密级：

**工作流引擎设计任务书**

**编制——————**

**审核——————**

**审查——————**

**会签——————**

**会签——————**

**会签——————**

**会签——————**

**批准——————**

**无锡芯享**

**2022年5月**

[2022年5月 1](#_Toc17387)

[1、前言 2](#_Toc25428)

[2、主要功能与组成 3](#_Toc1474)

[2.1主要功能： 3](#_Toc20201)

[2.2技术架构 6](#_Toc2767)

[2.2硬件组成： 6](#_Toc22274)

[2.3软件组成 6](#_Toc24807)

1、前言

**任务来源：**传统EAP编程主要根据Scenario设计文档使用通用编程语言(C# Java)等实现相关接口规范与函数功能,但当系统规模急剧扩大时,也产生了非常多的问题如:

1. 使用传统编程语言由于不同人对不同编程语言的使用方式和习惯不尽相同,所以写出的代码可能会风格迥异甚至对统一份scenario都会出现完全不同的多种代码实现,随着项目规模的扩大这些代码可能会变得相当难以维护。
2. 代码实现体现不出scenario上所描述的流程之间的关系,使得很多时候非常让人费解也很难从代码反向看出scenario中描述的流程,这样当其中任何一个步骤需要改变时,很可能会牵一发而动全身
3. 多个SECS标准函数库的实现版本,SECS协议本是可以作为协议标准形成一个高度封装的库函数,但在这种由不同开发人工程师”自治”的代码结构中每个工程师都在实现自己的SECS协议类库而且互不兼容,这极大地破坏了整个编程高内聚低耦合的原则,对整个EAP的产品结构造成了极大的破坏，甚至出现问题时都很难修改,大大降低了系统的健壮性。
4. 极高的实施成本,由于scenario流程图和scenario具体代码之间的双向转换存在巨大的鸿沟和差异所以EAP工程师在现场实施时要费尽心思做这层极不友好的转换,使得实施成本和时间急剧增加。
5. 对于运行态的scenario只能依靠日志判断运行状态缺乏可视化和直观的图形界面展示，使得发现运行中的问题比较困难。

所有以上这些问题已经出现在了现有的EAP系统中,并已经产生了非常负面的影响。在这种背景环境下，研发工作流引擎为应对和解决这些问题的想法应运而生。

**本产品名称**：工作流引擎。

**用途：**用于对scenario流程图文档转换为scenario代码这个过程进行可视化流程设计和编程。

# **主要功能与组成**

**2.1功能概要描述:**

工作流引擎的核心任务定义为:将scenario设计流程文档中的描述细节化并使用可视化可拖拽的流程图引擎设计器将符合文档设计的流程图的形式画出,并直接由这张流程图生成易于修改和维护的scenario代码,同时具备如可插入自定义代码以提高灵活性、SECS标准类库的实现以提高可维护性、节点点亮功能、流程包打包和分组功能等等为实现这个核心功能目标围绕在其周围展开的各个子功能的设计和开发实现(即基于流程图设计界面的scenario可视化编程)

## 2.2主要功能：

1. 图形界面

(1.1)基于Canvas的工作流图形引擎(重要,自研)

(1.2)拖拽布局

(1.3)调整画布大小

(1.4)缩放

(1.5)时间旅行(前进、后退)

(1.6)网格

(1.7)导入导出

(1.8)自动排列(待实现)

(1.9)连线

(1.10)弯折功能(待优化)

(1.11)设置连接点

(1.12)移动元素

(1.13)删除元素

(1.14)元素选中功能和事件高亮

(1.15)元素和连线的状态运行时高亮

(1.16)设置跳转条件和连接终结点

(1.17)终结点设计和实现

(1.18)业务规则连线规则的设计和实现

(1.19)优化代码以方便未来扩展分组功能

(1.20)工具栏元素改进为可导入可配置的以便配置业务组件

(1.21)节点点亮功能(优化)

(1.22)布局结构的优化

1. 内核部分

(2.1)前端Parser(改进)

(2.2)生成AST时支持回环结构同时对标准化函数和接口预留可配置的选项。

(2.3)为未来需要支持的分组结构做底层的支持和改进。

(2.4)重构AST生成Parser使之更易读和易理解。

(2.5)是否采用状态机模型(设想,暂不做实施)

(2.6)增加非跳转普通代码节点的支持组件

(2.7)是否未来自行研发scenario生成的DSL以及其他扩展功能(设想,暂不做实施)

(2.8)后台Parser(增强、改进),仍然围绕回环结构的支持展开

(2.9)抽象语法树生成 AST Generator(增强、改进)

(2.10)增强非跳转case 的代码节点的支持

(2.11)业务函数生成支持

(2.12)是否采用状态机模型(邻接表or switch待讨论)

(2.13)同步和异步代码如何解决生成问题对接第三方系统对接适配器

(2.14)模板代码以何种方式允许用户自定义和修改处理

(2.15)DSL生成支持是否启用如(dot语言,未想好暂不做实施)

(2.16)对未来会支持的流程包和分组功能需要设计和预留接口需改进现有功能和提前设计如何结合生成.

(2.17)抽象语法树业务处理AST Business Process(上述问题也需要同步处理一边)

(2.18)代码生成Code Generator(上述问题也需要同步处理一边)

以上引擎的核心改进在于支持回坏结构和对分组结构预留接口同时做一定程度的重构(优化设计)和整理。

1. 业务部分

(3.1)工作流对象元素的新增向导

(3.2)工作流左侧工具栏元素的配置向导

(3.3)工作流左侧工具栏元素的维护(CRUD)

(3.4)工作流其他的方便用户使用的CRUD用户图形界面功能和界面体验改进(待补充)

1. SECS标准库函数部分(标准化重要)

(4.1)控制元素如开始结束分支元素if、switch元素等的调整(优化)

(4.n)S\*F\*的实现(此处每个函数的实现都有工作量所以需要列全,待补充)

1. 第三方系统接口部分(标准化重要)

第三方接口适配器的定义和抽象以及设计如何和EAP对接

1,同步调用

2,异步消息调用

具体第三方系统接口:

MES

FDC..(此处每个函数的实现都有工作量所以需要列全,待补充)

## 2.2技术架构：

根据工作流引擎的核心功能定义可知,该引擎等价于一个基于流程图设计器的可视化变成系统内核。

由于整个可视化编程已经有些先驱在尝试如(Google blockly)

所以参考其架构思想得知实现该系统本质核心上即为实现一个类编译器的代码转换结构将描述流程图的代码转换为scenario代码

即为一个从图到scenario代码的简易编译(代码转换)结构

其详细说明和工作原理参看《[代码生成及转换规则](代码生成及转换规则.docx)》以及《[结构示意图](结构示意图.pdf)》,此处不再赘述。

## 2.2硬件组成：

本文档讨论的核心系统主要由EAP和Workflow以及这两个系统所涉及的平台硬件(如数据库,Redis)等共同组成，此处不展开说明。

## 2.3软件组成

本文档讨论的核心系统主要由EAP和Workflow以及这两个系统所涉及的平台软件(如数据库,Redis,仿真模拟软件)等共同组成，此处不展开说明。