**南京大学本科生毕业设计中文摘要**

毕业论文题目： 基于Javascript的词法分析器自动生成工具——AliceLex的设计与实现

软件学院 院系 软件工程 专业 2012 级本科生姓名： 葛羽航

指导教师（姓名、职称）： 葛季栋， 讲师

摘要：

随着Web2.0的不断发展，尤其是HTML5的推动，

关键词：Lex，词法分析器自动生成工具，NFA，DFA，JavaScript，编译原理

**南京大学本科生毕业设计英文摘要**

THESIS： Study and Implement on A Workflow Modeling Tool based on PNML

DEPARTMENT： Software Institute

SPECIALIZATION: Software Engineering

UNDERGRADUATE: 2012

MENTOR: Instructor Ge Jidong

ABSTRACT：

w

KEY WORDS: Lex, Lexical Analyzer Generator, NFA, DFA, JavaScript, Compiler Principle

# 目 录

[目 录 IV](#_Toc263258548)

[**第一章 概述** 1](#_Toc263258549)

[1.1 项目背景 1](#_Toc263258550)

[1.2 国内外工作流建模技术研究现状 2](#_Toc263258551)

[1.2.1 底层文件存储技术 2](#_Toc263258552)

[1.2.2 仿真技术和分析技术 3](#_Toc263258553)

[1.3 论文主要工作 3](#_Toc263258554)

[1.4 论文组织结构 4](#_Toc263258555)

[**第二章 基于PNML的工作流建模工具相关技术概述** 5](#_Toc263258556)

[2.1 Petri网及相关理论 5](#_Toc263258557)

[2.1.1 Petri网基本定义 5](#_Toc263258558)

[2.1.2 Petri网动态性质 6](#_Toc263258559)

[2.2 工作流网(Workflow Net，WF-net) 7](#_Toc263258560)

[2.3 着色Petri网(Colored Petri Net, CP-Net) 8](#_Toc263258561)

[2.3.1 着色Petri网的静态结构 8](#_Toc263258562)

[2.3.2着色Petri网的动态行为 9](#_Toc263258563)

[2.4 Petri Net Kernel (PNK) 10](#_Toc263258564)

[2.5 Petri Net Markup Language (PNML) 11](#_Toc263258565)

[**第三章 PNML对着色Petri网性质支持的研究** 14](#_Toc263258566)

[3.1 基于PNML的着色Petri网扩展标准 14](#_Toc263258567)

[3.2着色Petri网的PNML文件结构 16](#_Toc263258568)

[**第四章基于PNML的工作流建模工具概述** 19](#_Toc263258569)

[4.1 系统需求设计 19](#_Toc263258570)

[4.1.1系统的功能需求设计 19](#_Toc263258571)

[4.1.2 系统的非功能需求设计 20](#_Toc263258572)

[4.2 系统概要设计 21](#_Toc263258573)

[4.3 系统包结构设计 22](#_Toc263258574)

[**第五章基于PNML的工作流建模工具详细设计及实现** 25](#_Toc263258575)

[5.1着色Petri网在PNK中的定义和实现 25](#_Toc263258576)

[5.2着色Petri网内存结构模块 26](#_Toc263258577)

[5.2.1pnk.kernel包 26](#_Toc263258578)

[5.2.2 pnk.netElementExtension包 28](#_Toc263258579)

[5.3着色Petri网仿真器模块 28](#_Toc263258580)

[5.3.1仿真器详细设计 30](#_Toc263258581)

[5.3.2 Firing Rule算法实现 31](#_Toc263258582)

[5.4文件输出和文件管理模块 31](#_Toc263258583)

[5.4.1文件输入和管理详细设计 31](#_Toc263258584)

[5.4.2自动布局算法详细设计和实现 32](#_Toc263258585)

[5.5 界面显示模块 34](#_Toc263258586)

[5.5.1 界面显示设计及流程图 34](#_Toc263258587)

[5.5.2 界面显示实现 35](#_Toc263258588)

[5.6 基于概率技术的数据流分析算法 36](#_Toc263258589)

[5.6.1 算法相关定义 37](#_Toc263258590)

[基于概率的数据流分析算法实现 37](#_Toc263258591)

[**第六章 系统与BeehiveZ系统集成设计实现** 39](#_Toc263258592)

[6.1 数据库更改和设计 39](#_Toc263258593)

[6.2 BeehiveZ系统界面更改和显示 41](#_Toc263258594)

[**第七章 总结与展望** 43](#_Toc263258595)

[7.1 总结 43](#_Toc263258596)

[7.2 基于PNML的工作流建模工具的局限性 43](#_Toc263258597)

[7.3 基于PNML的工作流建模工具的改进方向和发展前景 44](#_Toc263258598)

[参考文献 45](#_Toc263258599)

[致谢 47](#_Toc263258600)

# 第一章 引言

## 1.1 项目背景

随着

。

## 1.2 国内外lex相关技术研究现状

基于

：

### 1.2.1 词法

目前，

### 1.2.2 正则表达式

针对

包括：

1. 网
2. 。

对于。

### 1.2.3 DFA压缩

## 1.3 论文主要工作

本文的主要工作包括以下几个方面：

1. 介绍了Web2.0的发展现状，阐述了编译原理，尤其是词法分析在JavaScript上面的应用前景。
2. 介绍了编译原理的相关理论知识，词法、文法的相关概念。
3. 着重介绍了词法分析的的相关理论知识，包括正则表达式，NFA，DFA，以及相关的算法理论。
4. 介绍了词法分析器自动生成工具和其实现的难点要点，展现当前Lex工具的现状。
5. 详细介绍了基于JavaScript的词法分析自动生成工具AliceLex的设计与实现。
6. 展示了基于AliceLex的应用场景，包括Lex经典示例以及一款使用其作为词法分析核心的在线代码编辑工具。

## 1.4 论文组织结构

本文主要介绍了词法分析和词法分析自动生成工具的相关理论知识，以及一个基于JavaScript的Lex工具——AliceLex的设计和实现。主要是对词法分析相关的编译原理进行描述，同时描述一个基于JavaScript的Lex的设计和实现过程以及这个过程中出现的一些问题。

第一章：概述和前言部分，主要介绍了项目背景，当前词法分析和词法分析自动生成工具的研究及现状 ，并描述了该论文的主要工作。

第二章：主要介绍基于JavaScript的词法分析器自动生成工具的开发和实现过程中用到的相关理论知识和理论研究。

第三章：介绍了基于JavaScript的Lex工具与传统Lex工具相比较的特点和实现难点。

第四章：从需求分析和总体设计两个方面，描述基于JavaScript的Lex工具的提出背景和开发者信息，同时分析和总结出功能性和非功能性需求。

第五章：对AliceLex工具的详细设计和实现进行描述，重点包括Lex工具相关算法的设计和具休实现。

第六章：展示了经典Lex示例在AliceLex上的实现，同时引入一个基于AliceLex核心的在线代码编辑器。

第七章：总结该项目已实现的功能，探讨项目的缺点和不足，并指出该项目未来的扩展和发展方向

# 第二章 lex工具相关技术概述

## 2.1 词法分析及相关理论

### 2.1.1 Petri网基本定义

Petri网[1]是用来描述分布式系统的一种模型。Petri网的概念是1962年由德国科学家Carl Adam Petri在他的博士论文中首先提出来的。Petri网既有严格的数学表述方式，也有直观的图形表达方式，既有丰富的系统描述手段和系统行为分析技术，又为计算机科学提供坚实的概念基础。研究领域趋向认为Petri网是所有流程定义语言之母。它分经典Petri网理论和高级Petri网理论。

