



(12) 发明专利申请

(10) 申请公布号 CN 103500099 A

(43) 申请公布日 2014. 01. 08

(21) 申请号 201310456133. 5

(22) 申请日 2013. 09. 27

(71) 申请人 摩卡软件(天津)有限公司

地址 300384 天津市南开区天津新技术产业
区华苑产业区榕苑路1号鑫茂软件大
厦A区1810-1811室

(72) 发明人 刘东要 曹明远 卢学军

(74) 专利代理机构 北京庆峰财智知识产权代理
事务所(普通合伙) 11417

代理人 刘元霞

(51) Int. Cl.

G06F 9/44 (2006. 01)

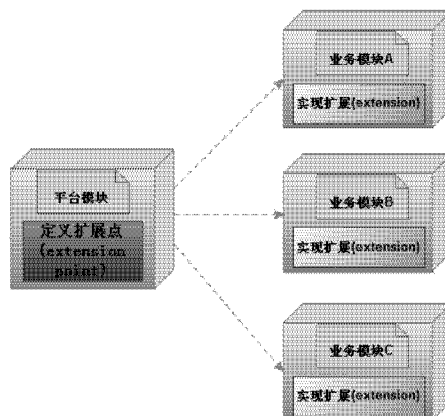
权利要求书1页 说明书3页 附图2页

(54) 发明名称

一种通过扩展点和扩展实现软件快速二次开发的方法

(57) 摘要

本发明公开了一种通过扩展点和扩展实现软件快速二次开发的方法,其特征在于:在软件中定义插件具备的扩展点,并且定义如何使用这些扩展点。由于本发明用xml来描述扩展点和扩展的内容,同时提供与扩展点相对应的Schema定义,这样二次开发人员能够方便快捷的通过xml配置将新的功能点加入到产品已有框架中,同时大大降低了与产品已有代码的耦合性。



1. 一种通过扩展点和扩展实现软件快速二次开发的方法,其特征在于:
在软件中定义插件具备的扩展点,并且定义如何使用这些扩展点。
2. 如权利要求 1 所述的方法,其特征在于:
所述扩展点和扩展均通过 Schema 定义。
3. 如权利要求 1 所述的方法,其特征在于:
通过可视化工具或者手动进行 xml 配置来实现扩展点和扩展的定义。
4. 如权利要求 1 所述的方法,其特征在于:
每个扩展点可以有多个扩展实现,且每个扩展实现之间采用插件化的开发机制。
5. 如权利要求 1 所述的方法,其特征在于,在定义了扩展点和扩展后,相应的平台启动流程包括:
 - S1、模块插件注册;
 - S2、解析插件 MANIFEST 文件;
 - S3、判断是否定义了扩展点,如否则直接执行到步骤 S5;
 - S4、解析插件扩展点;
 - S5、判断是否实现了扩展,如否则直接执行到步骤 S7;
 - S6、解析插件扩展;
 - S7、解析插件依赖的其它资源;
 - S8、启动完毕。

一种通过扩展点和扩展实现软件快速二次开发的方法

技术领域

[0001] 本发明涉及计算机领域,特别涉及一种通过扩展点和扩展实现软件快速二次开发的方法。

背景技术

[0002] 在信息化的大趋势下,随着 OA 的普及,企业对 OA 应用的要求也越来越高,在实际应用中,越来越多的 OA 用户需要进行二次开发才能满足企业全面需求,这也是由管理软件的特点所决定的。原因主要归纳为两点:一方面,OA 产品是商品化软件,属于行业通用型软件,但每个企业都有其自身的特点,因此需要对原来的 OA 系统进行适当的修改,以提高 OA 系统适用性。另一方面,随着 OA 项目的实施,客户对 OA 系统有了更深的了解,应用不断深入,对 OA 系统就会提出更多的要求,例如增加很多新的功能模块,这些要求也形成了 OA 二次开发的另一来源。

