



(12)发明专利申请

(10)申请公布号 CN 109144517 A

(43)申请公布日 2019.01.04

(21)申请号 201810866286.X

(22)申请日 2018.08.01

(71)申请人 北京京东金融科技控股有限公司
地址 101111 北京市北京经济技术开发区
科创十一街18号C座2层221室

(72)发明人 钟雅峰

(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 8/41(2018.01)

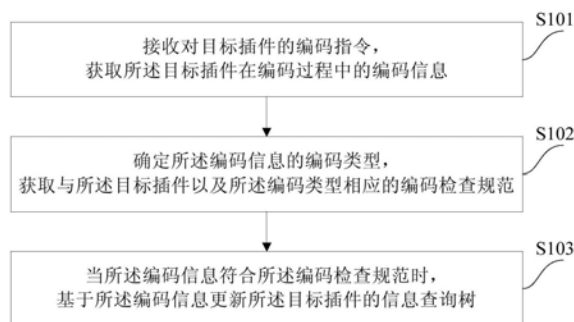
权利要求书2页 说明书10页 附图4页

(54)发明名称

一种检查编码的方法和装置

(57)摘要

本发明公开了一种检查编码的方法和装置，涉及计算机技术领域。该方法的一具体实施方式包括：接收对目标插件的编码指令，获取目标插件在编码过程中的编码信息；确定编码信息的编码类型，获取与所述目标插件以及所述编码类型相应的编码检查规范；当编码信息符合编码检查规范时，基于编码信息更新目标插件的信息查询树。该实施方式在编码的过程中对编码进行检查，增强了编码处理的及时性，便于后续及时修改，提高整体编码的有效性；整体过程人工参与部分较少，减少了人工成本、并降低误判的情况；根据规范模板编辑规范，可以有效降低规范编写成本，同时提高了作业人员的沟通便捷性，易扩展、易编写。



1. 一种检查编码的方法,其特征在于,包括:
接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;
确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;
当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树。
2. 根据权利要求1所述的方法,其特征在于,所述确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范,包括:
微粒化所述编码信息,根据预定正则方式,分析微粒化后编码信息的微粒类型,以确定与所述目标插件以及所述微粒类型相应的编码检查规范;其中,所述微粒类型至少包括类、变量、方法、函数、属性以及宏定义。
3. 根据权利要求1所述的方法,其特征在于,所述当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树,还包括:
遍历所述信息查询树,获取所述信息查询树中与所述编码类型相应的树节点,当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。
4. 根据权利要求3所述的方法,其特征在于,所述当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树,包括:
确定所述编码信息的关系摘要,当所述关系摘要与所述节点信息的关系摘要的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。
5. 根据权利要求1所述的方法,其特征在于,所述基于所述编码信息更新所述目标插件的信息查询树之后,还包括:
编译所述编码信息以生成编译信息,当所述编译信息符合与所述目标插件相应的编译检查规范时,基于所述编译信息更新所述信息查询树。
6. 根据权利要求1-5中任一项所述的方法,其特征在于,在所述接收对目标插件的编码指令之前,还包括:
当检测到所述目标插件第一次启动、且不存在相应的检查规范时,根据预定检查规范模板以及所述目标插件的预定参数,生成相应的检查规范;其中,所述检查规范至少包括所述编码检查规范以及所述编译检查规范。
7. 根据权利要求1-5中任一项所述的方法,其特征在于,
所述编码检查规范至少包括类规范、函数规范、属性规范、宏定义规范以及代码块规范;
所述编译检查规范至少包括词法规范、语法规则以及语意规范。
8. 一种检查编码的装置,其特征在于,包括:
信息获取模块,用于接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;
规范获取模块,用于确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;
查询树更新模块,用于当所述编码信息符合所述编码检查规范时,基于所述编码信息

更新所述目标插件的信息查询树。

9. 根据权利要求8所述的装置,其特征在于,所述规范获取模块,用于:

微粒化所述编码信息,根据预定正则方式,分析微粒化后编码信息的微粒类型,以确定与所述目标插件以及所述微粒类型相应的编码检查规范;其中,所述微粒类型至少包括类、变量、装置、函数、属性以及宏定义。

10. 根据权利要求8所述的装置,其特征在于,所述查询树更新模块,用于:

遍历所述信息查询树,获取所述信息查询树中与所述编码类型相应的树节点,当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

11. 根据权利要求10所述的装置,其特征在于,所述查询树更新模块,用于:

确定所述编码信息的关系摘要,当所述关系摘要与所述节点信息的关系摘要的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

12. 根据权利要求8所述的装置,其特征在于,所述查询树更新模块,还用于:

编译所述编码信息以生成编译信息,当所述编译信息符合与所述目标插件相应的编译检查规范时,基于所述编译信息更新所述信息查询树。

13. 根据权利要求8-12中任一项所述的装置,其特征在于,还包括规范生成模块,用于:

当检测到所述目标插件第一次启动、且不存在相应的检查规范时,根据预定检查规范模板以及所述目标插件的预定参数,生成相应的检查规范;其中,所述检查规范至少包括所述编码检查规范以及所述编译检查规范。

14. 根据权利要求8-12中任一项所述的装置,其特征在于,

所述编码检查规范至少包括类规范、函数规范、属性规范、宏定义规范以及代码块规范;

所述编译检查规范至少包括词法规范、语法规则以及语意规范。

15. 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1-7中任一所述的方法。

16. 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1-7中任一所述的方法。

一种检查编码的方法和装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种检查编码的方法和装置。

背景技术

[0002] 目前iOS开发工具主要是XCode,在编码的过程中,只能提示一些简单的词法分析结果。但对于编码规范,只能通过人工审核才能发现问题。除此之外,可以在编码结束后,通过一些专业的代码分析工具,对整个工程进行扫描分析,最后得出分析结果。

[0003] 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:

[0004] 1) 需要借助人力检查代码,极大增加了人工成本,也会存在检查遗漏、检查错误等问题;

[0005] 2) 编码后再使用工具做扫描,实时性差,问题处理具有滞后性,无法在编码期间及时发现问题并解决;

[0006] 3) 通常的静态扫描工具只有一些简单的约定,而往往不能满足实际开发需求,但有些工具无法满足扩展要求,即使扩展,实现也不易。因此对于使用工具扫描,规则自定义不强,编写规则成本较高,后期维护困难。

发明内容

[0007] 有鉴于此,本发明实施例提供一种检查编码的方法和装置,至少能够解决现有技术中人工成本高、处理滞后、维护困难等问题。

[0008] 为实现上述目的,根据本发明实施例的一个方面,提供了一种检查编码的方法,包括:

[0009] 接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;

[0010] 确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;

[0011] 当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树。

[0012] 可选的,所述确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范,包括:

[0013] 微粒化所述编码信息,根据预定正则方式,分析微粒化后编码信息的微粒类型,以确定与所述目标插件以及所述微粒类型相应的编码检查规范;其中,所述微粒类型至少包括类、变量、方法、函数、属性以及宏定义。

[0014] 可选的,所述当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树,还包括:

[0015] 遍历所述信息查询树,获取所述信息查询树中与所述编码类型相应的树节点,当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

[0016] 可选的,所述当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树,包括:

[0017] 确定所述编码信息的关系摘要,当所述关系摘要与所述节点信息的关系摘要的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

[0018] 可选的,所述基于所述编码信息更新所述目标插件的信息查询树之后,还包括:编译所述编码信息以生成编译信息,当所述编译信息符合与所述目标插件相应的编译检查规范时,基于所述编译信息更新所述信息查询树。

[0019] 可选的,在所述接收对目标插件的编码指令之前,还包括:当检测到所述目标插件第一次启动、且不存在相应的检查规范时,根据预定检查规范模板以及所述目标插件的预定参数,生成相应的检查规范;其中,所述检查规范至少包括所述编码检查规范以及所述编译检查规范。

[0020] 可选的,所述编码检查规范至少包括类规范、函数规范、属性规范、宏定义规范以及代码块规范;所述编译检查规范至少包括词法规范、语法规则以及语意规范。

[0021] 为实现上述目的,根据本发明实施例的另一方面,提供了一种检查编码的装置,包括:

[0022] 信息获取模块,用于接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;

[0023] 规范获取模块,用于确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;

[0024] 查询树更新模块,用于当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树。

[0025] 可选的,所述规范获取模块,用于:微粒化所述编码信息,根据预定正则方式,分析微粒化后编码信息的微粒类型,以确定与所述目标插件以及所述微粒类型相应的编码检查规范;其中,所述微粒类型至少包括类、变量、装置、函数、属性以及宏定义。

[0026] 可选的,所述查询树更新模块,用于:遍历所述信息查询树,获取所述信息查询树中与所述编码类型相应的树节点,当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

[0027] 可选的,所述查询树更新模块,用于:确定所述编码信息的关系摘要,当所述关系摘要与所述节点信息的关系摘要的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

[0028] 可选的,所述查询树更新模块,还用于:编译所述编码信息以生成编译信息,当所述编译信息符合与所述目标插件相应的编译检查规范时,基于所述编译信息更新所述信息查询树。

[0029] 可选的,还包括规范生成模块,用于:当检测到所述目标插件第一次启动、且不存在相应的检查规范时,根据预定检查规范模板以及所述目标插件的预定参数,生成相应的检查规范;其中,所述检查规范至少包括所述编码检查规范以及所述编译检查规范。

