

基于 Petri 网化简技术的工作过程模型结构验证

徐晶明, 杜宝珠

(北京理工大学 研究生学院, 北京 100081)

摘 要:目前, workflow 系统向大型化发展, 这使得基于可达图的验证技术在对大型模型进行验证时面临着状态空间爆炸的问题。因此, 在过程验证之前, 对大型模型进行化简是必要的。文中介绍两种化简规则。这些规则将一个大的 Petri 网化简为更小的 Petri 网, 同时保持合理性属性, 保证化简后的 Petri 网和原有的 Petri 网具有相同的属性。介绍了 Petri 网、Workflow Petri 网和过程合理性定义; 讨论了针对 Petri 网的两种化简技术; 提出了工作流过程模型结构合理性验证过程。

关键词: Petri 网; 工作流; 合理性; 模型化简

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2009)06-0051-04

Workflow Process Model Structure Verification Based on Petri Net Reduction Techniques

XU Jing-ming, DU Bao-zhu

(Graduate School of Beijing Institute of Technology, Beijing 100081, China)

Abstract: Now, the workflow system develop to large size. As a result, when verification for a large size model, the verification techniques of based on reachability graph face with the problem of state space explode. In that way, before process verification, it is necessary to reduce a large size net. In this paper, introduce two types reduction rules. These rules can reduce a large petri net to more smaller petri net, and maintain a soundness attributes, guarantee reduced petri net have the same attributes with original petri net. Introducing petri net, workflow petri net and process soundness definition; discussing two reduction methods for petri net; giving a procedure of structure soundness verification for workflow process model.

Key words: petri net; workflow; soundness; model reduction

0 引言

工作流过程定义中出现的错误可分为语法错、结构错和语义错^[1]。相应地, 有语法验证、结构验证和语义验证^[2]。语法验证检查过程定义是否符合特定描述语言的语法; 结构验证检查过程定义是否会导致错误的执行; 而语义验证是层次最高、最完整的验证, 保证过程定义与实际业务过程的目标一致性。

文中讨论的是工作流过程模型的结构合理性验证。Petri 网有它自己的语法定义和图形表示方法。许多年来人们已经在 Petri 网的分析和它的属性上作了许多研究工作。文中重点研究的属性是合理性。合理性在 Workflow Petri 网上的定义是^[3]: 对应于源库所的每一个标记, 最终会有且仅有一个标记出现在汇

结库所中; 当汇集库所中出现标记时, 其它所有的库所都是空的; 对每个变迁, 从初始状态都能够到达该变迁就绪状态。

但是, 现在常用的基于可达图的验证技术面临着状态空间爆炸^[4]的问题^[5]。因此, 需要化简 Petri 网^[6], 并且保证化简后的 Petri 网和原有的 Petri 网具有相同的属性。文中介绍两种化简规则。这些规则将一个大的 Petri 网化简为更小的 Petri 网, 同时保持合理性属性。

1 Petri nets

这里对在文中将要用到的 Petri 及 Petri 网子类作一些概念上的介绍。

Petri 网模型是一个基于图的数学形式化描述模型, 用来描述并发系统。Petri 网是一个双边有向图, 用库所(用圆圈表示)和变迁(用矩形表示)作为节点来表示。Petri 网的状态用库所中标记(用黑点表示)的分布来描述。一个变迁的实施条件是在它的所有输入

收稿日期: 2008-09-01; 修回日期: 2009-02-20

基金项目: 内蒙古自然科学基金(200607010810)

作者简介: 徐晶明(1981-), 男, 青海西宁人, 硕士研究生, 研究方向为工作流技术及数据集成; 杜宝珠, 硕士, 研究方向为工作流技术及数据集成。

库所中至少有一个标记,一旦一个变迁被实施,就从它的输入库所中取走一个标记,并往它的输出库所中增加一个标记。

定义 1: Petri 网是一个三元组 (P, T, F) :

- * P 是有限个库所的集合;
- * T 是有限个变迁的集合 $(P \cap T = \emptyset)$;

而且

- * $F(P \times T) \cup (T \times P)$ 是弧的集合。

定义 2: 自由选择 Petri 网。一个 Petri 网 $N = (T, P, F)$ 是自由选择 Petri 网, 如果:

$$\forall t_1, t_2 \in T: \cdot t_1 \cap \cdot t_2 \neq \emptyset \Rightarrow \cdot t_1 = \cdot t_2$$

