|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 密级: |  | | 阶段: |  | |
| 发动机异常状态分析与评估系统  软件设计方案 |
| 文件编号：HM2620\_SDD\_V1001  现行版本：V1001  总 页 数：49 |
| 北京旋极信息技术股份有限公司 |
|  |

|  |
| --- |
|  |
| 发动机异常状态分析与评估系统  软件设计方案 |
|  |
| |  |  | | --- | --- | |  |  | | 编 制： |  | | 审 核： |  | | 会 签： |  | |  |  | | 标 准 化： |  | | 批 准： |  | |  |  | |

更改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目 | 签名 | 日期 | 产品型号及名称 | （图册编号） | |
| 设计 |  |  |  |  |
| 校对 |  |  |  |  |
| 审核 |  |  | 第张 | 共张 |
| 标准化 |  |  | 空司通信修配厂制 | |
| 批准 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 更改日期 | 更改方法/内容/原因 | 更改人 | 批准 |
| V1000 | 2023/11/14 | 创建 | 林枫 |  |
| V1001 | 2023/12/12 | 根据上会专家意见修改完善 | 林枫 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 次

[1 主题内容与适用范围 1](#_Toc153956442)

[2 规范性引用文件 1](#_Toc153956443)

[3 术语、定义和符号 1](#_Toc153956444)

[4 概述 1](#_Toc153956445)

[4.1 系统概述 1](#_Toc153956446)

[4.2 运行环境 2](#_Toc153956447)

[4.2.1 硬件环境 2](#_Toc153956448)

[4.2.2 软件环境 3](#_Toc153956449)

[4.3 软件设计目标 3](#_Toc153956450)

[5 软件设计 4](#_Toc153956451)

[5.1 设计原则 4](#_Toc153956452)

[5.2 软件架构设计 4](#_Toc153956453)

[5.2.1 应用层 6](#_Toc153956454)

[5.2.2 决策生成层 6](#_Toc153956455)

[5.2.3 诊断评估层 6](#_Toc153956456)

[5.2.4 状态监测层 6](#_Toc153956457)

[5.2.5 数据预处理层 6](#_Toc153956458)

[5.2.6 数据采集层 6](#_Toc153956459)

[5.2.7 数据库 6](#_Toc153956460)

[5.3 软件接口设计 6](#_Toc153956461)

[5.3.1 采集装置与终端上位机接口 7](#_Toc153956462)

[5.3.2 运行软件内部接口 7](#_Toc153956463)

[5.4 嵌入式运行软件功能模块设计 8](#_Toc153956464)

[5.4.1 传感器数据采集模块 8](#_Toc153956465)

[5.4.2 特征提取模块 11](#_Toc153956466)

[5.4.3 故障诊断模块 12](#_Toc153956467)

[5.4.4 数据监控模块 13](#_Toc153956468)

[5.5 上位机运行软件功能模块设计 16](#_Toc153956469)

[5.5.1 状态监测模块 16](#_Toc153956470)

[5.5.2 异常状态评估模块 19](#_Toc153956471)

[5.5.3 决策生成模块 20](#_Toc153956472)

[5.5.4 人机界面 22](#_Toc153956473)

[6 数据库设计 26](#_Toc153956474)

[6.1 数据库接口 26](#_Toc153956475)

[6.2 数据库表设计 27](#_Toc153956476)

[6.2.1 组织层级表 27](#_Toc153956477)

[6.2.2 系统设计表 32](#_Toc153956478)

[6.2.3 数据输出表 34](#_Toc153956479)

[6.2.4 人员器件表 39](#_Toc153956480)

[6.2.5 位置信息 43](#_Toc153956481)

[7 软件研制流程 45](#_Toc153956482)

[7.1 数据采集 46](#_Toc153956483)

[7.2 训练算法模型 47](#_Toc153956484)

[7.3 开发平台构建 48](#_Toc153956485)

发动机异常状态分析与评估系统软件设计方案

# 主题内容与适用范围

本文档用于对“发动机异常状态分析与评估系统”运行软件的总体设计方案进行阐述，软件开发人员在软件研制的各个阶段和相关测试均应遵循本文档进行设计。

# 规范性引用文件

技术开发合同-发动机异常状态分析与评估系统.docx

OSA-CBM UML Specification 3.3.1 Release – June. 29, 2010

CRIS\_V3-2-3\_Documentation.doc

# 术语、定义和符号

OSACBM：Open System Architecture for Condition Based Maintenance 开放式健康管理架构体系

PHM：Prognostics and Health Management 故障诊断与健康管理

# 概述

## 系统概述

发动机异常状态分析与评估系统的运行软件（以下简称“异常状态检测软件”）主要针对某装备发动机系统，通过采集安装在发动机设备上的传感器数据，来实现系统异常状态检测与故障诊断功能。

