



(12) 发明专利申请

(10) 申请公布号 CN 101727490 A

(43) 申请公布日 2010.06.09

(21) 申请号 200910254385.3

(22) 申请日 2009.12.22

(71) 申请人 福建星网锐捷网络有限公司

地址 350002 福建省福州市仓山区金山大道
618号桔园州工业园19号楼

(72) 发明人 卢曙亮

(74) 专利代理机构 北京同达信恒知识产权代理
有限公司 11291

代理人 黄志华

(51) Int. Cl.

G06F 17/30(2006.01)

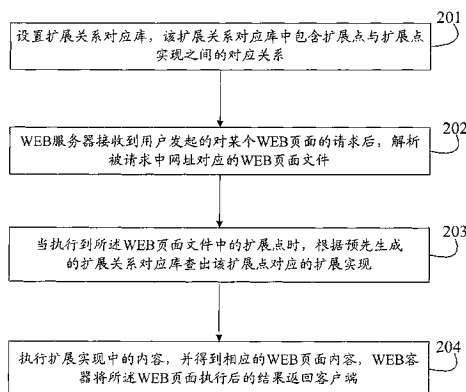
权利要求书 2 页 说明书 7 页 附图 3 页

(54) 发明名称

WEB 应用中实现页面插件化的方法和 WEB 服务器

(57) 摘要

本发明公开了 WEB 应用中实现页面插件化的方法和 WEB 服务器,应用于网络技术领域,该方法包括:WEB 服务器接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。应用本发明实施例提供的方法和装置能够解决现有技术中 WEB 应用中各模块在界面展示层相互耦合,难于扩展和组装的问题。



1. 一种 WEB 应用中实现页面插件化的方法,其特征在于,包括:

Web 服务器接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;

当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;

将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

2. 如权利要求 1 所述的方法,其特征在于,生成所述扩展关系对应库包括:

扫描 WEB 应用程序的各功能模块的配置文件,并从所述配置文件中解析出扩展点与扩展实现的对应关系;

记录所述扩展点与扩展实现的对应关系,生成扩展关系对应库。

3. 如权利要求 1 所述的方法,其特征在于,还包括:

当删除或添加 WEB 页面中的任一扩展实现时,确定所述扩展实现所对应的扩展点,并在所述扩展关系对应库中删除或添加所述扩展点与所述扩展实现的对应关系;

在所述扩展实现对应的功能模块的配置文件中删除或添加所述扩展实现的相关信息。

4. 如权利要求 1~3 任一所述的方法,其特征在于,所述扩展点对应多个扩展实现,所述多个扩展实现的相关信息分别存储在不同功能模块的配置文件中。

5. 如权利要求 1 所述的方法,其特征在于,所述扩展点在所述 WEB 页面文件以标签的形式呈现。

6. 如权利要求 1 所述的方法,其特征在于,所述扩展点包括多种类型,并且每种类型的扩展点对应多个相关属性。

7. 一种 WEB 应用中实现页面插件化的装置,其特征在于,包括:

解析模块,用于接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;

页面组件获取模块,用于当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;

页面生成模块,用于将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

8. 如权利要求 7 所述的装置,其特征在于,该装置还包括:

页面组件删增模块,用于当删除或添加 WEB 页面中的任一扩展实现时,确定所述扩展实现所对应的扩展点,并在所述扩展关系对应库中删除或添加所述扩展点与所述扩展实现的对应关系,在所述扩展实现对应的功能模块的配置文件中删除或添加所述扩展实现的相关信息,其中,所述扩展点对应多个扩展实现,所述多个扩展实现的相关信息分别存储在不同功能模块的配置文件中。

9. 如权利要求 8 所述的装置,其特征在于,该装置还包括:

