

基于 Petri 网的 SysML 活动图的分析与验证

王松锋 熊选东 付建丹 张亮忠

(解放军信息工程大学电子技术学院 郑州 450004)

摘 要 系统建模语言 (Systems Modeling Language, SysML) 是目前国际上系统工程领域最新的标准建模语言, 它包括语义和表示法两部分, 缺乏分析和验证的手段。为了弥补这一不足, 研究了 SysML 活动图到 Petri 网的转换方法, 主要定义了将 SysML 活动图转换为相应 Petri 网可执行模型的 6 种转换规则。应用这些规则可以将活动图转换为 Petri 网模型, 进而对其进行化简、分析和验证, 同时可检测 SysML 的行为规范与并发相关的性质, 如死锁、有界性等。采用列举法和模拟法验证了所建模型的一致性。证书申请活动图的实例表明, 该方法是可行的。

关键词 系统建模语言, Petri 网, 活动图, 建模

中图分类号 TP301 **文献标识码** A

Analyzing and Verifying of SysML Activity Diagram Based on Petri Net

WANG Song-feng XIONG Xuan-dong FU Jian-dan ZHANG Liang-zhong

(Electronic Technology Institute, Information Engineering University, Zhengzhou 450004, China)

Abstract Systems modeling language is the latest international standard systems engineering modeling language. It contains two parts of semantics and notations which is lack of analysis and verification tools. For solving the questions, this paper presented a method for conversing the SysML activity diagram to the Petri net, mainly defined six transformation rules for converting SysML activity diagram to a Petri net executable model. Using these transformation rules, SysML activity can be transformed into Petri net, and thus realizing its simplification, analysis and verification. Besides of this, it can detect concurrency-related code of conduct nature, such as deadlocks, boundness, etc. And the consistency of the model was verified by the list method and simulate method. Finally, an example was used to verify the feasibility of the method.

Keywords SysML, Petri net, Activity diagram, Modeling

系统建模语言 (Systems Modeling Language, SysML) 是 2006 年由对象管理组织 (Object Management Group, OMG) 和国际系统工程学会 (International Council On Systems Engineering, INCOSE) 为系统体系结构设计提出的多用途建模语言, 是统一建模语言 UML 在系统工程应用领域的延伸和扩展, 可用于由软硬件、数据和人综合而成的复杂系统的集成体系结构说明、分析和设计^[1]。

SysML 针对系统工程领域中系统设计与建模的特点, 提供了可视化、图形化的系统建模支持。自 SysML 提出以来, 其已得到了 IBM、I-Logi、Sparx 等众多公司的支持, 广泛应用于复杂系统建模。然而, SysML 和 UML 一样, 为了保持描述的清晰易懂, 采用了半形式化的描述方式, 使用自然语言描述约束和详细语义, 力求实现形式严格和易于理解之间的平衡^[1]。这意味着, SysML 本身缺乏分析和验证的手段, 只能进行静态建模, 不能进行动态仿真。也就是说, 对于 SysML 描述的图形, 目前尚缺乏严密有效的验证和分析方法, 同时也难以在模型实现之前进行仿真。针对这一问题, 可采取的办法是将活动图转化为形式化描述的模型, 然后对其进行分析、

验证。

本文给出了 SysML 活动图的 Petri 建模方法, 即通过将 SysML 活动图转换到 Petri 网, 在非形式化的图形表示和形式化图形定义之间建立映射关系。主要目的就是采用 SysML 建模的同时, 将其转换为 Petri 网并利用 Petri 网的分析技术对 SysML 活动图进行验证, 以避免形式化方法对数学基础要求较高、直接使用起来困难的局限性, 并弥补 SysML 缺少模型分析、无法进行分析和仿真等缺点。最后采用列举法和模拟法验证了转换模型的一致性。

