(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 111240972 B
(45)授权公告日 2022.03.08

(21)申请号 202010011072.1

(22)申请日 2020.01.06

(65)同一申请的已公布的文献号
申请公布号 CN 111240972 A

(43)申请公布日 2020.06.05

(73)专利权人 上海丰蕾信息科技有限公司
地址 200062 上海市普陀区云岭西路600弄6号801室

(72)发明人 史建琦 焦明月 黄滟鸿 孙文圣 战云龙 郭欣

(74)专利代理机构 北京辰权知识产权代理有限公司 11619
代理人 付婧

(51)Int.Cl.
G06F 11/36(2006.01)
G06F 30/20(2020.01)

(56)对比文件
CN 108376221 A,2018.08.07
CN 102520925 A,2012.06.27
CN 106874200 A,2017.06.20
US 2014019943 A1,2014.01.16
CN 108376221 A,2018.08.07
杨丽君.SEDS在星载综合电子系统中的应用设计.《计算机测量与控制》.2018,第26卷(第11期),
F. Zhang等.Formal Verification of Behavioral AADL Models by Stateful Timed CSP.《IEEE Access》.2017,第5卷
陶勇等.AADL模型的代码自动生成及集成技术.《计算机工程》.2009,第35卷(第8期),

审查员 陈玉艳

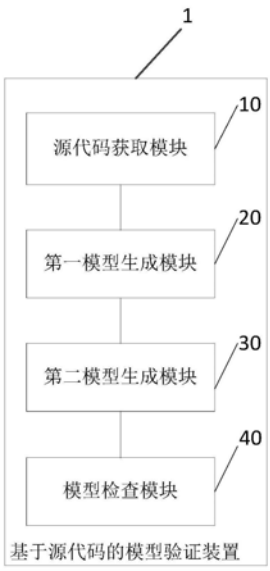
权利要求书2页 说明书13页 附图11页

(54)发明名称

一种基于源代码的模型验证装置

(57)摘要

本发明公开了一种基于源代码的模型验证装置,所述装置包括:源代码获取模块,用于获取目标源代码;第一模型生成模块,用于根据AADL建模技术将所述目标源代码进行建模后生成AADL模型;第二模型生成模块,用于基于预设转换方式将所述AADL模型转换生成时间自动机模型;模型检查模块,用于利用预设模型检测器对所述时间自动机模型进行检查。因此,采用本申请实施例,可以提高验证效率。



CN 111240972 B

1.一种基于源代码的模型检查装置,其特征在于,所述装置包括:

源代码获取模块,用于获取目标源代码;

第一模型生成模块,用于根据AADL建模技术将所述目标源代码进行建模后生成AADL模型;其中,所述第一模型生成模块,包括:

源代码转换单元,用于将所述目标源代码转换成AADL所需的构件集合,所述构件集合包括软件构件、执行平台构件及系统构件;

包获取单元,用于获取所述目标源代码对应的包;

系统生成单元,用于基于所述包将所述构件集合组织后生成实时系统;

第二模型生成单元,用于将所述实时系统确定为AADL语言子集,生成AADL模型;其中,

所述系统生成单元,包括:

属性特征获取子单元,用于获取所述包对应的属性和特征;

软件和硬件建模子单元,用于利用所述构件集合对系统中的软件和硬件进行建模;

性质建模子单元,用于利用所述属性和特征对系统中的功能性质和非功能性质进行建模;

第一模型生成子单元,用于当建模结束后生成AADL模型;

第二模型生成模块,用于基于预设转换方式将所述AADL模型转换生成时间自动机模型;

模型检查模块,用于利用预设模型检测器对所述时间自动机模型进行检查;

其中,所述装置还包括:

源代码生成模块,用于采用电子数据表单对系统、设备和软件接口进行处理后生成目标源代码;

其中,所述源代码生成模块,包括:

表单生成单元,用于采用电子数据表单对系统、设备和软件接口进行处理后生成处理后的电子数据表单;

第一模型生成单元,用于基于形式化模型转换技术将所述处理后的电子数据表单进行转换生成电子数据表单模型;

数据获取单元,用于基于形式化的性质描述语言获取所述电子数据表单模型中的关键结构和属性;

源代码获取单元,用于根据关键结构和属性获取关键部分源代码,将所述关键部分源代码作为目标源代码;其中,

所述第一模型生成模块,还用于:

基于形式化模型转换技术将SEDS中关于系统、设备、软件接口部分进行转换后形成SEDS模型;

根据SEDS中接口描述、协议和程序的说明及文档将其对应的功能逻辑,基于形式化的性质描述语言进行功能逻辑的刻画。

2.根据权利要求1所述的装置,其特征在于,所述第二模型生成模块,包括:

第三模型生成单元,包括基于预设时间抽象状态机将所述AADL模型映射生成时间抽象状态机模型;

模型转换单元,用于根据转换算法将所述时间抽象状态机模型转换成时间自动机模

型。

3.根据权利要求2所述的装置,其特征在于,所述第三模型生成单元,包括:

语法获取子单元,用于获取预设抽象状态机的抽象语法;

映射关系获取子单元,用于基于所述抽象语法获取所述AADL模型和所述预设抽象状态机的映射关系;

第二模型生成子单元,用于根据所述映射关系将所述AADL模型映射生成时间抽象状态机模型。

4.根据权利要求1所述的装置,其特征在于,所述模型检查模块,包括:

第一结果生成单元,用于根据预设模型检测器对所述目标源代码对应的逻辑正确性进行验证生成验证结果;

死锁去除单元,用于当所述验证结果中出现死锁时,通过获取死锁产生路径来去除死锁。

5.根据权利要求1所述的装置,其特征在于,所述模型检查模块,包括:

第二结果生成单元,用于根据模型检测器对所述目标源代码对应的变量进行检查生成检查结果;

信息输出单元,用于当所述检查结果和预设结果不一致时,输出错误信息。

一种基于源代码的模型验证装置

技术领域

[0001] 本发明涉及计算机技术领域，特别涉及一种基于源代码的模型验证装置。

背景技术

[0002] 随着信息技术的迅猛发展，软件系统的功能日益繁多、复杂，规模越来越大，其面临的安全性挑战也越来越严峻。特别是对于航空航天、汽车电子、工业控制、军事系统这一类安全攸关系统，软件的可信验证显得尤为重要。形式化验证技术是在形式化规约的基础上，通过深入分析与理解系统，验证系统是否满足其给定的性质，或发现系统不能满足该性质的条件。

[0003] 目前主要的形式化验证方法是定理证明。定理证明主要利用逻辑和数学手段，通过演绎推理来验证软件的模型或代码是否满足设定的性质。由于需要富有经验的用户提供大量的公理、前提条件及其他的系统相关信息才能满足验证，从而降低了验证效率。

发明内容

[0004] 本申请实施例提供了一种基于源代码的模型验证装置。为了对披露的实施例的一些方面有一个基本的理解，下面给出了简单的概括。该概括部分不是泛泛评述，也不是要确定关键/重要组成元素或描绘这些实施例的保护范围。其唯一目的是用简单的形式呈现一些概念，以此作为后面的详细说明的序言。

