

WF-net 模型到 PNML 的转换方法研究与实现

周建涛 海晓军

(内蒙古大学计算机学院 呼和浩特 010021)

摘要 Petri 网标记语言 PNML 是一种基于 XML 的支持不同 Petri 网工具之间文件交换的标准格式,对于解决不同 Petri 网变种之间的互操作性问题具有重要意义。工作流网 WF-net 是用于工作流系统建模和定性、定量分析的重要形式化技术之一。工作重点是将 WF-net 模型转换为 PNML 格式。首先,在标准 PNML 元模型的基础上,通过元素扩展提出 WF-net 的 PNML 元模型。然后,依据元模型,提出 WF-net 模型中各元素到 PNML 格式的转换规则,包括起始/结束库所转换、四种触发器转换和四种分支/合并结构转换。最后设计并实现了 WF-net 描述到 PNML 描述自动转换的工具软件。这些工作为 Petri 网标准交换格式的制定做了有力的探索。

关键词 元模型,WF-net,PNML,转换规则

中图法分类号 TP301.2 **文献标识码** A

Research and Implementation of Transition from WF-net to PNML

ZHOU Jian-tao HAI Xiao-jun

(College of Computer Science, Inner Mongolia University, Hohhot 010021, China)

Abstract Petri Net Markup Language (PNML) is an interchange format for different kinds of Petri nets based on XML. It plays an important role for inter-operability among different variations of Petri nets. Workflow net (WF-net) is one of formal techniques for workflow modeling and qualitative and quantitative analysis. The work done in this paper is transition from WF-net model to PNML format. Firstly, PNML meta model for WF-net was presented based on standard meta model by extending those elements not in the original model. Secondly, ten transition rules from WF-net to PNML were given based on the meta model presented above, including source place / sink place transition, four triggers transition, and four fork/join constructs transition. Finally, an automatic transition tool was designed and implemented. The work is exploratory research on standard, formal interchange format of Petri net.

Keywords Meta model, WF-net, PNML, Transition rule

1 引言

Petri 网作为一种图形化描述与严格数学定义相融合的形式化模型,已成为描述具有分布、并发、异步特征的离散事件动态系统的有效建模工具。目前, Petri 网已经有很多变种,一些有影响的研究机构还开发了工具软件,用以编辑、保存并对 Petri 网进行分析。然而,不同 Petri 网、不同 Petri 网支持工具的差异导致了其间较低的互操作性,这一问题很大程度上限制了 Petri 网的深入发展和广泛使用。

为了解决 Petri 网互操作性问题,使 Petri 网存储格式标准化,在 2000 年 Petri 网应用和理论国际会议上,研究者们对这一问题进行了研讨, Petri 网标记语言 (Petri Net Markup Language, PNML)^[1] 就是其中的方案之一。之后, PNML 进一步发展^[2-5],成为基于 XML 的支持不同 Petri 网工具之间文件交换的标准格式。

工作流网 (WorkFlow-net, WF-net)^[6] 是一种基于 Petri

网的、用于工作流过程描述和分析的形式化工具。为了扩展 WF-net 的使用方式和范围,并使它与更多的 Petri 网支持工具兼容,论文研究了 WF-net 到 PNML 的转换问题。

本文第 2 节介绍 WF-net 和 PNML 的基本理论知识;第 3 节提出 WF-net 的 PNML 元模型,并按类别阐述 WF-net 到 PNML 的转换规则;第 4 节介绍自动转换工具的设计与实现;最后总结全文的工作。

2 基础知识

2.1 工作流网

WF-net 是基于 Petri 网的工作流建模工具,可以描述 workflows 的控制过程,包括 4 种典型的控制流结构:顺序、并行、分支和循环。一个 WF-net 有一个输入库所和一个输出库所,任何被 WF-net 描述的过程实例都在它进入工作流管理系统时被创建,在处理完毕时被删除;换句话说,就是 WF-net 可以描述一个过程实例的生命周期。

收稿日期:2010-01-25 返修日期:2010-03-29 本文受国家自然科学基金(60863015),国家科技部春晖计划项目(Z2007-1-01042),内蒙古自然科学基金项目(20080404ZD20)资助。

