基于 NuSMV 的 SysML 模型形式化验证

邓刘梦,葛晓瑜,宛伟健
(南京航空航天大学 计算机科学与技术学院,江苏 南京 211106)

摘 要:航空航天道路交通等高安全领域的系统开发需要保证高安全、高可靠,对于该类系统的合理建模以及模型验证则尤为重要。当前模型驱动开发方法已经广泛应用于安全关键系统的开发过程中,它支持在早期就对系统进行安全分析和验证,有效地控制开发时间和成本,并降低系统出现风险的可能性。但与此同时,需求与设计模型之间仍然存在着沟壑,设计模型是否很好地满足用户所提出的需求在完成系统设计后仍需验证。针对系统建模语言缺乏精确形式化语义难以进行模型验证的问题,文中给出一套从 SysML 设计模型到 NuSMV 模型转换的语义规则,实现了一个自动转换程序,将 SysML 模型文件转换成 NuSMV 输入文件,进而利用 NuSMV 实现 SysML 模型的验证。最后通过一个铁路控制系统的案例证明了该方法的有效性。

关键词:需求工程;模型转换;形式化验证;模型驱动开发
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2019)10-0153-04
doi:10.3969/j.issn.1673-629X.2019.10.030

Formal Verification of SysML Model Based on NuSMV

DENG Liu-meng, GE Xiao-yu, WAN Wei-jian
(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: System development in aerospace road traffic and other high security areas needs to ensure high security and high reliability. It is especially important for the reasonable modeling and model verification of such systems. The model driven development (MDD) method has been widely used in the development of safety-critical systems, which supports the safety analysis and verification of the system at an early stage, effectively controlling development time and cost and reducing the possibility of system risk. At the same time, there is still a gap between the textual requirement and the design model. Whether the design model can well meet the user's requirements still needs to be verified after the completion of the system design. Addressing the problem of the lack of precise formal semantics for the systems modeling language (SysML), a set of semantic rules for the transformation from SysML design model to NuSMV model is given. An automatic conversion program is implemented to convert the SysML model file into NuSMV input file, and then to verify the SysML model by NuSMV. In the end, we prove the effectiveness of this method through the case of railway control system.

Key words: requirement engineering; model transformation; formal verification; model driven development

0 引 言

在过去多年,软件开发面临了多个挑战,新的需求和存在系统不断增长,系统也变得越来越复杂,以至于很难及时地对它们进行构建。为了解决这些问题,出现了很多新的方法,其中最突出的一个就是模型驱动开发。

模型驱动开发代表了一套理论和工业化软件开发的方法框架,在软件开发全生命周期中系统的使用模型作为主要工件,主要是为了解决软件的两个根本危机:复杂性和变更能力。但与此同时,模型驱动开发也带来了一些问题:使用自然语言描述的需求与严格定义的模型之间的鸿沟无法很好地连通[1]。此外,对于 SysML 描述的图形化模型,目前缺乏严格有效的分析和验证方法。

针对以上存在的问题,文中给出了从 SysML 模型到 NuSMV 输入模型的转换规则,并实现自动化程序完成这一转换。接着利用 NuSMV 模型检测的方法来验证 SysML 模型的正确性。

收稿日期:2018-11-25 **修回日期**:2019-03-26 **网络出版时间**:2019-06-26
基金项目:国家自然科学基金(617722770);南京航空航天大学开放基金(kfjj20171606)
作者简介:邓刘梦(1995-),男,硕士研究生,CCF 会员(88983G),研究方向为软件安全、嵌入式系统建模与分析、需求可追踪性。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20190626.0829.038.html

1 SysML 系统建模

SysML 是目前业界常用的系统体系结构建模语言,可用于由软硬件、数据和人综合而成的复杂系统的分析与设计。然而,为了保证一定的易读性,SysML 采用半形式化的描述方法来定义语义,使用自然语言描述约束和详细语义,力求在形式严格和易于理解间找到平衡[2]。在实际中,其图形化的建模方式十分简洁直观,关系链接与约束描述等方式也进一步缩小了模型驱动开发过程中需求描述与模型设计制品间的沟壑。但是,其牺牲的部分就是缺乏精确的语义,难以进行严格的语义分析以及正确性验证。

SysML 是一种图形化建模语言,是对象管理组织(object management group,OMG)在对 UML2.0 的子集进行重用和扩展的基础上提出的一种新建模语言[3]。它为软件体系结构建模提供了丰富的图标,涵盖了从系统需求到设计阶段的各项要求,广泛应用于航空航天软件开发过程。它致力于建模具有众多组件、难以描述、理解、预测、管理、设计以及更改的系统,并提供了表达个人需求及其组成的手段,已被学术界和工业界所广为接受,作为一种标准建模语言[4]。