[0005] 本申请实施例提供了一种基于源代码的模型验证装置，所述装置包括：

[0006] 源代码获取模块，用于获取目标源代码；

[0007] 第一模型生成模块，用于根据AADL建模技术将所述目标源代码进行建模后生成AADL模型；

[0008] 第二模型生成模块，用于基于预设转换方式将所述AADL模型转换生成时间自动机模型；

[0009] 模型验证模块，用于利用预设模型检测器对所述时间自动机模型进行检查。

[0010] 可选的，所述装置还包括：

[0011] 源代码生成模块，用于采用电子数据表单对系统、设备和软件接口进行处理后生成目标源代码。

[0012] 可选的，所述源代码生成模块，包括：

[0013] 表单生成单元，用于采用电子数据表单对系统、设备和软件接口进行处理后生成处理后的电子数据表单；

[0014] 第一模型生成单元，用于基于形式化模型转换技术将所述处理后的电子数据表单进行转换生成电子数据表单模型；

[0015] 数据获取单元，用于基于形式化的性质描述语言获取所述电子数据表单模型中的关键结构和属性；

[0016] 源代码获取单元，用于根据关键结构和属性获取关键部分源代码，将所述关键部

分源代码作为目标源代码。

[0017] 可选的，所述第一模型生成模块，包括：

[0018] 源代码转换单元，用于将所述目标源代码转换成AADL所需的构件集合，所述构件集合包括软件构件、执行平台构件及系统构件；

[0019] 包获取单元，用于获取所述目标源代码对应的包；

[0020] 系统生成单元，用于基于所述包将所述构件集合组织后生成实时系统；

[0021] 第二模型生成单元，用于将所述实时系统确定为AADL语言子集，生成AADL模型。

[0022] 可选的，所述系统生成单元，包括：

[0023] 属性特征获取子单元，用于获取所述包对应的属性和特征；

[0024] 软件和硬件建模子单元，用于利用所述构件集合对系统中的软件和硬件进行建模；

[0025] 性质建模子单元，用于利用所述属性和特征对系统中的功能性质和非功能性质进行建模；

[0026] 第一模型生成子单元，用于当建模结束后生成AADL模型。

[0027] 可选的，所述第二模型生成模块，包括：

[0028] 第三模型生成单元，包括基于预设时间抽象状态机将所述AADL模型映射生成时间抽象状态机模型；

[0029] 模型转换单元，用于根据转换算法将所述时间抽象状态机模型转换成时间自动机模型。

[0030] 可选的，所述模型生成单元，包括：

[0031] 语法获取子单元，用于获取预设抽象状态机的抽象语法；

[0032] 映射关系获取子单元，用于基于所述抽象语法获取所述AADL模型和所述预设抽象状态机的映射关系；

[0033] 第二模型生成子单元，用于根据所述映射关系将所述AADL模型映射生成时间抽象状态机模型。

[0034] 可选的，所述模型验证模块，包括：

[0035] 第一结果生成单元，用于根据预设模型检测器对所述目标源代码对应的逻辑正确性进行验证生成验证结果；

[0036] 死锁去除单元，用于当所述验证结果中出现死锁时，通过获取所述死锁产生路径来去除死锁。

[0037] 可选的，所述利用预设模型检测器对所述时间自动机模型进行检查，包括：

[0038] 第二结果生成单元，用于根据模型检测器对所述目标源代码对应的变量进行检查生成检查结果；

[0039] 信息输出单元，用于当所述检查结果和预设结果不一致时，输出错误信息。

[0040] 本申请实施例提供的技术方案可以包括以下有益效果：

[0041] 在本申请实施例中，首先获取目标源代码，再根据AADL建模技术将所述目标源代码进行建模后生成AADL模型，然后基于预设转换方式将所述AADL模型转换生成时间自动机模型，最后利用预设模型检测器对所述时间自动机模型进行检查。在本申请实施例中，由于模型检验的可操作性好，自动化程度高，可以通过工具提供的详细反例来定位和解决系统

中存在的安全隐患，并且在在检验过程中，可以只对部分状态空间进行搜索，无需建立整个系统的状态空间，从而可以提高验证效率。

[0042] 应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本发明。

附图说明

[0043] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本发明的实施例，并与说明书一起用于解释本发明的原理。

[0044] 图1是本申请实施例提供的一种基于源代码的模型验证方法的流程示意图；

[0045] 图2是本申请实施例提供的一种以星载软件为例进行基于源代码的模型验证过程的过程示意图；

[0046] 图3是本申请实施例提供的一种将源代码进行转换后的构件集合示意图；

[0047] 图4是本申请实施例提供的一种基于源代码的模型验证装置的结构示意图；

[0048] 图5是本申请实施例提供的另一种基于源代码的模型验证装置的结构示意图；

[0049] 图6是本申请实施例提供的一种源代码生成模块的结构示意图；

[0050] 图7是本申请实施例提供的一种第一模型生成模块的结构示意图；

[0051] 图8是本申请实施例提供的一种系统生成单元的结构示意图；

[0052] 图9是本申请实施例提供的一种第二模型生成模块的结构示意图；

[0053] 图10是本申请实施例提供的一种第三模型生成模块的结构示意图；

[0054] 图11是本申请实施例提供的一种模型检查模块的结构示意图；

[0055] 图12是本申请实施例提供的一种终端的结构示意图。

具体实施方式

[0056] 以下描述和附图充分地示出本发明的具体实施方案，以使本领域的技术人员能够实践它们。

[0057] 应当明确，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它实施例，都属于本发明保护的范围。

[0058] 下面的描述涉及附图时，除非另有表示，不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本发明相一致的所有实施方式。相反，它们仅是如所附权利要求书中所详述的、本发明的一些方面相一致的装置和方法的例子。

[0059] 在本发明的描述中，需要理解的是，术语“第一”、“第二”等仅用于描述目的，而不能理解为指示或暗示相对重要性。对于本领域的普通技术人员而言，可以具体情况理解上述术语在本发明中的具体含义。此外，在本发明的描述中，除非另有说明，“多个”是指两个或两个以上。“和/或”，描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。字符“/”一般表示前后关联对象是一种“或”的关系。

[0060] 到目前为止，对于系统形式化验证，目前主要的形式化验证方法是定理证明。定理

证明主要利用逻辑和数学手段，通过演绎推理来验证软件的模型或代码是否满足设定的性质。由于需要富有经验的用户提供大量的公理、前提条件及其他的系统相关信息才能满足验证，从而降低了验证效率。为此，本申请提供了一种基于源代码的模型验证方法、装置、存储介质及终端，以解决上述相关技术问题中存在的问题。本申请提供的技术方案中，由于模型检验的可操作性好，自动化程度高，可以通过工具提供的详细反例来定位和解决系统中存在的安全隐患，并且在在检验过程中，可以只对部分状态空间进行搜索，无需建立整个系统的状态空间，从而可以提高验证效率，下面采用示例性的实施例进行详细说明。

[0061] 下面将结合附图1-附图3，对本申请实施例提供的基于源代码的模型验证方法进行详细介绍。该方法可依赖于计算机程序实现，可运行于基于冯诺依曼体系的基于源代码的模型验证装置上。该计算机程序可集成在应用中，也可作为独立的工具类应用运行。其中，本申请实施例中的基于源代码的模型验证装置可以为用户终端，包括但不限于：个人电脑、平板电脑、手持设备、车载设备、可穿戴设备、计算设备或连接到无线调制解调器的其它处理设备等。在不同的网络中用户终端可以叫做不同的名称，例如：用户设备、接入终端、用户单元、用户站、移动站、移动台、远方站、远程终端、移动设备、用户终端、终端、无线通信设备、用户代理或用户装置、蜂窝电话、无绳电话、个人数字处理(personal digital assistant，PDA)、5G网络或未来演进网络中的终端设备等。

[0062] 请参见图1，为本申请实施例提供了一种基于源代码的模型验证方法的流程示意图。如图1所示，本申请实施例的所述方法可以包括以下步骤：

[0063] S101，获取目标源代码；

[0064] 其中，目标源代码是需要进行验证的系统核心代码片段。在本申请实施例中目标源代码是来自星载系统的源代码。

[0065] 通常，本申请实施例提供的基于源代码的模型验证方法以提供的星载软件为例进行说明，实际的软件以具体应用场景进行选择，此处不做限定。

[0066] 在本申请实施例中，首先采用电子数据表单对系统、设备和软件接口进行处理后生成处理后的电子数据表单，再基于形式化模型转换技术将所述处理后的电子数据表单进行转换生成电子数据表单模型，然后基于形式化的性质描述语言获取所述电子数据表单模型中的关键结构和属性，最后根据关键结构和属性获取关键部分源代码，将所述关键部分源代码作为目标源代码。