[0030] 可选的,所述编码检查规范至少包括类规范、函数规范、属性规范、宏定义规范以及代码块规范;

[0031] 所述编译检查规范至少包括词法规范、语法规则以及语意规范。

[0032] 为实现上述目的,根据本发明实施例的再一方面,提供了一种检查编码的电子设
备。

[0033] 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检查编码的方法。

[0034] 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检查编码的方法。

[0035] 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:在编码的过程中对编码进行检查,增强了编码处理的及时性,便于后续及时修改,提高
整体编码的有效性;整体过程人工参与部分较少,减少了人工成本、并降低误判的情况;根
据规范模板编辑制定规范,可以有效降低规范编写成本,同时提高了作业人员的沟通便捷
性,易扩展、易编写。

[0036] 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

[0037] 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:

[0038] 图1是根据本发明实施例的一种检查编码方法的主要流程示意图;

[0039] 图2是本发明查询树的示意图;

[0040] 图3是根据本发明实施例的一种检查编码装置的主要模块示意图;

[0041] 图4是本发明实施例可以应用于其中的示例性系统架构图;

[0042] 图5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

[0043] 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。

[0044] 需要说明的是,本发明实施例目前仅适用于IOS系统,对于安卓、Windows等系统,
本发明暂不考虑。

[0045] 对于本发明所涉及的词语做解释如下:

[0046] iOS:苹果公司所开发的移动操作系统。

[0047] Xcode:苹果公司向开发人员提供的集成开发环境(非开源),用于开发Mac OS X、
iOS的应用程序。

[0048] 词法分析结果:是编译过程的第一个阶段,是编译的基础。这个阶段的任务是从左
到右逐个字符地读入源程序,即对构成源程序的字符流进行扫描然后根据构词规则识别单
词(也称单词符号或符号)。

[0049] 编码规范:只是大家在程序范围内达成一致的约定,这样大家的代码就可以互相

看懂,维护起来更加容易,思想更畅快的交流,经验更快的得到传播。

[0050] 树:树状图是一种数据结构,它是由 $n(n \geq 1)$ 个有限节点组成一个具有层次关系的集合,看起来像是一棵倒挂的树。树中每个节点有零个或多个子节点;没有父节点的节点称为根节点;每一个非根节点有且只有一个父节点。

[0051] 抽象语法树(Abstract Syntax Tree,AST)或者语法树(syntax tree):是源代码的抽象语法结构的树状表现形式,这里特指编程语言的源代码。抽象语法树作为程序的一种中间表示形式,在程序分析等诸多领域有广泛的应用。利用抽象语法树可以方便地实现多种源程序处理工具,比如源程序浏览器、智能编辑器、语言翻译器等

[0052] 具体语法树(concrete syntaxtree),通常称作分析树(parse tree):和抽象语法树相对。一般的,在源代码的翻译和编译过程中,语法分析器创建出分析树。

[0053] 参见图1,示出的是本发明实施例提供的一种检查编码的方法的主要流程图,包括如下步骤:

[0054] S101:接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;

[0055] S102:确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;

[0056] S103:当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树。

[0057] 本发明包含两部分:代码规范的制定以及代码规范的检查。编码在代码规范检查阶段,主要针对于编码和编译两个阶段进行检查,而两者较为重要的是编码检查。

[0058] 上述实施方式中,对于步骤S101,对于插件程序(后面简称插件)启动后的编码输入识别,主要依据编码文件的MD5(Message Digest Algorithm MD5,文件摘要)进行。

[0059] 每个编码文件都有一个自己的MD5摘要,类似于“数字指纹”。当文件内容发生修改时,相应的MD5摘要,也就是对应的“数字指纹”也会发生修改。

[0060] 另外,为解决现有技术中编码结束后再检查的非实时问题,本发明采用边编码边检查的方式,对于异常编码,可以在编码的过程中解决。

[0061] 对于步骤S102,当插件识别到有编码输入时,会自动识别编码检查规范。

[0062] 例如,当编码类型为方法类型时,所获得的编码检查规范为方法规范。具体地:

[0063] 1) 将编码信息的文件内容微粒成节点信息(或微粒信息);其中,节点信息是指文件被微粒化的细小单元,例如,类、函数、变量等;

[0064] 2) 依据正则方式,判断当前的编码内容是属于何种微粒。例如,当输入@insterface xxxx:NSObject这时就可以知道xxxx是一个类名。

[0065] 插件在第一次启动时,会在指定位置处,例如C盘programs处,查询是否存在相应规范,例如,插件1-规范1。也可以是单独设置一个文件夹,用以存储各个插件的规范;或是在各个插件的安装位置处,例如文件夹XX,仅存储该插件的规范。

