SHANGHAI JIAO TONG UNIVERSITY



课程名称： 过程建模及优化

学生姓名/学号: 侯国鑫/118037930074

学生姓名/学号: 林许亚伦/118037920085

学生姓名/学号: 姚朋/118037910039

专 业: 软件工程

指导教师: 蔡鸿明

学院(系): 电子信息与电气工程学院

一、系统主要需求 3

二、业务问题分析 3

2.1、核心流程识别 3

2.2、流程模型 4

三、功能场景识别 4

3.1、功能场景识别 4

3.2、实体交互图 5

四、功能用例构造 6

4.1、用例图 6

4.2、功能清单 7

五、数据流识别 8

5.1、数据分析 8

5.2、数据流图 9

5.3、类图 10

六、状态及控制行为识别 10

6.1、分析状态变化 10

6.2、状态图 10

七、基于Petri建模 11

7.1、基于Petri网进行场景描述 11

7.2、流程模型 11

7.3、流程模型仿真 12

# 一、系统主要需求

1. 作者可以注册、登录、提交游记、追踪游记、管理游记功能。
2. 作者可以撤销自己提交的游记，游记审稿过程中不得撤销。
3. 编辑查看自己的审稿任务，编辑审稿意见，查看和改游记相关联的所有审稿意见。
4. 栏目主编分配审稿任务，最终决定游记评审结果，可以决定游记被拒、重审或通过。
5. 游记检索：用户可以查询所有以发布游记，需支持条件查询（如关键字、标签、作者、发表年份等信息）。
6. 游记归档：提供为游记添加标签的功能，每篇文章可以拥有多个标签；提供吧数据库中保存的游记转化为以RDF格式存储的文件的功能。
7. 能够通过消息全程告知用户的游记信息（包括投稿、审稿结果、是否发表）。
8. 系统需要包括手机端和Web端，以及服务器端。

# 二、业务问题分析

## 2.1、核心流程识别

1、（（**用户**）发布征稿需求 🡪）（**作者**）投稿（可修改） 🡪 （**栏目主编**）分配编辑 🡪 （**编辑**）初审 🡪 （**栏目主编**）终审 🡪 （**栏目主编**）发布 🡪 （**作者/编辑**）添加标签 🡪 （**作者/编辑**）存档归档