1 国内外研究现状

由于 SysML 还处于发展阶段, 针对上述问题的研究还比较少, 因此本文主要借鉴 UML 模型的形式化。在 UML 模型的形式化方面, 国内外专家学者进行了大量研究, 概括起来主要有以下几种方法: (1) 基于模型方法 Z 的转换^[2,3]: 将状态机图中各种状态模式转换为 Z 语言的模式运算来表达; (2) 基于逻辑方法的转换^[4,5]: 将类图以及带有时间扩展的状态图转换到通用高阶逻辑形式规范语言 PVS 或者使用实时动作

到稿日期: 2011-12-15 返修日期: 2012-03-20

王松锋 (1986—), 男, 硕士生, 主要研究方向为系统工程、Petri 网等, E-mail: 375574648@qq.com; 熊选东 (1965—), 男, 研究员, 主要研究方向为系统工程及计算机应用; 付建丹 (1986—), 男, 硕士生, 主要研究方向为系统工程; 张亮忠 (1985—), 男, 硕士生, 主要研究方向为系统工程。

逻辑 RAL 对状态机图进行形式化描述与精化, 进而进行验证; (3) 基于自动机的转换^[6,7]: 将状态机图的操作映射到一个特定形式的自动机, 使用基于自动机理论的模型检验方法来验证状态机图的线性时态逻辑性质, 或使用自动机建立序列图的形式化模型, 然后进行验证; (4) 基于验证工具语言的转换^[8]: 将 UML 协作图和状态图转换为模型检验器 SPIN 的输入语言 Promela 后, 使用模型检验器 SPIN 来验证系统设计模型是否满足某些关键性质需求; (5) 基于 Petri 网的转换^[9-12]: 将 UML 图映射为高级 Petri 网, 利用 Petri 网的理论知识对所得到的网模型进行分析和验证。

上述方法结合 UML 和形式化方法的优点, 将非形式化的 UML 图转换为具有精确语义定义的形式化规范, 在非形式化的图形表示与形式化定义之间建立映射关系。其主要特点是采用 UML 进行系统建模, 然后使用形式化方法对模型进行分析和验证, 这类方法既避免了形式化方法不易理解的局限性, 又克服了 UML 缺少模型分析、验证手段的不足。目前很多研究者对 UML 形式化时大多采用将建立的 UML 图转换为某种形式规范的方法。他们的工作为在 SysML 领域中进行相关研究奠定了基础。

本文在一定程度上借鉴了前人的工作, 选择 Petri 网^[13,14]作为 SysML 的活动图转换的规范语言。Petri 网是一种图形化、形式化的建模语言, 它以数学和图论知识为理论基础, 具有精确的语法和语义, 在协议工程、计算机网络、逻辑推理和工作流建模等领域得到了广泛应用。近年来, Petri 网在 UML 和 SysML 模型的形式化建模与分析方面开始得到较多研究, 涵盖了一般 Petri 网、时间 Petri 网、面向对象 Petri 网和有色 Petri 网等多种类型。

文献^[15]给出了 UML 状态机图和协作图转化为有色 Petri 网的方法; 文献^[16]给出了基于 UML 的 CPN 模型的测试框架; 文献^[17]提出了基于 UML 和 Petri 网的层次建模方法, 该方法采用 Petri 网描述系统的动态行为, 经过分析优化后, 将其转化成状态图和序列图; 文献^[18]阐述了状态图到 Petri 网模型的转换规则; 文献^[19]探讨了一种基于面向对象 Petri 网与 UML 模型的系统层次建模方法, 其按照一定的映射算法将 Petri 网模型转化成 UML 模型的状态图和顺序图。

对于活动图的 Petri 网转化, 国内外不同学者进行了研究。UML 活动图到 Petri 网的转换主要有两种, 一种是非形式化的, 一种是形式化的。英国的 GLimore 和 Kloul 研究了将 UML 2 活动图模型转化为性能评价进程代数 PEPA, 但是没有解释转化过程, 其看起来是非形式化的^[20]。西班牙的 Lopez-Grao 和 Campos J 把活动图转换为一种良好定义的形式化的标记通用随机 Petri 网 LGSPNs, 该方法利于性能分析, 但是最终的网是活动图的简化^[21]。LaQuSo 把活动图转化为简单 Petri 网用于安全性分析^[22]。这些研究为应用 Petri 网对 SysML 构建的模型进行分析和验证提供了重要基础。