关于Petri网的基本定义，引文[2]中进行了详细描述。为研究Petri网的基本性质，我们需要如下关于Petri网的基本定义。

****定义2.1(Petri网)****

三元组称为有向网(Direct Net，简称Net)的充要条件是：

它们分别是F的定义域和值域。

其中S和T分别称为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，流关系是从它们构造出来的。每个库所代表一种资源，资源的流动由流关系规定，所以变迁只能与库所有直接的流关系。

****定义2.2（输入前集、后集）****

设

称为x的前集或输入集，为x的后集或输出集。

****定义2.3 （标志Petri网）****

设为一个网，映射称为网N的一个标识(marking)。四元组称为一个标志网。

用图形表示一个标志网时，对则表示库所s的小圆圈内有k个黑点，标识库所s有k个标志(token)。

****定义2.4 （初始标识Petri网系统）****

一个网系统是一个标志网并具有下面的变迁发生规则(transition firing rule):

1. 对于变迁如果：

则说变迁t在标识M有发生权(enabled)，记为M[t>。

1. 若M[t>，则在标识M下，变迁t可以发生(fire)，从标识M发生变迁t得到一个新的标识M’，对于，

一个网系统有一个初始标志(initial marking)。记为M0，它描述了被模拟系统的初始状态。

### 2.1.2 Petri网动态性质

使用Petri 网对实际系统进行建模，主要目的之一就是借助系统模型来分析实际系统的性质和功能。

Petri网的动态性质(Dynamic property)是指网系统在运行过程中的一些性质。这些性质同Petri网所模拟的实际系统运行过程中的某些方面的性质有密切关联。关于Petri网的动态性质，在引文[3]中有详细叙述。在本文的研究中，主要用到以下几种Petri网的动态性质：

****定义2.5 (可达性)****

设为一个Petri网，如果存在t∈T，使M[t>M’，则称M’为从M直接可达的。如果存在变迁序列t1, t2,….tk和标识序列M1,M2,…Mk, 使得M[t1>M1[t2>M2... Mk-1[tk>Mk, 则称Mk为从M直接可达的。从M可达的一切标识的集合记为R(M)。约定M∈R(M)。

用Petri网模拟一个实际系统时，以网(S,T;F)描述系统的结构，初始标识M0表示系统的初始状态，R(M0)给出系统运行过程中可能出现的全部状态的集合。

****定义2.6 (活性)****

设为一个Petri网，M0为初始标识，t∈T。如果对任意M∈R(M0)，都存在M’ ∈R(M)，使得M’[t>，则称变迁t是活的。如果每个t∈T都是活的，则称为活的Petri网。

****定义2.7(有界性，安全性)****

设为一个Petri网，s∈S。若存在正整数B，使得M∈R(M0)：M(S)B,则称库所s为有界的(bounded)，并称满足此条件的最小正整数B为库所的界，记为B(s)。即：

当B(s)=1时，称库所s为安全的。

设为一个Petri网，如果每个s∈S都是有界的，则称为有界Petri网。称：为的界。当时，称为安全的。

对于比较复杂的网系统，需要一些通用的分析技术分析网模型，以此来确定被模拟系统的性能。网论对Petri网已提出了多种分析方法，主要有可达标识图与可覆盖树，关联矩阵与状态方程，Petri网语言和Petri网进程等。

## 2.2 词法分析器自动生成

对工作流的控制流维度建模的Petri网被称为WF-net[4]，它定义同一类案例共同动态行为。工作流网由W.M.P. van der Aalst提出，是被广泛接受的描述过程控制流维度的Petri网，它的提出对Petri网的控制流研究作出了重大贡献。

****定义2.8(工作流网)****

Petri网是一个WF-net，当且仅当：

1. 存在一个源库所i∈S，使得;
2. 存在一个汇结库所o∈P，使得；而且
3. 每一个节点x∈PT都位于从i到o的一条路径上。

在控制流中，构造块(如AND-split、AND-join、OR-split、OR-join)被用来建模顺序、条件、并行和迭代路由。在工作流网中，一个库所对应着一个条件，能够用作任务的前条件和/或后条件。工作流网中常见的控制结构如下：

1. 顺序(Sequential)结构

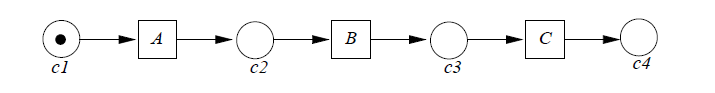


图2.1顺序结构

1. 并行(Parallel)结构

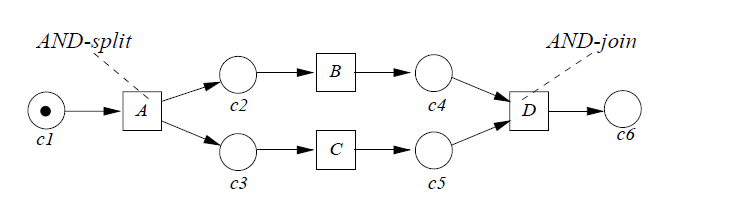


图2.2 并行结构

1. 选择(Excusive)结构

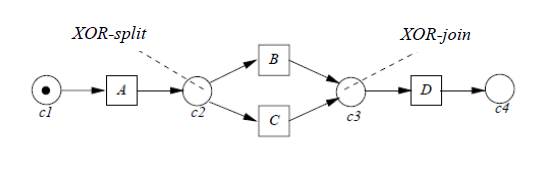


图2.3 选择结构

此外，Moe Wynn提出了Reset-Net，来处理WF-net无法形式化描述OR-split，OR-join，cancel region等高级控制流模式的问题，并将Reset-Net用在YAML工具中。但是，工作流网关注Petri网的控制流，对数据维度的支持不够。

## 2.3 着色Petri网(Colored Petri Net, CP-Net)

Jensen提出的着色Petri网[5]是对Petri网的著名扩展。着色Petri网是对Petri网中的标志着以不同的颜色。其实质就是对标志进行分类，以实现对网系统的折叠。颜色是与库所和变迁有关的，当变迁发生时，根据库所和变迁之间的弧上的标签，一定数量的托肯从进入库所中被拿走。

### 2.3.1 着色Petri网的静态结构

着色Petri网[6]涉及到多重集(Multi-Set)的定义和操作。多重集是集合理论的扩展，它与普通集合的不同之处是：多重集允许集合元素的多次出现，而普通集合中，每个元素只能出现零次或一次。

