基于扩展 Petri 网的除冰软件安全需求建模和验证

李 震 刘 斌 陆民燕 殷永峰

(北京航空航天大学 可靠性与系统工程学院, 北京 100191)

摘 要: Petri 网是形式化的系统建模方法,以严格的数学基础来保证系统的正确构建,但在支持复杂软件建模和自动化验证方面存在不足. 扩展了 Petri 网的形式语义,区别定义了状态型和数值型库所,区别定义了变迁的激发和抑制状态,引入了无前置、一元和组合判断规则,同时根据形式化定义将模型自动转换为检验程序实施安全性验证. 最后给出了以上方法在典型安全关键软件-除冰系统上的应用,过程和结果表明扩展的模型和方法增强了 Petri 网对复杂软件系统的建模能力,提高了软件的安全性,从模型到验证代码的自动转换解决了完善模型时人工修改相应代码的工作量和因此而引入人为错误的重复工作量的问题.

关 键 词: 软件安全性; 软件工程; Petri 网; 模型检验

中图分类号: TP 311.5; N 941.5

文献标识码: A **文 章 编 号**: 1001-5965(2012)01-0064-05

Modeling and verification of de-icing software safety requirement based on expanded Petri net

Li Zhen Liu Bin Lu Minyan YinYongfeng

(School of Reliability and Systems Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: As a formal method, Petri net can ensure the correct construction of system but lacks the ability of modeling complicated system and automatic verification. Firstly, the formal semantics of Petri net was expanded, the place of state and numeric were distinctively defined, the transition state of fired and inhibitory were distinctively defined, the non-precondition, unitary precondition and composite precondition were introduced, and the model was transformed into verification program to implement the safety verification according to formal definitions. And then, the effectiveness of this methodology was illustrated with an application in an aircraft de-icing system which is a typical safety-critical system. The process and result show that the expanded model and methodology enhance the ability of Petri net to model complicated software system, improve the software safety, and the automatic transform from model to codes settle the workload of rewrite corresponding code manually and the meaningless faults introduced by that when the model should be modified.

Key words: software safety; software engineering; Petri net; model checking

在安全关键系统中,软件所占的比例越来越高. 软件安全性是指软件运行不引起危险和灾难的能力[1]. 文献[2]指出形式化方法以严格的数学基础保证软件的正确构建,其中航空航天为重要的优先发展和示范领域.

Petri 网是描述动态系统很直观的形式化方法,目前已经存在着一些针对性的应用和扩展[3-6]. 软件可以更好地控制和处理复杂系统,所处理的变量不仅包括状态量,还有数值量. 但是对 Petri 网进行扩展以支持复杂软件系统建模和验

收稿日期: 2010-05-14; **网络出版时间**: 2012-01-06 15:11; **DOI**: CNKI:11-2625/V.20120106.1511.016

网络出版地址: www.cnki.net/kcms/detail/11.2625.V.20120106.1511.016.html

基金项目: "十一五"重点预研课题(51319070101); 航空科学基金资助项目(20095551025); 国家自然科学基金资助项目(70971056); 机载软件工程化研究专题(DY09Z11926)

作者简介: 李 震(1977-),男,江苏兴化人,博士生,现为江苏科技大学电子信息学院讲师,justlz@163.com.

证还缺乏相对深入的研究.

模型检验以其自动化和基于反例的错误分析得到工业界的青睐[7]. 在直接编写复杂软件系统验证程序之前,对系统进行图形化可以更好地展示系统的运行和逻辑结构,帮助分析和开发者深入理解系统. 同时,在系统开发早期的需求和设计存在相对频繁的变动,这使得验证程序也需要对应的修改,人工的频繁修改程序会导致大量无谓的错误、矛盾和调试工作量[8].

鉴于以上问题,本文在文献[9 - 11]的基础上扩展了 Petri 网的形式化定义,以增强其对复杂软件系统的建模和分析能力;建立了扩展 Petri 网和模型检验工具 NuSMV 程序语言的语义映射,从模型自动生成验证代码,对软件系统需求实施安全性验证.

