(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112631563 A
(43)申请公布日 2021.04.09

(21)申请号 202011450281.2

(22)申请日 2020.12.09

(71)申请人 北京飞讯数码科技有限公司
地址 100176 北京市大兴区北京经济技术开发区景园北街2号57幢四层A

(72)发明人 梁璇

(74)专利代理机构 北京品源专利代理有限公司 11332
代理人 孟金喆

(51)Int.Cl.
G06F 8/30(2018.01)

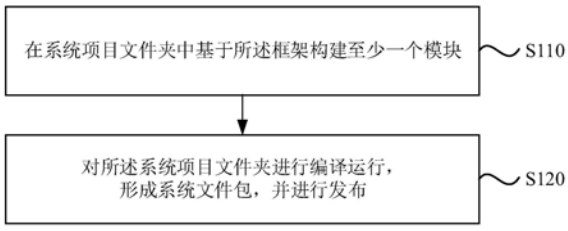
权利要求书1页 说明书11页 附图4页

(54)发明名称

基于框架的系统开发方法、装置、计算机设备及存储介质

(57)摘要

本发明实施例公开了一种基于框架的系统开发方法、装置、计算机设备及存储介质。所述方法包括:在系统项目文件夹中基于所述框架构建至少一个模块,其中,所述框架包括静态文件库和通用组件库,所述通用组件库包括用于至少两个所述模块调用的相同数据,各所述模块与所述通用组件库存在依赖关系,各所述模块依赖的静态文件存储于所述静态文件库中;对所述系统项目文件夹进行编译运行,形成系统文件包,并进行发布。本发明实施例,可以提高系统的开发和更新效率。



CN 112631563 A

1.一种基于框架的系统开发方法,其特征在于,包括:

在系统项目文件夹中基于所述框架构建至少一个模块,其中,所述框架包括静态文件库和通用组件库,所述通用组件库包括用于至少两个所述模块调用的通用组件和/或至少两个所述模块获取的通用数据,各所述模块与所述通用组件库存在依赖关系,各所述模块依赖的静态文件存储于所述静态文件库中;

对所述系统项目文件夹进行编译运行,形成系统文件包,并进行发布。

2.根据权利要求1所述的方法,其特征在于,所述在系统项目文件夹中基于所述框架构建至少一个模块,包括:

在开发环境中,在系统项目文件夹中构建至少一个模块的文件夹;

获取至少一个模块的源代码文件并分别存储于各所述模块匹配的文件夹中,所述源代码文件基于所述框架生成;

将各所述源代码文件依赖的静态文件夹存储于所述静态文件库中;其中,各所述源代码文件配置用于调用所述通用组件库关联的代码文件。

3.根据权利要求1所述的方法,其特征在于,所述静态文件库还包括:通用层叠样式表库;所述通用层叠样式表库中层叠样式表用于各所述模块读取并生成相应样式网页元素。

4.根据权利要求1所述的方法,其特征在于,还包括:

在所述系统项目文件夹中,获取网页跳转路由配置文件;

将各所述模块的路由配置信息存储在所述网页跳转路由配置文件中。

5.根据权利要求1所述的方法,其特征在于,还包括:

更新目标模块;

对更新后的目标模块进行编译运行,更新所述系统文件包;或者

对更新后的系统项目文件夹进行编译运行,更新所述系统文件包。

6.根据权利要求1所述的方法,其特征在于,所述通用组件库,包括下述至少一项:通用工具组件、数据库、通用用户界面资源和鉴权组件。

7.根据权利要求1所述的方法,其特征在于,所述系统为音视频调度传输系统;所述模块包括下述至少一项:视频会议模块、视频监控模块、视频通话模块和视频调度模块。

8.一种基于框架的系统开发装置,其特征在于,包括:

基于架构开发模块,用于在系统项目文件夹中基于所述框架构建至少一个模块,其中,所述框架包括静态文件库和通用组件库,所述通用组件库包括用于至少两个所述模块调用的通用组件和/或至少两个所述模块获取的通用数据,各所述模块与所述通用组件库存在依赖关系,各所述模块依赖的静态文件存储于所述静态文件库中;

打包发布模块,用于对所述系统项目文件夹进行编译运行,形成系统文件包,并进行发布。

9.一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1-7中任一所述的基于框架的系统开发方法。

10.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1-7中任一所述的基于框架的系统开发方法。

基于框架的系统开发方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明实施例涉及软件开发领域，尤其涉及一种基于框架的系统开发方法、装置、计算机设备及存储介质。

背景技术

[0002] 随着平台业务增加，业务逻辑的重复代码增多，功能归属混乱，业务冗余功能耦合性逐渐变强，前后端资源加载逐渐遇到性能瓶颈，开发、测试、部署和维护等愈发困难。

[0003] 在产品和项目开发的过程中，用户需求涉及的功能和业务既有不同点又有一致性。为了节约开发成本，提高开发效率，采用模块化的思路是分属同一功能或业务的代码进行隔离，按照项目功能需求划分成不同类型的模块业务框架，模块独立运行，采用横向分块(业务框架层)架构定位，实现高耦合、低内聚、无重用和易部署等。

[0004] 但上述方法，横向模块的代码重复性高，根据需求变更时，每个模块里的静态文件或重复代码需要替换或修改。

发明内容

[0005] 本发明实施例提供了一种基于框架的系统开发方法、装置、计算机设备及存储介质，可以提高系统的开发和更新效率。

[0006] 第一方面，本发明实施例提供了一种基于框架的系统开发方法，包括：

[0007] 在系统项目文件夹中基于所述框架构建至少一个模块，其中，所述框架包括静态文件库和通用组件库，所述通用组件库包括用于至少两个所述模块调用的相同数据，各所述模块与所述通用组件库存在依赖关系，各所述模块依赖的静态文件存储于所述静态文件库中；