异常状态监测软件包括嵌入式运行软件和上位机运行软件两部分。

其中，嵌入式运行软件安装在装备上的数据采集装置中，首先通过传感器线缆获取发动机的实时传感器数据（缸体振动信号、喷油压力信号等），然后通过数据处理模块对传感器数据实现滤波、去噪、提取特征等数据预处理功能，同时将数据存储在本地数据库中。

上位机运行软件安装部署在管理终端中，通过以太网与数据采集装置连接。主要通过以太网获取处理后的数据，通过智能算法实现发动机系统的故障诊断与异常状态评估功能；同时，通过人机交互界面实现图形化健康状态展示、故障告警、维修维护等功能。

对于故障诊断算法，后续可根据实际情况选择安装在数据采集装置中。

系统的整体连接示意图如下所示：



1. 系统连接图

## 运行环境

### 硬件环境

数据采集装备的硬件配置如下表所示：

1. 数据采集装置配置表

|  |  |  |
| --- | --- | --- |
| 序号 | 模块名称 | 数量 |
| 1 | CPU iMX6Q | 1 |
| 2 | 硬盘8G | 1 |
| 3 | 内存1G | 1 |
| 4 | CAN卡 | 1 |
| 5 | 网卡 | 1 |

管理终端的硬件配置如下表所示：

1. 综合管理终端配置表

|  |  |  |
| --- | --- | --- |
| 序号 | 模块名称 | 数量 |
| 1 | CPU | 1 |
| 2 | 硬盘 | 1 |
| 3 | 内存 | 1 |
| 4 | 显示器 | 1 |
| 5 | CAN卡 | 1 |
| 6 | 网卡 | 1 |

### 软件环境

系统的软件环境如下：

1. 操作系统：国产麒麟系统或windows系统；
2. 开发环境：QT Creator4.4.1；
3. 软件开发语言：QT5.9.2，C++和C语言；
4. 数据库：MySQL数据库。

## 软件设计目标

异常状态检测软件采用标准的软件结构和框架，简化了不同软硬件构件的集成过程。软件标准化设计提供了一个集成多种互异构件的方法，并且通过规范构件之间的输入输出，使集成的过程变得更容易。软件基于OSA-CBM标准以及OSA-EAI-CRIS数据体系，实现软件分层的标准化，功能模块的标准化，模块间接口的标准化以及数据库设计的标准化。

整个异常状态检测软件实现的功能主要包括：

1. 能实现基于事件的状态监控，采集发动机的有效工作数据；
2. 满足OSA-CBM模块接口标准、支持web-service跨平台通信；
3. 以图形化方式分级显示健康状态；
4. 具有故障告警显示功能；
5. 具有维护建议显示功能；
6. 具有定时维护提醒功能；
7. 具有历史数据检索功能；
8. 具有数据曲线显示功能。

本项目软件主要开发工作主要包括以下部分：

1. 开发基于传感器数据采集解析、预处理的功能模块；
2. 通过台架、实车获取数据，通过算法工具箱软件训练算法模型；
3. 开发用于发动机异常状态检测的评估算法；
4. 开发用于发动机故障诊断算法；
5. 开发人机交互界面以及数据库功能模块；
6. 通过开发平台软件集成构建运行软件。

# 软件设计

## 设计原则

异常状态检测软件结合上述系统环境和自身的功能需求，基于以下设计原则设计实现：

1. 标准化

异常状态检测软件的设计基于OSA-CBM开放式健康管理体系架构，实现软件架构的标准化，功能模块的标准化，模块间接口的标准化。

异常状态检测软件按照分层软件架构模式将软件的功能模块按照不同层来划分，每一层实现该层对应的功能，数据自下而上逐层传递；同时，异常状态检测软件内每层的功能模块同样基于标准化设计，各功能模块通过统一的模块配置文件定义模块输入、输出以及模块内算法信息，模块间通过标准通信接口实现数据通信。

1. 跨平台

异常状态检测软件的跨平台设计主要基于QT自身的跨平台特性，可以确保在源代码一致的情况下基于不同的操作系统都能够成功编译生成目标代码；同时，在异常状态检测软件实现过程中，通过获取当前操作系统相关信息，区分处理不同操作系统需要用到的库文件、函数等，来实现在跨平台过程中软件代码的一致性。