****定义(2.8) 多重集[****7****]****

在有限非空集S上的一个多重集b是一个函数：，其中N为非负整数集合。在S上的一个多重集b可以表示成：，其中b(s)表示s在多重集b中出现的次数。

在S上的所有多重集的集合表示为SMS。SMS= 。

****定义(2.9)****着色Petri网[8]

着色Petri网是一个九元组其中：

1. 是有限非空类型的集合，称为颜色集(Color Sets)；
2. 是有限的库所(Place)集合；
3. 是有限的变迁(Transition)集合；
4. 是有限的弧的集合，满足：；
5. 是节点的函数，定义；
6. 是颜色函数，定义；
7. 是警卫函数，是变迁T映射到布尔类型的表达式，满足

，B为布尔集合；

1. E是弧表达式函数，是弧A映射到颜色多重集的表达式，满足：

，s(a)是N(a)的库所；

1. I是初始化函数，定义了开始运行时P包含的颜色多重集：

。

其中，Type(x)表示表达式x的类型；Var(x)表示表达式x中包含的变量集合。

由以上定义可以看出，着色Petri网作为Petri网的扩展，它是高级网的一种，因此，着色Petri网的动态性质和分析方法都可以参照基本Petri网。

### 2.3.2着色Petri网的动态行为

Petri网通过定义发生规则(Firing Rule)定义Petri网的动态行为。Firing Rule的每次发生都导致token在库所中动态变化，从而表现系统变化着的状态。在着色Petri网中，由于变量和颜色的引入，变迁上的描述首先需要进行绑定(赋值)，才能引起变迁的发生。

定义(2.10) 绑定

一个变迁的绑定(Binding)是一个函数b，它的定义域为t的变量集Var(t)，且满足条件：

1. ；
2. 。

变迁t的绑定就是把属于Var(t)的变量用一种适当的颜色代替，且警卫函数必须为真。

定义(2.11) Token元素、绑定元素、发生步

一个Token元素是一个二元组，其中，；一个绑定元素是一个二元组，其中，。所有Token元素构成的集合记为TE，所有绑定元素构成的集合记为BE。一个Token是定义在TE上由Token元素构成的一个多重集，而一个步骤(Step)Y是定义在BE上由绑定元构成的一个多集。

定义(2.12) 使能

如果一个步骤Y满足，则称Y在状态M下使能。

当一个步骤使能后，该步骤就可以发生(Occur)。这意味着在相应的输入库所中删除Token，在输出库所中增加Token。Token的颜色和数量由弧上表达式的求值结果决定。

定义(2.13) 发生规则

当步骤Y在状态M1下使能，步骤Y可以发生，网的状态从M1变化到M2，即：。则从M1通过步骤Y是直接可达M2的，记为M1[Y>M2。

# 第三章 基于JavaScript的Lex工具的研究

与传统的词法分析自动生成工具相比，基于JavaScript语言的Lex工具——AliceLex具有不同的特点

## 3.1

根据2.5节的介绍，PNML在理论上可以支持任意类型的Petri网，这就为基于PNML实现着色Petri网的性质提供了理论上的可能性。着色Petri网作为基本Petri网的扩展，除了具有基本Petri网的基本元素外，还应具有相应的扩展。PNML文件对基本Petri网的支持以及PNML文件的基本结构和元模型已在第二章中进行了介绍，为实现PNML对着色Petri网的支持，需要使用PNML结构中的Type Define Interface和Feature Define Interface 定义着色Petri网类型，即对元模型中的Label进行限定，从而实现对着色Petri网在Place、Arc、Transition和整个网上的扩展。表3.1给出了在基本结构的基础上，着色Petri网的扩展元素。

表3.1 着色Petri网的扩展元素

|  |  |
| --- | --- |
| 基本结构 | 扩展 |
| Place | Marking、Initial Marking、Color Set |
| Arc | Inscription |
| Transition | Guard Function |
| Net | Signature，其中Signature中包含颜色集声明，函数声明和变量声明 |

为实现着色Petri网[11]，需要在Net上做Signature的扩展，在Place上进行ColorSet、Marking、InitialMarking的扩展，在Arc上进行Inscription扩展，同时在Transition上进行GuardFunction的扩展。

1. Net元素上的扩展

为实现着色Petri网，Net元素需要包含整个网的全局数据类型即颜色集的声明。同时，为满足Place、Transition和Arc上的扩展，Net元素中需要包含整个网的变量声明和函数生命。将Net元素上的扩展命名为Signature，它的UML模型图如图3.1所示。Signature是Petri网本身的扩展，主要包括三部分，SortDecl(颜色集声明)、VariableDecl(变量声明)、FunctionDecl(函数声明)。其中变量声明时，每一个变量必须指定它的数据类型，即它所属的颜色集。

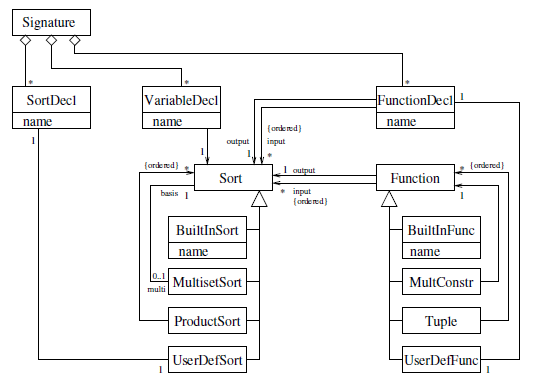


图3.1 Signature 扩展[12]

1. Arc元素上的扩展

在着色Petri网中，Arc需要指定一次变迁发生时，其输入弧和输出弧上的token流动情况。Token的流动情况一般通过弧上的变量或函数即弧上表达式进行表达。将Arc上的扩展命名为Term，它的UML模型图如图扩展：Term的结构如图3.2所示。

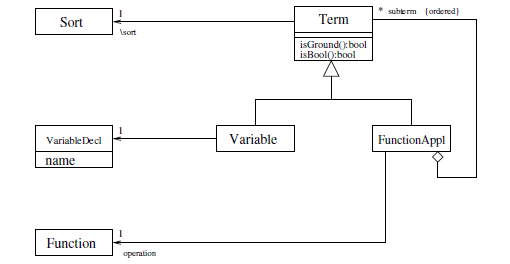


图3.2 Term扩展[12]

它主要包括两部分：Variable(变量) 和FunctionAppl(函数)。它们分别代表弧上表达式的两种不同的形式。在着色Petri网进行动态模拟时，弧上表达式需要与全局的函数或变量声明进行绑定。

1. Place元素上的扩展

Place上的扩展主要是对Signature中颜色集声明的引用，可以用Text表示。

基于以上研究和讨论，可以得出关于整个着色Petri网扩展的UML模型。对于整个着色Petri网，它的扩展UML模型如图3.3所示。图3.3给出了实现着色Petri网需要在PNML的Meta Model的基础上实现的扩展的UML模型图。

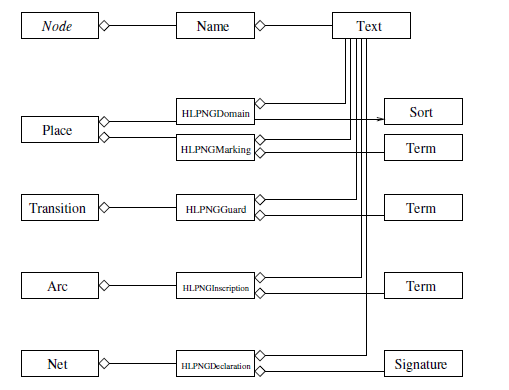


图3.3 Petri网扩展元素图[12]