2 SysML 活动图到 Petri 网的转化规则

活动建模强调活动的输入、输出、顺序和条件。活动图通过动作(action)的一系列控制序列, 描述输入到输出的转换。SysML 活动图对 UML 活动图进行了扩展, 包括把控制作为数据、可以表示连续的系统 and 加入概率等。在 UML 活动图中, 控制只能使动作开始, SysML 可以使正在执行的动作结束。SysML 活动图主要包括活动、变迁、分叉点、连接点、判断等元素。

SysML 活动图的 Petri 网建模之关键是 SysML 活动图到 Petri 网的转换过程, 只有完成这个转换过程, 得到活动图的 Petri 网模型后才能对模型进行分析和验证。本节首先解释了 SysML 活动图中的一些常用元素如活动、变迁、分叉点、连接点、判断等的含义, 然后给出了其转换为等价 Petri 网的转换规则, 重点是在 UML 活动图的元素与 Petri 网之间建立一个明确的映射。

规则 1(活动的转换) 活动是活动图的基本元素之一。活动图中的一个活动转化为一个基本的 Petri 网模型, 如图 1 所示: 库所 In_A 表示活动的开始, 变迁 t_A 表示活动的发生, 库所 Out_A 表示活动的结束。

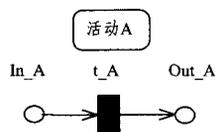


图 1 活动的转换

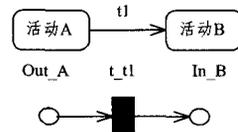


图 2 变迁的转换

规则 3(起始点和结束点的转换) 起始点映射为一个基本的 Petri 网模型, 如图 3 所示: 库所 $IniSts_A$ 表示活动的起始, 该库所里用一个托肯表示初始标识, 也表示 SysML 活动图的实际状态; 变迁 t_in_A 表示活动开始, In_A 表示活动 A 即将发生。结束点映射为一个基本的 Petri 网模型, 如图 4 所示: 库所 Out_A 表示活动 A 的结束, 变迁 t_end_A 表示活动结束动作本身, 库所 Out_A 表示活动的结束。

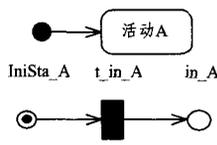


图 3 起始点的转换

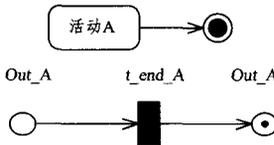


图 4 结束点的转换

规则 4(分叉点转换) 分叉点 folk 映射为一个基本的 Petri 网模型, 如图 5 所示: 库所 Out_A 表示活动 A 的结束, 变迁 $t_folk_B_C$ 表示活动 B 和 C 的分叉, 库所 brc_B 和变迁 brc_C 分别表示分叉活动, 变迁 t_in_B 和 t_in_C 分别表示即将进入活动 B 和活动 C, 库所 In_B 和 In_C 表示活动 B 和活动 C。

规则 5(连接点转换) 连接点(join)folk 映射为一个基本的 Petri 网模型, 如图 6 所示: 库所 Out_A 和 Out_B 分别表示活动 A 和活动 B 的结束, 变迁 $t_J_B_C$ 表示活动 A 和 B 的连接, 库所 J_A_B 表示活动 A 和 B, 变迁 t_in_C 表示即将进入活动 C, 库所 In_C 表示活动 C。

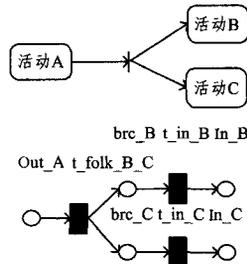


图 5 分叉点的转换

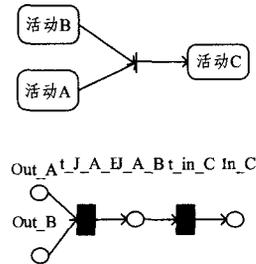


图 6 连接点的转换

规则 6(判断的转换) 判断(decision)映射为一个基本的 Petri 网模型,如图 7 所示:库所 Out_A 表示活动 A 的结束,变迁 t_in_Deci 表示判断的触发,库所 $Deci_B_C$ 表示判断,变迁 t_in_B 和 t_in_C 分别表示即将进入活动 B 和活动 C,库所 In_B 和 In_C 表示活动 B 和活动 C。

