



(12)发明专利

(10)授权公告号 CN 105373401 B

(45)授权公告日 2018.09.07

(21)申请号 201510706186.7

(22)申请日 2015.10.27

(65)同一申请的已公布的文献号
申请公布号 CN 105373401 A

(43)申请公布日 2016.03.02

(73)专利权人 上海找钢网信息科技股份有限公司

地址 201815 上海市嘉定区嘉定工业区汇源路55号6幢2层E区208室

(72)发明人 孙维杰

(74)专利代理机构 上海翼胜专利商标事务所
(普通合伙) 31218

代理人 翟羽

(51)Int. Cl.

G06F 8/60(2018.01)

(56)对比文件

CN 1798153 A,2006.07.05,

CN 103019732 A,2013.04.03,

CN 103793217 A,2014.05.14,

CN 101453367 A,2009.06.10,

CN 104765620 A,2015.07.08,

US 2002066073 A1,2002.05.30,

杨敏强.利用IBM UrbanCode Deploy实现企业应用部署自动化.《https://www.ibm.com/developerworks/cn/rational/1401_yangmq_urbancode/》.2014,

审查员 唐丹颖

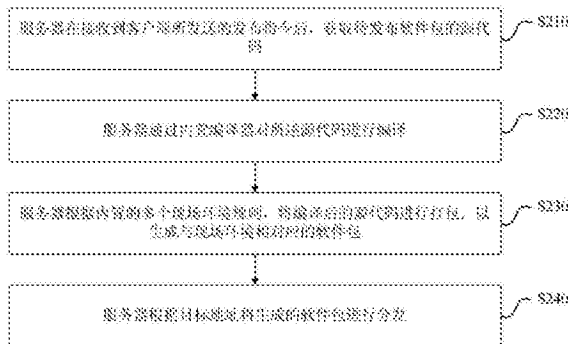
权利要求书3页 说明书8页 附图5页

(54)发明名称

软件包自动发布方法和系统

(57)摘要

本发明提供了一种软件包自动发布方法和系统,所述方法包括以下步骤:(1)服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码;(2)服务器通过内置编译器对所述源代码进行编译;(3)服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包;(4)服务器根据目标地址将生成的软件包进行分发。本发明不仅能够建立重复、可靠的发布流程,以提高软件包的发布质量;而且能够将整个发布流程自动化,提升整个软件包的发布效率;再者,其能够使多个开发人员同时参与,以增加软件开发的透明度。



1. 一种软件包自动发布方法,其特征在于,包括:
 - (1) 服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码;
 - (2) 服务器通过内置编译器对所述源代码进行编译;
 - (3) 服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包;
 - (4) 服务器根据目标地址将生成的软件包进行分发;在步骤(1)中进一步包括:
 - (11) 服务器提供一配置点,以确定现场环境中对应的目标服务器;
 - (12) 服务器提供一目标需求列表,以选择与目标需求相对应的代码包,进而确定与目标需求相对应的现场环境。
2. 根据权利要求1中所述的软件包自动发布方法,其特征在于,在步骤(3)中进一步包括:
 - (31) 服务器调用所有现场环境配置的模板库;
 - (32) 服务器通过一多环境配置文件生成工具并根据与目标需求相对应的代码包,在所有现场环境配置的模板库中查找与现场环境相对应的配置文件数据;
 - (33) 在所有现场环境配置的模板库中成功提取到相对应的配置文件数据之后,将所提取的配置文件数据并入至配置文件模板,以生成相应的配置文件;
 - (34) 将所生成的配置文件添加至编译后的源代码之中,以执行后继的打包操作。
3. 根据权利要求1中所述的软件包自动发布方法,其特征在于,在步骤(4)中进一步包括:
 - (41) 配置软件包自动发布的发布策略;
 - (42) 根据发布策略提供相应的审批权限。
4. 根据权利要求1中所述的软件包自动发布方法,其特征在于,在步骤(4)中进一步包括:
 - (51) 配置软件包自动发布的发布策略,以设定多个软件包之间的顺序;
 - (52) 根据所设定的多个软件包之间的顺序,执行多个软件包一次性同时发布的操作。
5. 根据权利要求1中所述的软件包自动发布方法,其特征在于,在步骤(4)中进一步包括:
 - (61) 配置软件包自动发布的发布策略,以设定软件包的版本信息;
 - (62) 当发现当前软件包内容存有错误时,根据发布策略中所设定的软件包的版本信息,查找到当前软件包的前一次版本所对应的软件包内容;
 - (63) 执行回退操作,以将前一次版本所对应的软件包内容替换当前软件包内容。
6. 根据权利要求1中所述的软件包自动发布方法,其特征在于,在步骤(4)中进一步包括:
 - (71) 提供一服务器管理界面,以对服务器的配置进行监控。
7. 根据权利要求2中所述的软件包自动发布方法,其特征在于,在步骤(3)中,进一步包括:
 - (81) 基于所生成的配置文件,添加伪数据信息至编译后的源代码,以执行后继的打包操作;

(82) 当打包至一设定现场环境相对应的软件包时,在配置文件中匹配所述伪数据信息;

(83) 若匹配到所述伪数据信息,则在敏感信息管理数据库中提取与伪数据信息相对应的敏感数据信息,并将所述伪数据信息替换为所述敏感数据信息。

8. 一种软件包自动发布系统,其特征在于,包括:

源代码获取单元,用以服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码;