[0003] 传统的 OA 产品一般会控制其源代码的开放,这样一来,如果企业要对已有的 OA 系统进行二次开发,只能继续由现有 OA 系统的开发者提供服务。更为严重的是如果在现有 OA 基础上做二次开发,则产品升级(升级的目的一般包括添加模块、增加新功能、功能升级、错误的修复、增强安全性、改善性能等)后一般是不包含用户二次开发的功能,选择升级则需要重新将二次开发的功能整合到升级的版本中。一般的选择二次开发服务的技术人员水平不是很高,设计能力有限,很多都是直接在 OA 系统的源码中进行修改,这种做法带来了两方面的不良后果,一是在产品中引入新的错误需要经过一段时间的运行后稳定下来,二是如果使用新升级的版本则需重新整合代码、代价比较高。

[0004] 随着 OA 产品技术的逐渐成熟,越来越多的 OA 厂商开始提供开源型 OA,支持源代码的开放,对企业用户来说这无疑是一大福音,借助源代码,用户可以自主进行 OA 的二次开发,增加了企业在信息化运用方面的主动性。然而不得不提的是,即使提供了源代码,OA 系统的二次开发也不是一蹴而就的事。由于 OA 系统的二次开发接口,主要针对专职程序设计人员,往往因为实施周期长、风险大、费用高影响了企业 OA 的正常应用。

[0005] OA 系统个性化功能的二次开发,首先应该在产品的体系结构上做到对修改封闭、对扩展开发,尽量避免直接在已有代码中进行修改,这样会引入新的错误。一个好的 OA 软件系统除了功能、安全性、稳定性等重要因素外,如果需要进行二次开发则其可扩展性、可集成性就显得非常重要了。

[0006] 因此,现有技术需要一种能够克服 OA 二次开发的弊端的方法。

发明内容

[0007] 鉴于现有技术中存在的问题,本发明目的在于提供一种通过扩展点和扩展实现软件快速二次开发的方法,其能够提高系统的扩展性,降低二次开发的代码与产品代码的耦合度,提高二次开发的效率。

[0008] 本发明提供了一种通过扩展点和扩展实现软件快速二次开发的方法,其特征在

于：

- [0009] 在软件中定义插件具备的扩展点,并且定义如何使用这些扩展点。
- [0010] 优选的,所述扩展点和扩展均通过 Schema 定义。
- [0011] 优选的,通过可视化工具或者手动进行 xml 配置来实现扩展点和扩展的定义。
- [0012] 优选的,每个扩展点可以有多个扩展实现,且每个扩展实现之间采用插件化的开发机制。
- [0013] 优选的,在定义了扩展点和扩展后,相应的平台启动流程包括：
 - [0014] S1、模块插件注册；
 - [0015] S2、解析插件 MANIFEST 文件；
 - [0016] S3、判断是否定义了扩展点,如否则直接执行到步骤 S5；
 - [0017] S4、解析插件扩展点；
 - [0018] S5、判断是否实现了扩展,如否则直接执行到步骤 S7；
 - [0019] S6、解析插件扩展；
 - [0020] S7、解析插件依赖的其它资源；
 - [0021] S8、启动完毕。

附图说明

- [0022] 图 1 例示了本发明实施例扩展点与扩展的关系示意图；
- [0023] 图 2 例示了本发明实施例提供的平台启动流程图。

具体实施方式

[0024] 为使本发明的上述目的、特征和优点更加明显易懂,下面结合附图和具体实施方式对本发明作进一步详细的说明。

[0025] 为了实现将新开发的功能点快速融入系统框架中,因此本发明提出了扩展点和扩展的概念。扩展点是引入了面向对象中接口的思想,就是说可以有多个扩展来实现这个接口,它定义插件具备的扩展点并且定义了如何使用这些扩展点,这里通过 Schema 定义。而扩展则是根据扩展点中定义的 Schema 来描述扩展的内容。

[0026] 优选的,本发明可通过可视化工具或者手动进行 xml 配置来实现扩展点和扩展的定义。

[0027] 图 1 例示了本发明实施例扩展点与扩展的关系示意图。如图 1 所示,可以在平台模块上定义扩展点,在业务模块上具体实现相应的扩展。每个扩展点都可以有多个扩展实现,而且每个扩展实现之间因为采用了插件化的开发机制,所以扩展实现之间是相互隔离的。

[0028] 图 2 例示了本发明实施例提供的平台启动流程图。如图 2 所示,通过在原程序中定义了扩展点和扩展,相应的平台启动流程包括：

- [0029] S1、模块插件注册；
- [0030] S2、解析插件 MANIFEST 文件；
- [0031] S3、判断是否定义了扩展点,如否则直接执行到步骤 S5；
- [0032] S4、解析插件扩展点；
- [0033] S5、判断是否实现了扩展,如否则直接执行到步骤 S7；

