基于模型检测的策略冲突检测方法

吴 丹[1,2]，危胜军[2,3]

(1. 北京理工大学机电学院 北京 海淀区 100081; 2. 北京理工大学软件安全工程技术北京市重点实验室 北京 海淀区 100081;
3. 北京理工大学软件学院 北京 海淀区 100081)

【摘要】提出一种基于模型校验的策略冲突检测新方法。首先通过形式化描述语言进行系统建模，采用时态逻辑表征策略冲突的系统属性，然后利用NuSMV模型检测器验证属性的可满足性，并根据模型检测器产生的反例轨迹追溯策略冲突点。该方法可提高策略冲突检测的效率。

关 键 词 反例; 模型检测; NuSMV; 策略冲突
中图分类号 TP399 文献标志码 A doi:10.3969/j.issn.1001-0548.2013.05.019

Policy Conflict Detection Method Based on Model Checking

WU Dan[1,2] and WEI Sheng-jun[2,3]

(1. School of Mechatronical Engineering, Beijing Institute of Technology Haidian Beijing 100081;
2. Beijing Key Laboratory of Software Security Engineering Technology, Beijing Institute of Technology Haidian Beijing 10008;
3. School of Software, Beijing Institute of Technology Haidian Beijing 100081)

Abstract A new policy conflict detection method is proposed based on model checking. In this method, the model of system is specified with formal description language, the properties of system depending on different types of policy conflicts is represented with temporal logic, and the violations of properties is detected by using NuSMV model checker, which can provide the counterexample and trace it back to the policy conflict point. The result shows that the method can improve the efficiency of policy conflict detection.

Key words counterexample; model checking; NuSMV; policy conflict

策略是网络安全管理系统的重要组成部分，其一致性检测是实现系统协同首先需要解决的问题。有关研究大多集中在对概念、原因、分类等方面的定性分析上，需要大量人工参与，形式化证明存在困难，制约了策略安全管理的自动化、智能化进程[1-2]。

文献[3-4]认为在两条或多条策略的目标不能同时满足时将发生策略冲突，基于主体、目标和行为的重叠定义了策略冲突。

基于策略描述语言，文献[5]等研究了分布式环境下防火墙策略冲突检测；文献[6]研究了IPSEC的策略冲突检测，该策略冲突检测方法针对具体安全设备的策略，如防火墙策略、IPsec、VPN策略和路由策略。由于安全设备策略的描述语言具有规范格式，冲突检测算法较易理解；但难以从整体上把握安全需求，方法不具有可重复性。

为避免上述缺点，近年来的研究一般采用基于信息模型的策略冲突检测方法，从更高的抽象层次来表示更复杂的实体[7]。文献[8]用模型来描述系统的结构，以形式化逻辑符号描述事件演算(Event Calculus)、以形式化语言表征策略冲突，实现策略冲突分析；文献[9]基于时间逻辑对策略的形式化规范进行描述，建立了基于有限状态自动机的Ponder语言策略建模方法；文献[1-2]等将策略冲突检测转化为有向图的连通性问题，通过分析邻接矩阵、可达性矩阵和路径矩阵间的运算结果实现了冲突检测。上述方法由于涉及大量逻辑公式和矩阵的推导证明，理论性强但在实际应用中不易实现[10]。本文提出一种基于模型检测的策略冲突检测方法，旨在利用自动化工具减少逻辑推导过程及形式化证明的人工参与程度，提高冲突检测效率。

收稿日期：2011 – 12 – 31；修回日期：2012 – 06 – 19
基金项目：国家863项目(2009AA01Z433)
作者简介：吴丹(1983 –)，男，博士，主要从事多级安全策略与网络安全方面的研究.

1 模型检测

模型检测是一种形式化分析验证方法，其思想是基于状态空间搜索，验证给定的系统模型与特定性质[10](时态逻辑公式所表示)的满足关系，采用时态逻辑描述系统属性、Kripke结构表示系统状态空间，实现系统属性与状态空间关系的自动穷举证明。

为了验证系统-性质间的满足关系，须完成3个步骤：

1) 采用模型检测器的描述语言对系统进行建模，得到模型M。

2) 使用时态逻辑描述系统的关键属性，产生时态逻辑公式Φ。

3) 以M和Φ为输入，运行模型检测器。模型检测器自动证明系统属性是否满足状态空间，如果不满足，则会生成系统失败行为轨迹。

图1为模型检测过程。模型检测是对问题$M \models \Phi$成立与否的判断过程，其中Φ是基于时态逻辑描述的待验证系统关键属性(系统性质)，M为待验证系统的适当模型，$\models$为满足关系。

