



(12)发明专利申请

(10)申请公布号 CN 110895472 A

(43)申请公布日 2020.03.20

(21)申请号 201811069076.4

(22)申请日 2018.09.13

(71)申请人 北京京东尚科信息技术有限公司
地址 100195 北京市海淀区杏石口路65号
西杉创意园四区11号楼东段1-4层西
段1-4层
申请人 北京京东世纪贸易有限公司

(72)发明人 孙晓野

(74)专利代理机构 中原信达知识产权代理有限
责任公司 11219
代理人 张一军 李阳

(51)Int.Cl.
G06F 8/71(2018.01)

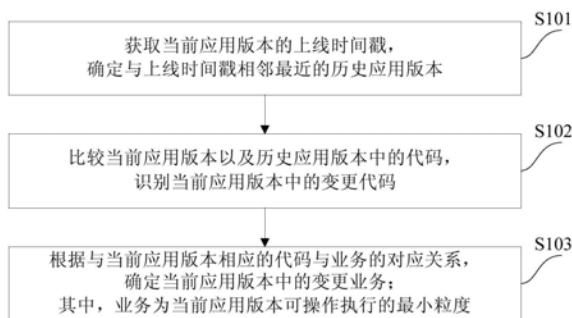
权利要求书2页 说明书10页 附图3页

(54)发明名称

一种识别业务变更的方法和装置

(57)摘要

本发明公开了一种识别业务变更的方法和装置,涉及计算机技术领域。该方法的一具体实施方式包括:获取当前应用版本的上线时间戳,确定与上线时间戳相邻最近的历史应用版本;比较当前应用版本以及历史应用版本中的代码,识别当前应用版本中的变更代码;根据与当前应用版本相应的代码与业务的对应关系,确定当前应用版本中的变更业务;其中,业务为当前应用版本可操作执行的最小粒度。该实施方式能够自行识别当前版本与历史上线版本中,与所修改的代码相应的业务名称,并自动为所识别到的变更业务添加报警监控。



1. 一种识别业务变更的方法,其特征在于,包括:

获取当前应用版本的上线时间戳,确定与所述上线时间戳相邻最近的历史应用版本;

比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码;

根据与所述当前应用版本相应的代码与业务的对应关系,确定所述当前应用版本中的变更业务;其中,所述业务为所述当前应用版本可操作执行的最小粒度。

2. 根据权利要求1所述的方法,其特征在于,所述比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码,包括:

获取所述当前应用版本中的模块,确定所述模块的标识信息;其中,所述标识信息至少包括所述模块的名称;

根据所述标识信息,确定所述历史应用版本中与所述标识信息相应的历史模块,比对所述模块以及所述历史模块中的代码,确定所述模块中的变更代码;或

当所述历史应用版本中不存在与所述标识信息相应的历史模块时,确定所述模块中的所有代码为变更代码。

3. 根据权利要求1所述的方法,其特征在于,在所述确定所述当前应用版本中的变更业务之后,还包括:

传输所述变更业务的业务信息至监控平台,以基于所述业务信息,对所述当前应用版本中所述变更业务的执行参数进行监控;以及

当接收到所述监控平台所传输的报警请求时,生成相应的报警指令并传输至报警平台进行报警操作。

4. 根据权利要求1所述的方法,其特征在于,还包括:

获取所述当前应用版本中的模块,根据与所述应用版本相应的模块分支列表,确定所述模块所处上线分支;

提取所确定的所述上线分支中与所述当前应用版本相应的提交记录,组合所获取的提交记录并打标。

5. 一种识别业务变更的装置,其特征在于,包括:

版本确定模块,用于获取当前应用版本的上线时间戳,确定与所述上线时间戳相邻最近的历史应用版本;

代码识别模块,用于比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码;

业务确定模块,用于根据与所述当前应用版本相应的代码与业务的对应关系,确定所述当前应用版本中的变更业务;其中,所述业务为所述当前应用版本可操作执行的最小粒度。

6. 根据权利要求5所述的装置,其特征在于,所述代码识别模块,用于:

获取所述当前应用版本中的模块,确定所述模块的标识信息;其中,所述标识信息至少包括所述模块的名称;

根据所述标识信息,确定所述历史应用版本中与所述标识信息相应的历史模块,比对所述模块以及所述历史模块中的代码,确定所述模块中的变更代码;或

当所述历史应用版本中不存在与所述标识信息相应的历史模块时,确定所述模块中的

所有代码为变更代码。

7. 根据权利要求5所述的装置,其特征在于,还包括业务监控模块,用于:

传输所述变更业务的业务信息至监控平台,以基于所述业务信息,对所述当前应用版本中所述变更业务的执行参数进行监控;以及

当接收到所述监控平台所传输的报警请求时,生成相应的报警指令并传输至报警平台进行报警操作。

8. 根据权利要求5所述的装置,其特征在于,还包括打标记录模块,用于:

获取所述当前应用版本中的模块,根据与所述应用版本相应的模块分支列表,确定所述模块所处上线分支;

提取所确定的所述上线分支中与所述当前应用版本相应的提交记录,组合所获取的提交记录并打标。

9. 一种电子设备,其特征在于,包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1-4中任一所述的方法。

10. 一种计算机可读介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时实现如权利要求1-4中任一所述的方法。

一种识别业务变更的方法和装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种识别业务变更的方法和装置。

背景技术

[0002] 随着互联网的广泛应用,计算机应用系统架构从单集群冷热互备演化为多集群分布式架构,以提高系统负载能力、降低模块间的耦合、增强模块的内聚。

[0003] 现有应用普遍采用分布式架构,普遍使用RPC (Remote Procedure Call Protocol,远程过程调用协议)框架,且通常由多个不同功能的模块进行组装。

[0004] 在实现本发明的过程中,发明人发现现有技术至少存在如下问题:

[0005] 1) 上线版本的内容更改较多,目前通用的代码管理工具git只能记录到文件级别,而不会记录到业务变更级别;

[0006] 2) 监控平台对于所修改的每个方法的监控,需要开发人员手工配置,工作量过大、且较为繁琐;若监控不及时,可能会影响现有生产的大规模作业;

[0007] 3) 每次所上线的版本,其提交记录(commit ID)、标签等通常由人工记录,存在人工维护版本的成本较高、错误出现较多的问题。

发明内容

[0008] 有鉴于此,本发明实施例提供一种识别业务变更的方法和装置,至少能够解决现有技术中对于上线版本中所修改的业务,监控需要人工参与,工作量大且繁琐的问题。

[0009] 为实现上述目的,根据本发明实施例的一个方面,提供了一种识别业务变更的方法,包括:

[0010] 获取当前应用版本的上线时间戳,确定与所述上线时间戳相邻最近的历史应用版本;

[0011] 比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码;

[0012] 根据与所述当前应用版本相应的代码与业务的对应关系,确定所述当前应用版本中的变更业务;其中,所述业务为所述当前应用版本可操作执行的最小粒度。

[0013] 可选的,所述比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码,包括:

[0014] 获取所述当前应用版本中的模块,确定所述模块的标识信息;其中,所述标识信息至少包括所述模块的名称;

[0015] 根据所述标识信息,确定所述历史应用版本中与所述标识信息相应的历史模块,比对所述模块以及所述历史模块中的代码,确定所述模块中的变更代码;或

[0016] 当所述历史应用版本中不存在与所述标识信息相应的历史模块时,确定所述模块中的所有代码为变更代码。

[0017] 可选的,在所述确定所述当前应用版本中的变更业务之后,还包括:

[0018] 传输所述变更业务的业务信息至监控平台,以基于所述业务信息,对所述当前应用版本中所述变更业务的执行参数进行监控;以及

[0019] 当接收到所述监控平台所传输的报警请求时,生成相应的报警指令并传输至报警平台进行报警操作。

[0020] 可选的,还包括:

[0021] 获取所述当前应用版本中的模块,根据与所述应用版本相应的模块分支列表,确定所述模块所处上线分支;

[0022] 提取所确定的所述上线分支中与所述当前应用版本相应的提交记录,组合所获取的提交记录并打标。

[0023] 为实现上述目的,根据本发明实施例的另一方面,提供了一种识别业务变更的装置,包括:

[0024] 版本确定模块,用于获取当前应用版本的上线时间戳,确定与所述上线时间戳相邻最近的历史应用版本;

[0025] 代码识别模块,用于比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码;

[0026] 业务确定模块,用于根据与所述当前应用版本相应的代码与业务的对应关系,确定所述当前应用版本中的变更业务;其中,所述业务为所述当前应用版本可操作执行的最小粒度。

[0027] 可选的,所述代码识别模块,用于:

[0028] 获取所述当前应用版本中的模块,确定所述模块的标识信息;其中,所述标识信息至少包括所述模块的名称;

[0029] 根据所述标识信息,确定所述历史应用版本中与所述标识信息相应的历史模块,比对所述模块以及所述历史模块中的代码,确定所述模块中的变更代码;或

[0030] 当所述历史应用版本中不存在与所述标识信息相应的历史模块时,确定所述模块中的所有代码为变更代码。

[0031] 可选的,还包括业务监控模块,用于:

[0032] 传输所述变更业务的业务信息至监控平台,以基于所述业务信息,对所述当前应用版本中所述变更业务的执行参数进行监控;以及

[0033] 当接收到所述监控平台所传输的报警请求时,生成相应的报警指令并传输至报警平台进行报警操作。

[0034] 可选的,还包括打标记录模块,用于:

[0035] 获取所述当前应用版本中的模块,根据与所述应用版本相应的模块分支列表,确定所述模块所处上线分支;

[0036] 提取所确定的所述上线分支中与所述当前应用版本相应的提交记录,组合所获取的提交记录并打标。

[0037] 为实现上述目的,根据本发明实施例的再一方面,提供了一种识别业务变更的电子设备。

[0038] 本发明实施例的电子设备包括:一个或多个处理器;存储装置,用于存储一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理

器实现上述任一所述的识别业务变更的方法。

[0039] 为实现上述目的,根据本发明实施例的再一方面,提供了一种计算机可读介质,其上存储有计算机程序,所述程序被处理器执行时实现上述任一所述的识别业务变更的方法。

[0040] 根据本发明所述提供的方案,上述发明中的一个实施例具有如下优点或有益效果:能够自行识别当前版本与历史上线版本中,与所修改的代码相应的业务名称,并自动为所识别到的变更业务添加报警监控。

[0041] 上述的非惯用的可选方式所具有的进一步效果将在下文中结合具体实施方式加以说明。

附图说明

[0042] 附图用于更好地理解本发明,不构成对本发明的不当限定。其中:

[0043] 图1是根据本发明实施例的一种识别业务变更的方法的主要流程示意图;

[0044] 图2是根据本发明实施例的一种识别业务变更的装置的主要模块示意图;

[0045] 图3是本发明实施例可以应用于其中的示例性系统架构图;

[0046] 图4是适于用来实现本发明实施例的移动设备或服务器的计算机系统的结构示意图。

具体实施方式

[0047] 以下结合附图对本发明的示范性实施例做出说明,其中包括本发明实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本发明的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0048] 需要说明的是,本发明实施例可适用于具有功能/业务更新的应用系统,例如,仓储系统以及其他涉及到仓储系统的应用系统,本发明以仓储系统为例进行说明。

[0049] 另外,本发明除了适用于对已上线应用版本的监控之外,还可以对测试环境中测试版本的监控。只不过已上线的应用版本较之测试版本较为固定,所管理的有效数据更多一些,因此,本发明主要针对上线的应用版本进行描述。

[0050] 参见图1,示出的是本发明实施例提供的一种识别业务变更的方法的主要流程图,包括如下步骤:

[0051] S101:获取当前应用版本的上线时间戳,确定与上线时间戳相邻最近的历史应用版本;

[0052] S102:比较当前应用版本以及历史应用版本中的代码,识别当前应用版本中的变更代码;

[0053] S103:根据与当前应用版本相应的代码与业务的对应关系,确定当前应用版本中的变更业务;其中,业务为当前应用版本可操作执行的最小粒度。

[0054] 上述实施方式中,对于步骤S101,本发明所涉及的版本,指的是应用程序版本,是一个抽象概念,对于应用版本后续简化称为版本。同商品上架时具有特定名称一样(例如2018秋季款限定J品牌包),应用每次新功能上线都会有一个版本号,例如V2.01,且该版本

名称通常由工作人员根据每次所上线的业务内容进行定义。

[0055] 同一应用可能具有多个历史版本,但各个版本的功能可能都不尽相同。每个版本都是较之之前相邻最近的历史版本进行改进的,因此所比较的对象就是当前版本以及历史相邻最近的版本。

[0056] 而对于历史版本的获取依据主要是当期版本的上线时间戳、以及当前版本的应用名称,例如,X商场v1.00-2018.08.01上线、X商场v1.02-2018.08.14上线。且通常情况下,历史版本存储于上线服务器中。

[0057] 对于步骤S102,由于开发环境、测试环境以及上线环境的不同,对于当前所上线的新应用版本,其较之历史应用版本所做出的功能改进/更改,需要对其进行监控,若执行出现异常,则出示报警提醒。

[0058] 对于最新上线的版本,其功能改进/变更的模块,主要通过比较两个应用版本的代码获得,具体地:

[0059] 1) 分别确定当前版本、历史版本的模块以及所包含的代码

[0060] 在版本管理系统中,一个版本的上线即是所包含模块的上线,例如,2018.08.09X商城上线,包括APP(Application)上线、后台模块1以及后台模块2上线。

[0061] Java语言所开发的应用程序,可以将固定功能的模块打包成一个以jar为后缀的压缩文件,用来实现指定功能。例如,模块1-接收用户名和密码、模块2-校验用户名和密码是否匹配。

