

(12) 发明专利

(10) 授权公告号 CN 102043626 B

(45) 授权公告日 2013. 12. 11

(21) 申请号 201010603412. 6

(22) 申请日 2010. 12. 23

(73) 专利权人 中国农业银行股份有限公司
地址 100005 北京市东城区建国门内大街
69 号

(72) 发明人 杨鹏 朱玥 邹德庆

(74) 专利代理机构 北京集佳知识产权代理有限
公司 11227
代理人 逯长明 王宝筠

(51) Int. Cl.
G06F 9/44 (2006. 01)

(56) 对比文件
JP 特开 2008-176520 A, 2008. 07. 31, 全文.
陈迟兵. 一种基于标签库的 Webservice 二次
快速开发模式. 《软件导刊》. 2009, 第 8 卷 (第 5

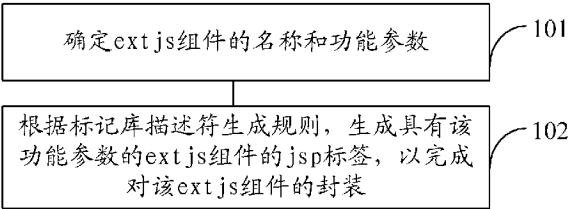
期), 第 126 至 128 页.
awdxzc. 自定义 jsp 标签. 《http://
ericxiong. iteye. com/blog/831535》. 2010, 第 1
页.

审查员 孙韬敏

权利要求书2页 说明书7页 附图1页

(54) 发明名称
extjs 组件的封装方法及装置、用户界面的生
成方法及系统

(57) 摘要
本发明提供一种 extjs 组件的封装方法及装
置、用户界面的生成方法及系统。一种 extjs 组件
的封装方法,包括:确定 extjs 组件的名称和功能
参数;根据标记库描述符生成规则,生成具有所
述功能参数的所述 extjs 组件的 jsp 标签,以完成
对所述 extjs 组件的封装,其中,所述 jsp 标签的
名称与所述 extjs 组件的名称相对应。在本发明
实施例中,通过生成 extjs 组件的对应 jsp 标签,
将复杂的 extjs 组件封装成简单易用的 jsp 标签,
封装了复杂的页面逻辑代码,可简单的实现代码
的复用,使用方便,易于理解,满足了银行业等应
用系统复杂多变的界面展示需求,节省了开发时
间、降低了开发难度,为信息系统前端开发提供了
充分的支持。



1. 一种 extjs 组件的封装方法,其特征在于,包括:

确定 extjs 组件的名称和功能参数;

根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,以完成对所述 extjs 组件的封装,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应,包括:

在生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签。

2. 根据权利要求 1 所述的方法,其特征在于,还包括:

设置调取所述 jsp 标签的路径。

3. 一种用户界面的生成方法,其特征在于,包括:

确定 extjs 组件的名称和功能参数;

根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应,包括:

在生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签;

根据用户界面所需 extjs 组件的名称,调取与所述所需 extjs 组件的名称相对应的 jsp 标签;

根据调取的 jsp 标签生成所述用户界面。

4. 根据权利要求 3 所述的方法,其特征在于,在所述生成所述用户界面之前,还包括:

设置调取的所述 jsp 标签的属性值。

5. 根据权利要求 4 所述的方法,其特征在于,所述根据用户界面所需 extjs 组件的名称,调取与所述所需 extjs 组件的名称相对应的 jsp 标签,具体为:

根据用户界面所需 extjs 组件的名称,确定与所述所需 extjs 组件的名称相对应的 jsp 标签,根据所述 jsp 标签的调取路径,调取所述 jsp 标签。

6. 一种 extjs 组件的封装装置,其特征在于,包括:

组件确定单元,用于确定 extjs 组件的名称和功能参数;

标签生成单元,用于根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,以完成对所述 extjs 组件的封装,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应,其中:

所述组件确定单元,还用于在所述标签生成单元生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

所述标签生成单元,具体用于根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签。

7. 根据权利要求 6 所述的装置,其特征在于,还包括:

