

基于 Spin 的 SysML 时序图与活动图一致性检测

胡良文, 马金晶, 孙 博

(南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016)

摘要: 系统建模语言 (Systems Modeling Language, SysML) 对复杂系统多视角建模时, 容易造成多视图描述语义冲突、矛盾等不一致问题, 可以通过形式化验证方法, 来提高模型的一致性。然而, 受制于传统的形式化检测方法不能做到完全自动化, 并且需要繁杂的公式推理, 导致多数验证方法仅限少数专家使用并且非常耗时。为了解决 SysML 时序图与活动图模型之间存在的一致性问题, 提出一种自动转换验证框架。首先基于已构建的模型和转换规则, 将时序图进行分解转换为活动图, 然后分别映射为 Spin 的输入模型, 并对模型的交互一致性执行自动化验证。实验结果表明, 该方法可以有效识别和转换时序图, 并能准确地向 Promela 实施映射和验证, 为一致性验证的演化提供支持。

关键词: 系统建模语言; 模型检测; 时序图; 活动图

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2015)09-0031-06

doi: 10.3969/j.issn.1673-629X.2015.09.007

Consistency Check Between SysML Sequence and Activity Diagram Based on Spin

HU Liang-wen, MA Jin-jing, SUN Bo

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China)

Abstract: As systems modeling language models multiple views for complex system, it can lead to a variety of inconsistent problem in the model. Formal verification methods can be used to improve consistency of the model. In the reason of that traditional formal methods can't be complete automation and need complex formula deduction, most verification can only be used by experts and it's very time-consuming. To address the consistent problems of the SysML sequence diagram and activity diagram, propose an automated transition and checking consistency approach. The sequence diagram can be decomposed and transformed to an activity diagram using the mapping rules. The diagrams are mapped to the input model of Spin. Then, the models are analyzed and verified by Spin. The experimental results show that the approach can correctly transform complex sequence diagrams in real projects and effectively verify consistency of them. This indicates that the approach is helpful for model checking evolution.

Key words: SysML; model checking; sequence diagram; activity diagram

0 引言

系统工程是以研究解决复杂工程问题而逐渐发展起来的多学科方法, 由于目前软件系统规模及复杂性持续发展使得要求更加严格和系统的工程方法。为了响应这种要求, 系统工程实践正在经历一场从基于文档的方法到基于模型的方法的深刻转型, 基于模型的方法可以帮助管理和处理系统的复杂性, 同时能够改善设计质量及缩短开发周期。

目前, SysML 是一种被广泛应用的标准工业建模语言, 它能够对多种复杂系统进行详细设计、分析和验

证^[1]。SysML 架构是基于四层元模型结构, 提供不同视图对系统进行建模和描述, 这些不同的视图能够帮助架构师处理复杂的软件系统^[2]。由于 SysML 多种视图从软件系统的不同视角进行描述, 所以它们彼此之间不可避免地会发生重叠并且存在很强的依赖性。换句话说, SysML 是一种半形式化的建模语言, 缺乏精确的形式化描述和规格说明^[3]。当对一个复杂软件系统进行建模或描述时, SysML 各种视图很难保证其一致性。而当各种视图描述的语义或语法存在冲突和矛盾时, 将给软件系统后期开发和维护带来严重的错

收稿日期: 2014-10-27

修回日期: 2015-01-29

网络出版时间: 2015-08-26

基金项目: 国防科工局“十二五”重大基础科研项目 (c0420110005)

作者简介: 胡良文 (1987-), 男, 硕士生, 研究方向为模型检测、软件工程。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150826.1556.024.html>