周建涛(1974—),女,博士,教授,CCF 会员,主要研究方向为计算机软件、计算机网络,E-mail:csztjao@imu.edu.cn;海晓军(1980—),男,硕士,主要研究方向为基于形式化方法的软件建模技术。

WF-net 中,变迁表示可执行的活动;库所表示活动的使能条件。被使能的活动要真正地被执行,必须得到相应的触发器的触发,WF-net 的触发器分为 4 种类型:

(1)自动触发:活动一旦使能就可以立即开始执行。这种类型的触发器常用于无需外界干预的情况下,由程序自动执行。

(2)人工触发:活动的执行者通过属于自己的工作流任务的列表选取活动来执行,出现在该任务列表中的所有活动都已处在使能状态。

(3)消息触发:由系统外部消息(如用户登录、新邮件到达)来触发活动使之执行。

(4)时间触发:由控制时间的定时器来触发已使能的活动。

在使用 WF-net 进行工作流建模时,需要将这些触发方式标注在对应的变迁节点上,以表示对应的活动或任务的触发方式。

2.2 PNML

PNML 是一种基于 XML 的支持不同 Petri 网工具之间文件交换的标准格式,它的出现满足不同类型的 Petri 网文件交换与 Petri 网存储格式标准化的实际需求。PNML 由三部分组成,第一部分是基本概念、定义和图形注释等;第二部分是对转换格式的定义,已经定义的 Petri 网类型包括 P/T 网、高级 Petri 网(High-level Petri Net Graphs, HLPNG)和对称网(Symmetric Nets)等;第三部分为后续的扩展作保留。

PNML 文件由几个部分组成,各部分之间的关系如图 1 所示。元模型(Meta Model)定义 PNML 文件的基本结构、相关概念及其间关系。特性定义接口(Feature Definition Interface)允许新型 Petri 网特性的定义,类型定义接口(Type Definition Interface)允许新型 Petri 网类型的定义,这几个部分是固定不变的。而协议文档(Conventions Document)是可变化的,是随着特性定义接口和类型定义接口中定义的新的特性和类型而变化的。

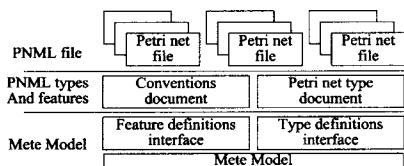


图 1 PNML 文件组织结构

3 WF-net 到 PNML 的转换

WF-net 作为一种基于工作流的形式化建模工具,已经在 Petri 网特性证明和正确性验证方面开展了很多研究工作,并取得了相当重要的研究成果,使得 WF-net 在工作流建模领域得到了较广泛的使用。但是,目前尚没有在 PNML 中给出相应的概念和结构等定义。

WF-net 是 P/T 网,其基础部分可以由 PNML 元模型中的 P/T Net 部分直接描述,其它扩展部分由下一小节提出的扩展元模型描述。

3.1 扩展元模型

模型中的 Place, Transition, Arc 等几个基本元素的元模型已经在 P/T Net 元模型中提出并给出定义。WF-net 与 P/T

Net 差异部分包括:唯一的开始库所、唯一的结束库所、变迁的触发器以及分支/合并等特殊构造,因此扩展的 PNML 元模型的 WF-net 部分如图 2 所示。

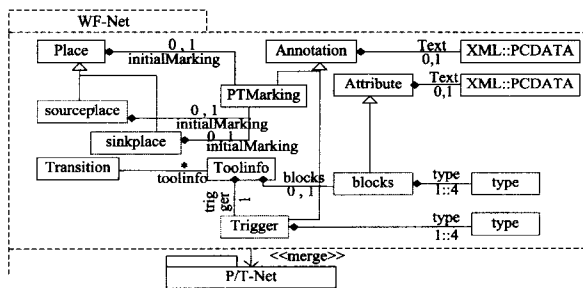


图 2 WF-net 元模型

3.2 转换规则

依据元模型给出 WF-net 元素到 PNML 的转换规则,共三类。分别是:起始库所/结束库所的转换规则;4 类触发器的转换规则;4 种分支构造模块的转换规则。

3.2.1 起始库所/结束库所的转换

WF-net 有且仅有一个起始库所,工作流过程模型的起始库所中有托肯(token)就意味着工作流实例可以开始一次执行。起始库所由新定义的标签对 `<sourceplace>` 和 `</sourceplace>` 表示,其中子元素 `<graphics>` `</graphics>` 标签描述库所的图形信息,也就是在图形编辑器中的具体位置信息;子元素 `<name>` `</name>` 描述库所的名字信息;子元素 `<initialMarking>` `</initialMarking>` 描述库所的初始标识;子元素 `<capacity>` `</capacity>` 表示起始库所的容量,而且由 `<value>` `</value>` 标签对表示的值为 1,即库所的容量为 1,最多含有一个托肯。

