**通信接口及协议配置组件设计方案**

# 引言

## 应用背景

在业务系统开发过程中经常遇到的类型定义、接口通信、协议解析等重复而繁琐的开发工作，这些内容处理起来不仅费时，而且经常容易出错，修改维护也比较困难。通信接口及协议配置组件正是针对这一难题，将繁复的开发任务进行自动化处理；只需输入少量配置信息便可自动生成指定的通信代码库，通信代码库将受控对象的类型定义、接口驱动调用、通信报文格式解析等技术细节进行内部封装，对外提供一组简单统一的API接口供业务系统开发时使用。

通信接口及协议配置组件（以下简称“配置系统”）是一套专用于实现快速创建软件系统底层通信模块的工具软件，配置系统主要用来实现：创建配置信息，并根据创建的配置信息自动生成通信代码库，通信代码库将数据输入输出操作封装成受控对象的方法提供给业务系统使用。在开发业务系统时，开发人员只需直接使用通信代码库中的API接口（受控对象的方法）即可实现完整的数据通信功能，而无需直接处理通信协议的底层细节。使用配置系统即省去许多繁复的开发工作，又增强了业务系统的健壮性与可维护性，正真使开发人员更集中精力去处理系统的业务逻辑。

## 定义

* 配置系统

配置系统是指实现配置信息录入、编辑，及自动生成通信代码功能的软件系统的统称。配置系统的输入内容是配置信息，输出内容是通信代码库。

* 配置项

配置项是按照自定义格式书写的配置代码，指配置信息中的条目，每个项目的配置信息均由一系列配置项组成，每一配置项均与面向对象中的域对应。

* 通信代码库

通信代码库是配置系统的输出内容，是一组C#或C++源代码，对外有统一的API可供访问，内部实现了完善的通信要素管理。

* 受控对象

受控对象用以指代业务系统中的功能组件，是实现独立功能的分系统；受控对象拥有自己的属性，并使用通信接口资源进行数据报文收发。受控对象是配置系统中的核心处理对象。

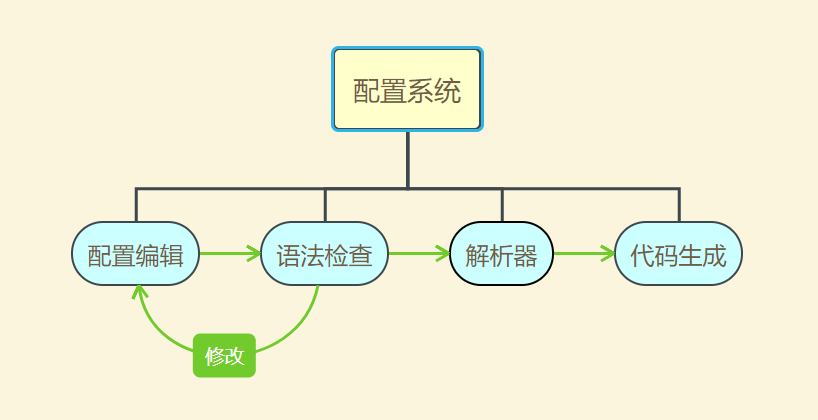
* 数据帧

数据帧是按照报文格式排列的一段数据内容，是受控对象进行数据通信时的最小数据单位。

