(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 103530163 B
(45)授权公告日 2017.05.31

(21)申请号 201310512831.2

(22)申请日 2013.10.25

(65)同一申请的已公布的文献号
申请公布号 CN 103530163 A

(43)申请公布日 2014.01.22

(73)专利权人 北京奇虎科技有限公司
地址 100088 北京市西城区新街口外大街28号D座112室(德胜园区)
专利权人 奇智软件(北京)有限公司

(72)发明人 孔令飞 吴军 任寰

(74)专利代理机构 北京市浩天知识产权代理事务所(普通合伙) 11276
代理人 宋菲 刘云贵

(51)Int.Cl.
G06F 9/445(2006.01)
G06F 17/30(2006.01)

(56)对比文件
CN 101395604 A,2009.03.25,????????.
CN 1902679 A,2007.01.24,????????.
US 2003/0069874 A1,2003.04.10,全文.

审查员 许光华

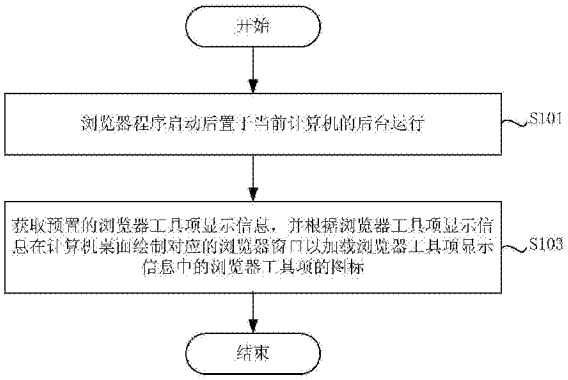
权利要求书2页 说明书11页 附图3页

(54)发明名称

加载浏览器工具项的方法及浏览器

(57)摘要

本发明公开了一种加载浏览器工具项的方法及浏览器，其中，加载浏览器工具项的方法包括：浏览器程序启动后置于当前计算机的后台运行；获取预置的浏览器工具项显示信息，并根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标；其中，所加载的浏览器工具项的图标的响应逻辑为后台运行的浏览器程序中的各相应工具项的响应逻辑。上述加载浏览器工具项的方法及浏览器，根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标，可以定制属于自己的个性化浏览器，满足不同用户的需求，提高用户效率，减少使用负担。



CN 103530163 B

1.一种加载浏览器工具项的方法，包括：

浏览器程序启动后置于当前计算机的后台运行；

获取预置的浏览器工具项显示信息，并根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口以加载所述浏览器工具项显示信息中的浏览器工具项的图标；

其中，所加载的浏览器工具项的图标的响应逻辑为所述后台运行的浏览器程序中的各相应工具项的响应逻辑。

2.根据权利要求1所述的方法，其特征在于，所述浏览器工具项包括：

浏览器默认自带组件、网页类型组件和扩展插件。

3.根据权利要求2所述的方法，其特征在于：

当所述浏览器工具项为浏览器默认自带组件时，所述根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口，包括：

根据所述浏览器工具项显示信息，在浏览器主进程启动后在所述计算机桌面绘制对应的浏览器窗口；

当所述浏览器工具项为网页类型组件或者扩展插件时，所述根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口，包括：

根据所述浏览器工具项显示信息，在浏览器启动后在所述计算机桌面绘制对应的浏览器窗口。

4.根据权利要求1-3任一权利要求所述的方法，其特征在于，每个浏览器窗口的绘制都是在主线程中完成的。

5.根据权利要求4所述的方法，其特征在于，该方法还包括：

当所述浏览器工具项为网页类型组件时，根据当前计算机的性能建立进程。

6.根据权利要求1或2所述的方法，其特征在于，所述浏览器工具项显示信息包括：所述浏览器工具项的图标和名称。

7.根据权利要求2所述的方法，其特征在于，该方法还包括：

在绘制的浏览器窗口中的所述网页类型组件或扩展插件被触发后，以弹窗的形式加载后台运行的浏览器程序在浏览器中打开的页面。

8.根据权利要求1所述的方法，其特征在于，所述获取预置的浏览器工具项显示信息之前，该方法还包括：

从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息。

9.根据权利要求1所述的方法，其特征在于，该方法还包括：

记录绘制的浏览器窗口的位置；

以一个浏览器窗口作为父窗口，将其他一个或多个浏览器窗口作为子窗口绘制在该父窗口中。

10.一种浏览器，包括：

运行模块，适于在浏览器程序启动后置于当前计算机的后台运行；

绘制加载模块，适于获取预置的浏览器工具项显示信息，并根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口以加载所述浏览器工具项显示信息中的浏览器工具项的图标；