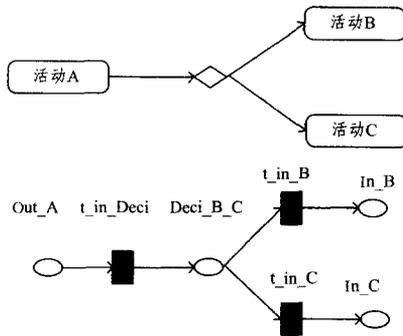


图 7 判断的转换

3 Petri 网模型化简规则

由第 2 节的转换规则可以看出,从活动图转换为的 Petri 网模型节点过多,包含一些对于性质分析多余的库所和变迁;此外,若对大型、复杂系统建模,转换后的 Petri 网模型往往存在模型状态空间爆炸问题,即模型的状态空间随实际系统规模的增大而呈指数增长,给网的分析带来了困难,因此需要对转换后的 Petri 网模型进行化简。本文化简 Petri 网的方法和技巧来源于文献[13],具体技巧如图 8 所示。

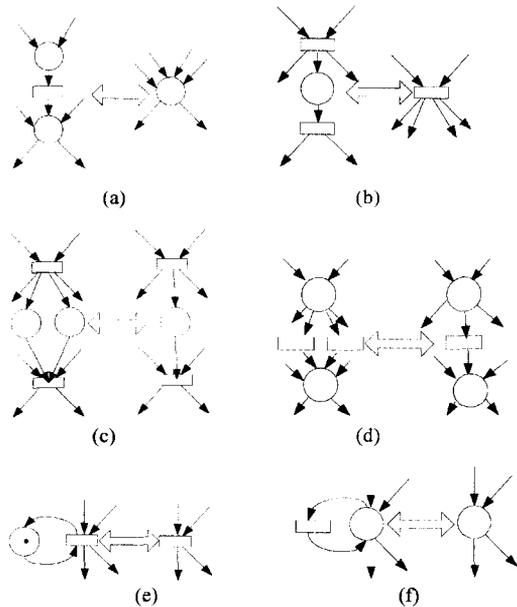


图 8 6 种保持活性、有界性和安全性的化简技巧

所谓 Petri 网的化简,是指在某种性质不变的基础上,采用等效变换,以达到缩小状态空间、简化系统分析的目的。经证明,由这 6 种化简技巧化简的 Petri 网可保持原来的活性、安全性以及有界性,其详细介绍及证明可参阅文献[13]。

4 动态行为的可执行验证过程

将活动图转换为 Petri 网并化简后,就可以利用 Petri 网的性质对动态行为实施可执行验证。动态行为可执行验证的整个过程可通过以下流程实现(见图 9)。

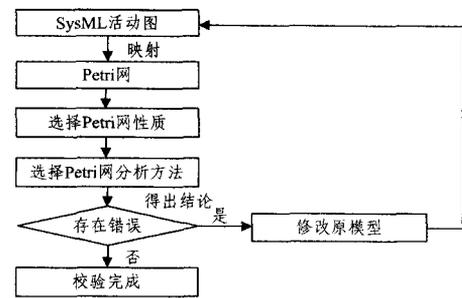


图 9 动态行为的可执行验证过程

在校验动态行为时,首先根据校验的目的选择模型所需满足的性质,只有当模型表现出一组所需要的性质时,才认为模型是正确的。一旦将这些性质放到模拟的系统的上下文中来解释,就可以使校验人员识别出该模型具有该应用领域特有的哪些功能,以及还缺少哪些功能。

