

基于 Petri 网的软件项目管理建模方法

祁方民¹, 鱼滨², 史立军¹, 牟力科¹

(1.西北大学信息科学与技术学院, 陕西 西安 710127; 2.西安电子科技大学计算机学院, 陕西 西安 710071)



摘要:一种新的对于 Petri 网的拓展方法—库所拓展提出, 并在以软件项目管理过程为例分析建模对象的动态特性的基础上, 说明了库所拓展 Petri 网在描述过程管理方面的特点和优势。同时分析了库所拓展 Petri 网如何迎合建模对象动态性特点及解决由此带来的设计管理工具难等问题, 以尽量达到最优的设计目的。

关键词: Petri 网; 建模; 库所扩展; 软件项目管理

中图分类号: TP311.5

文献标识码: A

文章编号: 1004-731X (2007) S1-0075-04

A Modeling Method of Software Project Management Based on Petri Nets

QI Fang-min¹, YU Bin², SHI Li-jun¹, MOU Li-ke¹

(1.Northwest University School of Information Science and Technology, Xi'an 710127, China;

2.Xidian University School of Computer Science & Technology, Xi'an 710071, China)

Abstract: A new Petri nets modeling method —Place expanding Petri nets is given. This tool has the dynamic modeling characteristics which are suitable to aid managing the project management process, and the software project management process as an example to introduce the features and advantages of Place expanding Petri nets in the management process modeling. Place expanding Petri nets can solve the question about how to build the management tools to the dynamic process which is difficult for traditional tools so as to achieve the optimal design purpose. This method is very common, for example, in software project management, and other aspects of modeling.

Key words: Petri nets; model; place expanding; software project management

引言

Petri Nets (PN) 既有严格的数学表述方式, 也有直观的图形表示^[3], 是可应用于多种系统和领域的图形以及数学建模工具。适合于描述异步的、并发的计算机系统模型, 可对离散事件系统的控制流并发特性和异步行为分析系统的异步并发冲突问题进行描述^[7]。在实际的应用过程中, 由于所建模对象的动态变化特性, 使得传统Petri网对过程的描述能力不是很得心应手。例如对软件项目管理过程的建模: 软件项目开发是一个动态的过程, 对一个软件项目的管理也是一个动态的过程, 具有不可预知性; 而传统的Petri网在建模过程中对这些项目开发过程中的动态因素不能很好地控制或分析。

本文在概述分析软件项目管理过程动态特点的基础上, 提出了利用库所扩展概念下的Petri网对软件项目管理过程建模, 来满足动态的管理过程特点。

1 软件项目管理概述

早期软件应用需求范围较小, 需求简单, 质量问题不突出, 只要求程序能够正常运行; 随着用户对软件和软件系统

的功能要求日益增长, 软件设计复杂, 软件的质量问题开始凸现; 而软件项目管理者任务是要协调有限的资源和经费, 开发出高质量软件。所以, 软件项目管理过程成功与否将关系到整个项目的最终结果。

软件项目管理的提出是在20世纪70年代中期的美国, 是为了使软件项目能够按照预定成本、进度、质量顺利完成, 而对成本、人员、进度、质量、风险等进行分析和管理的活动。软件项目管理和其它项目管理相比有一定的特殊性: 软件项目管理将开发人员的个人开发能力转化成整个开发团队的开发能力; 软件是纯知识产品, 其开发进度和质量很难估计和度量, 生产效率也难以预测和保证; 软件开发相对不需要使用大量物质资源, 主要是人力资源, 其开发的产品主要是程序代码和技术文件; 软件系统的复杂性也导致了开发过程中各种风险难以预见和控制。

归纳可以发现: 软件项目的管理是一个动态过程, 具有不可预知性, 但是可以预测。软件项目管理是决定软件产品质量的主要因素, 在软件项目开发过程中, 管理日益受到人们关注, 管理是开发出好的软件的前提条件, 而好的管理离不开好的工具。