路径设置单元,用于设置调取所述 jsp 标签的路径。

8. 一种用户界面的生成系统,其特征在于,包括:

组件确定模块,用于确定 extjs 组件的名称和功能参数;

标签生成模块,用于根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应,并且在生成所述 jsp 标签之前,确定所述 extjs 组件的属性,并根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签;

标签调取模块,用于根据用户界面所需 extjs 组件的名称,调取与所述所需 extjs 组件的名称相对应的 jsp 标签;

界面生成模块,用于根据调取的 jsp 标签生成所述用户界面。

9. 根据权利要求 8 所述的系统,其特征在于,还包括:

属性设置模块,用于在所述界面生成模块生成所述用户界面之前,设置调取的所述 jsp 标签的属性值。

10. 根据权利要求 9 所述的系统,其特征在于,

所述标签调取模块,具体用于根据用户界面所需 extjs 组件的名称,确定与所述所需 extjs 组件的名称相对应的 jsp 标签,根据所述 jsp 标签的调取路径,调取所述 jsp 标签。

11. 根据权利要求 8 至 10 中任意一项所述的系统,其特征在于,所述 extjs 组件包括下列之一或任意组合:

显示数字金额时同步显示汉字金额的组件、右键菜单组件、工具栏位置设置组件、表格联动同屏编辑组件、树形列表窗口组件、对列表中的数据实现小计的组件、对列表中的数据实现总计的组件、汇总表头组件、信息提示组件、表单注释组件、联想输入组件。

extjs 组件的封装方法及装置、用户界面的生成方法及系统

技术领域

[0001] 本发明涉及数据处理技术领域,尤其涉及一种 extjs 组件的封装方法及装置、用户界面的生成方法及系统。

背景技术

[0002] extjs 是一个强大的 js 类库,主要用于创建前端用户界面,通过参考 JavaSwing 等机制来组织可视化组件,无论从 UI 界面上 CSS 样式的应用,到数据解析上的异常处理,都可算是一款不可多得的 JavaScript 客户端技术的精品。

[0003] 现有技术中,用户界面的创建或生成方法是根据 extjs 框架提供的组件,直接编写至 JavaScript 代码,通过运行代码调用 extjs 组件来生成相应的用户界面,然而,由于对 extjs 框架提供的组件没有固定的使用模板或限制,一个用户界面的组件可能存在多种编写方式,这使得调用组件生成用户界面的过程变得复杂和多样化,后续若要重复使用一段代码而无需重新编写,以提高工作效率时,也很难确定需要复用的代码段,从而增加了代码复用的难度。

发明内容

[0004] 本发明实施例提供一种 extjs 组件的封装方法及装置、用户界面的生成方法及系统,能够形成模板化的用户界面组件,以便于形成用户界面和代码复用。

[0005] 为了解决上述技术问题,本发明实施例的技术方案如下:

[0006] 一种 extjs 组件的封装方法,包括:

[0007] 确定 extjs 组件的名称和功能参数;

[0008] 根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,以完成对所述 extjs 组件的封装,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应。

[0009] 进一步,还包括:

[0010] 在生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

[0011] 所述根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,具体为:

[0012] 根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签。

[0013] 进一步,还包括:

[0014] 设置调取所述 jsp 标签的路径。

[0015] 一种用户界面的生成方法,包括:

[0016] 确定 extjs 组件的名称和功能参数;

[0017] 根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标

签,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应;

[0018] 根据用户界面所需 extjs 组件的名称,调取与所述所需 extjs 组件的名称相对应的 jsp 标签;

[0019] 根据调取的 jsp 标签生成所述用户界面。

[0020] 进一步,还包括:

[0021] 在生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

[0022] 所述根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,具体为:

[0023] 所述根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签。

[0024] 进一步,在所述生成所述用户界面之前,还包括:

[0025] 设置调取的所述 jsp 标签的属性值。

[0026] 进一步,所述根据用户界面所需 extjs 组件的名称,调取与所述所需 extjs 组件的名称相对应的 jsp 标签,具体为:

[0027] 根据用户界面所需 extjs 组件的名称,确定与所述所需 extjs 组件的名称相对应的 jsp 标签,根据所述 jsp 标签的调取路径,调取所述 jsp 标签。