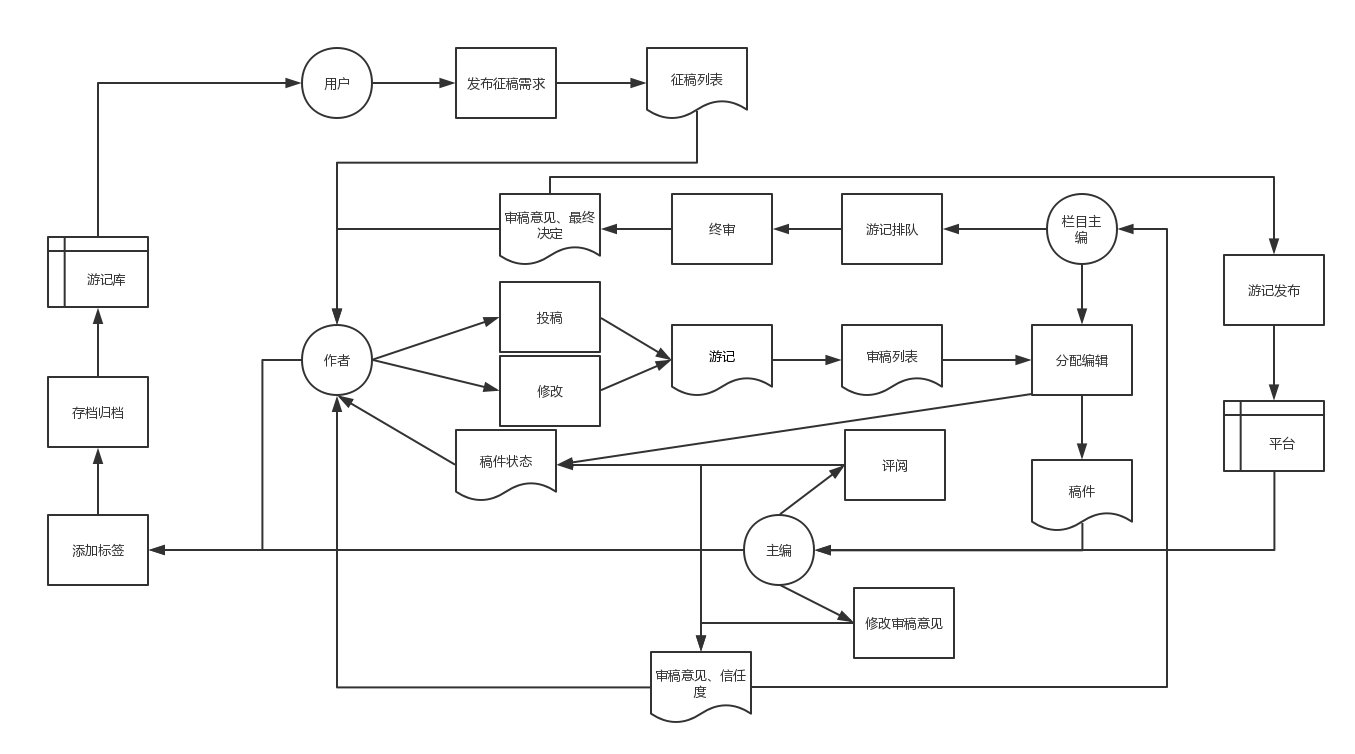
2、（**用户**）查询游记

3、（**作者**）查看初审和终审信息

4、（**栏目主编**）查看初审信息

5、（**主编**）查看已有初审信息

## 2.2、流程模型



# 三、功能场景识别

## 3.1、功能场景识别

**1、作者**

（1）作者在投稿的过程中，可进行游记投稿、投稿修改、查询审稿的意见并进行回复、对审核通过的游记进行标记和存档。

（2）对已经投递的稿件进行查询。

（3）对审核通过的游记进行标记并归档。

**2、栏目主编**

（1）对作者投稿的游记分配给编辑。

（2）对编辑审阅过的游记进行排序。

（3）对游记进行复审，并给出审阅意见和最终决定。