误,因此对系统的不一致进行分析、检测和解决就显得尤为重要。

由于 SysML 时序图与活动图对复杂系统的动态行为进行描述时存在着重叠,视图间可能产生不一致问题。为了解决该问题,文中提出将一致性转换和一致性比较分开,先将时序图转换为活动图,然后结合模型检测工具 Spin 进行一致性的验证。与文献[4-6]相比,文中方法的优势是不需要借助层次自动机、有限自动机等第三方模型。该框架首先根据文中提出的转换规则将构建完成的时序图源模型转换生成相应的活动图模型;然后将转换而成的活动图和原始活动图映射为 Spin 的输入 Promela 模型;最后利用检测工具 Spin 对模型进行一致性验证。实验结果表明,基于文中算法可以有效识别活动图和时序图,并能准确向 Promela 实施映射和一致性验证,同时也证明了所提出算法的有效性和正确性。

1 SysML 时序图与活动图

1.1 时序图与活动图概念

时序图(Sequence Diagram)是强调对象按时间交互的行为图,用于表达信息系统的动态行为^[7]。其主要用途是为表达对象间的行为和事件发生的交互序列,提供泛化与直观的解释,即把用例描述的宏观需求,转化并表示为进一步、更加具体层次的微观表述。其中时序图由对象、生命线、同步消息、异步消息、创建消息、返回消息和激活,以及一些组合片段构成。SysML 定义了 11 种交互的组合片段,但在日常建模工作中,最可能用到的只有选择、并行、循环和可选四种^[8],限于篇幅对组合片段只对这四种进行讨论。

定义 1(时序图):时序图可以定义为一个四元组 $SD = (Lif, Act, Msg, Com)$ 。其中: Lif 表示时序图中的对象集; Act 表示时序图中的激活元素集; $Msg \subseteq S \times R \times I$ 表示时序图中的消息集, $S \subseteq Lif$ 表示消息的发送对象集, $T \subseteq Lif$ 表示消息对象接收者集, I 表示消息的唯一标识; Com 表示组合片段集, $Com = \{alt, opt, par, loop \dots\}$ 。

定义 2(时序图语义):消息的发生位置关系可以定义为: $\prec_{pre} = \{(m, m'), m, m' \in Msg \text{ such that: } time(m) < time(m')\}$ 。同时时序图 SD 能表达多种逻辑语义: $[SD] = \langle S_1, S_2, \dots, S_n \rangle$, 其中每种逻辑语义 S_i 由一组事件踪迹来表达,即 $S_i = \langle m_1, m_2, \dots, m_k \rangle$, $\forall m_i \in Msg$ 并且 $m_i \prec_{pre} m_{i+1}$ 。

活动图(Activity Diagram)也是 SysML 动态行为视图之一,类似于时序图主要对信息系统的动态行为进行建模,描述复杂系统的控制流程和并发行为,在用例图之后提供了系统分析和充分描述,并用于展现参与

行为的类所进行的各种活动的顺序关系。

定义 3(活动图):活动图可以定义为一个五元组, $AD = (Nodes, Edges, Initial, Final, Type)$ 。其中, Nodes 表示活动图的节点元素集,定义 IDNodes 为活动节点的唯一标识; $Edges \subseteq S \times R \times I$ 表示活动图的控制流转移, $S \subseteq Nodes$ 表示控制流的发送节点集, $T \subseteq Nodes$ 表示接收节点集, I 表示节点的标识集; Initial 表示起始节点; Final 表示终止节点; Type 表示节点的类型集, $Type = \{start, end, activity, join, fork, merge, decision\}$, 分别表示:起点、终点、活动、汇合、分岔、合并、分支等节点类型,定义类型函数 $type: Nodes \rightarrow Type$ 。

在 SysML 所有模型中可以分为两种:静态图和动态图。其中时序图和活动图都是动态图,时序图主要用途就是泛化用例功能和需求,描述传送消息的时间顺序;而活动图是在用例图之后展现参与对象进行的各种控制活动顺序。两者的建模内容存在了重叠,在系统设计初期可能发生冲突和矛盾等不一致问题,这将给软件系统的开发后期带来致命后果。

1.2 一致性验证框架

标准建模语言 SysML 能够支持对同一个系统进行多视角描述,但也可能导致多个视角描述的内容发生重叠并出现不一致问题。针对此问题,提出一种针对 SysML 时序图与活动图之间的一致性验证方法。