[0066] 需要说明的是,插件执行所使用的规范,可以与系统的默认规范所处位置不同。其中,系统所默认的规范可以视为是源模板(例如行业类标准),例如包含1~10个不同的类型规范,而插件1的规范根据该模板所生成,例如仅需要其中5个。

[0067] 另外,由于编码检查规范是对编码的检查,规范所存储的位置可以与编码所缓冲/存储的位置不同,以避免编码被篡改的情况。

[0068] 通常情况下,插件所有的自定义规范以及所选择的规范,都会保存在相应指定位置处。若该位置没有规范,证明开发者没有配置规范。因此,在插件第一次启动、且没有规范的情况下,需要制定规范。

[0069] 传统的代码分析工具,规范的制定需要由专业人员编写,本发明摒弃这一思想,建立规范模板。任何开发者或者测试者,都可以通过界面选择的方式,自主配置规范。

[0070] 针对于插件的编码进程以及编译进程,可以将规范拆分为两类:

[0071] 1) 编码检查规范:主要分为类规范、函数规范、属性规范、宏定义规范以及代码块规范等;

[0072] 2) 编译检查规范:主要分为词法规范、语法规则、语意规范等。

[0073] 需要说明的是,对于各规范中的参数,可以根据具体显示要求设置格式,之后再形成具体的规范,例如,类名限制5~20个字符,函数名限制在5~20个字符等。插件在运行时,就可以根据这个规范去检查编码。

[0074] 另外,针对规范/模板不全的问题,可以根据本发明的规范制定思路,先编写特定模板,然后再生成特定的规范。有了规范之后,在插件启动时,会自动加载这些规范,以便在后续的编码和编译时通过规范检查代码。

[0075] 对于步骤S103,在确定编码检查规范后,即进行编码检查阶段。

[0076] 对于编码检查阶段,需要遍历查询树中的所有节点信息。为减少遍历工作量,只针对性的遍历与编码类型相应的树节点即可。例如,编码类型为类方法,则需要首先确定类型为类的树节点,然后再从中挑出类型属于方法的树节点。

[0077] 之后,对于编码信息的检查,即是将编码信息与所筛选到的树节点的节点信息进行相似度比对,以判断是否为违规信息。

[0078] 1) 当没有节点信息与编码信息相似、或者相似度都低于预定阈值(例如,50%)时,即表示该编码信息查询树中没有,可以将其添加至查询树中;

[0079] 2) 但所有节点信息与编码信息的相似度超出预定阈值时,可以添加该编码信息至查询树中、或者替换原节点信息等,也可以不添加;例如,同样的要求节点中的编码A可以实现、本次编码的编码B也可以实现,则无需替换。

[0080] 本发明中的查询树使用OC、C++语言做对比,也可以使用类似python语言做文字比对。

[0081] 为了加速查找速度,所有文件、类、函数、变量等信息,在插件启动时,会以字典形式组织成查询树,后续检索主要是根据字母方式检索,查询树的形式参见图2所示。

[0082] 需要说明的是,除了字典形式构建查询树之外,还可以是关键字等其他形式。其原理与使用字典查询汉字类似,先按照汉字拼音的首字母查询,然后在所得范围中查找次字母,以此类推,最后找到需要的汉字。

[0083] 之后所有的编码检查,会依据这个查询树进行。每次查询时,会根据当前的编码内容,找到相应的树节点,然后比对规范,最后做出判断。

[0084] 进一步的,在确定树节点、比对规范前,还需要对树节点进行遍历查询。因为规范是根据编码的微粒信息来指定的,例如,判断两个方法是否相同或相似,就需要对比项目中所涉及的所有文件,但判断对象是方法,故只需要考虑方法定义这部分,而不需要考虑类、变量、宏等其他信息。因此,在实际判断中,只需要将项目中所设计的所有文件、以及文件中

方法这部分信息提取出来就可以了。

[0085] 另外,不同微粒(类、方法、变量等)的对比方式可以是不同的。对于编码信息是否符合规范的比对,需要针对不同的情况具体实施:

[0086] 1) 变量一般只涉及到文件内,因此变量的比对只是简单的对规范阈值进行比对(方法名、参数等与变量类似);例如,变量是否在5~10个字符内、是否以规定的字母为开头、是否是指定大写/小写等;

[0087] 2) 方法的比对较为复杂,有些规范阈值是需要实现计算得到,之后再对比;

[0088] 例如,方法的相似度,在查询树中,方法下面有代码块的节点,这个节点是将代码微粒化,以“{}”为块,去掉空格、换行等无关信息,只留下类型、变量及其关系等,组成一个关系摘要。

[0089] 每次对比方法时,就对比这个关系摘要,得出一个相似度。然后对比规范预先设定的相似度阈值,判断两个方法是否相同。例如:

```

[0090]     -(void) printInfo:(NSString*)info {
            if (info.count == 10) {
                print("A");
            }
            else{
                print("B");
            }
        }
[0091] 所得关系摘要为:-v m(s) {if(==)do el do re}
[0092]     -(int) getInfo:(NSString*)message {
            if (message.count > 10) {
                return 10;
            }
            else{
                return 0;
            }
        }

