

(12) 发明专利申请

(10) 申请公布号 CN 102722442 A

(43) 申请公布日 2012. 10. 10

(21) 申请号 201210193257. 4

(22) 申请日 2012. 06. 13

(71) 申请人 禹芙蓉

地址 518000 广东省深圳市南山区科技园科
发路 1 号

(72) 发明人 不公告发明人

(74) 专利代理机构 北京科亿知识产权代理事务
所（普通合伙） 11350

代理人 汤东凤

(51) Int. Cl.

G06F 11/36 (2006. 01)

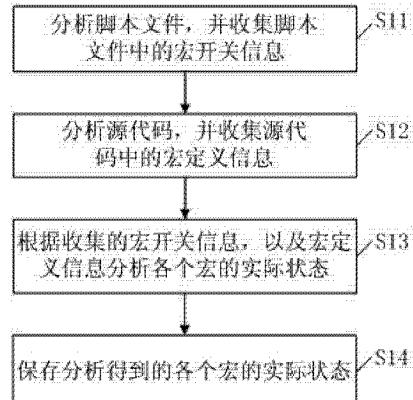
权利要求书 1 页 说明书 5 页 附图 2 页

(54) 发明名称

一种宏状态自动分析方法及其装置

(57) 摘要

本发明实施例提供了一种宏状态自动分析方法，通过分析脚本文件，并收集脚本文件中的宏开关信息；以及通过分析源代码，并收集源代码中的宏定义信息，再根据收集的该宏开关信息和宏定义信息分析各个宏的实际状态，并存储宏的实际状态，从而实现宏状态的自动分析，不需要编译验证，也不需要搜索后再进行筛选，进而提高了效率，也提高了可靠性。相应地，本发明实施例还提供了一种对应的宏状态自动分析装置。



1. 一种宏状态自动分析方法,其特征在于,包括:

分析脚本文件,并收集所述脚本文件中的宏的开关信息;

分析源代码,并收集所述源代码中的宏定义信息;

根据收集的所述宏的开关信息和所述宏定义信息,分析各个宏的实际状态;

保存分析得到的宏的实际状态。

2. 如权利要求1所述的宏状态自动分析方法,其特征在于,所述分析脚本文件,并收集所述脚本文件中的宏开关信息的步骤,具体包括:

解析相关脚本,得到预定义宏信息;

根据所述预定义宏信息,分析得到各个宏的开关信息,并收集各个宏的开关信息。

3. 如权利要求1所述的宏状态自动分析方法,其特征在于,所述根据所收集的各个宏的开关信息和所述宏定义信息,分析得到各个宏的实际状态的步骤,具体为:

根据各个宏的开关信息和宏定义信息,对当前状态显示为未定义的宏进行分析,并得到所述当前状态显示为未定义的宏的实际状态。

4. 如权利要求1至3中任意一项所述的宏状态自动分析方法,其特征在于,还包括:

根据保存的宏的实际状态,更新源代码中的宏的状态。

5. 一种宏状态自动分析装置,其特征在于,包括:

脚本分析模块,用于分析脚本文件,得到所述脚本文件中的宏的开关信息;

源代码分析模块,用于分析源代码,得到所述源代码中的宏定义信息;

信息收集模块,用于收集所述脚本分析模块分析得到的所述宏的开关信息,以及所述源代码分析模块分析得到的所述宏定义信息;

状态分析模块,用于根据所述信息收集模块收集的所述宏的开关信息和所述宏定义信息,分析各个宏的实际状态;

存储模块,用于存储所述状态分析模块分析得到的宏的实际状态。

6. 如权利要求5所述的宏状态自动分析装置,其特征在于,所述脚本分析模块包括:

解析子模块,用于解析相关脚本,得到预定义宏信息;

分析子模块,用于根据所述解析子模块解析得到的预定义宏信息,分析得到宏的开关信息。

7. 如权利要求5所述的宏状态自动分析装置,其特征在于,所述状态分析模块包括:

选择子模块,用于从各个宏当中选择当前状态显示为未定义的宏;

分析子模块,用于根据所述宏的开关信息和所述宏定义信息,对所述选择子模块所选择的宏的实际状态进行分析,并得到该宏的实际状态。

8. 如权利要求5至7中任意一项所述的宏状态自动分析装置,其特征在于,还包括:

更新子模块,用于根据存储模块存储的宏的实际状态,更新源代码中宏的状态。

一种宏状态自动分析方法及其装置

技术领域

[0001] 本发明涉及嵌入式领域,尤其涉及一种宏状态自动分析方法及其装置。

背景技术

[0002] 随着科技的发展,电子设备广泛应用到人们的日常生活中。电子产品的功能离不开软件的开发,如今嵌入式开发是最方便的方式。在嵌入式开发过程中,需要广泛的使用到宏,其与编译命令、配置环境等相关。由于宏既可以定义在源代码中,如 *.c, *.cpp, *.h 文件中,也可以定义在 makefile, python 脚本中,如 *.mk, *.scons。因此,在查看代码时,需要查看关联的文件才能判断宏是否定义,从而需要花很长时间。

[0003] 目前,代码查看工具为 Source Insight。Source Insight 是一种支持多种开发语言(java, c, c++, x86 汇编语言程序等等)的编辑器,其查找、定位、彩色显示等功能的强大。由于其内置对 C/C++, Java 和 x86 汇编语言程序的解析,并且有自己的动态数据库,因此,其不仅能分析源代码,并在工程师工作的同时动态维护它自己的符号数据库,且自动显示有用的上下文信息,包括有关的函数,宏,参数等。查看代码时,只要在源代码中 (*.C, *.Cpp, *.h) 定义了宏,在 Source Insight 中就显示为红色,表示宏已经定义,不管其前提是否有条件;如果是在汇编里或 make file 等脚本里定义的宏,由于 Source Insight 不能够解析,因此,显示为黑色,表示宏未定义。

[0004] 但是在实际操作过程中,则存在两个问题:

- (1) 若显示为红色的宏的上级条件宏没有定义,则该宏实际上也没有定义;
- (2) 若显示为黑色的宏是定义在汇编或 make file 等脚本里,则该宏实际上已经定义。

[0005] 因此,使用当前的 Source Insight 工具,并不能够得到宏的真实状态。

[0006] 一般嵌入式开发工程师在碰到上述问题时,只有两种方法处理:

- 1、故意写错误语法代码,编译验证;
- 2、使用软件搜索功能搜索,却不能使用全字符匹配,在搜完了之后,需要再做筛选。

[0007] 这两种方法需要花大量的时间去分析,判断宏是否开启,效率较低,可靠性也较低。

发明内容

[0008] 本发明实施例所要解决的技术问题在于,提供一种宏状态自动分析装置,能够更有效地得到宏的真实状态,且可靠性较高。

[0009] 为了解决上述技术问题,本发明实施例提供了一种宏状态自动分析方法,包括:

- 分析脚本文件,并收集所述脚本文件中的宏的开关信息;
- 分析源代码,并收集所述源代码中的宏定义信息;
- 根据收集的所述宏的开关信息和所述宏定义信息,分析各个宏的实际状态;
- 保存分析得到的宏的实际状态。

[0010] 相应地,本发明还提供了一种宏状态自动分析装置,包括:

脚本分析模块,用于分析脚本文件,得到所述脚本文件中的宏的开关信息;

源代码分析模块,用于分析源代码,得到所述源代码中的宏定义信息;

信息收集模块,用于收集所述脚本分析模块分析得到的宏的开关信息,以及所述源代码分析模块分析得到的所述宏定义信息;

状态分析模块,用于根据所述信息收集模块收集的所述宏的开关信息和所述宏定义信息,分析各个宏的实际状态;

存储模块,用于存储所述状态分析模块分析得到的宏的实际状态。

[0011] 实施本发明实施例,具有如下有益效果:

本发明实施例提供的一种宏状态自动分析方法及其装置,通过分析脚本文件得到脚本中宏的开关信息,同时通过分析源代码,得到源代码中的宏定义信息,并收集这两者,再根据收集的宏开关信息和宏定义信息来分析各个宏的实际状态,得到宏的实际状态,并存储,从而实现宏状态自动分析,不需要编译验证,也不需要搜索后再进行筛选,进而提高了效率,也提高了可靠性。

附图说明

[0012] 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的前提下,还可以根据这些附图获得其他的附图。

[0013] 图1是本发明的一种宏状态自动分析方法的一实施例的流程图;

图2是本发明的一种宏状态自动分析方法的又一实施例的流程图;

图3是本发明的一种宏状态自动分析装置的一实施例的功能模块图;

图4是本发明的一种宏状态自动分析状态的又一实施例的功能模块图。

具体实施方式

[0014] 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作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。