[0034] S6、解析插件扩展；

[0035] S7、解析插件依赖的其它资源；

[0036] S8、启动完毕。

[0037] 从图 2 可以很清楚的看到，每个模块插件在被注册到系统平台中时，系统都会试图解析其所定义的扩展点和其所实现的扩展，解析结果将被存储到系统内存中，以方便通过系统接口快速的调用。

[0038] 以下，通过一具体的扩展及扩展点应用实例，来更为清楚地理解本发明。

[0039] 通过对系统平台功能点的分析，我们在系统中定义了如下扩展点：添加快速链接、添加 js 到首页、更换首页默认加载页面、添加内容到跑马灯、添加首页状态、添加个性设置页签、自定义 portlet、添加快速操作、添加导航框架、添加日程类型、添加文档操作、添加文档处理、模块添加提醒功能等。

[0040] 在这里以添加快速链接为例对这种工作原理做一详细阐述：

[0041] 1) 在系统平台的其中一个插件模块中定义一个名为 framework_navi.headerlink 的扩展点，其对应的 Schema 中定义了 1 个 link 节点，该节点下定义了如下属性：text（链接显示的文本）、path（链接路径）、opentype（链接打开方式）、index（排序序号）；

[0042] 2) 在系统中的其中一个业务模块中实现 framework_navi.headerlink 的扩展，即对 link 节点下的属性做具体的描述，如：text="个性设置"、path="/jsp/userconfig/userconfig.jsp"、opentype="maskOpen"、index="10"；

[0043] 3) 系统平台启动时，将依次解析 framework_navi.headerlink 的扩展点和其所对应的所有扩展实现，并且在系统内存中会建立一个该扩展点与其所有扩展实现的对应关系，当其中某个业务模块被卸载后，该业务模块中所有的扩展实现也将被从系统内存中卸载掉，而且不会影响其他业务模块中的扩展实现；

[0044] 4) 在系统的快速链接区域做数据展示时就不用像传统模式开发时那样，通过硬编码指定固定的链接、或者通过读取常用的配置文件来进行展示，而是很简单的调用系统提供的接口，获取到快速链接的所有扩展实现，将它们依次按照排序序号进行展示就可以了，这样系统快速链接区域与其相关的业务模块的代码耦合性就被大大的降低。

[0045] 5) 系统平台在启动运行状态时如果需要将一个快速链接展示在快速链接区域，则只需要安装新的一个快速链接扩展实现所在的业务模块即可，系统便像平台启动流程图所示，重复解析、加载的过程，新的快速链接便很容易被展示到系统中。

[0046] 以上是对本发明的优选实施例进行的详细描述，但本领域的普通技术人员应该意识到，在本发明的范围内和精神指导下，各种改进、添加和替换都是可能的，例如使用可实现同种功能目的的算法、使用不同的编程语言（如 C、C++、Java 等）实现等。这些都在本发明的权利要求所限定的保护范围内。