异常状态检测软件目前设计运行在国产麒麟操作系统，基于跨平台的设计思想可以保证在源代码不变的情况下完成到Windows、Linux等其他操作系统的移植工作。

3）高可配置性。

高可配置性是指可以异常状态检测软件可根据不同的需求和环境灵活配置，异常状态检测软件的模块加载、模块输入源的信息、处理输入数据采用的算法描述、输出列表、输入输出规范，这些都可以在配置文件中进行定义。异常状态检测软件在运行初始化阶段加载配置文件，并根据配置文件自动设置系统运行参数和数据处理路径。

基于高可配置性的设计原则，异常状态检测软件可以通过系统配置灵活、方便的删减软件需要加载的功能模块，有利于后续软件的扩展和更新。

## 软件架构设计

异常状态检测软件使用QT编程语言开发，数据库采用开源MySQL数据库。异常状态检测软件基于模块化、面向对象的软件设计思想，在整个软件系统中，根据功能的不同将软件划分为不同的功能层，每个功能层中的模块在软件启动后独立运行，各功能模块间通过对外接口实现模块间的数据交互。异常状态检测软件的整体架构如下图所示：



1. 系统整体架构图

该分层架构设计可以解决系统所需功能，不同层数据将保存到数据库中，同时在应用层显示状态监控数据和历史数据查询。

该设计可让数据处于流动状态，数据由下而上流动。由数据采集层采集原始数据；原始数据流转到特征提取层，该层对原始数据进行特征提取处理；提取出的特征数据流转到状态检测层，生成事件和告警；生成的事件和告警数据首先流转到应用层显示，同时数据还将流转到诊断评估层，对生成的事件和告警数据进行诊断评估；诊断评估结果流转到维护建议层，对评估结果生成维修建议。每个层流转的关键数据将保存到数据库中，在应用层数据管理界面中可进行查询。整个流程形成闭环管理。

### 应用层

应用层设计用于解决人机交互问题，位于所有模块顶层。

显控界面包括图形化健康状态显示、数据管理、维修维护管理、系统管理等。其中，健康状态显示主要用于显示当前多任务系统的实时状态，包括：关键监控参数、故障告警、健康状态、设备基本信息等；数据管理界面主要显示当前多任务系统的历史数据信息，包括：历史采集数据、故障数据、维修维护数据等；维修维护管理主要用于显示定时维护提醒、填写维修履历等；系统管理用于实现日志管理、用户管理等系统配置功能。

