



(12)发明专利申请

(10)申请公布号 CN 109118065 A

(43)申请公布日 2019.01.01

(21)申请号 201810831094.5

(22)申请日 2018.07.26

(71)申请人 深圳市小赢信息技术有限责任公司

地址 518000 广东省深圳市前海港合作区

前湾一路1号A栋201室

(72)发明人 李查 李威

(74)专利代理机构 上海精晟知识产权代理有限公司

公司 31253

代理人 杨军

(51) Int. Cl.

G06Q 10/06(2012.01)

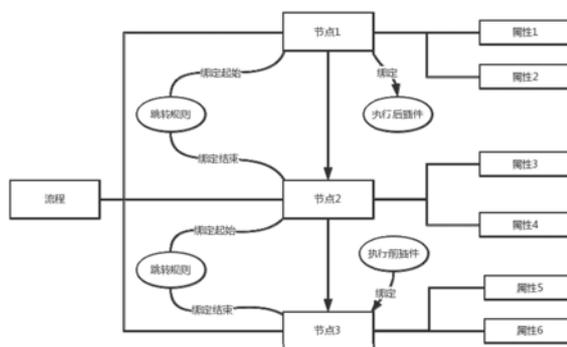
权利要求书2页 说明书7页 附图3页

(54)发明名称

一种交互式 workflow 系统及其运行方法

(57)摘要

本发明提供一种交互式 workflow 系统及其运行方法,交互式 workflow 系统包括存储模块、配置中心、发布中心、权限中心、消息中心、监控中心。本发明所述的流程一旦发起,所有的流程运转交由 workflow 系统接管,业务不需要关心其内部运行机制。流程逻辑全部采用 workflow 配置的方式,彻底释放业务压力,避免重复开发,维护过重的流程逻辑。



1. 一种交互式 workflow 系统, 包括存储模块、配置中心、流程中心、权限中心、消息中心、监控中心; 其特征是:

所述存储模块, 从流程配置到流程实例的整个生命周期内的所有数据落地, 操作记录增量保存, 方便后续流程追溯; 粒度到参数的每个流程单独存储, 对以后的数据分析统计提供基础数据;

所述配置中心, 包括: 流程配置、节点配置、参数配置、插件配置、跳转配置;

流程配置: 定义一个流程的基础信息;

节点配置: 同一个流程能可以配置多个节点, 每个节点代表流程中的一个过程;

参数配置: 可以为不同节点定义差异化参数, 从而保证不同节点的处理人只需要关心其特定参数;

插件配置: 在流程、节点的生命周期内会存在多个不同的状态, 基于其状态系统支持多种事件机制, 为对应节点的对应事件可以配置额外的业务逻辑;

跳转配置: 系统是以节点参数以及节点操作做为条件来配置节点间的跳转的, 从而实现流程的流程节点间的流转逻辑;

所述流程中心, 根据已定义的流程、传入要求的参数实例化一个流程, 实例化完成后, 流程会自动渲染一个相对应的工作流视图, 对流程进行流转操作, 系统流转路径会依据配置的跳转来实现流程流转, 继续或打回; 流程执行生命期中, 如流程初始化、节点开始前、节点开始后等, 预设的插件会根据配置的生命期执行调用, 流程结束调用消息中心通知提单人, 过程中所有的数据存储于存储模块;

存储模块、配置中心、流程中心、权限中心、消息中心、监控中心相互之间通过网络连接。

2. 一种交互式 workflow 系统的运行方法, 其特征是:

1) 定义基本结构

1.1) 定义流程, 用于区分各个流程的独立个体;

1.2) 定义节点, 用于确定流程执行步骤;

1.3) 定义节点属性, 用于充分展现每个步骤需要执行的工作视图, 一个步骤由多个属性组合;

1.4) 定义跳转规则, 用于流程继续的判断, 当条件满足时, 流程才可继续执行;

1.5) 定义插件, 用于跟使用方对接的调用钩子, 在流程不同生命期跟使用方交互;

2) 流转

2.1) 将1.2) 步中的节点添加到1.1) 步的集合中; 流程按节点设定序列执行流转;