定义 3: Workflow Petri 网。一个 Petri 网 $N = (T, P, F)$ 是 Workflow Petri 网, 如果:

1. 存在一个源库所 $i \in P$, 使得 $\cdot i = \emptyset$;
2. 存在一个汇结库所 $o \in P$, 使得 $o \cdot = \emptyset$; 而且
3. 每一个节点 $x \in P \cup T$ 都位于从 i 到 o 的一条路径上。

定义 4: 合理性。由一个 Workflow Petri 网 $N = (T, P, F)$ 建模的过程是合理的, 当且仅当:

1. 对于每一个从状态 i 可达的状态 M , 存在一个实施序列, 从状态 M 通往状态 o , 形式化表示为:

$$\forall M (i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} o);$$

2. 状态 o 是从状态 i 可达的唯一最终状态, 且结束时其中至少会有一个标记, 形式化表示为:

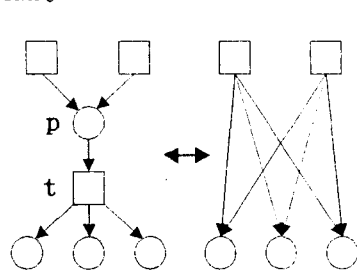
$$\forall M (i \xrightarrow{*} M \wedge M \geq o) \Rightarrow (M = o);$$

3. 在 (N, i) 中没有死变迁, 形式化表示为:

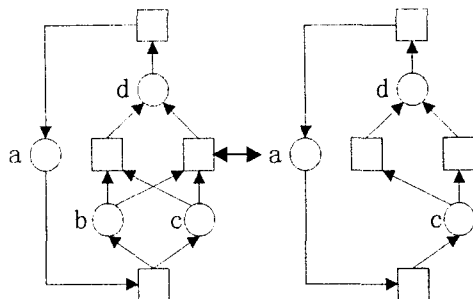
$$\forall t \in T \exists M, M' i \xrightarrow{*} M \xrightarrow{t} M'.$$

2 化简技术

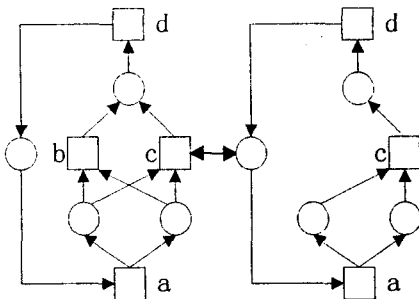
下面讨论的所有化简技术都保持原有网的属性: 活性、有界性(活性和有界性等价于合理性^[6])。首先介绍针对自由选择 Petri 网的化简技术 Desel 和 Esparza, 最后介绍可以针对任何 Petri 网的化简技术 Murata。



抽象规则



线性相关库所规则



线性相关变迁规则

图 1 Desel 和 Esparza 定义三条化简规则

2.1 Desel 和 Esparza 化简技术

Desel 和 Esparza 定义了三条针对自由选择 Petri 网的化简规则^[7]: 抽象规则、线性相关库所规则、线性相关变迁规则^[8], 分别简称为 #A、#P、#T。如图 1 所示。

2.2 Murata 化简技术

Murata 定义了六条针对任何 Petri 网的化简规则^[9]: 合并串联库所、合并串联变迁、合并平行库所、合并平行变迁、消除自环路库所、消除自环路变迁, 分别简称为 M1、M2、M3、M4、M5、M6。如图 2 所示。

以上就是介绍的化简技术, 但是从上面的定义可以得出这两个化简规则的特点:

① Murata 的合并串联库所和合并串联变迁规则是 Desel 和 Esparza 的抽象规则的进一步细化; Murata 的合并平行库所规则是 Desel 和 Esparza 的线性相关库所规则的进一步细化; Murata 的合并平行变迁规则是 Desel 和 Esparza 的线性相关变迁规则的进一步细化。

② 如果满足 Murata 化简条件, 那么一定满足 Desel 和 Esparza 化简条件; 如果满足 Desel 和 Esparza 化简条件, 不一定满足 Murata 化简条件。

③ 如果满足 Murata 化简条件, 那么节点和节点之间的路径一定小于等于满足 Desel 和 Esparza 化简条件的节点之间的路径。这就使得任何一个大规模的网, 化简之前优先考虑 Murata 化简规则。