在通过建模工具构建 SysML 时序图与活动图源模型,并保存为 XMI 文档后,框架首先解析图形信息并将其元素信息解析成为树状结构;然后将时序图转换为活动图;继而基于文中提出的活动图映射规则,将原始活动图和由时序图转换生成的活动图分别映射为 Promela 模型,最终通过验证工具 Spin 检测时序图与活动图之间的一致性。

2 SysML 时序图转换

SysML 根据软件系统的层次级别和建模阶段将其抽象为面向对象的交互描述和行为描述。为了实现时序图与活动图之间的映射描述及一致性检测,本节首先实现一致性的转换阶段,参考文献[9-11]定义了消息的动作关系和顺序关系,将时序图转换生成成为活动图。

2.1 基于消息的动作关系

在时序图中,对象间消息传递的同时传递了流的控制,每个对象只控制自身消息的发送与接收动作,可以随时接收不同对象发来的消息。用 ! m 表示发送消息动作,用 ? m 表示接收消息的动作。针对异步消息,分别定义顺序图中消息发送和接收动作的两种动作关系。

(1) 发送-发送相继关系。

如果两个消息 m 与 m' 由同一对象 o 发送,并且 m 和 m' 是无间断连续,那么动作 $!m$ 和 $!m'$ 满足发送-发送相继关系,如图 1(a)和(b)所示。

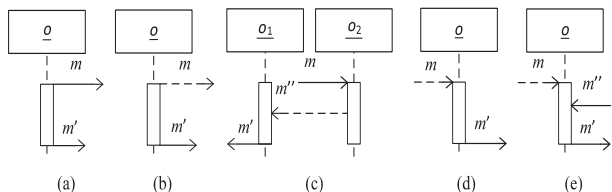


图1 相继与因果关系

而如果对象 o 发送消息 m 与 m' ,而它们是有间断连续,同时在它们之间存在一个消息 m'' ,且 m'' 是 m 的返回消息,并且 m 是异步消息,仍然称动作 $!m$ 与 $!m'$ 之间满足发送-发送相继关系,如图 1(c)所示。

根据如上语义,定义如下的动作关系:

$$<_{SSF,o} =$$

$$\left\{ \begin{array}{l} (!m, !m'); m, m' \in M, \exists o \in O; \text{ such that: } \\ o = \text{from}(m) = \text{from}(m') \text{ and } m <_{\text{pre}} m' \text{ or} \\ m'' \in M, m <_{\text{pre}} m'' <_{\text{pre}} m', o = \text{from}(m) = \\ \text{from}(m') = \text{to}(m'') \end{array} \right\}$$

$\Rightarrow m$ is an asyn. m'' is the return message of m

(2) 因果关系。

动作之间的因果关系是指,两个动作 a_1 与 a_2 之间存在了一种时序关系,当动作 a_1 能以某种方式引起动作 a_2 的发生,同步消息、异步消息与其返回消息之间具有一种平凡的满足因果关系的情况。

两个消息 m 与 m' , m 由对象 o 接收, m' 由对象 o 发送,并且 m 和 m' 是不间断连续的相继关系,那么称 $?m$ 和 $!m'$ 之间满足因果关系,如图 1(d)所示。

如果两个消息 m 与 m' , m 由对象 o 接收, m' 由 o 发送,在它们之间有且只有 o 接收的消息,而没有由 o 发送的消息,并且这些由 o 接收的消息与 m 不是同一个对象发送的,它们与 m 的关系是不能确定的。此时认为动作 $?m$ 与动作 $!m'$ 之间仍是满足因果关系,如图 1(e)所示。根据上述分析,定义关系:

$$<_{CS,o} =$$

$$\left\{ \begin{array}{l} (?m, !m'); m, m' \in M, \exists o \in O; \\ \text{such that } o = \text{to}(m) = \text{from}(m'); \text{ or} \\ \forall m'' \in M, \text{to}(m'') = o \text{ and } \text{from}(m'') \neq \text{from}(m) \end{array} \right\}$$