[0008] 对所述系统项目文件夹进行编译运行，形成系统文件包，并进行发布。

[0009] 第二方面，本发明实施例提供了一种基于框架的系统开发装置，包括：

[0010] 基于架构开发模块，用于在系统项目文件夹中基于所述框架构建至少一个模块，其中，所述框架包括静态文件库和通用组件库，所述通用组件库包括用于至少两个所述模块调用的相同数据，各所述模块与所述通用组件库存在依赖关系，各所述模块依赖的静态文件存储于所述静态文件库中；

[0011] 打包发布模块，用于对所述系统项目文件夹进行编译运行，形成系统文件包，并进行发布。

[0012] 第三方面，本发明实施例还提供了一种计算机设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序所述处理器执行所述程序时实现如本发明实施例中任一所述的基于框架的系统开发方法。

[0013] 第四方面，本发明实施例还提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现如本发明实施例中任一所述的基于框架的系统开发方法。

[0014] 本发明实施例通过预先配置包括通用组件库和静态文件库的框架，并基于该框架构建模块，将多个模块中相同的功能和数据进行提取并配置于通用组件库中，以便各模块进行调用，同时，将各模块依赖的静态资源配置于同一静态文件库中，便于集中管理，解决了现有技术中横向开发模块，模块之间代码重复率高，更新和修改成本高的问题，可以集中管理和更新模块之间的相同数据，减少代码重复性开发，提高系统开发效率，以及提高系统更新效率。

附图说明

[0015] 图1是本发明实施例一中的一种基于框架的系统开发方法的流程图；

[0016] 图2a是本发明实施例二中的一种基于框架的系统开发方法的流程图；

[0017] 图2b是本发明实施例二中的一种静态文件库的示意图；

[0018] 图2c是本发明实施例二中的一种通用组件库的示意图；

[0019] 图3是本发明实施例三中的一种基于框架的系统开发装置的结构示意图；

[0020] 图4是本发明实施例四中的一种计算机设备的结构示意图。

具体实施方式

[0021] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅仅用于解释本发明，而非对本发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与本发明相关的部分而非全部结构。

[0022] 实施例一

[0023] 图1为本发明实施例一中的一种基于框架的系统开发方法的流程图，本实施例可适用于系统开发的情况，该方法可以由本发明实施例提供的基于框架的系统开发装置来执行，该装置可采用软件和/或硬件的方式实现，并一般可集成计算机设备中。如图1所示，本实施例的方法具体包括：

[0024] S110，在系统项目文件夹中基于所述框架构建至少一个模块，其中，所述框架包括静态文件库和通用组件库，所述通用组件库包括用于至少两个所述模块调用的相同数据，各所述模块与所述通用组件库存在依赖关系，各所述模块依赖的静态文件存储于所述静态文件库中。

[0025] 系统项目文件夹用于承载构成系统的多个模块关联的数据。在系统开发时，编写的文件配置于该系统项目文件夹中。框架用于为各模块提供通用数据。框架为预先自定义的代码文件以及关联文件。其中，框架可以是各模块提取出的相同功能的数据形成的类。具体的，框架可以是根据需要自定义的类。具体的，框架包括静态文件库和通用组件库。静态文件库用于承载各模块依赖的静态文件，通用组件库用于承载各模块依赖的通用组件以及通用数据。通用组件库包括用于至少两个模块调用的相同通用组件和/或至少两个模块获取的相同通用数据。其中，通用组件库中各通用组件依赖的静态文件存储于该静态文件库中。通用组件用于模块在运行时调用，通用数据和静态文件用于在模块运行时读取。

[0026] 可以提取各模块中相同功能提取出形成通用组件，并在模块运行时调用；同时可以提取各模块中相同数据提取出形成通用数据，并在模块运行时获取，存储于通用组件库中，便于统一管理。而且，在需要对各模块的通用功能进行增加、删除和修改等时，仅需要对

通用组件库中通用组件进行增加、删除和修改等。同时，提取各模块依赖的静态文件存储于静态文件库中，便于统一管理。在需要对各模块的静态文件进行增加、删除和修改等时，仅需要在静态文件库中进行查询对应的静态文件进行增加、删除和修改等。

[0027] S120，对所述系统项目文件夹进行编译运行，形成系统文件包，并进行发布。

[0028] 对系统项目文件夹进行编译运行，形成系统文件包，实现对系统项目文件夹里的数据进行打包，形成系统文件包。系统文件包用于提供为任意计算机设备进行下载，部署到任意计算机设备中运行，实现系统的功能。系统文件包发布用于向各用户端公开下载链接，以使用户端下载系统文件包。

[0029] 本发明实施例通过预先配置包括通用组件库和静态文件库的框架，并基于该框架构建模块，将多个模块中相同的功能和数据进行提取并配置于通用组件库中，以便各模块进行调用，同时，将各模块依赖的静态资源配置于同一静态文件库中，便于集中管理，解决了现有技术中横向开发模块，模块之间代码重复率高，更新和修改成本高的问题，可以集中管理和更新模块之间的相同数据，减少代码重复性开发，提高系统开发效率，以及提高系统更新效率。

[0030] 实施例二

[0031] 图2a为本发明实施例二中的一种基于框架的系统开发方法的流程图，本实施例以上述实施例为基础进行具体化。本实施例的方法具体包括：

[0032] S210，在开发环境中，在系统项目文件夹中构建至少一个模块的文件夹。

[0033] 开发环境可以是指进行模块和系统开发的环境。模块的文件夹用于承载模块开发形成的源代码文件等。

[0034] 本发明实施例未详尽的描述可以参考前述实施例。

[0035] S220，获取至少一个模块的源代码文件并分别存储于各所述模块匹配的文件夹中，其中，所述源代码文件基于所述框架生成，所述框架包括静态文件库和通用组件库，所述通用组件库包括用于至少两个所述模块调用的通用组件和/或至少两个所述模块获取的通用数据，各所述模块与所述通用组件库存在依赖关系，各所述模块依赖的静态文件存储于所述静态文件库中。