其中，所加载的浏览器工具项的图标的响应逻辑为所述后台运行的浏览器程序中的各相应工具项的响应逻辑。

11.根据权利要求10所述的浏览器，其特征在于，所述浏览器工具项包括：

浏览器默认自带组件、网页类型组件和扩展插件。

12.根据权利要求11所述的浏览器，其特征在于：

当所述浏览器工具项为浏览器默认自带组件时，所述绘制加载模块，具体适于：根据所述浏览器工具项显示信息，在浏览器主进程启动后在所述计算机桌面绘制对应的浏览器窗口；

当所述浏览器工具项为网页类型组件或者扩展插件时，所述绘制加载模块，具体适于：根据所述浏览器工具项显示信息，在浏览器启动后在所述计算机桌面绘制对应的浏览器窗口。

13.根据权利要求10-12任一所述的浏览器，其特征在于，每个浏览器窗口的绘制都是在主线程中完成的。

14.根据权利要求13所述的浏览器，其特征在于，所述绘制加载模块，还适于：

当所述浏览器工具项为网页类型组件时，根据当前计算机的性能建立进程。

15.根据权利要求10或11所述的浏览器，其特征在于，所述浏览器工具项显示信息包括：所述浏览器工具项的图标和名称。

16.根据权利要求11所述的浏览器，其特征在于，所述绘制加载模块，还适于：在绘制的浏览器窗口中的所述网页类型组件或扩展插件被触发后，以弹窗的形式加载后台运行的浏览器程序在浏览器中打开的页面。

17.根据权利要求10所述的浏览器，其特征在于，该浏览器还包括：

获取更新模块，适于在所述绘制加载模块获取预置的浏览器工具项显示信息之前，从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息。

18.根据权利要求10所述的浏览器，其特征在于，所述绘制加载模块，还适于：记录绘制的浏览器窗口的位置；以一个浏览器窗口作为父窗口，将其他一个或多个浏览器窗口作为子窗口绘制在该父窗口中。

加载浏览器工具项的方法及浏览器

技术领域

[0001] 本发明涉及计算机技术，具体涉及一种加载浏览器工具项的方法及浏览器。

背景技术

[0002] 浏览器是最经常使用到的客户端程序，目前浏览器的界面通常加载默认的所有的界面组件，例如，地址栏、搜索栏、Tab页、起始页、插件扩展栏、状态栏、工具栏等。

[0003] 现有技术中的浏览器界面都是固定化的，各个组件显示的区域都是固定的，即使可以调整也是限定在浏览器的窗口中，其中的各个组件也是固定好位置的，用户只能进行配置来规定其是否显示这些组件。

[0004] 由于浏览器界面中的组件是固定的，无法进行组合的加载显示，并且界面的组织形式单一，不方便用户使用。

发明内容

[0005] 鉴于上述问题，提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的加载浏览器工具项的方法及浏览器。

[0006] 根据本发明的一个方面，提供了一种加载浏览器工具项的方法，包括：

[0007] 浏览器程序启动后置于当前计算机的后台运行；

[0008] 获取预置的浏览器工具项显示信息，并根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标；

[0009] 其中，所加载的浏览器工具项的图标的响应逻辑为后台运行的浏览器程序中的各相应工具项的响应逻辑。

[0010] 根据本发明的另一方面，提供了一种浏览器，包括：

[0011] 运行模块，适于在浏览器程序启动后置于当前计算机的后台运行；

[0012] 绘制加载模块，适于获取预置的浏览器工具项显示信息，并根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标；

[0013] 其中，所加载的浏览器工具项的图标的响应逻辑为后台运行的浏览器程序中的各相应工具项的响应逻辑。

[0014] 本发明实施例，根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标，可以定制属于自己的个性化浏览器，满足不同用户的需求，提高用户效率，减少使用负担。

[0015] 上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，而可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂，以下特举本发明的具体实施方式。

附图说明

[0016] 通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。在附图中：

[0017] 图1a示出了根据本发明一个实施例的加载浏览器工具项的方法的流程图；

[0018] 图1b示出了根据本发明另一个实施例的加载浏览器工具项的方法的流程图；

[0019] 图2示出了根据本发明一个实施例的浏览器工具项显示信息的示意图；

[0020] 图3示出了根据本发明一个实施例的显示结果页面的示意图；

[0021] 图4示出了根据本发明一个实施例的插件组合样式的示意图；

[0022] 图5示出了根据本发明另一个实施例的浏览器的结构示意图。

具体实施方式

[0023] 下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

[0024] 图1a示出了根据本发明一个实施例的加载浏览器工具项的方法的流程图。如图1a所示，该加载浏览器工具项的方法包括：

[0025] 步骤S101、浏览器程序启动后置于当前计算机的后台运行；

[0026] 步骤S103、获取预置的浏览器工具项显示信息，并根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标。