由于不可再现或不为情景测试所覆盖[10]，并发系统中的并发错误是测试方法(运行若干重要情景模拟的活动)最难发现的一种错误类型，而策略冲突正是一种典型的并发错误。模型检测利用相关自动化工具实现策略冲突检测，适用于发现这种并发错误，提供系统失败行为轨迹，并可通过“逆向追踪”找到策略冲突点，同时采用模型检测思想，可有效减少复杂的逻辑公式手工推导和形式化证明过程。

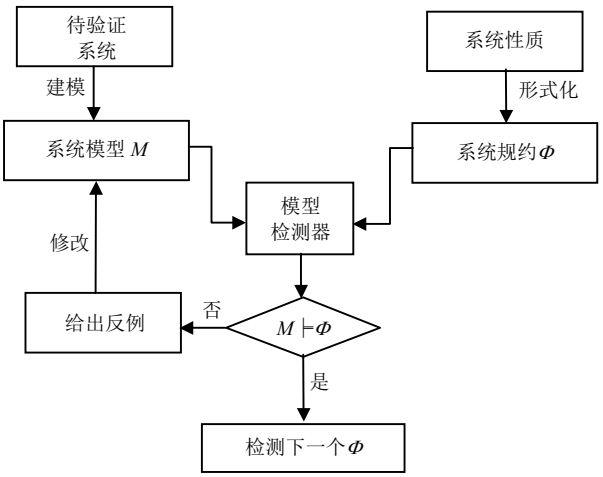

图1 模型检测过程

2 基于模型检测的策略冲突检测方法

基于模型检测的策略冲突检测，通过系统模型M及系统关键属性Φ，可将系统描述和系统行为策略分离。一些系统属性的检测，如活性、安全性、原子性等可由模型检测器自动完成。

2.1 系统建模

UML(unified modeling language)已成为标准建模语言，可描述系统的静态拓扑结构和动态行为。但UML是一种元模型，只有静态语义，没有形式化的动态语义，不能对系统进行自动推理和证明，因此UML模型不能直接作为模型检测器的输入。在本文方法中，首先给出UML图符的操作语义，再构造系统状态迁移图，才能生成模型检测代码。

线性时态逻辑(LTL)和计算树逻辑(CTL)类系统可用迁移系统建模，迁移系统[10]涉及状态(静态结构)和迁移(动态结构)两种建模要素。

定义 1 迁移系统定义为：$M=(S,\rightarrow,L)$。其中，S为状态集合，$\rightarrow$为迁移关系，L：$S\rightarrow p(\text{Atoms})$为标记函数。

模型状态集S所对应的每个状态s的其原子命题集合为$L(s)$，用p(Atoms)表示Atoms的幂集；迁移关系$\rightarrow$表示系统如何从一个状态转向另一个状态，即S上的二元关系，使得对每一$s\in S$存在某个$s'\in S$，满足$s\rightarrow s'$。

基于定义1，可用有向图表示迁移系统M的所有信息，其中，状态表述为图的结点，包含了该状态为真的所有原子命题。对于一个只有3个状态(S_0，S_1和S_2)的系统，如果状态间的可能迁移为：$S_0\rightarrow S_1$，$S_2\rightarrow S_0$，$S_1\rightarrow S_0$，$S_1\rightarrow S_2$，则该迁移系统M如图2所示，图中，$L(S_0)=\{a\}$，$L(S_1)=\{b\}$，$L(S_2)=\{c\}$。

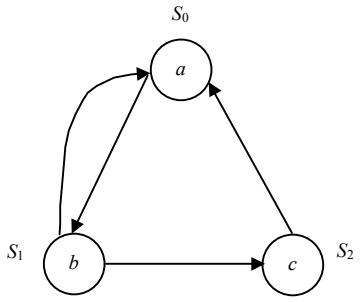

图2 迁移系统$M=(S,\rightarrow,L)$作为有向图的简明表示

2.2 线性时态逻辑LTL和计算树逻辑CTL

系统关键属性Φ可通过线性时态逻辑LTL或计算树逻辑CTL表示。LTL和CTL的语法定义如下[10]：

定义 2 线性时态逻辑LTL语法采用Backus Naur范式定义o ：

$$\Phi ::= \text{T} \mid \bot \mid p \mid (\neg \Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \rightarrow \Phi) \mid (\text{X}\Phi) \mid (\text{F}\Phi) \mid (\text{G}\Phi) \mid (\Phi \text{U} \Phi) \mid (\Phi \text{W} \Phi) \mid (\Phi \text{R} \Phi)$$

