**项目**

**概要设计**

**版本号：V0.1**

©

|  |  |  |  |
| --- | --- | --- | --- |
| **修订记录** | | | |
| **版本** | **修订日期** | **修订人员** | **修订摘要** |
| V0.1 | 2019-08-02 | 黄晓东 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 引言 4](#_Toc1884861368)

[1.1 编写目的 4](#_Toc1333735079)

[1.2 项目背景 4](#_Toc651165367)

[1.3 参考资料 5](#_Toc559658057)

[2 任务概述 5](#_Toc194590139)

[2.1 项目目标 5](#_Toc2006355439)

[2.1.1 业务目标 5](#_Toc1027638079)

[2.1.2 技术指标 5](#_Toc1449704579)

[2.2 需求概述 6](#_Toc1982884038)

[2.3 条件与限制 6](#_Toc1680792520)

[3 总体设计 6](#_Toc1079991002)

[3.1 处理流程 6](#_Toc876986170)

[3.2 总体架构 6](#_Toc1326289829)

[3.2.1 总体架构设计 6](#_Toc72900143)

[3.2.2 整体应用设计 7](#_Toc1167024611)

[3.2.3 系统组件清单 7](#_Toc1214489026)

[3.3 功能分配 8](#_Toc84995247)

[3.3.1 规则系统-定义界面规则（流） 8](#_Toc438491074)

[3.3.2 对接产品-装配自动生成规则文件 8](#_Toc1703087861)

[3.4 人工处理过程 8](#_Toc2135632611)

[4 接口设计 9](#_Toc535617119)

[4.1 产品组件 9](#_Toc2012954456)

[4.1.1 服务流程 9](#_Toc268167154)

[4.1.2 接口设计 9](#_Toc1664665872)

[4.2 系统公共组件 10](#_Toc622357588)

[4.2.1 服务流程 10](#_Toc1718620626)

[4.2.2 接口设计 10](#_Toc1201809032)

[5 数据结构设计 12](#_Toc1720700789)

[5.1 逻辑结构设计要点 12](#_Toc1803370221)

[5.2 物理结构设计要点 12](#_Toc1806594236)

[5.3 数据结构与程序的关系 12](#_Toc158039519)

[5.3.1 产品组件 12](#_Toc1880408141)

[5.3.2 系统公共组件 12](#_Toc1650276535)

[6 权限控制设计 13](#_Toc1446422740)

[6.1 需求 13](#_Toc512107140)

[6.2 适用对象 13](#_Toc2017728451)

[6.3 总体方案 13](#_Toc1047806180)

[6.3.1 客户用户分类 13](#_Toc1112561860)

[6.3.2 员工角色定义 14](#_Toc687066591)

[6.3.3 创建新员工流程 14](#_Toc508625018)

[7 运行设计 14](#_Toc1475762466)

[7.1 运行模块的组合 14](#_Toc1851126859)

[7.2 运行控制 15](#_Toc1293525124)

[7.2.1 客户登录运行控制 15](#_Toc1299800487)

[7.3 运行时间 15](#_Toc1543127725)

[8 出错处理设计 15](#_Toc187669256)

[8.1 出错处理对策 16](#_Toc1651191796)

# 引言

## 编写目的

本文档目的是根据各次讨论的需求来阐述系统概要，指导后续详细设计及开发。本文档明确项目的主要功能模块，数据库设计及接口概要设计。同时阐述系统错误处理方案。

本文档适读于产品-规则转换系统业务人员、需求分析人员、架构设计人员、开发人员以及系统测试人员。

## 项目背景

建信人寿现在系统不存在产品定义功能，每次产品上线均需要进行开发。开发流程包括程序新增程序代码，新增数据库模型，新增规则引擎对应的规则包等。各个部分独立进行，不但花费了更多的时间，而且容易造成各个环节需求的变形和实现的不一致，给测试、上线带来了较多异常情况。同时新产品上线的负责过程也造成公司不能很好的响应周围的变化，影响了产品上线的时间。