2.2 转换规则

时序图能给出多个对象交互过程中消息发生的顺序关系,但当时序图中出现异步消息时,就可能引起消息的并发,这时呈现的消息顺序就不能真实反映交互顺序^[9]。为明确消息的执行顺序,根据上述对消息间的动作关系,即发送-发送相继关系和因果关系,对消息之间的执行顺序表示为:

$$<= \{ (m, m'), \text{ such that } m, m' \in M \text{ and } (!m$$

$$<_{SSF,o} !m' \text{ or } ?m <_{CS,o} !m' \}$$

其中,称 m 是 m' 的前继, m' 是 m 的后继。

从时序图模型到活动图模型的转换是根据消息的相继顺序关系,并且基于相应一致性转换规则为基础,得到描述时序图并发控制流的活动图。借鉴文献[9, 12-13]提出时序图各种元素到活动图基本元素的转换规则。

规则1:时序图中的消息对应转换为活动图中的活动节点;其中消息的相继顺序关系 $<$ 对应于控制流,在活动图体现为边,如图 2(a)所示。

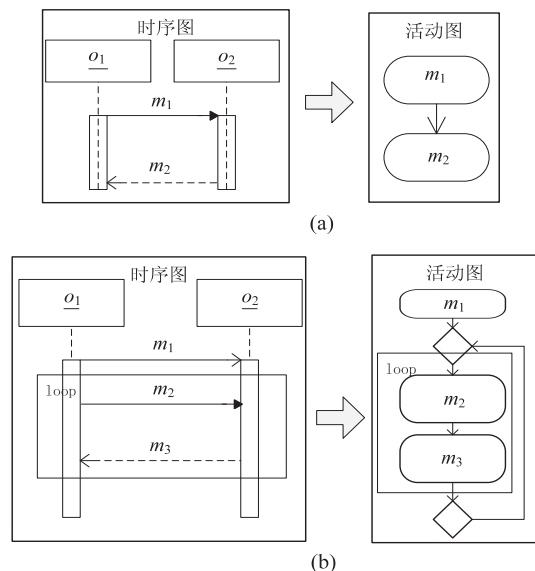


图2 消息的转换规则

规则2:如果一个消息有多个后继时,则此消息对应的活动图节点和其后继节点之间由一个分叉节点(fork)进行连接。而当一个消息有多个前驱,则此消息对应的活动节点和其前驱之间由一个汇合节点(join)连接,实现并发控制流的同步支持。

规则3:对以没有前驱的消息,则将起始节点作为此消息的前驱节点;而没有后继的消息,则将终止节点作为此消息的后继节点。

规则4:对以每个复合片段的内部作为一个独立的控制流。其中如果复合片段在前,则只考虑复合片段中的最后一个消息与外部消息的关系;如果复合片段在后,则只考虑复合片段第一个消息与外部消息的关系,如图 2(b)所示。消息映射规则同上述规则,对于不同的复合片段由下述映射:

- (1) loop/alt/opt/break 片段对应着一个分支节点(decision);
- (2) par 片段对应一个分叉节点(fork);
- (3) ref 片段对应一个活动子图。

2.3 模型转换

由于对待同一个系统,时序图模型与活动图模型之间的结构以及描述方法等关注点存在差别。时序图

强调消息按时间顺序进行交互,它描述了对象之间传送消息的时间顺序;而活动图描述系统的工作流程和并发行为,展现了参与行为的类所进行的各种活动的顺序关系,使得不同视角上的模型存在较大差异,这种差异使得模型之间映射关系复杂,难以定义。

用活动图来建模时序图的并发控制流,称其为基于消息语义的控制流图,为实现时序图生成活动图,提出如下的转换算法:

算法 1:模型转换算法。

输入:XML 形式的时序图;

输出:时序图对应活动图。

Step1:加载 XML 源文件,解析其中的时序图节点信息;