[0036] 模块匹配的文件夹可以是指存储该模块开发形成的各项文件的文件夹。示例性的，可以针对模块确定标识信息，并将该标识信息确定为该模块匹配的文件夹的标识信息，例如，标识信息为文件夹的名称，此时，模块可以是未开发完成的模块。可以基于代码开发软件对模块进行开发，并生成源代码文件存储于该模块匹配的文件夹内。或者，直接将已经预先开发完成的源代码文件移动到该模块匹配的文件夹内。模块匹配的文件夹相互独立，不同模块匹配的文件夹不同，例如，存储目录不同。模块用于实现特定功能，实现的功能内容可以根据实际情况进行设定。示例性的，模块可以用于实现图像采集、图像处理、音频采集、音频处理和数据传输等功能。不同模块的功能可以相互独立，也可以相互依赖。例如，一个模块的输出数据是另外一个模块的输入数据，例如，图像采集模块的输出数据作为图像处理模块的输入数据。又如，图像处理模块的功能和音频处理模块的功能相互独立。源代码文件基于框架生成，可以是指源代码文件依赖通用组件库，例如，源代码文件中包括调用通用组件库中通用组件的代码。

[0037] 示例性的，可以将鉴权、通用工具类(utils)和数据库等重复的代码提取出来合并

成为通用组件库中通用组件，通用组件库最重要的就是重用(复用)，位于框架最底层，其他功能都依赖于通用组件库，供所有的模块使用，独立性强。

[0038] 各模块与通用组件库存在依赖关系，可以是指，模块调用通用组件库中的通用组件和/或模块获取通用组件库中的通用数据。示例性的，各模块需要执行鉴权任务，各模块均可以调用通用组件库中的鉴权组件。各模块需要采用统一的显示风格，各模块可以从通用组件库中获取通用数据，如相同的用户界面资源，从而，各模块的交互页面在用户界面上的显示风格相同，例如颜色相同，或者部分控件颜色和样式相同等。

[0039] 其中，用户可以通过模块匹配的页面中的控制输入项输入控制指令。模块匹配的页面可以根据模块依赖的静态文件生成。模块依赖的静态文件用于生成模块匹配的页面，以运行和控制该模块。

[0040] 可选的，所述通用组件库，包括下述至少一项：通用工具组件、数据库、通用用户界面资源和鉴权组件。

[0041] 通用工具组件可以是指多个模块均使用的工具组件，即常用的工具组件，例如，通用工具组件可以包括帮助组件等。数据库用于存储通用组件库中各组件所需的数据。通用用户界面资源可以是指对软件的人机交互、操作逻辑和界面美观的资源，例如，图片资源和文本资源等。鉴权组件用于对用户身份进行确认，即验证用户是否拥有访问系统的权利。

[0042] 通过配置通用组件库，可以灵活配置模块中相同重复的功能，并进行同一管理，实现通用功能的快速维护。

[0043] S230，将各所述源代码文件依赖的静态文件夹存储于所述静态文件库中；其中，各所述源代码文件配置用于调用所述通用组件库关联的代码文件。

[0044] 静态文件可以是指内容固定的文件，例如被模块使用，且不包括没有代码的文件。例如，静态文件可以包括下述至少一项：js库、前端样式文件和图片等。将各模块以及通用组件依赖的静态文件存储于公共文件夹(即静态文件库)中，使得当静态文件被修改或替换时，能够立即体现在页面上，不需要重新构建项目和项目中的模块，集中管理样式，以及方便风格的统一。

[0045] 可选的，所述静态文件库还包括：通用层叠样式表库；所述通用层叠样式表库中层叠样式表用于各所述模块读取并生成相应样式网页元素。

[0046] 层叠样式表(Cascading Style Sheets，CSS)用于描述网页元素的样式和布局，例如标题元素的字体、颜色、尺寸和标题背景图像等。通用层叠样式表用于构建统一风格网页元素。多个模块使用同一层叠样式表，可以生成相同样式网页元素。通用层叠样式表库存储多个模块所使用的相同的网页元素样式的层叠样式表。模块可以在运行时读取层叠样式表，进行网页渲染和布局，形成模块匹配的网页。

[0047] 同时，采用Less的CSS预处理语言对CSS文件进行开发，在修改层叠样式表时，可以直接采用Less语言在通用层叠样式表库中对层叠样式表文件修改，可以将读取同一层叠样式表的模块中对应的网页元素的样式进行统一修改。

[0048] 通过将各模块中相同的层叠样式表进行提取，并存储于通用层叠样式表库中进行统一管理，可以快速对读取同一层叠样式表的模块中对应的样式进行修改，提高网页元素样式的修改效率

[0049] S240，对所述系统项目文件夹进行编译运行，形成系统文件包，并进行发布。

[0050] 可选的，所述基于框架的系统开发方法，还包括：在系统项目文件夹中，获取网页跳转路由配置文件；将各所述模块的路由配置信息存储在所述网页跳转路由配置文件中。

[0051] 网页跳转路由配置文件用于统一配置各模块的网页页面对应的跳转路径。模块的路由配置信息用于在读取时跳转到模块的网页页面。通过将各模块的路由配置信息存储在网页跳转路由配置文件中，可以对各模块的路由信息进行统一管理，加快路由信息的配置效率。