扩展关系对应库生成模块,用于扫描 WEB 应用程序的各功能模块的配置文件,并从所述配置文件中解析出扩展点与扩展实现的对应关系;记录所述扩展点与扩展实现的对应关

系,生成扩展关系对应库。

10. 一种 WEB 服务器,包括权利要求 7 ~ 9 任一权项所述的装置。

WEB 应用中实现页面插件化的方法和 WEB 服务器

技术领域

[0001] 本发明涉及网络技术领域,尤其涉及 WEB 应用中实现页面插件化的方法和 WEB 服务器。

背景技术

[0002] 近些年,随着 Internet 的爆炸性发展,基于 Web 的各种应用也得到了巨大的发展,随着 Web 技术在企业级应用领域的发展,基于 Web 的应用程序逐渐成为企业应用开发的首选。典型的 Web 应用程序通常基于 3 层架构。从后端到前端分别是数据访问层、业务逻辑层和界面表示层。其中,界面表示层通常是由服务端动态页面技术来实现,如 JSP 页面技术和 ASP 页面技术等。在比较复杂的 Web 应用中,模块与模块之间往往存在业务上的耦合,直观地表现在界面表示层的页面上就是,在一个页面中,存在着由各个模块贡献的界面元素,如一个菜单项、一个链接、一个内容面板等,由于一个页面本身包含的界面逻辑在开发期就已经确定了下来,这使得应用的开发者,没有办法,根据需要来定制页面中的界面逻辑包含的范围,并根据特定的需要,来方便的增加或者减少已经有页面中包含的界面元素。

[0003] 在现有的技术方案中,开发者会在开发期就确定了各个页面的内容,比如一个 WEB 应用程序包含三个模块,其中在系统内,开发者开发了一个首页面,这个首页面用来展示一个综合性的信息内容,信息内容中混合着三个模块的内容。

[0004] 目前通用的技术方案如下:

[0005] 场景一:进行全新开发时:

[0006] 首先分别开发三个模块;

[0007] 然后开发首页面,在首页面中分别调用三个模块的程序接口,把接口返回的内容组成所需页面。

[0008] 场景二:需要新增加一个模块时:

[0009] 首先开发新的模块;

[0010] 然后修改首页面,在首页面中新增调用新模块的程序接口,把所有接口返回的内容组成本页面。

[0011] 场景三:需要删除一个模块时:

[0012] 修改首页面,在首页面中删掉原来调用该模块的程序代码,把剩余接口返回的内容组成本页面。

[0013] 从上面的三个应用开发过程常见的场景中,可以很清晰地看到,为了新增加或者减少一个模块,开发者需要频繁地修改首页面,而首页面是全局性的,为所有模块服务的一个页面,在修改中,稍有不慎,将对整个应用的运行造成影响。在当前的技术方案中,每块模块的变动都将影响整个应用。如上面的场景,首页面成为了耦合各个模块的问题滋生点之一。

发明内容

[0014] 本发明提供 WEB 应用中实现页面插件化的方法和 WEB 服务器,用于解决现有 WEB

应用中各模块在界面展示层相互耦合,难于扩展和组装的问题。

[0015] 一种 WEB 应用中实现页面插件化的方法,包括:

[0016] Web 服务器接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;

[0017] 当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;

[0018] 将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

[0019] 其中,生成所述扩展关系对应库包括:

[0020] 扫描 WEB 应用程序的各功能模块的配置文件,并从所述配置文件中解析出扩展点与扩展实现的对应关系;

[0021] 记录所述扩展点与扩展实现的对应关系,生成扩展关系对应库。

[0022] 当删除或添加 WEB 页面中的任一扩展实现时,确定所述扩展实现所对应的扩展点,并在所述扩展关系对应库中删除或添加所述扩展点与所述扩展实现的对应关系;

[0023] 在所述扩展实现对应的功能模块的配置文件中删除或添加所述扩展实现的相关信息。

[0024] 所述扩展点对应多个扩展实现,所述多个扩展实现的相关信息分别存储在不同功能模块的配置文件中。

[0025] 其中,所述扩展点在所述 WEB 页面文件以标签的形式呈现。

[0026] 所述扩展点包括多种类型,并且每种类型的扩展点对应多个相关属性。

[0027] 一种 WEB 应用中实现页面插件化的装置,包括:

[0028] 解析模块,用于接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;

[0029] 页面组件获取模块,用于当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;

[0030] 页面生成模块,用于将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

[0031] 本发明提供的方法和装置,应用扩展点和扩展实现将 WEB 页面中所需展现的内容灵活的组合在一起,在用户请求时才将各功能模块中扩展实现的内容插入 WEB 页面中,易于扩展和组装。

附图说明

[0032] 图 1 为本发明实施例应用扩展点和扩展实现结合实现 WEB 页面的示意图;

[0033] 图 2 为本发明实施例一种 WEB 应用中实现页面插件化的方法的流程图;