2.2) 将1.3) 步的属性添加到1.2) 步的集合中, 控制节点展示视图;

2.3) 所述1.4) 步绑定1.2) 步的起始节点ID及结束节点ID, 同时存储判断条件;

2.4) 所述1.5) 步绑定1.2) 步的节点ID, 同时设定执行时序, 可为流程任意生命期, 任意绑定。

3. 根据权利要求2所述的一种交互式 workflow 系统的运行方法, 其特征是: 1.2) 步节点均需设置执行阶段, 设名称为seq, 值为2, 即流程的第二步; 同时seq值为2的节点可以多个, 配合跳转规则条件, 确定跳转走向。

4. 根据权利要求2所述的一种交互式 workflow 系统的运行方法, 其特征是:

1.4) 步跳转规则需设置起始节点和结束节点;

设置条件规则,当条件规则满足时,起始节点到结束节点的流转才可成功;多个节点及跳转规则可互相绑定,可并行或可单行。

5. 根据权利要求2所述的一种交互式 workflow 系统的运行方法,其特征是:

1.5) 步中插件作为流程调用的钩子,非必选;

插件触发点,按流程生命期,可分为节点执行前,节点执行后,流程开始时,流程结束时。

一种交互式 workflow 系统及其运行方法

技术领域

[0001] 本发明涉及互联网领域,尤其涉及 workflow 类后台业务系统及运行方法。

背景技术

[0002] 随着互联网技术快速发展,公司化运营趋于成熟,团队日益壮大。线上需要协同化的工作越来越多。

[0003] 以互联网产品版本发布举例,一个较大的版本发布迭代,可能涉及产品经理,前后端开发,测试,运维等各个角色。不同角色需要跟进处理的模块各不相同,操作时序各不相同,在人员调度,资源利用上将耗费大量精力,一个统一协作的操作平台显得迫切需要。当然,不同发布流程不尽相同,一个小的发布可能仅需要一两人处理,发布也就一两步,在口头、当面协作即可解决;而一个大的发布可能需要几十人,几十个步骤,扎堆协作的方式显得力不从心。那么,要支持到这样各类场景,同时还具备配置化、调度、推送、可定制化等能力,以适应变化。workflow 解决方案应运而生。

发明内容

[0004] 本发明所要解决的技术问题在于,针对现有系统的缺陷,提供一种交互式 workflow 系统及其运行方法。

[0005] 为了解决上述问题本发明的技术方案是这样的:

[0006] 一种交互式 workflow 系统,包括存储模块、配置中心、流程中心、权限中心、消息中心、监控中心;其中:

[0007] 所述存储模块,从流程配置到流程实例的整个生命周期内的所有数据落地,操作记录增量保存,方便后续流程追溯;粒度到参数的每个流程单独存储,对以后的数据分析统计提供基础数据;

[0008] 所述配置中心,包括:流程配置、节点配置、参数配置、插件配置、跳转配置;

[0009] 流程配置:定义一个流程的基础信息;

[0010] 节点配置:同一个流程能可以配置多个节点,每个节点代表流程中的一个过程;

[0011] 参数配置:可以为不同节点定义差异化参数,从而保证不同节点的处理人只需要关心其特定参数;

[0012] 插件配置:在流程、节点的生命周期内会存在多个不同的状态,基于其状态系统支持多种事件机制,为对应节点的对应事件可以配置额外的业务逻辑;

[0013] 跳转配置:系统是以节点参数以及节点操作做为条件来配置节点间的跳转的,从而实现流程的流程节点间的流转逻辑;

[0014] 所述流程中心,根据已定义的流程、传入要求的参数实例化一个流程,实例化完成后,流程会自动渲染一个相对应的工作流视图,对流程进行流转操作,系统流转路径会依据配置的跳转来实现流程流转,继续或打回;流程执行生命期中,如流程初始化、节点开始前、节点开始后等,预设的插件会根据配置的生命期执行调用,流程结束调用消息中心通知提

单人,过程中所有的数据存储于存储模块;

[0015] 存储模块、配置中心、流程中心、权限中心、消息中心、监控中心相互之间通过网络连接。

[0016] 一种交互式工作流系统的运行方法:

[0017] 1) 定义基本结构

[0018] 1.1) 定义流程,用于区分各个流程的独立个体;

[0019] 1.2) 定义节点,用于确定流程执行步骤;

[0020] 1.3) 定义节点属性,用于充分展现每个步骤需要执行的工作视图,一个步骤由多个属性组合;

[0021] 1.4) 定义跳转规则,用于流程继续的判断,当条件满足时,流程才可继续执行;

[0022] 1.5) 定义插件,用于跟使用方对接的调用钩子,在流程不同生命期跟使用方交互;

[0023] 2) 流转

[0024] 2.1) 将1.2)步中的节点添加到1.1)步的集合中;流程按节点设定序列执行流转;

[0025] 2.2) 将1.3)步的属性添加到1.2)步的集合中,控制节点展示视图;

[0026] 2.3) 所述1.4)步绑定1.2)步的起始节点ID及结束节点ID,同时存储判断条件;

[0027] 2.4) 所述1.5)步绑定1.2)步的节点ID,同时设定执行时序,可为流程任意生命期,任意绑定。

[0028] 在1.2)步节点均需设置执行阶段,设名称为seq,值为2,即流程的第二步;同时seq值为2的节点可以多个,配合跳转规则条件,确定跳转走向。

[0029] 在1.4)步跳转规则需设置起始节点和结束节点;

[0030] 设置条件规则,当条件规则满足时,起始节点到结束节点的流转才可成功;多个节点及跳转规则可互相绑定,可并行或可单行。

[0031] 在1.5)步中插件作为流程调用的钩子,非必选;

[0032] 插件触发点,按流程生命期,可分为节点执行前,节点执行后,流程开始时,流程结束时。

[0033] 有益效果,本发明所述的工作流系统,流程一旦发起,所有的流程运转交由工作流系统接管,业务不需要关心其内部运行机制。流程逻辑全部采用工作流配置的方式,彻底释放业务压力,避免重复开发,维护过重的流程逻辑。

附图说明

[0034] 下面结合附图和具体实施方式来详细说明本发明;

[0035] 图1为本发明所述的工作流系统模块设计图。

[0036] 图2为本发明所述的工作流系统流程关系图。

[0037] 图3为本发明所述的减免罚息场景示意关系图。

[0038] 图4为本发明所述的流程运转场景I关系图。

[0039] 图5为本发明所述的流程运转场景II关系图。

具体实施方式

[0040] 为了使本发明实现的技术手段、创作特征、达成目的与功效易于明白了解,下面结

合具体图示,进一步阐述本发明。

- [0041] 参看图1,要声明一个 workflow,首先需要在配置中心配置流程、节点、跳转、插件等。
- [0042] 1.之后,根据已定义的流程,传入要求的参数以实例化一个流程
- [0043] 2.实例化完成后,流程会自动渲染一个相对应的工作流视图
- [0044] 3.在工作流视图中,可对流程进行流转操作
- [0045] 4.系统流转路径会依据配置的跳转来实现流程流转,继续或打回。
- [0046] 5.流程执行生命期中,如流程初始化、节点开始前、节点开始后等,预设的插件会根据配置的生命期执行调用
- [0047] 6.流程结束调用消息中心通知提单人
- [0048] 7.过程中所有的数据存储于存储模块
- [0049] 所述存储模块,从流程配置到流程实例的整个生命周期内的所有数据落地,操作记录增量保存,方便后续流程追溯;粒度到参数的每个流程单独存储,对以后的数据分析统计提供基础数据;
- [0050] 所述配置中心,包括:流程配置、节点配置、参数配置、插件配置、跳转配置;
- [0051] 流程配置:定义一个流程的基础信息;
- [0052] 节点配置:同一个流程能可以配置多个节点,每个节点代表流程中的一个过程;
- [0053] 参数配置:可以为不同节点定义差异化参数,从而保证不同节点的处理人只需要关心其特定参数;
- [0054] 插件配置:在流程、节点的生命周期内会存在多个不同的状态,基于其状态系统支持多种事件机制,为对应节点的对应事件可以配置额外的业务逻辑;
- [0055] 跳转配置:系统是以节点参数以及节点操作做为条件来配置节点间的跳转的,从而实现流程的流程节点间的流转逻辑;
- [0056] 所述流程中心,根据已定义的流程、传入要求的参数实例化一个流程,实例化完成后,流程会自动渲染一个相对应的工作流视图,对流程进行流转操作,系统流转路径会依据配置的跳转来实现流程流转,继续或打回;流程执行生命期中,如流程初始化、节点开始前、节点开始后等,预设的插件会根据配置的生命期执行调用,流程结束调用消息中心通知提单人,过程中所有的数据存储于存储模块;
- [0057] 存储模块、配置中心、流程中心、权限中心、消息中心、监控中心相互之间通过网络连接。
- [0058] 参看图2,
- [0059] 1.一个流程对应多个节点
- [0060] 2.每个节点对应多个不同参数
- [0061] 3.流程主体、节点可以绑定插件
- [0062] 4.节点间跳转依据已配置的跳转规则
- [0063] 目前较多 workflow 跟业务耦合过重,针对现在存在的技术缺陷,本方案提供一种流程独立化,释放业务的工作流解决方法。流程一旦发起,所有的流程运转交由 workflow 系统接管,业务不需要关心其内部运行机制。流程逻辑全部采用 workflow 配置的方式,彻底释放业务压力,避免重复开发,维护过重的流程逻辑。
- [0064] 为实现上述目的,本发明提供如下方案:

- [0065] 1.首先抽象基本结构:
- [0066] 1.1定义流程,用于区分各个流程的独立个体
- [0067] 1.2定义节点,用于确定流程执行步骤
- [0068] 1.3定义节点属性,用于充分展现每个步骤需要执行的工作视图,一个步骤由多个属性组合
- [0069] 1.4定义跳转规则,用于流程继续的判断,当条件满足时,流程才可继续执行
- [0070] 1.5定义插件,用于跟使用方对接的调用钩子,在流程不同生命期跟使用方交互
- [0071] 2.根据该设计,为实现流程运转,需要采用如下方法:
- [0072] 2.1将上述1.3添加到1.1的集合中。将1.3添加到1.2的集合中。流程按节点设定序列执行流转
- [0073] 2.2上述1.4绑定1.2的起始节点ID及结束节点ID,同时存储判断条件
- [0074] 2.3上述1.5绑定1.2的节点ID,同时设定执行时序,可为流程任意生命期,任意绑定
- [0075] 3.为实现流程运转,节点配置,需要采用如下方法:
- [0076] 3.1上述1.2节点均需设置执行阶段,假设名称为seq,值为2,意思即流程的第二步
- [0077] 3.2以3.1的举例,同时seq值为2的节点可以多个,配合跳转规则条件,确定跳转走向
- [0078] 4.为实现流程运转,跳转规则,需要采用如下方法:
- [0079] 4.1上述1.4跳转规则需设置起始节点和结束节点
- [0080] 4.2设置条件规则,当条件规则满足时,起始节点到结束节点的流转才可成功
- [0081] 4.3多个节点及跳转规则可互相绑定,可并行可单行
- [0082] 5.为实现流程运转,插件采用,需要采用如下方法:
- [0083] 5.1上述1.5插件作为流程调用的钩子,非必选
- [0084] 插件触发点,按流程生命期,可分为节点执行前,节点执行后,流程开始时,流程结束时。
- [0085] 流程可为串行流程,也可为并行流程,为保证流程顺利执行,流程节点步骤间需要特定条件执行以支撑流程顺利执行,默认按节点配置的顺序执行。
- [0086] 应用举例:网贷减免罚息系统场景,涉及资金审批判断规则、审核对象不同、审核步骤不同等,业务审批运转如图3所示。根据工作流的设计方法,对不同条件判断,配置跳转规则条件,节点走向,并配置不同步骤,可以由以下结构体描述配置效果:

```
[
    'flowId' => 1 //流程 ID
    'flowCode' => 'penaltyRemit' //流程码
    'flowName' => '减免罚息' //流程名称
    'flowDescription' => '减免罚息 workflow' //流程描述
    'nodeList' => [ //节点列表
        1 => [
            'nodeId' => '9' //节点 ID
            'nodeName' => '减免罚息' //节点名称
            'nodeDescription' => '' //节点描述
            'users' => '' //节点操作人
            'role' => '' //节点操作角色
            'fieldList' => [ //属性列表
                1 => [
                    'fieldId' => '39' //属性 ID
                    'fieldType' => 'label' //属性类型
                    'fieldName' => '项目' //属性名称
                    'fieldDescription' => '' //属性描述
                    'config' => '' //属性配置
                ]
            ]
        ]
    ]
]
```