其次,选择 Petri 网的分析方法验证模型是否满足这些性质。Petri 网支持多种不同精度的验证技术,如模拟法、结构分析法、列举法等。模拟法通过在所建立的网系统模型上授权变迁发生,模拟 Petri 网的运行。它能够揭示缺陷,并为最终用户提供重要的信息反馈,因此它支持对定义的验证。结构分析用于研究网系统的行为特征和结构之间的关系,依赖于状态方程,与 Petri 网的拓扑有关,主要分析与起始标识无关的特性。列举法基于可达图验证,该方法用以描述 Petri 网的可达标识集,它既与 Petri 网的结构有关,也与 Petri 网的初始标识有关。通过分析 Petri 网的可达图,可了解 Petri 网的许多重要性质,如有界性、安全性、守恒性、可达性、覆盖性、死锁和活性等。

经过校验之后,如果存在问题,则通过回溯原模型找出其问题所在并进行修改;然后对修改之后的活动图重新建立 Petri 网可执行模型,对其实施校验,直到不再存在问题为止。

5 案例:证书申请活动图到 Petri 网的转换与验证

5.1 证书申请活动图

公钥基础设施(PKI)为大规模、分布式开放网络环境下的信息安全问题提供了行之有效的解决办法,PKI 的核心技术就是围绕数字证书的申请、颁发、使用、更新和撤销等整个生命周期展开的^[23]。本文以证书申请为例,说明如何使用 SysML 和 Petri 网建模。

证书申请活动图如图 10 所示。终端实体提出证书申请,进行申请证书的活动。当终端实体完成这一活动后,将申请信息交给注册机构。注册机构对信息进行审核,完成信息审核活动;如果审核通过,则将申请信息转交给认证中心进行后继的活动;若不通过,则由注册机构将失败的结果返回给终端实体,同时证书申请过程结束。当认证机构得到经过审核的申请信息后,进行证书签发活动。如果签发成功,便执行证书发布活动;若不成功,返回失败的消息。可以看到,证书发布活动需要认证机构与资料库共同参与完成。在这一个用例中,认证机构负责证书的发布,也就是将证书发送到资料库中,而资料库则存储接收到的证书。完成证书发布活动后,认证机构将签发成功的证书发往注册机构进行证书制作活动。在注册机构完成证书制作活动后,这一个证书申请的用例得到完成,返回结果到终端实体。

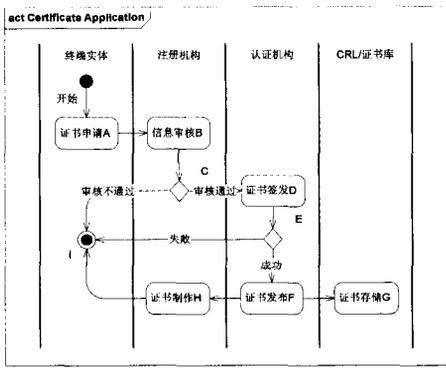


图 10 证书申请活动图

5.2 证书申请活动图向 Petri 网模型的转化与化简

根据转换规则 1—规则 6,把证书申请活动图转化为基本的 Petri 网模型,如图 11 所示。

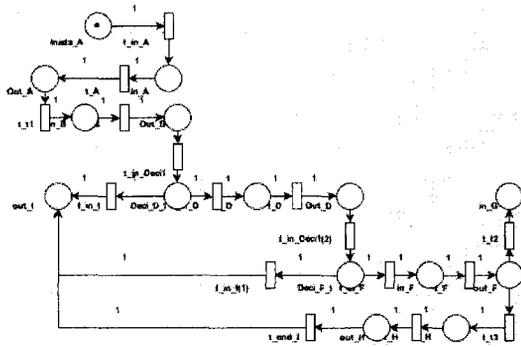


图 11 证书申请活动图的 Petri 网模型

根据第 3 节提到的保持活性、有界性、安全性的化简规则,将上述 Petri 网模型化简:

(1) 根据图 8 中的化简技巧(a),将从库所 $Inista_A$ 经变迁 t_in_A 、 t_A 、 t_l1 、 t_B 到达库所 Out_B 这一系列库所变迁化简为一个库所 $P0$ 。