[0034] 图 3 为本发明实施例一种 WEB 应用中实现页面插件化的装置的结构图。

具体实施方式

[0035] 本发明实施例提供一种 WEB 应用中实现页面插件化的方法,应用于网络技术领域,该方法包括:Web 服务器接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

[0036] 下面结合说明书附图对本发明的具体实施方式进行详细说明。

[0037] 在对本发明进行详细的说明之前,以下为本发明中应用到的相关名词的解释:

[0038] A、扩展点:

[0039] 扩展点是存在于 WEB 页面文件上的“钩子”,系统在全局扫描完所有的配置后,通过这个钩子,确定该扩展点对应的多个扩展实现(即在这个钩子上挂接了多少个“扩展实现”),从而在运行时,把扩展实现的具体内容展示在这个扩展点所在位置。

[0040] 具体实现时,扩展点可以以自定义标签的形式出现在页面代码中,但并不局限于这一种方式,则扩展点也可称为扩展点标签,如下为在主页面中的一个扩展点标签的定义:

[0041] `<extpoint:content id = " foundation.mainPage.content" />`

[0042] 上述扩展点标签表示这个扩展点是“content”类型的一个扩展点,它的 ID 标识是“foundation.mainPage.content”。

[0043] 在被扩展页面中设置扩展点(该扩展点也可称之为扩展点声明)时,可以采用自定义标签的方式,也可以采用先封闭成界面组件再使用的方式,本发明不限定采用什么方式,这儿以 JSP 的自定义 tag 标签为例进行说明。

[0044] JSP 的自定义 tag 标签,是指用户定义的一种 JSP 标记。当一个含有自定义的 tag 标签的 JSP 页面被 jsp 引擎编译成 Servlet 时,tag 标签被转化成了对一个称为 tag 处理类的对象进行的操作。于是当 JSP 页面被 jsp 引擎转化为 Servlet 后,实际上 tag 标签被转化成为了对 tag 处理类的操作。

[0045] Servlet 是用 Java 编写的 Server 端程序,它与协议和平台无关。Servlet 运行于 WEB 服务器(如 Tomcat)中。Java Servlet 可以动态地扩展 Server 的能力,并采用请求-响应模式提供 Web 服务,即能提供服务,供用户通过浏览器对 Web 程序进行访问。

[0046] 另外,根据 WEB 页面中包含的内容,该扩展点的类型也对应的包括多种:

[0047] 扩展点的类型包括但不限于导航菜单、超链接、内容片断等。每种类型的扩展点有其自己所特有的一些属性,如导航菜单类型的扩展点属性包括但不限于名称、目标地址、菜单文字;而内容片断类型的扩展点属性包括但不限于名称、片断页面的路径等。

[0048] 扩展点的属性包括但不限于名称、类型、渲染顺序等。

[0049] B、扩展实现:

[0050] 与“扩展点”的概念对应的扩展实现,是指具体挂接在某个扩展点上的实现内容(实现内容是指扩展实现代表的内容,如果是一个“内容片断”类型的扩展,那么实现内容可能是“一段文字”,若是“菜单”类型的扩展,那么实现内容可能就是“一个菜单项”),以下内容为一个对应的扩展实现的定义:

[0051] `<extension:content id = " moduleD.content"`

[0052] extensionPoint = " foundation.mainPage.content"

[0053] url = " /content.xhtml" />

[0054] 该扩展实现表示,模块 D 中存在一个 ID 标识是“moduleD.content”的扩展点实现,对应的扩展点是上面提到的“foundation.mainPage.content”对应的页面片断是“/content.xhtml”。

[0055] 上述扩展点和扩展实现,由系统在运行时组装在一起,用户通过浏览器访问主页面时,则可以看到上述扩展实现中所指定的“/content.xhtml”页面中的内容。

[0056] 根据上述扩展实现和扩展点的实现方式,则可以在所述“foundation.mainPage.content”下挂接多个扩展实现,这样,用户通过浏览器访问主页面时,则能看到所有扩展实现页面中的内容集合。

[0057] 利用上述方法提供的扩展点和扩展实现,生成如图 1 所示,一个 WEB 页面中同时包括内容 A、内容 B 和内容 C,具体的实现方式可以是:

[0058] 在 Web 应用程序的主页面上创建了一个名称为“mainPage.headerline”且类型为“内容片断”的扩展点(此类型的扩展,可以将各子模块中扩展实现所指定的内容片断自动插入到主页面中,并在运行时组成一个完整的页面供用户访问);