与起始库所相对应,WF-net 有且仅有一个结束库所,工作流过程模型的结束库所中流入托肯就意味着工作流过程实例的一次执行结束。结束库所由新定义的标签对 `<sinkplace>` 和 `</sinkplace>` 表示,其余标签的定义与起始库所类似。

以起始库所为例,转换规则如图 3 所示。

WF-Net模型元素	PNML文本
	<pre> <sourceplace id="i"> <graphics> <position x=" " y=" " /> </graphics> <name> <value> /> </name> <graphics> <offset x=" " y=" " /> </graphics> <name> <value> /> </name> <initialmarking> <value>1</value> </initialmarking> <capacity> <value>1</value> </capacity> </sourceplace> </pre>

图 3 起始库所的转换

3.2.2 触发器变迁的转换

图 4 所示为自动触发器变迁的转换,描述触发器的标签 `<trigger>` `</trigger>` 的属性 `<type>` 应当是 Auto。有一点需要注意的是,在 WF-net 中的普通变迁的触发器同样可以认为是自动触发,因为变迁的触发是由系统随机点火的,适用于一般 Petri 网变迁的点火规则。

其他三种类型的触发器:人工、消息和时间触发器的转换规则中除了表示触发器类型的属性值 `<type>` 有所不同之外,其他标签的定义与自动触发器的定义类似。

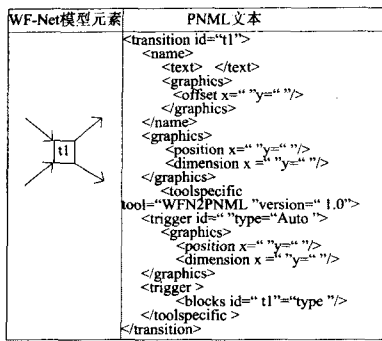


图4 自动触发器的转换

3.2.3 分支构造模块的转换

控制流分支构造模块变迁的转换规则与普通变迁的转换有所不同,应当区别对待。变迁的描述与库所的描述不同之处在于:对于变迁来说,有描述变迁触发器和控制流构造模块的属性,这些属性由<toolspecific></toolspecific>描述。其中子元素<blocks></blocks>描述控制流构造模块,构造模块类型由<type>属性描述。图5所示为带有自动触发器的与分支构造模块的转换。<blocks>标签的<type>属性描述的是 AND_SPLIT 变迁,表示构造块类型是与分支。

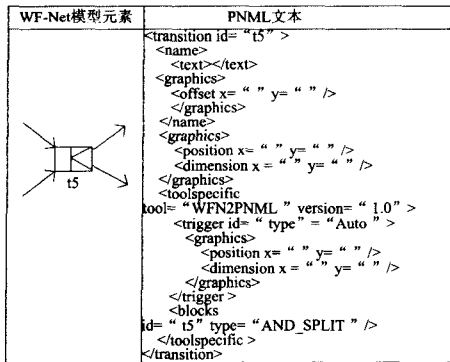


图5 与分支结构的转换

在 WF-net 其他控制流程分支构造模块的转换规则的定义中,除了表示类型的属性<type>不相同之外,描述属性的其余标签的定义类似。

4 转换工具的设计与实现

4.1 相关工作介绍

针对实际业务流程建立 WF-net 模型、模拟业务过程的执行并进行分析离不开建模仿真工具的支持。在论文工作开始时还没有发现 WF-net 和 PNML 相结合的工作。2009 年 3 月国外的一些科研组织发布了名为 WoPeD(The WorkFlow Petrinet Designer)^[7] 的工作流建模工具软件,该工具软件支持控制流构造模块的建模,使用通用的 PNML 格式存储模型等。