(2) 根据图 8 中的化简技巧(b),将从变迁 In_D 经库所 t_D 、 Out_D 到达变迁 $t_in_Deci1(2)$ 这一系列库所变迁化简为一个新的变迁 $T2$ 。

对于本例而言,在经上述两步化简后,只需重复利用技巧(a)和(b),调整库所变迁命名,就可得到最终化简图,如图 12 所示。

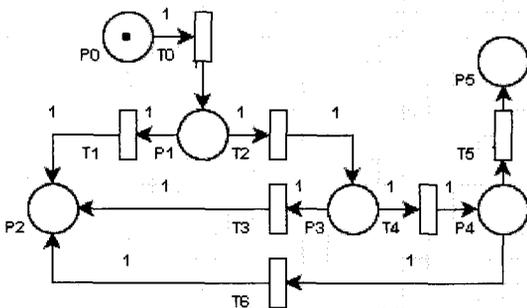


图 12 化简后的 Petri 网模型

由上述化简图可以清晰地看出,原来含有 15 个库所、16 个变迁的 Petri 网模型,经化简后,最终只含有 6 个库所、7 个变迁;而且原有模型中存在的并发、选择在化简后的 Petri 网中仍然刻画得非常清晰。

其中,库所 $P0$ 表示证书申请信息,变迁 $T0$ 发送证书申

请和审核请求,库所 $P1$ 表示待审核的证书信息;变迁 $T2$ 表示审核通过,发送证书签发请求,变迁 $T1$ 表示审核不通过; $P3$ 表示待签发的证书; $T4$ 表示签发成功,进行证书发布; $T3$ 表示签发失败; $P4$ 表示待发布的证书; $T5$ 表示证书存储; $P5$ 表示存储的证书; $T6$ 表示证书制作。

5.3 Petri 网模型分析

本例研究的对象是数字证书申请,主要考虑模型可达的状态、是否存在死锁、模型能否结束、系统是否有界等问题,因此,这里主要分析 Petri 网的可达性、无死锁性、活性和有界性。

本例中,Petri 网的初始状态已经确定,可利用模拟法和列举法来分析模型。

5.3.1 模拟法

模拟法也即从 Petri 网的初始标识开始,驱动变迁的连接发生和标识的不断变化,观察网系统的运行,分析所建模型是否完成了预期的功能。

观察前面所得的化简后的 Petri 网模型,假设 $M_i (i=0, 1, \dots, 5)$ 为标识, M_0 为其初始标识。系统的可达标识集为 $[M_0] = \{M_0, M_1, \dots, M_5\}$ 。由图 12 可知,其初始标识为 $(1, 0, 0, 0, 0, 0)$,根据变迁发生规则,可发生的变迁为 $T0$,即可对证书申请进行审核,此时标识为 $(0, 1, 0, 0, 0, 0)$ 。在此标识下有两个变迁有发生权: $T1$ 或者 $T2$,两者只能发生一个。当 $T1$ 发生时,即审核失败,此时系统标识为 $(0, 0, 1, 0, 0, 0)$,运行结束。当 $T2$ 有发生权时,即审核通过,网系统的运行行为: $T2 \rightarrow T4$,此时标识为 $(0, 0, 0, 0, 1, 0)$,在此标识下有两个变迁有发生权: $T5$ 或者 $T6$,两者只能发生一个。当 $T6$ 有发生权时,即证书签发失败,此时系统标识为 $(0, 0, 1, 0, 0, 0)$,运行结束。

检查活动图,发现活动图在语义上都能表达用户的需求;同时,模型表达语义恰当,表达能力充分,能够按照要求实现证书申请,各对象都完成了期望的功能,对象之间也能进行正确交互。