Step2:针对时序图所有对象,找出每个对象生命线上消息可能的动作关系,并进一步确定各个消息的相继顺序;

Step3:对所有关系对进行排序,读取集合中的关系对。

(1)匹配当前节点类型,并读取相应某一条转换规则;

(2)如果是终止元素,则跳转到 Step4;否则转到(3)。

(3)按照当前的转换规则转换当前元素为活动图元素,完成后跳转至 Step3。

Step4:结束转换。

3 一致性验证

应用 Spin 验证时序图与活动图一致性的关键问题是如何将模型映射为 Spin 的输入模型 Promela。本节首先介绍基于进程同步的方法,然后介绍 SysML 模型到 Promela 模型的转换算法。

3.1 进程同步

经 XML 解析和时序图转换生成两个独立的活动图:SysML 活动图(AD)和时序图转换图(SD)。文中借鉴文献[14-15]的转换算法,提出以活动节点为单位转换为 Promela 模型的独立进程,同时为保证验证算法能够高效并发执行,从横向和纵向上实现进程之间的同步通信,如图 3 所示。

进程纵向通信是指 SD 或 AD 内的进程进行通信,即 SD 与 AD 进程之间不进行通信。即 SD 进程通过共享通道 SDChan 进行交互,SD 进程只有在接收到其前驱节点的消息后才执行,并向后继节点发送消息;而 AD 则通过共享通道 ADChan 实现。

进程横向通信是指 SD 与 AD 进程之间的同步消息交互,即 SD 进程将验证信息发向共享通道 synchChan,而 AD 进程在 synchChan 通道上等待验证消息,

接收验证一致后通过共享通道 replyChan 向 SD 发送回复消息。

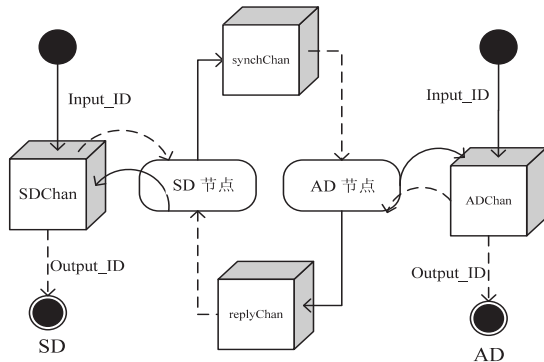


图 3 进程间的同步机制

3.2 验证算法

由于 SysML 时序图模型与活动图模型之间在对同一个复杂软件系统进行建模时,两者的建模内容存在了重叠,容易引发系统语法或语义之间不一致问题,从而导致系统开发后期出现一系列不可修复性问题,因此保证时序图与活动图之间的一致性就显得尤为重要。由此定义时序图与活动图之间的一致性如下:

定义 4(一致性检测准则):一个时序图 SD 包含 n 条语义踪迹 $[SD] = \langle S_1, S_2, \dots, S_n \rangle$, $S_i = \langle msg_1, msg_2, \dots, msg_k \rangle$, $1 \leq i \leq n$; 一个活动图 AD 能组成 m 条语义踪迹 $[AD] = \langle T_1, T_2, \dots, T_m \rangle$ 。如果 $[SD]$ 中的每条踪迹都和 $[AD]$ 中的一条踪迹相符合,即 $\forall S_i \in [AD]$, 那么这个时序图就与所关联的活动图是一致的。

文中借助 Spin 的安全性和正确性检测,同时借鉴文献[16]的轨迹等价性,从纵向和横向上提出同步验证算法检测时序图与活动图之间的一致性,具体细节如算法 2 所示。

算法 2:时序图与活动图的一致性验证算法。

输入:SD 活动图和 AD 活动图;

输出:验证结果。

Step1:SD 和 AD 起始节点进程向后继进程发送可执行消息;

Step2:每个进程按如下步骤执行:

(1)进入前驱进程消息等待状态,如果最终能够接收到可执行消息则转至(2),否则跳转至 Step3;