[0067] S102，根据AADL建模技术将所述目标源代码进行建模后生成AADL模型；

[0068] 其中，AADL(Architecture Analysis and Design language)是一种应用于嵌入式系统领域的体系结构建模语言，支持航空、航天、汽车等领域复杂实时的安全关键系统的设计与分析。

[0069] 例如，根据星载系统的特性，选取一段基于SEDS的形式化描述，根据其对应源代码，然后将其转换为一个较为完整的AADL语言子集，并形式地描述该语言子集的抽象语法。

[0070] 在本申请的一个实施例中，步骤S102包括：

[0071] 步骤S102-1：使用SEDS对系统、设备、软件接口进行形式化说明；

[0072] 具体的，基于形式化模型转换技术将SEDS中关于系统、设备、软件接口部分形成SEDS模型。

[0073] 根据SEDS中接口描述、协议和程序的说明及文档将其相关的功能逻辑，基于形式

化的性质描述语言如LTL进行功能逻辑的刻画。

[0074] 步骤S102-2:针对关键部分源代码,使用AADL建模技术对其进行建模;

[0075] 具体的,通过对关键结构与属性的抽取,无关属性的移除,选择关键部分源代码。

[0076] AADL作为一种基于模型驱动的设计语言,不仅能够对系统软件进行建模,也能对软件相关的硬件进行抽象描述,不仅能通过构建描述对系统的静态结构建模,也能通过构件交互、行为附件等描述系统的动态特征。

[0077] AADL通过构件、连接等概念描述系统的软、硬件体系结构;通过特征、属性描述系统功能与非功能性质;通过模式变换描述运行时体系结构演化;通过用户定义属性和附件支持可扩展。

[0078] 在本实施例中,将选择的源代码转换成AADL中软件构件、执行平台构件及系统构件如图3所示。

[0079] 其中,构件和连接将对系统中的软件与硬件进行建模;属性与特征实现了对系统中的功能性质和非功能性质进行建模;模式转换对系统运行时的体系结构的变化进行了描述。若是想要系统支持可扩展性,则只需要通过用户定义属性和扩展附件完成。

[0080] 通过Package,利用组织的方式,构建复杂的实时系统,得到一个较为完整的AADL语言子集,并形式地描述该语言子集的抽象语义。

[0081] 在本实施例中,AADL的语法规则可以如下形式表述:

[0082] Mode::=System+System.Impl

[0083] System::=system Identifier Feature?Annex?

[0084] end;

[0085] Features::=features Feature+

[0086] Feature::=Identifier:{in|out}{event|data|eventdata}port;

[0087] SystemImpl::=system implementation Identifier.System Identifier

[0088] Identifier Subcomponents?Connections?

[0089] end;

[0090] Subcomponents::=subcomponents Subcomponent+

[0091] Subcomponents::=Identifier:system Identifier;

[0092] Connections::=connections Connection

[0093] Connection::={event|data|enentdata}port Identifier.Identifier->Identifier.Identifier

[0094] 由语法规则可知,一个AADL模型由一个或者多个系统构件组成,系统构件可以定义类型和其对应的实现,同时可以定义特征属性(输入输出端口)和扩展附件;系统构件的实现内可以定义子程序以及构件间的连接,子程序可以看成一个子系统,和系统构件的定义和实现是一样的,连接则是线程间或者进程间的通信,分为数据端口、事件端口或者数据事件端口。通过以上可以完整的定义一个系统模型。

[0095] S103,基于预设转换方式将所述AADL模型转换生成时间自动机模型;

[0096] 具体的,先形式地给出转换中所使用的中级语言,即时间抽象状态机(TASM)的抽象语法;然后,基于语义映射函数描述AADL到TASM的整体映射关系,并基于类似ML的元语言形式定义转换规则,利用从TASM模型到TA模型的转换算法实现最终的模型转换功能。

[0097] 该关键技术是实现星载软件安全性分析验证的前提,可为AADL模型的分析验证提供重要的前提保障。

[0098] 在本申请的一个实施例中,步骤S103包括:

[0099] 步骤S103-1:将时间抽象状态机(TASM)作为中间语言,使得半形式化的AADL模型映射成形式化的TASM模型;

[0100] 在本实施例中,形式地给出转换中所使用的中级语言,即时间抽象状态机(TASM)的抽象语法,相较于BIP,Fiacre,TLA+,Signal等形式语言,TASM能够同时支持功能、时间以及资源等行为的表达,且可读性较好。TASM的抽象语法描述如下:

[0101] P::=x:=exp

[0102] |skip

[0103] |if Bexp then P

[0104] |else then P

[0105] $|\mathrm{time}(t_{min}, t_{max}) \triangleright P$

[0106] $|\mathrm{time\ next} \triangleright P$

[0107] $|\mathrm{resource\ r}(r_{min}, r_{max}) \triangleright P$

[0108] |p⊕p

[0109] |p⊗p

[0110] TASM::=<E,P‖P‖…‖P>

[0111] 对于该抽象语法,假设P为主机的行为,x:=exp表示更新受控变量x的值,time说明了P执行的时间,P在执行过程中所消耗的资源用resource表示,并且资源的名称用小写的r来表示;p⊕p是在一个主机范围内关联多个规则的选择操作符,p⊗p是一个同步并行操作符,P‖P是一种关联着主机的并行操作符,它们共享相同的环境E。

[0112] AADL中的每个线程转换成TASM中的两种状态机,其一用于执行,其二用于分派。端口通信(port communication)、模式转换自动机(mode change automata)和带有调度协议定义的线程构件也分别转换成端口通信机(machines of port communication)、系统操作模式状态机(System Operation Mode,SOM)和调度状态机(Scheduler)。这些生成的状态机将通过使用环境变量(environment variables)进行彼此同步通信。

[0113] 更进一步的,基于语义映射函数描述AADL到TASM的整体映射关系;并基于类似ML的元语言形式定义转换规则。

[0114] 步骤S103-2:利用从TASM模型到TA模型的转换算法,将形式化的TASM模型转换到形式化建模语言时间自动机(TA:Time Automata)模型;

[0115] S104,利用预设模型检测器对所述时间自动机模型进行检查。

[0116] 具体的,对于上一过程中得到的星载软件形式化模型,需要构造模型检测器,对相关性质进行形式化验证。对于得到的形式化模型,使用计算树逻辑描述其规范,星载软件模型检测器验证规范在指定模型中是否成立,如果不成立给出反例。

[0117] 星载软件模型检测器对传统的时间自动机的状态和通道都做了改进,并进行了一些扩展。每一个过程被描述为由有限控制结构、实数值时钟和变量组成的时间自动机,过程之间通过信道和共享变量通信,信道保证不同自动机中的两个或多个转换同步进行。星载

软件模型检测器的验证机制能够避免状态空间爆炸的问题,因此可通过仿真验证有效发现所建模型的错误之处,进而发现系统设计的不足。

[0118] 星载软件模型检测工具的主要有三个部分,分别是系统编辑器(System Editor),仿真器(Simulator)和验证器(Verifier)。针对星载软件,该模型检测器主要包括以下特点:

[0119] 1.采用符号模型检测技术。该技术的基本原理是将系统的状态转换关系用逻辑公式表示。在这方法中,二叉图是用以表示逻辑公式的重要手段,它能较为紧凑地表示状态转换关系,以降低系统模型所需的内存空间。另外,状态转换的计算可以以集合为单位,以提高搜索的效率。本工具采用有序的二进制判决图(OBDD),为提供了处理大系统进行符号模型检验的可能性。