```

[0093] 所得关系摘要为:-i m(s) {if(>)re el re};然后对比两者的关系摘要,来计算两者代码的相似度。

[0094] 当用户输入编码完毕之后,所有所输入的编码信息,将会自动加入到查询树中,每次修改内容,查询树的信息也会相应改变。

[0095] 需要说明的是,修改是文件被保存的特殊时刻,每个文件被保存时,会有一个MD5。当文件被修改后,这个MD5就会变化。然后根据所修改的位置以及所修改的内容,对查询树

进行更新。这里的修改每次都只会影响到新加内容的微粒对应的节点,其他节点是不需要修改和变化的,因此,每次对比和调整的信息不会很多。

[0096] 在加入编码信息至树节点完毕之后,会自动进入编译流程。当插件识别到开始编译时,会自动收集编译信息,待编译结束后,根据编译检查规范检查所编译的信息,最后提示分析结果。

[0097] 编译器在编译时,会生成编译日志,因此可以根据是否有编译日志判断是否开始编译。编译日志中记载有编译的时间、编译方式等信息。

[0098] 编译模板主要依据抽象语法树,借助编译工具(例如XCode开发工具)在编译代码时将源码分析成抽象语法树,然后遍历语法树节点,分析节点信息(例如,变量、方法名等),对比规则,记录违规信息。

[0099] 上述实施例所提供的方法,在编码的过程中对编码进行检查,增强了编码处理的及时性,便于后续及时修改,提高整体编码的有效性;整体过程人工参与部分较少,减少了人工成本、并降低误判的情况;根据规范模板编辑规范,可以有效降低规范编写成本,同时提高了作业人员的沟通便捷性,易扩展、易编写。

[0100] 参见图3,示出了本发明实施例提供的一种检查编码的装置200的主要模块示意图,包括:

[0101] 信息获取模块301,用于接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;

[0102] 规范获取模块302,用于确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;

[0103] 查询树更新模块303,用于当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件的信息查询树。

[0104] 本发明实施装置中,所述规范获取模块302,用于:微粒化所述编码信息,根据预定正则方式,分析微粒化后编码信息的微粒类型,以确定与所述目标插件以及所述微粒类型相应的编码检查规范;其中,所述微粒类型至少包括类、变量、装置、函数、属性以及宏定义。

[0105] 本发明实施装置中,所述查询树更新模块303,用于:遍历所述信息查询树,获取所述信息查询树中与所述编码类型相应的树节点,当所述编码信息与所述树节点中节点信息的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

[0106] 本发明实施装置中,所述查询树更新模块303,用于:确定所述编码信息的关系摘要,当所述关系摘要与所述节点信息的关系摘要的相似度低于预定阈值时,基于所述编码信息更新所述信息查询树。

[0107] 本发明实施装置中,所述查询树更新模块303,还用于:编译所述编码信息以生成编译信息,当所述编译信息符合与所述目标插件相应的编译检查规范时,基于所述编译信息更新所述信息查询树。

[0108] 本发明实施装置还包括规范生成模块304(图中未标出),用于:

[0109] 当检测到所述目标插件第一次启动、且不存在相应的检查规范时,根据预定检查规范模板以及所述目标插件的预定参数,生成相应的检查规范;其中,所述检查规范至少包括所述编码检查规范以及所述编译检查规范。

[0110] 本发明实施装置中,所述编码检查规范至少包括类规范、函数规范、属性规范、宏

定义规范以及代码块规范;所述编译检查规范至少包括词法规范、语法规则以及语意规范。

[0111] 另外,在本发明实施例中所述的检查编码装置的具体实施内容,在上面所述检查编码方法中已经详细说明了,故在此重复内容不再说明。

[0112] 图4示出了可以应用本发明实施例的检查编码方法或检查编码装置的示例性系统架构400。

[0113] 如图4所示,系统架构400可以包括终端设备401、402、403,网络404和服务器405(仅仅是示例)。网络404用以在终端设备401、402、403和服务器405之间提供通信链路的介质。网络404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。

[0114] 用户可以使用终端设备401、402、403通过网络404与服务器405交互,以接收或发送消息等。终端设备401、402、403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)。

[0115] 终端设备401、402、403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。

[0116] 服务器405可以是提供各种服务的服务器,例如对用户利用终端设备401、402、403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。

[0117] 需要说明的是,本发明实施例所提供的检查编码方法一般由服务器405执行,相应地,检查编码装置一般设置于服务器405中。

[0118] 应该理解,图4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。

[0119] 下面参考图5,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500的结构示意图。图5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。