1　扩展 Petri 网

Petri 网是一种适合并发、异步、分布式软件系统规格与分析的形式化方法[12]. 基本 Petri 网的形式定义为一个六元组:

$$PN=(P,T,F,\boldsymbol{K},W,\boldsymbol{M}_0)=(N,\boldsymbol{K},W,\boldsymbol{M}_0)$$

其中,(P,T,F)是一个网结构;$\boldsymbol{K}$ 是位置上的容量函数;$W:F\to\mathbf{Z}^+$,令牌传递中的加权系数;$\boldsymbol{M}:P\to\mathbf{Z}$(非负整数集合)是位置标识向量,必须满足 $\boldsymbol{M}(p)\leqslant\boldsymbol{K}(p)$;$\boldsymbol{M}_0$ 是初始标识向量.

本文在文献[11]的基础上扩展了一种软件 Petri 网——SEPN. $SEPN=(P,T,F,W,\boldsymbol{M}_0)$,其扩展定义如下:

1) 库所 $P=\{\text{kind},\text{value}\}$ 由类型、值构成,具体的形式定义见文献[11].

2) 扩展变迁集合为

$$T=\{\text{kind},\text{condition},\text{state},\text{pre},\text{post}\}$$

其中,$\text{kind}\in\{0,1\}$,表示触发条件为一元简单组合判断条件或者复杂组合判定条件;condition 为复杂组合判定条件表达式,kind = 0 时 condition 为空;$\text{state}\in\{0,1\}$,表示变迁处于抑制或者激发状态;$t\in T$. t. pre 为前集也记为$'t$,$'t\in P$;t. post 为后集也记为 t',$t'\in P$. 约定变迁的前后集均为库所,使语义清晰,方便自动化处理.

3) 流 $F\subseteq(P,T)\times(T,P)$.

4) 扩展权

$$W=\{s,d,\text{kind},\text{operation},\text{opkind},\text{value},\text{enable},\text{state}\}$$

其中,s 为权起点,$s\in P\cup T$;d 为权终点,$d\in P\cup T$,s 和 d 不可同为库所或变迁,即$((s\in P)\to(d\in T))\wedge((s\in T)\to(d\in P))$;$\text{kind}\in\{0,1\}$,表示权运算为一元简单运算或者复杂组合运算;operation 为复杂组合运算表达式,kind = 0 时,operation 为空;$\text{opkind}\in\{0,1,2,3,4,5,6\}$,当权起点为库所,终点为变迁,对应于 = =,! =, <, >, < =, > = 和无前置约束. 当权起点为变迁,终点为库所,则对应于 =, +, -, *, /, mod,无运算;value 为运算数;$\text{enable}\in\{0,1\}$,表示权在变迁处于 0 抑制或 1 激发状态下可发生,在变迁的图形描述上以虚线或实线对应;$\text{state}\in\{0,1\}$,对应于权是否满足触发条件.

5) 迁移条件和结果:

① 前集可触发条件为满足文献[11]中的前集可触发条件或者$((w.\text{kind}=0)\wedge(w.\text{condition}=\text{true}))$,否则 w. state = 0.

② 后集在激发状态下可触发的条件为满足文献[11]中的后集在激发状态下可触发的条件或者$((w.\text{kind}=1)\wedge(\text{result}(w.\text{operation})\in[P(d).\text{value}.\min,P(d).\text{value}.\max]))$,或者当后集在抑制状态下可激发的条件满足:

$$\forall t,w,(w=t')\wedge(w.s.\text{kind}=1)\wedge(w.d.\text{kind}=0)\wedge(w.\text{enable}=0)\wedge(w.\text{state}=0)$$

w. state = 0 同样必须满足以上 w. opkind 为 0 ~ 6 或者 operation 结果受限的情况下的运算赋值判断来决定抑制状态下跃迁是否可触发.

③ 在网系统的一次变化中,任意的库所不能被不同的权所触发,形式约束定义见文献[11].

2　模型检验

模型检验的特点是自动化并且验证速度快,当所检验的性质未满足时,给出的反例路径常常反映了系统设计中的细微失误,因而对于用户排错有很大的帮助[13]. 建立从形式化的图形模型到验证语言的自动转化将减少人工编写、修改和调试产生的无谓错误和工作量,将大大提高验证工作的效率.

2.1　计算树逻辑

计算树逻辑(CTL, Computational Tree Logic)是一种离散分支时间命题逻辑[12]. CTL 是在命题逻辑中增加了时态算子和路径量词:□always 算子,□A 表示 A 总是为真或者 A 永远为真;◇sometimes 算子,◇A 表示 A 最终为真或者 A 有时为真;○next 算子,○A 表示 A 在下一时刻为真;▷until 算子,$A\triangleright B$ 表示 A 一直为真,直到 B 为真;A 表示所有未来路径;E 表示至少某一路径.