源代码编译单元,用以服务器通过内置编译器对所述源代码进行编译;

打包单元,用以服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包;

软件包分发单元,用以服务器根据目标地址将生成的软件包进行分发;

在所述源代码获取单元中,进一步包括:

配置点提供模块,用以服务器提供一配置点,以确定现场环境中对应的目标服务器;

目标需求列表提供模块,用以服务器提供一目标需求列表,以选择与目标需求相对应的代码包,进而确定与目标需求相对应的现场环境。

9. 根据权利要求8所述的软件包自动发布系统,其特征在于,所述打包单元进一步包括:

模板库调用模块,用以服务器调用所有现场环境配置的模板库;

配置文件数据查找模块,用以服务器通过一多环境配置文件生成工具并根据与目标需求相对应的代码包,在所有现场环境配置的模板库中查找与现场环境相对应的配置文件数据;

配置文件生成模块,用以在所有现场环境配置的模板库中成功提取到相对应的配置文件数据之后,将所提取的配置文件数据并入至配置文件模板,以生成相应的配置文件;

配置文件添加模块,用以将所生成的配置文件添加至编译后的源代码之中,以执行后继的打包操作。

10. 根据权利要求8所述的软件包自动发布系统,其特征在于,所述软件包分发单元进一步包括:

发布策略配置模块,用以配置软件包自动发布的发布策略;

审批权限提供模块,用以根据发布策略提供相应的审批权限。

11. 根据权利要求8所述的软件包自动发布系统,其特征在于,所述软件包分发单元进一步包括:

发布策略配置模块,用以配置软件包自动发布的发布策略,以设定多个软件包之间的顺序;多软件包发布模块,用以根据所设定的多个软件包之间的顺序,执行多个软件包一次性同时发布的操作。

12. 根据权利要求8所述的软件包自动发布系统,其特征在于,所述软件包分发单元进一步包括:

发布策略配置模块,用以配置软件包自动发布的发布策略,以设定软件包的版本信息;

软件包内容查找模块,用以当发现当前软件包内容存有错误时,根据发布策略中所设定的软件包的版本信息,查找到当前软件包的前一次版本所对应的软件包内容;

回退操作执行模块,用以执行回退操作,以将前一次版本所对应的软件包内容替换当前软件包内容。

13.根据权利要求8所述的软件包自动发布系统,其特征在于,所述软件包分发单元进一步包括:

服务器配置监控模块,用以提供一服务器管理界面,以对服务器的配置进行监控。

14.根据权利要求9所述的软件包自动发布系统,其特征在于,所述打包单元进一步包括:

伪数据信息添加模块,用以基于所生成的配置文件,添加伪数据信息至编译后的源代码,以执行后继的打包操作;

伪数据信息匹配模块,用以当打包至一设定现场环境相对应的软件包时,在配置文件中匹配所述伪数据信息;

敏感数据信息替换模块,用以若匹配到所述伪数据信息,则在敏感信息管理数据库中提取与伪数据信息相对应的敏感数据信息,并将所述伪数据信息替换为所述敏感数据信息。

软件包自动发布方法和系统

技术领域

[0001] 本发明涉及计算机领域,尤其涉及一种软件包自动发布方法和系统。

背景技术

[0002] 随着软件需求的增大,技术的发展,以及人们对软件的应用或者产品的特定设置的要求的提高,各种应用或产品或软件功能的完善不断推陈出新。现有技术中,对应用或产品或软件的研发需要开发人员在本地开发并编译,然后远程登入服务器,并且将编译后的软件包上传至服务器,同时根据不同的现场环境规则,在服务器上修改与现场环境相对应的配置文件,接着根据目的地址而执行分发操作,以完成发布任务。例如,参见图1所示:1) 开发本地生成发布包;2) 登录服务器;3) 上传发布包,并修改配置文件;4) 操作IIS。

[0003] 由上述过程能够看出,对应用或产品或软件的发布或更新需要花费大量人力以及时间。因人工发布软件包占时较长,且不稳定性高,容易造成人为失误,且效率较低,进而使得用户无法更加快速地使用更新后的应用或产品或软件,降低用户体验。再者,整个软件包发布过程不具有透明性,且不适合多人合作环境。

[0004] 因此,亟需提供一种软件包自动发布方法及系统。

发明内容

[0005] 为了解决上述问题,本发明提供一种软件包自动发布方法及系统,其不仅能够建立重复、可靠的发布流程,以提高软件包发布的布质量;而且能够将整个发布流程自动化,提升整个软件包的发布效率;再者,其能够使多个开发人员同时参与,以增加软件开发的透明度。

[0006] 依据本发明的一方面,本发明提供了一种软件包自动发布方法,其包括:(1) 服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码;(2) 服务器通过内置编译器对所述源代码进行编译;(3) 服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包;(4) 服务器根据目标地址将生成的软件包进行分发。

[0007] 依据本发明的另一方面,本发明还提供了一种软件包自动发布系统,其包括:源代码获取单元,用以服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码;源代码编译单元,用以服务器通过内置编译器对所述源代码进行编译;打包单元,用以服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包;软件包分发单元,用以服务器根据目标地址将生成的软件包进行分发。

[0008] 本发明的优点在于,通过采用本发明所述的软件包自动发布方法及系统,其能够建立重复、可靠的发布流程,以提高软件包的发布质量,进而避免人为失误,而且能够将整个发布流程自动化,提升整个软件包的发布效率,进而减少相关人员浪费时间在毫无技术含量的重复劳动之中,再者,其能够使多个开发人员同时参与,以增加软件开发的透明度,进而避免团队中的其他人员无法及时得知整个发布进程。

