



(12) 发明专利申请

(10) 申请公布号 CN 103473034 A

(43) 申请公布日 2013. 12. 25

(21) 申请号 201210185412. 8

(22) 申请日 2012. 06. 06

(71) 申请人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四
层 847 号邮箱

(72) 发明人 熊照

(74) 专利代理机构 北京润泽恒知识产权代理有
限公司 11319
代理人 苏培华

(51) Int. Cl.
G06F 9/44 (2006. 01)
G06F 9/455 (2006. 01)

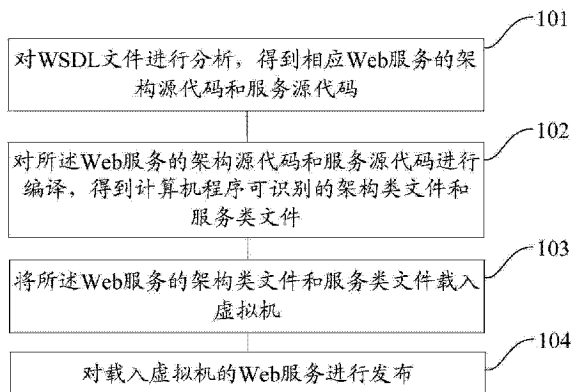
权利要求书2页 说明书7页 附图2页

(54) 发明名称

一种动态发布 Web 服务的方法和装置

(57) 摘要

本申请提供了一种动态发布 Web 服务的方法和装置,其中的方法具体包括:对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;将所述 Web 服务的架构类文件和服务类文件载入虚拟机;对载入虚拟机的 Web 服务进行发布。本申请能够降低发布耗费的人力成本,提高 Web 服务的发布效率,以及提高服务器集群的稳定性。



1. 一种动态发布 Web 服务的方法,其特征在于,包括:
对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;
对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;
将所述 Web 服务的架构类文件和服务类文件载入虚拟机;
对载入虚拟机的 Web 服务进行发布。
2. 如权利要求 1 所述的方法,其特征在于,所述对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码的步骤,进一步包括:
对所述 WSDL 文件进行解析,得到相应 Web 服务的 schema 和服务信息;
依据所述 schema 生成相应 Web 服务的架构源代码;
依据所述服务信息生成相应 Web 服务的服务源代码。
3. 如权利要求 1 所述的方法,其特征在于,所述对载入虚拟机的 Web 服务进行发布的步骤,进一步包括:
通过调用发布应用程序接口,对载入虚拟机的 Web 服务的服务功能进行发布。
4. 如权利要求 1 所述的方法,其特征在于,所述对载入虚拟机的 Web 服务进行发布的步骤,进一步包括:
依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布;所述动态脚本用于描述 Web 服务对应的业务信息。
5. 如权利要求 4 所述的方法,其特征在于,所述依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布的步骤,进一步包括:
采用所述动态脚本调用架构应用程序接口,对载入虚拟机的 Web 服务的业务功能进行发布。
6. 如权利要求 4 或 5 所述的方法,其特征在于,所述动态脚本为 Groovy 脚本或 Java 脚本。
7. 一种动态发布 Web 服务的装置,其特征在于,包括:
分析模块,用于对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;
编译模块,用于对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;
加载模块,用于将所述 Web 服务的架构类文件和服务类文件载入虚拟机;及
发布模块,用于对载入虚拟机的 Web 服务进行发布。
8. 如权利要求 7 所述的装置,其特征在于,所述分析模块进一步包括:
解析子模块,用于对所述 WSDL 文件进行解析,得到相应 Web 服务的 schema 和服务信息;
第一生成子模块,用于依据所述 schema 生成相应 Web 服务的架构源代码;
第二生成子模块,用于依据所述服务信息生成相应 Web 服务的服务源代码。
9. 如权利要求 7 所述的装置,其特征在于,所述发布模块进一步包括:
第一发布子模块,用于通过调用发布应用程序接口,对载入虚拟机的 Web 服务的服务功能进行发布。

10. 如权利要求 7 所述的装置,其特征在于,所述发布模块进一步包括:

第二发布子模块,用于依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布;所述动态脚本用于描述 Web 服务对应的业务信息。

11. 如权利要求 10 所述的装置,其特征在于,所述第二发布子模块,具体用于采用所述动态脚本调用架构应用程序接口,对载入虚拟机的 Web 服务的业务功能进行发布。