(2)如果是 SD 进程则发送验证消息,然后进入回复消息等待状态;如果是 AD 进程直接进入验证消息等待状态,转(3);

(3)如果 SD 能够最终接收到一致性回复消息,而如果 AD 进程最终能够接收到验证消息,并验证一致则发送回复消息,然后都转(4),否则转 Step3;

(4)SD 和 AD 进程如果是终止进程,直接跳转 Step4;否则先向后继进程发送可执行消息,然后跳转

Step2。

Step3:返回不一致结果;

Step4:返回一致结果。

4 案例分析

本节通过对高铁票务系统的订单验证实例进行建模,并经文中所提的验证框架对所构建的模型进行解析、转换和验证,最后对实验结果产生的反例进行分析可发现错误的原因,从而来说明所提出的 SysML 一致性验证框架对检测模型的可行性。

4.1 问题描述

高铁订票系统提供在线查询、预定和购票的功能。这能给旅客提供一个便捷、快速的购票途径,同时为售票窗口减轻工作负担。在旅客进入系统并提交订单后,系统首先根据旅客账户所提交的信息对订单进行验证。如果订单验证成功,系统将会并发完成三件工作:向旅客发送交易的确认消息,更新账户信息以反映交易结果,以及通过借记账户的资金变动来结清交易。当上述三项进程并发完成后,系统将会合并为单个进程结束订单;如果订单执行失败,系统会给旅客发送失败通知,并结束订单。

4.2 SysML 活动图与时序图建模

根据高铁订票系统的功能需求可对订单验证进行活动图建模,如图 4 所示。在客户向系统提交订单后,系统将进行订单的验证和执行过程,当验证成功或失败时会进行不同的处理过程,同时都会通过短信发送平台向客户返回信息。

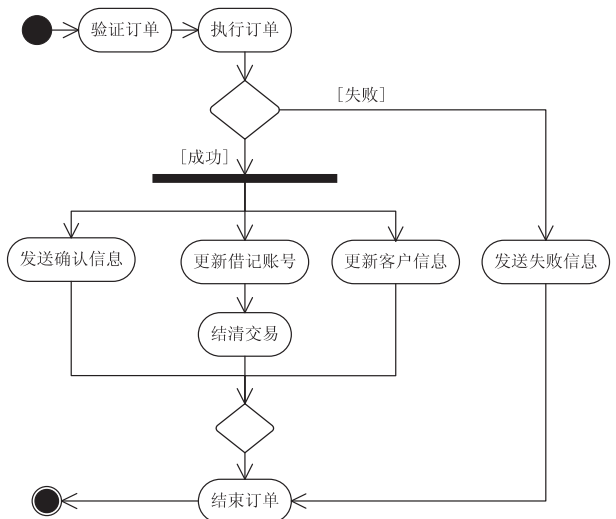


图 4 订票系统的活动图

同样根据高铁订票系统的功能需求进行时序图建模,能构建如图 5 所示的时序图。首先用户向订购模块提交请求,那么订购模块与数据模块将进行一系列的信息交互来完成订单的验证;最后会通过数据模块向用户发送验证结果。