根据上面总结的特点, 提出一套自己的验证过程。

3 验证过程

本节介绍如何用前面讲到的 Petri 化简技术来分析 Workflow Petri 网的合理性。如图 3 所示的主过程图阐述了这个过程。首先检查给定的 Petri 网是不是 Workflow Petri 网(用定义), 不是的话结束分析; 否则开始分析合理性, 图 3 的子过程图阐述了这个过程。

步骤 1: 检查是否可以用 Murata 化简技术来化简, 如果不可以跳到步骤 3, 如果可以, 用 Murata 化简技术进行化简, 直到不能化简为止;

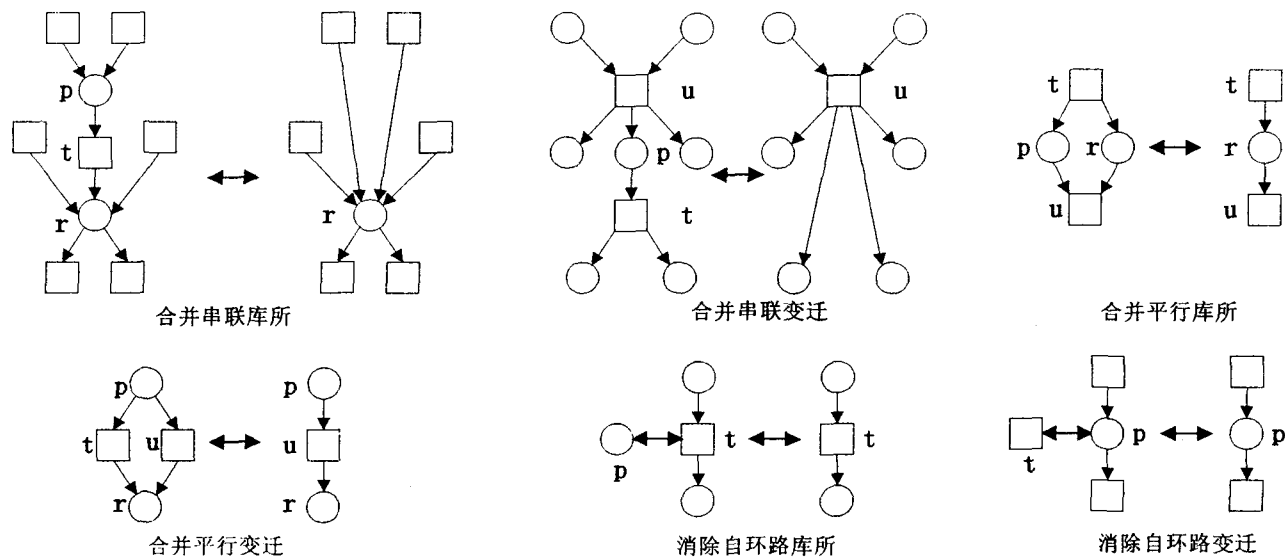


图 2 Murata 定义的六条化简规则