附图说明

[0009] 根据下文结合附图对本发明具体实施方式的详细描述,本领域技术人员将会更加明白本发明上述以及其他目的、优点和特征。

[0010] 通过阅读下文具体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。附图仅用于示出具体实施方式的目的,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。

[0011] 图1是现有技术中软件包发布的步骤流程图;

[0012] 图2是本发明一实施方式中软件包自动发布方法的步骤流程图;

[0013] 图3是本发明一实施方式中软件包自动发布系统的框架示意图;

[0014] 图4是本发明一实施方式中源代码获取单元的结构示意图;

[0015] 图5是本发明一实施方式中打包单元的结构示意图;

[0016] 图6是本发明一实施方式中软件包分发单元的结构示意图;

[0017] 图7是本发明另一实施方式中软件包分发单元的结构示意图;

[0018] 图8是本发明另一实施方式中软件包分发单元的结构示意图;

[0019] 图9是本发明另一实施方式中软件包分发单元的结构示意图;

[0020] 图10是本发明另一实施方式中打包单元的结构示意图。

具体实施方式

[0021] 下面将参考附图更详细地描述本发明的示例性实施例。虽然附图中显示了本发明的示例性实施例,然而应当理解,可以以各种形式实现本发明而不应所阐述的实施例限制。相反,提供这些实施例是为了能够更透彻地理解本发明,并且能够将本发明的范围完整地传达给本领域的技术人员。

[0022] 参见图2所示,本发明提供一种软件包自动发布方法,该方法包括以下步骤:

[0023] 步骤S210:服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码。

[0024] 在此步骤中,待发布软件包的源代码可以存储在开发人员的本地设备中或者存储在一集中式管理的源代码管理服务器(如TFS模式的服务器)中,且不限于此。开发人员或测试人员或其他相关人员可以通过客户端或前台所提供的(网站)站点,选择待发送的软件包及其相应的现场环境、相关参数等,以执行发布指令。服务器在接收到开发人员或测试人员或其他相关人员的发布请求之后,可以根据软件包的名称、版本等信息调用并获取相应软件包的源代码。在本实施方式中,可以进一步包括以下步骤:(11)服务器提供一配置点,以确定现场环境中对应的目标服务器。(12)服务器提供一目标需求列表,以选择与目标需求相对应的代码包,进而确定与目标需求相对应的现场环境。其中,在步骤(11)中,通过一配置点,可以确定待发送的代码包所对应的现场环境(例如开发环境、测试环境、生产环境等),每一现场环境对应于相应的目标服务器(包括目标服务器的数量等)。在步骤(12)中,当选择与目标需求相对应的代码包(即程序)时,可以确定该软件包将发送到相应的现场环境(例如测试环境、生产环境)。

[0025] 步骤S220:服务器通过内置编译器对所述源代码进行编译。

[0026] 在此步骤中,服务器可以通过内置的编译器,例如C、PHP、Python、Java等多种语言的编译器进行编译。该编译器可以为本领域技术人员所常用的编译器,在此不再一一列举。

[0027] 步骤S230:服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包。

[0028] 在编译完代码后,根据不同的配置文件而产生不同的软件包。

[0029] 在本实施方式中,可以进一步包括以下步骤:(31)服务器调用所有现场环境配置的模板库;(32)服务器通过一多环境配置文件生成工具并根据与目标需求相对应的代码包,在所有现场环境配置的模板库中查找与现场环境相对应的配置文件数据;(33)在所有现场环境配置的模板库中成功提取到相对应的配置文件数据之后,将所提取的配置文件数据并入至配置文件模板,以生成相应的配置文件;(34)将所生成的配置文件添加至编译后的源代码之中,以执行后继的打包操作。

[0030] 其中,在步骤(31)中,所有现场环境配置(例如测试环境、生产环境等)的模板库可以存储于一文件之中,该文件类似于文本文件。

[0031] 在步骤(32)中,当开发人员或测试人员或其他相关人员在选择待发布软件包时,就已经自动确认了相应的环境(例如开发环境、测试环境、生产环境),而且每一个现场环境均有其对应的配置文件数据,因此,利用存储于上述文件的起始位置的环境标识,并且通过多环境配置文件生成工具(configtools)在所有现场环境配置的模板库中查找与现场环境相对应的配置文件数据,其中该环境标识具有唯一性,且用以区分不同的环境(例如开发环境、测试环境、生产环境)。

[0032] 在步骤(33)中,在所有现场环境配置的模板库中成功提取到相对应的配置文件数据(其为xml文件)之后,将所提取的配置文件数据并入至配置文件模板(其为tpl文件),以生成相应的配置文件(其为cfg文件)。上述tpl文件用以配置节点,xml文件用以根据不同环境产生相应的配置文件数据,相当于数据源,用以提供每一环境所对应的具体值,例如用户名/密码等(每一个环境中的用户名/密码是不同的)。

[0033] 在步骤(34)中,基于上述编译源代码所生成的代码包(例如一次生成所有环境的代码包),并且将配置文件分别置入于相应的代码包中,以生成相应的软件包。在本实施方式中,可以一次生成4种不同环境(例如开发环境、测试环境、生产环境)的代码包。