2.2 NuSMV 验证代码生成

本文建立了 SEPN(Software Expanded Petri Net)和 NuSMV 核心语法的形式化映射. 根据语义映射,本文设计了并实现了从 SEPN 模型到 NuSMV 程序的自动转换,程序设计如图 1 所示.

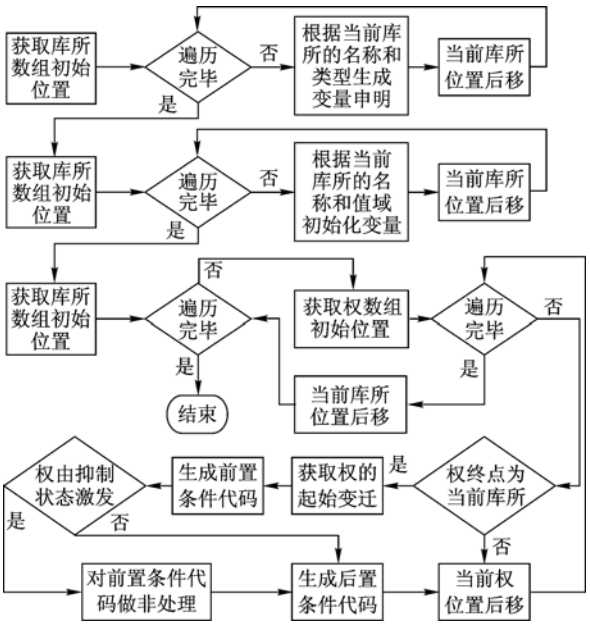

图 1 程序流程图

3 除冰软件实例

本文以某除冰系统为实例,不失一般性,实例中涉及的数据均为虚拟数据. 该系统通过除冰开关和软件对引气活门加断电来控制其开关. 当出现结冰告警时,飞行员按下除冰开关,启动软件,防冰活门加电打开,除冰系统工作. 结冰信号消失 2 min 后,飞行员按出防冰开关,防冰活门断电关闭,防冰系统停止工作. 防冰系统仅允许在飞行状态下工作,在地面检测 30 s 后会自动关闭.

本文对除冰系统进行建模和验证,需求的初始功能模型见图 2. 初始需求功能模型中库所类型和意义见表 1. 初始变迁类型和意义见表 2.

表 1 初始库所类型和意义

库所	类型	意义
Ice_Signal	状态	结冰信号
Cb_On	状态	除冰按钮按下(1)或抬起(0)
Software	状态	软件工作状态
ValveDoor	状态	引气活门打开(1)或关闭(0)
Cb_Timer	数值	除冰信号消失时间
Temperature_xx	数值	温度传感器采样数据
Pressure_xx	数值	压力传感器采样数据
Temperature	数值	温度数据综合值
Pressure	数值	压力数据综合值
Phase	数值	飞行阶段
Test_Time	数值	地面测试时间
Engine_Number	数值	正常工作的发动机数
t_8_fired	状态	活门关闭监控信号

表 2 初始变迁类型和意义

变迁	类型	意义
t_1	一元判断	收到结冰信号
t_2	一元组合判断	地面检测状态且按下除冰按钮
t_3	一元判断	除冰信号消失 120 s
t_4	无前置判断	自动打开活门条件
t_5	无前置判断	收集压力传感器采样数据
t_6	无前置判断	收集温度传感器采样数据
t_7	一元组合判断	是否关闭活门
t_8	一元判断	除冰信号是否消失

初始需求功能模型的验证代码自动转换如图 3 所示,需验证的形式需求如表 3 所示.

