



(12) 发明专利申请

(10) 申请公布号 CN 103207902 A

(43) 申请公布日 2013. 07. 17

(21) 申请号 201310100184. 4

(22) 申请日 2013. 03. 27

(71) 申请人 交通银行股份有限公司
地址 200120 上海市银城中路 188 号

(72) 发明人 李海翔

(74) 专利代理机构 北京三友知识产权代理有限公司 11127

代理人 任默闻

(51) Int. Cl.

G06F 17/30 (2006. 01)

H04L 29/08 (2006. 01)

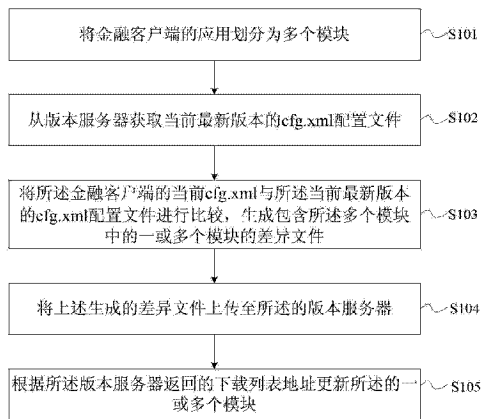
权利要求书2页 说明书6页 附图6页

(54) 发明名称

一种金融客户端数据更新方法及系统

(57) 摘要

本发明提供一种金融客户端数据更新方法及系统,该金融客户端数据更新方法包括:将金融客户端划分为多个模块;从版本服务器获取当前最新版本的cfg.xml配置文件;将所述金融客户端的当前cfg.xml与所述当前最新版本的cfg.xml配置文件进行比较,生成包含所述多个模块中的一或多个模块的差异文件;将所述的差异文件上传至所述的版本服务器;根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。本发明通过将客户端的应用划分为多个模块,根据用户的选择对需要更新的全部或部分模块进行更新,可以提高金融客户端更新的速度与效率。



1. 一种金融客户端数据更新方法,其特征在于,所述的金融客户端数据更新方法包括:

将金融客户端的应用划分为多个模块;

从版本服务器获取当前最新版本的 cfg.xml 配置文件;

将所述金融客户端的当前 cfg.xml 与所述当前最新版本的 cfg.xml 配置文件进行比较,生成包含所述多个模块中的一或多个模块的差异文件;

将所述的差异文件上传至所述的版本服务器;

根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。

2. 根据权利要求 1 所述的金融客户端数据更新方法,其特征在于,所述的将金融客户端的应用划分为多个模块,包括:根据功能将所述的金融客户端的应用划分为多个模块。

3. 根据权利要求 1 所述的金融客户端数据更新方法,其特征在于,所述从版本服务器获取当前最新版本的 cfg.xml 配置文件,包括:通过 HTTP 连接方式与所述版本服务器的地址进行交互,从版本服务器获取当前最新版本的 cfg.xml 配置文件。

4. 根据权利要求 1 所述的金融客户端数据更新方法,其特征在于,将所述金融客户端的当前 cfg.xml 与所述当前最新版本的 cfg.xml 配置文件进行比较之前,所述的方法还包括:所述的金融客户端对所述当前最新版本的 cfg.xml 配置文件进行解析。

5. 根据权利要求 1 所述的金融客户端数据更新方法,其特征在于,在将所述的差异文件上传至所述的版本服务器之前,所述的方法还包括:

所述的金融客户端将所述的差异文件展现给客户;

将所述的差异文件中用户选择更新的所述一或多个模块的对应部分差异文件上传至所述的版本服务器。

6. 一种金融客户端数据更新系统,其特征在于,所述的金融客户端数据更新系统包括:

模块划分单元,用于将金融客户端的应用划分为多个模块;

配置文件获取单元,用于从版本服务器获取当前最新版本的 cfg.xml 配置文件;

差异文件生成单元,用于将所述金融客户端的当前 cfg.xml 与所述当前最新版本的 cfg.xml 配置文件进行比较,生成包含所述多个模块中的一或多个模块的差异文件;

差异文件上传单元,用于将所述的差异文件上传至所述的版本服务器;

