第 36 卷 第 2 期 计 算 机 工 程 2010 年 1 月
Vol.36 No.2 Computer Engineering January 2010
·软件技术与数据库· 文章编号：1000—3428(2010)02—0039—03 文献标识码：A 中图分类号：TP311

基于 Petri 网的容错系统分层建模

周月明[1]，杜玉越[1,2]，刘 伟[1]
(1. 山东科技大学信息科学与工程学院，青岛 266510；2. 中国科学院计算机科学国家重点实验室，北京 100080)

摘 要：扩展 Petri 网的结构，将一般的控制管理系统改进为具有容错功能的控制管理系统，应用扩展 Petri 网对其进行分层建模，包括系统的基础层、检错层、恢复层。采用故障恢复策略并使用带有基于 Petri 网分层模型中状态标识的行为跟踪、行为检错和行为改错增加控制管理系统的容错功能，并分析了模型的正确性。

关键词：Petri 网；容错；分层建模；检错

Layer Modeling of Fault-tolerant System Based on Petri Nets

ZHOU Yue-ming[1], DU Yu-yue[1,2], LIU Wei[1]
(1. College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510;
2. State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100080)

【Abstract】This paper extends the structure of Petri nets. The generic controlling and managing system is improved into fault tolerance one. The system is layer modeled with Petri nets, which are basic layer, error detection layer and recover layer. Fault-tolerance is added into the system with recover fault policy and behavior tracking with state logos Petri-based layered modeling, error detection for software behavior and recovers the error. And correctness of this system is analyzed.

【Key words】Petri nets; fault-tolerance; layer modeling; error detection

1 概述

随着计算机技术的飞速发展和普及，系统的可信性变得越来越重要。如果软件出现故障就可能导致安全事故、财产损失。提高计算机可靠性的技术可以分为避错技术和容错技术[1]。软件容错策略可分为故障避免策略、故障屏蔽策略和故障恢复策略。直接开发容错应用是非常困难的，因为开发者不仅要处理复杂的应用逻辑，还要面对纷繁的容错逻辑。为方便快速地开发出容错软件，本文利用 Petri 网建模，采用故障恢复策略并使用了带有基于 Petri 网分层模型中状态标识的行为跟踪、行为检错和行为改错功能的控制管理系统容错软件。有关 Petri 网的基本概念术语可查阅文献[2-3]。

2 系统软件的容错模型

为开发出高质量低错误甚至无错误的系统软件，按照功能把系统软件设计为 3 层：

(1)基础层。此层是没有检错和改错功能的系统软件，做基本的且必不可少的行为，此层中可能有难以查找的错误和缺陷。

(2)检错层。跟踪基础层的软件行为，及时检查行为的错误、缺陷，并把检查出的错误、缺陷的行为名传递给修复层。

(3)修复层。根据从检错层收到的行为错误、缺陷，用该行为的其他版本进行改正基础层行为的失误。

2.1 系统软件的基础层

某地有一铁路与公路的相交口，铁路靠公路的一段信号灯显示其使用情况，单杠门控制公路交通。信号灯显红色代表有火车在使用铁路，红灯一亮单杠门关闭，汽车就无法通过铁路。

在铁路靠公路的最后一段的 2 个端点各有 1 个接收器，接收火车到来的信息。收到信息后，把信息传递给总控制系统，系统就把信号灯设置为红灯且把门关闭。当火车离开另一个端点时，该接收器把信息传递给总控制系统，系统把信号灯设置为绿灯且把门打开。图 1 为用 Petri 网表示的控制系统基础层。