[0062] 对于代码,通常存储于git-server中。这里git是个代码保存工具,但若代码保存在电脑中时候,易出现电脑损坏、代码丢失的情况,因此通常情况下会使用git开源工具,保存代码到远程服务器中,便于读取以及保存管理。

[0063] 2) 对比当前版本的模块较之历史版本的模块中的代码,确定存在代码变更的变更模块,并调用git diff命令,确定变更代码;

[0064] 命令形式如下:

[0065] diff—git a/f1b/f1

[0066] index 6f8a38c..449b072 100644

[0067] ---a/f1

[0068] +++b/f1

[0069] @@-1,7,+1,7@@

[0070] -a

[0071] +b

[0072] 需要说明的是,应用版本每次上线的模块数量不一定相同,同样所存储的位置也不一定完全一致,即相应的代码、代码所处行数等,都可能是不一样的。

[0073] 另外,历史版本以及当前本版本中的模块,需要针对性的一一对比,主要依据是其模块名称,且通常情况下,一个功能模块的名称不会更改。

[0074] 3) 每个应用上线前,其所包含的模块中的代码,与方法具有一定的对应关系,例如模块-业务-代码表。这里的业务,针对于执行方法,表示的是可操作执行的最小粒度。

[0075] 确定到代码之后,根据该表,确定其所对应的业务,即表示这部分方法/业务,较之历史版本是有所变更的。

[0076] 例如同一应用,2018.08.09的“包邮”模块,对应于2018.08.10的“包邮”模块,且识别存在代码替换、删减等,即表示对该功能做了更新,后续也是需要监控的;若发现历史模块中不存在该“包邮”模块,即表示2018.08.10的“包邮”模块为新增模块,包邮功能为新增功能。

[0077] 作为步骤2)的可替代方式,可以确定变更代码在相应模块中的所属行数,进而确定所发生变更的业务:

[0078] 1) 使用Java LineNumberReader,读取diff文件,确定所变更的代码对应的行数。通过正则表达式Pattern.compile("^+\\w+")匹配,将某个字符模式与所搜索的字符串进行匹配,并将变更行数记录在内存中,例如[1,3,6],表示第一行、第三行以及第六行的代码有所变更;

[0079] 2) 使用JDK (Java Development Kit)中java自带的tools包,对变更的代码行数进行解析,以分析出所修改过的文件中,业务的起始行数以及截止行数;

[0080] 具体地,使用JavaCompiler类对指定路径的类进行编译,得到编译后的编译对象;例如,将工作人员所编写的代码,翻译为计算机可识别的指令,Safwseoirfuweorfj这堆代码翻译成:我今天11点约好吃饭;

[0081] 使用JavaTask对象对编译后的对象进行解析,得到Compilation UnitTree对象,并对CompilationUnitTree进行循环,根据时间变化序列累计业务/代码,并得到发生变化的是哪个业务、以及该业务的起始行数与截止行数。

[0082] 这里的JavaTask是一个抽象工具,JavaCompiler借助JavaTask来对代码翻译;例如,把变更的代码行数6发送给JavaTask,就可以得知发生变更的代码所对应的是哪个业务。

[0083] 3) 将本次步骤1)所得代码变更行数以及步骤2)所得业务所对应的行数进行比较,这里可以有业务-代码-代码行数表,以确定代码落在哪个业务内。由于业务与业务相对应,进而确定当前版本较之历史版本,所发生变更的业务信息。

[0084] 另外,需要说明的是,业务位于文件内,一个文件内部是由不同的业务所构成的,例如定时关机功能:

[0085] 功能开发所生成的文件:定时关机.java

[0086] 文件关机.java内所包含的业务为:

[0087] 业务1 {检查当前系统所处时间是否等于预定关机时间};

[0088] 业务2 {关机};

[0089] 业务3 {调用业务1(),调用业务2()}。

[0090] 确定变更业务/业务之后,可以将当前版本较之历史版本所修改的业务/业务信息,统一发送给企业监控平台,用以监控当前版本对该业务的执行参数(例如,执行时长、执行结果是否成功等)。

[0091] 目前各企业基本都有各自独立的监控平台,且通常独立于应用系统,例如,通过对所监控的业务添加AOP (Aspect Oriented Programming,面向切面编程)注解的方式,监控记录该业务的执行参数,以此分析其运行效率。

[0092] 监控平台若监测到业务的执行出现异常时,可以执行报警提醒;例如,记录一次业务运行开始到结束的时间,若运行时长超过30s,就报警提醒,实现对新功能、新模块的控制