[0027] 其中，所加载的浏览器工具项的图标的响应逻辑为后台运行的浏览器程序中的各相应工具项的响应逻辑；上述浏览器工具项包括：浏览器默认自带组件、网页类型组件和扩展插件，该浏览器默认自带组件包括地址栏、搜索栏、收藏夹、主页、前进、后退和更新等；上述网页类型组件是指加载了指定统一资源定位符(URL)的浏览器网页的组件，例如起始页面中的特定网页频道等；上述扩展插件包括扩展和插件，“扩展”是基于浏览器本身增加的一些实用功能，而“插件”则是在浏览器之外独立编写的程序，用于显示网页中的特定内容；由于“扩展”往往只是一些以源码形式存储的脚本，所以文件都非常小，而“插件”多是经过编译的程序，体积比“扩展”要庞大得多。

[0028] 上述预置的浏览器工具项显示信息如图2所示，包括浏览器工具项的图标和名称，在用户点击后即可以启动在计算机桌面上对相应浏览器工具项图标的绘制。具体地，当上述浏览器工具项为浏览器默认自带组件时，根据上述浏览器工具项显示信息在上述计算机桌面绘制对应的浏览器窗口包括：根据上述浏览器工具项显示信息，在浏览器主进程启动后在上述计算机桌面绘制对应的浏览器窗口；当上述浏览器工具项为网页类型组件或者扩展插件时，根据上述浏览器工具项显示信息在上述计算机桌面绘制对应的浏览器窗口包括：根据上述浏览器工具项显示信息，在浏览器启动后在上述计算机桌面绘制对应的浏览器窗口。

[0029] 对于不同的浏览器工具项，需要建立不同的窗口绘制类。例如，对于浏览器默认自带组件中的搜索框组件，该搜索框组件可以继承窗口绘制的基类，重写(OVERRIDE)基类的

绘制图标，窗口名称，窗口大小等虚方法，实现点击事件的功能，具体地，可以进行如下设计：

```
class SearchMainWindow : public   views::WidgetDelegateView,
                         public   views::TextfieldController,
                         public   views::ButtonListener,
                         public   views::LinkListener,
                         public   SearchObserver,
                         public views::WidgetObserver{
public:
   SearchMainWindow ();
   ~ SearchMainWindow ();
   virtual void Show();
   bool Init();
```

[0030]

[0031] //重写WidgetDelegate(Override from WidgetDelegate)，WidgetDelegate是一个为Widget显示窗口时提供信息的接口，比如说窗口的标题，图标，以及是否可以被重设大小

```
 virtual std::string GetWindowName() const OVERRIDE;
virtual bool ShouldShowWindowIcon() const OVERRIDE {
   return false;
};
```

[0032]

[0033]

```
    virtual string16 GetWindowTitle() const OVERRIDE;
    virtual gfx::ImageSkia GetWindowIcon() OVERRIDE;
    virtual gfx::ImageSkia GetWindowAppIcon() OVERRIDE;
    virtual views::View* GetContentsView() OVERRIDE;
    virtual gfx::Size GetPreferredSize() OVERRIDE;
    virtual void Layout() OVERRIDE;
    //重写 Widget::Observe（Override from Widget::Observe）
    virtual void OnWidgetClosing(Widget* widget);
    // 重写 ButtonListener（Override from ButtonListener）
    virtual void ButtonPressed(views::Button* sender, const ui::Event& event) OVERRIDE;
  private:
    gfx::ImageSkia getBlendedBitmap(int id1,int id2);
    views::ImageButton* search_logo_btn_;
    views::ImageButton* search_action_btn_;
    views::Label* more_ways_label_;
    SearchTextfieldView* find_text_;
  };
```

[0034] 在本实施例中，只需绘制一个搜索图标，搜索输入框和搜索按钮。当监听到用户点击的是搜索组件图标，则实例化并显示该窗口。

[0035] seach_frame_win_=new SearchMainWindow();

[0036] params.native_widget=doctor_frame_win_->AsNativeWidget();

[0037] views::Widget::Init(params);

[0038] 对于内嵌网页的组件窗口，需要在窗口中建立tab页面；而对于扩展插件窗口，和普通组件绘制类似。而每个浏览器窗口的绘制都是在主线程中完成的，只是对于内嵌网页的网页类型组件窗口，可能会根据用户机器性能和实际情况建立进程。

[0039] 图2中的每一个组件被选中后，其后续都会起一个对应的浏览器窗口来绘制该组件对应的功能按钮或者图标，并启用在后台运行的对应的浏览器程序中的相应组件的逻辑功能。同时，可以在计算机桌面上进行显示，例如，在点击图标后生成对应的结果显示界面，如图3所示。对于扩展插件被点击后，会以POP弹窗的形式加载后台运行的浏览器程序在浏览器中打开的页面，该POP页面显示在该组件图标的附近。