[0052] 可选的，所述基于框架的系统开发方法，还包括：更新目标模块；对更新后的目标模块进行编译运行，更新所述系统文件包；或者对更新后的系统项目文件夹进行编译运行，更新所述系统文件包。

[0053] 每个模块都配置独立的编译运行代码，以及在系统项目文件夹中配置整个系统的编译运行代码，可以根据需要选择某个模块或者全部模块(即整个系统)进行编译运行，实现灵活选择部分系统或整个系统进行打包，可以仅通过对更新的部分进行打包，减少仅能通过整个系统进行打包，才能更新系统文件包，提高系统文件包的更新效率。

[0054] 在一个具体的应用场景中，可选的，所述系统为音视频调度传输系统；所述模块包括下述至少一项：视频会议模块、视频监控模块、视频通话模块和视频调度模块。

[0055] 可以基于本发明实施例中基于框架的系统开发方法，分别编写视频会议模块、视频监控模块、视频通话模块和视频调度模块等的代码文件，并各模块的代码文件中调用通用组件库中的组件，可选择读取静态文件库中通用层叠样式表库，同时关联的静态文件存储于该静态文件库，路由配置信息在网页跳转路由配置文件中进行设置。在开发完成后，对系统项目文件夹进行编译和运行，形成系统文件包。在打包完成后，需要对模块进行修改时，可以仅对模块进行编译和运行，更新系统文件包。将系统文件包安装在计算机设备中，建立音视频调度传输系统，并实现不同型号设备的运维。

[0056] 在一个具体的例子中：

[0057] 构建项目后的框架结构下所示，

[0058] 框架

[0059] ├———lib文件夹

[0060] ├———静态文件夹

[0061] ├———模板文件夹

[0062] ├———通用组件库jar包

[0063] ├———其他文件或文件夹

[0064] 其中，框架文件夹中包括lib文件夹、静态文件夹(静态文件库)、模板文件夹和通用组件库jar包等。lib文件夹用于存储函数调用的数据。模板文件夹(template)用于存储模板文件。框架中静态文件库如图2b所示，由于静态文件库构建在jar包外部，使得当静态文件被修改或替换时，能够立即体现在页面上。其中，层叠样式表为各模块使用的单独的层叠样式表。而通用层叠样式表，为各模块之间使用的相同的层叠样式表，并提供Less语言的修改接口，以便快速调整层叠样式表。

[0065] 在框架中，通用组件库jar包会判断项目部署环境(开发环境和正式环境)和运行的模块等，读取并加载相关的文件。因此每个模块引入通用组件库后，可以判断当前项目部署环境，检测模块标识，以及读取并加载相关的文件，能保证引入通用组件库的模块可以正

确加载相关的文件,以使模块可以实现通用组件库提供的功能。通用组件库的组成结构如图2c所示。

[0066] 模块化开发后,通用组件库结构如下所示:

[0067] 通用组件库

[0068] ├———excel文件夹

[0069] ├———lib文件夹

[0070] ├———log文件夹

[0071] ├———xml文件夹

[0072] ├———其他文件或文件夹

[0073] 其中,通用组件库文件夹中包括excel文件夹、lib文件夹、log文件夹和xml文件夹等。log文件夹用于存储日志信息,xml文件夹用于存储路由配置信息,即网页跳转路由配置文件。excel文件夹用于存储excel数据。

[0074] 示例性的,business模块文件夹结构如下所示:

[0075] business模块文件夹

[0076] ├———lib文件夹

[0077] ├———business模块文件夹jar包

[0078] ├———其他文件或文件夹

[0079] business模块的lib文件夹结构如下所示:

[0080] business模块文件夹的lib文件夹

[0081] ├———通用组件库jar包

[0082] ├———其他文件或文件夹

[0083] business模块的lib文件夹中包含通用组件库jar包,作为组件引用。当需要修改某模块里的功能时,只需要重启该模块的jar,不影响其他模块的功能使用。

[0084] 现有技术中,构建脚本只能构建整个项目。采用本发明实施例提供的方法进行开发后,在每个模块匹配的文件夹中都配置构建脚本文件,可以遍历各构建脚本文件,一键构建项目中所有模块及组件,具体的,可以对现有的项目构建脚本进行修改,添加构建新语句,例如,添加构建新语句的内容为遍历所有子模块的构建脚本文件,构建编译依赖以及打包jar的文件,也可以指定某个模块匹配的文件夹中构建脚本文件,按模块进行构建。

[0085] 示例性的,现有技术的系统项目文件夹结构为:

[0086] 系统项目文件夹

[0087] ├———资源设置

[0088] ├———任务

[0089] │├———项目构建

[0090] │├———项目安装

[0091] │├———文本文档

[0092] │├———帮助

[0093] │├———其他

[0094] │└———鉴权

[0095] ├———运行配置

[0096] 其中,项目构建文件仅能构建整个项目。

[0097] 本发明中系统项目文件夹结构为:

[0098] 系统项目文件夹

[0099] ├———资源设置

[0100] ├———任务

[0101] │├———项目构建

[0102] │├———项目安装

[0103] │├———文本文档

[0104] │├———帮助

[0105] │├———其他

[0106] │└———鉴权

[0107] ├———运行配置

[0108] ├———第一模块

[0109] │├———资源设置

[0110] │├———任务

[0111] ||├———模块构建

[0112] ||├———文本文档

[0113] ||├———帮助

[0114] ||├———其他

[0115] ||└———鉴权

[0116] │├———运行配置

[0117] ├———第二模块

[0118] │├———资源设置

[0119] │├———任务

[0120] ||├———模块构建

[0121] ||├———文本文档

[0122] ||├———帮助

[0123] ||├———其他

[0124] ||└———鉴权

[0125] │├———运行配置

[0126] ……

[0127] 其中,系统项目文件夹包括构建文件,各模块也包括构建文件,项目构建文件可以构建整个项目。各模块的模块构建文件可以构建单个模块。

[0128] 综上所述,模块化开发后,分离了原来冗余和耦合度高的代码,划分成模块,并把所有模块依赖的基础功能合并为通用组件库,并将静态文件构建为一个外置文件夹静态文件库,使修改和替换不需重启服务,方便了项目迁移、构建及模块的可插拔性。