### 决策生成层

维护建议生成模块主要依据故障诊断和健康评估的结果，生成对应的维修维护建议。异常状态检测软件的维修维护建议是文字描述。

### 诊断评估层

基于数据故障诊断模块，根据实装数据和台架数据，训练算法相关数据模型。在此基础上按照模型和相关故障诊断算法进行推理和诊断。

基于全工况的实车正常数据，实现发动机异常状态的评估。

### 状态监测层

事件生成，依据配置多任务系统的监控数据和相关逻辑关系配置，生成相关事件信息，生成日志保存到本地数据库中。

异常告警，依据配置的模块输出端口阈值生成异常告警（可根据严重程度划分分级）。

### 数据预处理层

主要基于采集的传感器数据，实现滤波、去噪等处理。通过对应的特征提取算法提取时域、频域或者时频域上的特征值，为诊断和评估提供数据支持。

### 数据采集层

传感器数据采集主要基于加装传感器和数据采集设备获取传感器的实时波形数据。

### 数据库

主要基于CIRS数据标准来实现数据库结构设计，包括系统组成相关信息表，各功能模块输出数据存储表，维修维护等工作管理相关数据表以及其他相关配置信息表。

## 软件接口设计

对于异常状态检测软件的内部接口，主要包括两种类型的接口：

1. 嵌入式运行软件和上位机运行软件间的接口：用于软件间实现数据交互；
2. 运行软件内部的接口：模块间传递数据。

### 采集装置与终端上位机接口

1. 接口类型：Webservice
2. 接口1：

方向：嵌入式软件→终端软件

功能：发送预处理数据；

数据内容：预处理数据；

1. 接口2：

方向：嵌入式软件→终端软件

功能：发送传感器原始数据；

数据内容：振动、压力波形数据。

### 运行软件内部接口

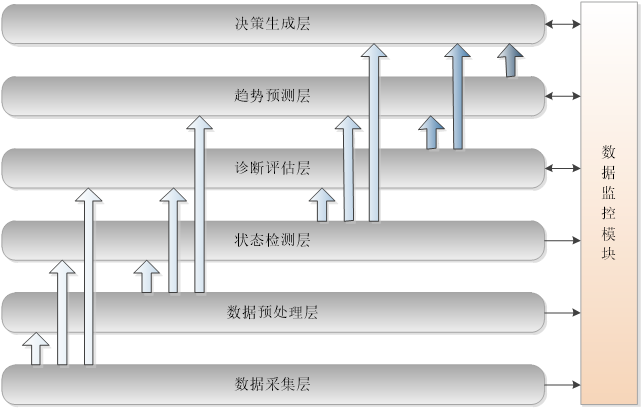
在异常状态检测软件中，运行软件中各功能块输出的数据设计封装为标准数据结构，通过标准数据通信接口实现。模块内在接收到输入数据后，通过解析数据中的数据值传递给算法输入，算法完成运算后，将算法的输出结果封装成标准数据发送出去。

各类功能模块的数据内容如下表所示：

1. 内部数据类型列表

|  |  |  |
| --- | --- | --- |
| **数据层级** | **特定内容** | **内容描述** |
| 采集数据 | 当前数值 | 包括int、char和blob三种数值。其中blob数据用于存放大数据块类型的数据，例如音视频、波形数据等。 |
| 预处理数据 | 当前数值 | 包括int、char和blob三种数值。 |
| 状态监测数据 | 状态事件  事件类型  发生时间  结束时间 | 输出系统状态信息。如系统上电，包括名称、上电时间、断电时间以及类型。 |
| 故障诊断数据 | 模糊组id  故障发生时间 | 模糊组id通过关联FMECA表示一个或多个可能故障的设备 |
| 健康评估数据 | 设备id  健康等级  健康度 | 表示对应设备的健康分数（0-1）和等级。不同的健康等级对应不同的维修维护策略。 |
| 决策数据 | 设备id  决策产生时间  决策描述  工作流程id | 表示对应的设备的维修维护建议。其中，工作流程id用于关联对应的维修维护工作内容，包括工作步骤、所需资源等信息。 |

各类功能模块见的接口示意图如下表所示：

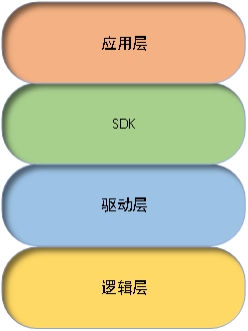


1. 模块内部接口图

## 嵌入式运行软件功能模块设计

### 传感器数据采集模块

传感器的采集需要实现对采集设备的驱动初始化、按照配置信息设置通道参数、采集数据预处理等功能，传感器的采集系统整体的结构设计按照分层的方式去实现，具体如下图所示：



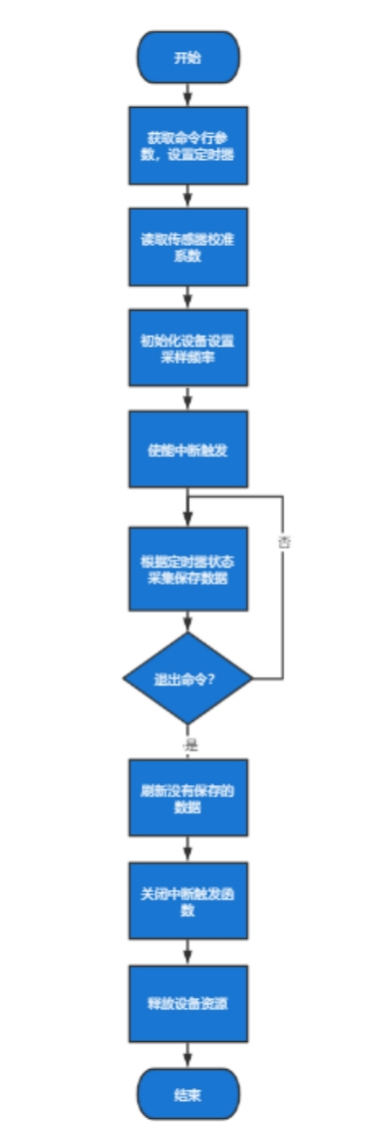
1. 数据采集分层结构图

传感器数据采集功能主要分为两部分功能：

1. 底层软件
2. 应用层功能模块

其中，底层软件主要包括逻辑层、驱动层、SDK层程序。逻辑层主要实现硬件的逻辑驱动，驱动层实现与硬件交互，SDK实现对ADC和驱动层的接口，将基础功能封装成方便调用的接口函数。应用层软件通过调用SDK提供的接口实现自检、采集配置、数据采集、校准功能、指示灯控制等功能。