[0120] 如图5所示,计算机系统500包括中央处理单元(CPU)501,其可以根据存储在只读存储器(ROM)502中的程序或者从存储部分508加载到随机访问存储器(RAM)503中的程序而执行各种适当的动作和处理。在RAM 503中,还存储有系统500操作所需的各种程序和数据。CPU 501、ROM 502以及RAM 503通过总线504彼此相连。输入/输出(I/O)接口505也连接至总线504。

[0121] 以下部件连接至I/O接口505:包括键盘、鼠标等的输入部分506;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507;包括硬盘等的存储部分508;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509。通信部分509经由诸如因特网的网络执行通信处理。驱动器510也根据需要连接至I/O接口505。可拆卸介质511,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510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508。

[0122] 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509从网络上被下载和安装,和/或从可

拆卸介质511被安装。在该计算机程序被中央处理单元(CPU) 501执行时,执行本发明的系统中限定的上述功能。

[0123] 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。

[0124] 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。

[0125] 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:一种处理器包括信息获取模块、规范获取模块、查询树更新模块。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,信息获取模块还可以被描述为“编码过程中的编码信息获取模块”。

[0126] 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:

[0127] 接收对目标插件的编码指令,获取所述目标插件在编码过程中的编码信息;

[0128] 确定所述编码信息的编码类型,获取与所述目标插件以及所述编码类型相应的编码检查规范;

[0129] 当所述编码信息符合所述编码检查规范时,基于所述编码信息更新所述目标插件

的信息查询树。

[0130] 根据本发明实施例的技术方案,在编码的过程中对编码进行检查,增强了编码处理的及时性,便于后续及时修改,提高整体编码的有效性;整体过程人工参与部分较少,减少了人工成本、并降低误判的情况;根据规范模板编辑规范,可以有效降低规范编写成本,同时提高了作业人员的沟通便捷性,易扩展、易编写。