[0087]

```

        'errMsg' => '' //错误提示
        'dftValue' => '' //默认值
    ]
]
]
],
'ruleList':[ //条件规则列表
    'fromNode' => '1' //起始节点
[0088] 'toNode' => '2' //结束节点
    'condition' => '' //条件表达式
]
'pluginList':[
    'nodeId' => '9' //节点 ID
    'calltime' => 'begin' //执行时间
    'config' => '' //调用配置
]
]
]

```

[0089] 此结构为配置生成的定义接口,提供接入参考使用。一级结构分别记录流程信息,条件规则列表及插件调用列表。

[0090] 关键字段设计说明:

[0091] 1.flowId作为流程实例主键,用于实例唯一标识。

[0092] 2.flowCode用于流程场景区分。

[0093] 3.nodeList是各个节点列表,按设置的排序排列,其中包含节点ID,节点审批人,节点属性列表等

[0094] ■users节点审批人,可以在实例化时,根据公司组织架构动态生成

[0095] ■role节点审批角色,为方便起见,与users单独区分,该字段只存角色名

[0096] ■fieldList节点属性,html结构的抽象定义,包括id,name,中文名称等,用于视

图展示。Config是json结构,如select的option,radio组等

[0097] 4.ruleList表示节点间执行条件,当实例化完成后,condition为真,fromNode到toNode的执行流转即生效。为保证流程的一致性,实例化后,一般多个ruleList只有一个condition可以为真,可参考图4,5

[0098] 5.pluginList作为节点调用插件使用,可以在多个节点处理点,根据流程执行时序在不同时段调用执行

[0099] ■ calltime可以枚举为init(初始化),begin(节点流转前),end(节点流转后),final(流程结束时)

[0100] ■ 设置init,final两个值的时候,nodeId设置为空。

[0101] ■ 设置begin,end两个值的时候,nodeId必填

[0102] config:json结构,关键字可以任意定义,自行遵循即可。包括不限于设置url,执行命令行及任何可执行的调用。

[0103] 完整的实施场景,可为图4,5。

[0104] 本发明所述的工作流系统,流程一旦发起,所有的流程运转交由工作流系统接管,业务不需要关心其内部运行机制。流程逻辑全部采用工作流配置的方式,彻底释放业务压力,避免重复开发,维护过重的流程逻辑。以上显示和描述了本发明的基本原理、主要特征和本发明的优点。本行业的技术人员应该了解,本发明不受上述实施例的限制,上述实施例和说明书中描述的只是说明本发明的原理,在不脱离本发明精神和范围的前提下本发明还会有各种变化和改进,这些变化和改进都落入要求保护的本发明范围内。本发明专利要求保护的的范围由所附的权利要求书及其等同物界定。