底层传感器数据采集以指定的采样速率，以时间或采样点数为依据设定工作状态，并保存采集数据到分配的内存中。流程如下：



1. 数据采集流程图

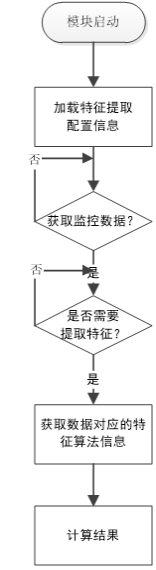
对于发动机系统的数据采集装置，应用软件启动后首先加载采集配置文件，读取当前的采集配置信息，包括：采集通道号、采样率、采样周期、传感器类型等信息；然后根据采集配置信息，通过底层API接口获取传感器信号数据。

本项目中传感器采集的信号为电信号，因此在获取到传感器信号数据后，需要根据传感器类型进行单位转化。例如：振动数据转化为加速度单位，温度转换为摄氏度等。转化后封装为标准数据格式，机械设备采集的原始数据使用。

### 特征提取模块

在嵌入式运行软件中，将获取到的原始传感器数据进行特征提取，用于后续故障诊断和评估使用。数据的特征提取主要是为了能够根据特征提取算法获取数据的故障或性能的表征，对于不同的设备和数据，需要不同的特征提取算法才能提取到最能表现故障或性能的指标。

对应发动机系统监控的振动、压力等信号，需要配置对应的特征提取算法和算法参数信息。存储在特征提取模块对应的算法配置文件中。当模块启动后，加载算法配置信息。获取监控数据后，根据配置信息中的定义加载对应的特征算法来计算，得出对应的特征值。特征提取的流程图如下所示：



1. 特征提取流程

当前嵌入式运行软件中，主要可以配置如下特征提取算法：

特征提取算法列表

|  |  |  |
| --- | --- | --- |
| **序号** | **算法类型** | **算法名称** |
| 1 | 时域分析 | 平均值 |
| 2 | 均方值 |
| 3 | 方差 |
| 4 | 有效值 |
| 5 | 峰值指标 |
| 6 | 脉冲指标 |
| 7 | 裕度指标 |
| 8 | 歪度指标 |
| 9 | 峭度指标 |
| 10 | 频域和时频域分析 | 快速傅里叶变换 |
| 11 | 短时傅里叶变换 |
| 12 | 小波分解变换 |
| 13 | 小波包分解变换 |
| 14 | 谱分析 | 功率谱分析 |
| 15 | 能量谱分析 |
| 16 | 相位谱分析 |
| 17 | 包络谱分析 |

### 故障诊断模块

故障诊断模块功能流程图如下所示：



1. 故障诊断流程图

该模块获取传感器数据采集模块采集的正常数据和故障振动数据，对振动数据进行滤波，去除噪声干扰，使用特征提取模块种的时域分析算法、小波包分解算法特征、包络谱分析算法提取所有样本的时域、小波能量和包络谱特征，构建基于BP网络的故障分类模型，使用正常和故障数据提取的特征训练优化该模型，故障诊断模块中保留最佳模型，后期实时采集测点的振动数据送入到故障诊断模块进行故障诊断，详细诊断过程见《发动机异常状态评估技术与故障诊断方案》第2.8小节。

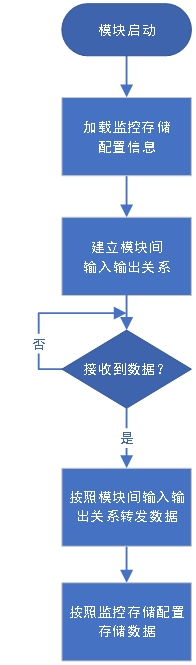
### 数据监控模块

对于嵌入式软件和上位机软件，都包含数据监控模块（后续相同部分不再赘述）。

数据监控模块主要用于异常状态检测软件中的各功能模块的数据转发和数据存储功能。数据监控模块在初始化过程中获取其他功能模块的配置信息，根据配置信息中的输入输出端口信息获取各功能模块数据的传递关系，并将此信息保存在自身模块内。