2 库所拓展 Petri 网在软件项目管理过程中的应用

2.1 Petri网的库所拓展定义

软件项目管理人员通常要完成的工作有: 组织、培训和

收稿日期: 2007-05-26

修回日期: 2007-07-02

作者简介: 祁方民(1979-), 男, 青海省平安县人, 硕士生, 研究方向为计算机软件与理论, 形式化方法; 鱼滨(1964-), 男, 陕西人, 副教授, 研究方向为(主要从事软件构件技术)分布式计算和多媒体技术; 史立军, 男, 硕士生, 研究方向为软件项目管理与软件度量。

激励职员;对工程规划、预算和调度;发现和解决资源冲突;持续监测整个项目,并且,在新的冲突出现时能采取正确的策略。在这样一个动态过程中,要应对在项目的执行的过程中出现的各种问题。现在很多项目管理工具可以协助完成不同功能,一般来说,这些工具包括项目调度、资源管理、项目跟踪、项目申报等职能。但是,当各种资源动态因素施加到项目的环境中时,其中一些工具就不能工作了^[1]。

Carl K. Chang 在 A net practice for software project management 中定义了一种新工具—PM-Net 来完成对其他工具所存在问题的改进,PM-Net 来源于 Petri 网,同时又添加了新的内容。事实上,可以以 Petri 网为基础,在不改变其结构的基础上辅助软件项目的管理过程。

一般系统的模型均由两类元素构成:表示状态的元素和表示变化的元素,Petri 网的状态元素和变化元素分别称为 S 元素和 T_元素,也简称为 S_元和 T_元^[3]。

下面首先给出 Petri 网 N 的定义,在此基础上给出适合软件项目管理的 Petri 网的库所拓展定义:

定义1:三元组 $N=(S, T, F)$ 为 Petri 网的充分必要条件^[3]:

- (1) $S \cap T = \emptyset$;
- (2) $S \cup T \neq \emptyset$;
- (3) $F \subseteq S \times T \cup T \times S$ (“ \times ”为笛卡儿积);
- (4) $\text{dom}(F) \cup \text{cod}(F) = S \cup T$, 其中, $\text{dom}(F) = \{x | \exists y: (x, y) \in F\}$ 和 $\text{cod}(F) = \{y | \exists x: (x, y) \in F\}$ 分别是 F 的定义域和值域

其中 S 和 T 分别被称为 N 的库所 (place) 集和变迁 (transition) 集。F 为流关系 (flow relation); 库所集和变迁集是有向网的基本成分,流关系是从中构造出来的,而连接 S 和 T 中元素的叫做弧 (arc), 弧是 F 中的元素,所以弧只能连接库所和变迁,不能连接两个库所或者是两个变迁。

库所拓展 Petri 网的定义:

定义2:三元组 $N=(S, T, F)$ 为库所拓展 Petri 网的充分必要条件是:

- (1) $S=\{s | s \text{ 是一个 } S \text{ 拓展后的 Petri 网或者 } s \subseteq S', S' \text{ 是 Petri 网 } N \text{ 的库所集}\}$
- (2) $S \cap T = \emptyset$;
- (3) $S \cup T \neq \emptyset$;
- (4) $F \subseteq S \times T \cup T \times S$ (“ \times ”为笛卡儿积);
- (5) $\text{dom}(F) \cup \text{cod}(F) = S \cup T$, 其中, $\text{dom}(F) = \{x | \exists y: (x, y) \in F\}$ 和 $\text{cod}(F) = \{y | \exists x: (x, y) \in F\}$ 分别是 F 的定义域和值域;
- (6) $\forall (x, y) \in F: x \in S \wedge y \in T \Rightarrow \exists z \in S, \text{使得 } (y, z) \in F$
- (7) $\forall (x, y) \in F: x \in T \wedge y \in S \Rightarrow \exists z \in S, \text{使得 } (z, x) \in F$

2.2 库所拓展后的 Petri 网元素

2.2.1 库所

通常,通过圆圈或椭圆来表示库所,每个库所代表一类

资源。为了便于利用 Petri 网对软件项目进行管理过程建模,我们将 Petri 网中的库所进行拓展,这里谈到的库所是一个递归的概念:库所可以是一个传统 Petri 网的库所来表示某一类资源,也可以是一个小库所拓展后的 Petri 网,这样,在建模过程中,可以将某个子项目或是某个模块作为库所来对待,既可以体现该子项目或模块在整个软件项目中的作用,也可以给软件管理者一种模块化视角,来加强对整个软件项目的管理和宏观规划。