[0028] 一种 extjs 组件的封装装置,包括:

[0029] 组件确定单元,用于确定 extjs 组件的名称和功能参数;

[0030] 标签生成单元,用于根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,以完成对所述 extjs 组件的封装,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应。

[0031] 进一步,所述组件确定单元,还用于在所述标签生成单元生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

[0032] 所述标签生成单元,具体用于根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签。

[0033] 进一步,还包括:

[0034] 路径设置单元,用于设置调取所述 jsp 标签的路径。

[0035] 一种用户界面的生成系统,包括:

[0036] 组件确定模块,用于确定 extjs 组件的名称和功能参数;

[0037] 标签生成模块,用于根据标记库描述符生成规则,生成具有所述功能参数的所述 extjs 组件的 jsp 标签,其中,所述 jsp 标签的名称与所述 extjs 组件的名称相对应;

[0038] 标签调取模块,用于根据用户界面所需 extjs 组件的名称,调取与所述所需 extjs 组件的名称相对应的 jsp 标签;

[0039] 界面生成模块,用于根据调取的 jsp 标签生成所述用户界面。

[0040] 进一步,所述组件确定模块,还用于在所述标签生成模块生成所述 jsp 标签之前,确定所述 extjs 组件的属性;

[0041] 所述标签生成模块,具体用于根据标记库描述符生成规则,生成具有所述功能参数及所述属性的所述 extjs 组件的 jsp 标签。

[0042] 进一步,还包括:

[0043] 属性设置模块,用于在所述界面生成模块生成所述用户界面之前,设置调取的所述 jsp 标签的属性值。

[0044] 进一步,所述标签调取模块,具体用于根据用户界面所需 extjs 组件的名称,确定与所述所需 extjs 组件的名称相对应的 jsp 标签,根据所述 jsp 标签的调取路径,调取所述 jsp 标签。

[0045] 进一步,所述 extjs 组件包括下列之一或任意组合:

[0046] 显示数字金额时同步显示汉字金额的组件、右键菜单组件、工具栏位置设置组件、表格联动同屏编辑组件、树形列表窗口组件、对列表中的数据实现小计的组件、对列表中的数据实现总计的组件、汇总表头组件、信息提示组件、表单注释组件、联想输入组件。

[0047] 在本发明实施例中,通过生成 extjs 组件的对应 jsp (Java Server Pages) 标签,将复杂的 extjs 组件封装成简单易用的 jsp 标签,封装了复杂的页面逻辑代码,可简单的实现代码的复用,使用方便,易于理解,满足了银行业等应用系统复杂多变的界面展示需求,节省了开发时间、降低了开发难度,为信息系统前端开发提供了充分的支持。

附图说明

[0048] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0049] 图 1 是本发明实施例一种 extjs 组件的封装方法流程图;

[0050] 图 2 是本发明实施例一种用户界面的生成方法流程图;

[0051] 图 3 是本发明实施例一种 extjs 组件的封装装置结构示意图;

[0052] 图 4 是本发明实施例一种用户界面生成系统的结构示意图。

具体实施方式

[0053] 为了使本领域技术人员能进一步了解本发明的特征及技术内容,请参阅以下有关本发明的详细说明与附图,附图仅提供参考与说明,并非用来限制本发明。

[0054] 下面结合附图和实施例,对本发明的技术方案进行描述。

[0055] 参见图 1,为本发明实施例一种 extjs 组件的封装方法流程图。

[0056] 该方法可以包括:

[0057] 步骤 101,确定 extjs 组件的名称和功能参数。

[0058] 在本实施例中,首先确定出需要生成标签的 extjs 组件,并确定该组件的名称和功能参数,以文本框组件进行说明,名称为文本框,功能参数为用于表征该文本框所具有的功能的参数,可以根据用户的需求设定或选择该文本框所具有的功能,例如文本的横向、纵向输入。设定组件的功能参数后,生成的标签也具有相同的功能,也即规定了该组件对应的标签只具有设定的功能。