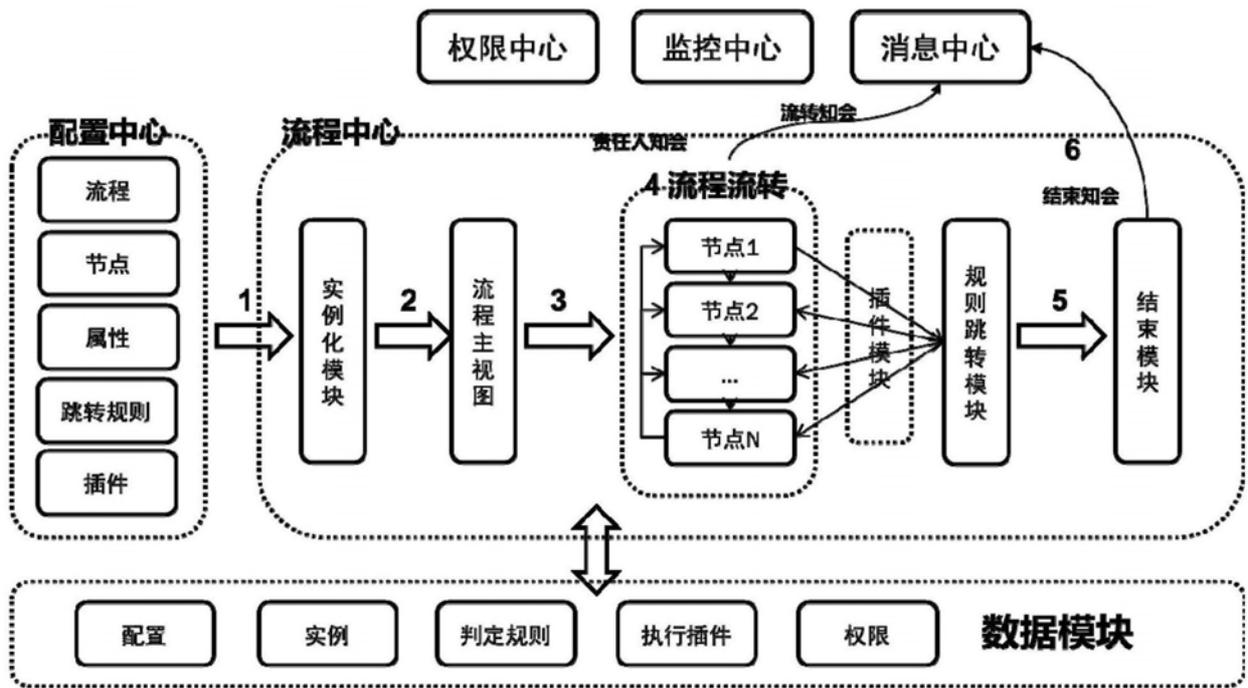


图1

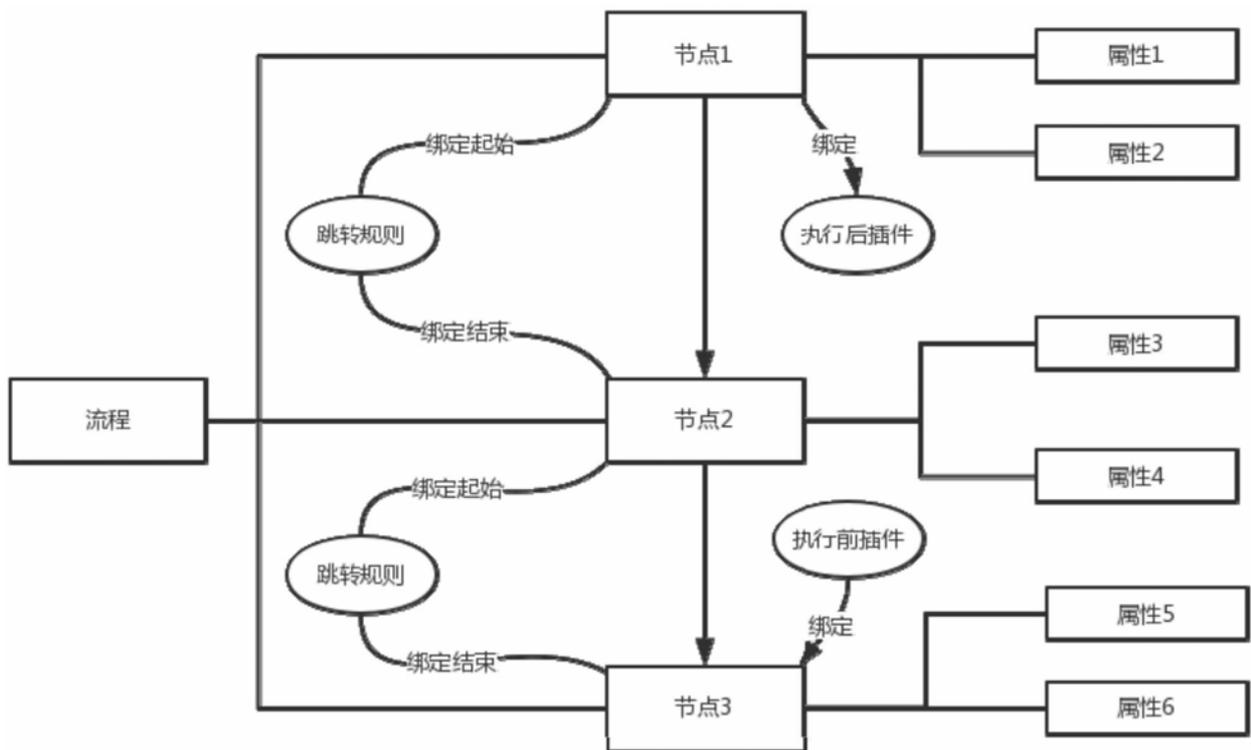