[0131] 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。

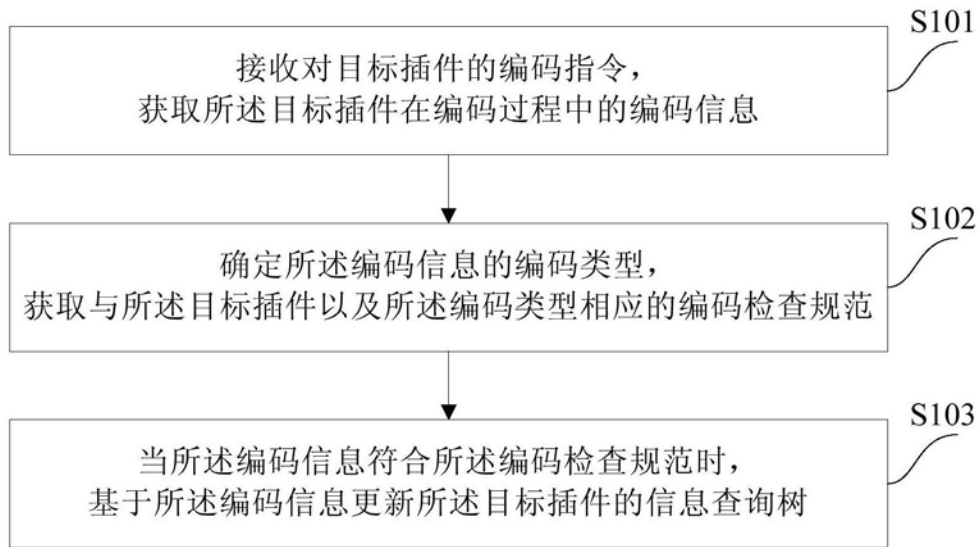


图1

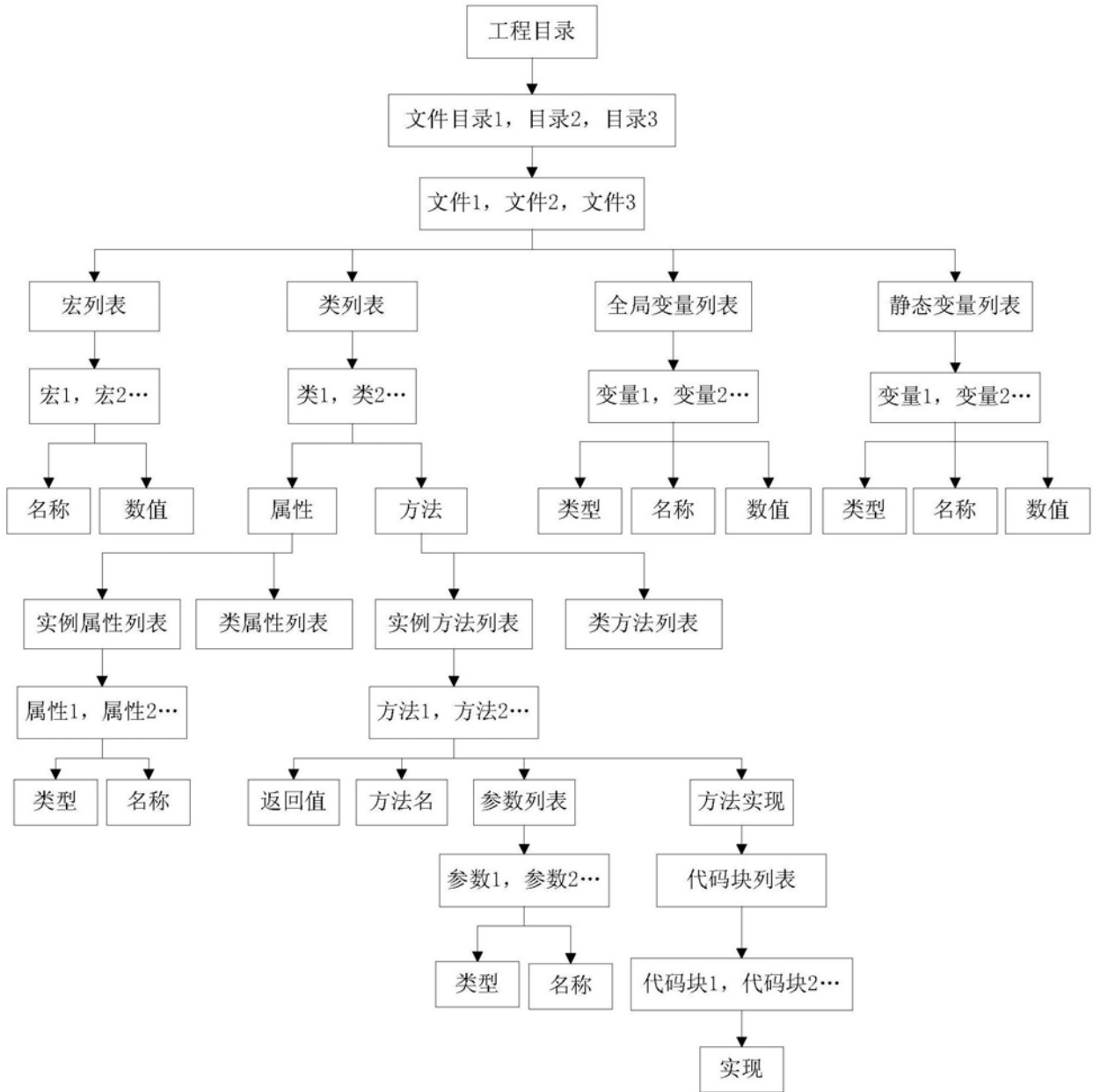


