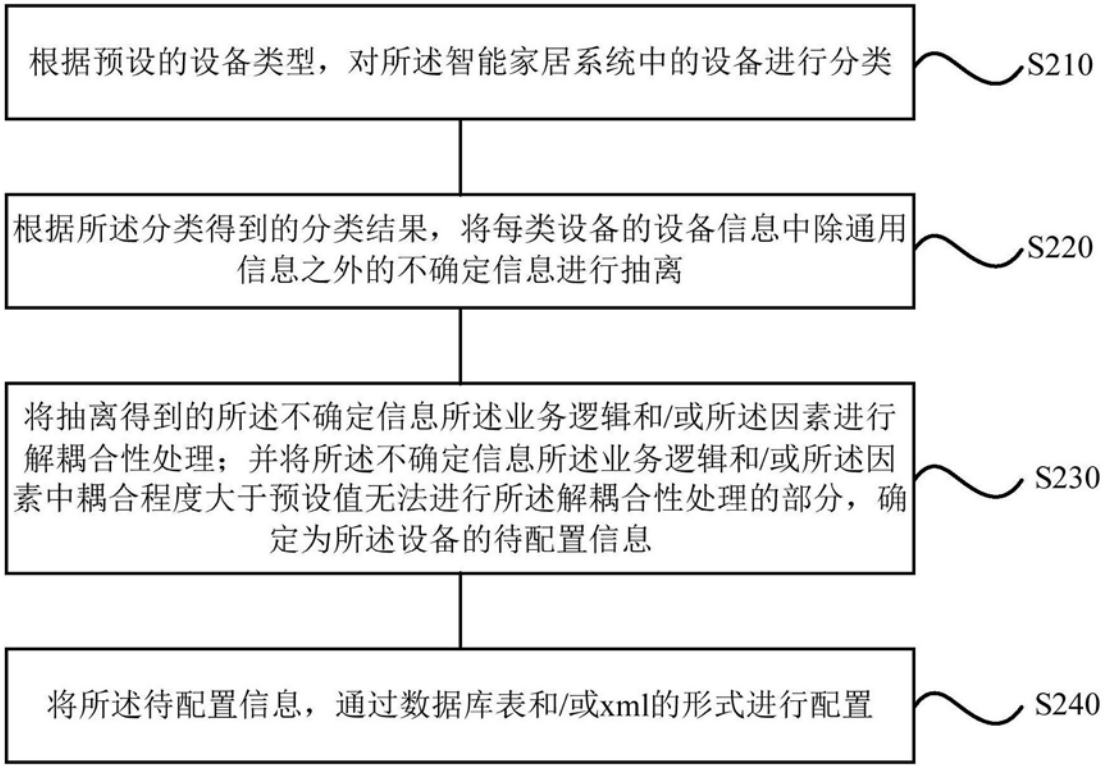

根据预设的设备类型，对所述智能家居系统中的设备进行分类
S210
根据所述分类得到的分类结果，将每类设备的设备信息中除通用信息之外的不确定信息进行抽离
S220
将抽离得到的所述不确定信息所述业务逻辑和/或所述因素进行解耦合性处理；并将所述不确定信息所述业务逻辑和/或所述因素中耦合程度大于预设值无法进行所述解耦合性处理的部分，确定为所述设备的待配置信息
S230
将所述待配置信息，通过数据库表和/或xml的形式进行配置
S240