其中，p为原子集Atoms中的任意命题原子，时态连接词X、F、G、U、R、W分别定义为：

X表示“下一个状态”(next)；F表示“某未来状态”(Future)；G表示“所有未来状态”(Globally)；U、R、W分别表示“直到(Until)”，“释放(Release)”和“弱-直到”(Weak-until)。

定义 3　计算树逻辑CTL采用Backus Naur范式定义为：

$$\Phi ::= \bot \mid T \mid p \mid (\neg \Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \rightarrow \Phi) \mid AX\Phi \mid EX\Phi \mid AF\Phi \mid EF\Phi \mid AG\Phi \mid EG\Phi \mid A[\Phi U \Phi] \mid E[\Phi U \Phi]$$

其中，时态算子U、F、G、X的含义与CTL定义相同，量词A和E分别表示“对所有路径”和“存在一条路径”。

2.3　用时态逻辑进行关键属性描述

策略冲突分类如图3所示，策略模态冲突发生在行为同时被允许和禁止时，目标冲突则与策略同时执行时行为对系统的影响相关[3-4]。

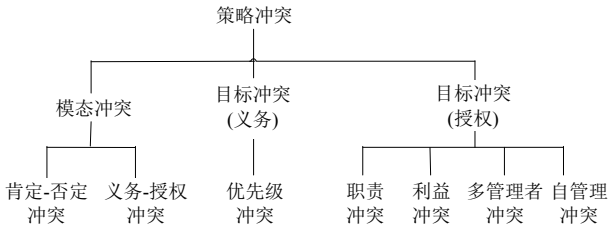

图3　策略冲突分类

尽管策略冲突类型的分类标准不尽相同，但是在形式化逻辑描述上都非常清晰。因此，根据线性时态逻辑LTL和计算树逻辑CTL的定义，可以用时态逻辑来对各种类型的策略冲突进行系统关键属性Φ的形式化描述。举例说明：

对于策略肯定-否定冲突，可用LTL算子G进行规约描述。如以下断言：

LTLSPEC G！((Subject.act =Allow) & (Subject.act =forbidden))可用来检测是否存在策略肯定-否定冲突，即不可能出现某主体的操作ALLOW成立，同时操作forbidden也成立的情况。

同样对于职责冲突，可用如下断言来判定：

LTLSPEC G！((Subject.role =teller) & (Subject.role =Auditor))

该断言表示，针对未来的系统状态，不会出现角色既是出纳，又是审计的情况。

2.4　模型检测器NuSMV

选取基于Kripke结构的NuSMV作为模型检测器。NuSMV以模型描述程序和时态逻辑规范文本作为输入。若规范成立，则输出为真，否则显示一个轨迹，该轨迹解释“规范为假”的原因。

如使用如下LTL公式：

LTLSPEC G!((p1 = true) & (p2 =true))

可检测策略间是否存在冲突，而且不可能出现策略p1成立，同时策略p2也成立的情况。

3　实例研究

以简化版“加锁数据库”系统为例[11]，该系统关注的是用户对数据库进行写操作时的数据安全，为防止多个用户同时对数据库进行写操作，系统采用了加锁和解锁的功能。为检测其是否存在策略冲突，使用图4描述系统的运行模式。