## 3.2着色Petri网的PNML文件结构

图3.3包含了一个着色Petri网中包含的所有扩展元素。根据2.5节的介绍，这些元素需要转化成PNML文件中元素的子元素。根据表2.1中Meta Model的对应关系，对表2.1进行改进和扩展，得到表3.2中着色Petri网与PNML文件中元素和属性的转换关系。

表3.2给出了着色Petri网在PNML文件中需要具有的主要元素和相应的属性。经过上述分析，将表3.2中的元素转化到XML文件中的标签，可以制定出支持着色Petri网的各个元素的XML Schema[13]，它的内容如图3.4所示。

|  |  |
| --- | --- |
| Signature Schema  <signature>  <tokenTypeDecl>  <name/>  </tokenTypeDecl>  <functionDecl>  <parameters>  <tokenType/>  <multiSetToken>  <tokenType/>  </multiSetToken>  </parameters>  </functionDecl>  <variableDecl>  <variable>  <tokenType/>  <multiSetToken>  <tokenType/>  </multiSetToken>  </variable>  </variableDecl>  </signature> | Place Schema  <place>  <marking>  <tokenType/>  <multiSetToken>  <tokenType/>  </multiSetToken>  </marking>  <initialMarking>  <value/>  <tokenType/>  <multiSetToken>  <tokenType/>  </multiSetToken>  </initialMarking>  <type/>  </place> |
| Transition Schema  <transition>  <variable>  <value/>  </variable>  <name>  <value/>  </name>  <cond>  <value/>  </cond>  <toolspecific/>  </transition> | Arc Schema  <arc>  <name>  <value/>  </name>  <inscription>  <value/>  </inscription>  <toolspecific/>  </arc> |

图3.4着色Petri网各元素的XML Schema

图3.4给出了着色Petri网各元素的XML Schema，它包含了一个完整的支持着色Petri网的PNML文件的结构。在该文件中定义了网上的颜色集扩展，库所上的标志扩展，弧上的描述扩展以及变迁上的警卫函数扩展。

本文在第三章中，介绍了有关PNML对着色Petri网支持的研究，并给出了着色Petri网的XML Schema。为将着色Petri网应用到工作流建模中，本文将介绍一个基于PNML的着色Petri网工作流建模工具的设计与实现。

表3.2 着色Petri网元素与PNML文件的对应关系[[1]](#footnote-1)

|  |  |  |  |
| --- | --- | --- | --- |
| 类 | XML 元素 | XML属性 | 父元素 |
| PetriNetFile | <pnml> |  |  |
| PetriNet | <net> | id: ID | <pnml> |
| type: anyURI |
| Signature | <declaration> |  | <net> |
| SortDel | <datatype> | type: Enumeration | <declaration> |
| VariaDel | <variable> |  | <declaration> |
| FuncDel | <function> |  | <declaration> |
| Place | <place> | id: ID |  |
| Text | <marking> |  | <place> |
| <initialmarking> |
| <type> |
| Transition | <transition> | id: ID |  |
| Term | <guard> |  | <transition> |
| Arc | <arc> | id: ID |  |
| source: IDRef(Node) |  |
| target: IDRef(Node) |  |
| Term | <inscription> |  | <arc> |
| Page | <page> | id: ID |  |
| ToolInfo | <toolspecific> | tool: string |  |
| version: String |  |
| Graphics | <graphics> |  |  |

# 第四章 基于JavaScript的Lex工具——AliceLex概述

本章介绍了一个基于JavaScript的词法分析器自动生成工具。

## 4.1 需求设计

### 4.1.1功能需求设计

作为一个词法分析器自动生成工具，本软件具有传统lex工具拥有的基本功能：

1. 对Petri工作流网基本结构的操作：用户可以通过可视化界面对模型进行结构上的操作，包括：创建工作流模型、修改模型、打开原有模型，同时，用户可以对创建的模型以及修改过的模型进行自动布局。
2. 对Petri工作流网动态行为的分析：包括模型分析和模型仿真

图4.1 给出了系统相应的用例图。

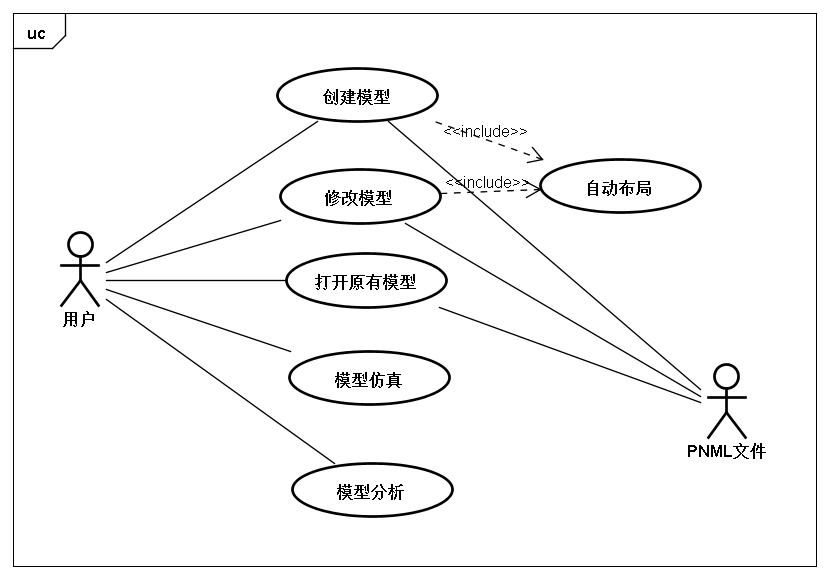


图4.1 系统用例图

本项目的任务范围主要集中在以下几个方面：

1. 模型展示：展示不同类型的模型，该工具作为一个统一的模型展示工具，它使用PNML作为底层的文件类型，通过制定自己的PNML标准，实现对多种网类型，包括着色Petri网的支持；
2. 添加对着色Petri网数据的支持：与其他工作流建模工具相比，该工具除了需要实现对控制流分析的支持外，还需要实现对数据流分析的支持。
3. 模型分析：研究现有的模型分析方法开发符合自己需要的模型分析算法；
4. 模型仿真：实现着色Petri网的模型仿真；

将开发的工具集成到BeehiveZ中：将BeehiveZ的模型展示模块移植到PNK上，使用PNK的Editor显示模型，并对模型进行仿真。

### 4.1.2 非功能需求设计

目前，本项目需要符合以下约束：

1. 目前我们用户需要的文件格式为pnml与cpn
2. 读取底层文件时允许忽略DTD验证
3. 模型分析模块接受的Petri网模型需要带有标识。
4. 系统需要处理多个工具生成的PNML文件，保证兼容性和可扩展性。