图2

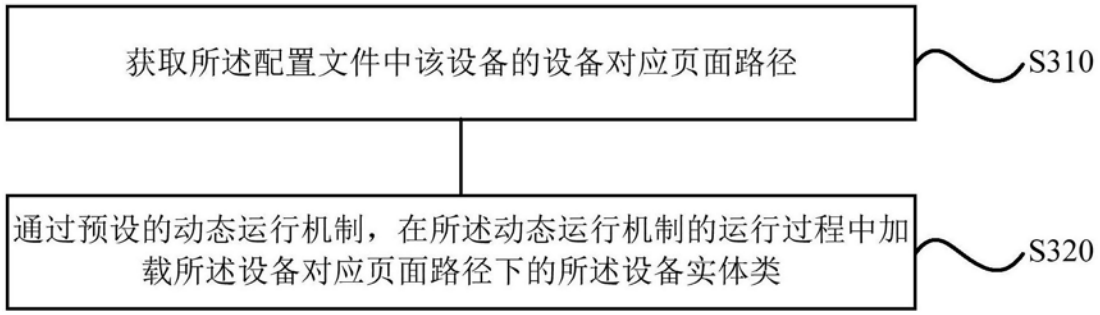

获取所述配置文件中该设备的设备对应页面路径
S310
通过预设的动态运行机制，在所述动态运行机制的运行过程中加载所述设备对应页面路径下的所述设备实体类
S320

图3

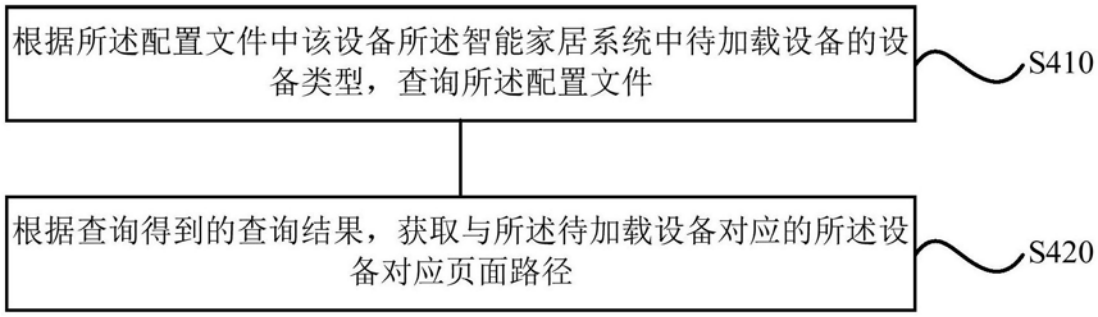

根据所述配置文件中该设备所述智能家居系统中待加载设备的设备类型，查询所述配置文件
S410
根据查询得到的查询结果，获取与所述待加载设备对应的所述设备对应页面路径
S420