[0034] 步骤S240:服务器根据目标地址将生成的软件包进行分发。

[0035] 在此步骤中,可以根据目标地址(根据先前在配置点中所设定的目标服务器的地址)将生成的软件包分发至相应的环境中。

[0036] 由此可见,通过执行上述步骤S210至S240,能够实现建立重复、可靠的发布流程,以提高软件包的发布质量,进而避免人为失误,而且能够将整个发布流程自动化,提升整个软件包的发布效率,进而减少相关人员浪费时间在毫无技术含量的重复劳动之中。

[0037] 在本发明的优选实施方式中,在步骤S240之中,可以进一步包括以下步骤:步骤(41)、配置软件包自动发布的发布策略;步骤(42)、根据发布策略提供相应的审批权限。

[0038] 其中,在步骤(41)中生成一发布单,用以配置软件包自动发布的发布策略。在所述发布单中,可以定义软件包的代号、版本号、版本说明、部署手册以及存放地址等信息。这样,多个开发人员可以根据所述发布单的信息而参与于同一项目,以增加软件开发的透明度,进而避免团队中的其他人员无法及时得知整个发布进程。

[0039] 在步骤(42)中,根据发布策略提供相应的审批权限。这样,具有相应审批权限的审批人员(例如开发、测试、PMO、运维人员)在接收到相关申请人员的不同等级的请求之后,对请求发布的软件包进行审批处理及注解,审批通过后,服务器自动发送所需的软件包或文档至申请人员所指定的位置。这样,通过上述步骤,能够实现软件包的版本申请、申请审批、软件包发布等功能,从而替代人工的版本发布的配置控制,提高软件包的发布效率,减少重复劳动,降低人工成本。

[0040] 可选的,在本发明的其他实施方式中,在步骤S240之中,可以进一步包括以下步骤:步骤(51)、配置软件包自动发布的发布策略,以设定多个软件包之间的顺序;步骤(52)、根据所设定的多个软件包之间的顺序,执行多个软件包一次性同时发布的操作。

[0041] 由于一些大型项目需要同时发布多个软件包,而多个软件包之间存在一定的依赖性,因此,需要通过配置软件包自动发布的发布策略,以设定多个软件包之间的先后顺序,相当于形成一任务链。在设定完多个软件包之间的先后顺序之后,可以执行多个软件包一次性同时发布的操作。

[0042] 可选的,在本发明的其他实施方式中,在步骤S240之中,可以进一步包括以下步骤:步骤(61)、配置软件包自动发布的发布策略,以设定软件包的版本信息;步骤(62)、当发现当前软件包内容存有错误时,根据发布策略中所设定的软件包的版本信息,查找到当前软件包的前一次版本所对应的软件包内容;步骤(63)、执行回退操作,以将前一次版本所对应的软件包内容替换当前软件包内容。

[0043] 其中,在步骤(61)中生成一发布单,用以配置软件包自动发布的发布策略。在所述发布单中,可以定义软件包的代号、版本号、版本说明、部署手册以及存放地址等信息。通过执行步骤(62)至步骤(64),可以在发现当前软件包内容存有错误而导致项目运行出现问题,或者是发布软件包的版本有误的时候,根据已存储的版本信息,查找之前已成功发布的软件包版本,并且进行重新软件包的发布,避免因当前发布的软件包出现错误之后,无法及时将已成功发布的软件包发布至服务器,进而影响用户的使用。综上,采用本发明实施方式所提供的软件包自动发布方法不仅能够实现对软件包的自动发布,以节省人力及时间,降低发布出错率,使得用户能够尽快使用更新后更加完善的软件包,并且避免因发布不成功而导致影响用户的使用。

[0044] 可选的,在本发明的其他实施方式中,在步骤S240之中,可以进一步包括以下步骤:提供一服务器管理界面,以对服务器的配置进行监控。

[0045] 具体而言,提供一服务器管理界面,以设定服务器所对应的站点,该站点是通过IIS进行管理。在现有技术中,若需要新增站点,那么相关人员需要登入至服务器进行IIS服务操作,以使新增站点指向相应的文件夹。而在本实施方式中,则不需要登入至服务器以进行IIS服务操作,只要在所提供的一服务器管理界面下,对服务器进行相关配置即可(例如建立站点,或者改变文件夹的路径)。这样,与服务器相连的客户端在接收到服务器所发布的软件包之后,执行解压软件包操作并且根据软件包的相关配置而改变用于存放文件的文件夹的路径。优先地,采用具有独立性的windows服务替代现有的IIS服务,且采用传输效率较高的TCP/IP协议替换现有安全性不高且传输效率较低的http协议方式,从而降低因IIS服务发生故障而影响整个发布系统。

[0046] 综上,采用本发明实施方式所提供的软件包自动发布方法不仅能够保障安全,避

避免因修改文件而影响整个服务器的权限,而且在无需登入服务器的情况下,根据所提供一服务器管理界面对服务器进行相关配置,同时降低犯错率。

[0047] 可选的,在本发明的实施方式中,在步骤S230之中,可以进一步包括以下步骤:步骤(81)、基于所生成的配置文件,添加伪数据信息至编译后的源代码,以执行后继的打包操作;步骤(82)、当打包至一设定现场环境相对应的软件包时,在配置文件中匹配所述伪数据信息;步骤(83)、若匹配到所述伪数据信息,则在敏感信息管理数据库中提取与伪数据信息相对应的敏感数据信息,并将所述伪数据信息替换为所述敏感数据信息。