在此基础上，本系统的性能需求如下：

表 4.1系统性能需求表

|  |  |
| --- | --- |
| 时间特性要求 | 系统解析模型文件的功能，要求解析时间不超过0.5秒 |
| 系统对模型的分析和仿真算法，要求运行时间不超过1.5秒 |
| 灵活性要求 | 本系统需要满足开源项目的要求，支持跨平台使用 |
| 本系统需要预留开发和扩展接口，以便后续开发 |
| 输入输出要求 | 输入文件格式为：pnml以及cpn |
| 输入文件格式与输出文件格式一致 |
| 故障处理要求 | 编写异常类，提供异常处理功能，系统出现错误能友好的提醒 |
| 对于开发人员，提示错误发生的位置和原因，以便于修改和维护。 |

同时，作为一个开源项目，本项目使用如下开发工具和平台：

表 4.2运行环境规定

|  |  |
| --- | --- |
| 开发语言 | Java JDk 1.6.0 |
| 开发工具 | Eclipse SDK 3.3.2/ PNK2.2 |
| 源代码管理 | SVN |
| 操作系统 | WindowsXP/Vista/Window7，可支持Linux |

## 4.2 概要设计

基于PNML的工作流建模工具[14]采用MVC[15]的总体设计框架，M代表模型，数据层(Model)，V代表视图，界面层(View)，C代表控制器(Controller)。图4.2给出了传统的MVC体系结构的基本框架。通过采用MVC模型，本系统被分割成三层：视图(用户界面)获取用户输入并把执行过程的相关状态和信息显示出来；模型负责处理从图形界面获得的数据，控制器在视图和模型之间传递数据。采用MVC架构，可以使三部分相互独立，从而使系统便于修改和扩展，同时可以使用多种设计模式，减少代码量。

在本系统中，数据层包括系统使用的底层PNML/CPN文件，以及集成到BeehiveZ中后，涉及到的内嵌数据库Derby。界面层提供用户与系统的操作接口，模型文件的打开，分析和仿真都在界面层实现，在界面层的开发中，在PNK自带的Editor的基础上，使用java的swing组件进行开发，减少代码量。在控制器的设计中，将系统按照功能划分模块，不同的功能分别由相应的模块实现，保证系统体系结构的清晰性和完整性，同时便于系统的后续扩展。

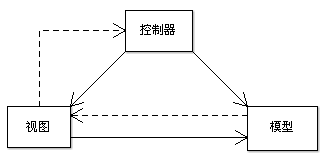


图4.2 传统MVC模式图

将系统功能进行模块划分，得到系统的结构模块[16]如图4.3所示。本系统共包括7个模块(不包括PNML/CPN文件模块)，其中基本功能模块包括：图形编辑器模块，输出管理器模块，文件管理器模块，模型分析器模块和模型仿真器模块。

另外，着色Petri网和语法规则库模块包括着色Petri网的的所有内存对象和与Firing Rule，分析方法等与Petri网动态性质相关的类。图形用户界面可以独立于其他模块实现，本系统中主要是基于原有的PNK的Editor实现。各个模块的主要功能如下：

1. 图形用户界面模块：图形用户界面模块的主要作用是显示系统模型，提供用户与模型之间的交互功能；
2. 图形编辑器模块：监听用户在图形界面上对Petri网的创建、修改、删除和打开操作，并将相应的操作传递给内存中相应的Petri网对象实例；
3. 输出管理器模块：将内存中Petri网实例的变化显示到图形界面上；
4. 文件管理器模块：处理文件的打开、保存等操作；
5. 模型分析器模块：对内存中的Petri网对象根据用户的需求选择相应的算法进行可达性等的分析，并将分析结果显示在图形界面上；
6. 模型仿真器模块：对内存中的Petri网模型进行动态仿真，在仿真的过程中，需要接受来自用户操作界面的输入，并将结果返回给操作界面；
7. 语法规则库和着色Petri网模型模块：提供表示Petri网静态结构的Place、Transition、Arc等的类实现，以及Petri网动态性质相关的Marking、Inscription、Firing Role、分析算法等的类实现；
8. PNML/CPN文件模块：PNML/CPN文件的底层存储系统。

图形用户界面

图形编辑器

模型分析器

模型仿真器

输出管理器

文件管理器

语法规则库和着色Petri网模型

PNML/CPN文件

图4.3 系统体系结构图

系统各个模块之间的关系如表4.3所示：

表4.3系统各模块之间的关系[[2]](#footnote-2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | white.png | white.png | white.png | black.png | black.png | black.png | black.png | white.png |
| 2 | white.png | white.png | black.png | white.png | white.png | white.png | black.png | white.png |
| 3 | black.png | white.png | white.png | white.png | white.png | white.png | black.png | white.png |
| 4 | black.png | white.png | white.png | white.png | white.png | white.png | black.png | black.png |
| 5 | black.png | white.png | black.png | white.png | white.png | white.png | black.png | white.png |
| 6 | black.png | white.png | black.png | white.png | white.png | white.png | black.png | white.png |
| 7 | white.png | black.png | black.png | white.png | black.png | black.png | white.png | white.png |
| 8 | white.png | white.png | white.png | black.png | white.png | white.png | black.png | white.png |

6. pnk.appControl：该包包含PNK上不同Application相互交互的内容。相应接口在pnk.appControl.base包中。

图4.4给出了系统的包结构设计。在本系统中，不同的包负责不同的功能，它们之间通过接口交互，从而实现高内聚低耦合。同时，它们之间相互协作，共同实现工作流建模工具的功能。