图4

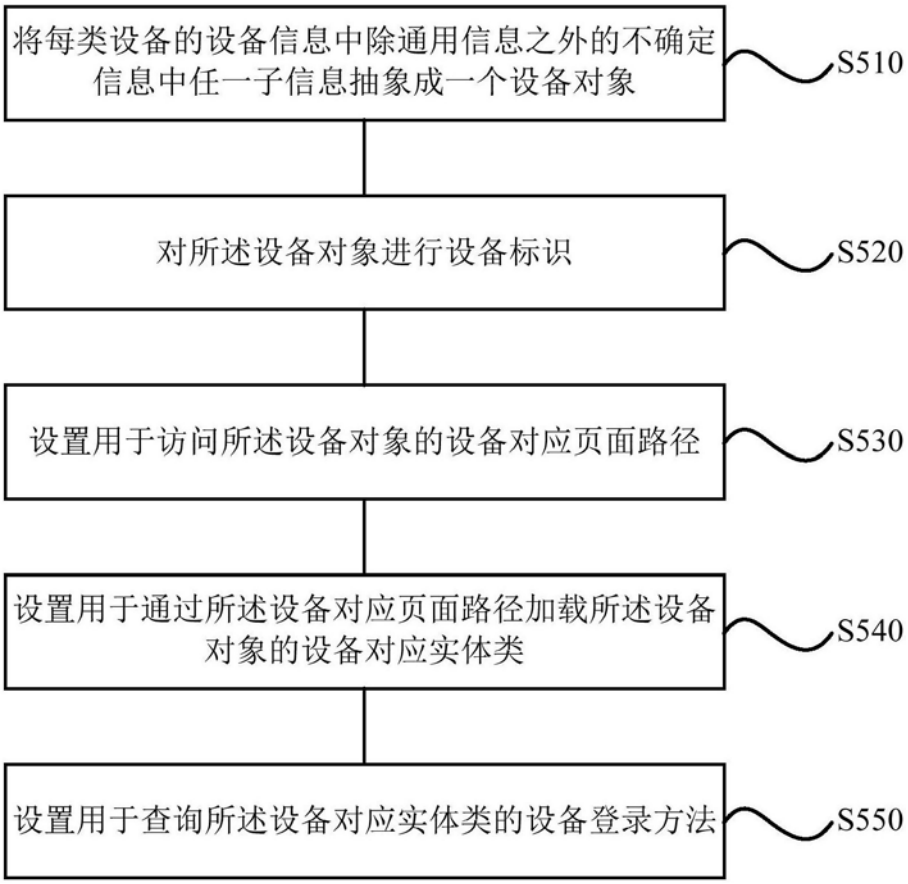

将每类设备的设备信息中除通用信息之外的不确定信息中任一子信息抽象成一个设备对象
S510
对所述设备对象进行设备标识
S520
设置用于访问所述设备对象的设备对应页面路径
S530
设置用于通过所述设备对应页面路径加载所述设备对象的设备对应实体类
S540
设置用于查询所述设备对应实体类的设备登录方法
S550

图5

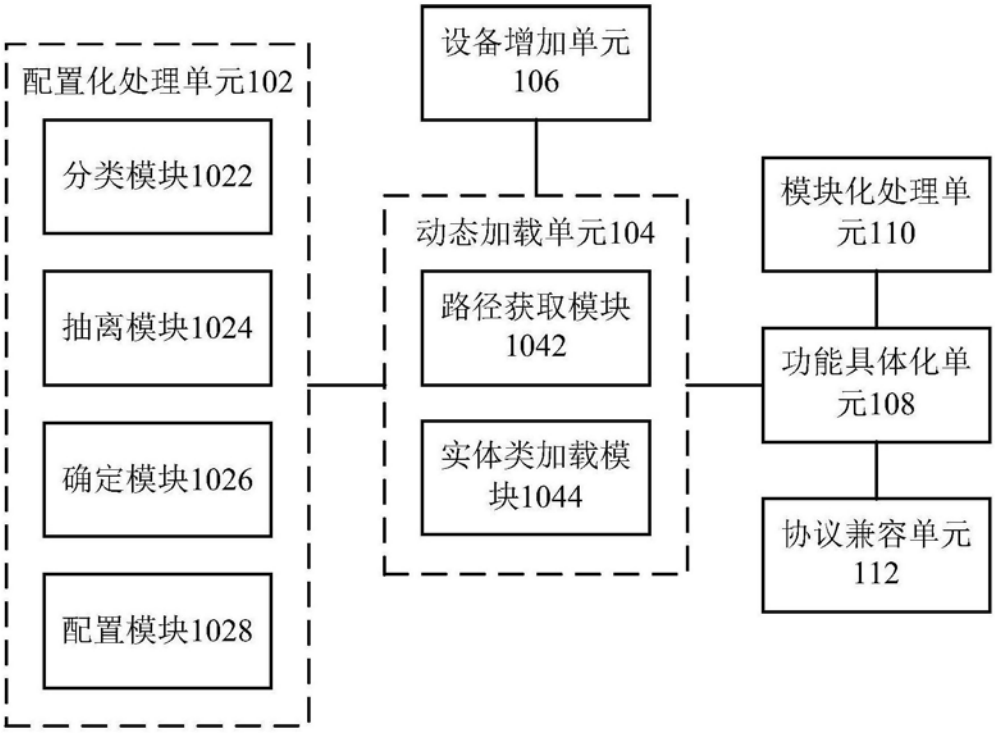

配置化处理单元102
分类模块1022
抽离模块1024
确定模块1026
配置模块1028
设备增加单元106
动态加载单元104
路径获取模块1042
实体类加载模块1044
模块化处理单元110
功能具体化单元108
协议兼容单元112