[0059] 其中, extjs 组件也即用于生成用户界面的组件,该组件可以是现有的表格组件、表单组件、工具栏等,还可以是根据组件的生成规则生成的新组件,如显示数字金额时同步显示汉字金额的组件 (Money 组件)、右键菜单组件、工具栏位置设置组件 (可以用于 form,

grid, editgrid, tree, region 中, 其中重要的属性是 Position, 为工具栏在容器内的位置, 可以在容器顶部或底部, 默认为在顶部)、表格联动同屏编辑组件 (该组件可以实现表格组件中的记录详细信息以表单方式在同一页面显示, 并可以实现同屏编辑)、树形列表窗口组件 (可以用于弹出一个窗口, 左侧树形结构, 右侧表格数据)、对列表中的数据实现小计的组件 (可以对列表中的数据实现小计功能)、对列表中的数据实现总计的组件 (可以对列表中的数据实现总计功能)、汇总表头组件 (可以对列表中的数据实现汇总表头功能)、信息提示组件 (可以对列表中的某条记录实现悬浮提示信息)、表单注释组件 (表单 field 等显示注释)、联想输入组件 (可以根据用户输入的不完整信息进行联想匹配) 等。

[0060] 步骤 102, 根据标记库描述符生成规则, 生成具有该功能参数的 extjs 组件的 jsp 标签, 以完成对该 extjs 组件的封装。

[0061] 设定该 extjs 组件后, 即可根据标记库描述符生成规则生成标记库描述文件, 也即 jsp 标签, 该标记库描述符生成规则为已知规则, 根据该规则生成标记库描述文件的过程为现有技术, 此处不再赘述。其中, jsp 标签的名称与 extjs 组件的名称相对应, 在本实施例中, 文本框组件生成文本框标签。

[0062] JSP (Java Server Pages) 是在传统的网页 HTML 文件 (*.htm, *.html) 中插入 Java 程序段 (Scriptlet) 和 JSP 标记 (tag), 从而形成 JSP 文件 (*.jsp)。用 JSP 开发的 Web 应用是跨平台的, 既能在 Linux 下运行, 也能在其他操作系统上运行。jsp 标签是很简单而且可重用的代码结构。

[0063] 标签实际是一种模板化, 把可以复用的代码封装起来对用户不可视, 但是为了适应用户的个性化需求, 还需要给出一些用户可定义的参数, 比如一个文本域, 我们需要用户给出宽度和高度参数, 这样结合用户给出的高度和宽度, 我们就可以绘制出一个文本域。

[0064] 在本发明另一实施例中, 在步骤 101 中, 还可以进一步确定该 extjs 组件的属性, 例如组件标识 ID、触发事件 onClick 等等; 还有一些个性属性, 比如菜单 menu 组件的属性有菜单上显示的文字 text, 菜单的提示信息 tooltip 等等。则在步骤 102 中可以根据标记库描述符生成规则, 生成具有该功能参数及该属性的 extjs 组件的 jsp 标签, 以便于在后续应用该标签式可以设置该标签的属性。

[0065] 在生成该 jsp 标签后, 还可以进一步设置调取该 jsp 标签的 URL 路径, 标签库的描述文件中指定了标签库的 url 路径, 在 jsp 页面头引入该路径就可以查找到对应的标签库, 例如 `<% @taglib uri = " /plantix-ui-ext" prefix = " p" %>`, 代码中, 指定了 uri 就是标签库的路径, 那 jsp 页面就知道到哪里去取得标签代码, 此外还指定了该标签库中的标签前缀为 p, 那该 jsp 标签库的标签都以该前缀开始。例如 `<p:ajaxcomboboxdata data = " ${data.dataMap}" total = " 2" valueProperty = " key" displayProperty = " value" />`。设置调取该 jsp 标签的 URL 路径, 以便在后续调取该标签时, 直接根据该 URL 路径调取标签, 生成用户界面。

[0066] 在本发明实施例中, 通过生成 extjs 组件的对应 jsp 标签, 将复杂的 extjs 组件封装成简单易用的 jsp 标签, 封装了复杂的页面逻辑代码, 可简单的实现代码的复用, 使用方便, 易于理解, 满足了银行业等应用系统复杂多变的界面展示需求, 节省了开发时间、降低了开发难度, 为信息系统前端开发提供了充分的支持。