异常状态检测软件在运行过程中，各功能模块间通过数据监控模块来转发数据，形成模块间的数据传递；同时，数据监控模块将各功能模块输出的数据缓存在模块内，基于先入先出规则将数据有序的存储到本地数据库中。通过数据监控模块保证了异常状态检测软件中各功能模块的独立性。

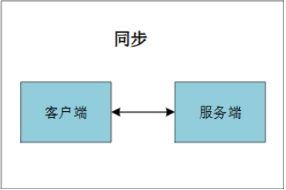
数据监控模块的流程图如下所示：



1. 数据监控模块的功能流程图

1. **数据转发**

对于运行软件的数据转发，主要基于OSA-CBM标准通过调用标准接口来实现数据的转发。对于传感器数据，通过同步通信方式来获取实时数据信息，调用同步接口返回数据内容，然后转发给其他功能模块。



1. 同步转发
2. **数据存储**

在数据监控模块中，主要通过监控报告来实现数据的触发存储。监控报告主要包含三部分内容：

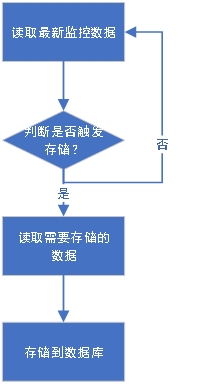
1）该报告对应的参数配置：报告名称、报告类型、监控频率（每隔多长时间监控一次）、存储数量、存储位置（存储的数据为监控触发前、触发中或者触发后）。

2）监控触发条件：监控参数数据id，参数触发判断（大于/等于/小于某数值等），与下一条监控参数信息的判断关系（与、或）。

3）存储内容：存储参数数据id等信息。

数据监控模块开始运行后，根据监控报告的内容，从数据缓存区读取监控参数的最新数据，然后判断是否满足触发存储的条件。如果条件判断满足，则从缓存区获取对应需要存储的数据存储到数据库中。数据监控模块中可以包括多个监控报告配置信息，每个监控报告的处理保持线程独立性（多线程处理）。

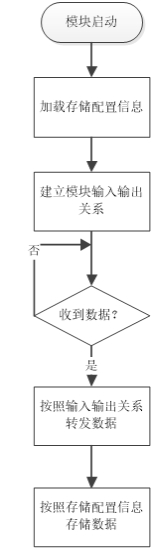
基于监控报告的存储过程如下：



1. 监控报告流程图

运行软件在运行过程中，各功能模块间通过数据交换与数据同步模块来转发数据，形成模块间的数据传递；同时，数据交换与数据同步模块将各功能模块输出的数据缓存在模块内，基于先入先出规则将数据有序的存储到数据库中。

实时数据存储流程图如下所示：



1. 实时数据存储流程图

## 上位机运行软件功能模块设计

### 状态监测模块

状态检测模块主要对收到的数据进行数据判读、状态判读、事件判读、指令判读、异常判读等，给出工作状态信息、是否发生特定事件、控制指令是否正确执行以及是否存在数据异常、告警等提示信息。状态检测对后续的诊断、评估起到数据剔除、提取等作用。同时为后续做实验分析、数据分析打下基础。为了对船载数据有一个初步的判断，系统对船载数据中的指令、参数数据进行分析，判读指令参数执行是否正确。

1. **告警监测**

告警监测通过监控参数数据，导入告警算法（阈值或告警方程），从而实时判断是否产生告警信息，如水温过高、电压超压等告警信息。此类告警信息主要用于后续诊断和评估功能模块的输入数据。

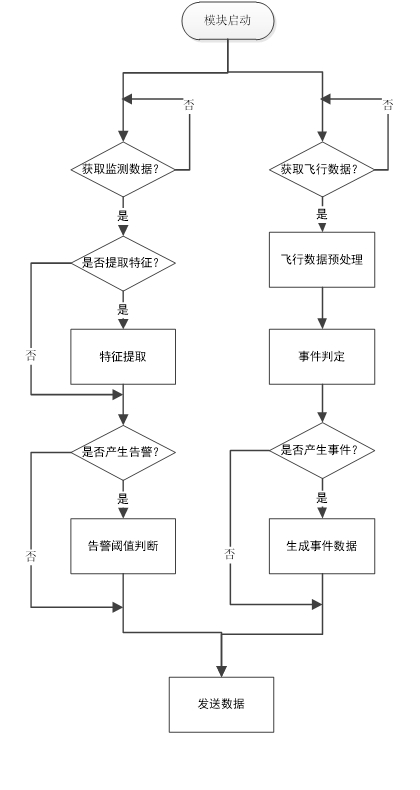
1. **基于事件的监测**

事件监测主要用于实时监测状态的改变或切换。例如：状态检测模块可以实时监控遥测参数，判断系统是否上电，还是可以监测系统的当前运行阶段等。基于事件的监测还可以用于当状态发生改变时，可以触发相关数据的存储功能，用于存储指定工况下的数据，支撑后续趋势分析和趋势预测功能。