12. 如权利要求 10 或 11 所述的装置,其特征在于,所述动态脚本为 Groovy 脚本或 Java 脚本。

一种动态发布 Web 服务的方法和装置

技术领域

[0001] 本申请涉及计算机软件技术领域,特别是涉及一种动态发布 Web 服务的方法和装置。

背景技术

[0002] 由于特有的平台无关、编程语言无关等特性,针对 SOAP 协议的(简单对象访问协议,Simple Object Access Protocol)Web 服务(WebService)成为系统集成中使用非常广泛的技术,并且促进了完全不同的应用程序和平台之间的通信。

[0003] 支持 Web 服务的一个重要标准就是 WSDL (Web 服务描述语言,WebServices Description Language),WSDL 是一种用来描述 Web 服务和说明如何与 Web 服务通信的 XML (可扩展标记语言,Extensible Markup Language)语言,每个 Web 服务都配有相关的 WSDL 文件。

[0004] 例如,在网关平台和银行通信的应用场景中,如果银行提供一个 WSDL 文件,则网关平台可以根据该 WSDL 文件的描述,生成相应的 Web 服务给银行来使用。

[0005] 现有技术一种发布 Web 服务的方法流程具体可以包括:

[0006] 步骤 1、依据 WSDL 文件,采用 WSDL2Java 等转换工具生成 Java 代码;

[0007] 步骤 2、把生成的 Java 代码导入需要发布的工程中;

[0008] 步骤 3、在工程中通过相关的 Web 服务引擎框架生成定制的服务代码;

[0009] 步骤 4、发布所述服务代码。

[0010] 可以看出,上述步骤 1-步骤 4 均需要人工介入,既需要耗费大量的人力成本,又会影响发布效率。

[0011] 另外,对于大多数应用场景来说,上述方法流程需要停机执行,容易造成应用服务器的长时间停止,而大型互联网环境基本上要保证服务器的 99.99% 的可用性,应用服务器的长时间停止有可能导致流量冲击到其它的服务器上,造成服务器集群的整体死机。

[0012] 总之,需要本领域技术人员迫切解决的一个技术问题就是:如何能够降低发布耗费的人力成本,提高 Web 服务的发布效率,以及提高服务器集群的稳定性。

发明内容

[0013] 本申请所要解决的技术问题是提供一种动态发布 Web 服务的方法和装置,能够降低发布耗费的人力成本,提高 Web 服务的发布效率,以及提高服务器集群的稳定性。

[0014] 为了解决上述问题,本申请公开了一种动态发布 Web 服务的方法,包括:

[0015] 对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;

[0016] 对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;

[0017] 将所述 Web 服务的架构类文件和服务类文件载入虚拟机;

[0018] 对载入虚拟机的 Web 服务进行发布。

[0019] 优选的,所述对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码的步骤,进一步包括:

[0020] 对所述 WSDL 文件进行解析,得到相应 Web 服务的 schema 和服务信息;

[0021] 依据所述 schema 生成相应 Web 服务的架构源代码;

[0022] 依据所述服务信息生成相应 Web 服务的源代码。

[0023] 优选的,所述对载入虚拟机的 Web 服务进行发布的步骤,进一步包括:

[0024] 通过调用发布应用程序接口,对载入虚拟机的 Web 服务的功能进行发布。

[0025] 优选的,所述对载入虚拟机的 Web 服务进行发布的步骤,进一步包括:

[0026] 依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布;所述动态脚本用于描述 Web 服务对应的业务信息。

[0027] 优选的,所述依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布的步骤,进一步包括:

[0028] 采用所述动态脚本调用架构应用程序接口,对载入虚拟机的 Web 服务的业务功能进行发布。

[0029] 优选的,所述动态脚本为 Groovy 脚本或 Java 脚本。

[0030] 另一方面,本申请还公开了一种动态发布 Web 服务的装置,包括:

[0031] 分析模块,用于对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;

[0032] 编译模块,用于对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;

[0033] 加载模块,用于将所述 Web 服务的架构类文件和服务类文件载入虚拟机;及

[0034] 发布模块,用于对载入虚拟机的 Web 服务进行发布。

[0035] 优选的,所述分析模块进一步包括:

[0036] 解析子模块,用于对所述 WSDL 文件进行解析,得到相应 Web 服务的 schema 和服务信息;