[0129] 本发明实施例通过在开发环境下的系统项目文件夹中分贝构建模块的文件夹,并基于框架编写各模块的源代码文件,以使源代码文件中调用通用组件库中各通用组件,并将静态文件提取出存储于静态文件库中,实现在模块独立开发,并依赖于同一基础组件,以

及将静态文件存储于同一文件库中，进行统一管理，减少重复性代码的开发，并将静态文件单独统一管理，便于各模块的修改，提高系统的可插拔性。

[0130] 实施例三

[0131] 图3为本发明实施例三中的一种基于框架的系统开发装置的示意图。实施例三是实现本发明上述实施例提供的基于框架的系统开发方法的相应装置，该装置可采用软件和/或硬件的方式实现，并一般可集成计算机设备中。

[0132] 相应的，本实施例的装置可以包括：

[0133] 基于架构开发模块310，用于在系统项目文件夹中基于所述框架构建至少一个模块，其中，所述框架包括静态文件库和通用组件库，所述通用组件库包括用于至少两个所述模块调用的相同数据，各所述模块与所述通用组件库存在依赖关系，各所述模块依赖的静态文件存储于所述静态文件库中；

[0134] 打包发布模块320，用于对所述系统项目文件夹进行编译运行，形成系统文件包，并进行发布。

[0135] 本发明实施例通过预先配置包括通用组件库和静态文件库的框架，并基于该框架构建模块，将多个模块中相同的功能和数据进行提取并配置于通用组件库中，以便各模块进行调用，同时，将各模块依赖的静态资源配置于同一静态文件库中，便于集中管理，解决了现有技术中横向开发模块，模块之间代码重复率高，更新和修改成本高的问题，可以集中管理和更新模块之间的相同数据，减少代码重复性开发，提高系统开发效率，以及提高系统更新效率。

[0136] 进一步的，所述基于架构开发模块310，具体用于：在开发环境中，在系统项目文件夹中构建至少一个模块的文件夹；获取至少一个模块的源代码文件并分别存储于各所述模块匹配的文件夹中，所述源代码文件基于所述框架生成；将各所述源代码文件依赖的静态文件夹存储于所述静态文件库中；其中，各所述源代码文件配置用于调用所述通用组件库关联的代码文件。

[0137] 进一步的，所述静态文件库还包括：通用层叠样式表库；所述通用层叠样式表库中层叠样式表用于各所述模块读取并生成相应样式网页元素。

[0138] 进一步的，所述基于框架的系统开发装置，还包括：路由配置模块，用于在系统项目文件夹中，获取网页跳转路由配置文件；将各所述模块的路由配置信息存储在所述网页跳转路由配置文件中。

[0139] 进一步的，所述基于框架的系统开发装置，还包括：系统更新模块，用于更新目标模块；对更新后的目标模块进行编译运行，更新所述系统文件包；或者对更新后的系统项目文件夹进行编译运行，更新所述系统文件包。

[0140] 进一步的，所述通用组件库，包括下述至少一项：通用工具组件、数据库、通用用户界面资源和鉴权组件。

[0141] 进一步的，所述系统为音视频调度传输系统；所述模块包括下述至少一项：视频会议模块、视频监控模块、视频通话模块和视频调度模块。

[0142] 上述基于框架的系统开发装置可执行本发明实施例所提供的基于框架的系统开发方法，具备执行的基于框架的系统开发方法相应的功能模块和有益效果。

[0143] 实施例四

[0144] 图4为本发明实施例四提供的一种计算机设备的结构示意图。图4示出了适于用来实现本发明实施方式的示例性计算机设备12的框图。图4显示的计算机设备12仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0145] 如图4所示,计算机设备12以通用计算设备的形式表现。计算机设备12的组件可以包括但不限于:一个或者多个处理器或者处理单元16,系统存储器28,连接不同系统组件(包括系统存储器28和处理单元16)的总线18。计算机设备12可以是挂接在高速工业控制总线上的设备。

[0146] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(Industry Standard Architecture,ISA)总线,微通道体系结构(Micro Channel Architecture,MCA)总线,增强型ISA总线、视频电子标准协会(Video Electronics Standards Association,VESA)局域总线以及外围组件互连(Peripheral Component Interconnect,PCI)总线。

[0147] 计算机设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被计算机设备12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0148] 系统存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。计算机设备12可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图4未显示,通常称为“硬盘驱动器”)。尽管图4中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如紧凑磁盘只读存储器(Compact Disc Read-Only Memory,CD-ROM),数字视盘(Digital Video Disc-Read Only Memory,DVD-ROM)或者其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。系统存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0149] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如系统存储器28中,这样的程序模块42包括——但不限于——操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0150] 计算机设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该计算机设备12交互的设备通信,和/或与使得该计算机设备12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(Input/Output,I/O)接口22进行。并且,计算机设备12还可以通过网络适配器20与一个或者多个网络(例如局域网(Local Area Network,LAN),广域网(Wide Area Network,WAN)通信。如图所示,网络适配器20通过总线18与计算机设备12的其它模块通信。应当明白,尽管图4中未示出,可以结合计算机设备12使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、(Redundant Arrays of Inexpensive Disks,RAID)系统、磁带驱动器以及数据备份存储系统等。