更新单元,用于根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。

7. 根据权利要求 6 所述的金融客户端数据更新系统,其特征在于,所述的模块划分单元根据功能将所述的金融客户端的应用划分为多个模块。

8. 根据权利要求 6 所述的金融客户端数据更新系统,其特征在于,所述配置文件获取单元具体用于:通过 HTTP 连接方式与所述版本服务器的地址进行交互,从版本服务器获取当前最新版本的 cfg.xml 配置文件。

9. 根据权利要求 6 所述的金融客户端数据更新系统,其特征在于,所述的系统还包括:配置文件解析单元,用于所述的金融客户端对所述当前最新版本的 cfg.xml 配置文件进行解析。

10. 根据权利要求 6 所述的金融客户端数据更新系统,其特征在于,所述的系统还包括:

展示单元,用于将所述的差异文件展现给客户;所述的差异文件上传单元将所述的差异文件中用户选择更新的所述一或多个模块的对应部分差异文件上传至所述的版本服务器。

一种金融客户端数据更新方法及系统

技术领域

[0001] 本发明是关于网络银行客户端技术,特别是关于一种金融客户端数据更新方法及系统。

背景技术

[0002] 随着互联网技术的发展与银行业务的拓展,网络银行的发展极大的方便了客户的使用,提高了银行的利润率,网络银行也成为各家银行争取客户的重要媒介。

[0003] 网络银行包括网上银行及手机银行等。目前的网上银行与手机银行在技术上有两种实现模式,一种是采用 IE 浏览器,另一种是采用客户端的方式。在手机银行(手机版、ipad 版)中,大多采用客户端方式对客户提供服务;在网上银行,大多采用 IE 浏览器提供服务,但对于一些专业客户也会提供金融客户端给客户提供服务。客户端的优势在于可以给客户提供更为安全,更为便捷,方便以及功能丰富的金融服务;其缺点是首次使用时需要下载客户端进行安装,而且每次客户端版本更新时,需要客户重新下载进行更新安装,给客户带来不便。

发明内容

[0004] 本发明提供一种金融客户端数据更新方法及系统,以提高客户端更新的速度与效率。

[0005] 为了实现上述目的,本发明提供一种金融客户端数据更新方法,该金融客户端数据更新方法包括:将金融客户端的应用划分为多个模块;从版本服务器获取当前最新版本的 cfg.xml 配置文件;将所述金融客户端的当前 cfg.xml 与所述当前最新版本的 cfg.xml 配置文件进行比较,生成包含所述多个模块中的一或多个模块的差异文件;将所述的差异文件上传至所述的版本服务器;根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。

[0006] 进一步地,所述的将金融客户端的应用划分为多个模块,包括:根据功能将所述的金融客户端的应用划分为多个模块。

[0007] 进一步地,所述从版本服务器获取当前最新版本的 cfg.xml 配置文件,包括:通过 HTTP 连接方式与所述版本服务器的地址进行交互,从版本服务器获取当前最新版本的 cfg.xml 配置文件。

[0008] 进一步地,将所述金融客户端的当前 cfg.xml 与所述当前最新版本的 cfg.xml 配置文件进行比较之前,所述的方法还包括:所述的金融客户端对所述当前最新版本的 cfg.xml 配置文件进行解析。

[0009] 进一步地,在将所述的差异文件上传至所述的版本服务器之前,所述的方法还包括:所述的金融客户端将所述的差异文件展现给客户;将所述的差异文件中用户选择更新的所述一或多个模块的对应部分差异文件上传至所述的版本服务器。

[0010] 为了实现上述目的,本发明还提供一种金融客户端数据更新系统,所述的金融客

户端数据更新系统包括：模块划分单元，用于将金融客户端的应用划分为多个模块；配置文件获取单元，用于从版本服务器获取当前最新版本的 cfg.xml 配置文件；差异文件生成单元，用于将所述金融客户端的当前 cfg.xml 与所述当前最新版本的 cfg.xml 配置文件进行比较，生成包含所述多个模块中的一或多个模块的差异文件；差异文件上传单元，用于将所述的差异文件上传至所述的版本服务器；更新单元，用于根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。