通过对验证结果和反例路径的分析,列出了以下部分改进措施:①增加飞行员强制打开和关闭引气活门的功能;②增加压力和温度数据异常

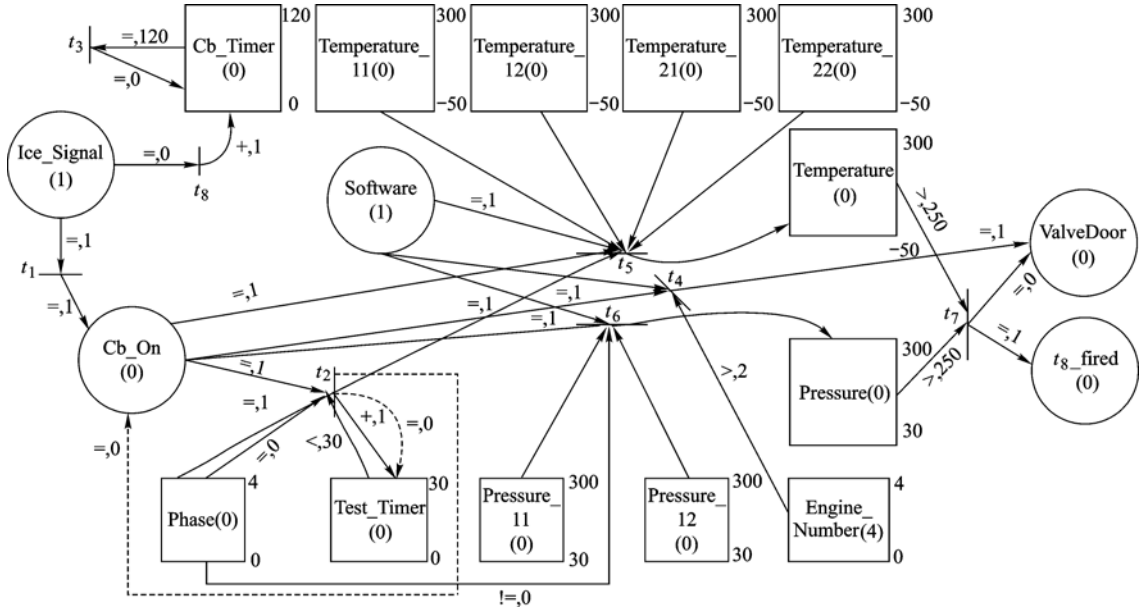

图 2 除冰系统的初始需求功能模型

判断和故障报警(连续 5 次或统计 20 次异常则置故障,否则视为虚警). 完善后的除冰系统需求功能模型如图 4 所示,增加的库所类型和意义见表 4,增加的变迁类型和意义见表 5.

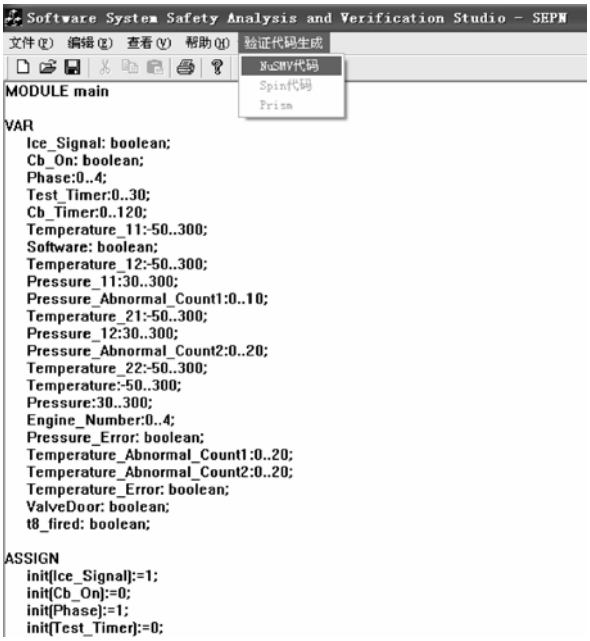

图 3　NuSMV 代码自动生成

表 3　需求的 CTL 描述和验证结果

需求描述	CTL 验证公式	验证结果
在所有执行路径分支中引起活门可打开	AF(ValveDoor = 1)	False
在所有执行路径分支中引起活门可关闭	AF(ValveDoor = 0)	True
在所有执行路径分支中引起活门不可一直处于关闭状态(即一直无法打开)	!EG(ValveDoor = 0)	False
在所有执行路径分支中引起活门不可一直处于打开状态(即一直无法关闭)	!EG(ValveDoor = 1)	False
在所有执行路径分支中,在软件发生故障的情况下,按下除冰按钮,引起活门仍可打开	AF(Cb_On = 1&Software = 0)&(ValveDoor = 1)	False
在所有执行路径分支中,在软件发生故障的情况下,抬起除冰按钮,引气活门仍可关闭	AF(Cb_On = 0&Software = 0)&(ValveDoor = 0)	False
在压力数据异常情况下,软件立即自动关闭引气活门,信号不可被自动触发	AG(Pressure_Error) - > AX(t_8_fired = 0)	初始需求未考虑
在温度数据异常情况下,软件立即自动关闭引气活门,信号不可被自动触发	AG(Temperature_Error) - > AX(t_8_fired = 0)	初始需求未考虑