[0015] 参见图1,为本发明的一种宏状态自动分析方法的一种实施例的流程图。具体实施时,本实施例的宏状态自动分析方法的实施过程如下:

S11,分析脚本文件,并收集该脚本文件中的宏开关信息。具体实施时,该步骤S11的具体实施过程如下:解析相关脚本,得到预定义宏信息,该预定义宏信息包括宏名,以及各个宏之间的依赖关系等信息;根据各个宏之间的依赖关系、宏名等信息,分析得到宏的开关信息,并收集宏的开关信息。在嵌入式开发中,可由make, python等通过解析相关脚本,得到预定义宏信息等等。在一具体实施例中:

.....

```
#ifdef 宏 A
```

```
#define 宏 B
```

```
#eldef 宏 C  
#endif  
.....
```

SourceInsight 工具分析相关脚本文件得到预定义宏 B 和预定义宏 C 均依赖宏 A, 根据该依赖关系得知 : 如果定义了宏 A, 则预定义宏 B 的开关信息为开, 预定义宏 C 的开关信息为关 ; 如果未定义宏 A, 则预定义宏 C 的开关信息为开, 预定义宏 B 的开关信息为关。

[0016] 在另一具体实施例中, 执行该步骤 S11 之前, 在 SourceInsight 工具中需要行编译环境的配置, 即在 SourceInsight 工具中进行 *.cmd 文件或 *.bat 文件等脚本文件的设置, 其主要包括编译工具的路径, 产品的宏开关定义等。由于 cmd 或 bat 是整个项目编译的起点, 也是后面脚本关联的起点, 因此, 在编译环境的配置过程中, 还需要查找这些脚本文件中的宏定义, 并记录查找到的宏定义, 同时表明其控制范围, 从而便于分析脚本文件时得到正确的宏的开关信息。具体实施时, 由于 *.cmd 文件或 *.bat 文件等脚本文件中的宏定义是从编译命令的输入开始进行宏定义的, 因此, 可直接通过 cmd 命令语法来查找脚本文件中的宏定义。一个定义的宏, 有可见区域, 如在一个头文件 A 中定义一个宏, 在一个源文件 B 中使用这个宏, 如果 B 文件没有直接或间接包含这个头文件 A, 那么对于 B 来说, 这个宏是不可见的。本实施例中, 查找到的 Cmd、Bat 的宏定义, 为整个项目可见, 级别最高。

[0017] S12, 分析源代码, 并收集该源代码中的宏定义信息, 该宏定义信息包括 : 该宏是否定义在源代码中, 以及该宏是否存在上级条件宏等。一般宏定义在头文件中, 也有一些定义在源文件中, 因此, 需要通过分析源代码文件来收集该源代码中的宏定义信息。在一具体实施中, SourceInsight 工具分析源代码, 即在源代码中搜索各个宏, 并得到宏定义信息, 再将该宏定义信息收集起来。

[0018] S13, 根据所收集的宏开关信息和宏定义信息, 分析各个宏的实际状态。具体实施时, 由于各种头文件的包含关系和嵌入关系, 因此, 只需要根据步骤 S11 中分析得到的宏开关信息和步骤 S12 中分析得到的宏定义信息, 对当前状态显示为未定义的宏进行分析即可, 从而得到该宏的实际状态。在一具体实施例中, 由于各种头文件的包含关系和嵌入关系, 且在 SourceInsight 工具基础上, 只需要对 SourceInsight 工具中当前状态颜色显示为黑色的宏进行分析即可, 从而分析出该宏的实际状态 ; 同理, 若红色部分的上级条件宏为黑色, 当经过步骤 S13 分析即可得到该呈黑色的上级条件宏的实际状态, 从而得到呈红色的宏的实际状态。

[0019] 本实施例中的宏的状态包括 : 该宏已经定义 ; 该宏还未定义。

[0020] S14, 存储分析得到的宏的实际状态。具体实施时, 将分析的宏的实际状态, 加到头文件中保存, 从而使得 sourceinsight 这样的工具识别, 便于 sourceinsight 工具后续根据该头文件中的各个宏的实际状态重新的同步, 进而更新原来的宏的颜色, 并正确显示。

[0021] 本实施例中, 当所有处理后, 需要同步更新所有代码, 作最后处理, 因此, 如图 2 所示, 本实施例的宏状态自动分析方法还可包括步骤 :