[0040] 对各种图标进行点击或者输入关键字进行搜索操作后，产生的后续流程与正常的浏览器中的操作一样，仅仅是改变后续显示结果页面的位置，在POP弹窗中显示结果页面。所有的操作都在该POP弹窗中进行，或者恢复到原浏览器窗口进行也是可以的。

[0041] 另外，在获取预置的浏览器工具项显示信息之前，如图1b所示，该方法还可以包括：

[0042] 步骤S102、从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息。

[0043] 上述配置文件的格式如下：

[0044]


```
"ahfgeienlihckogmohjhadlkjgocpleb": {
                    "active_permissions": {
                        "api": [ "appNotifications", "management", "tabs", "webstorePrivate" ]
                    },
                    "app_launcher_ordinal": "n",
                    "creation_flags": 1,
                    "from_bookmark": false,
                    "from_webstore": false,
                    "install_time": "13011254347332692",
                    "location": 5,
                    "manifest": {
                        "app": {
```

[0045]


```
"launch": {
    "web_url": "https://chrome.google.com/webstore"
},
"urls": [ "https://chrome.google.com/webstore",
"https://ext.chrome.360.cn", "https://skin.chrome.360.cn" ]
},
"description": "Web Store",
"icons": {
    "128": "webstore_icon_128.png",
    "16": "webstore_icon_16.png"
},
"key":
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCtl3tO0osjuzRsf6xtD2SKxPlTfuoy
7AWoObysitBPvH5fE1NaAA1/2JkPWkVDhdLBWLaIBPYeXbzlHp3y4Vv/4XG+aN5qFE3z+1
RU/NqkzVYHtIpVScf3DjTYtKVL66mzVGijSoAIwbFCC3LpGdaoe6Q1rSRDp76wR6jjFzsYw
QIDAQAB",
"name": "\u6269\u5C55\u4E2D\u5FC3",
"permissions": [ "appNotifications", "webstorePrivate", "management" ],
"version": "0.1"
},
"page_ordinal": "n",
path": :\\360chrome26\\src\\build\\Debug\\resources\\web_store",
"was_installed_by_default": false
}
```


[0046] 一个扩展的配置文件通常以扩展的标识为键，其值包括扩展标识，扩展拥有的权限，安装时间，图标(icons)、链接地址，所在目录等配置信息。

[0047] 通过上述配置信息中的icons字段可以获取到该扩展的大图标或小图标，其中，ahfgeienlihckogmohjhadlkjgocpleb是扩展的标识。

[0048] 其中，对于浏览器侧已经安装的扩展，在图2中进行图标显示时会从浏览器中获取

已经安装的扩展的信息，例如，读取浏览器侧extension目录下的profile文件中的关于扩展的描述信息，获取对应的扩展的名称以及其对应的图标信息。

[0049] 由于安装的浏览器扩展插件以及浏览器的Tab网页新功能(例如新频道)是在不断更新的，故图2所示界面中浏览器侧配置的对应的图标和信息也会跟着一起被更新，或者按时从网络侧进行更新。

[0050] 另外，各个组件还可以进行组合，以一个组件图标的窗口作为父窗口，将其它的组件图标可以拖拽后作为子窗口绘制在父窗口中，如图4所示。

[0051] 上述加载浏览器工具项的方法，不用开启浏览器就可以在桌面输入框里直接输入搜索内容，展现结果，从而可以提高使用效率，减少用户和工具的交互成本；另外，对于访问网址较固定的用户，可以保留收藏夹于桌面上，直接选择想访问的网站即可；当用户保留邮件插件时，即使没有打开邮箱也可以提示有新邮件；由此可见，上述加载浏览器工具项的方法，可以定制属于自己的个性化浏览器，满足不同用户的需求，提高用户效率，减少使用负担。

[0052] 图5示出了根据本发明另一个实施例的浏览器的结构示意图。如图5所示，该浏览器包括运行模块51和绘制加载模块52，其中：

[0053] 运行模块适于在浏览器程序启动后置于当前计算机的后台运行；

[0054] 绘制加载模块适于获取预置的浏览器工具项显示信息，并根据上述浏览器工具项显示信息在上述计算机桌面绘制对应的浏览器窗口以加载上述浏览器工具项显示信息中的浏览器工具项的图标；

[0055] 其中，所加载的浏览器工具项的图标的响应逻辑为上述后台运行的浏览器程序中的各相应工具项的响应逻辑。

[0056] 上述浏览器工具项包括：浏览器默认自带组件、网页类型组件和扩展插件，该浏览器默认自带组件包括地址栏、搜索栏、收藏夹、主页、前进、后退和更新等；上述网页类型组件是指加载了指定统一资源定位符(URL)的浏览器网页的组件，例如起始页面中的特定网页频道等；上述扩展插件包括扩展和插件，“扩展”是基于浏览器本身增加的一些实用功能，而“插件”则是在浏览器之外独立编写的程序，用于显示网页中的特定内容；由于“扩展”往往只是一些以源码形式存储的脚本，所以文件都非常小，而“插件”多是经过编译的程序，体积比“扩展”要庞大得多。