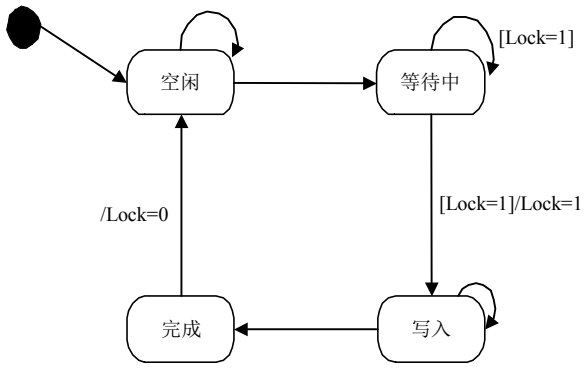

图4　加锁数据库状态迁移图

加锁数据库状态图由空闲状态开始，用户登入后，如果数据库处于加锁状态(Lock=1)，用户等待；直到解锁状态(Lock=0)，用户可以写入，完成操作。

与之相对应的NuSMV部分代码如图5所示。

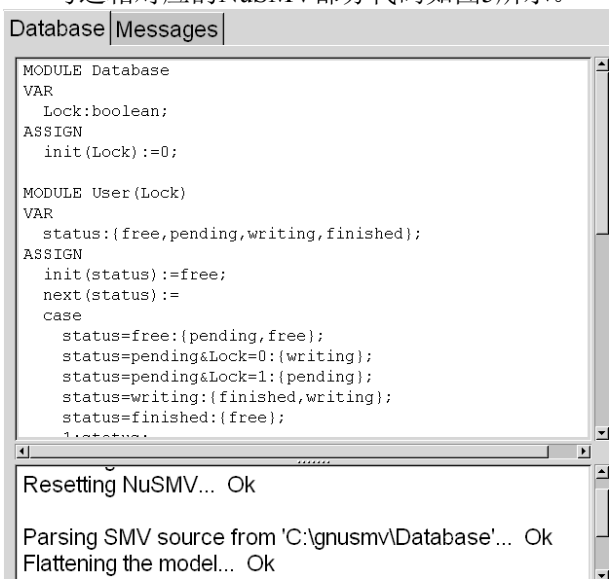

图5　加锁数据库部分NuSMV代码

采用NuSMV，实现如下两种类型冲突的检测：

1) 多用户同时对数据库进行写操作(目标冲

突)；2) 当一个用户登入数据库后总能够完成他的写入要求(模式冲突)。

两个待验证属性对应的LTL和CTL断言如下：

LTLSPEC !F(user1.status=writing&user2.status=writing)

AG (user1.status =pending->AF user1.status = writing)

4 实验结果及分析

将待验证属性输入NuSMV，检测前后结果如图6所示。

图6 属性验证前后对比

图中，LTL断言为“True”，表示该加锁数据库不存在多用户同时对数据库进行写操作的情况，即不存在“目标冲突”；CTL断言为“False”，表示该加锁数据库系统存在用户登入后不能完成数据库写入要求的情形，即存在策略冲突分类中的“模式冲突”。

通过模型检测器的“Show Trace”按钮，查看“False”的反例轨迹，可寻找策略冲突点，如图7所示。

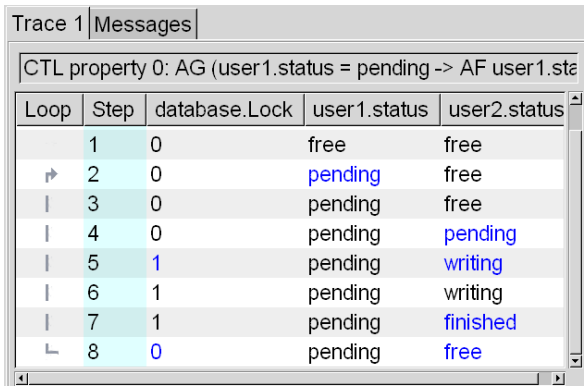

图7 反例轨迹

NuSMV的反例轨迹自动生成文本如图8所示。

图7和图8的结果表示，LOOP情况下user1一直维持等待(pending)状态，而user2一直维持空闲(free)状态，说明待检测系统存在设计缺陷；反例轨迹的描述文本进一步指示，由于没有考虑在user2进入加锁状态时，user1也可能申请写操作，造成user1一直无法进入写状态。

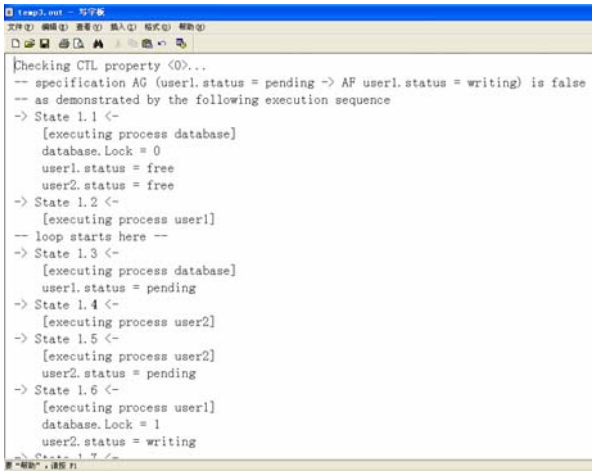

图8 反例轨迹的文本描述

综上所述，为了检测系统是否存在策略冲突，首先用状态迁移图抽象待检测系统的行为，然后用LTL和CTL逻辑公式表征不同类型策略冲突的关键属性，进而通过模型检测器快速判定系统是否存在策略冲突，同时利过反例轨迹能够快速找到策略冲突点，便于改进系统的设计缺陷。

5 结 束 语

在前人工作的基础上，基于模型检测的思想，本文提出了一种基于模型检测的策略冲突检测方法，并应用于策略冲突检测中。实验结果表明，该方法有效地降低了检测过程的人工参与程度，提高了冲突检测效率并可追溯策略冲突点。

参 考 文 献

[1] 姚键, 茅兵, 谢立. 一种基于有向图模型的安全策略冲突检测方法[J]. 计算机研究与发展, 2005, 42(7): 1108-1114.
YAO Jian, MAO Bing , XIE Li. A DAG-based security policy conflicts detection method[J]. Journal of Computer Research and Development, 2005, 42(7): 1108-1114.