S15, 根据保存得到的宏的实际状态, 更新源代码中宏的状态。具体实施时, 若 SourceInsight 工具中, 当前状态显示为黑色的宏的实际状态为已经定义, 则将源代码中该宏的状态更新为已经定义, 即在 SourceInsight 工具中将颜色由黑色更新为红色 ; 同理, 若当前状态显示为红色的宏的实际状态为未定义(其上级条件宏未定义), 则将源代码中该宏

的状态更新为未定义,即在 SourceInsight 工具中将颜色由红色更新为黑色。

[0022] 本实施例的宏状态自动分析方法,通过分析脚本文件,并收集脚本中宏的开关信息;以及通过分析源代码,并收集源代码中的宏定义信息,再根据该宏开关信息和宏定义信息分析各个宏的实际状态,即由 cmd 到 make 到源代码,这样一级一级解析下去,从而得到哪些宏是定义的,哪些未定义,实现了宏状态的自动分析,不需要编译验证,也不需要搜索后再进行筛选,进而提高了效率,也提高了可靠性。

[0023] 为了更高的可靠性,在另一具体实施例中,该宏状态自动分析方法除了包括上述实施例的步骤 S11 至 S14 之外,本实施例的宏状态自动分析方法,在执行步骤 S11 之前还可包括步骤:

获取项目自定义的参数,并解析输入参数,将其带入 makefile 文件。具体实施时,由于 cmd 是整个项目编译的起点,通常需要自定义 cmd 或 bat,而当一套代码对应多个项目时,则需要定义对应的产品、功能、特性宏等,这就需要通过简单的 cmd 带参数方式来解决,因此,项目自定义的参数也可能影响产品的功能开关,例如自定义了 gosomo.make :

```
.....
if /i "%GOSO_PROJECT_NAME%"=="ua6300"    goto ua6300
if /i "%GOSO_PROJECT_NAME%"=="ua1600"    goto ua1600
goto NoArg
.....
```

其中,gosomo.make 可输入参数 ua6300、ua1600 等。如果输入 ua1600,则对应 ua1600 的项目,定义 ua1600 的项目宏;如果输入 ua6300,则对应 ua6300 的项目宏。若要定义对应 ua6300 的项目宏,则通过 cmd.exe 输入命令: gosomo.make ua6300 即可。

[0024] 相应地,本发明实施例还提供了一种宏状态自动分析装置,下面将结合附图对其进行详细的说明。

[0025] 参见图 3,为本发明的一种宏状态自动分析装置的一种实施例的功能模块图。具体实施时,本实施例的宏状态自动分析装置的实施过程如下:

脚本分析模块 11,用于分析脚本文件,得到脚本文件中的宏开关信息;具体实施时,该脚本分析模块 11 具体包括:

解析子模块,用于解析相关脚本,得到预定义宏信息;该预定义宏信息包括该预定义宏名,以及各个预定义宏之间的依赖关系等信息;

分析子模块,用于根据该解析子模块解析得到的预定义宏信息,分析得到该宏的开关信息;

源代码分析模块 12,用于分析源代码,得到源代码中的宏定义信息;该宏定义信息包括该宏是否定义在源代码中,以及该宏是否存在上级条件宏等;

信息收集模块 13,用于收集脚本分析模块 11 分析得到的宏的开关信息,以及源代码分析模块 12 分析得到的宏定义信息;

状态分析模块 14,用于根据信息收集模块 13 收集到的宏的开关信息和宏定义信息,分析得到各个宏的实际状态;具体实施时,该状态分析模块 14 具体包括:

选择子模块,用于从各个宏当中选择当前状态显示为未定义的宏;

分析子模块,用于根据脚本分析模块 11 的分析子模块分析得到的宏的开关信息,以及

源代码分析模块 12 分析得到的宏定义信息,对选择子模块所选择的宏的实际状态进行分析,并得到宏的实际状态;

存储模块 15,用于存储状态分析模块 14 分析得到的宏的实际状态。

[0026] 当处理完成之后,编译代码之前,还需要对所有的代码进行更新,因此,如图 4 所示,本实施例的宏状态自动分析装置还可包括:

更新模块 16,用于根据存储模块 15 存储的宏的实际状态,更新源代码中宏的状态。

[0027] 在另一具体实施例中,宏状态自动分析装置包括上述实施例中的脚本分析模块 11 至存储模块 15,不同的是本实施例的装置还包括:

配置模块,用于配置编译环境;