在定义2中,条件(1)说明在 Petri 网的库所拓展中,库所的来源不仅仅是传统 Petri 网中的库所,同时也可以是一个小的库所拓展的 Petri 网。条件(6)是在 Petri 网的定义基础上对库所拓展的 Petri 网的限制,它的存在是必要的,因为按照 Petri 网 N 的定义,如图 1-a 是一个 Petri 网,假设存在库所拓展的 Petri 网 Σ' (如图 1-b), s_1 是 Σ' 的一个库所,令 $s_1 = \Sigma$,则将上述库所拓展的 Petri 网扩展开来可以得到扩展后的 Petri 网 Σ' (如图 1-c) 所示:弧 a_1 连接的是两个变迁,不符合 Petri 网的定义。由于在定义中,我们只谈到将库所集的定义可以延展为库所和子 Petri 网的集合,而库所的连接对象最多只有变迁,所以我们通过条件 (6) 对库所拓展的 Petri 网库所的概念进行加强。同时,我们也会注意到在图 2 中,如果令 $s_2 = \Sigma$,存在图 2-c 情况。

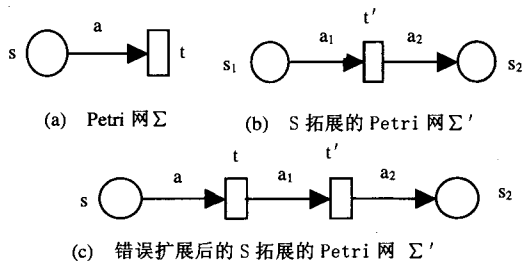


图 1 S 拓展的 Petri 网 Σ' 在 Petri 网 Σ 上的扩展方式一

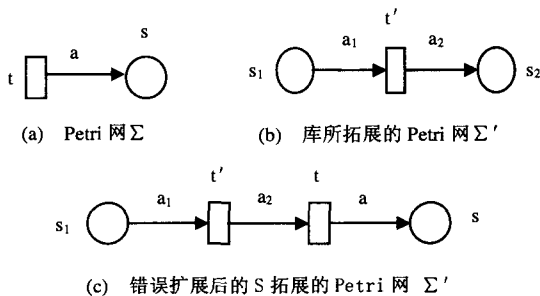


图 2 S 拓展的 Petri 网 Σ' 在 Petri 网 Σ 上的扩展方式二

所以,在库所拓展的 Petri 网的定义过程中,条件(6)、(7)的存在是必要的。

在软件项目管理过程中,软件项目管理者不可能拥有无限制的资源,同样需要诸如人力,物力和财力等等方面的约束信息。所以,在库所拓展的 Petri 网中,将所需资源的信息作为一个或者一类库所存在,将它们和加工、使用这部分资源的变迁通过弧来连接,以体现软件项目管理过程中外界环境的作用。例如在图 3 中, a 表示车辆管理系统开发不受

外界因素干扰, 而 b 表示需要有相应的开发经验; 类似的, 在软件项目管理的其他环节, 可以将约束信息设置为库所, 通过库所中资源存在与否来控制它相关的变迁是否可以发生。

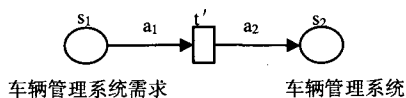
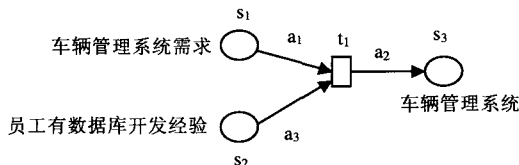
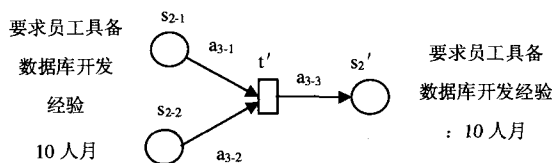
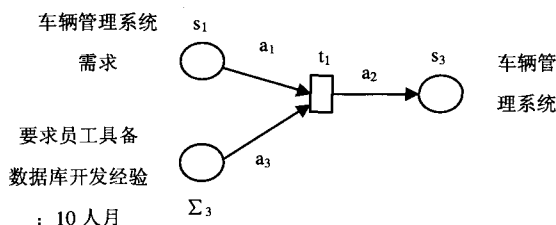
(a) S 拓展的 Petri 网 Σ_1 (b) 具有一个约束条件的 Petri 网 Σ_2 (c) 动态添加一个约束信息的 Petri 子网 Σ_3 (d) 动态添加一个约束条件后的 Petri 网 Σ_4 (令 $s_2 = \Sigma_3$)