[2] 李祥军, 孟洛明, 焦利. 网管系统策略冲突解决的结果中存在的问题及检测与解决方法[J]. 计算机研究与发展, 2006, 43(7): 1297-1303.
LI Xiang-jun, MENG Luo-ming, JIAO Li. Problems in results of policy conflict resolutions and detection and resolution methods in network management systems[J]. Journal of Compute Research and Development, 2006, 43(7): 1297-1303.

[3] LUPU E, SLOMAN M. Conflicts in policy-based distributed systems management[C]//IEEE Transactions on Software Engineering Management-Special Issue on Inconsistency Management. [S.l.]: IEEE, 1999.

[4] MOFFETT J, SLOMAN M. Policy conflict analysis in distributed system management[J]. Journal of Organizational Computing, 1994, 4(1): 1-22

(下转第768页)

[7] 朱清新. 离散和连续空间中的最优搜索理论[M]. 北京: 科学出版社, 2005.
ZHU Qing-xin. The optimal search theory in discrete and continuous spac[M]. Beijing: Science Press, 2005.
[8] NG See-kee, SEAH W K G. Game-theoretic approach for improving cooperation in wireless multihop networks[J]. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, 2010, 40(3): 559-574.
[9] MARBACH P, QIU Ying. Cooperation in wireless Ad hoc networks: a market-based approach[J]. IEEE ACM Transactions on Networking, 2005, 13(6): 1225-1338.
[10] URPI A, BONUCCELLI M, GIORDANO S. Modelling cooperation mmobile Ad hoc networks: a formal description of selfishness[C]//Proc WiOpt. France: Sophia-Antipolis, [s.n.], 2003: 3-5.
[11] MARTIN J O, ARIEL R. A course in game theory[M]. London: MIT Press, 1994.
[12] HEINZELMAN W B, CHANDRAKASAN A P, BALAKRISHNAN I I. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE Trans, 2002, 1(4): 660-670.

编 辑 漆 蓉

(上接第748页)

[5] AL-SHAER E, HAMED H, BOUTABA R, et al. Conflict classification and analysis of distributed firewall policies[J]. IEEE Journal on Selected Areas in Communications, JSAC, 2005, 23(10): 2069-2084.
[6] LIN C, XUE C, ZHITANG L. Analysis and classification of IPSec security policy conflicts[C]//Proc Japan-China Joint Workshop on Frontier of Computer Science and Technology. [S.l.]: [s.n.], 2006.
[7] 吴蓓. 安全策略转换关键技术研究[D]. 郑州: 解放军信息工程大学, 2010.
Wu Bei. Research on technology of security policy transformation[D]. Zhenzhou: PLA Information Engineering university, 2010.
[8] BANDARA A K, LUPU E C, RUSSO A. Using event calculus to formalize policy specification and analysis[C]// Proc of the 4th IEEE Workshop on Policies for Distributed Systems and Networks(Policy 2003). [S.l.]: IEEE, 2003: 1-14.
[9] BALIOSIAN J, SERRAT J. Finite state transducers for policy evaluation and conflict resolution[C]//Proc of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks. [S.l.]: IEEE, 2004.
[10] 哈斯·瑞安. 面向计算机科学的数理逻辑: 系统建模与推理[M]. 第2版. 何伟, 樊磊, 译. 北京: 机械工业出版社, 2007.
HUTH M, RYAN M. Logic in computer science: modelling and reasoning about systems[M]. 2nd ed. Translated by HE Wei, FAN Lei. Beijing: China Machine Press, 2007.
[11] 朱泽涛. 基于模型检测的UML形式化验证及其系统实现[D]. 广州: 中山大学, 2005.
ZHU Ze-tao. A formal verification system for UML models based on model checking[D]. Guangzhou: Sun Yat-Sen University, 2005.
[12] 程亮, 张阳. 基于UML和模型检测的安全模型验证方法[J]. 计算机学报, 2009, 32(4): 699-708.
CHENG Liang, ZHANG Yang. A verification method of security model based on UML and model checking[J]. Chinese Journal of Computers, 2009, 32(4): 699-708.
[13] XU De-sheng, XIA Ke-jian, ZHANG De-zheng, et al. Model checking the inconsistency and circularity in rule-based expert systems[J]. Compute and Information Science, 2009, 2(1): 12-17.

编 辑 税 红