图2

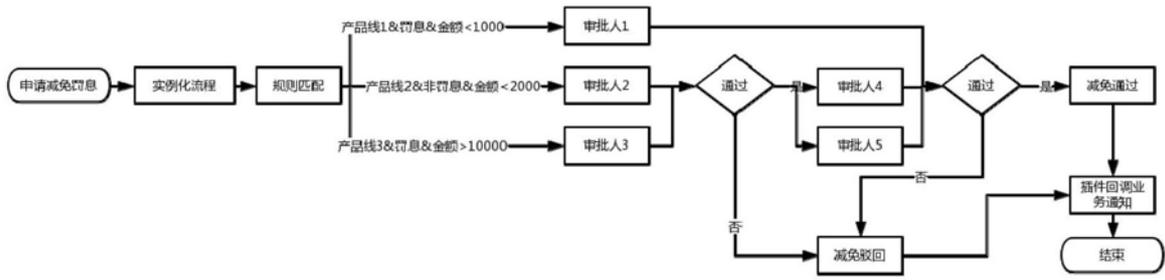


图3

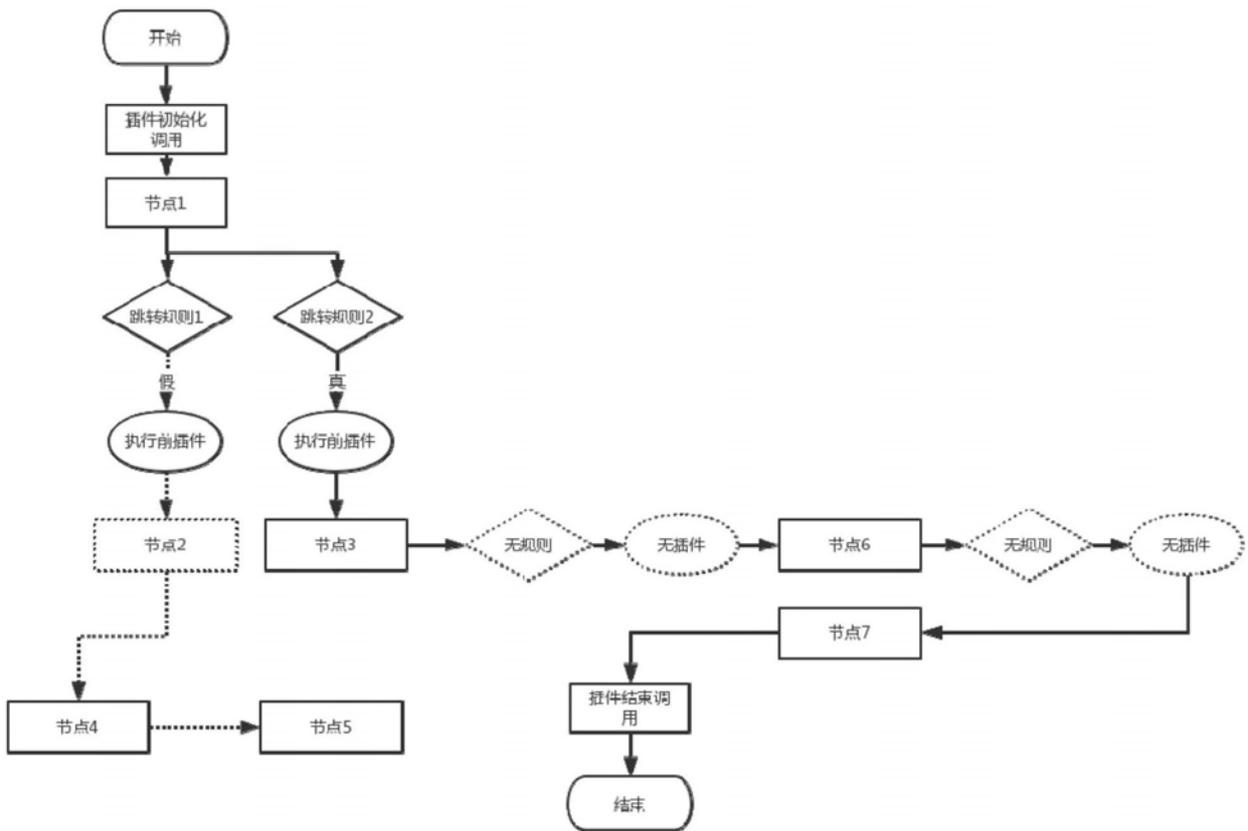


图4

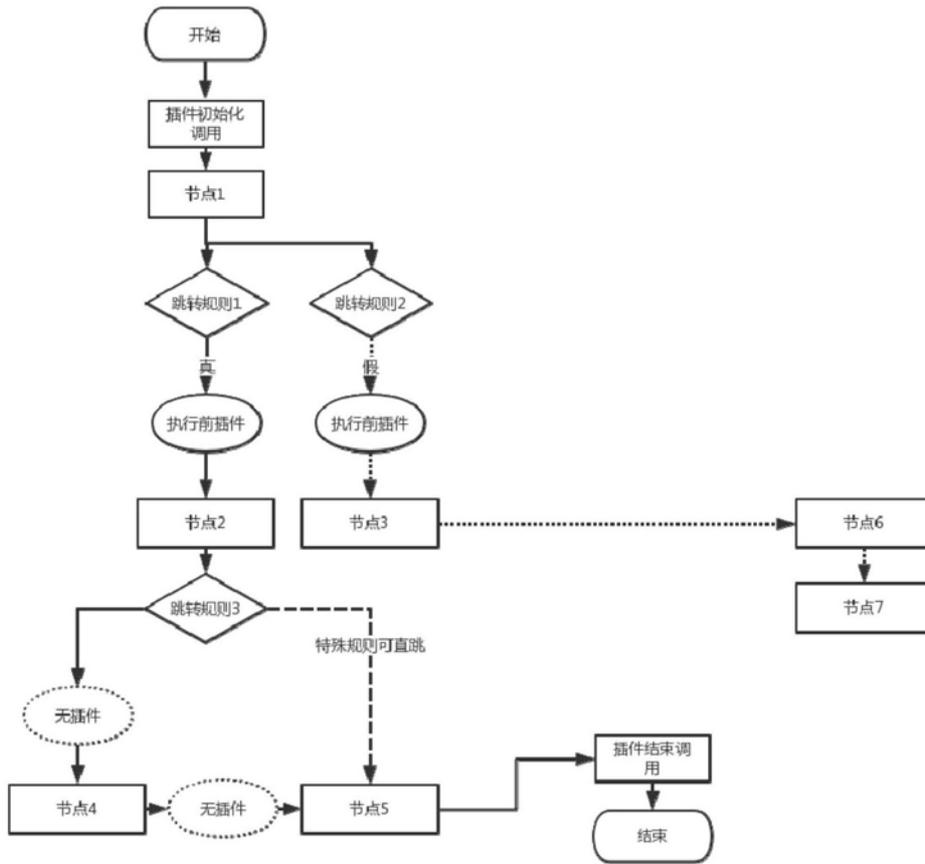


图5