## 参考资料

* 产品装配相关资料;
* 规则引擎相关资料;
* 建信人寿模板。

# 任务概述

## 项目目标

### 业务目标

* 一阶段：实现产品装配参数到规则的转化；
* 二阶段：保险产品生命周期配置；
* 三阶段：分布式规则服务；
* 四阶段：生成前端判断规则。

### 技术指标

* **松耦合：**在方案设计过程中，从产品装配工厂的输出产生的规则不再做手工的修改，单向生成，无反向能力；通过中间模型适配后，依赖产品装配又具备一定的独立能力；
* **高内聚：**组件需设计成相对对立，在统一层次内每个组件需设计成支持完成多种同类型的功能；
* **重服务：**每个组件需对外提供统一的接口服务，便于组件与组件之间、层与层之间灵活调用；
* **易运维：**自动打包、自动发布、定时生效；稳定性、可靠性满足A级系统要求；流水号和错误码唯一，规则码可追溯；良好的日志跟踪机制；支持全链路监控；

## 需求概述

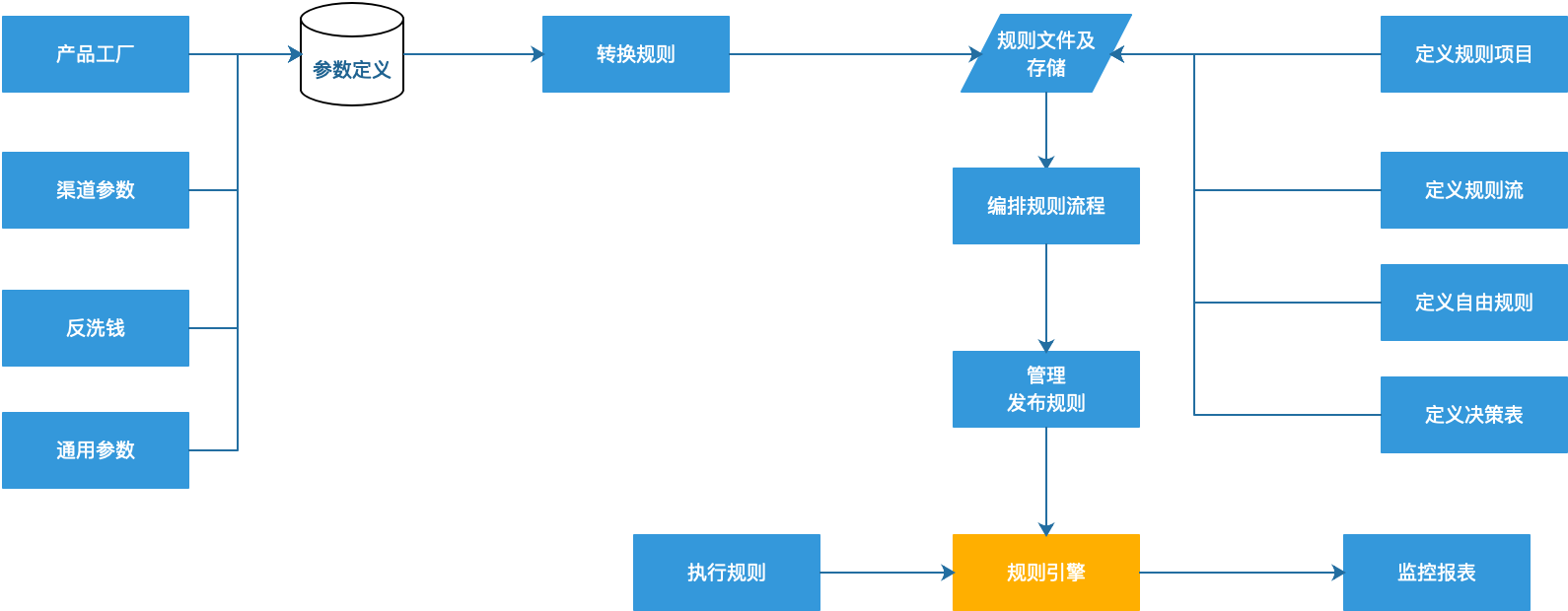
自主需求。