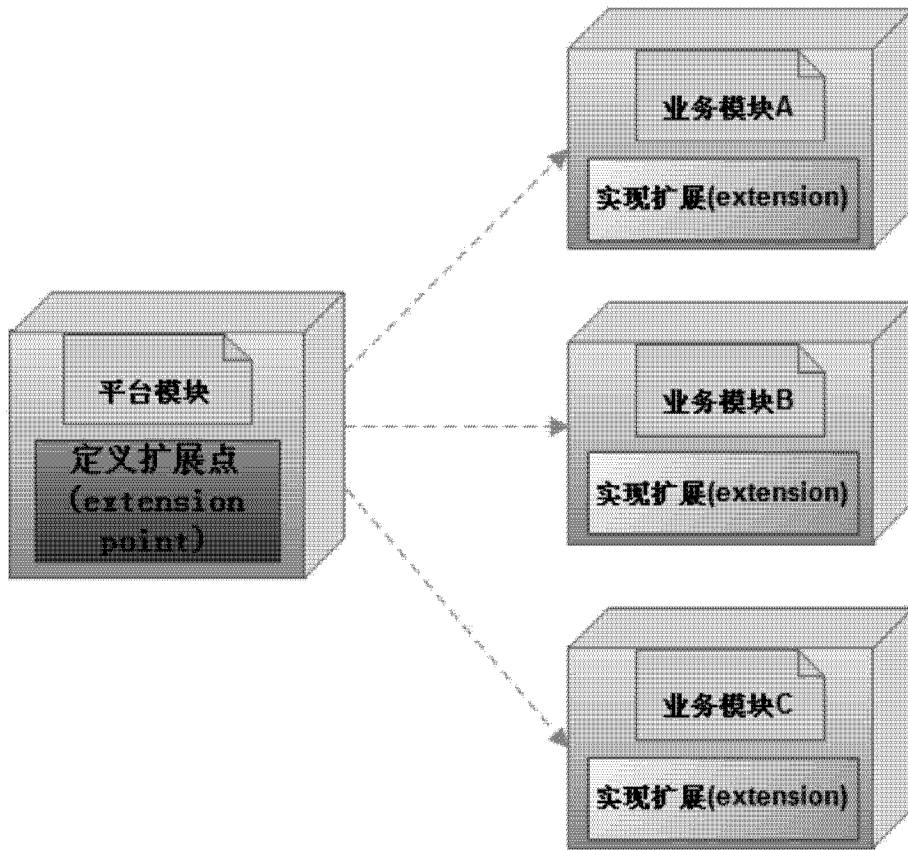


图 1

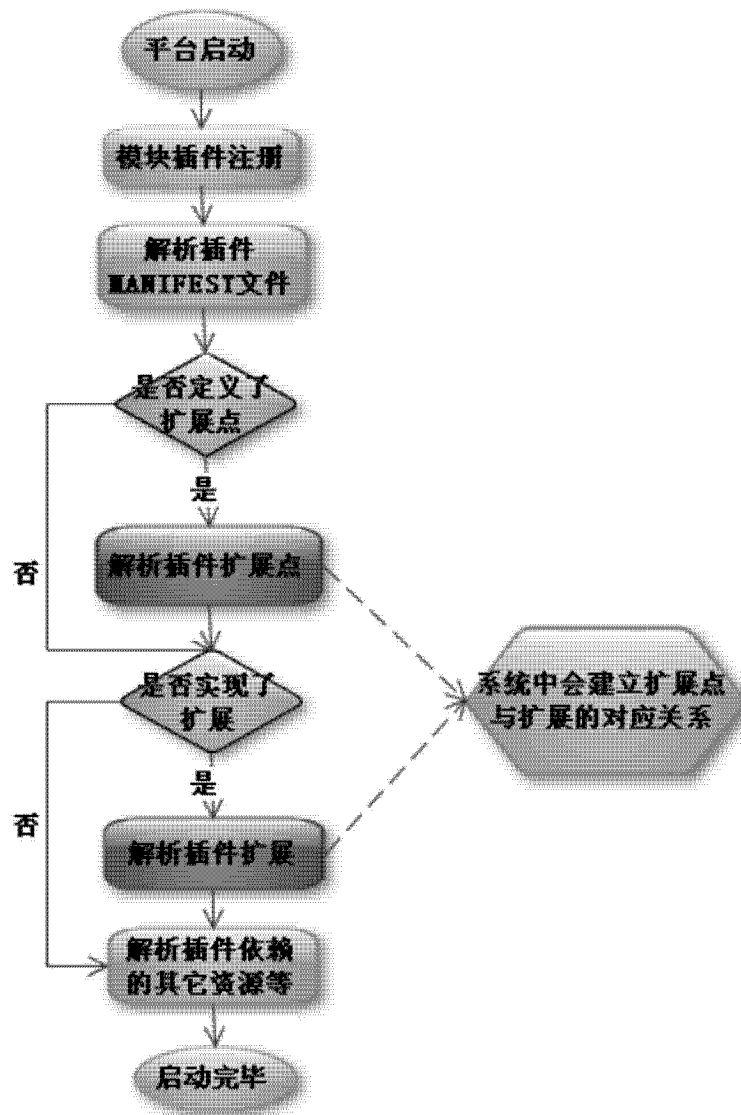


图 2