[0048] 在步骤(81)中,由于每一环境中(例如测试环境、生产环境等)的用户名/密码是不同的,并且各个环境的保密度也不同,更重要的是,为了避免一特定环境(例如开发环境)中的用户名/密码在其他环境中(例如生产环境)被公开,因此,在步骤(81)中,基于所生成的配置文件,添加伪数据信息至编译后的源代码,其中所述伪数据信息可以为字符或字符串等(例如"ABCD"),且不具有任意意义。在步骤(82)中,当打包至一设定现场环境(例如生产环境)相对应的软件包时,在配置文件(如cfg文件)中匹配所述伪数据信息。在步骤(83)中,若匹配到所述伪数据信息,则在敏感信息管理数据库中提取与伪数据信息(例如"ABCD")相应的敏感数据信息(例如"ABCD"对应的用户名/密码),并且将所述伪数据信息替换为所述敏感数据信息。

[0049] 综上,通过采用本发明所述的软件包自动发布方法能够建立重复、可靠的发布流程,以提高软件包的发布质量,进而避免人为失误,而且能够将整个发布流程自动化,提升整个软件包的发布效率,进而减少相关人员浪费时间在毫无技术含量的重复劳动之中,再者,其能够使多个开发人员同时参与,以增加软件开发的透明度,进而避免团队中的其他人员无法及时得知整个发布进程。

[0050] 另外,参见图3所示,本发明还提供一种软件包自动发布系统。所述系统包括:源代码获取单元310、源代码编译单元320、打包单元330和软件包分发单元340。

[0051] 其中,所述源代码获取单元310用以服务器在接收到客户端所发送的发布指令后,获取待发布软件包的源代码。待发布软件包的源代码可以存储在开发人员的本地设备中或者存储在一集中式管理的源代码管理服务器(如TFS模式的服务器)中,且不限于此。开发人员或测试人员或其他相关人员可以通过客户端或前台所提供的(网站)站点,选择待发送的软件包及其相应的现场环境、相关参数等,以执行发布指令。服务器在接收到开发人员或测试人员或其他相关人员的发布请求之后,可以根据软件包的名称、版本等信息调用并获取相应软件包的源代码。

[0052] 所述源代码编译单元320用以服务器通过内置编译器对所述源代码进行编译。所述内置编译器可以为例如C、PHP、Python、Java等多种语言的编译器。所述编译器可以为本领域技术人员所常用的编译器,在此不再一一列举。

[0053] 所述打包单元330用以服务器根据内置的多个现场环境规则,将编译后的源代码进行打包,以生成与现场环境相对应的软件包。

[0054] 所述软件包分发单元340用以服务器根据目标地址将生成的软件包进行分发。根据目标地址(根据先前在配置点中所设定的目标服务器的地址)将生成的软件包分发至相应的环境中。

[0055] 在本发明一实施方式中,在所述源代码获取单元310中进一步包括:配置点提供模

块311和目标需求列表提供模块312,如图4所示,其中,所述配置点提供模块311用以服务器提供一配置点,以确定现场环境中对应的目标服务器。通过所述配置点可以确定待发送的代码包所对应的现场环境(例如开发环境、测试环境、生产环境等),每一现场环境对应于相应的目标服务器(包括目标服务器的数量等)。所述目标需求列表提供模块312用以服务器提供一目标需求列表,以选择与目标需求相对应的代码包,进而确定与目标需求相对应的现场环境。也就是说,当选择与目标需求相对应的代码包(即程序)时,可以确定该软件包将发送到相应的现场环境(例如测试环境、生产环境)。

[0056] 进一步,参见图5所示,所述打包单元330可以包括:模板库调用模块331、配置文件数据查找模块333、配置文件生成模块335和配置文件添加模块337。

[0057] 其中,所述模板库调用模块331用以调用所有现场环境配置的模板库。所有现场环境配置(例如测试环境、生产环境等)的模板库可以存储于一文件之中,该文件类似于文本文件。

[0058] 所述配置文件数据查找模块333用以通过一多环境配置文件生成工具并根据与目标需求相对应的代码包,在所有现场环境配置的模板库中查找与现场环境相应的配置文件数据。当开发人员或测试人员或其他相关人员在选择待发布软件包时,就已经自动确认了相应的环境(例如开发环境、测试环境、生产环境),而且每一个现场环境均有其对应的配置文件数据,因此,利用存储于上述文件的起始位置的环境标识,并且通过多环境配置文件生成工具(configtools)在所有现场环境配置的模板库中查找与现场环境相对应的配置文件数据,其中该环境标识具有唯一性,且用以区分不同的环境(例如开发环境、测试环境、生产环境)。

[0059] 所述配置文件生成模块335用以在所有现场环境配置的模板库中成功提取到相对应的配置文件数据之后,将所提取的配置文件数据并入至配置文件模板,以生成相应的配置文件。也就是说,在所有现场环境配置的模板库中成功提取到相对应的配置文件数据(其为xml文件)之后,将所提取的配置文件数据并入至配置文件模板(其为tpl文件),以生成相应的配置文件(其为cfg文件)。上述tpl文件用以配置节点,xml文件用以根据不同环境产生相应的配置文件数据,相当于数据源,用以提供每一环境所对应的具体值,例如用户名/密码等(每一个环境中的用户名/密码是不同的)。