## 条件与限制

当前系统用户量保守估计支持100用户，10用户同时在线。

# 总体设计

## 处理流程



## 总体架构

### 总体架构设计



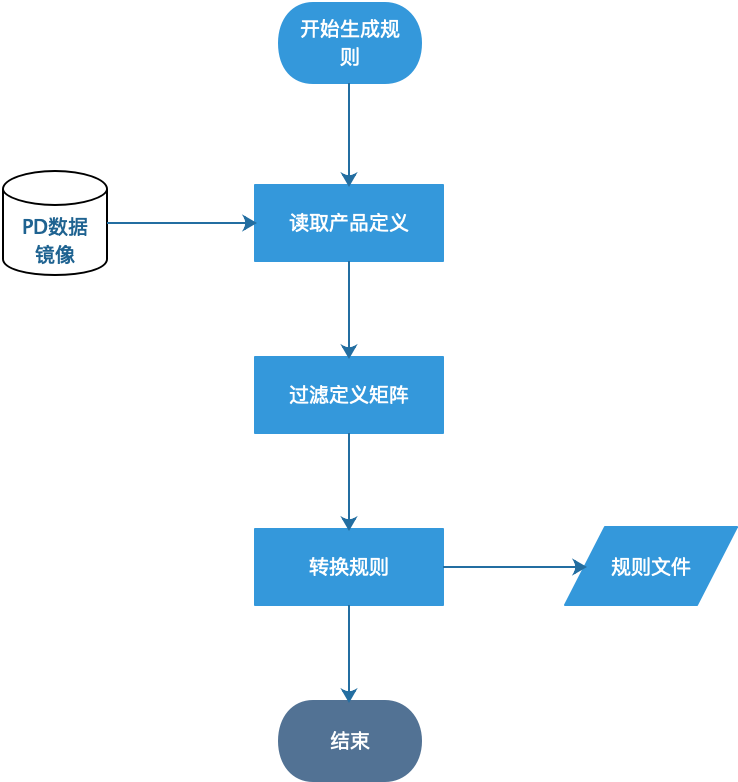
### 整体应用设计

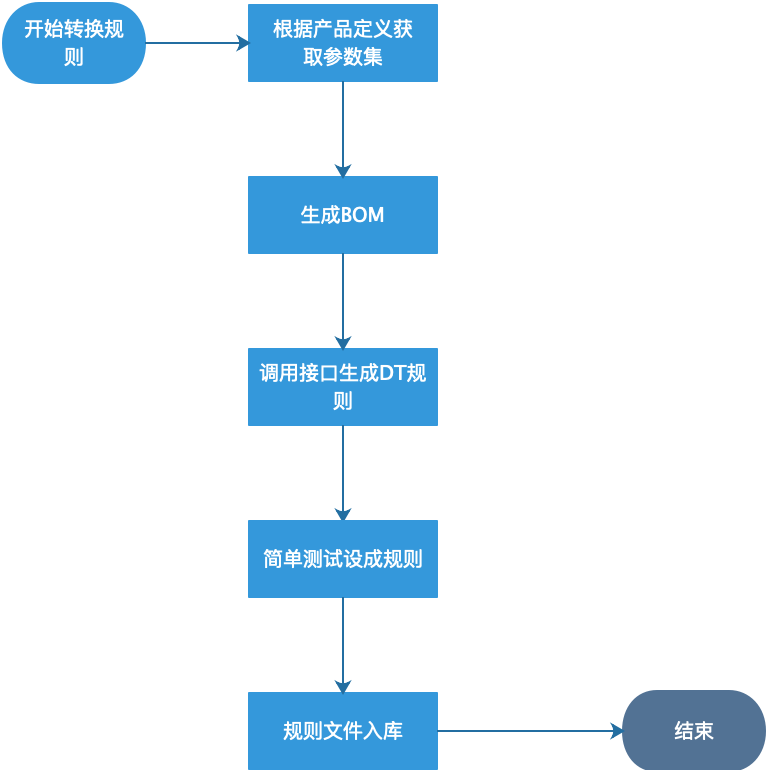
### 系统功能清单

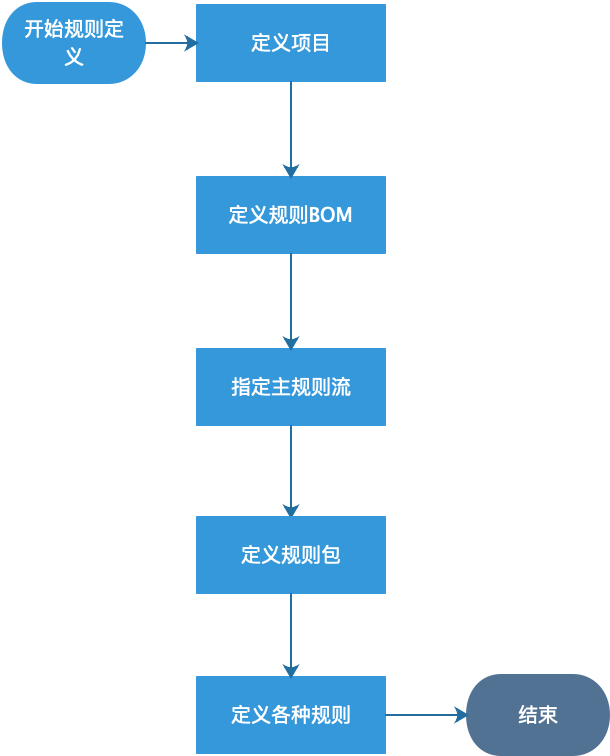
|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | | 服务 | 服务状态 |
| 服务层 | 转换规则 | 获取产品参数服务 | 已存在 |
| 标记产品参数服务 | 待开发 |
| 转换参数服务 | 待开发 |
| 获取产品销售定义服务 | 待定 |
| 获取产品渠道定义服务 | 待定 |
| 获取 |  |
| 参数转换规则服务 | 参考demo |
| 产品定义流程 | 待定 |
| 定义规则 | 定义（增删查改）项目服务 | 参考workbench |
| 定义（增删查改）规则流服务 | 参考activiti |
| 定义（增删查改）规则包服务 | 参考workbench |
| 定义（增删查改）自由规则服务 | 参考workbench |
| 定义（增删查改）规则表服务 | 参考workbench |
| 定义（增删查改）评分表服务 | 参考workbench |
| 管理发布规则 | 管理规则版本服务 | 参考workbench |
| 定义发布基线服务 | 参考workbench |
| 打包规则服务 | 参考workbench |
| 定义规则生效时间、状态服务 | 参考workbench |
| 推送规则服务 |  |
| 执行引擎 | 执行规则服务（封装多服务？） |  |
| 查询引擎状态服务 |  |
| 查询规则命中情况服务 |  |
| 手工刷新规则服务 |  |
| 查询规则仓库服务 |  |
| 查询发布计划服务 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