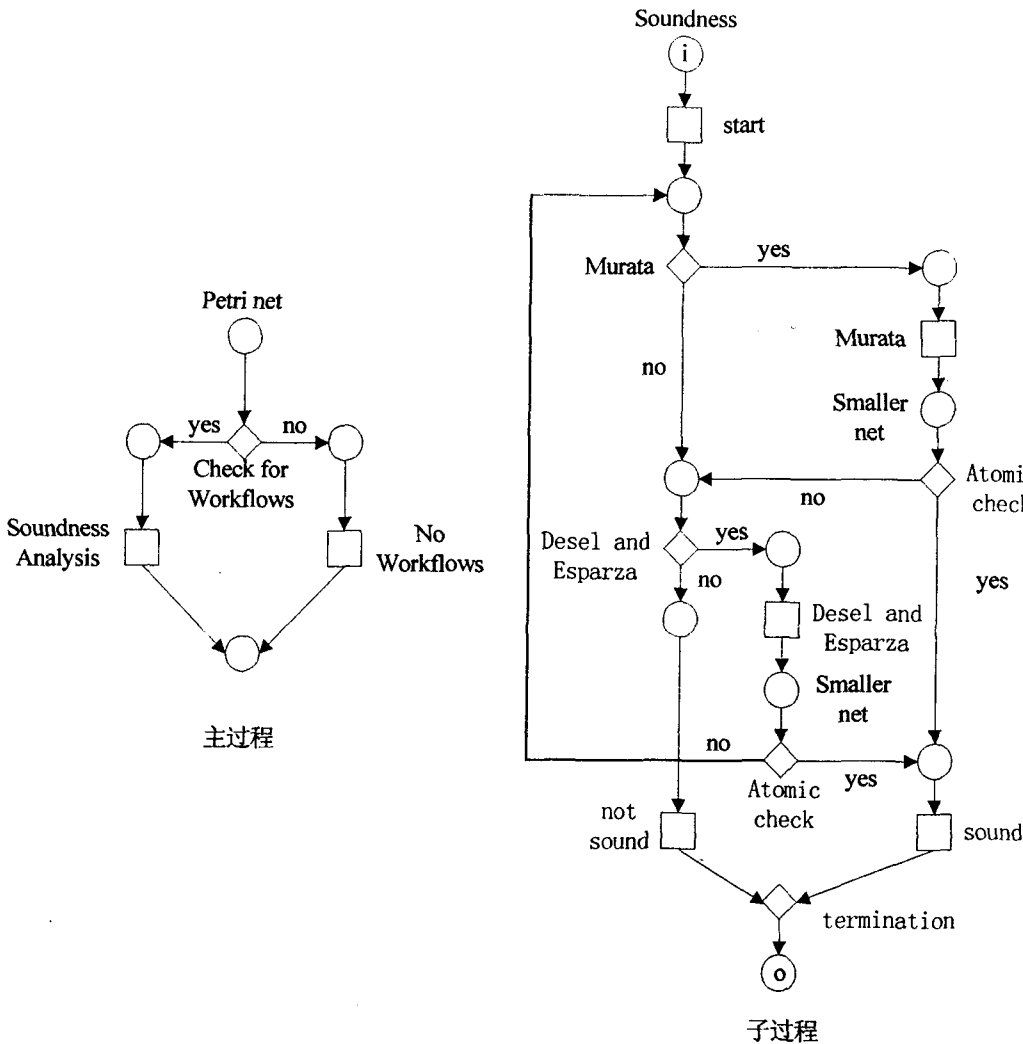


图 3 Petri 网合理性分析主过程和
Petri 网合理性分析子过程

步骤 2:判断步骤 1 得到的网是不是原子网($|T| = 1$ 或 $|P| = 1$),如果是跳到步骤 5,如果不是到步

骤 3;

步骤 3:检查是否可以用 Desel 和 Esparza 化简技术来化简,如果不可以跳到步骤 6,如果可以,用 Desel 和 Esparza 化简技术来化简,直到不能化简为止;

步骤 4:检查步骤 3 得到的网是不是原子网,如果是到步骤 5,如果不是跳到步骤 1;

步骤 5: Workflow Petri 网合理;