[0059] 然后,模块 A 中编写了一个扩展实现,名称为“moduleA.extension”;模块 B 中编写了一个扩展实现,名称为“moduleB.extension”;模块 C 中编写了一个扩展实现,名称为“moduleC.extension”。

[0060] 这三个实现,分别在三个不同的模块中进行独立编写,物理上文件之间没有耦合。在三个模块的扩展点配置文件中,标识出扩展实现对应的扩展点为“mainPage.headerline”,并指定内容片断的页面路径(内容片断也是一个页面,只不过不是完整的页面,而只是页面的部分内容)。

[0061] 因此,在逻辑上名称为“mainPage.headerline”的扩展点拥有三个扩展实现。在 Web 应用程序运行时,系统通过扩展点全局管理模块,扫描各子模块的扩展点配置文件后,发现名称为“mainPage.headerline”存在着三个扩展实现,于是将此对应关系保存在“扩展关系对应库”中,当主页面被访问时,页面解析引擎发现页面上存在一个名称为“mainPage.headerline”的扩展点标签,通过查找“扩展关系对应库”,发现此扩展点在系统中存在着三个扩展实现,然后将每个扩展实现对应路径下的内容片断页面的内容插入到当前页面中,并向访问者返回一个完整的页面内容。

[0062] 扩展点定义不限制用哪种编程语言实现,本发明以 JSP 自定义 tag 标签这种实现方式为例进行说明,如:

[0063] 在 WEB 页面中我们使用<extpoint:content id = " foundation.mainPage.content" />这个自定义的 tag 标签,是指声明了一个 content 类型的扩展点,扩展点的 id 为“foundation.mainPage.content”。

[0064] 而对于扩展实现的定义,同样不限制用哪种编程语言实现,本发明以 XMLSchema 语言(XML Schema 是一种用来描述 XML 文档结构的语言)为例进行说明:

[0065] <xsd:element name = " content" >

[0066] <xsd:complexType>

[0067] <xsd:complexContent>

```

[0068]         <xsd:extension base = " beans:identifiedType" >
[0069]     <xsd:attribute name = " extensionPoint " type = " xsd:string " use
= " required" />
[0070]         <xsd:attribute name = " url " type = " xsd:string " use
= " required" />
[0071]         </xsd:extension>
[0072]     </xsd:complexContent>
[0073] </xsd:complexType>
[0074] </xsd:element>

```

[0075] 上面定义了 content 类型的扩展实现的属性,包括指明是哪个扩展点的实现 (extentsionPoooint 属性)、本扩展实现的具体页面 (url 属性)。

[0076] 如图 2 所示,本发明实施例提供一种 WEB 应用中实现页面插件化的方法,具体步骤包括:

[0077] 步骤 201,设置扩展关系对应库,该扩展关系对应库中包含扩展点与扩展点实现之间的对应关系;

[0078] 其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;

[0079] 其中生成该扩展关系对应库的具体实现方式可以是:

[0080] 扫描 WEB 应用程序各功能模块的配置文件,并从所述配置文件中解析出的扩展点与扩展实现的对应关系;

[0081] 记录所述扩展点与扩展实现的对应关系,生成扩展关系对应库。

[0082] 在具体的应用中,需要记录扩展点与挂接的扩展实现的对应关系时,在具体的扩展关系对应库中记录的是扩展点标签与扩展实现的对应的关系。

[0083] 扩展关系对应库的具体形式可以如表 1 所示的形式:

扩展点	扩展实现
mainPage.headerline	moduleA.extension
	moduleB.extension
	moduleC.extension
扩展点二	扩展点二的扩展实现 A
	扩展点二的扩展实现 B
扩展点三	扩展点三的扩展实现

[0084] 表 1

[0086] 步骤 202,WEB 服务器接收到用户终端发起的对 WEB 页面请求时,根据所述请求中的网址确定对应的 WEB 页面文件,并解析所述 WEB 页面文件;

[0087] 步骤 203,当解析到所述 WEB 页面文件中的设置的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的所有扩展实现;

[0088] 在具体的实现中,如 WEB 页面文件和扩展关系对应库中扩展点的具体表现形式一

一般为扩展点标签,所以在解析 WEB 页面文件时,解析到的是扩展点对应的扩展点标签,在查找关系库以及关系库中存储扩展点与扩展实现的对应关系时,都是应用的扩展标签具体标示扩展点。