（4）将审核通过的游记进行发布。

**3、编辑**

（1）对分配到的游记进行评阅并给出审核意见，在最终决定前可进行修改。

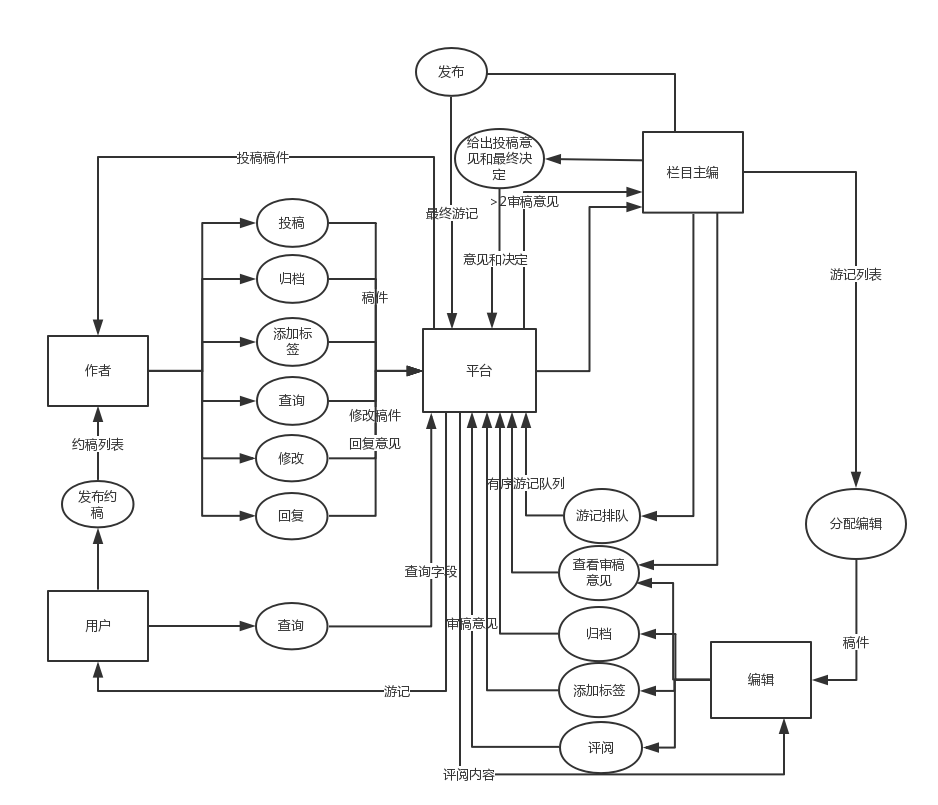
（2）对审核通过的游记进行标记并归档。

**4、用户**

（1）查询存档中的游记。

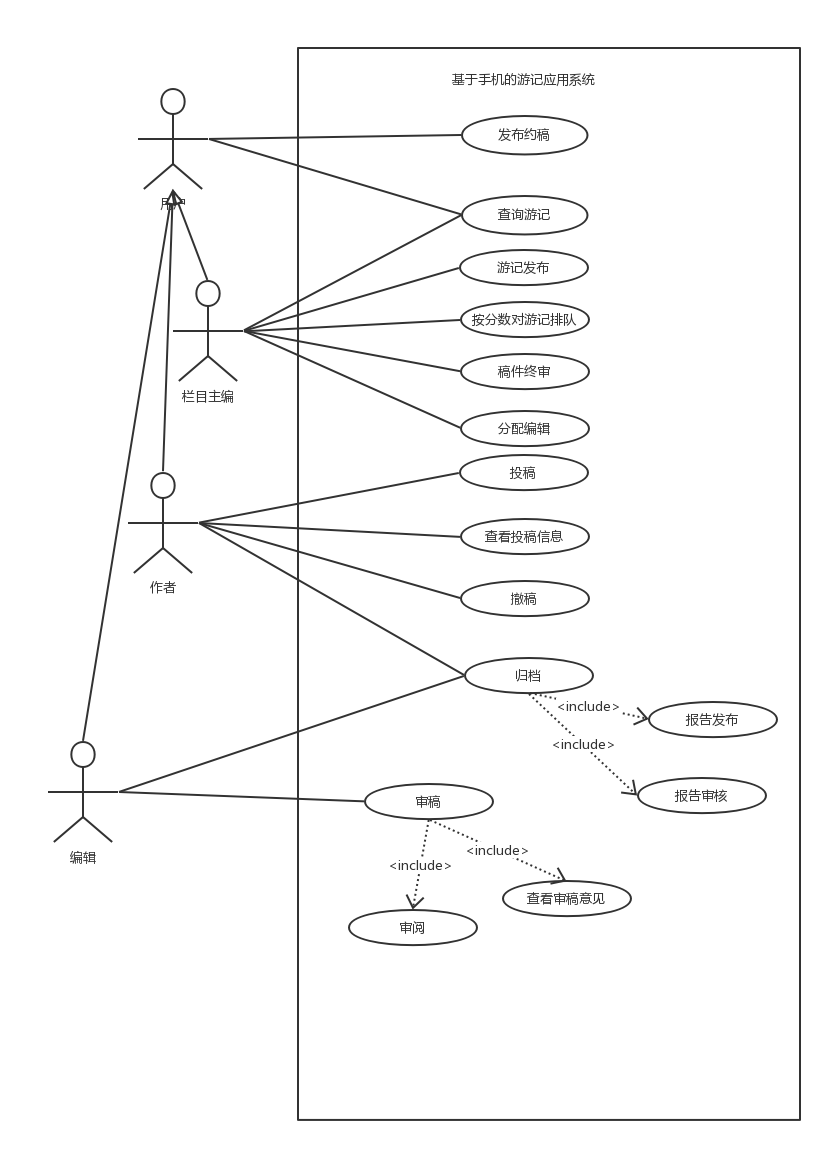
（2）在平台上发布征稿需求。

## 3.2、实体交互图



# 四、功能用例构造

## 4.1、用例图



## 4.2、功能清单

因为栏目主编、作者、编辑均是用户的衍生，所以他们都继承自用户。

|  |  |
| --- | --- |
| **用户** | |
| **查询** | 基础功能，普通用户仅能查询已归档的游记。 |