[0060] 所述配置文件添加模块337,用以将所生成的配置文件添加至编译后的源代码之中,以执行后继的打包操作。基于上述编译源代码所生成的代码包(例如一次生成所有环境的代码包),并且将配置文件分别置入于相应的代码包中,以生成相应的软件包。在本实施方式中,可以一次生成4种不同环境(例如开发环境、测试环境、生产环境)的代码包。

[0061] 综上,通过所述软件包自动发布系统能够实现建立重复、可靠的发布流程,以提高软件包的发布质量,进而避免人为失误,而且能够将整个发布流程自动化,提升整个软件包的发布效率,进而减少相关人员浪费时间在毫无技术含量的重复劳动之中。

[0062] 可选的,在本发明的一实施方式中,所述软件包分发单元340进一步包括:发布策略配置模块341和审批权限提供模块342,如图6所示。

[0063] 其中,所述发布策略配置模块341用以配置软件包自动发布的发布策略。通过所述发布策略配置模块341可以生成一发布单,在所述发布单中,可以定义软件包的代号、版本号、版本说明、部署手册以及存放地址等信息。这样,多个开发人员可以根据所述发布单的

信息而参与于同一项目,以增加软件开发的透明度,进而避免团队中的其他人员无法及时得知整个发布进程。

[0064] 所述审批权限提供模块342用以根据发布策略提供相应的审批权限。这样,具有相应审批权限的审批人员(例如开发、测试、PMO、运维人员)在接收到相关申请人员的不同等级的请求之后,对请求发布的软件包进行审批处理及注解,审批通过后,服务器自动发送所需的软件包或文档至申请人员所指定的位置。于是,能够实现软件包的版本申请、申请审批、软件包发布等功能,从而替代人工的版本发布的配置控制,提高软件包的发布效率,减少重复劳动,降低人工成本。

[0065] 可选的,在本发明的另一实施方式中,所述软件包分发单元340进一步包括:发布策略配置模块341和多软件包发布模块343,如图7所示。其中,所述发布策略配置模块341用以配置软件包自动发布的发布策略,以设定多个软件包之间的顺序。所述多软件包发布模块343,用以根据所设定的多个软件包之间的顺序,执行多个软件包一次性同时发布的操作。

[0066] 具体而言,由于一些大型项目需要同时发布多个软件包,而多个软件包之间存在一定的依赖性,因此,需要通过配置软件包自动发布的发布策略,以设定多个软件包之间的先后顺序,相当于形成一任务链。在设定完多个软件包之间的先后顺序之后,可以执行多个软件包一次性同时发布的操作。

[0067] 可选的,在本发明的另一实施方式中,所述软件包分发单元340进一步包括:发布策略配置模块341、软件包内容查找模块344和回退操作执行模块345,如图8所示。其中,所述发布策略配置模块341用以配置软件包自动发布的发布策略,以设定软件包的版本信息。所述软件包内容查找模块344用以当发现当前软件包内容存有错误时,根据发布策略中所设定的软件包的版本信息,查找到当前软件包的前一次版本所对应的软件包内容。所述回退操作执行模块345用以执行回退操作,以将前一次版本所对应的软件包内容替换当前软件包内容。这样,在发现当前软件包内容存有错误而导致项目运行出现问题,或者是发布软件包的版本有误的时候,根据已存储的版本信息,查找之前已成功发布的软件包版本,并且进行重新软件包的发布,避免因当前发布的软件包出现错误之后,无法及时将已成功发布的软件包发布至服务器,进而影响用户的使用。综上,采用上述模块不仅能够实现对软件包的自动发布,以节省人力及时间,降低发布出错率,使得用户能够尽快使用更新后更加完善的软件包,并且避免因发布不成功而导致影响用户的使用。

[0068] 可选的,在本发明的另一实施方式中,所述软件包分发单元340进一步包括:服务器配置监控模块346,如图9所示。具体而言,所述服务器配置监控模块346用以提供一服务器管理界面,以对服务器的配置(例如建立站点,或者改变文件夹的路径)以进行监控。这样,不仅能够保障安全,避免因修改文件而影响整个服务器的权限,而且在无需登入服务器的情况下,通过服务器配置监控模块对服务器进行相关配置,同时降低犯错率。

[0069] 可选的,在本发明的另一实施方式中,所述打包单元330进一步包括:伪数据信息添加模块332、伪数据信息匹配模块334和敏感数据信息替换模块336,如图10所示。

[0070] 其中,所述伪数据信息添加模块332用以基于所生成的配置文件,添加伪数据信息至编译后的源代码,以执行后继的打包操作。所述伪数据信息匹配模块334用以当打包至一设定现场环境相对应的软件包时,在配置文件中匹配所述伪数据信息。所述敏感数据信息

替换模块336用以当匹配到所述伪数据信息时,在敏感信息管理数据库中提取与伪数据信息相应的敏感数据信息,并将所述伪数据信息替换为所述敏感数据信息。

[0071] 具体而言,由于每一环境中(例如测试环境、生产环境等)的用户名/密码是不同的,并且各个环境的保密度也不同,更重要的是,为了避免一特定环境(例如开发环境)中的用户名/密码在其他环境中(例如生产环境)被公开,因此,通过所述伪数据信息添加模块332以在基于所生成的配置文件的情况下,添加伪数据信息至编译后的源代码,其中所述伪数据信息可以为字符或字符串等(例如"ABCD"),且不具有任意意义。通过所述伪数据信息匹配模块334以当打包至一设定现场环境(例如生产环境)相对应的软件包时,在配置文件(如cfg文件)中匹配所述伪数据信息。通过敏感数据信息替换模块336以当匹配到所述伪数据信息时,在敏感信息管理数据库中提取与伪数据信息(例如"ABCD")相应的敏感数据信息(例如"ABCD"对应的用户名/密码),并且将所述伪数据信息替换为所述敏感数据信息。