[0089] 步骤 204,将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

[0090] 另外,当需要删除或添加 WEB 页面中的任一扩展实现时,本发明实施例具体事项方式可以是:

[0091] 当删除或添加 WEB 页面中的任一扩展实现时,确定所述扩展实现所对应的扩展点,并在所述扩展关系对应库中删除或添加所述扩展点与所述扩展实现的对应关系;

[0092] 在所述扩展实现对应的功能模块的配置文件中删除或添加所述扩展实现的相关信息。

[0093] WEB 应用程序包括多个功能模块,其中每个功能模块包含多个扩展实现,各功能模块的配置文件中记载了各扩展实现的实现内容。当需要修改 WEB 页面中的内容,即相当于删除或添加扩展实现时,则可直接从对应功能模块的配置文件中删除或添加相应扩展实现的实现内容。

[0094] 根据上述方法,用户请求具体的 WEB 页面时的实现方式可以是:

[0095] 如图 3 所示,根据上述方法本发明实施例还提供一种 WEB 应用中实现页面插件化的装置,该装置包括解析模块 301、页面组件获取模块 302 和页面生成模块 303:

[0096] 解析模块 301,接收到用户终端发起的 WEB 页面请求后,根据所述请求中的网址确定对应的 WEB 页面文件,并解析;

[0097] 页面中的需要展现的内容是通过本发明实施例中提供的扩展点和扩展实现结合在一起生成的。

[0098] 页面组件获取模块 302,用于当解析到所述 WEB 页面文件中的扩展点时,根据预先生成的扩展关系对应库查出该扩展点对应的扩展实现,其中,所述扩展点用于挂接扩展实现,所述扩展实现是指挂接到对应扩展点的实现内容;

[0099] 页面生成模块 303,用于将所述扩展点对应的扩展实现的实现内容插入所述 WEB 页面文件,并将所述 WEB 页面文件运行后的结果返回用户终端。

[0100] 另外,为了灵活的修改页面的内容,且不影响页面的整体运行,本发明装置还包括:

[0101] 页面组件删增模块 304,用于当删除或添加 WEB 页面中的任一扩展实现时,确定所述扩展实现所对应的扩展点,并在所述扩展关系对应库中删除或添加所述扩展点与所述扩展实现的对应关系,在所述扩展实现对应的功能模块的配置文件中删除或添加所述扩展实现的相关信息。

[0102] 进一步,本发明装置还包括扩展关系对应库生成模块 305,该扩展关系对应库生成模块 305,用于扫描 WEB 应用程序的各功能模块的配置文件中,并从所述配置文件中解析出扩展点与扩展实现的对应关系;记录所述扩展点与扩展实现的对应关系,生成扩展关系对应库。

[0103] 其中,所述扩展点对应多个扩展实现,所述多个扩展实现的相关信息分别存储在不同功能模块的配置文件中。

[0104] 另外,本发明实施例提供的装置可以置于 WEB 服务器。

[0105] 应用本发明实施例提供的方法和装置,Web 应用程序的开发者通过本方法,可以使 Web 应用程序组织灵活,结构清晰,并且支持模块级的组装,可以方便地在需要时,增加或删除一个新的模块,从而组装出一个新的应用程序,而这种改变,不需要改动其它模块或者全局性的程序文件,极大地降低了因为模块变动,而带来的整个系统的修改。

[0106] 应用本发明提供的提供的方法和装置可以让开发者在 Web 应用程序的开发中,先开发出各自独立的模块,开发时是相互不受影响的,在 Web 应用程序运行时,通过本发明提供的方法和装置,将各模块的内容动态地组装在一个页面上,供用户访问。通过这样的灵活处理,实现了分散开发,整合运行。

[0107] 将本来属于一个页面的各界面元素,物理上隔离在业务模块内部,互不影响,达到模块间界面逻辑解耦的目的,为 Web 应用程序的开发和维护提供极大的便利。

[0108] 本发明所述的方法并不限于具体实施方式中所述的实施例,本领域技术人员根据本发明的技术方案得出其它的实施方式,同样属于本发明的技术创新范围。显然,本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样,倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。