步骤 6: Workflow Petri 网不合理。

下面用一个实例来描述化简过程。图 4(a)是文献[10]中图 9 的模型映射出来的 Workflow Petri 网。用上面的化简步骤对其进行化简,具体化简过程

如图 4 所示。

化简 1:图 4(a)中标记为 M1、M2 的节点分别满足

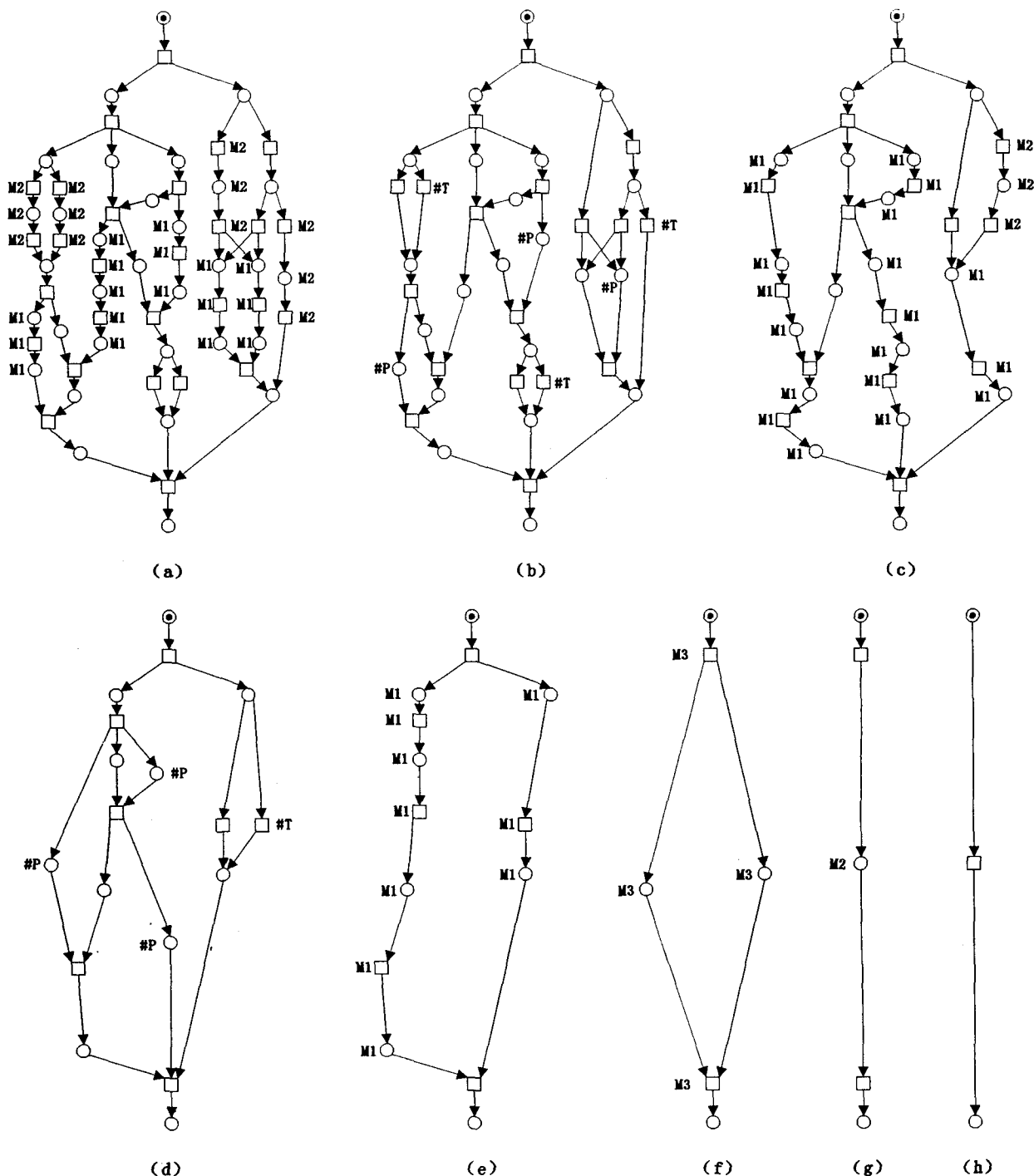


图 4 实例化简过程

M1、M2 的化简条件,因此,应用化简规则 M1、M2 对图(a)进行化简,结果为图(b)。图(b)不是原子网。

化简 2:图 4(b)中标记为 #P、#T 的节点分别满足 #P、#T 的化简条件,因此,应用化简规则 #P、#T 对图(b)进行化简,结果为图(c)。图(c)不是原子网。

化简 3:图 4(c)中标记为 M1、M2 的节点分别满足 M1、M2 的化简条件,因此,应用化简规则 M1、M2 对图(c)进行化简,结果为图(d)。图(d)不是原子网。

化简 4:图 4(d)中标记为 #P、#T 的节点分别满

足 #P、#T 的化简条件,因此,应用化简规则 #P、#T 对图(d)进行化简,结果为图(e)。图(e)不是原子网。

化简 5:图 4(e)中标记为 M1、M2 的节点分别满足 M1、M2 的化简条件,因此,应用化简规则 M1、M2 对图(e)进行化简,结果为图(f)。图(f)不是原子网。

化简 6:图 4(f)用 M3 化简,得到图(g),图(g)用 M2 化简得到图(h)。图(h)是原子网。

图 4(a)是文献[10]中图 9 的模型映射出来的一个

(下转第 57 页)

ArrayList 存放若干个 Strategy 角色,处理者调用 `handleRequest()` 方法时,该方法通过遍历集合中的 Strategy 角色,并委托该 Strategy 角色调用 `handleRequest()`,并判断 Strategy 角色的 `isHandle()` 方法返回的值是否为 false,如果是 false 就将用户的请求传递给下一个处理者,即执行 `getNextHandler().handleRequest()`,并结束 `handleRequest()` 方法。如果 Strategy 角色的 `isHandle()` 方法返回的值是 true,标明具体处理者选择了当前策略处理了用户的请求,不再将用户的请求传递给下一个处理者,并结束 `handleRequest()` 方法。