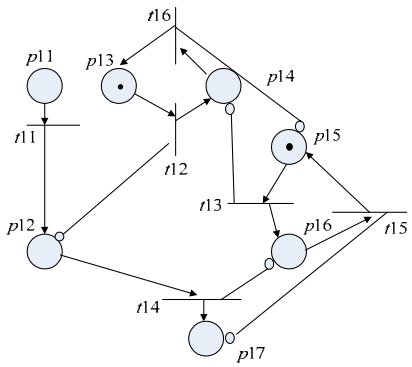

图 1 用 Petri 网表示的控制系统基础层

图 1 中的 $p11$ 是由端点接收器把火车信息传送到系统时增加一个 token。$p11$ 有 token 后系统就进入工作阶段。关于其中变迁和库所的含义如表 1 所示。

基金项目：国家自然科学基金资助项目(60773034, 90818023, 60803032)；山东省科技发展计划基金资助项目(2008GG30001024)；山东科技大学 2009-2010 年度研究生创新基金资助项目(YCA090319)
作者简介：周月明(1984－)，女，硕士研究生，主研方向：工作流，可信软件，Petri 网理论与应用；杜玉越，教授、博士、博士生导师；刘 伟，博士研究生
收稿日期：2009-07-30 **E-mail：**zhouyueming168@163.com

表 1　图 1 中变迁和库所的含义

库所	含义	变迁	含义
$p11$	火车进路段	$t11$	接收器接收并传递信息
$p12$	火车在该道行驶中	$t12$	信号灯由绿变红
$p13$	信号灯显绿色	$t13$	关闭单杠门
$p14$	信号灯显红色	$t14$	火车过交叉口
$p15$	单杠门开状态	$t15$	打开单杠门
$p16$	单杠门关状态	$t16$	信号灯由红变绿
$p17$	火车已过交叉口		

2.2　系统软件的检错层

系统软件的检错层由行为跟踪和行为判断 2 个部分组成。行为跟踪是对系统的动作进行跟踪，向行为判断传递状态信息。行为判断部分判断系统的执行是否正确，如不正确则向恢复层发送信息。

2.2.1　软件的行为跟踪与检错

为说明检错层如何从基础层接收信息，首先给出几个定义。

定义 1 设 Petri 网(P,T,F,M_0)，对任意 $p_i\in P$，存在 $t_1, t_2\in T$, $p_j\in P$, $p_i^{\bullet}=t_1$, $t_1^{\bullet}=p_j$，则称 p_i 是 p_j 的直接前库所。$p_j^{\bullet}=t_2$, $t_2^{\bullet}=p_k$，则称 p_i 是 p_k 的间接前库所，称 p_i, p_j 是 p_k 的前库所。

定义 2 设 Petri 网(P,T,F,M_0)，$P=(p_1,p_2,\cdots,p_n)$是基本有序库所集，当且仅当对于所有 t_i, $i<n$，在 $P1=(p_{i+1},p_{i+2},\cdots,p_n)$中不存在库所 p_j，且 $P2=(p_1,p_2,\cdots,p_i)$中的 p_k, p_j 是 p_k 的前库所。

定义 3 五元组 $DPN=(P,T,F,M_0,B)$称为带库所集标识的 Petri 网，当且仅当：

(1)$PN=(P,T,F,M_0)$是一般 Petri 网。

(2)其 $P=(p_1,p_2,\cdots,p_n)$是基本有序状态集，B 是一个 n 位二进制数，B 与 P 的 n 个库所从左到右相对应的，当 p_i 有 token 时，B 的第 i 位是 1，否则第 i 位是 0，则称这个数字 B 为库所集标识。

定义 4 设 Petri 网(P,T,F,M)，对于 $t_i\in T$，且 $p_j\in P$, $t_i^{\bullet}=p_j$，则称 t_i 是 p_j 的激发变迁。

检错层的接收装置与基础层的每个库所连接，该接收装置输出系统基础层的状态标识。每个基础层变迁地执行更新状态标识，更新后的状态标识传递给检错层，检错层再根据状态标识判断基础层刚执行过的变迁，即状态标识的二进制中 1 相对应的库所的激发变迁，再对这些变迁检查是否执行正确。如果执行错误就发送信息到恢复层。