上位机运行软件中，当获取到数据采集模块解析后的数据，对于所有监测数据，按照配置信息对指定的数据首先进行特征提取计算获取特征值；然后根据告警配置信息进行数据阈值判断，给对应的数据添加是否告警的状态标签。

对于获取的流程数据，按照是数据间的逻辑关系运算计算出当前的装备运行状态、任务状态、工况的信息，生成事件数据。

状态监测处理功能的流程图如下所示。



1. 状态监测处理流程图

状态监控模块基于多线程的方式实现，模块完成初始化工作后，基于告警、事件的配置信息对输入的船载数据进行算法处理，判断是否产生告警、事件信息；如果产生则存储到模块缓存区。发送线程判断当前缓存区是否存在数据，如果有则按照队列先入先出规则发送数据，发送完成后删除队列中对应的数据。对于接收和发送线程，缓存区是互锁的。

### 异常状态评估模块



1. 异常状态评估流程图

该模块获取传感器数据采集模块采集的正常数据，对正常状态的振动数据进行滤波，去除噪声干扰，使用特征提取模块种的时域分析算法、小波包分解算法特征、包络谱分析算法提取所有样本的时域、小波能量和包络谱特征，构建基于DeepCross网络的状态异常检测模型，DeepCross网络由交叉网络和感知机网络组成，使用正常数据提取的特征训练优化该模型，模型输出系统状态异常检测结果，异常状态评估模块模块中保留最佳模型，后期实时采集测点的振动数据进行滤波、特征提取后送入到异常状态评估模块进行系统状态评估，输出系统是否异常，详细诊断过程见《发动机异常状态评估技术与故障诊断方案》第2.7小节。

### 决策生成模块

状态监测、故障诊断及评估的数据输入到维修辅助决策模块；维修辅助决策模块直接读取数据，通过系统工作状态判别、数据分类、模型处置、关联维修流程等步骤，输出辅助维修决策。同时根据信息管理内容，关联维修备品备件等信息生成辅助维修报告内容。

其中，维修辅助决策通过当前的状态监测数据，获取当前系统的工作状态信息；通过实时故障诊断结果，根据FMECA不同故障模式的故障概率来给出优先检查、维修哪个设备；然后结合当前子系统/设备的工作状态，给出立即维修或者推迟维修的决策。

同时，维修辅助决策结合最近的评估结果，基于决策模型，通过优先级来输出最终的维修决策结果，生成辅助维修决策报告。

上位机运行软件的维修辅助决策模块提供对应的IETM链接接口，支持跳转到对应的IETM章节来指导维修工作。同时，还可以根据内置的维修流程内容来加载显示整个维修流程信息。维修流程模板设计如下所示：

维修流程内容表

|  |  |
| --- | --- |
| **录入条目名称** | **解释说明** |
| 分系统/设备代号 | 字符串类型，能够唯一标识分系统/设备 |
| 维修id | 字符串类型，能够唯一标识当前维修流程 |
| 维修程序名称 | 字符串类型 |
| 维修类型 | 枚举类型 |
| 设备批次 | 字符串类型，同一设备不同批次产品可能对应不同的维修程序 |
| 安全等级 | 枚举类型（  0：轻微的；  1：中度的；  2：严重的） |
| 安全等级提示 | 字符串类型，提示文本信息 |
| 前置条件 | 字符串类型，为了执行本维修程序的先决条件 |
| 维修步骤编号 | 字符串类型，如果本维修程序有多个步骤，分为多行录入 |
| 维修步骤描述 | 字符串类型 |
| 维修步骤关联图文件名 | 字符串类型，在当前维修步骤时通过图片方式指导维修工作，图文件必须放在当前文本同目录下 |
| 维修步骤关联视频文件名 | 字符串类型，在当前维修步骤时通过视频方式指导维修工作，图文件必须放在当前文本同目录下 |
| 下一步维修步骤编号 | 定义本步骤后需要执行的步骤，最后一步设置为空 |
| 维修区域/位置 | 字符串类型，没有则忽略，不填写 |
| 维修设计的维修口盖/面板编码 | 字符串类型，没有则忽略，不填写 |
| 维修人员级别 | 枚举类型（  0：高级；  1：中级；  2：初级） |
| 维修人员数量 | 整数类型 |
| 维修工作估计（小时） | 浮点数类型 |
| 所需工具名称 | 字符串类型 |
| 所需工具型号 | 字符串类型，标识工具规格 |
| 所需工具数量 | 整数类型 |
| 所需耗材编号 | 字符串类型，没有则忽略，不填写 |
| 所需耗材数量 | 整数类型，没有则忽略，不填写 |