图6


抽离模块1024
对象抽象子模块10242
标识设置子模块10244
路径设置子模块10246
实体类设置子模块10248
方法设置子模块10250

图7


路径获取模块1042
查询子模块
10422
获取子模块
10424

图8

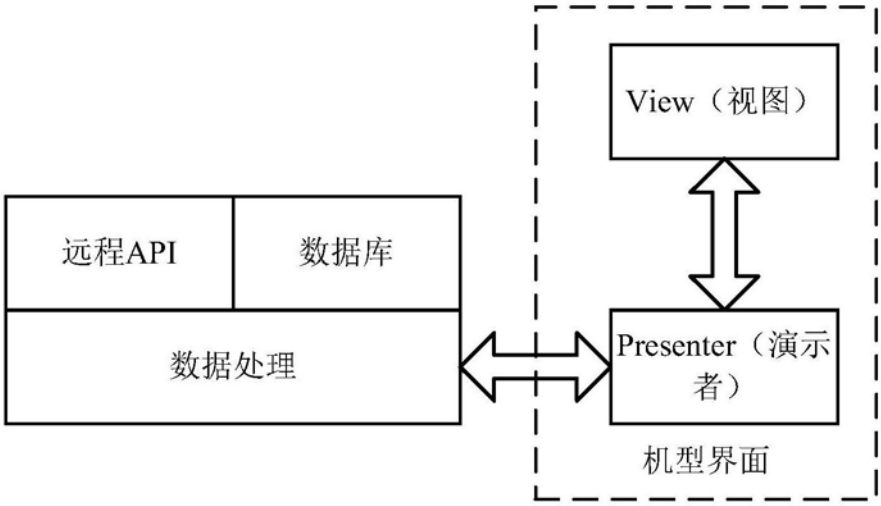

View（视图）
远程API
数据库
数据处理
Presenter（演示者）
机型界面

图9

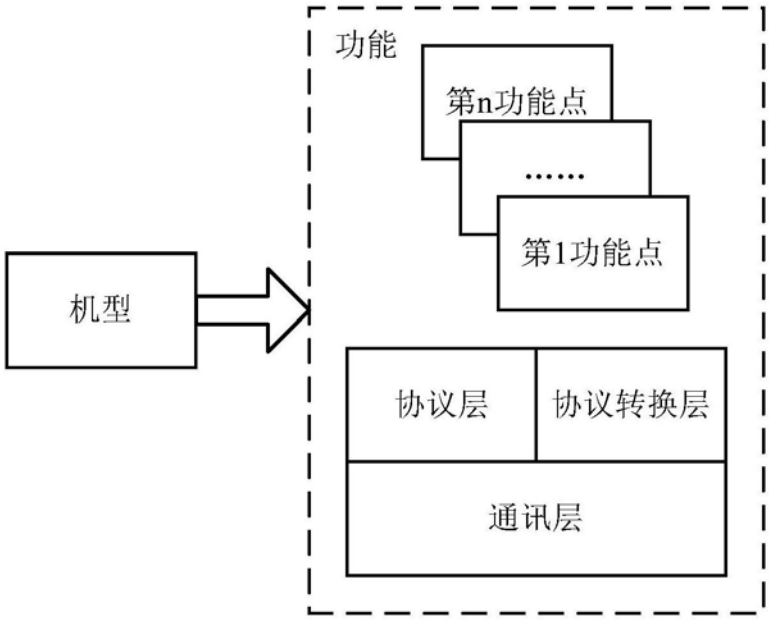


图10