定义 5 设 $DPN=(P,T,F,M_0,B)$，B 对应位为 1 的库所集 $P1$，则 $T1=\{t|t_\in T\wedge p_\in P_1\wedge p^{\bullet}=t\}$为 B 的后置变迁集。

检错层内有一个计时器，当接收到信息时计时器置零，指定时间后状态标识还没改变，就根据上次的状态标识得出后置变迁集。把后置变迁集发送给检错层，检错层对后置变迁集的行为进行检错，再把计时器置零。

2.2.2　Petri 网的 OR-split 的扩展结构

系统主要包括顺序、并行、选择、循环 4 种结构，Petri 网的路由也主要包括这 4 种结构。传统 Petri 网采用规范的模型语言，在表示这 4 种结构时具有一定的严谨性，选择结构用 OR-split 表示。当涉及的路由数较少时，分辨还比较容易，一旦过程较为复杂时，混淆性则极大地增加，很难快速、清晰地辨别。文献[4]中 OR-split 结构用符号⊠表示。为减少 Petri 网混淆性，本文选用更为特殊、规律化、接近程序语言的符号表示。

定义 6 在 Petri 网中变迁的输出端用符号 $\boxed{\begin{matrix}\text{If}(p)\\ \text{else}\end{matrix}}$ 表示 OR-split，则称符号 $\boxed{\begin{matrix}\text{If}(p)\\ \text{else}\end{matrix}}$ 为 OR-split 的扩展形式。其中，p 是逻辑表达式或逻辑变量；If, else 是 C 语言中的选择语言。

图 2 是 OR-split 的扩展图。其中，$p2$ 是逻辑表达式 p 的值是 true 的输出结果；否则输出 $p3$。其 OR-split 的图示表示使得模型分析较为简单、明确。

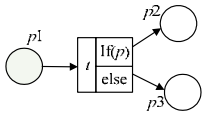

图 2　Petri 网中 OR-split 扩展结构表示

图 3 是用 Petri 网表示检错层的内部结构。$t23$ 中 p 为促发变迁的执行正确。如果被判断促发变迁执行正确，就在 $p25$ 增加 1 个 token，否则 $p24$ 增加 1 个 token。表 2 是图 3 中库所和变迁的意义。

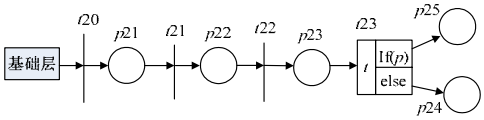

图 3　交叉口控制系统的检错层

表 2　图 3 中库所与变迁的含义

库所	含义	变迁	含义
$p21$	库所集标识	$t20$	接收基础层的库所集标识
$p22$	B 的促发变迁集的标识	$t21$	分析基础层的库所信息
$p23$	执行 B 的促发变迁集错误的标识	$t22$	算出 B 的促发变迁
$p24$	促发变迁执行失效或错误的标识	$t23$	判断促发变迁的执行的正确性
$p25$	促发变迁执行正确的标识		

2.3　系统的恢复层

为实现改正基础层错误行为功能，在恢复层存储了基础层行为的其他 2 种版本，它们和基础层的行为版本是用不同的算法实现行为的模块。为区分这 3 种版本，把基础层存储变迁集称为主本，恢复层中变迁的 2 种版本分别称为主动副本和被动副本。如果主版本发生失效率大，说明变迁的主本可能有错误，该主版本用主动副本覆盖，则被动副本就改为主动副本。如果这 3 个版本用相同的软件版本，当软件失效发生时，容错措施就完全丧失了其防护能力。图 4 为用 Petri 网表示检错层的内部结构。表 3 是图 4 中库所和变迁的意义。