5 结束语

* 复合模式的优点:将策略模式嵌入到责任链模式中形成一种复合模式,从依赖的角度看,具体处理者和策略角色是组合依赖关系,当系统增加新的具体策略时,具体处理者不需修改代码即可使用新的具体策略。复合模式不仅保留了责任链模式的优点,而且增强了责任链模式处理用户请求的能力,使得责任链具有更好的灵活性,非常适合应用于动态响应用户的请求。责任链在维护和扩展方面具有很好的灵活性,因为系统不需修改其它处理者的代码就可以随时为每个处理者增减策略。

(上接第 54 页)

无结构冲突的 Workflow Petri 网,因此,用上面的化简过程得到一个原子网,说明此 Workflow Petri 网合理。

4 结束语

介绍了 Petri 网及 Petri 的两个子类,并且介绍了两种化简技术,最后结合这两种化简技术对 Workflow Petri 网进行化简。把原来的大网化简为更小的网为止,同时这里讲到的两种化简技术都保持原有网的属性。这样就解决了现有的分析技术存在的库所和变迁膨胀问题,提高对 Petri 网的分析效率。

参考文献:

- [1] Zhou Jiantao, Shi Meilin, Ye Xinming. Formal Verification Techniques in Workflow Process Modeling[J]. Journal of Computer Research and Development, 2005, 42(1): 1-9.
- [2] 周建涛,史美林,叶新铭.一种基于 Petri 网化简的工作流过程语义验证方法[J]. 软件学报, 2005, 16(7): 23-32.
- [3] van der Aalst W M P. Verification of Workflow nets[M]//In Azema P, Balbo G. Application and Theory of Petri nets 1997, volume 1248 of Lecture Notes in Computer Science.

* 复合模式的适用性:需要设计多个对象来处理用户的请求,哪个对象处理该请求运行时刻自动确定,并为每个处理请求的对象动态地配置多个算法。

参考文献:

- [1] Erich G, Helm R, Johnsn R. Design Pattern Elements of Reusable Object - Oriented Software[M]. Reading, Mass: Addison - Wesley, 1995: 115-128.
- [2] Martin R, Riehle D, Buschmann F. Pattern Languages of Program Design 3[M]. Reading, Mass: Addison - Wesley, 1998: 77-96.
- [3] Harrison N, Foote B, Rohnert H. Pattern Languages of Program Design 4[M]. Mass: Addison - Wesley, 1998: 151-172.
- [4] Cooper J W. Java Design Patterns[M]. 北京:清华大学出版社, 2003: 103-115.
- [5] 薛振伟,吴志杰.模型驱动的软件开发模式研究[J]. 计算机技术与发展, 2008, 18(2): 24-26.
- [6] 徐正权,但蕾蕾.基于 Java 源码的设计模式探测初探[J]. 计算机工程与科学, 2005, 27(1): 99-101.
- [7] 顾明.基于软件设计模式的 GIS 中间件描述[J]. 计算机应用与软件, 2008, 25(3): 86-88.
- [8] 杨年华,张礼平. Java 类库中的策略模式[J]. 计算机应用与软件, 2004, 21(1): 20-22.

Berlin:Spring-verlag, 1997:407-426.

- [4] 孙建召,曾巧明.基于面向对象 Petri 网的工作流建模及性能分析[J]. 计算机技术与发展, 2007, 17(10): 73-75.
- [5] Zhou Conghua, Chen Zheyu. Model Checking Workflow Net Based on Petri Net[J]. Wuhan University Journal of Natural Sciences, 2006, 11(5): 2-16.
- [6] 秦凯,姜浩.一种基于 Petri 网的工作流模型分解方法[J]. 计算机技术与发展, 2008, 18(1): 97-100.
- [7] Desel J, Esparza I. Free choice Petri Nets[M]//volume 40 of Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press, 1995.
- [8] Cheng A, Esparza J, Palsberg J. Complexity results for 1-safe nets[C]//In Shyamasundar R K. Foundations of software technology and theoretical computer science, volume 761 of Lecture Notes in Computer Science. Berlin: Springer - Verlag, 1993: 326-337.
- [9] Murata T. Petri Nets: Properties, Analysis and Applications[J]. Proceedings of the IEEE, 1989, 77(4): 541-580.
- [10] Sadiq W, Orlawska M E. Analyzing Process Model Using Graph Reduction Techniques[J]. Information systems, 2000, 25(2): 117-134.