[0067] 参见图 2, 为本发明实施例一种用户界面的生成方法流程图。

- [0068] 该方法可以包括：
- [0069] 步骤 201, 确定 extjs 组件的名称和功能参数。
- [0070] 在本步骤中, 还可以进一步确定 extjs 组件的属性。
- [0071] 步骤 202, 根据标记库描述符生成规则, 生成具有功能参数的 extjs 组件的 jsp 标签。
- [0072] 其中, jsp 标签的名称与 extjs 组件的名称相对应。
- [0073] 若在上步骤中, 同时确定了组件的属性, 则在本步骤中可以生成具有前述功能参数及属性的 extjs 组件的 jsp 标签。
- [0074] 以上步骤 201 ~ 202 与前述实施例类似, 此处不再赘述。
- [0075] 步骤 203, 根据用户界面所需 extjs 组件的名称, 调取与所需 extjs 组件的名称相对应的 jsp 标签。
- [0076] 若上步骤中同时确定了组件的属性, 则在本步骤之前, 还可以先设置调取的 jsp 标签的属性值, 然后再调取相应的 jsp 标签, 并且根据预先设置的该 jsp 标签的调取路径调取 jsp 标签。
- [0077] 步骤 204, 根据调取的 jsp 标签生成用户界面。
- [0078] jsp 是由应用服务器解释的, 并被翻译成 html 代码, 最终在浏览器显示。不管标准标签, 自定义标签, 包括其它的 jsp 元素, 都是由服务器解释。浏览器只显示服务器解释完后的 html 静态网页。
- [0079] 本发明实施例对 extjs 组件封装完善, 不需要开发人员编写 javascript 代码, 只需要在 jsp 页面中写入封装后的 jsp 标签即可生成用户界面。本发明实施例不仅实现了 extjs 框架下组件的标签化, 而且提供了大量可以应用于银行业行业的新组件。既可简单的实现代码的复用, 使用方便, 易于理解, 又满足了银行业等应用系统复杂多变的界面展示需求, 节省了开发时间、降低了开发难度, 为信息系统前端开发提供了充分的支持。
- [0080] 以上是对本发明实施例方法的描述, 下面对实现上述方法的装置和系统进行介绍。
- [0081] 参见图 3, 为本发明实施例一种 extjs 组件的封装装置结构示意图。
- [0082] 该封装装置可以包括：
- [0083] 组件确定单元 301, 用于确定 extjs 组件的名称和功能参数。
- [0084] 标签生成单元 302, 用于根据标记库描述符生成规则, 生成具有功能参数的 extjs 组件的 jsp 标签, 以完成对 extjs 组件的封装, 其中, jsp 标签的名称与 extjs 组件的名称相对应。
- [0085] 在本实施例中, 组件确定单元 301 确定出需要生成标签的 extjs 组件, 并确定该组件的名称和功能参数, 设定组件的功能参数后, 生成的标签也具有相同的功能, 也即规定了该组件对应的标签只具有设定的功能。其中, extjs 组件也即用于生成用户界面的组件, 该组件可以是现有的表格组件、表单组件、工具栏等, 还可以是根据组件的生成规则生成的新组件, 如显示数字金额时同步显示汉字金额的组件 (Money 组件)、右键菜单组件、工具栏位置设置组件 (可以用于 form, grid, editgrid, tree, region 中, 其中重要的属性是 Position, 为工具栏在容器内的位置, 可以在容器顶部或底部, 默认为在顶部)、表格联动全屏编辑组件 (该组件可以实现表格组件中的记录详细信息以表单方式在同一页面显示, 并