[0037] 第一生成子模块,用于依据所述 schema 生成相应 Web 服务的架构源代码;

[0038] 第二生成子模块,用于依据所述服务信息生成相应 Web 服务的源代码。

[0039] 优选的,所述发布模块进一步包括:

[0040] 第一发布子模块,用于通过调用发布应用程序接口,对载入虚拟机的 Web 服务的功能进行发布。

[0041] 优选的,所述发布模块进一步包括:

[0042] 第二发布子模块,用于依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布;所述动态脚本用于描述 Web 服务对应的业务信息。

[0043] 优选的,所述第二发布子模块,具体用于采用所述动态脚本调用架构应用程序接口,对载入虚拟机的 Web 服务的业务功能进行发布。

[0044] 优选的,所述动态脚本为 Groovy 脚本或 Java 脚本

[0045] 与现有技术相比,本申请具有以下优点:

[0046] 本申请动态发布 Web 服务的流程具体可以包括:依据 WSDL 文件自动生成源代码—将源代码编译成为计算机程序可识别的类文件—将类文件动态加载至虚拟机—发布载入

虚拟机的 Web 服务。一方面,所述动态发布 Web 服务的流程无需人工介入,故能够降低发布耗费的人力成本,提高 Web 服务的发布效率;另一方面,所述动态发布 Web 服务的流程无需停机无需重启系统即可热部署运行,因此,能够有效避免停机发布导致的服务器集群的整体死机等现象,从而能够提高服务器集群的稳定性。

附图说明

- [0047] 图 1 是本申请一种动态发布 Web 服务的方法实施例 1 的流程图;
- [0048] 图 2 是本申请一种动态发布 Web 服务的方法实施例 2 的流程图;
- [0049] 图 3 是本申请一种动态发布 Web 服务的应用示例;
- [0050] 图 4 是本申请一种动态发布 Web 服务的装置实施例的结构图。

具体实施方式

[0051] 为使本申请的上述目的、特征和优点能够更加明显易懂,下面结合附图和具体实施方式对本申请作进一步详细的说明。

[0052] 本申请提供了一种动态发布 Web 服务的方案。一方面,相对于现有技术中依赖人工介入的方式执行生成 Java 代码—Java 代码导入工程—生成服务代码—部署发布的操作流程,本申请的方案自动执行整个发布的操作流程,故能够降低发布耗费的人力成本,提高 Web 服务的发布效率;另一方面,现有技术中发布的操作流程需要停机执行,容易造成应用服务器的长时间停止,而本申请的方案可以依据 WSDL 文件进行相应 Web 服务的动态发布,由于动态发布无需停机无需重启系统即可热部署运行,因此,能够有效避免停机发布导致的服务器集群的整体死机等现象,从而能够提高服务器集群的稳定性。

[0053] 参照图 1,示出了本申请一种动态发布 Web 服务的方法实施例 1 的流程图,具体可以包括:

[0054] 步骤 101、对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;

[0055] 步骤 102、对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;

[0056] 步骤 103、将所述 Web 服务的架构类文件和服务类文件载入虚拟机;

[0057] 步骤 104、对载入虚拟机的 Web 服务进行发布。

[0058] WSDL 文件是一种用来描述 Web 服务和说明如何与 Web 服务通信的 XML,WSDL 文件中通常记录有两大块内容:Web 服务的 schema (架构)和服务信息,其中,Web 服务的 schema 通常定义有 Web 服务能够发送和接收的消息种类,以及使用方必须提供给 Web 服务的数据,以便该 Web 服务能够执行;Web 服务的服务信息通常可以包括 Web 服务的功能,Web 服务在 Web 上的位置,使用 Web 服务的指令等信息。

[0059] 在本申请的一种优选实施例中,所述对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码的步骤,可以进一步包括:

[0060] 步骤 A1、对所述 WSDL 文件进行解析,得到相应 Web 服务的 schema 和服务信息;

[0061] 步骤 A2、依据所述 schema 生成相应 Web 服务的架构源代码;

[0062] 步骤 A3、依据所述服务信息生成相应 Web 服务的服务源代码。

[0063] 在本申请的一种应用示例中,所述架构源代码可以包括 POJO (简单的 Java 对象,

Plain Old Java Objects) 的 javacode, 此时 schema 中可以记录有 POJO 的描述信息。

[0064] 在本申请的一种应用示例中, schema 中可以记录有 POJO 的描述信息, 此时生成的架构源代码中可以包括 POJO(简单的 Java 对象, Plain Old Java Objects) 的 javacode。