SysML 为系统的结构模型、行为模型、需求模型和参数模型定义了语义。结构模型强调系统的层次以及对象之间的相互连接关系,包括类和装配。行为模型强调系统中对象的行为,包括它们的活动、交互和状态历史[5]。

文中主要使用 SysML 的块定义图对系统的静态结构进行描述,使用状态机图对系统的动态迁移进行描述。SysML 中,块(block)是系统描述的最小建模单位,可以用来描述每一个单独的组件,同时也是描述系统结构特征和行为特征的单元。SysML 块以 UML 类图为基础,并扩展了 UML 复合结构的一些特征[6]。块定义图(block definition diagram)则是用于描述块信息的图。它描述了块的属性值、块的组成部分、块的操作以及对其他块的参考等[7]。而状态机图(state machine diagram)则是用来描述系统的状态迁移情况[8]。其中状态转移用来描述对象对事件的响应情况。关于 SysML 块定义图以及状态机图的实例将在下一节给出。

2 NuSMV 模型

针对 SysML 模型进行验证,采用 NuSMV 作为验证工具。

2.1 输入语言分析

NuSMV 模型中,系统被描述为模块化的层次结构,支持定义组件的重用[9]。其支持的数据类型主要有枚举类型、布尔类型和固定数组等。基本上,一个完整的 NuSMV 模型文件主要由两部分组成:系统模型和待验证的系统性质[10-11]。

NuSMV 系统模型部分主要用于描述系统的状态以及状态迁移关系,刻画出系统的静态结构与动态行为[12]。通过关键字 MODULE 来定义模块,通常每一个模块对应一个系统组件[13]。通过主模块中的 SPEC 字段进行待验证需求性质描述,同时支持计算树逻辑(computation tree logic,CTL)和线性时序逻辑(linear-time temporal,LTL)的表达式[14-15]。

2.2 SysML 模型到 NuSMV 模型的转换

本节根据 SysML 模型与 NuSMV 模型的特点[16],给出转化规则,并实现工具完成 NuSMV 模型实例的自动化生成。

首先对 SysML 中的静态图进行转换,文中主要使用的是块定义图。

规则 1:模块名声明。

描述:对于 NuSMV 中的模块名根据块定义图中的名称进行命名。

规则 2:静态变量声明。

描述:对于块定义图中定义的属性都须在相应的模块中通过 VAR 关键字进行声明。

规则 3:变量初值。

描述:对于块定义图中所有属性的初值都须在相应的模块中通过 ASSIGN 关键字进行声明。

接下来对 SysML 中的动态行为模型进行转换。SysML 中主要通过状态机图对系统的状态迁移进行刻画,系统中可能出现的状态迁移,都存在对应的状态机图[12]。从另一个侧面来看,状态图也可以理解为对块定义图动态的补充,故在转换中,应将其放入相应的模块中,而不是重新声明模块。

规则 4:状态机图声明。

描述:对于状态机图转换不重新进行模块声明,将其状态迁移关系通过 TRANS、next 等关键字描述加入到从属的块定义图对应的模块中。

例如:Car 对应的状态机图,其描述的状态迁移关系都应该加入到 MODULE car 中。

规则 5:状态变量声明。

描述:如果一个块定义图存在一个对应的状态机图,则应该在此块对应的模块中通过 VAR 声明一个 state 变量。

规则 6:状态变量赋值。

描述:状态机图中 state 的取值是由去除初始状态和结束状态后所有状态值构成的枚举集合,其初始值应为状态机图中 Initial 节点指向的第一个状态,通过 ASSGIN 声明。

例如:对于汽车 car,通过一个状态机图描述其运

行状态可能存在运行或者停止两种状态(见图 1),那么 MODULE car 中 VAR 字段就需加入 car_state:{stop,running}声明,初始状态为 stop,通过 ASSGIN init(car_state):=stop 字段进行声明。

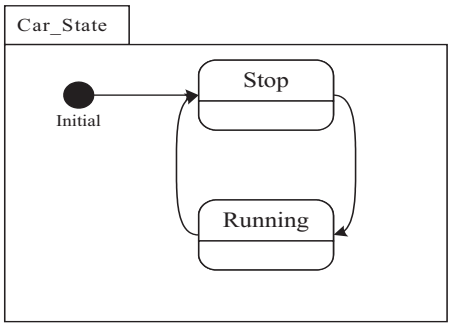

图 1　汽车运行状态机图实例

规则 7:状态迁移。

描述:状态机图确定的状态转变使用 next 进行描述,并通过 case 来表达分支情况。

例如:car_state 当前状态为 stop 下一状态为 running 和当前状态为 running 下一状态为 stop 表示如下:

next(car_state):=
case
car_state=stop:{running};
car_state=running:{stop};
esac

规则 8:状态迁移条件。

描述:如果状态机图中的状态迁移存在迁移条件,则需将该守卫条件加入到迁移描述字段中(见图 2)。

例如:在汽车启动前应确定车门是关闭的,否则无法启动。

next(car_state):=
car_state=stop&door. closed=1:{running};