[0057] 上述预置的浏览器工具项显示信息如图2所示，包括浏览器工具项的图标和名称，在用户点击后即可以启动在计算机桌面上对相应浏览器工具项图标的绘制。具体地，当上述浏览器工具项为浏览器默认自带组件时，上述绘制加载模块根据上述浏览器工具项显示信息，在浏览器主进程启动后在上述计算机桌面绘制对应的浏览器窗口；当上述浏览器工具项为网页类型组件或者扩展插件时，上述绘制加载模块根据上述浏览器工具项显示信息，在浏览器启动后在上述计算机桌面绘制对应的浏览器窗口。

[0058] 对于不同的浏览器工具项，需要建立不同的窗口绘制类。对于内嵌网页的组件窗口，需要在窗口中建立tab页面。而每个浏览器窗口的绘制都是在主线程中完成的，只是对于内嵌网页的网页类型组件窗口，可能会根据用户机器性能和实际情况建立进程。

[0059] 另外，上述绘制加载模块，还适于：在绘制的浏览器窗口中的上述网页类型组件或扩展插件被触发后，以弹窗的形式加载后台运行的浏览器程序在浏览器中打开的页面。

[0060] 进一步地，该浏览器还包括获取更新模块53，该获取更新模块，适于在上述绘制加载模块获取预置的浏览器工具项显示信息之前，从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息，具体的配置文件格式可参见方法实施例，此处不赘述。

[0061] 另外，上述绘制加载模块还适于：记录绘制的浏览器窗口的位置；以一个浏览器窗口作为父窗口，将其他一个或多个浏览器窗口作为子窗口绘制在该父窗口中，如图4所示。

[0062] 上述浏览器加载浏览器工具项的过程可参见图1a-1b，此处不赘述。

[0063] 上述浏览器加载的搜索框模块，不需要开启浏览器就可以在桌面输入框里直接输入搜索内容，展现结果，从而可以提高使用效率，减少用户和工具的交互成本；上述浏览器加载的收藏夹模块，对于访问网址较固定的用户，可以将收藏夹保留在桌面上，直接选择想访问的网站即可；上述浏览器加载的邮件模块，当用户保留邮件插件时，没有打开邮箱也可以提示有新邮件。

[0064] 由此可见，上述浏览器可以定制属于自己的个性化浏览器，满足不同用户的需求，提高用户效率，减少使用负担。

[0065] 在此提供的算法和显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述，构造这类系统所要求的结构是显而易见的。此外，本发明也不针对任何特定编程语言。应当明白，可以利用各种编程语言实现在此描述的本发明的内容，并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

[0066] 在此处所提供的说明书中，说明了大量具体细节。然而，能够理解，本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中，并未详细示出公知的方法、结构和技术，以便不模糊对本说明书的理解。

[0067] 类似地，应当理解，为了精简本公开并帮助理解各个发明方面中的一个或多个，在上面对本发明的示例性实施例的描述中，本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而，并不应将该公开的方法解释成反映如下意图：即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说，如下面的权利要求书所反映的那样，发明方面在于少于前面公开的单个实施例的所有特征。因此，遵循具体实施方式的权利要求书由此明确地并入该具体实施方式，其中每个权利要求本身都作为本发明的单独实施例。

[0068] 本领域那些技术人员可以理解，可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件，以及此外可以把它们分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外，可以采用任何组合对本说明书（包括伴随的权利要求、摘要和附图）中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述，本说明书（包括伴随的权利要求、摘要和附图）中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0069] 此外，本领域的技术人员能够理解，尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征，但是不同实施例的特征的组合意味着处于本发明的

范围之内并且形成不同的实施例。例如，在下面的权利要求书中，所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0070] 本发明的各个部件实施例可以以硬件实现，或者以在一个或者多个处理器上运行的软件模块实现，或者以它们的组合实现。本领域的技术人员应当理解，可以在实践中使用微处理器或者数字信号处理器(DSP)来实现根据本发明实施例的浏览器中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序(例如，计算机程序和计算机程序产品)。这样的实现本发明的程序可以存储在计算机可读介质上，或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到，或者在载体信号上提供，或者以任何其他形式提供。

[0071] 应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制，并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中，不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中，这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

[0072] 本发明还公开了A1、一种加载浏览器工具项的方法，包括：

[0073] 浏览器程序启动后置于当前计算机的后台运行；

[0074] 获取预置的浏览器工具项显示信息，并根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口以加载所述浏览器工具项显示信息中的浏览器工具项的图标；