[0120] 2.采用偏序规约技术。星载软件进程之间往往是异步的,状态数就会指数级增加。两个事件若是独立的,如果不论它们按什么顺序执行,其结果是相同的整体状态。可以将某些状态的次序固定,以减少重复验证,使用偏序规约方法可以有效地解决星载软件中异步进程的状态爆炸问题,

[0121] 3.采用On-the-fly技术。使用On-the-fly技术,在模型检测进行过程中,它可以根据需要而展开系统路径所包含的状态,避免预先生成系统中所包含的所有状态。解决传统模型检测方法中的空间爆炸问题。

[0122] 4.采用SAT技术,进行有界模型检查。随着布尔可满足性问题(SAT)研究的的进展,我们可以以其为基础,使用有界模型检验(BMC)对星载软件进行形式化验证。极大的提高模型检测的灵活性。

[0123] 随着相关领域最新技术的应用,星载软件模型检测器可以实现下列功能:

[0124] 1.星载软件系统仿真。对于建立的星载软件系统形式化模型,该检测工具可以对星载软件系统进行仿真。对软件系统的系统逻辑与正确性进行初步验证,减少后期相关工作量。

[0125] 2.变量分析。星载软件模型检测器允许在用户指定的路径检查变量,便于跟踪变量,检查数据中可能出现的冲突与缺陷。

[0126] 3.反例输出。该工具在对星载软件进行模型检测过程中,若发现与用户规范不符合的情况,会输出相关缺陷与冲突的执行路径,帮助用户准确的定位缺陷,及时发现与解决问题。

[0127] 4.并发竞争分析与定位。并发系统中,两个或两个以上的进程在执行过程中,由于竞争资源或者由于彼此通信而造成的一种阻塞的现象,若无外力作用,它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁,这些永远在互相等待的进程称为死锁进程。通过星载软件模型检测器对时间自动机模型进行模型检测,可以检测到系统中是否存在死锁问题,若存在死锁问题,可以通过分析死锁产生的路径消除死锁,从而完善软件,保证星载软件的安全性。

[0128] 5.可达性分析。可达性是系统各种性质中最为简单和基本的属性。可达性通常用于执行对所设计模型的健全性检查。若一个模型部分构件是不可达的,该模型肯定是不健全的模型。星载软件模型检测工具主要通过快速搜索机制来验证时钟约束和可达性。它的主要优点是其高效性和使用的方便性。

[0129] 6.安全性保证。星载软件模型检测工具通过形式化技术，保证软件安全性，确保不会发成引起系统灾难性后果的行为。软件的安全性可以表述为:“灾难性的后果将永远不会发生”。

[0130] 7.活性保证。和安全性相对应，活性是指软件中期望的状态最终能够到达，“好”事情终会发生。星载软件模型检测工具可以通过活性验证来检验模型是否符合系统预先设计的要求。

[0131] 在本申请实施例中，通过直接将AADL模型映射成一种形式化模型时间抽象状态机(TASM)，实现异构模型的相互转换，从而使得安全攸关实时系统能够利用现有的形式化验证工具进行分析和验证，提高系统的安全可靠性和开发效率；其次，通过符号模型检测技术，将系统的状态转换关系用形式化逻辑公式表示，并且只对部分状态空间进行搜索，无需建立整个系统的状态空间，可操作性好，搜索效率高；其次，通过偏序规约技术，有效地解决软件中异步进程的状态爆炸问题；其次，通过On-the-fly技术，解决传统模型检测方法中的空间爆炸问题；再次，通过SAT技术，进行有界模型检查，极大的提高模型检测的灵活性。

[0132] 下述为本发明装置实施例，可以用于执行本发明方法实施例。对于本发明装置实施例中未披露的细节，请参照本发明方法实施例。

[0133] 请参见图4，其示出了本发明一个示例性实施例提供的基于源代码的模型验证装置的结构示意图。该基于源代码的模型验证装置可以通过软件、硬件或者两者的结合实现成为终端的全部或一部分。该装置1包括源代码获取模块10、第一模型生成模块20、第二模型生成模块30、模型检查模块40。

[0134] 源代码获取模块10，用于获取目标源代码；

[0135] 第一模型生成模块20，用于根据AADL建模技术将所述目标源代码进行建模后生成AADL模型；

[0136] 第二模型生成模块30，用于基于预设转换方式将所述AADL模型转换生成时间自动机模型；

[0137] 模型检查模块40，用于利用预设模型检测器对所述时间自动机模型进行检查。

[0138] 可选的，如图5所示，所述装置1还包括：

[0139] 源代码生成模块50，用于采用电子数据表单对系统、设备和软件接口进行处理后生成目标源代码。

[0140] 可选的，如图6所示，所述源代码生成模块50，包括：

[0141] 表单生成单元510，用于采用电子数据表单对系统、设备和软件接口进行处理后生成处理后的电子数据表单；

[0142] 第一模型生成单元520，用于基于形式化模型转换技术将所述处理后的电子数据表单进行转换生成电子数据表单模型；

[0143] 数据获取单元530，用于基于形式化的性质描述语言获取所述电子数据表单模型中的关键结构和属性；

[0144] 源代码获取单元540，用于根据关键结构和属性获取关键部分源代码，将所述关键部分源代码作为目标源代码。

[0145] 可选的，如图7所示，第一模型生成模块20，包括：

[0146] 源代码转换单元210，用于将所述目标源代码转换成AADL所需的构件集合，所述构

件集合包括软件构件、执行平台构件及系统构件；

[0147] 包获取单元220，用于获取所述目标源代码对应的包；

[0148] 系统生成单元230，用于基于所述包将所述构件集合组织后生成实时系统；

[0149] 第二模型生成单元240，用于将所述实时系统确定为AADL语言子集，生成AADL模型。

[0150] 可选的，如图8所示，所述系统生成单元230，包括：

[0151] 属性特征获取子单元2301，用于获取所述包对应的属性和特征；

[0152] 软件和硬件建模子单元2302，用于利用所述构件集合对系统中的软件和硬件进行建模；

[0153] 性质建模子单元2303，用于利用所述属性和特征对系统中的功能性质和非功能性质进行建模；

[0154] 第一模型生成子单元2304，用于当建模结束后生成AADL模型。

[0155] 可选的，如图9所示，所述第二模型生成模块30，包括：

[0156] 第三模型生成单元310，包括基于预设时间抽象状态机将所述AADL模型映射生成时间抽象状态机模型；

[0157] 模型转换单元320，用于根据转换算法将所述时间抽象状态机模型转换成时间自动机模型。

[0158] 可选的，如图10所示，所述第三模型生成单元310，包括：

[0159] 语法获取子单元3101，用于获取预设抽象状态机的抽象语法；

[0160] 映射关系获取子单元3102，用于基于所述抽象语法获取所述AADL模型和所述预设抽象状态机的映射关系；

[0161] 第二模型生成子单元3103，用于根据所述映射关系将所述AADL模型映射生成时间抽象状态机模型。

[0162] 可选的，如图11所示，所述模型检查模块40，包括：

[0163] 第一结果生成单元410，用于根据预设模型检测器对所述目标源代码对应的逻辑正确性进行验证生成验证结果；

[0164] 死锁去除单元420，用于当所述验证结果中出现死锁时，通过获取所述死锁产生路径来去除死锁。

[0165] 第二结果生成单元430，用于根据模型检测器对所述目标源代码对应的变量进行检查生成检查结果；

[0166] 信息输出单元440，用于当所述检查结果和预设结果不一致时，输出错误信息。

[0167] 需要说明的是，上述实施例提供的基于源代码的模型验证装置在执行基于源代码的模型验证方法时，仅以上述各功能模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能模块完成，即将设备的内部结构划分成不同的功能模块，以完成以上描述的全部或者部分功能。另外，上述实施例提供的基于源代码的模型验证装置与基于源代码的模型验证方法实施例属于同一构思，其体现实现过程详见方法实施例，这里不再赘述。