[0072] 显然,本领域技术人员应该明白,上述本发明实施方式的各模块或各步骤可以用通用的计算机装置来实现,它们可以集中在单个的计算机装置上,或者分布在多个计算机装置所组合成的网络上,可选地,它们可以用计算机装置可执行的软件包源代码来实现,从而,可以将它们存储在存储装置中由计算机装置来执行,并且在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤,或者将它们分别制成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本发明实施方式不限于任何特定的硬件和软件结合。

[0073] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

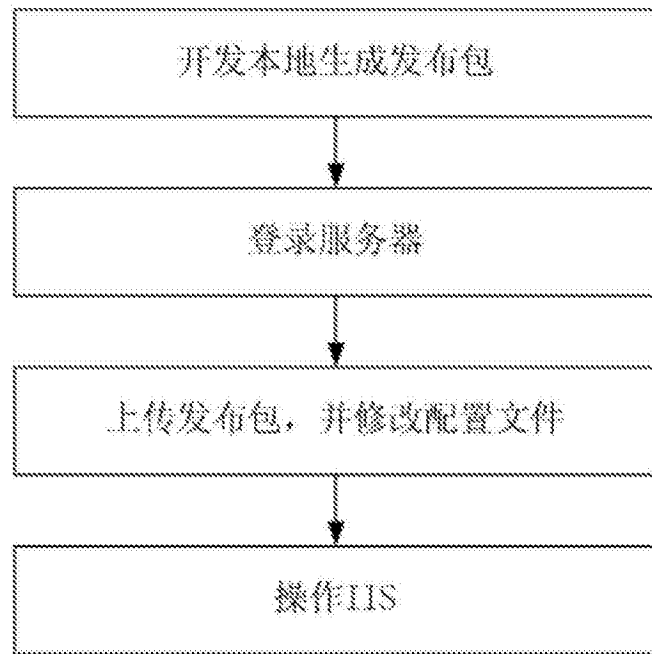