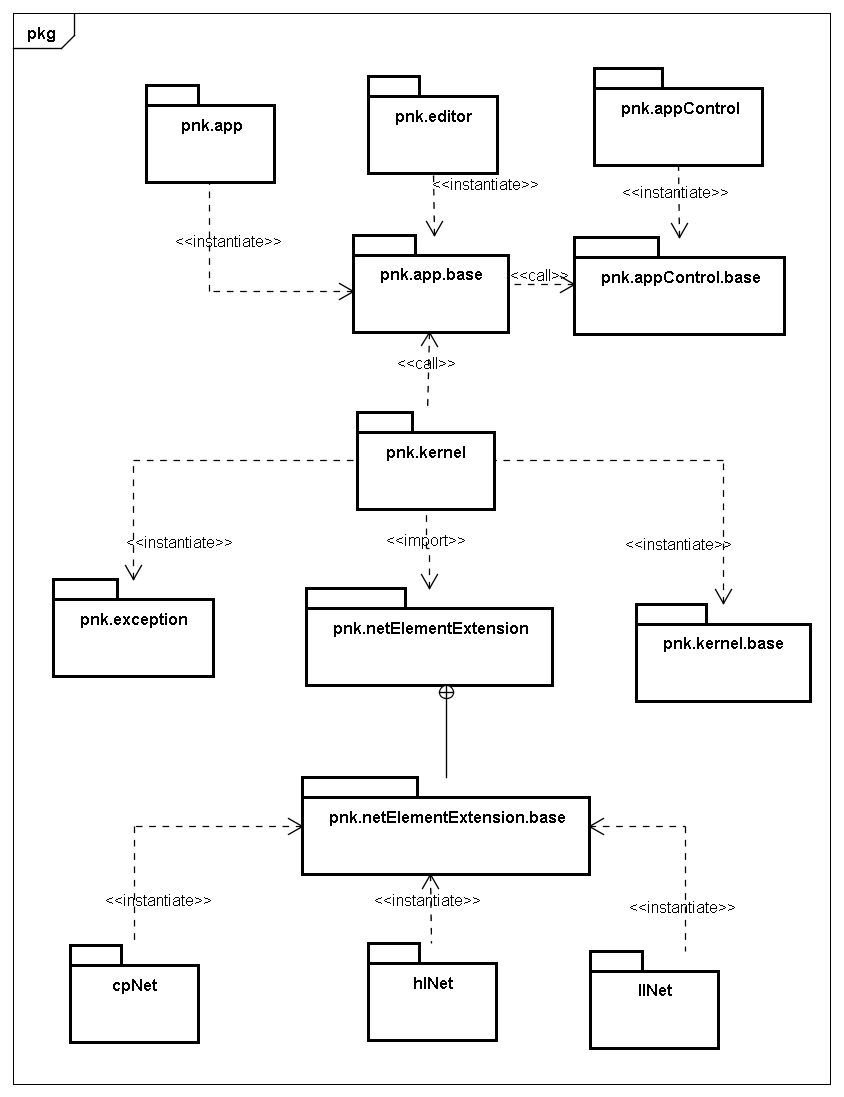


图4.4 系统包结构图

# 第五章 基于JavaScript的词法分析器自动生成工具详细设计及实现

由于该工具在PNK的基础上进行扩展实现，因此它的设计和实现必须符合PNK的开发规定。关于在Eclipse中集成PNK进行开发的步骤，在第二章中已经进行了详细介绍。根据4.2.1节的介绍和PNK的开发规则，该工作流建模工具的详细设计和实现如下。

## 5.1 流程图设计

|  |
| --- |
|  |

图5.1 PNK中着色Petri网定义的XML文件

为了在PNK中实现着色Petri网，需要首先在PNK下的netTypeSpecifications文件夹下编写相应的网定义文件，指定着色Petri网中基本元素Net、Place、Transition、Arc上的扩展，同时将它们与具体的内存实现类相关联。

着色Petri网的网定义文件的结构分为两部分：

1. 着色在基本元素上的扩展及扩展在最终的PNML文件中对应的标签名称。这部分内容已在第三章中详细介绍。
2. 着色Petri网的基本结构和扩展对应的具体实现类。

基于以上两部分，为实现着色Petri网，定义着色Petri网的文件CPNet.xml内容如图5.1所示。

## 5.2 算法设计

AliceLex的核心难点是词法分析相关算法在JavaScript语言上的具体实现，包括从字符串解析到正则表达式（NFA），NFA转换为DFA，DFA压缩

### 5.2.1 字符串转NFA

1. 详细设计

pnk.kernel包中主要实现Petri网的基本结构。为实现可扩展性，设计Extendable类，作为可扩展对象的基类。其中：Place、Transition、Arc、Net以及它们的扩展Extension均继承Extendable类，它们的扩展通过继承Extension的类实现。pnk.kernel包的类图及它们之间的关系如图5.2所示。

1. 实现

在这个包中的主要实现的类及它们的方法介绍如表5.1所示：

表5.1 pnk.kernel实现

|  |  |  |  |
| --- | --- | --- | --- |
| Place：  实现Petri网的基本结构Place的类   |  | | --- | | getInitialMarking()//得到初始化标识；  getMarking()//得到标识；  setMarkingAsInitial()//重置标识 | | Transition：  实现Petri网的基本结构Transition的类   |  | | --- | | getMode()//得到变迁的模式 | |
| Arc：  实现Petri网的基本结构Arc的类   |  | | --- | | getInscription ()//得到弧上表达式；  getPlace ()//得到Arc连接的Place；  getTransition ()//得到transition | | Net：  实现Petri网的类  它的主要作用是维护Petri网的Place\Transition\Arc的Vector，同时可以获得和访问它们 |

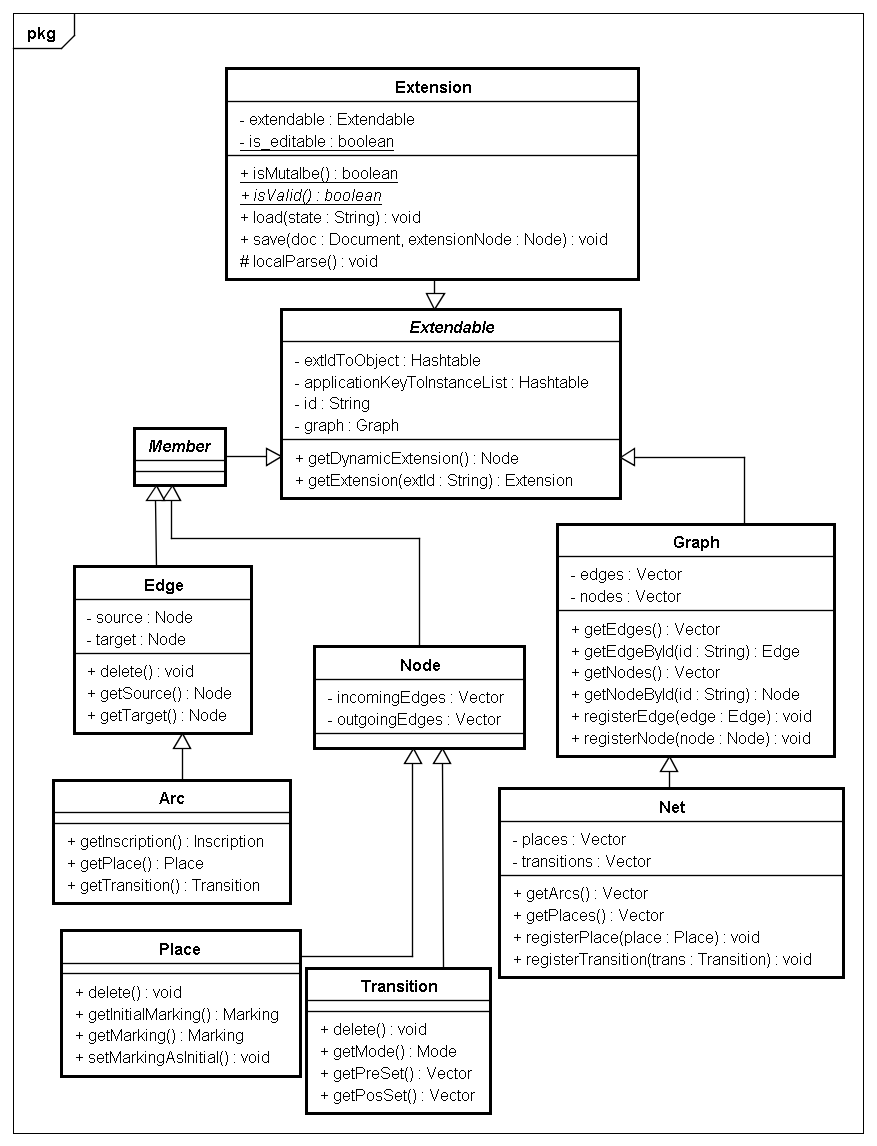


图5.2 pnk.kernel详细设计图

### 5.2.2 pnk.netElementExtension包

pnk.netElementExtension包括多个子包，其中pnk.netElementExtension.base包含对网进行扩展时，必须实现的三个抽象类和接口，它们分别是FiringRule，Inscription和Marking。