管理,降低业务运行的出错概率。

[0093] 但也有有的监控平台不具备报警功能,此时需要生成报警指令,传输至报警器进行报警提醒。

[0094] 在版本管理系统中,对于当前应用版本,其应用名称、版本编号等标识信息,通常需要依赖工作人员,通过界面基础工具配置进行设定,例如:

[0095] 1) 应用名称:主要是该应用第一次开发时录入。企业内部系统可能较多,为避免后续应用管理混淆,可以使用定义名称的方式对其进行区分,例如X商城、X物流、X仓储WMS (Warehouse Management System) 系统等;

[0096] 2) 应用所包含的模块以及模块名称:通常由业务方、产品类型以及产品功能定义,以此设计APP和后台代码的改造。例如,对于仓储系统,所包含的模块可以有:远程服务、web端、序列号管理平台、任务异步处理平台、接单平台、接单报文解析平台、打印平台、集中配置管理平台、库存服务平台。

[0097] 3) 各个模块的代码仓储git访问地址:可以通过git-local (即本地git仓库、客户端) 获取,

[0098] 4) 所能够访问代码仓库的git用户名和密码,并通过JGIT第三方jar包进行测试,校验应用功能是否可以正常连接使用;例如,银行网页登录,校验所输入的用户名和密码是否正确。当测试结果为正确/可行/能联通时,即表示测试通过,能够正常提供服务。

[0099] 应用上线前,需要约定其版本号,便于记录和后续回滚。例如,2018/07/09所上线的应用版本出现问题,则全套回滚至上线之前的版本中。

[0100] 其具体实施过程为:

[0101] 1) 选择已上线的应用;

[0102] 2) 根据本次上线内容,选择相应的模块进行组合;例如,X商城v1.00可以购买自营商品功能,v1.02可以购买第三方门店商品,v1.03可以使用货到付款,而1.02版本较之v1.00需通过开发5000行代码、100个文件实现。

[0103] 也可以是根据所变更的业务内容,输入相应的代码,并基于该代码生成方法/文件,并打包生成模块,然后与历史模块进行组合、删减、增加或替换,具体根据上线内容确定。

[0104] 另外,存在与所修改的内容相应的模块,例如,秒杀模块,若X商城需要添加该功能,则直接将该模块与原历史模块进行组合即可。

[0105] 例如,2018/08/09X商城上线版本v2.00,所包含的功能为后台性能提升,并能够支持1亿用户,这种升级只需要提升/添加后台模块即可;2018/09/15上线版本v2.02,这次版本增加了“晚上11点秒杀”的功能;2018/09/20上线版本v2.04,删除了“晚上11点秒杀”功能。

[0106] 3) 使用JGIT获取模块-分支列表,确定模块所处的上线分支;

[0107] 需要说明的是,分支是计算机中的一个代码管理的概念,每个分支中包含不同应用内容功能的改造。

[0108] 例如,不同小组针对同一应用进行不同的新功能开发,且不同功能所处的分支不同,A小组-白条支付-分支A、B小组-包邮-分支B,不同分支上的功能、代码等互不影响,体现了分支隔离的作用。

[0109] 但实际上线的时候,可能“包邮”的成本过高,故暂时不上线,便有根据实际情况选择上线分支的方式。但若选择功能A、B均一起上线的话,是需要选择分支A、B的。

[0110] 4) 使用JGIT获取所确定的上线分支下的提交记录(commit ID),选择此版本上线的commit ID,并进行打标记记录处理;通常提交记录会存储在wiki、excel或是word中。

[0111] 这样后续,就可以通过标签可以快速找到所对应的组合,例如:

[0112] 标签1:功能A+功能B;

[0113] 标签2:功能A+功能B+功能C;

[0114] 在代码开发中,标签对应于修改文件的组合,可以记录版本更新的阶段性变化。当版本上线后出现问题时,可以找到历史版本,回滚并重新上线。

[0115] 本发明实施例所提供的方法,能够自行识别当前版本与历史上线版本中,所修改的Java代码中所涉及的方法名称/业务名称,并自动为所识别到的方法/业务添加报警监控。

[0116] 参见图2,示出了本发明实施例提供的一种识别业务变更的装置200的主要模块示意图,包括:

[0117] 版本确定模块201,用于获取当前应用版本的上线时间戳,确定与所述上线时间戳相邻最近的历史应用版本;

[0118] 代码识别模块202,用于比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码;

[0119] 业务确定模块203,用于根据与所述当前应用版本相应的代码与业务的对应关系,确定所述当前应用版本中的变更业务;其中,所述业务为所述当前应用版本可操作执行的最小粒度。

[0120] 本发明实施装置中,所述代码识别模块202,用于:

[0121] 获取所述当前应用版本中的模块,确定所述模块的标识信息;其中,所述标识信息至少包括所述模块的名称;