为了实现可视化,以利于判断结果的正确性,可通过 Pipe3.0 对 Petri 网进行模拟。它能为模型验证人员提供动画演示过程,并给出模拟结果。

图 12 即为证书申请模拟运行的界面,通过动画演示,可以确定模型能够按照要求完成证书申请活动。

5.3.2 列举法

由上可知,模拟法在理解所建立的模型方面或在修改模拟过程中显现的问题方面具有极大的帮助作用。但模拟法不能保证模型不会出现期望之外的行为,即不能体现 Petri 网的相关性质。上述缺陷可以通过列举法来弥补。列举法基于建造可达图(RG)。可达图是分析 Petri 网的有效工具,可用于分析网系统的各种动态性质。

计算可达图的具体算法如下。

算法 1 计算可达图

输入:网系统 $S=(N, M_0)$

输出:对于有界网系统,有向图 $RG(S)=(V, E)$

初始化有向图 $RG(S)=(m_0, \emptyset)$; m_0 没有做标记

while 在集合 V 中还存在没有标记的节点 do

从集合 V 中任意选择一个没有标记的节点 $m \in V$, 并且标记它

for 每一个在标识 m 下可以发生的变迁 t

计算 m' 使得 $m \xrightarrow{t} m'$;

if 存在标识 $m'' \in V$ 使得 $m'' \xrightarrow{\sigma} m'$ 并且 m'' 大于 m'
then 算法失败, 终止, 输出“系统是无界的”; (检测完 S 的无界条件)
if 存在标识 $m'' \in V$ 使得 $m'' = m'$
then $V := V \cup \{M'\}$; (其中 m' 没有做标记)
 $E := E \cup \langle (m, t, m') \rangle$
算法成功输出可达图 $RG(S)$

在此, 计算证书申请活动图的 Petri 网模型的可达图, 如图 13 所示。

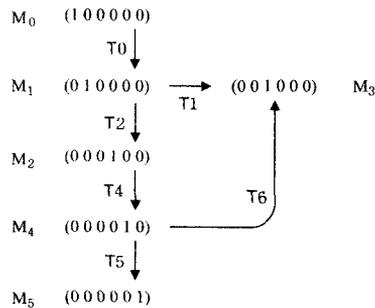


图 13 可达标识图

可达图给出了由初始标识开始的所有可达的状态, 因此可以保证其可达性。在图 13 中, 每个可达状态代表了证书申请时各对象所处的状态。通过回溯分析可知, 各对象的每一个功能也都能在可达图中找到其对象的状态。因此, 可以说模型能够正确体现各对象的功能。从图中也可以看出, 任意库所中的托肯数都是有限的, 因而是有界的, 同时也是安全的。有界性说明证书申请系统不会产生溢出, 安全性是有界性的一种特殊情况, 说明每次只能进行证书申请过程中的一个步骤。对于图中的任意变迁 t , 对任何可达标识 $M \in [M_0]$, 总有从 M 可以达到的标识 $M' \in [M_0]$, 使得 $M[t]$, 因此可以说该模型是活的, 那么就肯定不存在死锁。这可以保证概念模型中各对象的交互不存在死锁。随着模型的运行, 模型最终可以到达一个稳定的状态。

综合以上分析, 通过模拟法可知, 模型正确体现了各对象的功能以及对象之间的交互关系, 能够按照要求实现正确传输, 而且各图模型之间也保持了动态上的一致性; 通过列举法可知模型中没有出现不需要的对象或者不希望出现的状态, 保证了状态的可控性, 同时每个对象在最终的状态时仍拥有一个托肯, 为下一次传输作好准备。

通过实施本文所建立的校验过程, 保证了证书申请活动图模型的完整性、正确性、一致性。由此也可证明本文所建立的校验过程是可行、合理的。

结束语 系统建模语言 SysML 可用于由软硬件、数据和人综合而成的复杂系统的集成体系结构说明、分析和设计, 具有可视化的特性, 且容易理解; Petri 网是一种图形化、数学化的形式化建模语言, 便于对模型进行仿真与分析。本文给出了 SysML 活动图的 Petri 网建模, 其可以将 SysML 活动图转化为 Petri 网模型, 并利用 Petri 网的分析技术对转化后的模型进行仿真与分析, 同时采用列举法和模拟法验证了转换模型的一致性。