[0151] 处理单元16通过运行存储在系统存储器28中的程序，从而执行各种功能应用以及数据处理，例如实现本发明任意实施例所提供的一种基于框架的系统开发方法。

[0152] 实施例五

[0153] 本发明实施例五提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现如本申请所有发明实施例提供的基于框架的系统开发方法：

[0154] 也即，该程序被处理器执行时实现：在系统项目文件夹中基于所述框架构建至少一个模块，其中，所述框架包括静态文件库和通用组件库，所述通用组件库包括用于至少两个所述模块调用的相同数据，各所述模块与所述通用组件库存在依赖关系，各所述模块依赖的静态文件存储于所述静态文件库中；对所述系统项目文件夹进行编译运行，形成系统文件包，并进行发布。

[0155] 本发明实施例的计算机存储介质，可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、RAM、只读存储器(Read Only Memory，ROM)、可擦式可编程只读存储器(Erasable Programmable Read Only Memory，EPROM)、闪存、光纤、便携式CD-ROM、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0156] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括——但不限于——电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0157] 计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括——但不限于——无线、电线、光缆、无线电频率(RadioFrequency，RF)等等，或者上述的任意合适的组合。

[0158] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括LAN或WAN——连接到用户计算机，或者，可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0159] 注意，上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解，本发明不限于这里所述的特定实施例，对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此，虽然通过以上实施例对本发明进行了较为详细的说明，但是本发明不仅仅限于以上实施例，在不脱离本发明构思的情况下，还