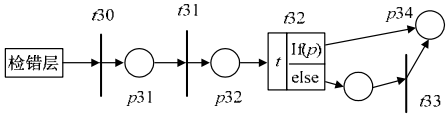

图 4　交叉口控制系统的恢复层

表 3　图 4 中变迁和库所的含义

库所	含义	变迁	含义
$p31$	从检错层接收的变迁集	$t30$	接收出错的变迁标识集
$p32$	执行完变迁的主副版本的状态	$t31$	执行变迁的主副本
$p33$	存出错率大于 0.5 的变迁	$t32$	判断变迁的出错率是否大于 0.5
$p34$	无出错率大于 0.5 的变迁恢复层结束	$t33$	出错率＞0.5 的主动副本覆盖主版本，被动覆盖主动副本

3　系统的整体模型与分析

3.1　系统的整体模型

系统的整体模型如图 5 所示。其中，$t20$ 是逻辑输入变迁[5]，$f_I=p11\vee p12\vee p13\vee p14\vee p15\vee p16\vee p17$。基础层为检错层提供信息，检错层检查基础层的变迁执行的有效性；恢复层接收检错层检查出基础层出错的变迁，对出错的变迁用执行其主动副本进行补充，对基础层出错率大的变迁用其主动副本进行自动改正。这 3 层的有机组合使系统始终处在安全的状态。

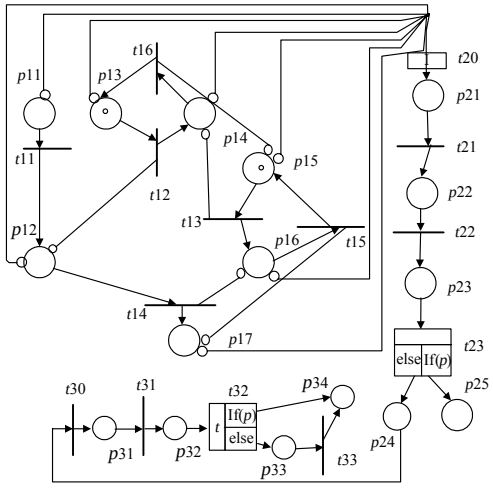

图 5　容错系统整体模型

3.2　分析模型的正确性

计算可达图是验证 Petri 网正确性的主要方法。只要能对 Petri 网模型做出可达图，就可以证明此模型是正确的。Petri 网的可达图是由以下算法构造的图结构，它的每一个节点 x 都有一个标记 Mx，Mx 是从到非负整数或 w 的映射，即 Mx: p->(0,1,2,…)∪{w}。

算法的步骤如下：

(1)$G(\sum)$的初值只有一个顶点 r，$Mr=M_0$。

(2)令 x 为 $G(\sum)$中的叶顶点。若任何变迁 Tt，在 Mx 均无发生权，则 Mx 为真叶顶点；若 $G(\sum)$的所有叶顶点均为真叶顶点，算法终止，否则执行(3)。

(3)若有叶节点 x，x 不是真叶顶点。于是在 Mx 标识下至少有一个变迁有发生权。在 Mx 对每一个有发生权的变迁添加一个顶点 y，添加一个从顶点 x 到 y 的有向弧，有向弧用变迁 t 标记。顶点 y 的标记 My 的定义如下：首先计算出 Mx 的后继 M'，即对所有 $p\in P$，如果 $Mx(p)=w$ 时，$M'(p)=w$；否则当 $p\in{}^\bullet t\text{-}t^\bullet$时，$M'(p)=Mx(p)$-1；当 $p\in t^\bullet\text{-}{}^\bullet t$时，$M'(p)=Mx(p)$+1。然后计算 My：对所有 $p\in P$，若从 r 到 y 的路径上有节点 z，使得 $Mz<M'$，且 $Mz(p)<M'(p)$，则 $My(p)=w$。否则，$My(p)=M'(p)$。再检查 $G(\sum)$的其他顶点标识是否与 My 相等，如果有存在顶点 $Ma=My$，则合并 2 个顶点。

(4)回到步骤(2)。

根据构造算法对系统 Petri 网系统做分析，可以得到系统模型的可达图 $G(\sum)$。从可达图可知，系统 Petri 网模型的每个转移都有发生权。因此，此模型是正确的。