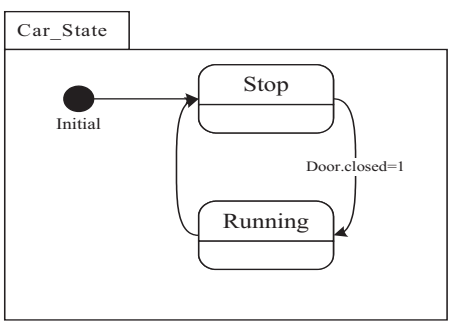

图 2　存在守卫条件的状态迁移实例

3　实验与案例分析

根据前几节的理论分析,实现了一个 SysML 模型到 NuSMV 模型自动转换的工具。下面通过案例演示。

案例的场景如下:在铁路控制系统中,在火车通过路口时需要关闭公路两侧的栅栏,保证在火车通过的过程中汽车无法驶入路口,避免发生火车汽车相撞的事故。首先通过 SysML 建模工具建立该场景的模型如下:

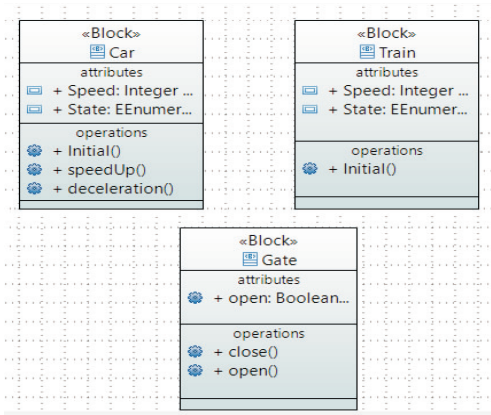

图 3　铁路案例 SysML 块定义图

图 3 中块定义图描述了系统的静态结构信息,图 4 中状态机图则描述了系统的状态以及迁移关系。

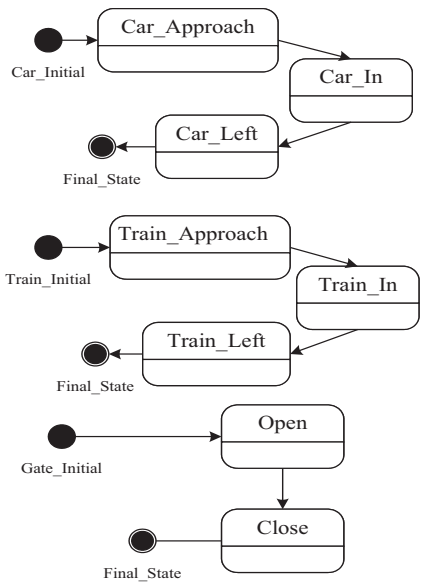

图 4　铁路案例 SysML 状态机图

在建模分别得到块定义图和状态机图后,利用工具将模型导出为 XMI 文件格式以供后续转换使用。接着将得到的 SysML 模型文件输入到自动转换工具中即可完成转换,得到铁路系统的 SMV 文件(见图 5)。

图 5　SysML 模型转换工具界面

得到设计模型的实例后,即可利用已有的 NuSMV 工具来检测系统需求是否被设计模型所实现。首先给出一条安全需求:该系统模型不得出现汽车和火车同时驶入路口的情况,避免事故发生。接着在得到的 SMV 文件中写入该需求性质 LTL 表达式:LTLSPEC G! ((Car_state = Car_in) & (Train_state = Train_in))。最后在 Windows10 下采用命令行形式运行 NuSMV 工具检测该 SMV 文件得到的结果如图 6 所示。