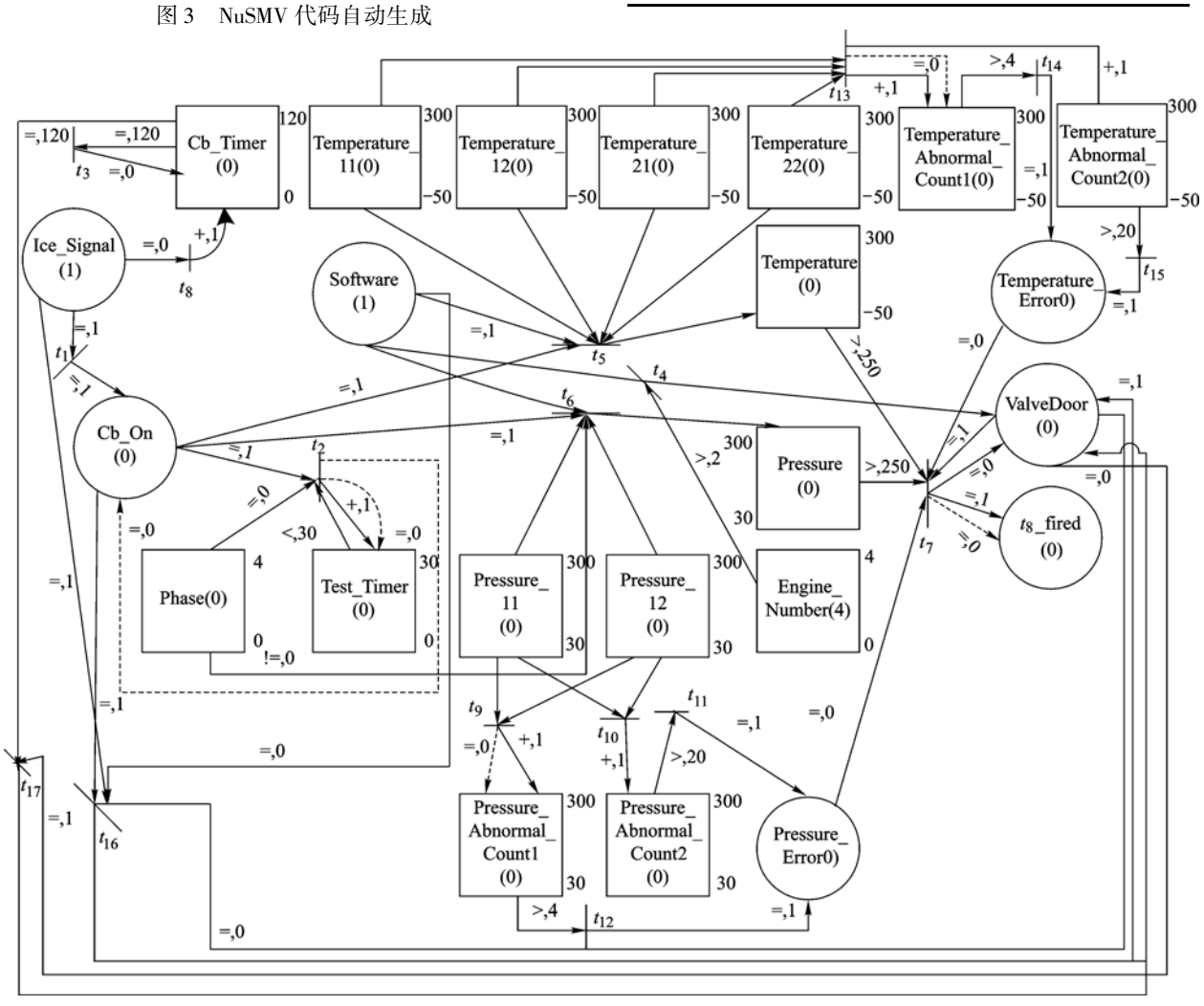

图 4　完善后的除冰系统需求功能模型

表 4 增加的库所类型和意义

库所	类型	意义
Temperature_Abnormal_Count1	数值	温度传感器数据异常的连续出现次数
Temperature_Abnormal_Count2	数值	温度传感器数据异常的统计出现次数
Temperature_Error	状态	温度传感器故障
Pressure_Abnormal_Count1	数值	压力传感器数据异常的连续出现次数
Pressure_Abnormal_Count2	数值	压力传感器数据异常的统计出现次数
Pressure_Error	状态	压力传感器故障