[0075] 其中，所加载的浏览器工具项的图标的响应逻辑为所述后台运行的浏览器程序中的各相应工具项的响应逻辑。

[0076] A2、根据A1所述的方法，其特征在于，所述浏览器工具项包括：

[0077] 浏览器默认自带组件、网页类型组件和扩展插件。

[0078] A3、根据A2所述的方法，其特征在于：

[0079] 当所述浏览器工具项为浏览器默认自带组件时，所述根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口，包括：

[0080] 根据所述浏览器工具项显示信息，在浏览器主进程启动后在所述计算机桌面绘制对应的浏览器窗口；

[0081] 当所述浏览器工具项为网页类型组件或者扩展插件时，所述根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口，包括：

[0082] 根据所述浏览器工具项显示信息，在浏览器启动后在所述计算机桌面绘制对应的浏览器窗口。

[0083] A4、根据A1-A3任一所述的方法，其特征在于，每个浏览器窗口的绘制都是在主线程中完成的。

[0084] A5、根据A4所述的方法，其特征在于，该方法还包括：

[0085] 当所述浏览器工具项为网页类型组件时，根据当前计算机的性能建立进程。

[0086] A6、根据A1或A2所述的方法，其特征在于，所述浏览器工具项显示信息包括：所述浏览器工具项的图标和名称。

[0087] A7、根据A2所述的方法，其特征在于，该方法还包括：

[0088] 在绘制的浏览器窗口中的所述网页类型组件或扩展插件被触发后，以弹窗的形式加载后台运行的浏览器程序在浏览器中打开的页面。

[0089] A8、根据A1所述的方法，其特征在于，所述获取预置的浏览器工具项显示信息之前，该方法还包括：

[0090] 从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息。

[0091] A9、根据A1所述的方法，其特征在于，该方法还包括：

[0092] 记录绘制的浏览器窗口的位置；

[0093] 以一个浏览器窗口作为父窗口，将其他一个或多个浏览器窗口作为子窗口绘制在该父窗口中。

[0094] 本发明还公开了B10、一种浏览器，包括：

[0095] 运行模块，适于在浏览器程序启动后置于当前计算机的后台运行；

[0096] 绘制加载模块，适于获取预置的浏览器工具项显示信息，并根据所述浏览器工具项显示信息在所述计算机桌面绘制对应的浏览器窗口以加载所述浏览器工具项显示信息中的浏览器工具项的图标；

[0097] 其中，所加载的浏览器工具项的图标的响应逻辑为所述后台运行的浏览器程序中的各相应工具项的响应逻辑。

[0098] B11、根据B10所述的浏览器，其特征在于，所述浏览器工具项包括：

[0099] 浏览器默认自带组件、网页类型组件和扩展插件。

[0100] B12、根据B11所述的浏览器，其特征在于：

[0101] 当所述浏览器工具项为浏览器默认自带组件时，所述绘制加载模块，具体适于：根据所述浏览器工具项显示信息，在浏览器主进程启动后在所述计算机桌面绘制对应的浏览器窗口；

[0102] 当所述浏览器工具项为网页类型组件或者扩展插件时，所述绘制加载模块，具体适于：根据所述浏览器工具项显示信息，在浏览器启动后在所述计算机桌面绘制对应的浏览器窗口。

[0103] B13、根据B10-B12任一所述的浏览器，其特征在于，每个浏览器窗口的绘制都是在主线程中完成的。

[0104] B14、根据B13所述的浏览器，其特征在于，所述绘制加载模块，还适于：

[0105] 当所述浏览器工具项为网页类型组件时，根据当前计算机的性能建立进程。

[0106] B15、根据B10或B11所述的浏览器，其特征在于，所述浏览器工具项显示信息包括：所述浏览器工具项的图标和名称。

[0107] B16、根据B11所述的浏览器，其特征在于，所述绘制加载模块，还适于：在绘制的浏览器窗口中的所述网页类型组件或扩展插件被触发后，以弹窗的形式加载后台运行的浏览器程序在浏览器中打开的页面。

[0108] B17、根据B10所述的浏览器，其特征在于，该浏览器还包括：

[0109] 获取更新模块，适于在所述绘制加载模块获取预置的浏览器工具项显示信息之前，从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息。

[0110] B18、根据B10所述的浏览器，其特征在于，所述绘制加载模块，还适于：记录绘制的浏览器窗口的位置；以一个浏览器窗口作为父窗口，将其他一个或多个浏览器窗口作为子窗口绘制在该父窗口中。