| **评论点赞** | 用户可对发布的游记进行点赞，评论。 |
| **征稿** | 可发布征稿需求。 |

|  |  |
| --- | --- |
| **作者** | |
| **查询** | 在用户查询的基础上，添加查询自己的投稿极它们的状态。 |
| **撤稿** | 作者可对已投稿的游记进行撤回。 |
| **存档归档** | 作者或主编可进行稿件的归档。归档包括标签的添加和格式的处理。 |

|  |  |
| --- | --- |
| **栏目主编** | |
| **查询** | 在用户查询的基础上，添加查看投稿列表、查看稿件初审信息。 |
| **分配编辑** | 栏目主编对稿件分配给评阅人进行审阅，一般每一篇游记2-3人。 |
| **稿件终审** | 审稿意见>2时且在截止日期前，稿件送至栏目主编处审稿，并给出审稿意见和最终决定。 |
| **对初审稿件排队** | 栏目主编可对游记进行排队。 |
| **游记发布** | 若游记评审通过，则将文章发布在平台上。 |

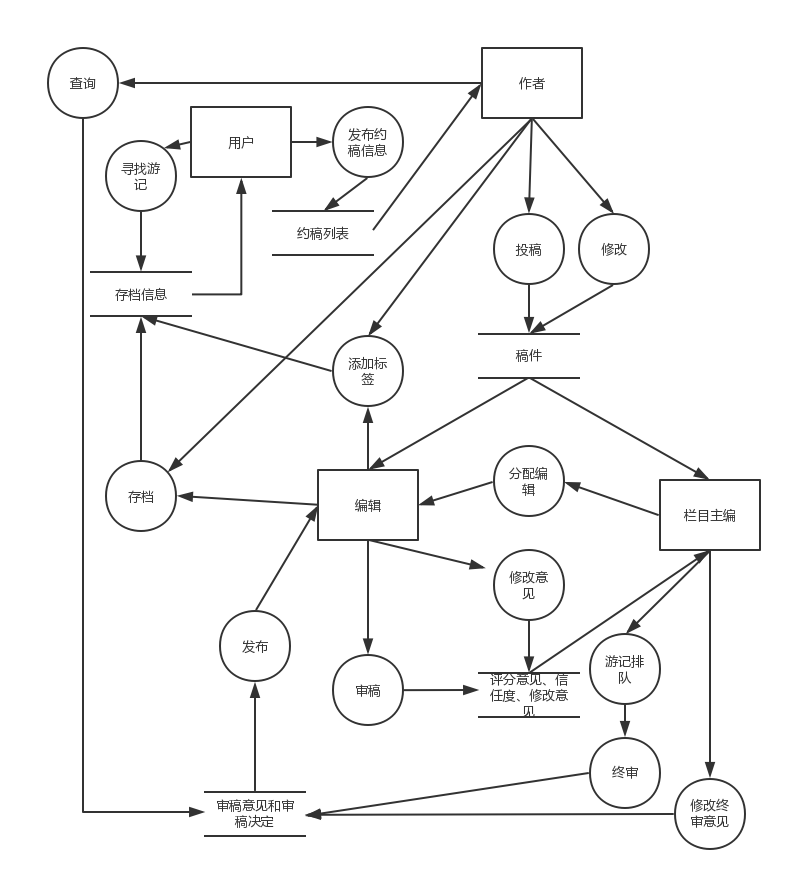
|  |  |
| --- | --- |
| **主编** | |
| **查询** | 评阅人可以查看已有的审稿意见。 |
| **稿件初审** | 对稿件进行审阅，并给出审稿意见分支1-100及对结果的信任度。 |
| **修改审稿意见** | 在游记最终结果之前可进自己意见的修改。 |
| **存档归档** | 作者或主编可进行稿件的归档。归档包括标签的添加和格式的处理。 |