图1

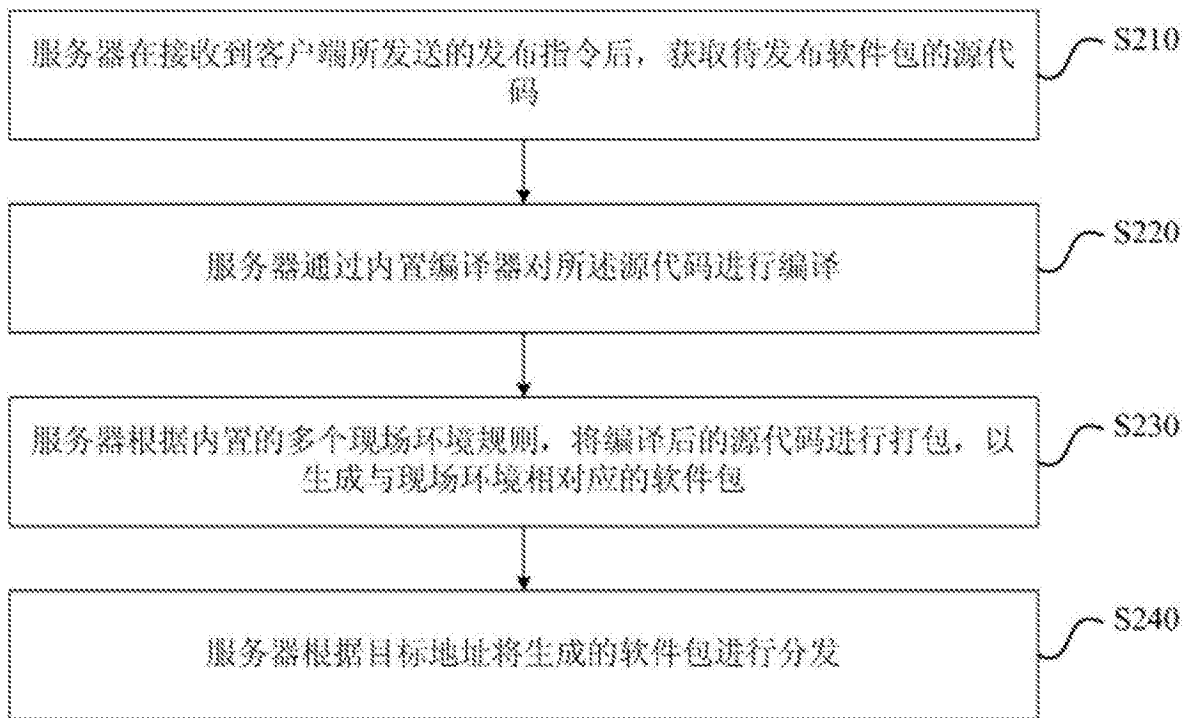


图2

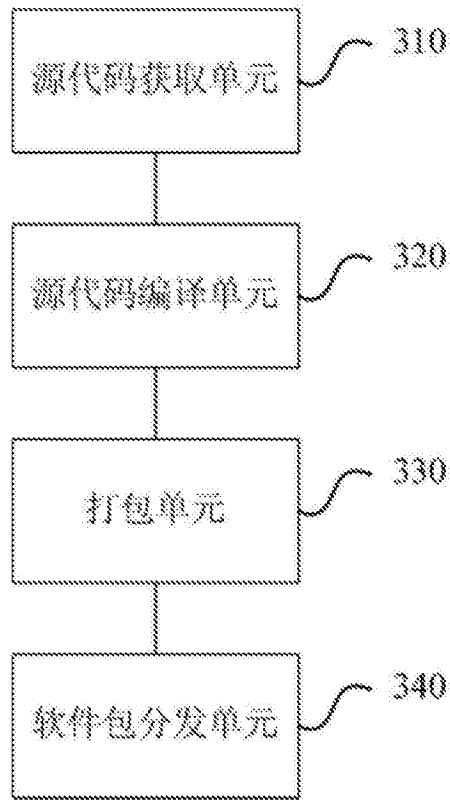


图3

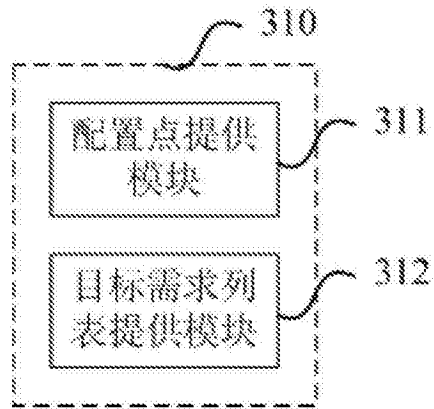


图4

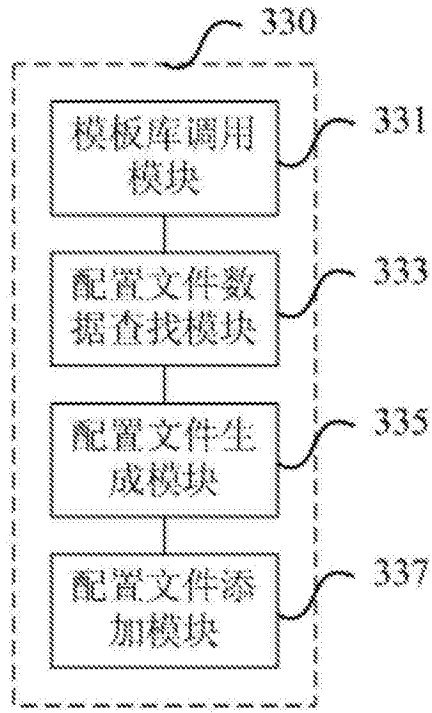


图5

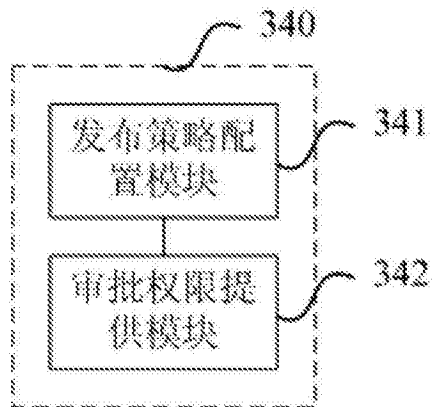


图6

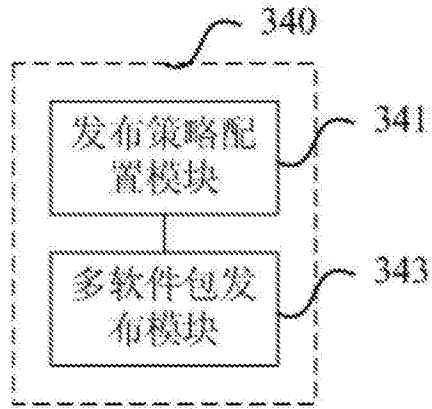


图7

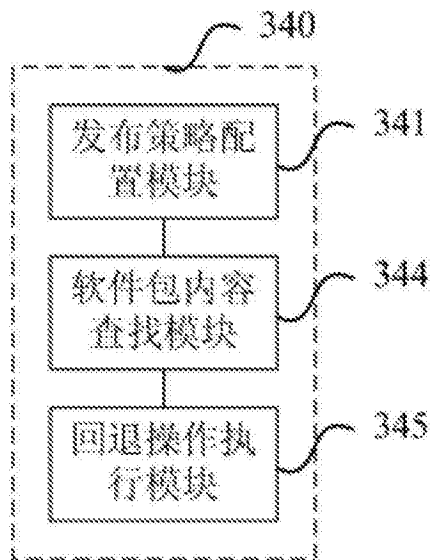


图8



图9

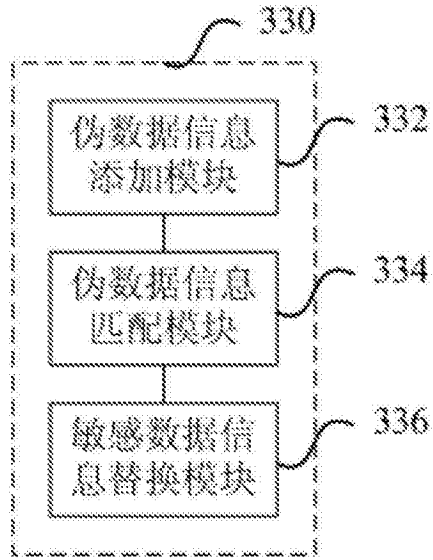


图10