[0065] 在本申请的另一种应用示例中, 所述服务信息可以包括命名空间(namingspace), 服务名称(serviceName), 属性名称(attributeName) 等。

[0066] 在具体实现中, 可以采用 WSDL4j 等工具进行 WSDL 文件的解析。在实际应用中, 可以将 Web 服务的架构源代码和服务源代码存储于预设的目录下。

[0067] 另外, 可以采用 axis (阿帕奇可扩展交互系统, Apache Extensible Interaction System) 的生成器来生成步骤 A2 和步骤 A3 的源代码。但是, 由于 axis 生成的源代码会包括大量的服务, 并且 axis 通常自带大量的冗余代码, 而这些大量的服务和冗余代码容易造成步骤 102 编译时间以及步骤 103 加载时间的加长。

[0068] 在本申请的一种优选实施例中, 可以采用 XFire 的生成器来生成步骤 A2 和步骤 A3 的源代码。XFire 是新一代的 Java Web 服务引擎, XFire 使得在 JavaEE (Java 平台企业版, Java Platform Enterprise Edition) 应用中发布 Web 服务变得轻而易举。和 axis 等其他 Web 服务引擎相比, XFire 的配置非常简单, 其生成的源代码不会包括大量的服务, 且不会自带大量的冗余代码, 故相对于 axis, 能够缩短步骤 102 的编译时间以及步骤 103 的加载时间。

[0069] 在计算机软件领域, 源代码是指一系列人类可读的计算机语言指令。在现代程序语言中, 源代码可以是以书籍或者磁带的形式出现, 但最为常用的格式是文本文件, 这种典型格式的目的是为了编译出计算机程序。源代码的最终目的是将人类可读的文本翻译成为计算机可以执行的二进制指令, 这种过程叫做编译, 可以通过编译器完成。

[0070] 在本申请的一种应用示例中, 可以采用 JDT (Java 开发环境, Java development tooling)、suntool (SUN 公司的开发工具) 等作为编译器, 将所述 Web 服务的架构源代码和服务源代码编译成为计算机程序可识别的架构类(Class) 文件和服务类文件。其中, 作为一种轻量级的 Eclipse (一个开放源代码的、基于 Java 的可扩展开发平台) 插件, JDT 能够实现即时编译, 缩短步骤 102 的编译时间。在实际应用中, 可以将编译得到的计算机程序可识别的架构类文件和服务类文件存储于预设的目录下, 架构类文件的一种典型示例是 POJO 类文件。

[0071] 在本申请的一种应用示例中, 可以采用类加载器(ClassLoader) 将所述 Web 服务的架构类文件和服务类文件载入虚拟机。以 JVM (Java 虚拟机, Java Virtual Machine) 为例, Class Loader 的目的是动态加载 Java 类文件和 Resource (资源), 这里的 Java 类是平台无关的, 标准的, 具有规范二进制文件格式的, 可由编译器生成, 且可以被任何一种 JVM 加载。Java 类的表现形式不仅只有类文件, 还可以为内存缓存, 或是网络数据流。

[0072] 综上, 本申请动态发布 Web 服务的流程具体可以包括: 依据 WSDL 文件自动生成源代码—将源代码编译成为计算机程序可识别的类文件—将类文件动态加载至虚拟机—发布载入虚拟机的 Web 服务。一方面, 所述动态发布 Web 服务的流程无需人工介入, 故能够降低发布耗费的人力成本, 提高 Web 服务的发布效率; 另一方面, 所述动态发布 Web 服务的流程无需停机无需重启系统即可热部署运行, 因此, 能够有效避免停机发布导致的服务器集群的整体死机等现象, 从而能够提高服务器集群的稳定性。

[0073] 在本申请的一种优选实施例中,所述对载入虚拟机的 Web 服务进行发布的步骤,可以进一步包括:通过调用发布应用程序接口,对载入虚拟机的 Web 服务的服务功能进行发布。

[0074] 在将所述 Web 服务的架构类文件和服务类文件载入虚拟机后,本领域技术人员可以选择对相应的 Web 服务发布或者不发布。

[0075] 在实际应用中,可以通过后台操作界面之类的点击等触发操作,触发所述对载入虚拟机的 Web 服务进行发布的操作,其中,本优选实施例采用发布 API (应用程序接口, Application Programming Interface) 作为触发操作和发布操作之间的桥梁。