[0011] 进一步地，所述的模块划分单元根据功能将所述的金融客户端的应用划分为多个模块。

[0012] 进一步地，所述配置文件获取单元具体用于：通过 HTTP 连接方式与所述版本服务器的地址进行交互，从版本服务器获取当前最新版本的 cfg.xml 配置文件。

[0013] 进一步地，所述的系统还包括：配置文件解析单元，用于所述的金融客户端对所述当前最新版本的 cfg.xml 配置文件进行解析。

[0014] 进一步地，所述的系统还包括：展示单元，用于所述的金融客户端将所述的差异文件展现给客户；所述差异文件上传单元将所述差异文件中用户选择更新的所述一或多个模块的对应部分差异文件上传至所述的版本服务器。

[0015] 本发明是实施例的有益效果在于，通过将客户端的应用划分为多个模块，根据用户的选择对需要更新的全部或部分模块进行更新，可以提高金融客户端更新的速度与效率。

附图说明

[0016] 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。

[0017] 图 1 为本发明实施例金融客户端数据更新方法流程图；

[0018] 图 2 为本发明实施例 Flex 端 Main window 内容示意图；

[0019] 图 3 为本发明实施例 cfg.xml 文件的模板示意图；

[0020] 图 4 为本发明实施例当前 cfg.xml 记录文件示意图；

[0021] 图 5 为本发明实施例最新 cfg.xml 配置文件示意图；

[0022] 图 6 为本发明实施例的差异文件示意图；

[0023] 图 7 为本发明实施例金融客户端数据更新系统结构示意图。

具体实施方式

[0024] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0025] 如图 1 所示，本实施例提供一种金融客户端数据更新方法，该金融客户端数据更新方法包括：

[0026] 步骤 S101 :将金融客户端的应用划分为多个模块。

[0027] 将金融客户端的应用按照功能进行分组,以网格化的方式将金融客户端分隔为不同独立的各个独立的功能模块,功能模块包括:用户权限管理模块(登入登出等),业务功能(实际业务操作),安全保障功能(相关木马病毒清理等),用户交互模块,数据池模块,HTTP 连接模块及版本控制模块等。其中,HTTP 连接模块与版本控制模块将相互配合进行客户端的版本更新。

[0028] 该金融客户端(网银客户端)可采用 Flex, java swing, javascript 作为前端显示界面。采用主窗体来进行统一主体容器,各个功能模块在其中作为分格或分层并列显示。采用标准 MVC 结构进行统一控制。每一个模块单独由控制层根据配置文件进行显示(或不显示),并由版本更新模块进行单独的版本变更。

[0029] 步骤 S102 :从版本服务器获取当前最新版本的 cfg. xml 配置文件。

[0030] 金融客户端的当前的配置文件 cfg. xml 存在于项目路径下,该配置文件应采用标准 xml 格式,文件内容本身为 XML. XML 类,子项集合为 XMLList 类型,子项为 XML. XML 类。每一单个子项可以唯一地解析为一个 spark. components. TitleWindow 类。Flex 端 Main window 内容如图 2 所示。

[0031] cfg. xml 文件的模板如图 3 所示,其中的 config/ml. xml 等文件具体描述模块中的业务内容,将交由子模块类本身进一步解析并显示,本申请不予具体描述。

[0032] 金融客户端每次登录时,首先以 HTTP 连接方式与指定的版本服务器地址进行交互,取得存有当前最新版本的 cfg. xml 配置文件。

[0033] 步骤 S103 :将所述金融客户端的当前 cfg. xml 与所述当前最新版本的 cfg. xml 配置文件进行比较,生成包含所述多个模块中的一或多个模块的差异文件。

[0034] 将所述金融客户端的当前 cfg. xml 与所述当前最新版本的 cfg. xml 配置文件进行比较之前,金融客户端需要对所述当前最新版本的 cfg. xml 配置文件进行解析,读懂该 cfg. xml 配置文件,然后将所述金融客户端的当前 cfg. xml 与所述当前最新版本的 cfg. xml 配置文件进行比较。

[0035] 举例说明如下:

[0036] 例如,当前 cfg. xml 记录文件如图 4 所示,登录金融客户端后从版本服务器端获取的最新 cfg. xml 配置文件如图 5 所示。

[0037] 对于从版本服务器端获取的最新配置文件,首先,遍历根节点的第一层子节点,获取到模块(module),与现存的版本记录文件进行匹配后,发现现有的模块 module1, module2, module3;并且发现当前未有的模块 module4。继续遍历下级子节点,匹配后发现 module3 现有的子节点 content 与当前版本存在差异。根据上述的差异,匹配结束后将生成差异文件如图 6 所示。

[0038] 步骤 S104 :将上述生成的差异文件上传至所述的版本服务器。

[0039] 生成差异文件后,金融客户端可以对差异文件进行解释,将差异文件展现给客户;用户可以根据差异文件选择需要更新的模块,比如仅更新上述的模块 3 或模块 4,或者同时更新模块 3 及模块 4。用户选择完成后,将差异文件中需要更新的资源(可以是用户选择更新的所述一或多个模块的对应部分差异文件或者步骤 S103 中生成的差异文件)上传至版本服务器。

[0040] 步骤 S105:根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。

[0041] 版本服务器端收到差异文件后,将返回一系列 HTTP/FTP 下载列表地址以下载更新文件。例如: `http://update.bankcomm.com/version/m3_2.xml`。客户端下载更新该文件后将重新启动,将该文件在金融客户端下次启动之前替换为最新下载版本,并且将 `cfg.xml` 文件也替换为新版本。

[0042] 经过如上的操作之后,金融客户端将发生如下变化:

[0043] 独立的功能模块内相应的有更新的文件发生变更,其他模块保持不变,并且不影响到客户端体验。版本控制文件也将发生相应变更以适应下一次的下载更新,在逐一遍历并请求下载了所有的更新文件后,该版本控制文件将更新资源进行整合,加载版本控制模块,从而实现以无冗余更新数据的方式,快捷安全地对客户端进行更新。对于版本控制文件,需要说明的是,在客户端进行了一次更新后,需要将当前客户端的各文件名,版本号等信息记录在一个文件(即为版本控制文件)中。该文件的存在是为了在下次的更新时,方便快速进行客户端与服务端的比对,从而实现以最小的更新量快速更新客户端。

[0044] 各功能模块通过动态脚本实现,通过动态更新脚本来局部更新金融客户端应用,在客户每次更新金融客户端的时候,可以有选择的只更新某一局部模块,而不是整个程序下载,通过这种方式,可以提高客户端下载速度与效率,提升客户体验。

[0045] 以上的变更将仅对于文件系统内存在更新模块的部分进行修改,从下载的数据量,更新时间,客户端稳定性等方面比较旧有的整体覆盖更新方式将有极大提升。

[0046] 如图 7 所示,本实施例提供一种金融客户端数据更新系统,该金融客户端数据更新系统包括:模块划分单元 701,配置文件获取单元 702,差异文件生成单元 703,差异文件上传单元 704 及更新单元 705。

[0047] 模块划分单元 701 用于根据功能将金融客户端的应用划分为多个模块。将金融客户端的应用按照功能进行分组,以网格化的方式将金融客户端分隔为不同独立的各个独立的功能模块,功能模块包括:用户权限管理模块(登入登出等),业务功能(实际业务操作),安全保障功能(相关木马病毒清理等),用户交互模块,数据池模块,HTTP 连接模块及版本控制模块等。其中,HTTP 连接模块与版本控制模块将相互配合进行客户端的版本更新。

[0048] 该金融客户端(网银客户端)可采用 Flex, java swing, javascript 作为前端显示界面。采用主窗体来进行统一主体容器,各个功能模块在其中作为分格或分层并列显示。采用标准 MVC 结构进行统一控制。每一个模块单独由控制层根据配置文件进行显示(或不显示),并由版本更新模块进行单独的版本变更。

[0049] 配置文件获取单元 702 通过 HTTP 连接方式与所述版本服务器的地址进行交互,从版本服务器获取当前最新版本的 `cfg.xml` 配置文件。金融客户端每次登录时,配置文件获取单元 702 以 HTTP 连接方式与指定的版本服务器地址进行数据交互,取得存有当前最新版本的 `cfg.xml` 配置文件。