可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

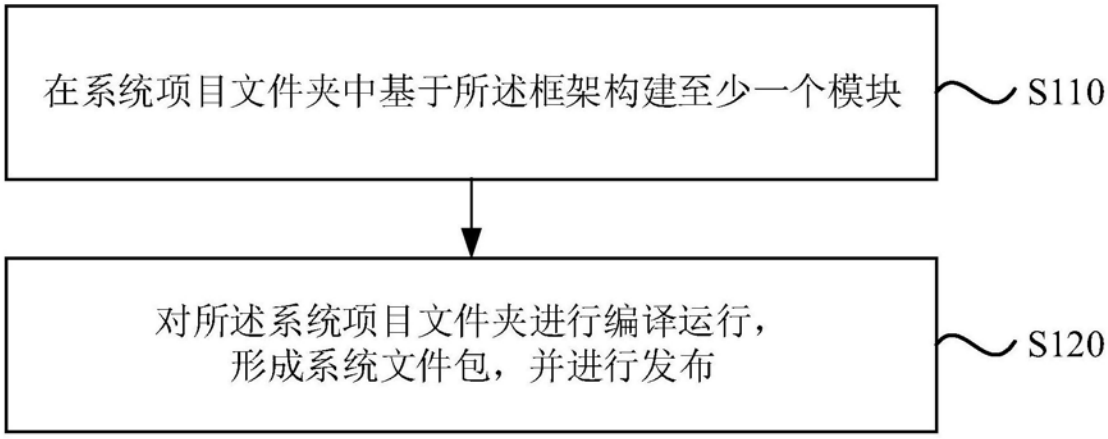

在系统项目文件夹中基于所述框架构建至少一个模块
S110
对所述系统项目文件夹进行编译运行，
形成系统文件包，并进行发布
S120

图1

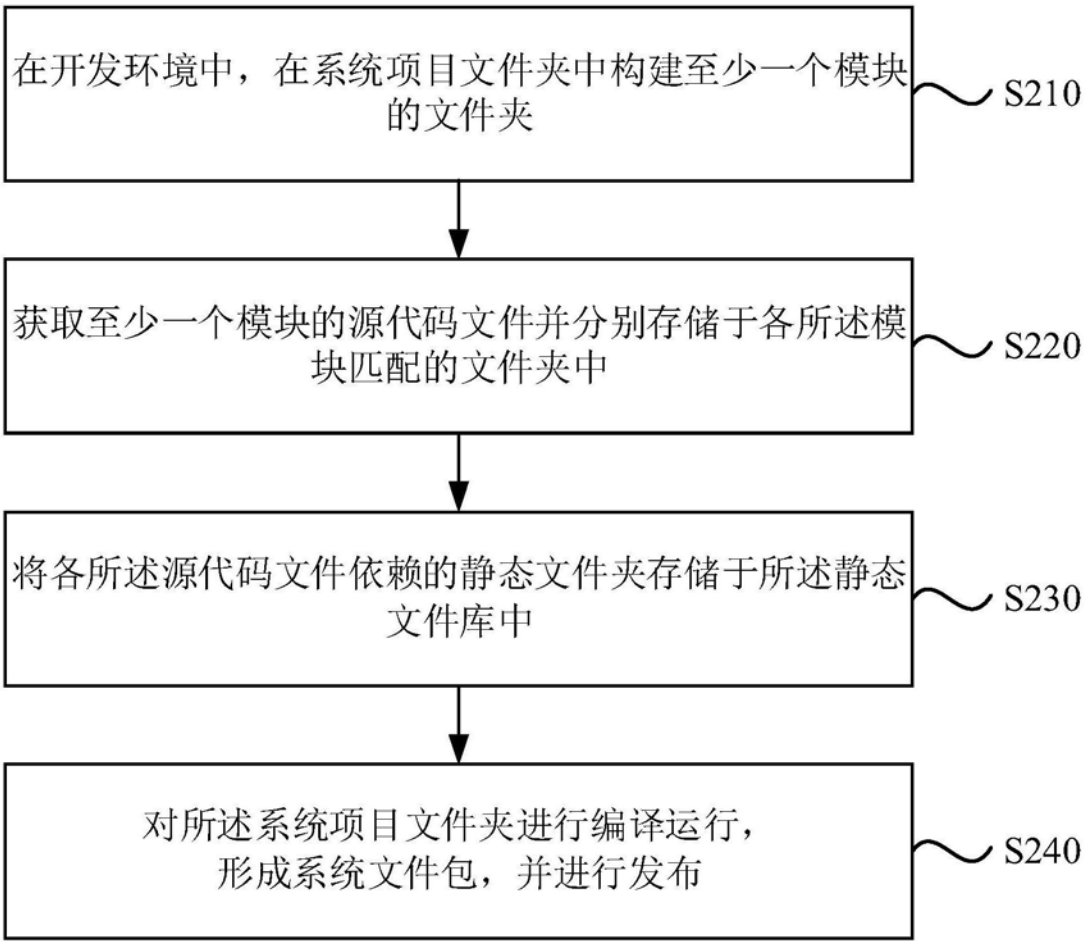

在开发环境中，在系统项目文件夹中构建至少一个模块
的文件夹
S210
获取至少一个模块的源代码文件并分别存储于各所述模
块匹配的文件夹中
S220
将各所述源代码文件依赖的静态文件夹存储于所述静态
文件库中
S230
对所述系统项目文件夹进行编译运行，
形成系统文件包，并进行发布
S240

图2a

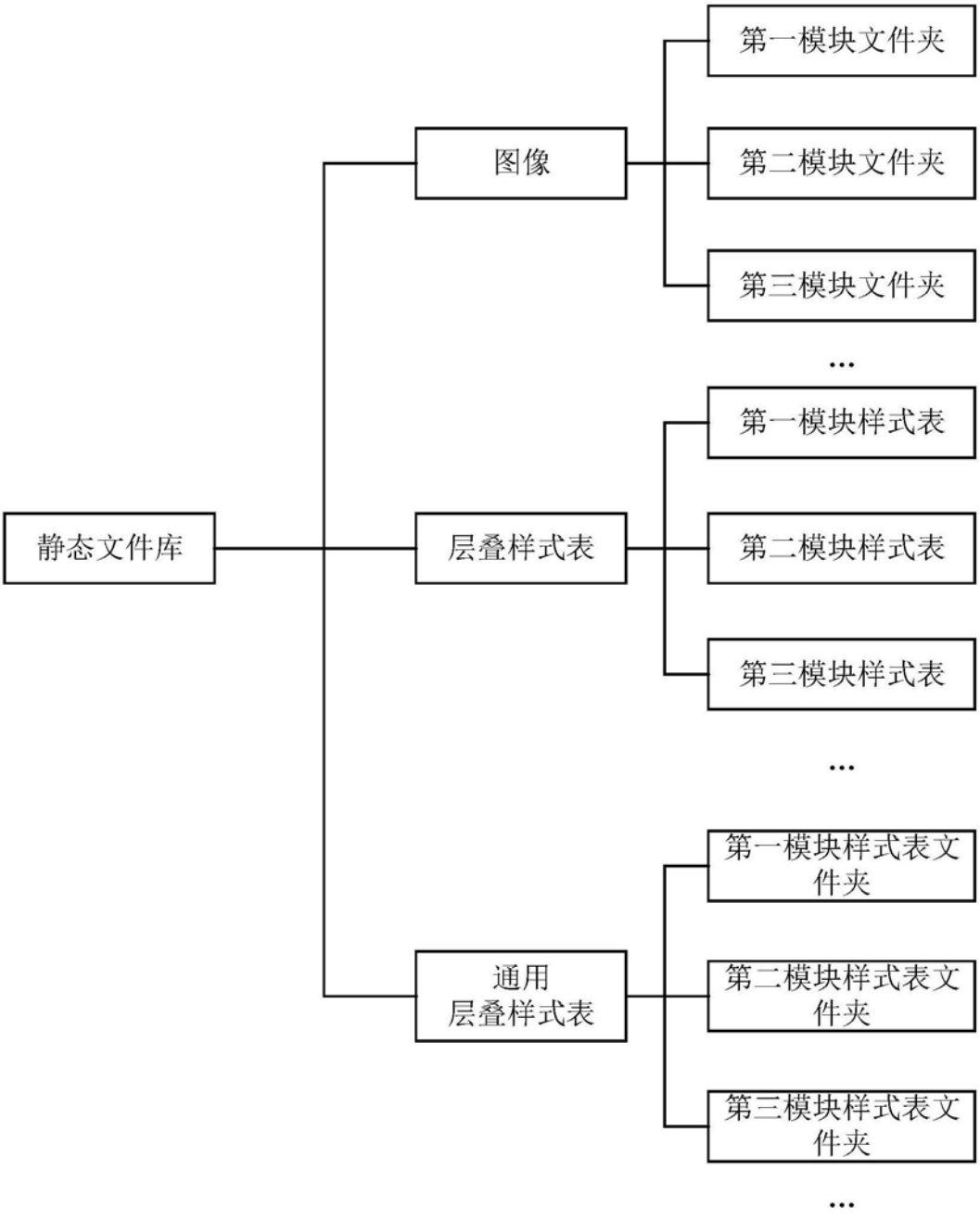

静态文件库
图像
层叠样式表
通用
层叠样式表
第一模块文件夹
第二模块文件夹
第三模块文件夹
...
第一模块样式表
第二模块样式表
第三模块样式表
...
第一模块样式表文
件夹
第二模块样式表文
件夹
第三模块样式表文
件夹
...