表 5 增加的变迁类型和意义

变迁	类型	意义
t_9	多元组合判断	判断压力传感器数据差异
t_{10}	多元组合判断	判断压力传感器数据差异
t_{11}	一元判断	判断压力传感器数据统计异常是否超过 20 次
t_{12}	一元判断	判断压力传感器数据连续异常是否超过 4 次
t_{13}	多元组合判断	判断温度传感器数据差异
t_{14}	一元判断	判断温度传感器数据连续异常是否超过 4 次
t_{15}	一元判断	判断压力传感器数据连续异常是否超过 20 次
t_{16}	一元组合判断	判断是否正常关闭活门
t_{17}	一元组合判断	判断是否正常打开活门

根据完善后的除冰系统需求模型,自动生成 NuSMV 验证代码后实施验证,最终通过所有的示例性质验证. 本文使用 NuSMV2.4.3 运行后验证结果如图 5 所示.

```
*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat
*** Copyright (c) 2003-2005, Niklas Een, Niklas Sorensson

-- specification AF ValveDoor = 1  is true
-- specification AF ValveDoor = 0  is true
-- specification !(EG ValveDoor = 1)  is true
-- specification !(EG ValveDoor = 0)  is true
-- specification (AG (Cb_On = 1 & Software = 0) -> AF ValveDoor = 1)  is true
-- specification (AG (Cb_On = 0 & Software = 0) -> AF ValveDoor = 0)  is true
-- specification (AG Pressure_Error = 1 -> AX t8_fired = 0)  is true
-- specification (AG Temperature_Error = 1 -> AX t8_fired = 0)  is true

C:\Program Files\NuSMV\2.4.3\bin>
```

图 5 通过验证的结果

4 结 论

本文提出了一种扩展 Petri 网以更好的支持复杂软件系统建模和验证,提高了 Petri 网建模和自动化验证能力,通过模型检验对软件的需求功能模型实施验证,提高了软件的安全性;将继续增加扩展 Petri 网的故障描述机制,基于故障的安全性等级等因素设计并实现 Petri 网的安全性划分算法,以提高利用扩展 Petri 网实施安全性模型检验的效率.

参考文献 (References)

[1] Leveson N G. Software safety: why, what, and how? [J]. ACM Computing Surveys, 1986, 18(2): 125 - 163

[2] 刘克,单志广,王戟,等. "可信软件基础研究"重大研究计划综述[J]. 中国科学基金,2008(3):145 - 151
Liu Ke, Shan Zhiguang, Wang Ji, et al. Overview on major research plan of trustworthy software [J]. Science Foundation in China, 2008(3): 145 - 151 (in Chinese)

[3] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社, 2005:51 - 75
Yuan Chongyi. The principle and application of Petri net [M]. Beijing: Electronics Industry Press, 2005: 51 - 75 (in Chinese)

[4] 姚洪英,范铁生,苏红丽,等. 基于 Petri 网的分布式实时多媒体同步模型[J]. 沈阳工业大学学报,2004,26(1):96 - 98
Yao Hongying, Fan Tiesheng, Su Hongli, et al. A distributed real-time multimedia synchronic model based on Petri net [J]. Journal of Shenyang University of Technolgy, 2004, 26(1): 96 - 98 (in Chinese)

[5] Cicirelli F, Furfaro A, Nigro L. An approach to protocol modeling and validation [C]//39th Annual Simulation Symposium. CA: IEEE Computer Society, 2006: 1 - 8

[6] Franck C, Olivier H R. Structural translation from time Petri nets to timed automata [J]. Journal of Systems and Software, 2006, 79(10): 1456 - 1468