[0122] 根据所述标识信息,确定所述历史应用版本中与所述标识信息相应的历史模块,比对所述模块以及所述历史模块中的代码,确定所述模块中的变更代码;或

[0123] 当所述历史应用版本中不存在与所述标识信息相应的历史模块时,确定所述模块中的所有代码为变更代码。

[0124] 本发明实施装置还包括业务监控模块204(图中未标出),用于:

[0125] 传输所述变更业务的业务信息至监控平台,以基于所述业务信息,对所述当前应用版本中所述变更业务的执行参数进行监控;以及

[0126] 当接收到所述监控平台所传输的报警请求时,生成相应的报警指令并传输至报警平台进行报警操作。

[0127] 本发明实施装置还包括打标记记录模块205(图中未标出),用于:

[0128] 获取所述当前应用版本中的模块,根据与所述应用版本相应的模块分支列表,确定所述模块所处上线分支;

[0129] 提取所确定的所述上线分支中与所述当前应用版本相应的提交记录,组合所获取的提交记录并打标。

[0130] 另外,在本发明实施例中所述的识别业务变更装置的具体实施内容,在上面所述

识别业务变更方法中已经详细说明了,故在此重复内容不再说明。

[0131] 本发明实施例所提供的装置,能够自行识别当前版本与历史上线版本中,所修改的Java代码中所涉及的方法名称/业务名称,并自动为所识别到的方法/业务添加报警监控。

[0132] 图3示出了可以应用本发明实施例的识别业务变更方法或识别业务变更装置的示例性系统架构300。

[0133] 如图3所示,系统架构300可以包括终端设备301、302、303,网络304和服务器305(仅仅是示例)。网络304用以在终端设备301、302、303和服务器305之间提供通信链路的介质。网络304可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0134] 用户可以使用终端设备301、302、303通过网络304与服务器305交互,以接收或发送消息等。终端设备301、302、303上可以安装有各种通讯客户端应用,例如购物类应用、网页浏览器应用、搜索类应用、即时通信工具、邮箱客户端、社交平台软件等(仅为示例)。

[0135] 终端设备301、302、303可以是具有显示屏并且支持网页浏览的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等。

[0136] 服务器305可以是提供各种服务的服务器,例如对用户利用终端设备301、302、303所浏览的购物类网站提供支持的后台管理服务器(仅为示例)。后台管理服务器可以对接收到的产品信息查询请求等数据进行分析等处理,并将处理结果(例如目标推送信息、产品信息—仅为示例)反馈给终端设备。

[0137] 需要说明的是,本发明实施例所提供的识别业务变更方法一般由服务器305执行,相应地,识别业务变更装置一般设置于服务器305中。

[0138] 应该理解,图3中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器的。

[0139] 下面参考图4,其示出了适于用来实现本发明实施例的终端设备的计算机系统400的结构示意图。图4示出的终端设备仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0140] 如图4所示,计算机系统400包括中央处理单元(CPU)401,其可以根据存储在只读存储器(ROM)402中的程序或者从存储部分408加载到随机访问存储器(RAM)403中的程序而执行各种适当的动作和处理。在RAM 403中,还存储有系统400操作所需的各种程序和数据。CPU 401、ROM 402以及RAM 403通过总线404彼此相连。输入/输出(I/O)接口405也连接至总线404。

[0141] 以下部件连接至I/O接口405:包括键盘、鼠标等的输入部分406;包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分407;包括硬盘等的存储部分408;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分409。通信部分409经由诸如因特网的网络执行通信处理。驱动器410也根据需要连接至I/O接口405。可拆卸介质411,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器410上,以便于从其上读出的计算机程序根据需要被安装入存储部分408。

[0142] 特别地,根据本发明公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本发明公开的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在

这样的实施例中,该计算机程序可以通过通信部分409从网络上被下载和安装,和/或从可拆卸介质411被安装。在该计算机程序被中央处理单元(CPU)401执行时,执行本发明的系统中限定的上述功能。

[0143] 需要说明的是,本发明所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本发明中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本发明中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0144] 附图中的流程图和框图,图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0145] 描述于本发明实施例中所涉及到的模块可以通过软件的方式实现,也可以通过硬件的方式来实现。所描述的模块也可以设置在处理器中,例如,可以描述为:一种处理器包括版本确定模块、代码识别模块、业务确定模块。其中,这些模块的名称在某种情况下并不构成对该模块本身的限定,例如,代码识别模块还可以被描述为“识别变更代码的模块”。

[0146] 作为另一方面,本发明还提供了一种计算机可读介质,该计算机可读介质可以是上述实施例中描述的设备中所包含的;也可以是单独存在,而未装配入该设备中。上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该设备执行时,使得该设备包括:

[0147] 获取当前应用版本的上线时间戳,确定与所述上线时间戳相邻最近的历史应用版本;

[0148] 比较所述当前应用版本以及所述历史应用版本中的代码,识别所述当前应用版本中的变更代码;

[0149] 根据与所述当前应用版本相应的代码与业务的对应关系,确定所述当前应用版本中的变更业务;其中,所述业务为所述当前应用版本可操作执行的最小粒度。

[0150] 根据本发明实施例的技术方案,能够自行识别当前版本与历史上线版本中,所修改的Java代码中所涉及的方法名称/业务名称,并自动为所识别到的方法/业务添加报警监控。

[0151] 上述具体实施方式,并不构成对本发明保护范围的限制。本领域技术人员应该明白的是,取决于设计要求和因素,可以发生各种各样的修改、组合、子组合和替代。任何在本发明的精神和原则之内所作的修改、等同替换和改进等,均应包含在本发明保护范围之内。

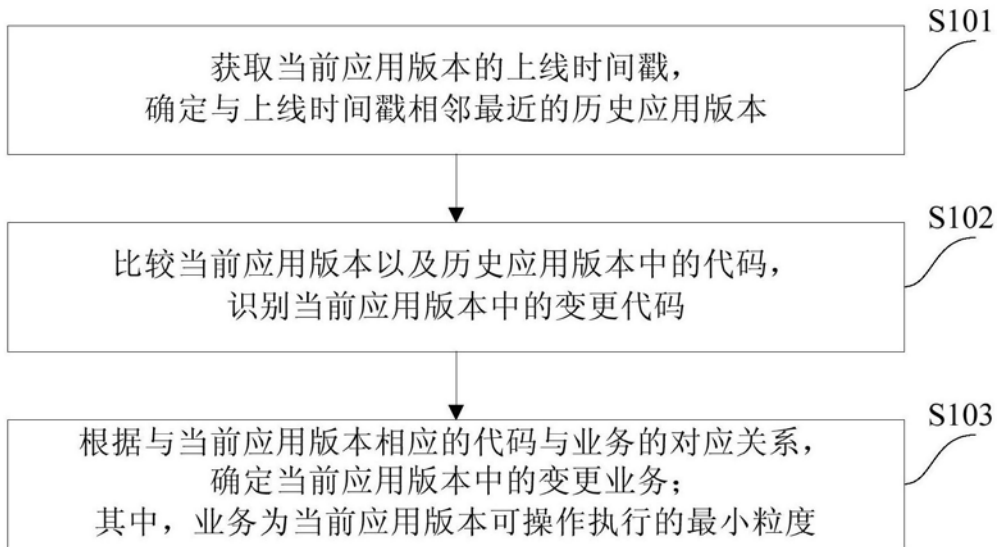


图1

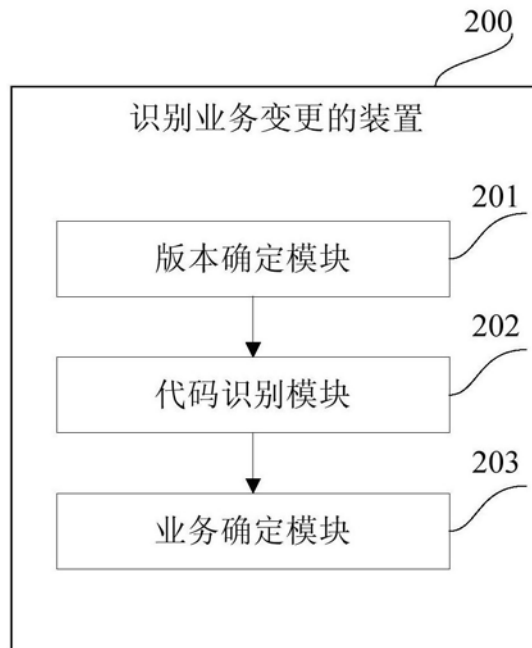


图2

300

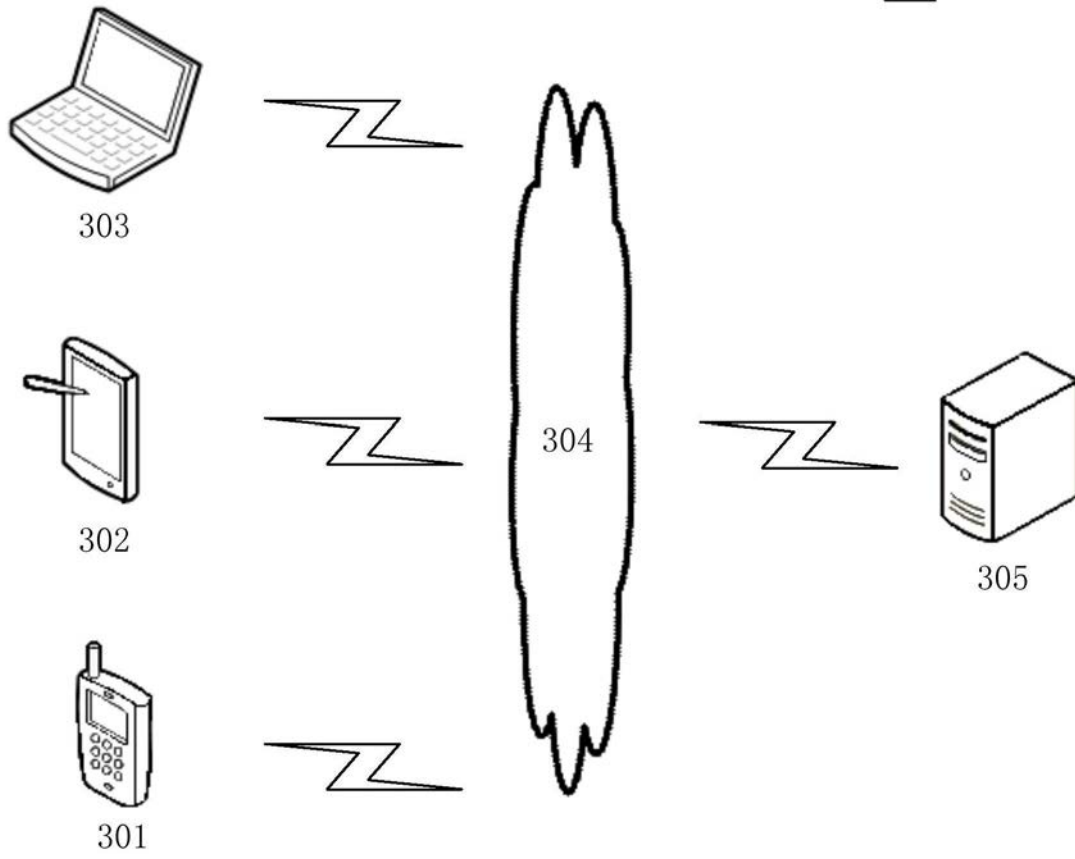


图3

400

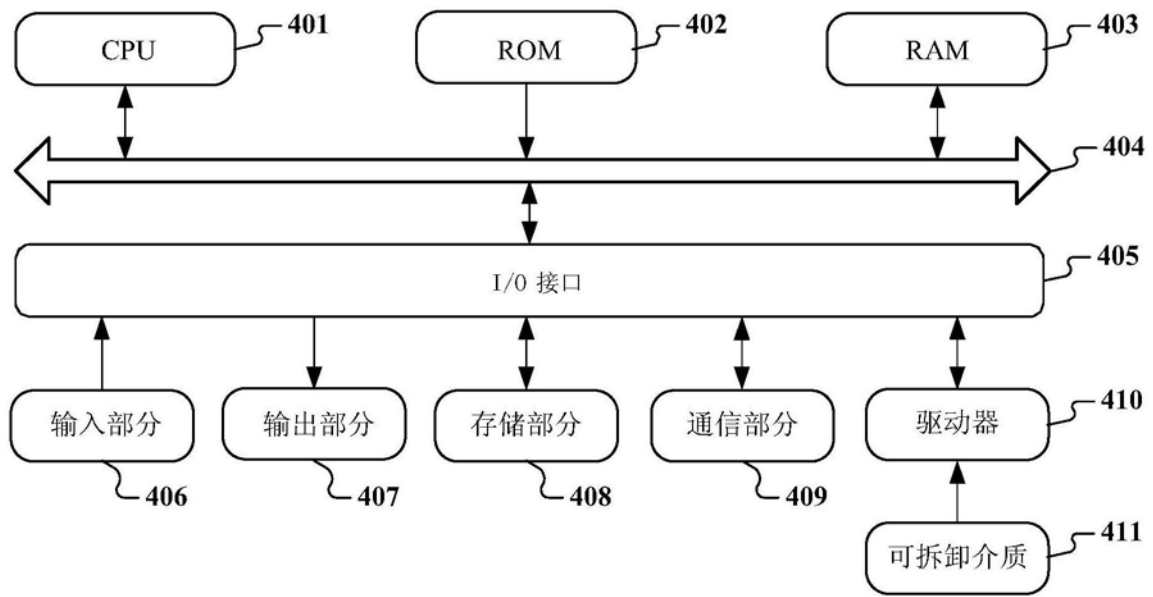


图4