[0050] 差异文件生成单元 703 用于将所述金融客户端的当前 `cfg.xml` 与所述当前最新版本的 `cfg.xml` 配置文件进行比较,生成包含所述多个模块中的一或多个模块的差异文件。

[0051] 金融客户端数据更新系统还包括:配置文件解析单元 706,用于在差异文件生成单元 703 比较金融客户端的当前 `cfg.xml` 与所述当前最新版本的 `cfg.xml` 配置文件之前,对所述当前最新版本的 `cfg.xml` 配置文件进行解析。

[0052] 将所述金融客户端的当前 `cfg.xml` 与当前最新版本的 `cfg.xml` 配置文件进行比较之前,金融客户端需要对所述当前最新版本的 `cfg.xml` 配置文件进行解析,读懂该 `cfg.xml` 配置文件,然后将所述金融客户端的当前 `cfg.xml` 与所述当前最新版本的 `cfg.xml` 配置文件进行比较。

[0053] 对于从版本服务器端获取的最新配置文件,首先,遍历根节点的第一层子节点,获取到模块(module),与现存的版本记录文件进行匹配后,发现现有的模块 `module1`, `module2`, `module3`;并且发现当前未有的模块 `module4`。继续遍历下级子节点,匹配后发现 `module3` 现有的子节点 `content` 与当前版本存在差异。根据上述的差异,匹配结束后将生成差异文件,如图 6 所示。

[0054] 差异文件生成单元 703 生成差异文件之后,差异文件上传单元 704 将所述的差异文件上传至所述的版本服务器。

[0055] 可选地,金融客户端数据更新系统还可以包括:展示单元 707,用于所述的金融客户端将所述的差异文件展现给客户,用户可以根据差异文件选择需要更新的模块,比如仅更新上述的模块 3 或模块 4,或者同时更新模块 3 及模块 4。用户选择完成后,差异文件上传单元 704 将差异文件中需要更新的资源(可以是用户选择更新的所述一或多个模块的对应部分差异文件或者步骤 S103 中生成的差异文件)上传至版本服务器。

[0056] 版本服务器端收到差异文件后,将返回一系列 HTTP/FTP 下载列表地址以下载更新文件。例如:`http://update.bankcomm.com/version/m3_2.xml`。更新单元 705 根据所述版本服务器返回的下载列表地址更新所述的一或多个模块。客户端下载更新该文件后将重新启动,将该文件在金融客户端下次启动之前替换为最新下载版本,并且将 `cfg.xml` 文件也替换为新版本。

[0057] 经过如上的操作之后,金融客户端将发生如下变化:

[0058] 独立的功能模块内相应的有更新的文件发生变更,其他模块保持不变,并且不影响到客户端体验。版本控制文件也将发生相应变更以适应下一次的下载更新,在逐一遍历并请求下载了所有的更新文件后,该控件将更新资源进行整合,加载版本控制模块,从而实现以无冗余更新数据的方式,快捷安全地对客户端进行更新。

[0059] 各功能模块通过动态脚本实现,通过动态更新脚本来局部更新金融客户端应用,在客户每次更新金融客户端的时候,可以有选择的只更新某一局部模块,而不是整个程序下载,通过这种方式,可以提高客户端下载速度与效率,提升客户体验。

[0060] 以上的变更将仅对于文件系统内存在更新模块的部分进行修改,从下载的数据量,更新时间,客户端稳定性等方面比较旧有的整体覆盖更新方式将有极大提升。

[0061] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0062] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算