但是,本文的工作作为 WF-net 到 PNML 转换工作的探索,仍然可以为后续的研究提供基础性借鉴。

4.2 工具体系结构

支持 WF-net 模型编辑、执行和转换的工具由 3 个模块组成:图形编辑模块、模型仿真模块和转换模块。这些模块都是由应用程序控制器控制协调的,应用程序控制器是整个工具的控制枢纽与控制中心。各个模块之间的关联关系如图 6 所示。

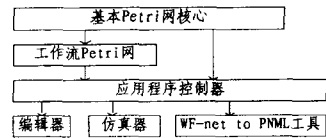


图6 系统体系结构

(1) 基本 Petri 网:描述基本 Petri 网元素的属性,包括库所、变迁、有向弧、点火规则、标识等。在设计和实现时应用面向对象技术和可扩展的思想,便于对基本的 Petri 网进行扩展,及实现各种不同类型的 Petri 网。

(2) 工作流 Petri 网:不同的应用领域对 Petri 网的应用有不同的需求,因此需要不同的 Petri 网,如时间 Petri 网、随机 Petri 网等。可在基本 Petri 网的基础上实现扩展,而可扩展的思想主要表现在对库所、变迁、弧、网这四个基本的 Petri 网元素的属性扩展上。

(3) 应用程序控制器:负责各种应用程序(编辑器、仿真工具、WF-net 转换为 PNML 工具)的调度,负责程序的输入输出。同时引入应用程序控制层可以方便地实现应用程序之间的相互调用,以及程序之间的相互通信。

(4) 编辑器:图形化界面实现 Petri 网的编辑、显示、修改等。

(5) WF-net 到 PNML 的转换工具:把用户建立的 WF-net 转变成 PNML,下一小节叙述它的基本转换过程。

4.3 基本转换过程

本文实现的 WF-net 到 PNML 的自动转换工具,不仅可以完成模型元素(库所、变迁、有向弧)的创建、编辑、删除、修改和存储等基本功能,还可以反向读取已有的 PNML 格式存储的 Petri 网文件,恢复成 Petri 网的图形表示。

从 WF-net 到 PNML 的转换过程为:通过图形编辑器的编辑界面捕捉模型元素,提取图形元素的坐标、大小等属性,依据如上定义的转换规则,分别得到各模型元素的 XML 描述,之后利用 Schema 模式对 XML 文本格式进行规范化处理,整合之后就可以得到模型的 PNML 格式文件,如图 7 所示。