图 3 动态添加约束信息

软件项目管理是一个动态过程, 其中很多因素在设计开始阶段没有办法得到和了解, 只有在项目实施过程中才能根据实际情况做出反应。传统的 Petri 网很难对这种动态“属性”进行考虑, 而这里给出的库所拓展的 Petri 网, 库所不限于表示资源信息, 同时是一个单独的小库所拓展的 Petri 网; 利用这条属性将新出现的外界的“动态信息”与已经存在的约束信息合并, 组成一个新的约束信息子 Petri 网, 来实现对整个管理过程的约束。例如在图 3(c)中, 如果在项目实施过程中还想加入工作量的信息, 则加入这条新的约束条件后 Petri 网可以描述为图 3(d)所示, 使得在图 3 中原本只有一个约束条件的 Petri 网 Σ 在 $s_2 = \Sigma_3$ 的条件下形成具有两个约束条件的 Petri 网 Σ_4 。

通常对软件项目管理者而言, 在项目开始规划阶段, 要从整体角度去看待整个工程, 而在库所拓展的 Petri 网的定义中可以发现, 利用库所拓展的 Petri 网中对库所的递归定义, 可以帮助软件管理者在项目实施开始阶段就将一部分暂时不用考虑的环节和资源等因素通过一个递归的库所来代替, 便于项目的宏观管理和规划。在定义这一类递归的库所时,

项目管理者不用关心该递归库所的内部运作过程, 但是此库所需要作为整个 Petri 网的一部分存在, 所以项目管理者在定义该类库所时, 必须要知道其输入信息和输出信息, 也就是这个递归的库所的托肯信息。

2.2.2 托肯

在 Petri 网 N 的定义中, 每个库所代表一种资源, 其中黑点表示该种资源的数量, 把黑点叫做托肯 (token) [3]。对于库所拓展的 Petri 网, 其中的库所可以是一个小的库所拓展的 Petri 网, 因此如果在库所拓展的 Petri 网中使用的库所, 表示某一种资源, 则该库所中的黑点代表的是该种资源的数量; 如果该库所代表一个小的库所拓展 Petri 网, 则该库所不能代表某类资源。按照对库所拓展的 Petri 网的定义, 库所拓展的 Petri 网要求每个变迁必须存在它的前驱库所和后继库所, 所以, 当整个一个库所拓展的 Petri 子网 Σ' 视为一个库所 s 置入到上一级的 Petri 网 Σ 中, 为了保证上一级 Petri 网的正确性, 要求库所 s 的托肯分为两类, 第一类可以称为输入托肯 (input token) 是所有库所拓展的 Petri 网 Σ' 中无前集的库所中的托肯的集合, 此时的托肯代表的是整个库所拓展的 Petri 网的子网 Σ' 开始工作时的初始情态; 第二类可以称为输出托肯 (output token) 是所有库所拓展的 Petri 网的子网 Σ' 中无后集的库所中的托肯的集合, 此时的托肯代表整个库所拓展的 Petri 网子网 Σ' 结束工作时的终结情态。在实际的库所拓展 Petri 网的应用中, 可以通过库所的打包将多个库所合并成一个新库所, 此新库所包含合并前参加合并的库所的托肯, 打包过程如图 4 所示:

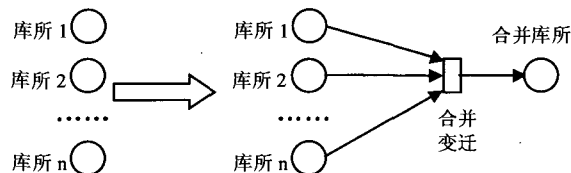


图 4 库所合并打包过程

库所打包后, 将合并库所的托肯称为托肯集, 其数量为 1, 集合中的元素是打包前的库所中的元素集合。经过上述的打包合并, 使得新库所的托肯值都为 1 (托肯集个数), 这样, 在以后的讨论中不再另外讨论库所中的托肯信息, 而且, 为了图形的简便, 我们也不对上述托肯在图形中标注, 默认每个库所都包含托肯集, 且个数为 1。