[0168] 上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

[0169] 在本申请实施例中，首先获取目标源代码，再根据AADL建模技术将所述目标源代

码进行建模后生成AADL模型，然后基于预设转换方式将所述AADL模型转换生成时间自动机模型，最后利用预设模型检测器对所述时间自动机模型进行检查。在本申请实施例中，由于模型检验的可操作性好，自动化程度高，可以通过工具提供的详细反例来定位和解决系统中存在的安全隐患，并且在在检验过程中，可以只对部分状态空间进行搜索，无需建立整个系统的状态空间，从而可以提高验证效率。

[0170] 本发明还提供一种计算机可读介质，其上存储有程序指令，该程序指令被处理器执行时实现上述各个方法实施例提供的基于源代码的模型验证方法。

[0171] 本发明还提供了一种包含指令的计算机程序产品，当其在计算机上运行时，使得计算机执行上述各个方法实施例所述的基于源代码的模型验证方法。

[0172] 请参见图12，为本申请实施例提供了一种终端的结构示意图。如图12所示，所述终端1000可以包括：至少一个处理器1001，至少一个网络接口1004，用户接口1003，存储器1005，至少一个通信总线1002。

[0173] 其中，通信总线1002用于实现这些组件之间的连接通信。

[0174] 其中，用户接口1003可以包括显示屏(Display)、摄像头(Camera)，可选用户接口1003还可以包括标准的有线接口、无线接口。

[0175] 其中，网络接口1004可选的可以包括标准的有线接口、无线接口(如WI-FI接口)。

[0176] 其中，处理器1001可以包括一个或者多个处理核心。处理器1001利用各种借口和线路连接整个电子设备1000内的各个部分，通过运行或执行存储在存储器1005内的指令、程序、代码集或指令集，以及调用存储在存储器1005内的数据，执行电子设备1000的各种功能和处理数据。可选的，处理器1001可以采用数字信号处理(Digital Signal Processing，DSP)、现场可编程门阵列(Field-Programmable Gate Array，FPGA)、可编程逻辑阵列(Programmable Logic Array，PLA)中的至少一种硬件形式来实现。处理器1001可集成中央处理器(Central Processing Unit，CPU)、图像处理器(Graphics Processing Unit，GPU)和调制解调器等中的一种或几种的组合。其中，CPU主要处理操作系统、用户界面和应用程序等；GPU用于负责显示屏所需要显示的内容的渲染和绘制；调制解调器用于处理无线通信。可以理解的是，上述调制解调器也可以不集成到处理器1001中，单独通过一块芯片进行实现。

[0177] 其中，存储器1005可以包括随机存储器(Random Access Memory，RAM)，也可以包括只读存储器(Read-Only Memory)。可选的，该存储器1005包括非瞬时性计算机可读介质(non-transitory computer-readable storage medium)。存储器1005可用于存储指令、程序、代码、代码集或指令集。存储器1005可包括存储程序区和存储数据区，其中，存储程序区可存储用于实现操作系统的指令、用于至少一个功能的指令(比如触控功能、声音播放功能、图像播放功能等)、用于实现上述各个方法实施例的指令等；存储数据区可存储上面各个方法实施例中涉及到的数据等。存储器1005可选的还可以是至少一个位于远离前述处理器1001的存储装置。如图12所示，作为一种计算机存储介质的存储器1005中可以包括操作系统、网络通信模块、用户接口模块以及基于源代码的模型验证应用程序。

[0178] 在图12所示的终端1000中，用户接口1003主要用于为用户提供输入的接口，获取用户输入的数据；而处理器1001可以用于调用存储器1005中存储的基于源代码的模型验证应用程序，并具体执行以下操作：

[0179] 获取目标源代码；

[0180] 根据AADL建模技术将所述目标源代码进行建模后生成AADL模型；

[0181] 基于预设转换方式将所述AADL模型转换生成时间自动机模型；

[0182] 利用预设模型检测器对所述时间自动机模型进行检查。

[0183] 在一个实施例中，所述处理器1001在执行所述获取目标源代码之前时，还执行以下操作：

[0184] 采用电子数据表单对系统、设备和软件接口进行处理后生成目标源代码。

[0185] 在一个实施例中，所述处理器1001在执行所述采用电子数据表单对系统、设备和软件接口进行处理后生成目标源代码之前时，具体执行以下操作：

[0186] 采用电子数据表单对系统、设备和软件接口进行处理后生成处理后的电子数据表单；

[0187] 基于形式化模型转换技术将所述处理后的电子数据表单进行转换生成电子数据表单模型；

[0188] 基于形式化的性质描述语言获取所述电子数据表单模型中的关键结构和属性；

[0189] 根据关键结构和属性获取关键部分源代码，将所述关键部分源代码作为目标源代码。

[0190] 在一个实施例中，所述处理器1001在执行所述根据AADL建模技术将所述目标源代码进行建模后生成AADL模型时，具体执行以下操作：

[0191] 将所述目标源代码转换成AADL所需的构件集合，所述构件集合包括软件构件、执行平台构件及系统构件；

[0192] 获取所述目标源代码对应的包；

[0193] 基于所述包将所述构件集合组织后生成实时系统；

[0194] 将所述实时系统确定为AADL语言子集，生成AADL模型。

[0195] 在一个实施例中，所述处理器1001在执行所述基于所述包将所述构件集合组织后生成实时系统时，具体执行以下操作：

[0196] 获取所述包对应的属性和特征；

[0197] 利用所述构件集合对系统中的软件和硬件进行建模；

[0198] 利用所述属性和特征对系统中的功能性质和非功能性质进行建模；

[0199] 当建模结束后生成AADL模型。

[0200] 在一个实施例中，所述处理器1001在执行所述基于预设转换方式将所述AADL模型转换生成时间自动机模型时，具体执行以下操作：

[0201] 基于预设时间抽象状态机将所述AADL模型映射生成时间抽象状态机模型；

[0202] 根据转换算法将所述时间抽象状态机模型转换成时间自动机模型。

[0203] 在一个实施例中，所述处理器1001在执行所述基于预设时间抽象状态机将所述AADL模型映射生成时间抽象状态机模型时，具体执行以下操作：

[0204] 获取预设抽象状态机的抽象语法；

[0205] 基于所述抽象语法获取所述AADL模型和所述预设抽象状态机的映射关系；

[0206] 根据所述映射关系将所述AADL模型映射生成时间抽象状态机模型。

[0207] 在一个实施例中，所述处理器1001在执行所述利用预设模型检测器对所述时间自

动机模型进行检查时，具体执行以下操作：

[0208] 根据预设模型检测器对所述目标源代码对应的逻辑正确性进行验证生成验证结果；

[0209] 当所述验证结果中出现死锁时，通过获取所述死锁产生路径来去除死锁。

[0210] 在一个实施例中，所述处理器1001在执行所述利用预设模型检测器对所述时间自动机模型进行检查时，具体执行以下操作：

[0211] 根据模型检测器对所述目标源代码对应的变量进行检查生成检查结果；

[0212] 当所述检查结果和预设结果不一致时，输出错误信息。

[0213] 在本申请实施例中，首先获取目标源代码，再根据AADL建模技术将所述目标源代码进行建模后生成AADL模型，然后基于预设转换方式将所述AADL模型转换生成时间自动机模型，最后利用预设模型检测器对所述时间自动机模型进行检查。在本申请实施例中，由于模型检验的可操作性好，自动化程度高，可以通过工具提供的详细反例来定位和解决系统中存在的安全隐患，并且在在检验过程中，可以只对部分状态空间进行搜索，无需建立整个系统的状态空间，从而可以提高验证效率。

[0214] 本领域技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。所属技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

[0215] 本文所披露的实施例中，应该理解到，所揭露的方法、产品(包括但不限于装置、设备等)，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