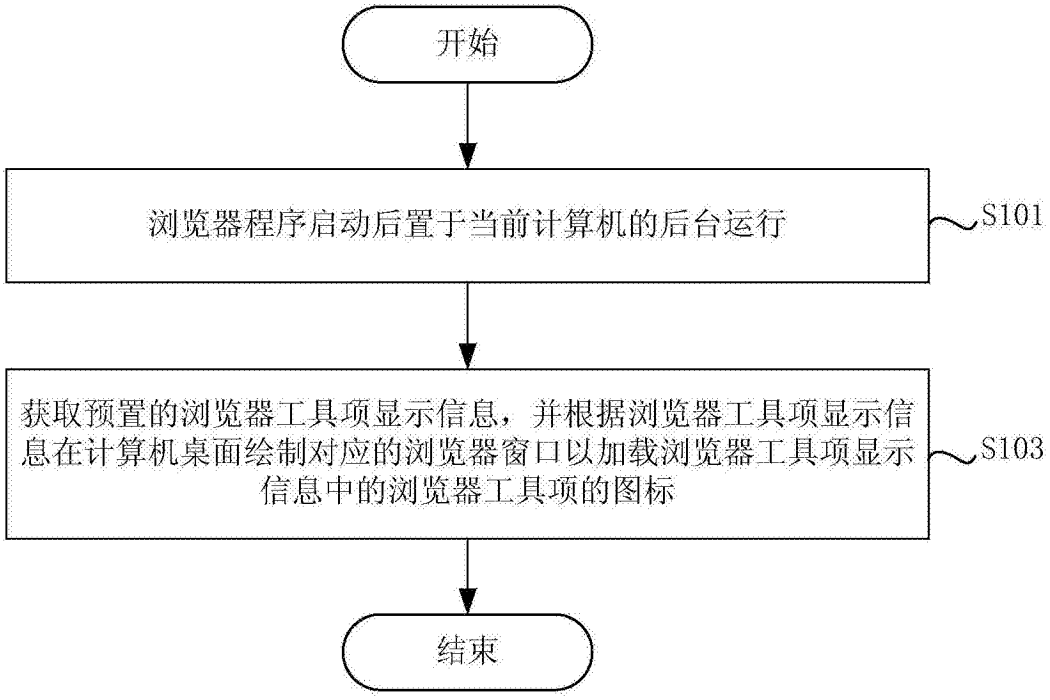

开始
浏览器程序启动后置于当前计算机的后台运行
S101
获取预置的浏览器工具项显示信息，并根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标
S103
结束