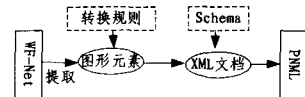


图7 基本转换过程

例如,图 8 所示的一个 PN-net 经工具自动转换后,得到如图 9 所示的 PNML。

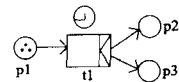


图8 一个 PN-net 例子



图9 转换后的 PNML 描述

- Architecture in Requirements Engineering[C]//Proceedings of the First International Conference on Requirements Engineering, 1994;239-245
- [36] Hall J G, Mistrík I, Nuseibeh B, et al. Relating Software Requirements and Architectures[J]. IEE Proceedings Software, 2005,152(4):141-142
- [37] Bass L, Clements P, Kazman R. Software Architecture in Practice(2nd Edition)[M]. SEI Series in Software Engineering, Addison-Wesley Professional, April 2003
- [38] Kruchten P, Obbink H, Stafford J. The Past, Present, and Future for Software Architecture[J]. IEEE Software, Special issue on Software Architecture, 2006,23(2):22-30
- [39] Perry D E. Software Engineering and Software Architecture[C]//Proceedings of the International Conference on Software: Theory and Practice, Beijing:Electronic Industry Press, 2000;1-4
- [40] Booch G. The Economics of Architecture-first [J]. IEEE Software, 2007,24(5):18-20
- [41] Royce W W. Managing the Development of Large Software Systems; Concepts and Techniques[C]//Proceedings of the 9th International Conference on Software Engineering (ICSE'87). 1987;328-338
- [42] Swartout W, Balzer R. On the Inevitable Intertwining of Specification and Implementation [J]. Communications of ACM, 1982,25(7):438-440
- [43] Garlan D, Perry D. Software Architecture: Practice, Potential, and Pitfalls[C]//Proceedings of the 16th International Conference on Software Engineering, May 1994;363-364
- [44] Rajasree M S, Reddy P J K, Janakiram D. Pattern-oriented Software Development: Moving Seamlessly from Requirements to Architecture[C] // Proceedings of the Second International Workshop on from Software Requirements to Architectures (ICSE'03), May 2003
- [45] Grünbacher P, Egyed A, Medvidovic N. Reconciling Software Requirements and Architectures with Intermediate Models[J]. Software and System Modeling, 2004,3(3):235-253
- [46] Chung L, Gross D, Yu E. Architectural Design to Meet Stakeholder Requirements[C]//Proceedings of the First Working IF-IP Conference on Software Architecture (WICSA). San Antonio, Texas, USA, 1999
- [47] Nuseibeh B. Weaving Together Requirements and Architectures [J]. IEEE Computer, 2001,34(3):115-119
- [48] Kozaczynski W. Requirements, Architectures and Risks [C]//Proceedings of IEEE Joint International Conference on Requirements Engineering, 2002;6-7
- [49] Jackson M. Problem Frames: Analyzing and Structuring Software Development Problems [M]. Addison-Wesley Longman Publishing Co., Inc. 2001
- [50] Diallo M H, Sim S E, Alspaugh T A. Case Study, Interrupted; the Paucity of Subject Systems that Span the Requirements-architecture Gap [C]//Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies; held in conjunction with the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE), 2007;43-48
- [51] de Boer R C, van Vliet H. On the Similarity Between Requirements and Architecture[J]. Journal of Systems and Software, 2009,82(3):544-550
- [52] Garlan D. Software Architecture; a Roadmap[C]//Proceedings of the Conference on the Future of Software Engineering, June 2000;91-101
- [53] 杨红. 适应性软件体系结构评价方法研究[D]. 大连:大连理工大学, 2007
- [54] 李刚, 金茂忠. 适应性软件体系结构研究[J]. 计算机科学, 2002, 29(2):90-93
- [55] Jun H, Colman A. The Four Major Challenges of Engineering Adaptive Software Architectures[C]//31st Annual International Computer Software and Applications Conference (COMPSAC), 2007;565-572

(上接第98页)

结束语 本文重点研究如何将 workflow 建模和分析的形式化模型 WF-net 转换为 PNML, 以便解决 WF-net 和其他 Petri 网模型的互操作问题。

本文首先在 PNML 元模型的基础上, 针对 WF-net 中的一些特殊元素, 如控制流构造模块和变迁触发器等, 提出了扩展的元模型, 然后基于扩展的元模型, 定义了 WF-net 模型元素到 PNML 的转换规则。最后, 在基于 Java 语言的 Eclipse 平台和通用交换格式 XML 的基础上, 设计并实现了一个支持 workflow 网编辑和 PNML 格式存储的软件工具, 取得了较好的效果, 从而证明了提出的 WF-net 的 PNML 元模型的正确性和转换规则的可行性。

本文为 Petri 网标准交换格式的制定做了有力的探索。进一步的工作是将工具做得更完善, 研究工作可以侧重通用的 Petri 网到 PNML 的转换方法。

参考文献

- [1] Matthias J, Ekkart K, Michael W. The Petri Net Markup Lan-

- guage [EB/OL]. http://www.informaditk.hu-berlin.de/top/pnml/download/JKW_PNML.ps, 2000
- [2] 周必水, 胡伟军. Petri net 可扩展性标记语言[J]. 系统仿真学报, 2003,15(8):49-52
- [3] Jonathan B, Soren C. The Petri Net Markup Language: Concepts, Technology, and Tools[C]//Proceedings in Applications and Theory of Petri Nets 2003; 24th International Conference, 2003. Berlin: Springer-Verlag, 2003;1023-1024
- [4] ISO/IEC/JTC1/SC7. ISO/IEC 15909-2 Software and Systems Engineering-High-level Petri Nets Part 2: Transfer Format International Standard[S]. 2009
- [5] 赵文, 袁崇义, 张世琨, 等. 一种模型驱动的工作流过程定义途径[J]. 计算机科学, 2006,33(12):10-15
- [6] Aalst W M P. The application of Petri nets to workflow management [J]. The Journal of Circuits, Systems and Computers, 1998,8(1):21-66
- [7] Thomas F, Andreas E. Workflow Petri Net Designer [EB/OL]. <http://www.woped.org>