# 总体架构

## 功能模块

配置系统由四个功能模块组成，分别为：配置编辑模块，语法检查模块、解析器和代码生成器。



配置系统功能模块

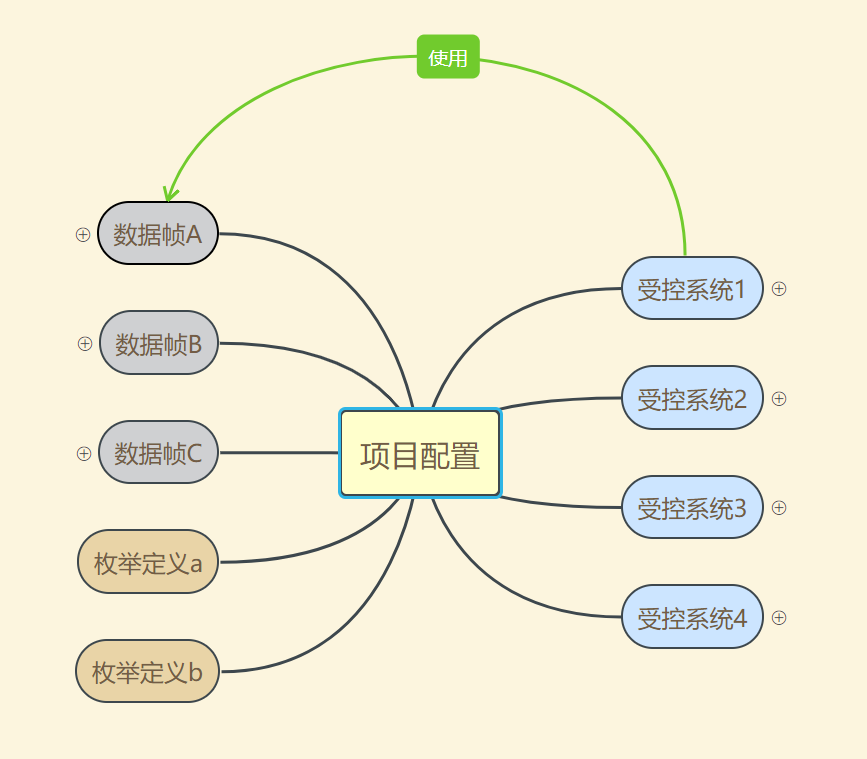
* 配置编辑模块

配置编辑模块用于录入配置信息，录入配置信息的方式分为可视化编辑和代码编辑两种可选模式，两种模式可以互相切换。

* 语法检查模块
* 解析器
* 代码生成器

## 配置信息分类

每个项目的配置信息都是一个文本格式的文件，由一组自定义格式的代码编写而成，配置信息由三大类配置项组成，分别为受控系统定义、数据帧定义和枚举定义。



配置信息的顶层分类

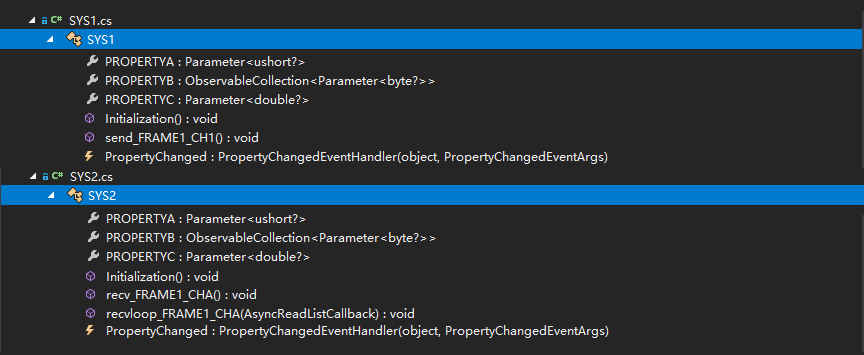
* 受控系统定义
* 数据帧定义
* 枚举定义

## 配置信息处理流程

## 通信代码库API

通信代码库由配置系统自动生成，通信代码库将数据通信中的需要处理的诸多要素和环节进行了隐藏和封装，仅通过简单的API调用便可实现通信功能；通信代码库的API以面向对象的形式进行封装，每个受控对象对应一个分系统类，受控对象的状态值对应分系统类的属性，输入输入输出动作对应分系统类的方法，而当受控对象状态值发生改变时，会自动触发相应的消息事件。

通信代码库API示例如下：



通信代码库API

* 属性
* 方法
* 事件

# 详细设计

## 配置项详细设计

* + 1. 分系统配置项
* 属性配置项
* 通道配置项
* IO配置项
  + 1. 数据帧配置项
* 字段配置项
* 字段类型
* 字段操作

## 关键处理环节

* + 1. 语法与语义检查
    2. 配置项解析
    3. 输出代码

## 通信代码库

* + 1. 分系统输出代码
    2. 数据帧输出代码

# 系统实现

## 开发与运行环境

## 关键技术路线

## 质量控制

基于成熟类库解析实现

异常处理机制

# 关于凯云

# 实操示例