图2

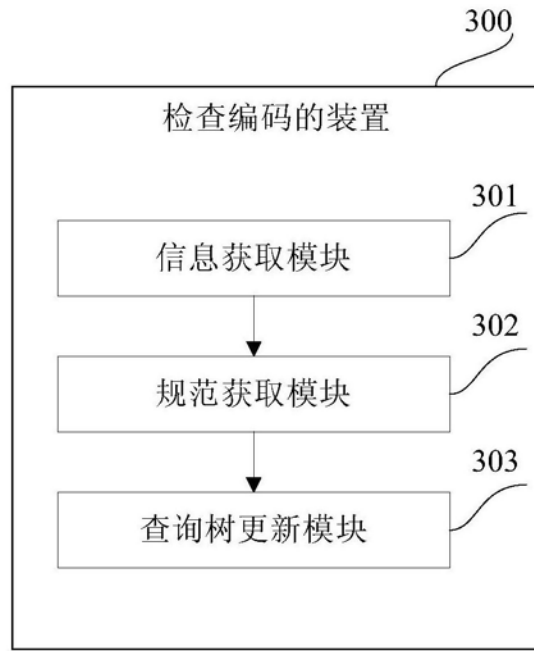


图3

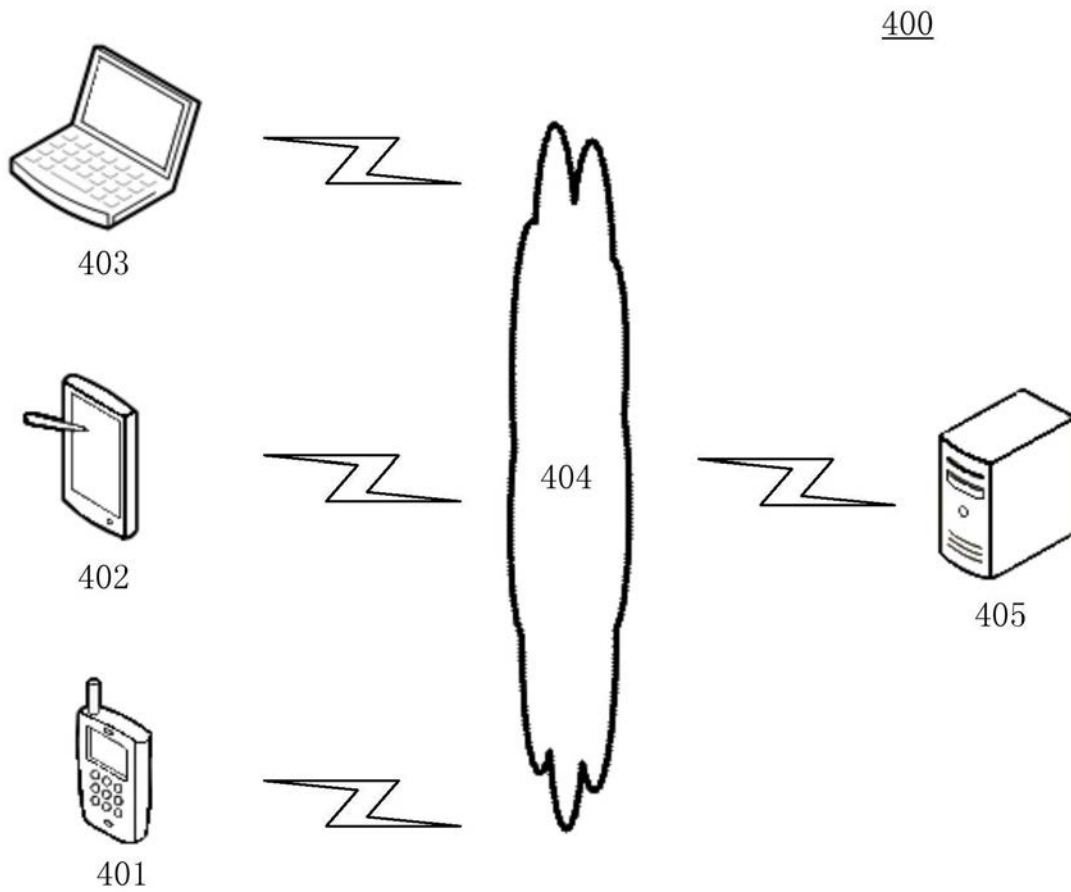


图4

500

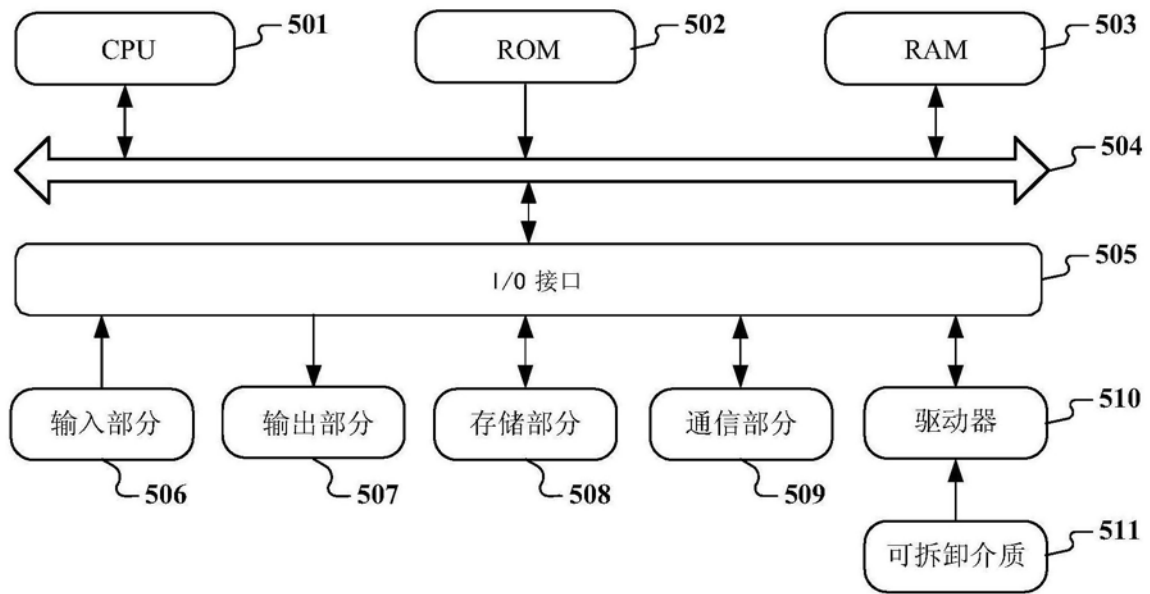


图5