[0216] 应当理解的是，附图中的流程图和框图显示了根据本发明的多个实施例的系统、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，所述模块、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，两个连续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。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动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。本发明并不局限于上面已经描述并在附图中示出的流程及结构,并且可以在不脱离其范围进行各种修改和改变。本发明的范围仅由所附的权利要求来限制。

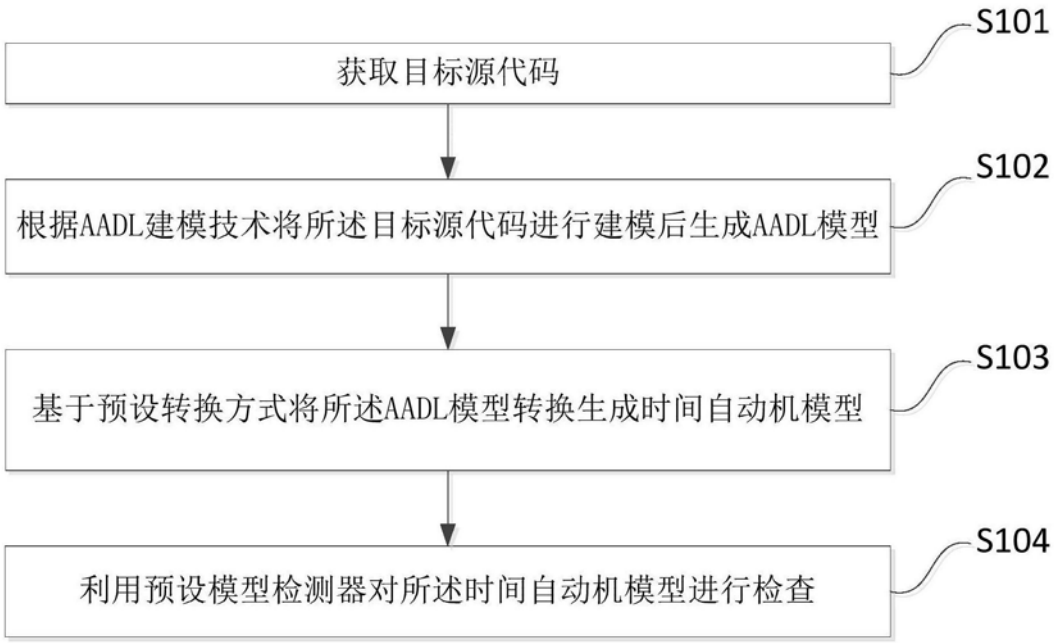

S101
获取目标源代码
S102
根据AADL建模技术将所述目标源代码进行建模后生成AADL模型
S103
基于预设转换方式将所述AADL模型转换生成时间自动机模型
S104
利用预设模型检测器对所述时间自动机模型进行检查