查找模块,用于在配置编译环境过程中,查找宏定义,并记录查找到的宏定义,同时表明其控制范围。

[0028] 在又一具体实施例中,宏状态自动分析装置包括上述实施例中的脚本分析模块 11 至配置模块、查找模块,不同的是本实施例的装置还包括:

项目参数获取模块,用于获取待分析项目自定义的参数;

输入模块,用于用户输入参数;

参数解析模块,用于结合所获取的待分析项目自定义参数对用户所输入的参数进行解析,并将其带入 makefile 文件。

[0029] 本实施例的宏状态自动分析装置,通过脚本分析模块来分析脚本文件,以及源代码分析模块来分析源代码,并采用信息收集模块来收集脚本中宏的开关信息和源代码中的宏定义信息,再根据收集的宏的开关信息和宏定义信息分析各个宏的实际状态,即由 cmd 到 make 到源代码,这样一级一级解析下去,从而得到哪些宏是定义的,哪些未定义,实现了宏状态的自动分析,不需要编译验证,也不需要搜索后再进行筛选,进而提高了效率,也提高了可靠性。

[0030] 本发明实施例是以 SourceInsight 工具为例来进行详细的说明,当然本发明实施例也可应用与其他编译工具。

[0031] 以上所揭露的仅为本发明较佳实施例而已,当然不能以此来限定本发明之权利范围,本领域普通技术人员可以理解实现上述实施例的全部或部分流程,并依本发明权利要求所作的等同变化,仍属于发明所涵盖的范围。