[7] Bell D G, Brat G P. Automated software verification & validation: an emerging approach for ground operations [C]//Aerospace Conference, 2008 IEEE. NY: IEEE Press, 2008: 1 - 8

[8] Heitmeyer C L, Archer M M, Leonard E I. Applying formal methods to a certifiably Secure software system [J]. Software Engineering, IEEE Transactions on, 2008, 34(1): 82 - 98

(下转第 105 页)

AIAA,2009

[6] Romero H, Salazar S, Lozano R. Real-time stabilization of an eight-rotor UAV using optical flow[J]. IEEE Transactions on Robotics,2009,25(4):809 - 817

[7] Barber D B, Griffiths S R, McLain T W, et al. Autonomous landing of miniature aerial vehicles[J]. Journal of Aerospace Computing, Information, and Communication,2007,4(5):770 - 784

[8] Griffiths S, Saunders J, Curtis A, et al. Maximizing miniature aerial vehicles[J]. IEEE Robotics & Automation Magazine,2006,13(3):34 - 44

[9] Rodrguez A F, Andersen E, Bradley J M, et al. Wind estimation using an optical flow sensor on a miniature air vehicle[C]// AIAA Guidance, Navigation and Control Conference and Exhibit. Hilton Head, South Carolina: AIAA,2007

[10] Srinivasan M V, Zhang S W, Chahl J S. Landing strategies in honeybees and possible applications to autonomous airborne vehicles[J]. The Biological Bulletin,2001,200:216 - 221

[11] Liu Xiaoming, Chen Wanchun, Yin Xingliang, et al. Optical flow sensor designed for altitude-holding flight[C]//2010 International Conference on Electrical Engineering and Automatic Control. Zibo:[s. n.],2010:114 - 117

[12] Ruffier F, Franceschini N. Optic flow regulation: the key to aircraft automatic guidance[J]. Robotics and Autonomous Systems,2005,50:177 - 194

[13] Netter T, Franceschini N. A robotic aircraft that follows terrain using a neuromorphic eye[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. EPFL Lausanne Switzerland:[s. n.],2002:129 - 134

[14] 孙虎章. 自动控制原理[M]. 北京:中央广播电视大学出版社,2002:59
Sun Huzhang. Auto control theory[M]. Beijing: Central Radio & TV University Press,2002: 59 (in Chinese)

[15] Barrows G L, Chahl J S, Srinivasan M V. Biomimetic visual sensing and flight control[C]//Proceedings Seventeenth International Unmanned Air Vehicle Systems Conference. Bristol, UK:[s. n.],2002:1 - 15

(编　辑:张　嵘)

(上接第 68 页)

[9] Li Zhen, Liu Bin, Ma Ning, et al. Formal testing applied in embedded software[C]//8th International Conference on Reliability, Maintainability and Safety. Washington, DC: IEEE Computer Society,2009,2:1 - 5

[10] Ma Ning, Bao Xiaohong, Li Zhen, et al. Verification of safety-critical software requirement based on Petri-net model checking [C]//Supplemental Proceedings 20th International Symposium on Software Reliability Engineering. Washington, DC: IEEE Computer Society,2009

[11] 李震,刘斌,殷永峰,等. 基于扩展 Petri 网的安全关键软件需求模型检验[J]. 沈阳工业大学学报,2011,33(1):113 - 120
Li Zhen, Liu Bin, Yin Yongfeng, et al. Requirement model checking of safety-critical software based on expanded Petri net [J]. Journal of Shenyang University of Technology, 2011, 33(1):113 - 120(in Chinese)

[12] 古天龙. 软件开发的形式化方法[M]. 1 版. 北京: 高等教育出版社,2005:47 - 51
Gu Tianlong. Formal methods of software development[M]. 1st ed. Beijing: Higher Education Press, 2005: 47 - 51 (in Chinese)

[13] 沈胜宇. 模型检验的反例解释[D]. 长沙: 国防科学技术大学计算机学院,2005:1 - 7
Shen Shengyu. Explaining counterexample of model checking [D]. Changsha: School of Computer, National University of Defense Technology,2005:1 - 7(in Chinese)

[14] Roberto C, Alessandro C, Charles A J, et al. NuSMV 2.4 user manual [EB/OL]. Italy, Trento: CMU and ITC-irst, 2005 [2009]. http://nusmv. irst. itc. it/

(编　辑:赵海容)