# 五、数据流识别

## 5.1、数据分析

1. 作者进行投稿，生成**稿件数据**（并可对稿件数据进行修改）。
2. 作者可以对自己已经投稿（可以没发布）的稿件进行查询查看**审稿意见**。
3. 栏目主编对收到的**稿件**进行分配，给编辑进行审阅。
4. 编辑进行审稿，给出**评分意见**、**信任度**和**修改意见**（在最终决定前可以修改）。
5. 栏目主编对编辑的**评分**进行排队，并进行终审，给出**评审意见**（此时还可进行修改）。
6. 栏目主编给出**最终决定**。
7. 若通过审核，进行**游记**的发布。
8. 编辑对将要发布的**游记**进行添加**标签**并存档。
9. 用户可以对**存档中的游记**进行查询操作。
10. 用户可在平台发布**约稿需求**。
11. 作者可以查看**约稿需求列表**，并进行投稿。

## 5.2、数据流图

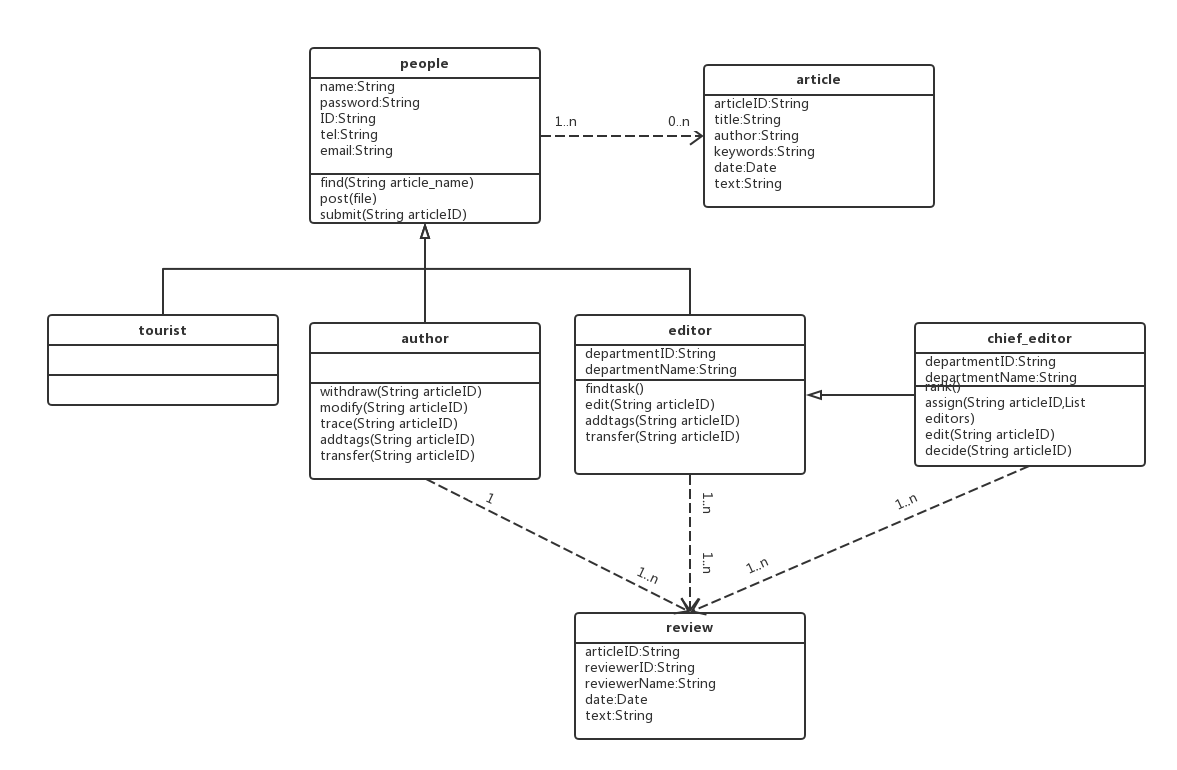


## 5.3、类图

类图描述了类之间的关系，其中我们系统涉及到多个角色，所以我们设计了一个公共的people类作为父类，其中定义了所有角色一些共有的属性和方法，其子类有三种，tourist，author，editor类，分别定义了他们各自的属性和方法，其中editor类又会衍生出一个chief\_editor类。

article类被所有的people类依赖，其是多对多关系，一个people可能有多篇article，一篇

article也可能被多个people拥有，rerview类被author，editor，cheif\_editor类依赖，其中author和review是一对多关系，一个author可能会有多篇review，editor和review是多对多关系，一个editor会有多个review，一篇review会有多个editor来进行review，同理chief\_editor和review类。



# 六、状态及控制行为识别

## 6.1、分析状态变化

状态图描述了一篇游记在其生命周期内的状态变化，首先是编写完成后进入待投稿状态，在投稿完成后进入已投稿状态，投稿完后可能会进入三种状态：

1.经过审稿进入已审稿状态

2.经过修改进入已修改状态

3.经过撤稿进入撤稿状态

状态分支1，进入审稿状态后，经过决定会进入三种状态:

a、经过通过进入已通过状态

b、经过拒绝进入已拒绝状态

3、经过重审进入已被审状态

状态分支a，通过后会进入已通过状态，在经过发布会进入已发布状态，已发布状态会进入两种状态：

i.经过存档进入已存档状态

ii.经过查询又会分为三种状态

状态分支i，存档后进入终止状态

状态分支ii，经过点赞进入已点赞状态，之后进入终止状态；经过评论进入已评论状态，之后进入终止状态；经过收藏进入已收藏状态，之后进入终止状态。

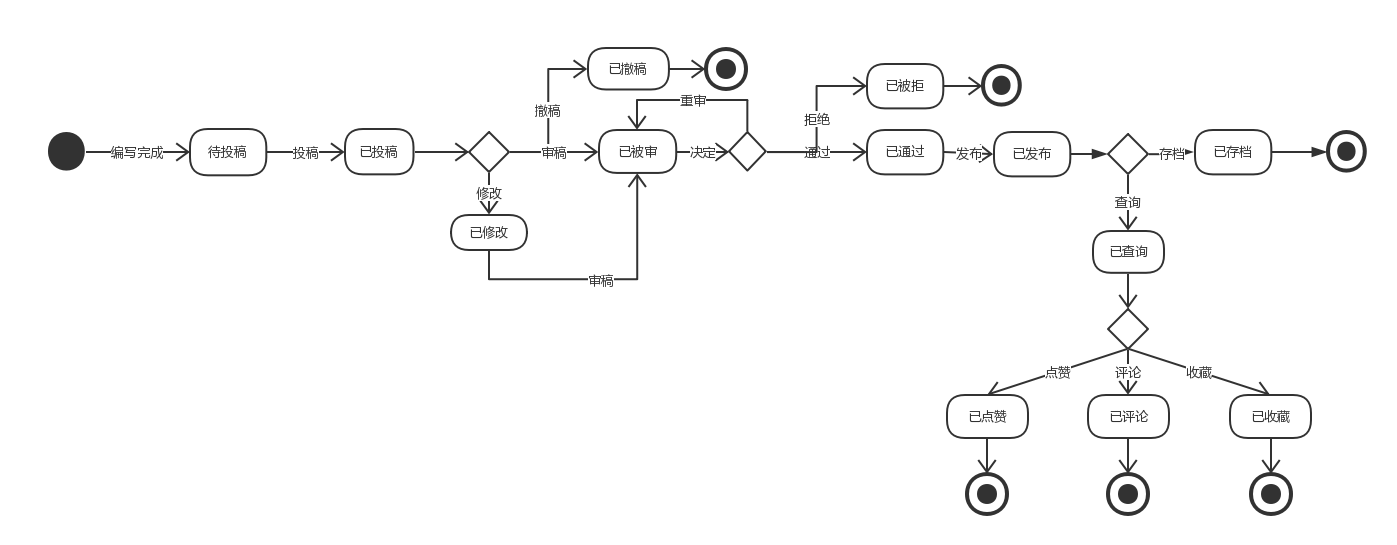
状态分支b，已拒绝状态之后直接进入终止状态

状态分支c，进入已被审状态后重新进入状态分支1

状态分支2，进入已修改状态后经过审稿进入分支1

状态分支3，进入已撤稿状态后进入终止状态

## 6.2、状态图

​​

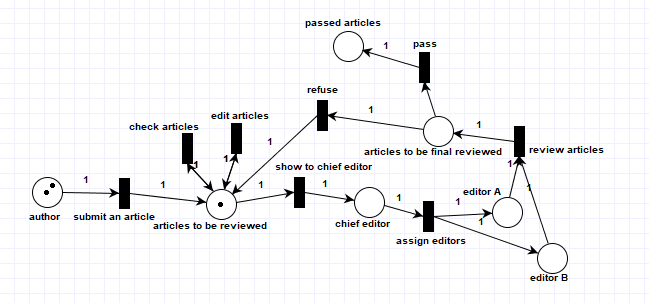
# 七、基于Petri建模

## 7.1、基于Petri网进行场景描述

该Petri网主要描述了系统的核心业务，审查流程。P生成的xml文件见附件中的petriNet.xml，该文件由 [PIPE 5.0.2](https://github.com/sarahtattersall/PIPE/releases) 生成。

|  |  |  |
| --- | --- | --- |
| **Event** | **Pre-condition** | **Post-condition** |
| submit an article | author | articles to be reviewed |
| check articles | articles to be reviewed | articles to be reviewed |
| edit articles | articles to be reviewed | articles to be reviewed |
| show to chief editors | articles to be reviewed | chief editor |
| assign editors | chief editor | editor A, editor B |
| review articles | editor A, editor B | articles to be final reviewed |
| pass | articles to be final reviewed | passed articles |
| refuse | articles to be final reviewed | articles to be reviewed |

## 7.2、流程模型



## 7.3、流程模型仿真

可以用三个Token来描述模型中的业务问题：

* 作者提交了一篇文章然后通过了
  + - author → articles to be reviewed → chief editor → editor A & editor B → articles to be final reviewed → passed articles
    - Events: submit an article, show to chief editors, assign editors, review articles, pass
* 作者提交一篇文章然后被拒绝了
  + - author → articles to be reviewed → chief editor → editor A & editor B → articles to be final reviewed → articles to be reviewed
    - Events: submit an article, show to chief editors, assign editors, review articles, refuse
* 作者查看提交的文章然后修改它
  + - articles to be reviewed → articles to be reviewed → articles to be reviewed
    - Events: check articles, edit articles

## 7.4、系统增值模式分析

游记是系统的主要价值。系统必须游记的质量和数量。越多的文章发布，就能吸引更多的用户流量；越好的文章发布，就能吸引越多的作者，形成一个良性循环，这就是为什么系统的main path是REVIEW，它涉及了所有级别用户的参与，并且是系统的核心业务。编辑和主编处理游记的效率是系统的瓶颈，因此KPI取决于主编和编辑。为了更好地管理文章，系统应该有良好的策略来存储游记并能够准确的给这些游记分类。