1. FiringRule：

FiringRule的接口图如图5.3所示。FiringRule是Net对象的扩展，是进行模型仿真时必须使用的类，它规定了Petri网的点火规则，决定了Transition是否能够执行，以及token在库所间的传递。FiringRule同时规定了在仿真的过程中是否允许模型与用户的交互。

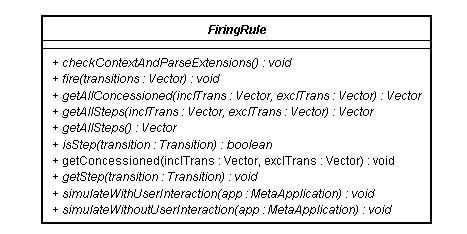


图5.3 Firing Rule 抽象类图

1. Inscription和Marking

Inscription的接口图和Marking类图如图5.4所示。其中：Inscription是Arc对象的扩展，它作为一个接口，只有一个方法evaluate，当我们编写具体的网类型的ArcInscription时，必须实现这个接口。Marking是Place对象的扩展，它继承Inscription，同时添加了自己方法add和sub。

## 5.3 其它设计

## 5.3 项目实现

# 第六章 基于JavaScript的lex工具的示例展示

在完

。

## 6.1 lex工具经典示例

为6.2 BeehiveZ系统界面更改和显示

BeehiveZ系统采用插件开发，每个插件有独立的界面。BeehiveZ中主要包括模型导入、模型相似度测量、模型挖掘、模型索引等功能模块，各个模块修改后的界面如下：

1. 模型导入界面：

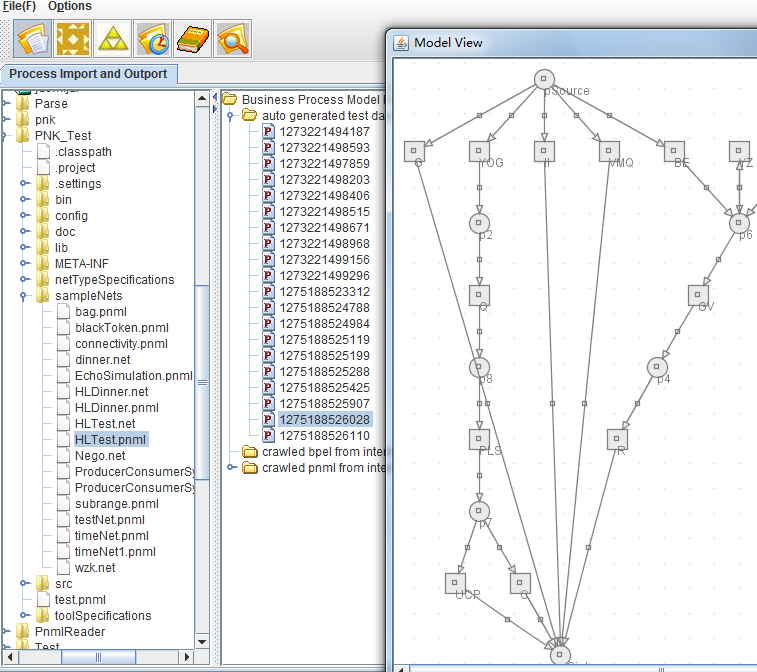


图6.1 模型导入界面

该模块主要负责PNML模型的导入。在模型导入的过程中，相应的文件作为Process记录到process表中，同时将相应文件的信息以输入流的形式存储到相应的字段中并刷新模型列表。

1. 模型相似度测量模块界面

## 6.2基于AliceLex内核的代码编辑器

图6.2模型相似度测量模块界面

该模块的主要功能是进行模型相似度测量，第一个标签进行模型选择，并将选择的模型以可视化的形式表示。第二个标签是相似度测量算法的选择标签，用户可以选择不同的相似度测量算法，对选中的模型进行相似度测量。

本文描述的独立开发的着色Petri网工作流建模工具，目前与BeehiveZ的集成还仅局限在模型显示方面，关于两者集成的后续工作，在第七章会详细介绍。

# 第七章 总结与展望

## 7.1 总结

着色Petri网作为传统Petri网的扩展，在Petri网化简、数据流分析等方面具有很大的优势。目前，虽然对于着色Petri网的理论研究取得了很大进展，但是并没有很好的工具来支持基于着色Petri网的工作流建模。在前人研究的基础上，本论文提出了一种基于PNML的着色Petri网工作流建模语言，并在开源PNK软件基础之上进行着色Petri网工作流建模工具的设计和实现。

尽管目前开发的系统还比较简单，并且具有一定的局限性，但是随着基于着色Petri网的工作流建模技术的发展，这个开源项目将会得到不断完善。

本文的主要工作为：

1. 介绍了以经典Petri网为例的有关Petri网的基本理论，其中包括Petri网的静态结构和动态分析方法的介绍。
2. 介绍了工作流网的相关理论知识，来阐释Petri网的控制结构的表示方法。
3. 着重介绍了着色Petri网相关的理论知识，包括着色Petri网的静态结构，绑定以及动态分析方法等。
4. 详细介绍了基于PNML对经典Petri网的支持，在现有PNML的基础上进行扩展，使其支持着色Petri网的理论和实现方法。
5. 介绍了基于上述描述的扩展PNML，在PNK中实现着色Petri网建模工具的详细的设计和实现。
6. 介绍了BeehiveZ系统的功能及现状，并描述了将上述着色Petri网建模工具。

## 7.2 基于JavaScript的lex工具的局限性

1. 系统对着色Petri网颜色集的支持不够完整

与CPN Tools相比，本系统只支持特定类型的颜色集：int、real、string、enumeration、list，不能像CPN Tools一样支持自定义的数据类型。这就造成了系统在实际应用时的局限性。

1. 系统对分析算法的实现不够完整

由于Petri网的分析算法相对比较复杂而且目前有很多基于Petri网的分析算法，因此对分析算法的研究和实现存在一定困难。因此本论文只介绍了基于概率技术的数据流分析算法，没有在系统中进行实现。

1. 系统的显示界面比较简陋

由于开发时间的问题，目前的界面显示只实现了简单的自动布局，不能实现模型在显示时的自动缩放等功能。同时，在创建网对象时，操作不够友好。

1. 系统与BeehiveZ的集成存在一定的问题

目前，本系统的界面层已经移植到BeehiveZ中，但是对于模型分析的相关算法，由于本系统还没有很好的实现，所以没有移植到BeehiveZ中。

## 7.3 基于JavaScript的lex工具的改进方向和发展前景

当前系统存在的问题已经在6.2节中进行了描述。基于上述问题，本系统需要以下扩展：

1. 开发符合特定需要的数据流分析算法，并尽快集成到BeehiveZ中。
2. 增强系统对颜色集的支持，改进CPN文件解析算法，在CPN文件向着色Petri网内存对象转化的过程中，尽可能多的保存CPN文件中信息，防止信息丢失。
3. 尽量采用开源的第三方Java自动布局项目，并将它应用到目前的界面显示中，解决当前显示界面的简陋性和不友好性问题。
4. 同时，着色Petri网在工作流建模系统中的应用远远超过目前项目所能涵盖的范围，有了本系统对着色Petri网内存对象的支持，在本系统上可以做其他的扩展。