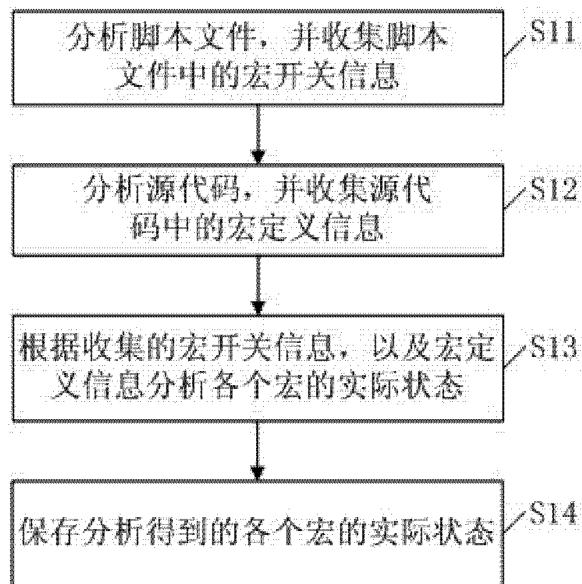


图 1

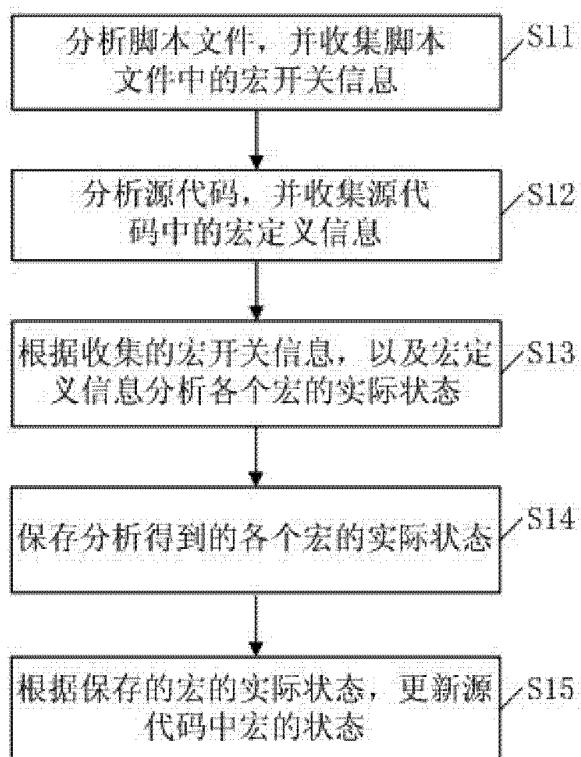


图 2

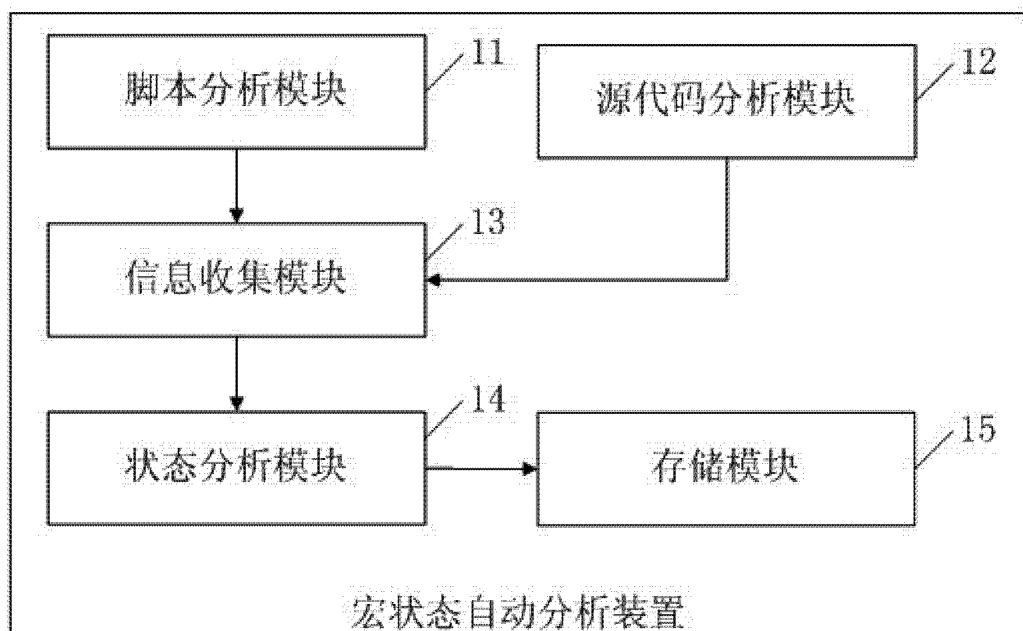


图 3

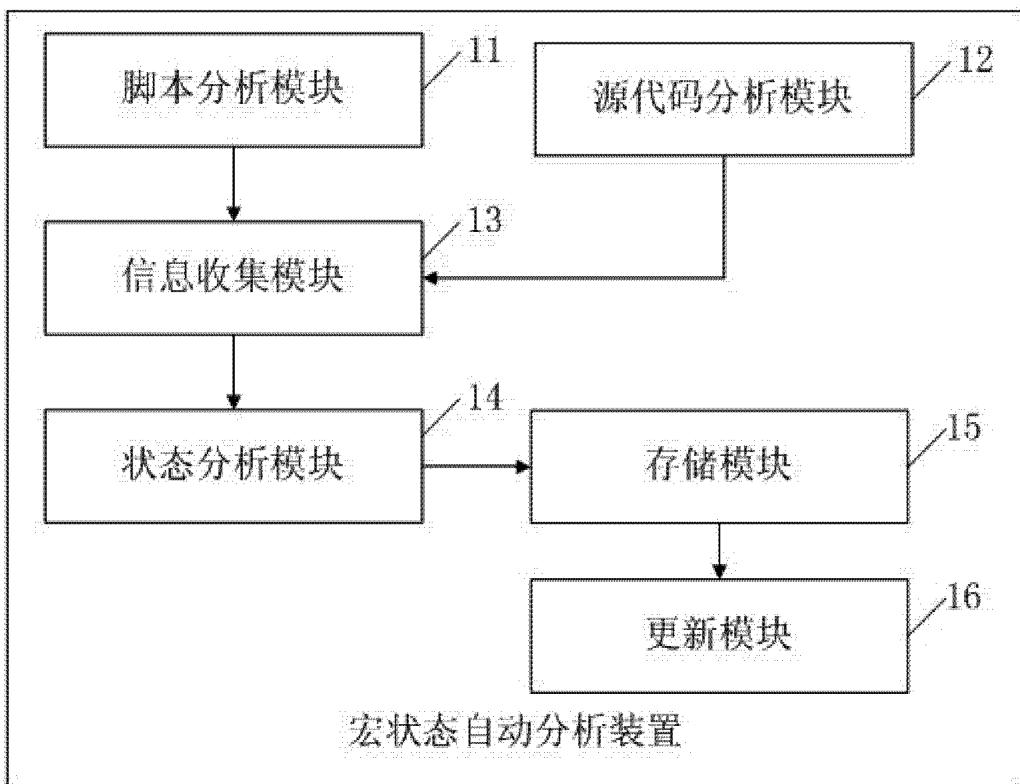


图 4