2.2.3 变迁

无关联的库所之间是通过变迁来连接的。同 Petri 网的概念相一致, 在库所拓展的 Petri 网中, 状态元素和变化元素分别用库所和变迁表示, 两者是平等的, 库所由变迁来改变, 而变迁由库所来描述, 两者相互依存 [3]。

2.2.4 弧

简单而言, 弧就是用来连接库所和变迁的, 弧只能由库所到变迁或从变迁到库所, 不能用来连接两个库所或两个变

迁。而且,根据在托肯集中的定义,以后库所拓展 Petri 网的图形描述中,每个变迁上传递的托肯集的数量都是 1,以后在不引起歧义的情况下对托肯集的数量不进行描述。

2.3 库所拓展的 Petri 网应用

从上述对库所拓展的 Petri 网的定义可以发现,库所拓展的 Petri 网同 Petri 网在本质上没有区别,建立在 Petri 网上的一些规则、性质、定理和公式以及判断的工具和手段都可以应用到库所拓展的 Petri 网上面来,为我们利用库所拓展 Petri 网去指导实际管理工作提供帮助。

2.3.1 已有软件项目管理工具的不足

目前的软件项目管理规划软件在项目管理过程方面存在很多不足和局限:1、通常受限只能关注项目的静态组成部分:对项目计划通常以静态的角度考虑,对在软件项目设计、开发的过程中出现的具备动态属性的信息考虑不周;2、不能很好的描述任务和资源之间的约束关系:针对项目内部或项目之间因竞争有限资源而引起的项目活动延迟等情况考虑不周;3、缺乏对软件项目的持续性管理:通常一个大的项目要延续很长的时间,在开始时,这些项目只能被简单地计划,后续步骤只能在第一步计划完成并产生结果后才能有比较详细的计划。因此,在项目计划时不同成果之间的因果依赖关系不能尽早取得。4、缺乏对软件项目的整体性控制。一个项目的规划一般是自顶向下,分成多层次具有一定次序的子任务,对每一级任务,都需要提供相应的输入、输出、资源、

时间等信息,而子任务信息的总合都要和上一级的任务的相关信息相匹配,通常的工具不容易处理这种情况。

2.3.2 库所拓展的 Petri 网在软件项目管理中的应用

对于目前软件项目管理工具存在的问题,可以利用库所拓展 Petri 网自身的属性来寻求解决的方法:库所拓展的 Petri 网可以利用库所的递归定义,当出现在原有项目计划之下有新的子项目或者动态条件信息加入时,可以将原来的库所由该库所与部分要添加的库所合并后形成一个新库所来代替,这个新库所就是一个库所拓展 Petri 网,这样,在软件项目规划阶段未能涉及的内容和细节,都可以利用库所拓展的 Petri 网的这个特点来动态的添加到原规划中,从而保证起始规划的有效性,并随时根据实际需要不断对新的过程进行持续加载和不断完善,以达到整个软件项目的要求。

库所拓展的 Petri 网在传统 Petri 网的概念基础之上,可以根据传统 Petri 网的概念来发现软件项目管理过程中变迁(相当于流程)的并发、顺序、冲突、冲撞和混感等问题。对于错误的构件过程,还可以根据传统 Petri 网的理论和手段,使其合理化和正确化。下面通过一个简单事例来简要描述库所拓展的 Petri 网控制软件项目的管理过程。