可以实现同屏编辑)、树形列表窗口组件(可以用于弹出一个窗口,左侧树形结构,右侧表格数据)、对列表中的数据实现小计的组件(可以对列表中的数据实现小计功能)、对列表中的数据实现总计的组件(可以对列表中的数据实现总计功能)、汇总表头组件(可以对列表中的数据实现汇总表头功能)、信息提示组件(可以对列表中的某条记录实现悬浮提示信息)、表单注释组件(表单 field 等显示注释)、联想输入组件(可以根据用户输入的不完整信息进行联想匹配)等。设定该 extjs 组件后,标签生成单元 302 即可根据标记库描述符生成规则生成标记库描述文件,也即 jsp 标签,其中,jsp 标签的名称与 extjs 组件的名称相对应。

[0086] 在本发明另一实施例中,组件确定单元 301,还可以用于在标签生成单元 302 生成所述 jsp 标签之前,确定 extjs 组件的属性。标签生成单元 302,可以用于根据标记库描述符生成规则,生成具有功能参数及属性的 extjs 组件的 jsp 标签。该装置也还可以进一步包括路径设置单元,用于设置调取 jsp 标签的路径。

[0087] 在本发明实施例中,通过上述单元生成 extjs 组件的对应 jsp 标签,将复杂的 extjs 组件封装成简单易用的 jsp 标签,封装了复杂的页面逻辑代码,可简单的实现代码的复用,使用方便,易于理解,满足了银行业等应用系统复杂多变的界面展示需求,节省了开发时间、降低了开发难度,为信息系统前端开发提供了充分的支持。

[0088] 参见图 4,为本发明实施例一种用户界面生成系统的结构示意图。

[0089] 该系统可以包括:

[0090] 组件确定模块 401,用于确定 extjs 组件的名称和功能参数。

[0091] 标签生成模块 402,用于根据标记库描述符生成规则,生成具有该功能参数的 extjs 组件的 jsp 标签,其中,jsp 标签的名称与 extjs 组件的名称相对应。

[0092] 标签调取模块 403,用于根据用户界面所需 extjs 组件的名称,调取与所需 extjs 组件的名称相对应的 jsp 标签。

[0093] 界面生成模块 404,用于根据调取的 jsp 标签生成用户界面。

[0094] 其中,组件确定模块 401、标签生成模块 402 与前述 extjs 组件的封装装置中的组件确定单元和标签生成单元类似,且组件确定模块 401,也还可以用于在标签生成模块 402 生成 jsp 标签之前,确定 extjs 组件的属性;标签生成模块 402,具体可以用于根据标记库描述符生成规则,生成具有该功能参数及属性的 extjs 组件的 jsp 标签。

[0095] 本实施例中,在组件确定模块 401、标签生成模块 402 对组件封装形成 jsp 标签后,标签调取模块 403 根据用户界面所需 extjs 组件的名称,调取与所需 extjs 组件的名称相对应的 jsp 标签,具体地,可以根据用户界面所需 extjs 组件的名称,确定与所需 extjs 组件的名称相对应的 jsp 标签,根据 jsp 标签的调取路径,调取 jsp 标签。若在组件确定模块 401 中确定了组件的属性,则本实施例中,该系统还可以包括属性设置模块,用于在界面生成模块 404 生成用户界面之前,设置调取的 jsp 标签的属性值。然后,再由界面生成模块 404 根据调取的 jsp 标签生成用户界面。

[0096] 本发明实施例通过上述单元对 extjs 组件封装完善,不需要开发人员编写 javascript 代码,只需要在 jsp 页面中写入封装后的 jsp 标签即可生成用户界面。本发明实施例不仅实现了 extjs 框架下组件的标签化,而且提供了大量可以应用于银行业行业的新组件。既可简单的实现代码的复用,使用方便,易于理解,又满足了银行业等应用系统复

杂多变的界面展示需求,节省了开发时间、降低了开发难度,为信息系统前端开发提供了充分的支持。

[0097] 以上装置和系统中各单元、模块的具体实现过程请参照前述方法实施例中的相应描述,此处不再赘述。

[0098] 以上所述的本发明实施方式,并不构成对本发明保护范围的限定。任何在本发明的精神和原则之内所作的修改、等同替换和改进等,均应包含在本发明的权利要求保护范围之内。