图2b

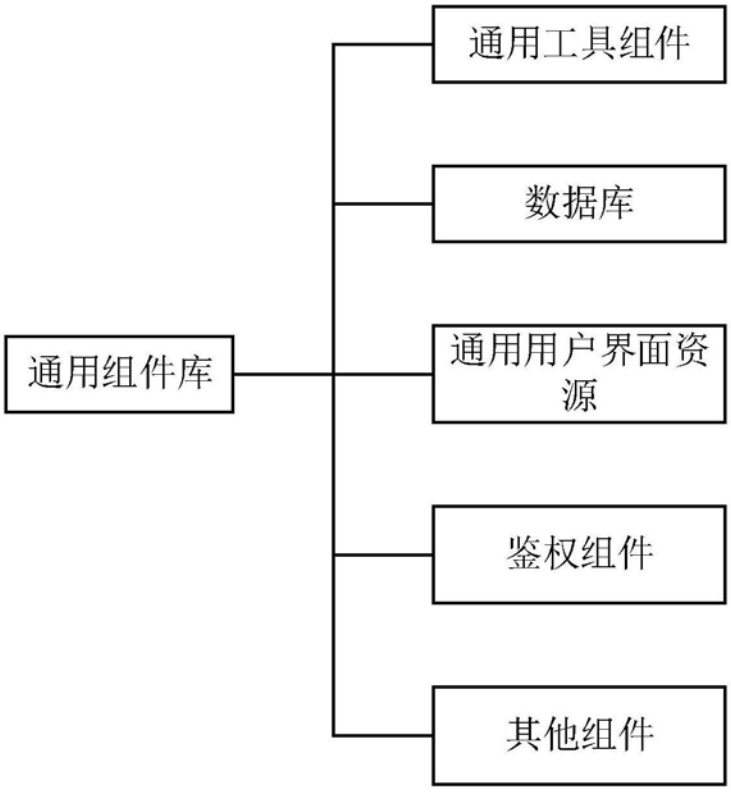

通用工具组件
数据库
通用组件库
通用用户界面资源
鉴权组件
其他组件

图2c


基于架构开发模块
310
打包发布模块
320

图3

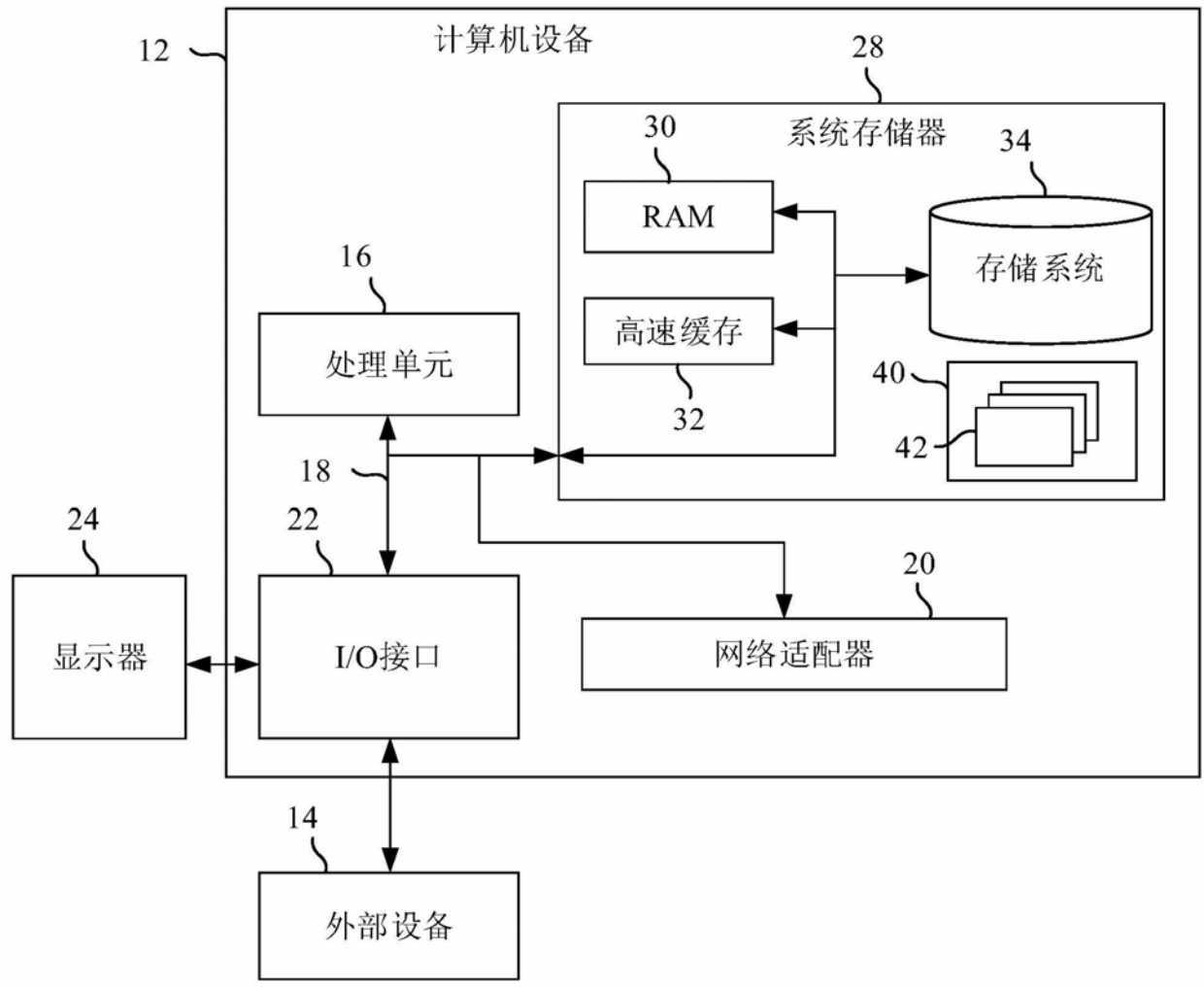

12
计算机设备
28
30
系统存储器
34
RAM
存储系统
16
处理单元
高速缓存
32
40
42
18
24
22
20
显示器
I/O接口
网络适配器
14
外部设备

图4