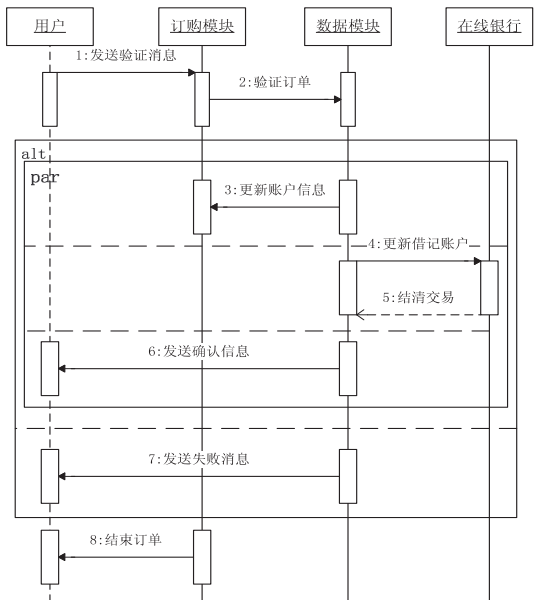


图 5 订票系统的时序图

4.3 模型验证与结果分析

根据文中提出的转换规则和映射算法,分别输入图 4 和图 5 所示的 XML 形式模型,对 SysML 时序图和活动图进行解析、转换和映射为 Promela 模型,最后经检测器 Spin 验证结果,如图 6 所示。

```
(Spin Version 6.2.3 -- 24 October 2012)
Warning: Search not completed
+ Partial Order Reduction

Full statespace search for:
never claim      - (none specified)
assertion violations  +
acceptance cycles - (not selected)
invalid end states  +

State-vector 2248 byte, depth reached 251, errors: 1
252 states, stored
0 states, matched
252 transitions (= stored+matched)
0 atomic steps
hash conflicts: 0 (resolved)
```

图 6 验证结果

结果表明,Spin 检测出的时序图模型与活动图模型之间存在不一致,即模型检测工具执行到 252 状态时发生了超时错误,说明在此处发生了不一致。通过错误路径跟踪可以发现在退出节点之前的汇合进程等待超时。经分析发现当验证成功后将有三个并发进程会汇合为单个进程,但误用了合并节点,从而导致了与时序图的不一致错误。

5 结束语

文中针对现有模型检测算法无法实现自动化验证 SysML 时序图与活动图之间的一致性,提出了一个基于 Spin 的 SysML 时序图与活动图检测框架。为检测 SysML 模型之间的一致性,定义了时序图转换为活动图模型的四种转换规则,并且提出了 SysML 模型到 Promela 模型的映射算法。通过此框架能够识别复杂的时序图与活动图,并能对其分解,以及转换为相应的

Promela 模型。最后,以高铁订票系统架构进行了实验和分析。结果表明,文中的验证方法能够检测到时序图与活动图之间存在的 inconsistency 问题,为 SysML 模型的一致性演化提供支持。

文中旨在利用自动化工具辅助工程师在软件开发的早期阶段发现系统设计的 inconsistency 错误,提高软件系统的开发速度和效率,减少传统形式化检测方法中对验证人员的数学背景要求,使其能从繁重的手工推导过程中解脱出来。由于没有考虑时间和空间的局限性,因此下一步工作是对算法进行优化,使其能在有限的条件下同样适用于复杂的大型软件系统建模中。

参考文献:

- [1] 刘军霞,熊选东,王松峰. 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] Florian M, Albert A, Daniel W, et al. Multi-view modeling in sysml: thematic structuring for multiple thematic views[J]. Procedia Computer Science, 2014, 28(1):531-538.
- [3] Ando T, Yatsu H, Kong W, et al. Translation rules of SysML state machine diagrams into CSP# toward formal model checking[J]. International Journal of Web Information Systems, 2014, 10(2):3-3.
- [4] 陈 卉, 窦万峰. UML 顺序图与状态图的一致性检查[J]. 计算机工程, 2008, 34(18):62-64.
- [5] 杜 杰, 江国华. 基于模型检测的 UML 状态图和顺序图一致性检测[J]. 电子科技, 2012, 25(2):100-104.
- [6] 汪 胜. UML 顺序图与状态图一致性检验[D]. 广州:中山大学, 2009.
- [7] Friedenthal S, Moore A, Steiner R, et al. A practical guide to SysML[M]. Waltham: Morgan Kaufmann, 2012.
- [8] Delligatti L. SysML distilled: a brief guide to the systems modeling language[M]. [s. l.]: Pearson Education, 2013.
- [9] 王洪媛. UML 行为模型之间模型转换的研究[D]. 长春: 吉林大学, 2007.
- [10] 张 频, 罗贵明. UML 模型检测方法的研究[J]. 计算机应用, 2007, 27(10):2493-2497.
- [11] 王松峰, 熊选东, 付建丹, 等. 基于 Petri 网的 SysML 活动图的分析与验证[J]. 计算机科学, 2012, 39(9):138-142.
- [12] Kaliappan P S, Koenig H. An approach to synchronize UML-based design components for model-driven protocol development[C]//Proc of 4th IEEE software engineering workshop. [s. l.]: IEEE, 2011:27-35.
- [13] Torre D, Labiche Y, Genero M. UML consistency rules: a systematic mapping study[R]. Carleton: Carleton University, 2014.
- [14] Staines T S. Transforming UML sequence diagrams into petri nets[J]. Journal of Communication and Computer, 2013, 10(1):72-81.
- [15] Ouchani S, Mohamed O A, Debbabi M. A property-based abstraction framework for SysML activity diagrams[J]. Knowledge-based Systems, 2014, 56(1):328-343.
- [16] Kaliappan P S, Vogel M, Koenig H. Component-oriented verification for protocol designs using the SPIN model checker[J]. International Journal, 2014, 1(1):37-49.
- [17] keypoint. International Journal of Computer Vision, 2004, 60(2):91-110.
- [18] Fischler M, Bolles R. Random sample consensus a paradigm for model fitting with application to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6):381-395.
- [19] Heymann S, Muller K, Smolic A. SIFT implementation and optimization for general-purpose GPU[C]//Proceedings of the 15th international conference in Central Europe on computer graphics, visualization and computer vision. [s. l.]: [s. n.], 2007:317-322.
- [20] 闫钧华, 杭谊青, 许俊峰, 等. 基于 CUDA 的高分辨率数字视频图像配准快速实现[J]. 仪器仪表学报, 2014, 35(2):380-386.
- [21] 张 舒, 褚艳利. GPU 高性能运算之 CUDA[M]. 北京: 中国水利水电出版社, 2009.
- [22] Jang K H, Jung S K, Lee M. Constructing cylindrical panoramic image using equidistant matching[J]. Electronics Letters, 1999, 35(20):1715-1716.
- [23] Shum H Y, Szeliski R. Construction and refinement of panoramic mosaics with global and local alignment[C]//Proc of sixth international conference on computer vision. Bombay: IEEE, 1998:953-956.
- [24] Muja M, Lowe D G. Fast approximate nearest neighbors with automatic algorithm configuration[C]//Proc of international conference on computer vision theory and applications. [s. l.]: [s. n.], 2009:331-340.
- [25] Lowe D G. Object recognition from local scale-invariant features[C]//Proc of the international conference on computer vision. [s. l.]: [s. n.], 1999:1150-1157.
- [26] Duplaquet Marie-Lise. Building large image mosaics with invisible seam lines[J]. Proc of SPIE, 1998, 3387:369-377.
- [27] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer[C]//Proceedings of the 28th annual conference on computer graphics and interactive techniques. [s. l.]: ACM, 2001:341-346.
- [28] Kuglin C, Hines D. The phase correlation image alignment method[C]//Proc of international conf on cybernetics and society. New York: IEEE, 1975:163-165.
- [29] 彭 勃, 何 宾. FPGA 在视频拼接中的应用与实现[J]. 计算机工程与设计, 2013, 34(5):1635-1639.

(上接第 26 页)

基于Spin的SysML时序图与活动图一致性检测

作者：[胡良文](#)，[马金晶](#)，[孙博](#)，[HU Liang-wen](#)，[MA Jin-jing](#)，[SUN Bo](#)

作者单位：[南京航空航天大学](#) [计算机科学与技术学院](#), [江苏](#) [南京](#), 210016

刊名：[计算机技术与发展](#) 

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015 (9)

引用本文格式：[胡良文](#). [马金晶](#). [孙博](#). [HU Liang-wen](#). [MA Jin-jing](#). [SUN Bo](#) [基于Spin的SysML时序图与活动图一致性检测](#) [期刊论文] - [计算机技术与发展](#) 2015 (9)