[0076] 在本申请的一种应用示例中,发布 API 函数可以为 pubService(String serviceName),调用该发布 API 函数即可触发接口中某个服务名(serviceName)。

[0077] 参照图 2,示出了本申请一种动态发布 Web 服务的方法实施例 2 的流程图,具体可以包括:

[0078] 步骤 201、对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;

[0079] 步骤 202、对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;

[0080] 步骤 203、将所述 Web 服务的架构类文件和服务类文件载入虚拟机;

[0081] 步骤 204、依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布;所述动态脚本可用于描述 Web 服务对应的业务信息。

[0082] 如果 Web 服务被业务系统使用,则现有技术中,开发人员需要在工程中编写符合发送/接收 SOAP 协议的业务代码,并将业务代码和服务代码一起发布。业务代码的编写又会耗费大量的人力成本,影响 Web 服务的发布效率;并且,停机发布容易造成很多业务中断,而长时间的业务中断不仅带来一些业务损失,甚至可能造成一些资金损失。

[0083] 具体到本申请动态发布 Web 服务的流程,则摆在本领域技术人员面前的的问题是:在某 Web 服务被发布后,该 Web 服务接收的数据如何完成业务功能?

[0084] 针对上述问题,实施例 2 将业务功能放在动态脚本中,具体而言,使用动态脚本来描述 Web 服务对应的业务信息,所述动态脚本可由开发人员依据业务信息编写,具体可以采用 Groovy 脚本或 Java 脚本进行编写。其中,Groovy 是一种新兴的 Java2 平台语言,由于其可以与 Java 代码很好的结合,可以用 Groovy 在 Java 平台上进行 Java 编程,因此 Groovy 被广泛的应用于各个领域。在实际中,可以将动态脚本存储在数据库或者是固定的目录下。

[0085] 参照图 3,示出了本申请一种动态发布 Web 服务的应用示例,其中,后台系统 301 从业务系统 302 的载入虚拟机的 Web 服务中选择一个,触发相应的发布操作,业务系统 302 依据该触发选择相应的动态脚本 303,并执行该 Web 服务的发布操作。

[0086] 在本申请的一种优选实施例中,所述依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布的步骤,可以进一步包括:采用所述动态脚本调用架构 API (应用程序接口),对载入虚拟机的 Web 服务的业务功能进行发布。

[0087] 在实际中,可以依据动态脚本中描述的业务信息设置架构 API。例如动态脚本中的业务信息表明,需要调用虚拟机中的 POJO 类文件,则可以使用 API Object getPOJO(String classpath) 作为架构 API,这里的 classpath 用于表示 POJO 类文件的目录或路径。假设有个 Hello.class 文件在 D 盘,则相应的 classpath 可以为 java classpath D:\Hello.

class。

[0088] 需要说明的是,在业务信息需要变更时,现有技术需要重复改动业务代码,然后重新发布,既耗费人力成本,又会影响发布效率。

[0089] 本申请的动态脚本主要是用于接收到服务后消息的处理的,使用本申请,只需更改动态脚本即可刷新相应的 Web 服务,因此,本申请的动态脚本能够随着业务信息的变更而灵活改动。例如,如果银行消息体发生变动,可以将老的动态脚本删除,使用新的动态脚本,这个过程只需配置即可,不需要做任何代码修改。

[0090] 另外,本申请可以根据需求选择使用或不使用脚本。例如,可以将动态脚本加载到 spring 上下文,使用时放在本地缓存,不需要时从缓存中删除即可。

[0091] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中未详述的部分,可以参见其他实施例的相关描述即可。

[0092] 并且,本领域技术人员易于想到的是:上述实施例 1—实施例 2 任意组合应用都是可行的,故上述实施例 1—实施例 2 之间的任意组合都是本发明的实施方案,但是由于篇幅限制,本说明书在此就不一一详述了。

[0093] 与前述方法实施例相应,本申请还公开了一种动态发布 Web 服务的装置实施例,参照图 4 所示的结构图,具体可以包括:

[0094] 分析模块 401,用于对 WSDL 文件进行分析,得到相应 Web 服务的架构源代码和服务源代码;