3 库所拓展的 Petri 网图形化软件项目管理描述举例

在图 5 中给出库所拓展的 Petri 网辅助软件项目管理初始规划图和进一步的细化图。

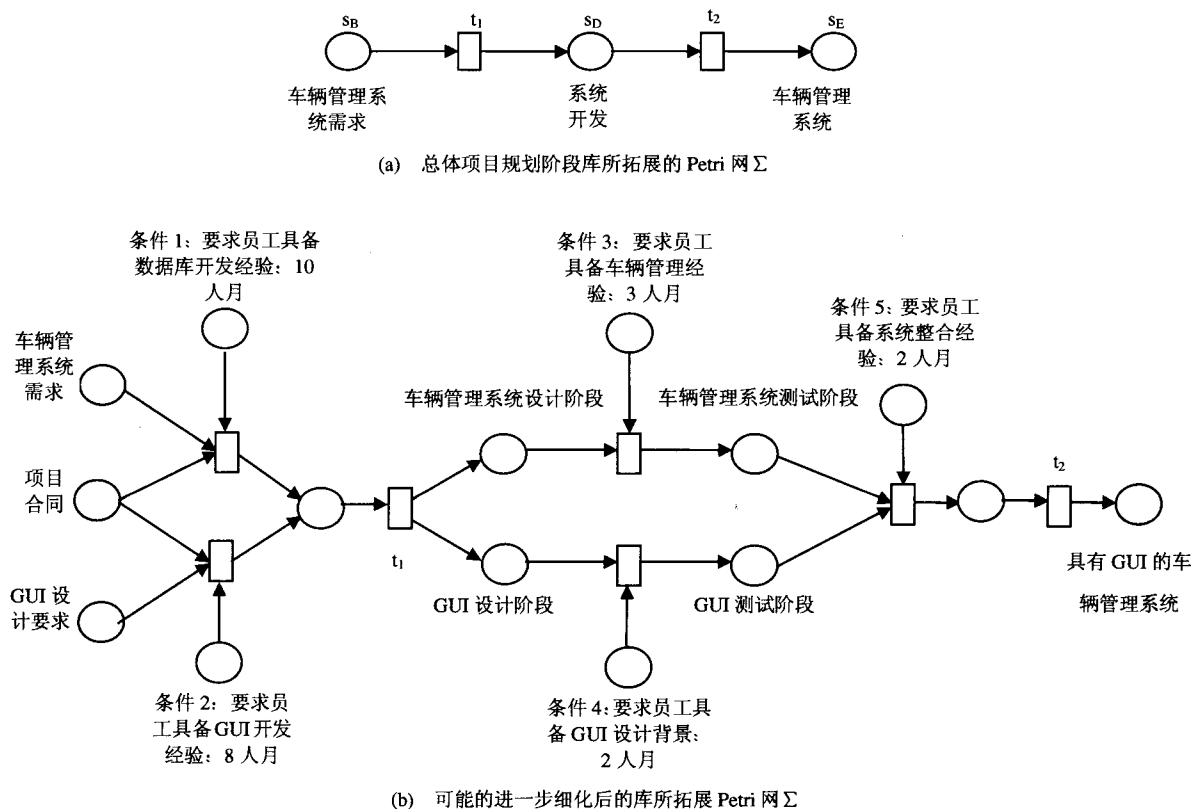


图 5 库所拓展的 Petri 网辅助设计举例

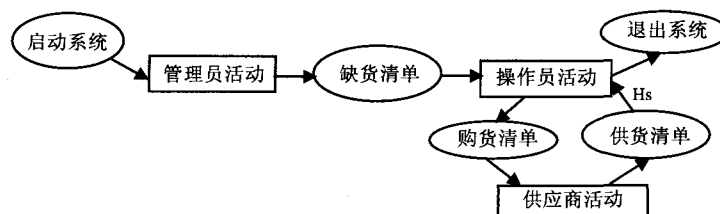


图 3 顶层 CPN 模型图

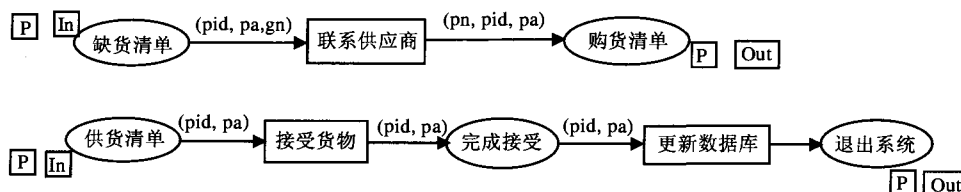


图 4 置换变迁操作员活动对应的 CPN 模型子页面

将顶层 CPN 模型中的置换变迁操作员活动建立相应的子页面, 见图 4。

若再给定 Petri 网初始标识后, 系统的静态模型(类图)就转化为由 CPN 表示的可执行模型, 并且可利用它来分析系统体系结构的模型的设计。

参考文献:

- [1] J L P. Petri Net Theory And The Modeling of Systems [Z]. 1980, 115-150.
- [2] Kurt Jensen. Coloured Petri Nets-Basic Concepts, Analysis Methods and Practical Use [Z]. 1980, 106-118.
- [3] 朱连章, 张红霞. 基于着色 Petri 网的电子商务工作流建模[J]. 中国石油大学学报, 2006, 30(4): 140-144.
- [4] 吴建, 等. UML 基础与 Rose 建模案例[M]. 北京: 人民邮电出版社, 2004.
- [5] 冯玲玲, 沈铁. 基于 UML 的需求分析与建模[J]. 科学技术与工程, 2005, 5(9): 605-606.
- [6] 曹晨. 基于 UML 的仓储管理系统的分析与设计[J]. 电子工程师, 2006, 32(6): 69-72.
- [7] YAN Zhi-jun, GAN Ren-chu. Formalization and Verification of Business Process Modeling Based on UML and Petri Nets [J]. Journal of Beijing Institute of Technology, 2005, 14(2): 212-216.
- [8] 田国会, 等. 基于层次着色 Petri 网的物资储备系统调度问题研究[J]. 机械工程学报, 2005, 41(4): 148-153.

(上接第 78 页)

4 结论

以上给出了基于 Petri 网的软件项目管理的新方法库所拓展的 Petri 网的设想, 该方法是在继承传统 Petri 网概念的基础上给出的, 同时继承了 Petri 网的优点, 对于动态软件项目管理提供了一定的帮助, 为今后软件项目的管理提供了一种可行的方法。Petri 网是一种具有严格的数学定义的形式化方法, 利用它拓展的库所拓展 Petri 网发挥了形式化方法的严谨的数学证明和推理的特点, 给项目的构件化管理提供了保障, 当然上述对软件项目管理建模的方法也可以运用到其他管理过程建模中。

对于库所拓展的 Petri 网还有一些更深入的内容将是作者今后在此方向上主要学习和继续研究的方向。

参考文献:

- [1] Carl K Chang, Christensen M. A net practice for software project management [J]. IEEE Software, 1999, 16(6): 80-88.
- [2] Frederick P Brooks, Jr. (Adams Wang translation). The MYTHICAL MAN-MONTH [M/CD]. Beijing: Tsinghua University Press, 2002.
- [3] 袁崇义. Petri 网原理与应用 [M]. 电子工业出版社, 2005.
- [4] 吴哲辉. Petri 网导论 [M]. 机械工业出版社, 2006.
- [5] Mauricio Varea, Bashir A I Hashimi. Dual Transition Petri Net Based Modeling Technique for Embedded Systems Specification [C]//Proceedings of Design, Automation and Test in Europe, 2001.
- [6] Z Peng, K Kuchcinski. Automated Transformation of Algorithms into Register-Transfer Level Implementations [C]// IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1994.
- [7] 郭军, 郝克刚. 基于双变迁 Petri 网的嵌入式系统建模 [J]. 系统仿真学报, 2005, 17(S1): 229-231.
- [8] Luis Alejandro C. Modeling and formal verification of embedded systems on a Petri net representation [J]. J. of Systems Architecture, 2003, 49: 571-589.
- [9] Spinner M P. Elements of project management: plan, schedule, and control [M]. 2nd ed. Englewood Cliff, NJ: Prentice Hall, 1992.
- [10] C ones. Software Management: The Weakest Link in the Software Engineering [J]. Computer, 1994, 27(5): 10-11.
- [11] 潘善亮, 陈刚, 董金祥. 支持并行规划和设计的项目管理模型研究 [J]. 计算机辅助设计与图形学学报, 2001, 13(8).
- [12] 徐立云, 张春慨, 邵惠鹤. 基于时延 Petri 网的项目管理方法 [J]. 系统工程理论方法应用, 2001, 10(1).