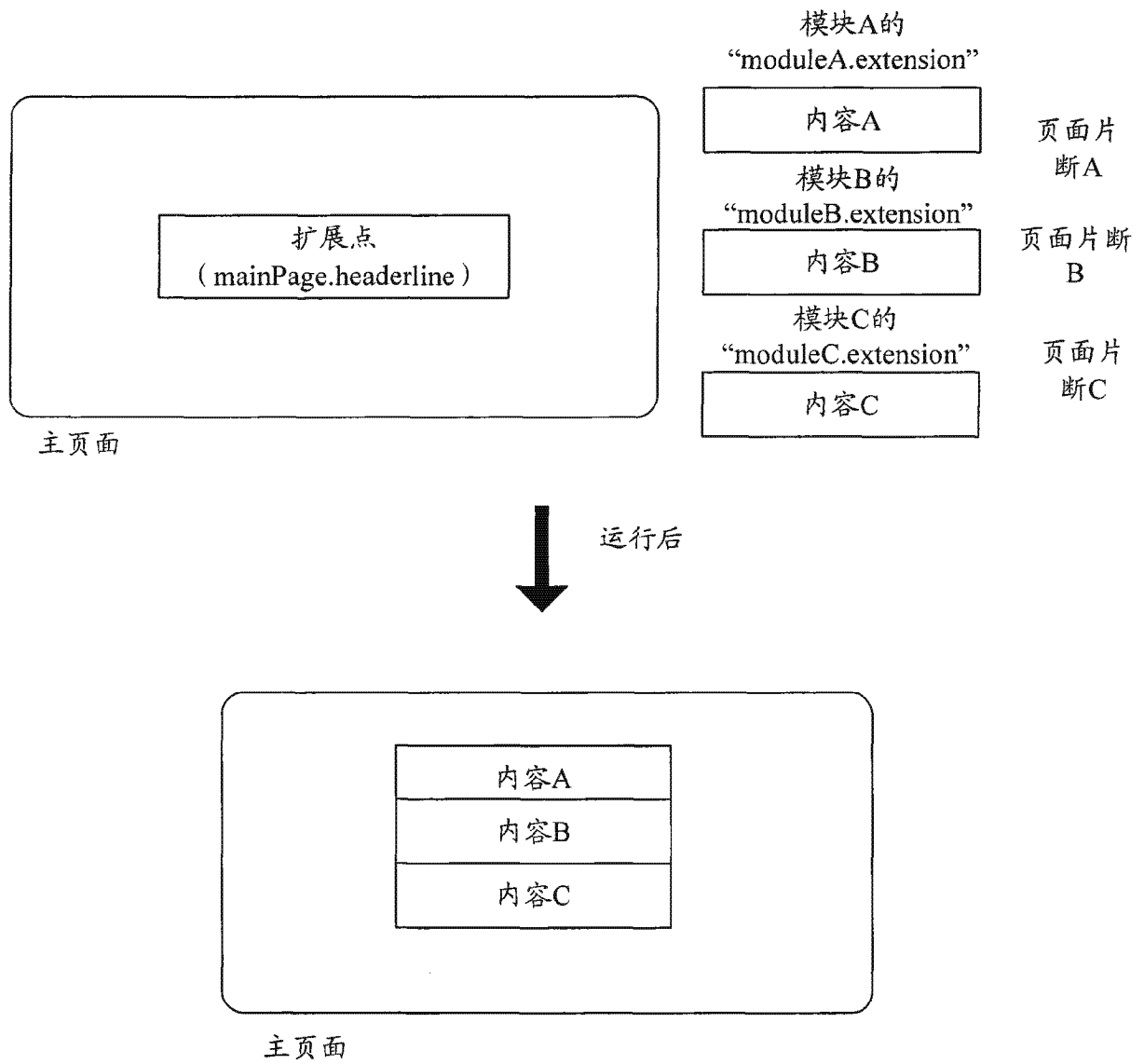


图 1

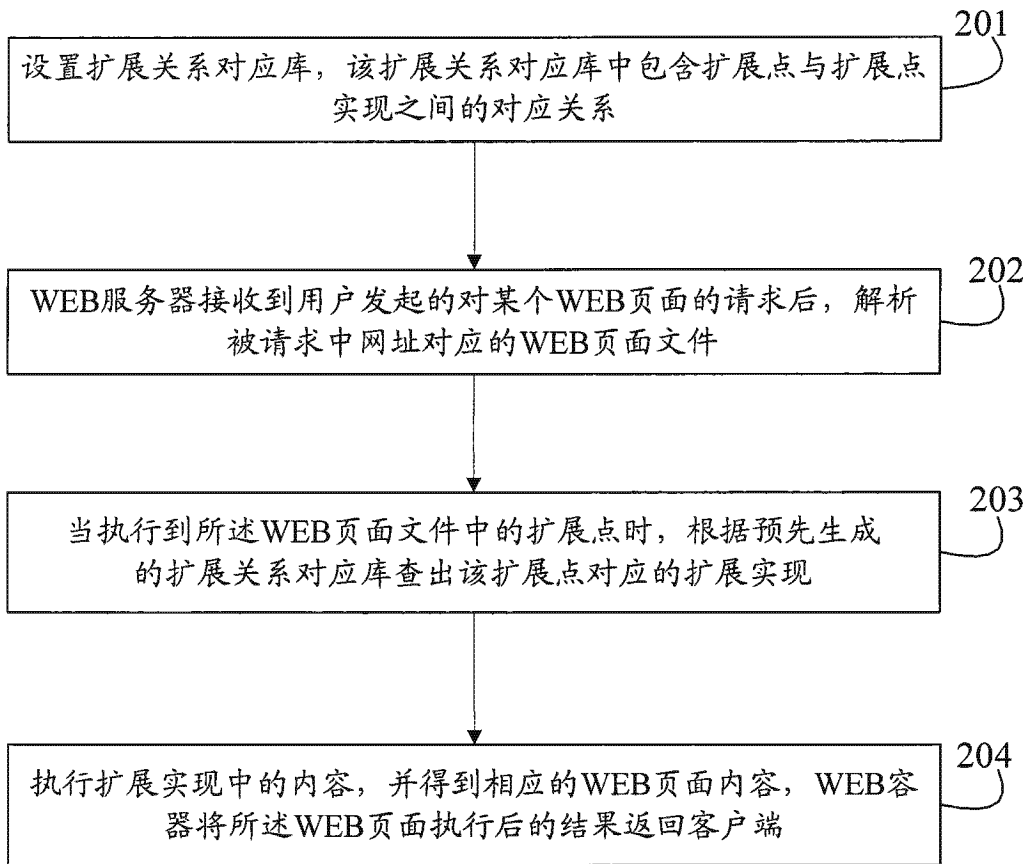