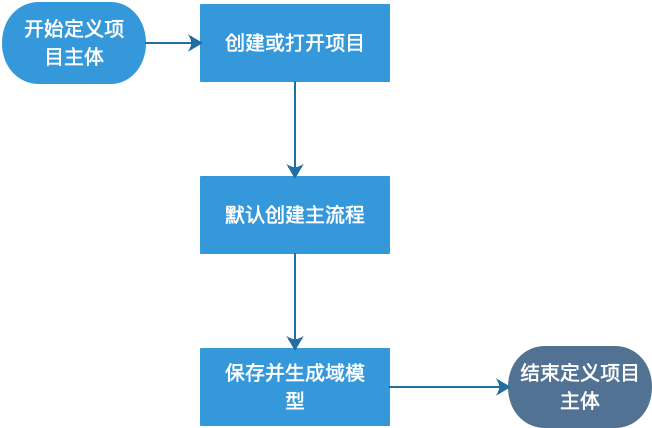
## 功能分配

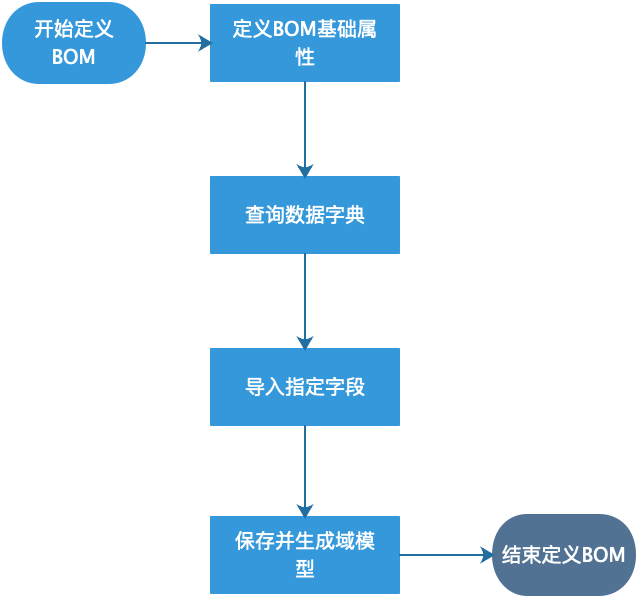
### 规则系统-转换规则

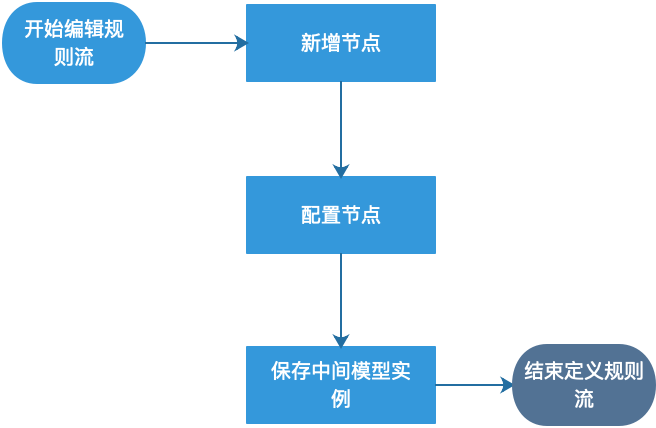












### 对接产品-装配自动生成规则文件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模块 | 子模块 | 功能点 | 功能点编号 | 服务组件 | 需求编号 |
|  |  |  |  |  |  |
|  |  |  |  |

## 人工处理过程

人工处理过程可能涉及：

* 手工批处理等。

# 接口设计

## 产品组件

### 服务流程

#### 产品配置

#### 产品模板设定

### 接口设计

#### 外部接口设计

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **接口名称** | **接口类型** | **服务说明** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

#### 内部接口设计

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **接口名称** | **接口类型** | **服务说明** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
| 20 |  |  |  |
| 21 |  |  |  |

## 系统公共组件

### 服务流程

### 接口设计

#### 外部接口

#### 内部接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **系统维护模块** | **接口名** | **备注** | **传入参数** | **返回值** |
| 系统资源 | 新增功能 | 新增一个系统功能 | 资源类型、名称等信息 | 成功或失败 |
| 启用功能 | 将该功能状态置为启用 | 是否启用 | 成功或失败 |
| 下线功能 | 将该功能状态置为不可用 | 是否下线 | 成功或失败 |
| 功能列表查询 | 查询所有功能列表 | 用户id | 成功或失败 |
| 角色管理 | 新增角色 | 新增一个角色并分配权限 | 角色名、权限信息 | 成功或失败 |
| 删除角色 | 删除一个角色，该角色下有用户则不允许删除 | 角色id | 成功或失败 |
| 修改角色 | 修改角色信息 | 角色id、角色信息 | 成功或失败 |
| 查询角色列表 | 查询所有角色信息，有权限限制 | 管理员id | 成功或失败 |
| 码表管理 | 查询码表 | 显示所有的码表数据 | 用户id | 码表数据 |
| 修改码表 | 对具体字段进行修改 | 用户id | 成功或失败 |