图1a

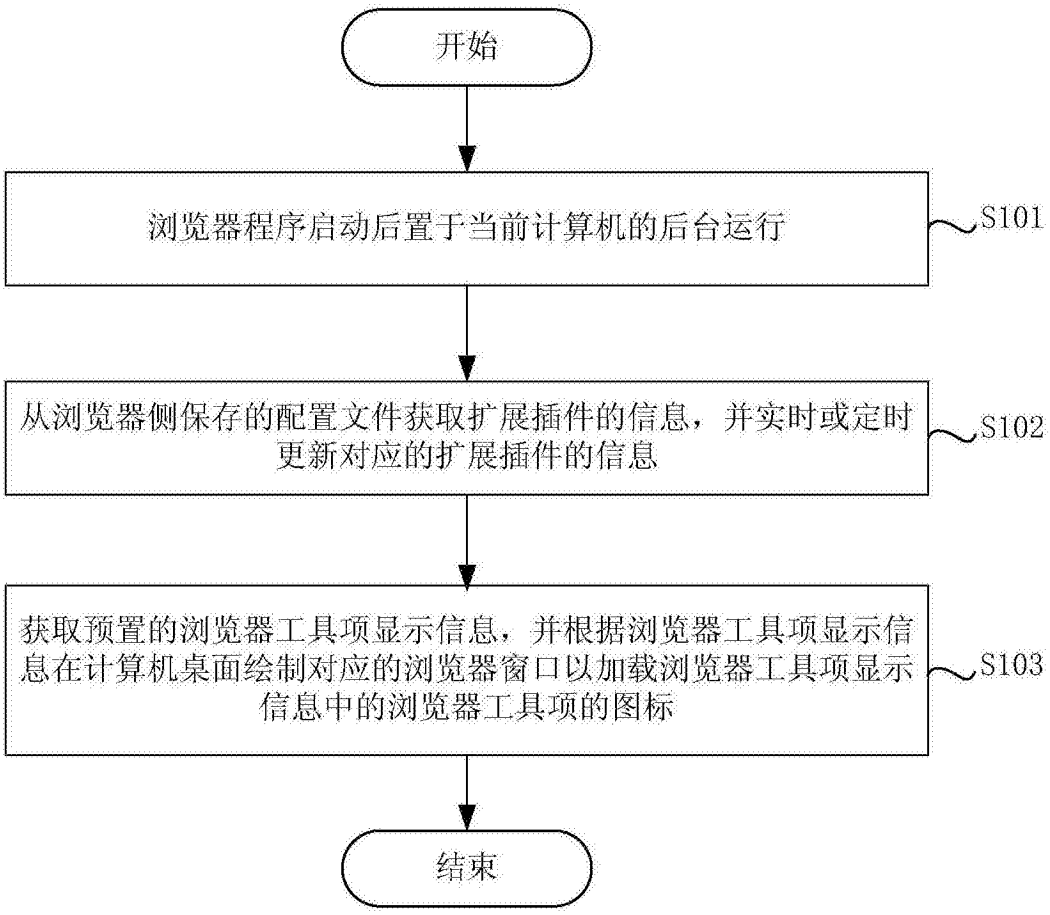

开始
浏览器程序启动后置于当前计算机的后台运行
S101
从浏览器侧保存的配置文件获取扩展插件的信息，并实时或定时更新对应的扩展插件的信息
S102
获取预置的浏览器工具项显示信息，并根据浏览器工具项显示信息在计算机桌面绘制对应的浏览器窗口以加载浏览器工具项显示信息中的浏览器工具项的图标
S103
结束

图1b

图2

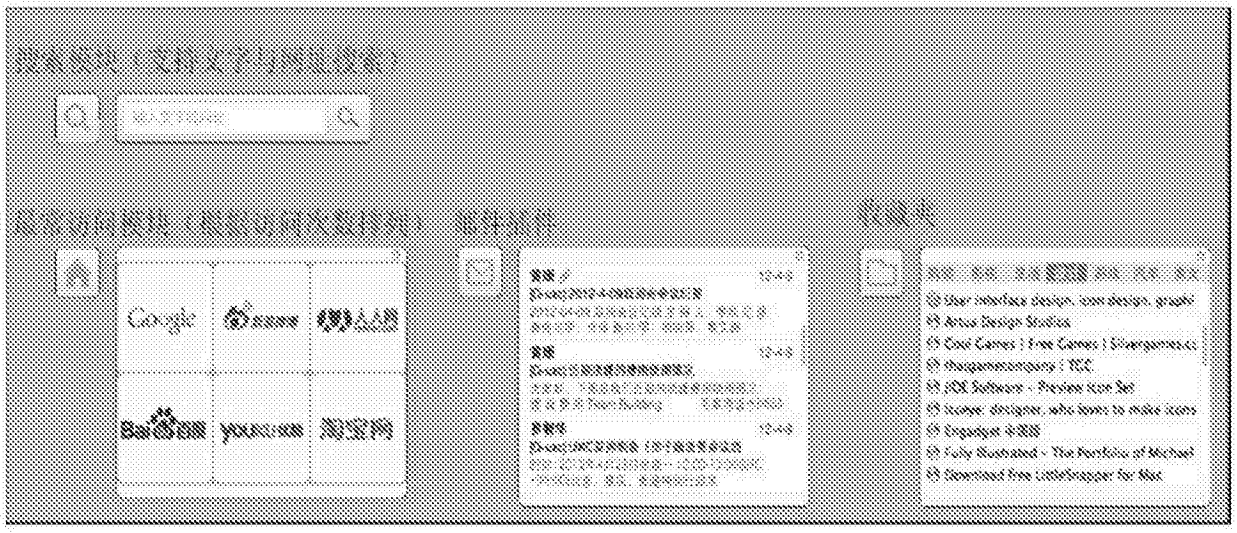


图3

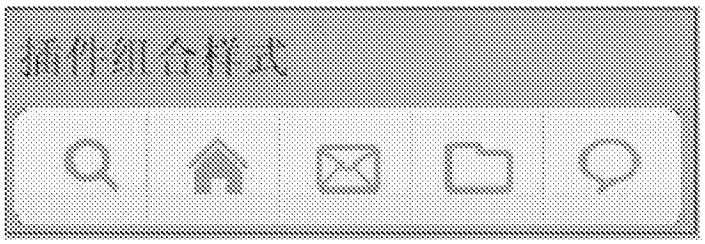


图4

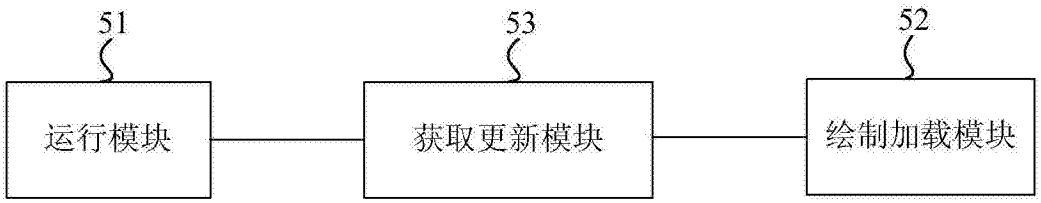


图5