机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

[0063] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

[0064] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

[0065] 本发明中应用了具体实施例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

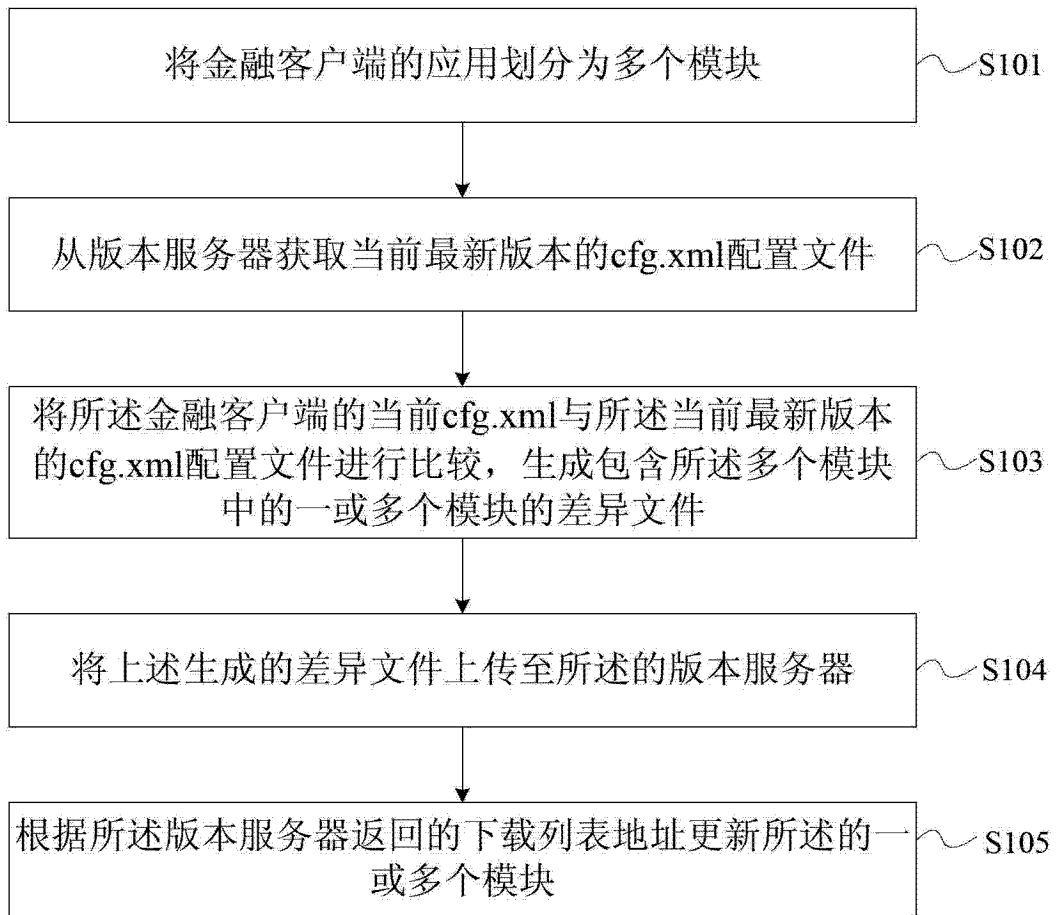


图 1

```
<?xml version="1.0" encoding="utf-8"?>
<s:TitleWindow xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" width="100%" height="100%"
    creationComplete="initHandler(event)">
    <fx:Script>
        <![CDATA[
            import mx.events.FlexEvent;

            var ul:URLLoader;

            protected function initHandler(event:FlexEvent):void
            {
                ul = new URLLoader();
                ul.load(new URLRequest("config/cfg.xml"));

                ul.addEventListener(Event.COMPLETE, ulComplete);
            }

            protected function ulComplete(event:Event):void
            {
                var xml:XML = (XML)ul.data;
                var xmlList:XMLList = xml.children();
                for (var i:int=0;i<xmlList.length();i++) {
                    var tw:TitleWindow = new TitleWindow();
                    tw.width = xmlList[i].width;
                    tw.height = xmlList[i].height;
                    tw.id = xmlList[i].name_en;
                    tw.init(xmlList[i].content);
                    grp.addElement(tw);
                }
            }
        ]]>
    </fx:Script>
    <s:Group id="grp" width="100%" height="100%"/>
</s:TitleWindow>
```

图 2

```
<?xml version="1.0" encoding="UTF-8"?>
<modules>
  <module>
    <name_en>module1</name_en>
    <name_cn>模块1</name_cn>
    <width>150</width>
    <height>150</height>
    <content>config/m1.xml</content>
  </module>
  <module>
    <name_en>module2</name_en>
    <name_cn>模块2</name_cn>
    <width>200</width>
    <height>150</height>
    <content>config/m2.xml</content>
  </module>
  <module>
    <name_en>module3</name_en>
    <name_cn>模块3</name_cn>
    <width>250</width>
    <height>300</height>
    <content>config/m3.xml</content>
  </module>
</modules>
```