为实现一个完整的建模工具，并将它应用到实践中，本系统以后需要做出如下扩展：

1. 添加模型相似度分析算法。本系统目前实现的功能基本是处理单个的着色Petri网，还没有考虑到比较不同模型的问题，本系统需要在模型管理方面作出扩展，尽量向工作流管理工具靠拢。
2. 添加对模型引擎、模型挖掘和相关算法的支持。模型引擎和挖掘作为模型分析和管理的另一个领域，有很广阔的研究前景，本项目可以在着色Petri网的基础上，进一步对这个领域进行研究。

# 参考文献

[1] W.Reisig. Petri Nets: An Introduction. Springer. New York. 1982.

[2] 吴哲辉.Petri网导论.北京.机械工业出版社.2006/4.页1-24

[3] 吴哲辉.Petri网导论.北京.机械工业出版社.2006/4.页27-41

[4] Aalst van der W.M.P. The Application of Petri Nets to Workflow Management. The Journal of Circuits. Systems and Computers. Vol. 8, No. 1. 1998.Pages 21-66.

[5] Kurt Jensen. Colored Petri Nets: A High Level Language for System Design and Analysis*.* Lecture Notes in Computer Science. Vol. 483. Advances in Petri Nets 1990. Pages 342-416. Berlin, Germany: Springer-Verlag. 1991.

[6] Jensen K. Colored Petri Nets- basic concepts, analysis methods and practical use. German: Springer, 1992

[7] Kurt Jensen. An introduction to the theoretical aspects of Coloured Petri Nets. Lecture Notes in Computer Science. Volume 803/1994. A Decade of Concurrency Reflections and Perspectives. Pages 230-272. Springer Berlin / Heidelberg

[8] Kurt Jensen. Colored Petri nets. Lecture Notes in Computer Science. Volume 254/1987. Petri Nets: Central Models and Their Properties. Pages 248-299. Spring Berlin/Heidelberg

[9] Ekkart Kindler and Michael Weber. The Petri Net Kernel An infrastructure for building Petri net tools. International Journal on Software Tools for Technology Transfer (STTT). Volume 3, Number 4 .September,2001.Pages 486-497

[10] Michael Weber and Ekkart Kindler. The Petri Net Kernel. Lecture Notes in Computer Science, Vol.2472/2003. Petri Net Technology for Communication- Based Systems. Pages 109-124. Berlin,Germany. Springer. 2004

[11] J. Billington, S. Christensen, K. van Hee, E. Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber. The petri net markup language: Concepts, technology, and tools in Applications and Theory of Petri Nets 2003: 24th International Conference. Eindhoven, The Netherlands. June 2003

[12] Ekkart Kindler .High-level Petri Nets – Transfer Format-Working Draft of the International Standard ISO/IEC 15909 Part 2 –Version 0.5.0.November 8, 2004

[13] M. Westergaard.Towards A High-Level Petri Net Type Definition. Workshop in the Definition, Implementation and Application of a Standard Interchange Format for Petri Nets in 25th International Conference On Application and Theory of Petri Nets.Bologna, Italy, June 21-26, 2004. Ekkart Kindler. June 2004

[14] Dongsheng Liu, Jianmin Wang, Stephen C.F.Chan, Jiaguang Sun, Li Zhang. Modeling workflow process with Colored Petri nets. Computers in Industry. 49(2002). Pages 267-281

[15] MVC. http://zh.wikipedia.org/zh/MVC

[16]丁彩红**.** Petri网建模工具的面向对象设计. 系统工程与电子技术. 2000年第八期, 卷 22.

[17] Alessandra Di Pierro, Chris Hankin and Herbert Wiklicky. On Probabilistic Techniques for Data Flow Analysis. Electronic Notes in Theoretical Computer Science 190. 2007. Pages 59–77

# 致谢

时光如白驹过隙，转眼间本科的时光已尽。在过去的四年里，感谢南京大学和南京大学软件学院的培养。四年的本科生涯，让我对软件工程有了系统的认识也具有了足够的实践经验。感谢软件学院给我提供的良好的软硬件条件，感谢软件学院勤劳善良的老师们的谆谆教导。

特别要感谢我的论文导师，南京大学软件学院的葛季栋老师，在论文开题和论文撰写的过程中对我的关心和指导。由于他的耐心提醒，让我很好的控制了项目实践和论文撰写的进度，并在理论研究上给了我很大帮助。感谢他在我修改论文的过程中，一次又一次细心认真的审阅。同时也感谢他在生活和学习中对我的帮助和指导。

在此要感谢郑涛老师在《软件构造》课程中讲述的编译原理对我起到的启蒙作用，使得我对编译原理产生了较浓兴趣，也才能自己去学习和体悟编译原理中的各种算法，并将其在毕业设计中一一实现。

感谢209宿舍的三个哥们，以及后来的211的众多舍友，是你们的关心支持和帮助，让我在大学四年里生活愉快而幸福，才能更好地投入到学习之中，才能专注地去研究和学习自己感兴趣的内容。

另外，衷心感谢一直以来默默支持关心我的家人和朋友们。

1. 灰色背景部分表示着色Petri网的扩展部分 [↑](#footnote-ref-1)
2. white.png表示相应模块之间没有关联关系；black.png表示相应模块之间存在关联关系

   ## 4.3系统包结构设计

   将上述的功能模块对应到相应的包中，得到系统的主要包结构图如图4.4所示，不同的包具有不同的功能。在该系统中，pnk.kernel为核心包，它实现一个Petri网基本结构的内存对象，其他包与pnk.kernel的关系如图4.4中所示。系统中主要包的功能及它们的关系说明如下：

   1. pnk.kernel：包含与Petri网基本结构相关的内存结构的类实现，包pnk.kernel.base包含pnk.kernel提供的接口信息，通过调用pnk.kernel.base中的方法，可以实现对pnk.kernel中的内存对象类的调用。
   2. pnk.netElementExtension：包含实现特定Petri网需要实现的扩展类。这个包由多个子包组成，其中：pnk.netElementExtension.base：包含实现Petri网扩展的接口。其中包含Marking、Inscription等多个抽象类。在PNK的基础上，本系统在实现base包中接口的基础上，借助原有hlNet，llNet包并作出相应修改和添加，实现cpNet，cpNet包中包含实现着色Petri网的类文件
   3. pnk.editor：本系统的图形用户界面，它是一个application，但由于太复杂，作为单独的Editor包。
   4. pnk.app：在PNK上开发的应用程序，都应放在这个包内。在PNK上的Application都实现pnk.app.base中的相应接口，若开发自己的应用程序，也需实现pnk.app.base包中的相应接口。
   5. pnk.exception：包含系统实现异常处理需要的类

   [↑](#footnote-ref-2)