[0095] 编译模块 402,用于对所述 Web 服务的架构源代码和服务源代码进行编译,得到计算机程序可识别的架构类文件和服务类文件;

[0096] 加载模块 403,用于将所述 Web 服务的架构类文件和服务类文件载入虚拟机;及

[0097] 发布模块 404,用于对载入虚拟机的 Web 服务进行发布。

[0098] 在本申请的一种优选实施例中,所述分析模块 401 可以进一步包括:

[0099] 解析子模块,用于对所述 WSDL 文件进行解析,得到相应 Web 服务的 schema 和服务信息;

[0100] 第一生成子模块,用于依据所述 schema 生成相应 Web 服务的架构源代码;

[0101] 第二生成子模块,用于依据所述服务信息生成相应 Web 服务的源代码。

[0102] 在本申请的另一种优选实施例中,所述发布模块 404 可以进一步包括:

[0103] 第一发布子模块,用于通过调用发布应用程序接口,对载入虚拟机的 Web 服务的业务功能进行发布。

[0104] 在本申请的再一种优选实施例中,所述发布模块 404 可以进一步包括:

[0105] 第二发布子模块,用于依据动态脚本对载入虚拟机的 Web 服务的业务功能进行发布;所述动态脚本用于描述 Web 服务对应的业务信息。

[0106] 在本申请的一种优选实施例中,所述第二发布子模块,可以具体用于采用所述动态脚本调用架构应用程序接口,对载入虚拟机的 Web 服务的业务功能进行发布。

[0107] 在本申请实施例中,优选的是,所述动态脚本可以为 Groovy 脚本或 Java 脚本。

[0108] 对于装置实施例而言,由于其与方法实施例基本相似,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0109] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序

产品。因此,本申请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,本申请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。

[0110] 本申请是参照根据本申请实施例的方法、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。

[0111] 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。

[0112] 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。

[0113] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0114] 本说明书中的各个实施例均采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似的部分互相参见即可。

[0115] 以上对本申请所提供的一种动态发布 Web 服务的方法和装置,进行了详细介绍,本文中应用了具体个例对本申请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申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,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申请的限制。