图 3

```
<?xml version="1.0" encoding="UTF-8"?>
<modules>
  <module>
    <name_en>module1</name_en>
    <name_cn>模块1</name_cn>
    <width>150</width>
    <height>150</height>
    <content>config/m1.xml</content>
  </module>
  <module>
    <name_en>module2</name_en>
    <name_cn>模块2</name_cn>
    <width>200</width>
    <height>150</height>
    <content>config/m2.xml</content>
  </module>
  <module>
    <name_en>module3</name_en>
    <name_cn>模块3</name_cn>
    <width>250</width>
    <height>300</height>
    <content>config/m3.xml</content>
  </module>
</modules>
```

图 4

```
<?xml version="1.0" encoding="UTF-8"?>
<modules>
  <module>
    <name_en>module1</name_en>
    <name_cn>模块1</name_cn>
    <width>150</width>
    <height>150</height>
    <content>config/m1.xml</content>
  </module>
  <module>
    <name_en>module2</name_en>
    <name_cn>模块2</name_cn>
    <width>200</width>
    <height>150</height>
    <content>config/m2.xml</content>
  </module>
  <module>
    <name_en>module3</name_en>
    <name_cn>模块3</name_cn>
    <width>250</width>
    <height>300</height>
    <content>config/m3_2.xml</content>
  </module>
  <module>
    <name_en>module3</name_en>
    <name_cn>模块3</name_cn>
    <width>250</width>
    <height>300</height>
    <content>config/m3_2.xml</content>
  </module>
  <module>
    <name_en>module4</name_en>
    <name_cn>模块4</name_cn>
    <width>400</width>
    <height>100</height>
    <content>config/m4.xml</content>
  </module>
</modules>
```

图 5

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<modules>  
  
  <module>  
  
    <content>config/m3_2.xml</content>  
  
  </module>  
  
  <module>  
  
    <name_en>module4</name_en>  
  
    <name_cn>模块4</name_cn>  
  
    <width>400</width>  
  
    <height>100</height>  
  
    <content>config/m4.xml</content>  
  
  </module>  
</modules>
```

图 6

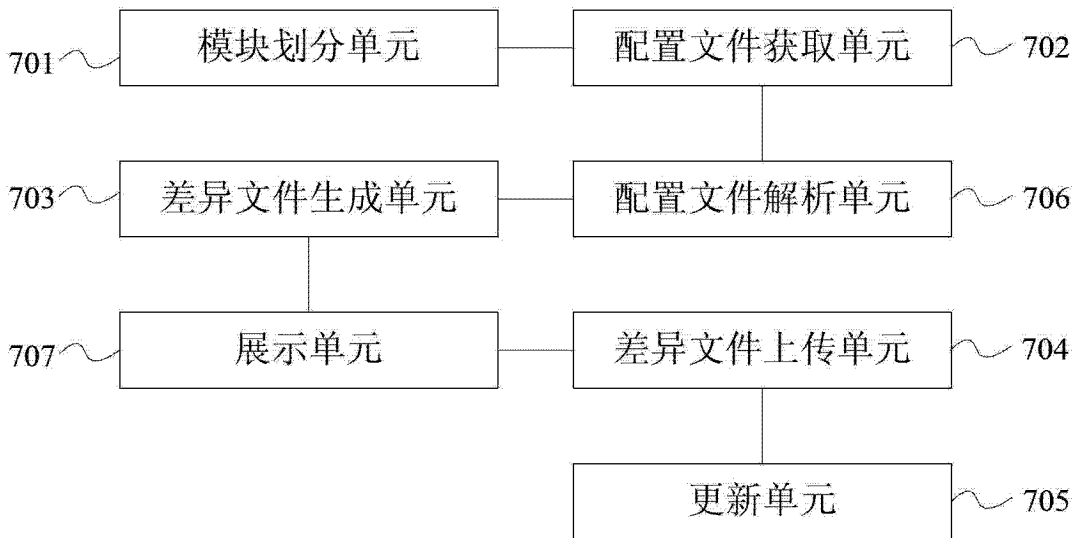


图 7