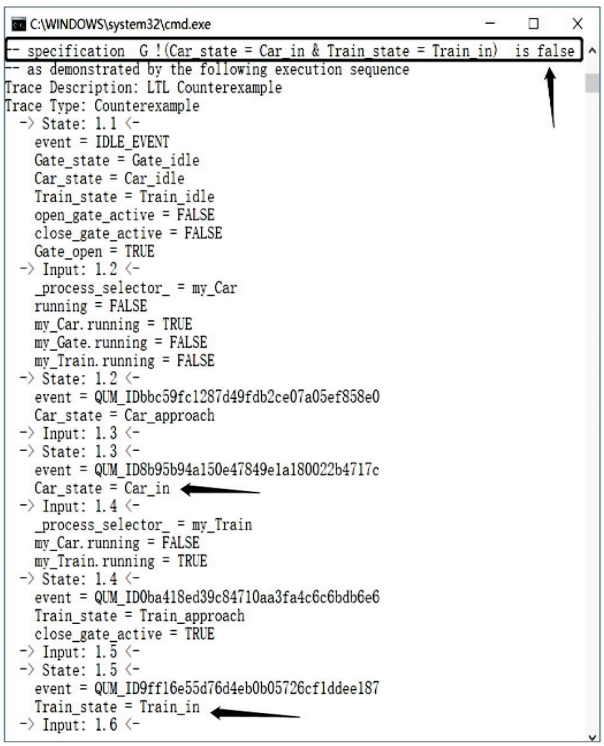

图 6 需求验证结果

得到 LTL 公式检测结果为 false,即该需求没有被满足。NuSMV 工具给出了反例,观察到在 1.5 状态时同时出现了汽车和火车均进入路口的情况。

同时表明文中转换工具得到的 SMV 文件可以很好地作为 NuSMV 工具的输入,证明了该方法的有效性。

4 结束语

针对 SysML 模型缺乏精确语义而难以进行模型正确性验证的问题,给出了一个通过模型转换技术实现模型验证的解决方法。实现了一个从 SysML 设计模型到 NuSMV 模型自动转换的工具,最后利用转换得到的 SMV 文件作为模型检测器的输入即可进行 SysML 模型的验证。

参考文献:

[1] 刘军霞,熊选东,王松锋. 基于随机 Petri 网的 SysML 状态机图的验证[J]. 计算机应用与软件,2013,30(6):202-208.

[2] 夏 宇. 基于 NuSMV 和 STPA 的 RBC 交接场景安全分析方法研究[D]. 北京:北京交通大学,2018.

[3] SALMAN Y D,HASHIM N L. Automatic test case generation from UML state chart diagram: a survey[M]//Advanced computer and communication engineering technology[s. l.]:Springer International Publishing,2016:123-134.

[4] FRIEDENTHAL S, MOORE A,STEINER R. A practical guide to sysML[M]. [s. l.]:[s. n.],2011:41-46.

[5] 曹德建,黄志球,阚双龙,等. 基于故障扩展 SysML 活动图的软件安全性分析方法研究[J]. 小型微型计算机系统,2015,36(9):2067-2074.

[6] 王 栋. 基于 SysML 的武器装备体系结构建模与仿真方法研究[D]. 长沙:国防科学技术大学,2009.

[7] NEJATI S,SABETZADEH M,ARORA C,et al. Automated change impact analysis between SysML models of requirements and design[C]//Proceedings of the 24th international symposium on foundations of software engineering. Seattle,WA,USA:ACM,2016:242-253.

[8] ANDRADE E,MACIEL P,CALLOU G,et al. A methodology for mapping SysML activity diagram to time petri net for requirement validation of embedded real-time systems with energy constraints[C]//International conference on digital society. Cancun,Mexico:IEEE,2009:266-271.

[9] CIMATTI A, CLARKE E, GIUNCHIGLIA F,et al. NUSMV:a new symbolic model checker[J]. International Journal on Software Tools for Technology Transfer,2000,2(4):410-425.

[10] ARCAINI P,GARGANTINI A,RICCOBENE E. Asmeta-SMV:a way to link high-level ASM models to low-level NuSMV specifications[C]//Proceedings of the second international conference on abstract state machines,alloy,B and Z. Orford,QC,Canada:Springer-Verlag,2010:61-74.

[11] 陈冬火,刘 全. 基于符号执行和 LTL 公式重写的测试用例产生方法[J]. 计算机研究与发展,2013,50(12):2661-2675.

[12] 周玉平. 基于 UML-NuSMV 模型的列控系统需求阶段的安全分析[D]. 北京:北京交通大学,2015.

[13] 何 洋,洪 玫,祁琳莹,等. 基于模型检测工具 NuSMV 的功能测试用例生成方法[J]. 计算机应用,2015,26(z2):155-159.

[14] 王 曦,徐中伟,梅 萌. 基于模型检测的软件安全性验证方法[J]. 武汉大学学报:理学版,2010,56(2):156-160.

[15] KAN Shuanglong,HUANG Zhiqiu. Detecting safety-related components in statecharts through traceability and model slicing[J]. Software: Practice and Experience,2017,48(2):428-448.

[16] 俞晓锋,王立松. SysML 状态图合理性验证研究与实现[J]. 电子科技,2014,27(5):127-131.