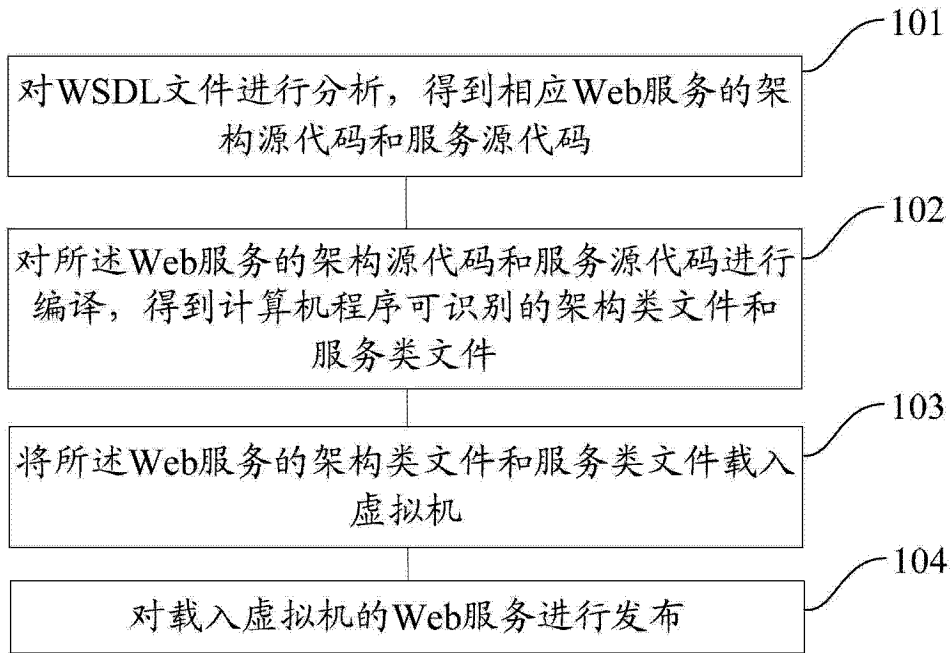


图 1

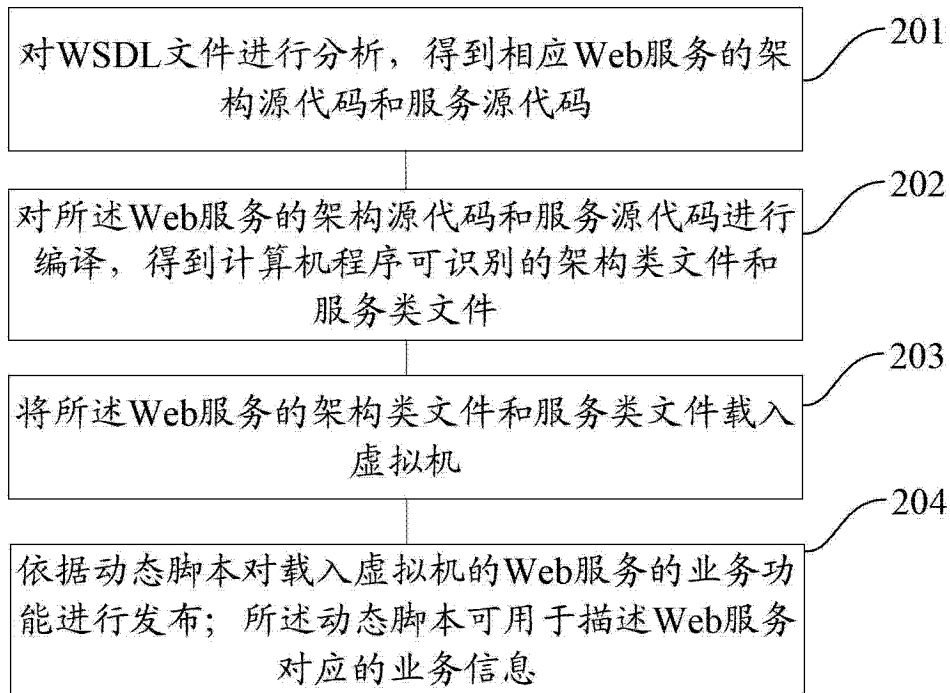


图 2

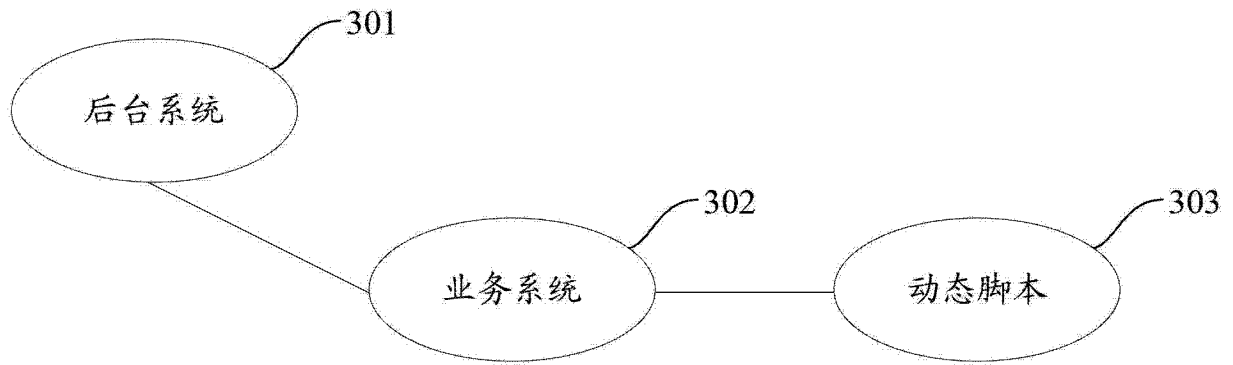


图 3

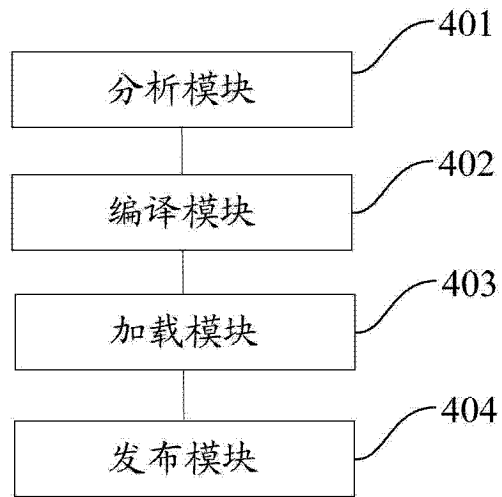


图 4