图1

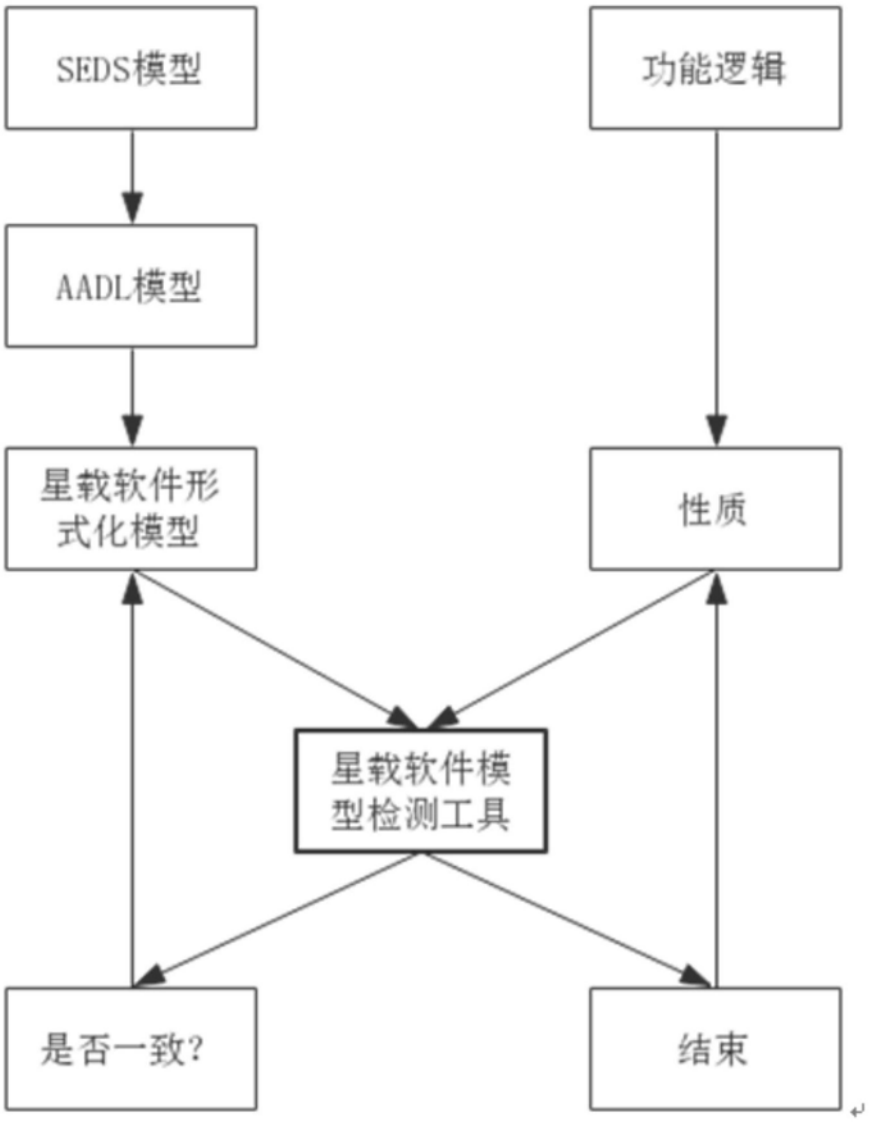


图2

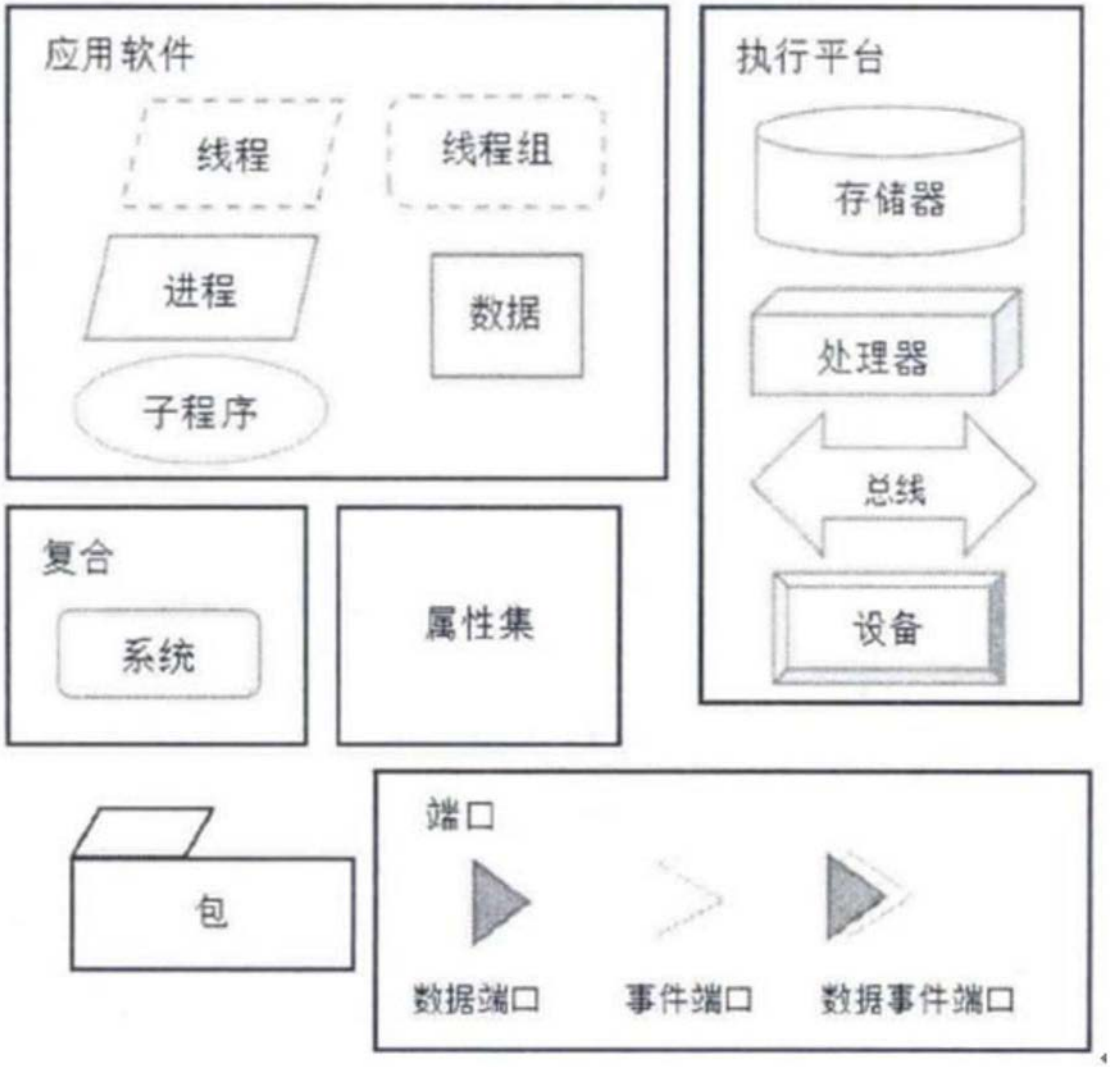

应用软件
线程
线程组
进程
数据
子程序
执行平台
存储器
处理器
总线
设备
复合
系统
属性集
包
端口
数据端口
事件端口
数据事件端口

图3

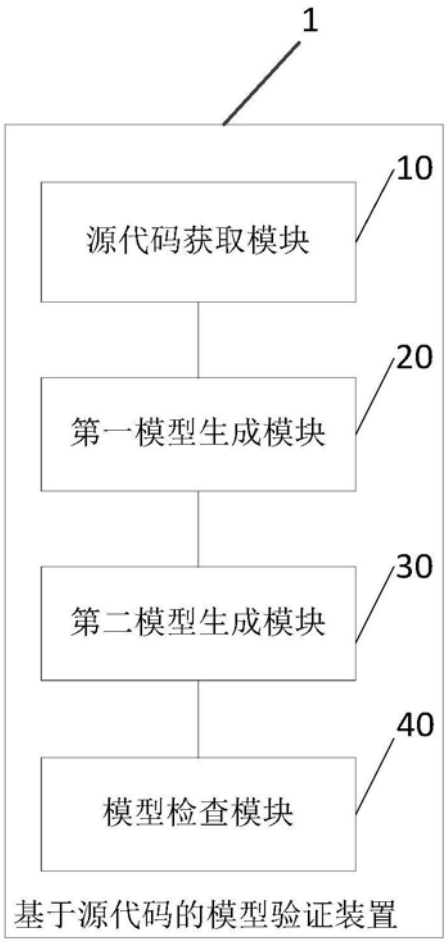

1
10
源代码获取模块
20
第一模型生成模块
30
第二模型生成模块
40
模型检查模块
基于源代码的模型验证装置

图4

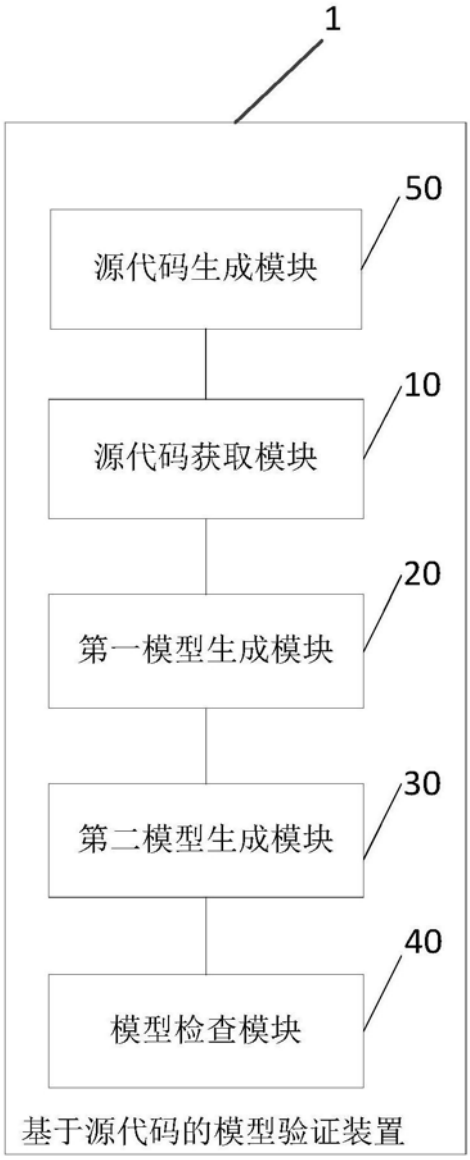

1
50
源代码生成模块
10
源代码获取模块
20
第一模型生成模块
30
第二模型生成模块
40
模型检查模块
基于源代码的模型验证装置

图5

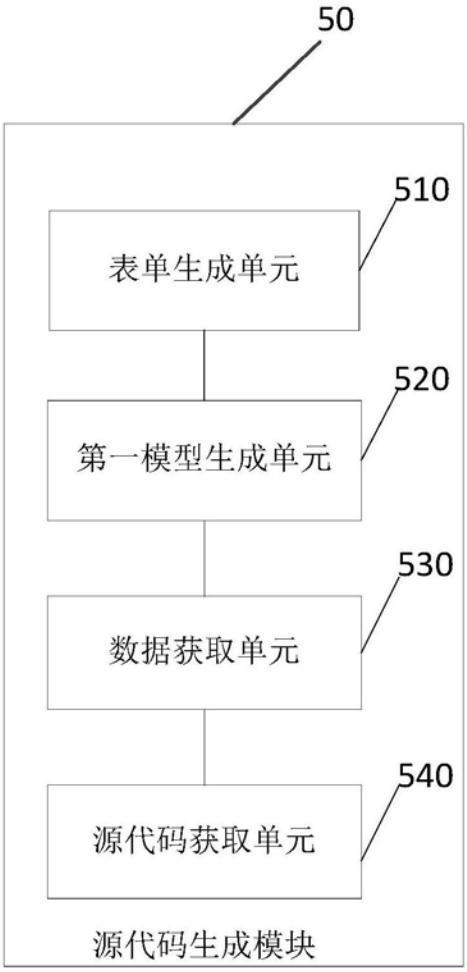


图6

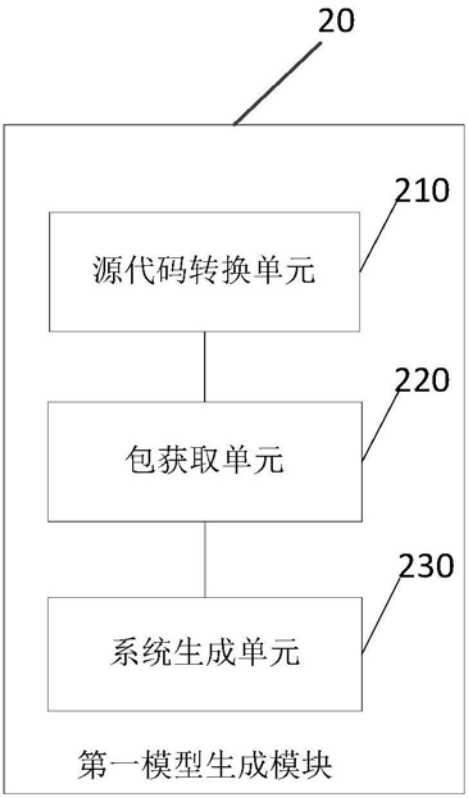