# 数据结构设计

## 逻辑结构设计要点

* 遵循数据E-R建模，以概念数据模型设计成数据库；
* 规范逻辑结构命名原则；
* 以规范化理论为指导，适当地修改、调整数据模型的结构。

## 物理结构设计要点

数据库物理设计过程中需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，选择一个优化方案作为数据库物理结构。在数据库物理设计中，最有效的方式是集中地存储和检索对象。确定数据库的物理结构包含下面四方面的内容：

* 确定数据的存储结构
* 设计数据的存取路径
* 确定数据的存放位置
* 确定系统配置

## 数据结构与程序的关系

### 产品组件

### 系统公共组件

# 权限控制设计

权限总体方案见 官网拓展一期—安全子项目概要设计—>权限设计说明

## 需求

本系统对用户权限控制的需求：

* 用户角色管理
* 角色权限配置
* 资源访问权限管理
* 功能操作权限管理

## 适用对象

权限控制设计适用于内部用户，包含：系统管理员、运维人员、运营人员、客服、IT等。

## 总体方案

通过对内部用户进行部门、职位等不同的角色划分，再对角色进行权限分配，权限对应了不同的资源访问限制和功能操作限制，从而实现对内部用户在访问系统时的权限控制。

### 客户用户分类

#### 客户类型

不涉及

#### 员工分类

|  |  |  |  |
| --- | --- | --- | --- |
| 员工岗位 | 说明 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 系统管理员 |  |  |  |
| 运维人员 |  |  |  |
| 安全审计员 |  |  |  |
| 开发支持人员 |  |  |  |

### 员工角色定义

见

### 创建新员工流程

见

# 运行设计

## 运行模块的组合

**系统主要以组件为模块，通过API适配方式调用，各模块之间主要通过传送数据来实现模块间的合作和数据共享。各模块之间相互独立，程序的可移植性好。**

通过下图，可了解运行模块的组合：

当系统外部访问时，最终都会访问服务层，即组件层。组件服务可以组合，支持着每种运行所历经的控制访问。

**应用组件清单:**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 系统层级 | 业务组件 | 组件说明 |
|  |  |  |  |
|  |  |  |  |

## 运行控制

### 客户登录运行控制

首先访问CDN获取静态数据显示页面，用户通过客户端登录页面登录，交易通过展现层服务器代理进入用户应用服务器，用户应用服务器调用应用组件服务器，查看该用户是否存在，如果存在缓存分布式Session，如果不存在则创建访客Session存储到缓存服务器。

## 运行时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **吞吐量**  **(秒)** | **最终用户响应时间** | **性能的严重度(高, 中, 低)** | **一年/三年内**  **最大并发最终用户数** |  |
| 1000 | 交易业务响应时间目标：交易平均响应时间 < 1秒 | 高 | 800/1200 |  |
|  |  |  |  |  |

# 出错处理设计

**第三段定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **系统名称** | **系统代码** | **子系统** | **子系统代码** | **name** |
|  |  |  |  |  |
|  |  |  |
| … |  |  |

**第四段定义：**

|  |  |  |
| --- | --- | --- |
| 系统代码 | 错误代码 | 说明（举例） |
| 520 | 00001 | 参数错误！ |
| 521 | 00020 | 类型{0}未找到！ |
| 530 | 00023 | 类型{0}的定义{1}无效！ |

## 出错处理对策