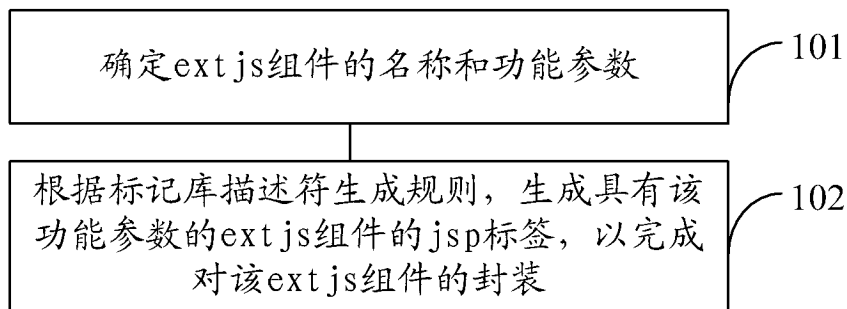


图 1

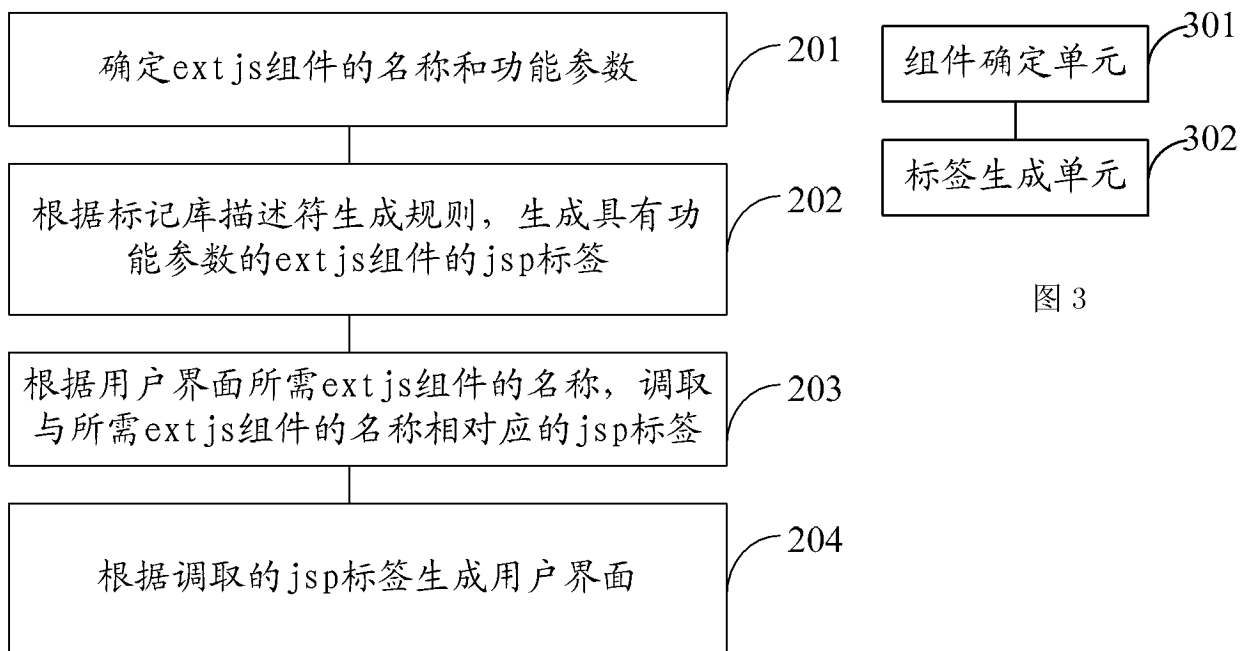


图 3

图 2

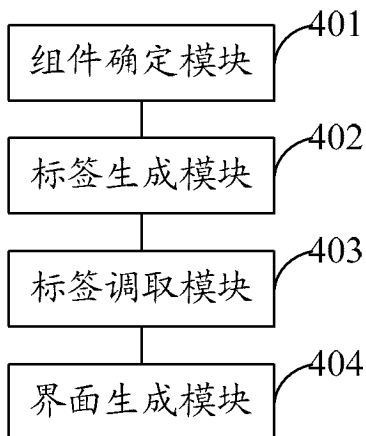


图 4