图7

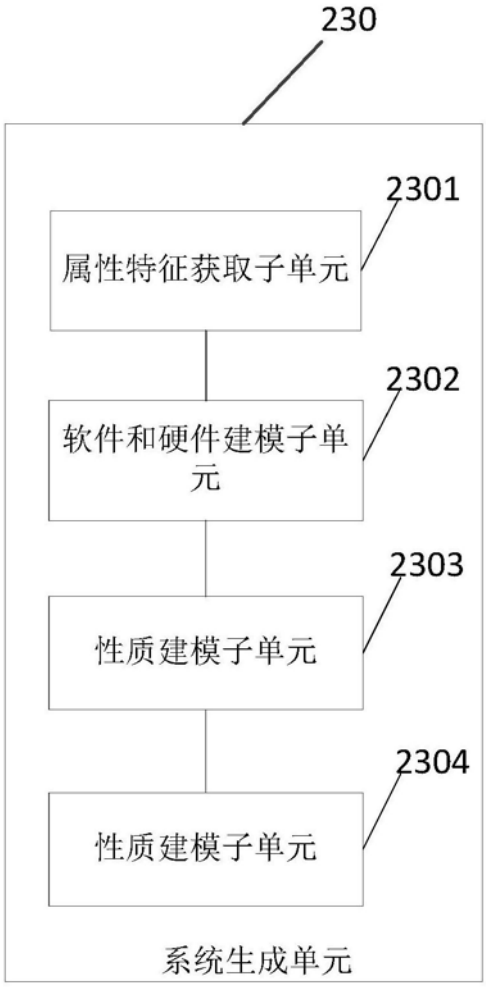

230
2301
属性特征获取子单元
2302
软件和硬件建模子单元
2303
性质建模子单元
2304
性质建模子单元
系统生成单元

图8

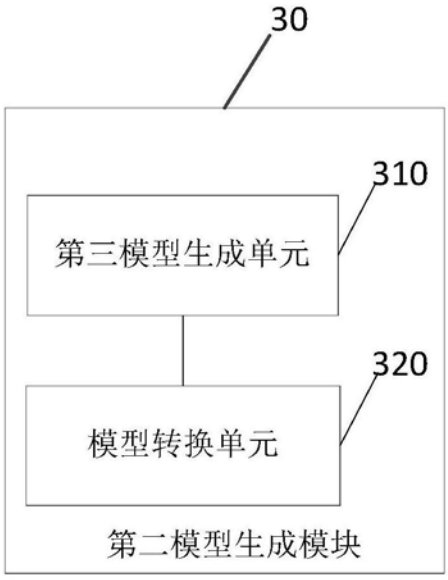

30
310
第三模型生成单元
320
模型转换单元
第二模型生成模块

图9

图10

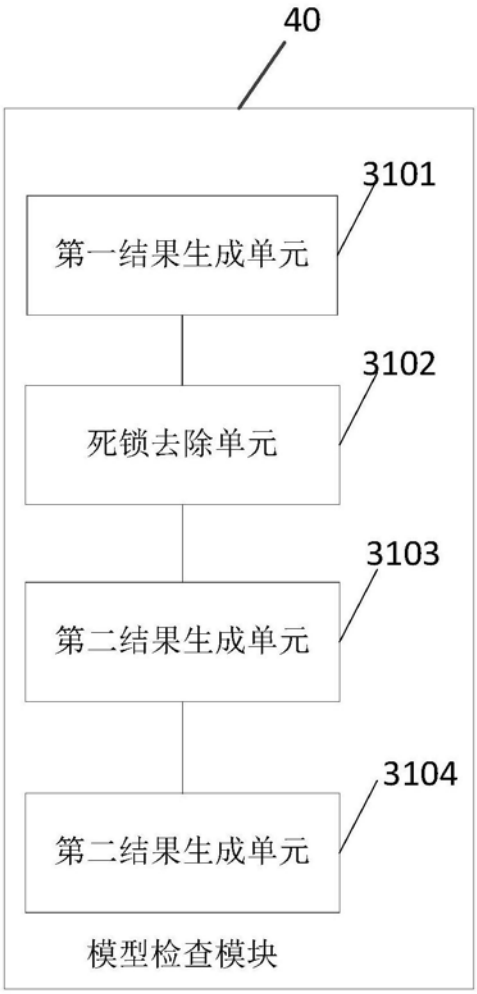

40
3101
第一结果生成单元
3102
死锁去除单元
3103
第二结果生成单元
3104
第二结果生成单元
模型检查模块

图11

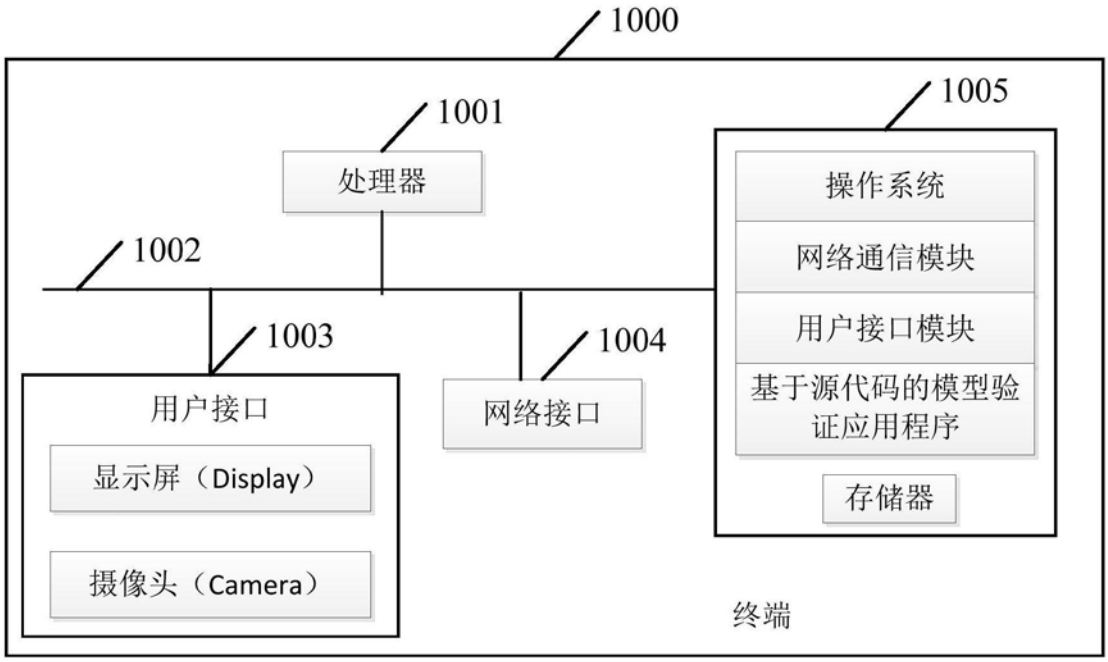

1000
1001
处理器
1002
1003
用户接口
显示屏（Display）
摄像头（Camera）
1004
网络接口
1005
操作系统
网络通信模块
用户接口模块
基于源代码的模型验证应用程序
存储器
终端

图12