参考文献

[1] OMG. SysML-v1. 2-PDF. pdf[EB/OL]. <http://www.omg.org/spec/SysML/1.2/>, 2010-06-01

[2] 李桂, 苏一丹. UML 状态图的形式化[J]. 广西大学学报, 2003, 28(4): 318-321

[3] 祝义, 张广泉. 基于 UML 和 Z 的软件体系结构求精方法及其应用[D]. 苏州: 苏州大学, 2006

[4] 尤晋元. 基于 UML 与 PVS 的严格建模技术研究——嵌入式实时系统的建模[D]. 上海: 上海交通大学, 2003

[5] 罗蜜, 张为群. 结合形式化方法的 UML 系统开发[J]. 西安师范大学学报, 2003, 28(2): 203-208

[6] 董威, 王戟, 齐志昌. UML Statecharts 的模型检验方法[J]. 软件学报, 2003, 14(4): 750-756

[7] 张琛, 段振华, 田聪. 基于事件确定有限自动机的 UML2.0 序列图描述与验证[J]. 软件学报, 2011, 22(11): 2625-2638

[8] 单卓为. 基于 SPIN 的 UML 模型验证技术的研究[D]. 西安: 西北大学, 2008

[9] Kerkouche E, Chaoui A, Bourennane E-B, et al. A UML and Colored Petri Nets Integrated Modeling and Analysis Approach using Graph Transformation[J]. Journal of Object Technology, 2010, 9(4): 25-43

[10] Kerkouche E, Chaoui A, et al. Modeling and verification of Dynamic behaviour in UML models, a graph transformation based approach[C]//Proceeding of SEDE. Las Vegas, Nevada, USA, 2009

[11] Spiteri Staines T. Intuitive Mapping of UML 2 Activity Diagrams into Fundamental Modeling Concept Petri Net Diagrams and Colored Petri Nets[C]//Proc. of the 15th IEEE International Symposium and Workshop on Engineering of Computer Based Systems. Belfast, Ireland, ECBS, 2008

[12] Shinkawa Y. Inter-Model Consistency in UML based on CPN Formalism[C]//Proc. of APSEC' 06. Bangalore India, 2006

[13] Murata T. Petri Nets; Properties, Analysis and Applications [J]. Proceedings of the IEEE, 1989, 77(44): 541-580

[14] Jensen K, et al. Colored Petri Nets and CPN Tools for modeling and validation of concurrent systems[J]. Int J Softw Tools Technol Transfer, 2007, 9: 213-254

[15] Kerkouche E, et al. A UML and Colored Petri Nets Integrated Modeling and Analysis Approach using Graph Transformation [J]. Journal of Object Technology, 2010, 9(4): 25-43

[16] 刘烁, 陈俊杰. 基于 UML 的 CPN 模型在软件测试中的应用 [J]. 计算机工程, 2008, 34(3): 119-121

[17] 江金龙. UML 和 Petri 网建模方法及其在 C4ISR 系统中的应用 [D]. 南京: 南京理工大学, 2006

[18] 杨喜刚, 等. 基于 UML 状态图和 Petri 网的冷却水实时控制系统建模与分析[J]. 苏州大学学报, 2009, 29(6): 7-11

[19] 周振喜, 等. 基于 OPN 和 UML 的导航装备维修建模方法研究 [J]. 航空计算技术, 2010, 40(1): 85-89

[20] Canevet C, Gilmore S, Hilliston J, et al. Analyzing UML Activity Diagrams in the Software Engineering Performance Process[C]//Proc. of WOSP'04. Redwood City CA, 2004

[21] Lopez-Grao J P, Campos J. From UML Activity Diagrams to Stochastic Petri Nets [J]. ACM Software Engineering Notes, 2004, 29(336): 25-36

[22] LaQuSo[EB/OL]. <http://www.Laquso.com>

[23] 荆继武, 林环镛, 冯登国. PKI 技术[M]. 北京: 科学出版社, 2008: 103-132