4　结束语

本文为使 Petri 网建模和分析更加清晰，扩展了 Petri 网的结构，对交通控制系统进行了分层建模。为构造具有容错功能的系统，使用了行为跟踪、行为检错和行为改错技术。并对模型的正确性进行了分析。此容错软件有众多的冗余单元，是一般软件的 2 倍多，增大了程序规模，增加了资源消耗，一般用于失效后果严重的场合。下一步工作包括 OR-split 的多分支结构及减少此容错软件的冗余。

参考文献

[1] 徐仁佐. 软件可靠性工程[M]. 北京: 清华大学出版社, 2007.

[2] Du Yuyue, Jiang Changjun. On the Design and Temporal Petri Net Verification of Grid Commerce Architecture[J]. Chinese Journal of Electronics, 2008, 17(2): 247-251.

[3] Du Yuyue, Jiang Changjun, Zhou Mengchu. Modeling and Analysis of Real-time Cooperative Systems Using Petri Nets[J]. IEEE Trans. on Systems, Man, and Cybernetics, 2007, 37(5): 643-654.

[4] 殷　鹏. 基于扩展 Petri 网的 EEPC 测试模型研究[J]. 计算机应用, 2007, 27(5): 1245-1247

[5] 杜玉越, 蒋昌俊. 一种基于逻辑 Petri 网的协同实时系统工作流模拟技术[J]. 计算机学报, 2004, 27(4): 471-481.

编辑　顾逸斐

(上接第 32 页)

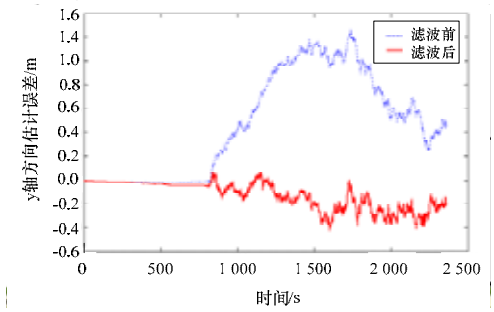

图 3　*Y* 轴方向的估计误差

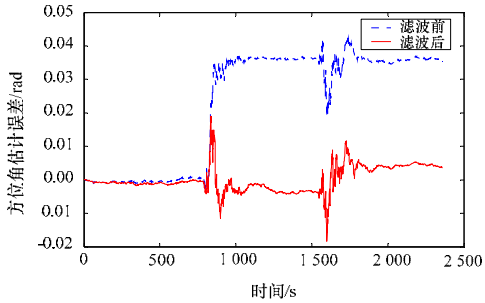

图 4　方位角的估计误差

7　结束语

本文以贝叶斯规则作为理论基础，建立了移动机器人 SLAM 算法的概率表示模型，通过 EKF 来实现 SLAM 算法，仿真实验表明该方法为实现 SLAM 算法提供了一种有效可靠的途径，但是由于 EKF 难以避免会出现不一致现象，存在模型误差、线性化误差等因素对其精度的影响。

参考文献

[1] 康叶伟, 黄亚楼, 孙凤池, 等. 一种基于 RBUKF 滤波器的 SLAM 算法[J]. 计算机工程, 2008, 34(1): 17-19.

[2] Thrun S, Fox D, Burgard W. Probabilistic Algorithms and the Interactive Museum Tour-guide Robot Minerva[J]. International Journal of Robotics Research, 2000, 19(11): 972-999.

[3] 厉茂海, 洪炳熔. 移动机器人的概率定位方法研究进展[J]. 机器人, 2005, 27(4): 380-384.

[4] 陈卫东, 张　飞. 移动机器人的同步自定位与地图创建研究进展[J]. 控制理论与应用, 2005, 22(3): 455-460.

[5] Bailey T. Mobile Robot Localisation and Mapping in Extensive Outdoor Environments[D]. Sydney, Australia: University of Sydney, 2002.

编辑　金胡考