图 2

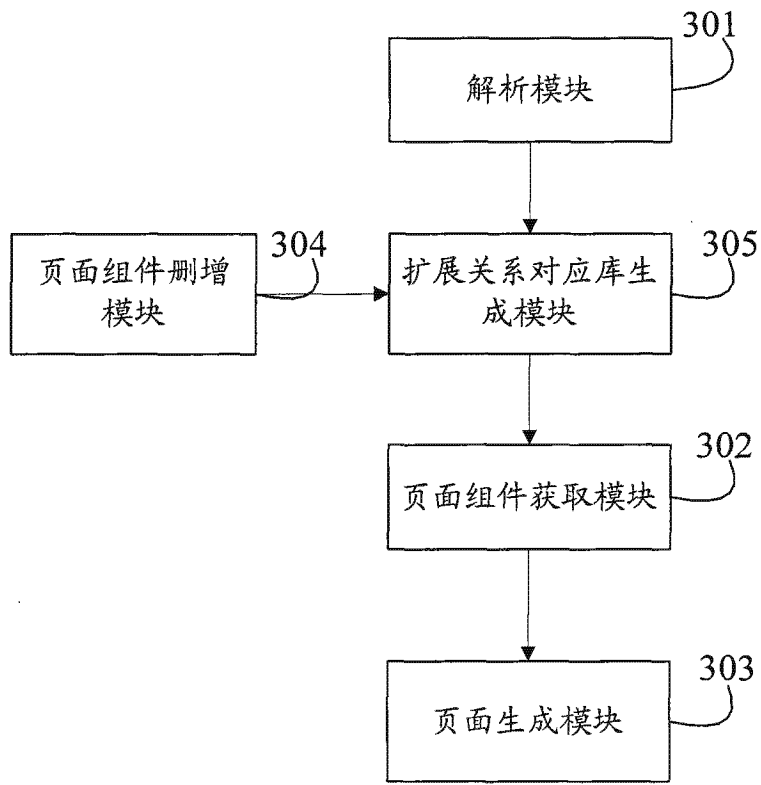


图 3