辅助维修决策同时提供定期维护提醒功能，发动机系统的维护项目按照维护周期可分为三种类型：

1）周期维护：按照时间周期维护的项目，如月维护、季度维护等；

2）阶段维护：按照某些指标周期维护的项目，如航行里程、运行时间等；

3）其他维护：基于条件触发的维护项目等；

辅助维修决策中可通过记录上次维护的维护数据和当前数据来分析判断维修项目是否需要维护，如果需要则给出维护提醒标识；

完成维修维护工作后，通过填写维保履历来确认工作完成，所填信息存储到数据库中。维修保养履历的内容应尽量避免使用人员人工输入相关内容，根据当前系统信息自动填写相关内容，其他内容尽量配置为选择项让使用人员选择。

维保履历主要包括三部分内容：

1）设备信息

包括当前设备信息，需要维修的设备名称、设备型号、出厂编号等信息；

2）故障告警信息

包括问题发生时间、发生阶段，问题获知方式，当前设备运行状态以及问题描述等信息；

3）维修信息

维修人员、维修开始时间、维修项目、更换设备编号、问题解决时间、备注等信息。

维修保养履历内容存储在数据库中，通过维修履历的积累来不断完整案例库。从而进一步提高维修决策模型的准确度。

### 人机界面

上位机运行软件的人机界面主要包括两部分内容：

1. 状态监控界面：主要通过获取实时状态数据来展示系统当前健康状态。包括关键参数展示、故障维修提醒、定期维护提醒、健康状态展示等；
2. 数据管理界面：主要通过查询数据库，获取各类型数据的统计分析结果。包括原始数据管理、故障数据管理、维修维护数据管理等。

#### 状态监控界面



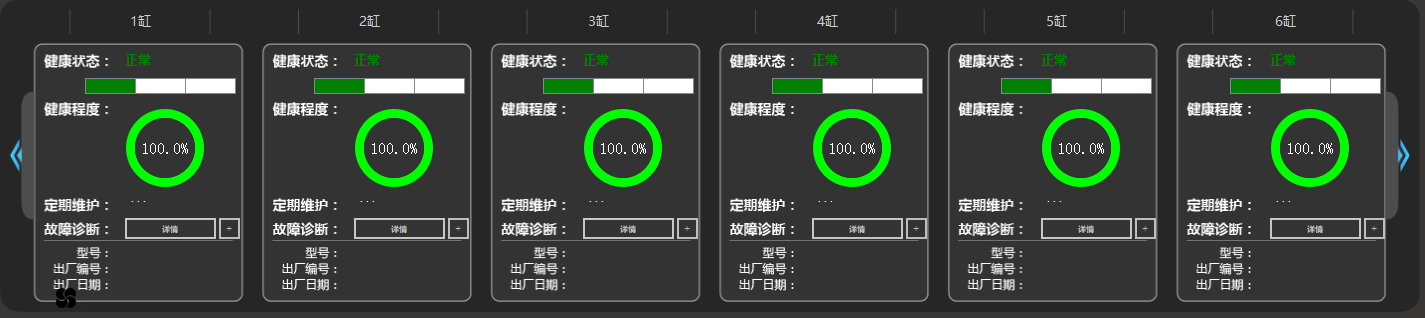
1. 状态监控界面示意图

如上图所示，状态监控的界面结构主要由三部分组成：

* **系统模型图**

系统模型图以框图的形式直观分层显示多任务系统各层级的健康状态，以及对应的监控参数实时数据，用户可通过点击界面上的图标进入到下一级系统/设备的组成模型图。

* **健康单元**

健康单元主要显示当前系统下组成单元的健康状态、定期维护、维护建议等信息，以及对应的型号、出厂编号、出厂日期等。健康状态框图界面如下所示： 

1. 健康评估界面示意图

其中，健康状态和健康程度用来显示通过健康评估功能模块输出的当前系统/设备的健康等级和健康度；目前PHM软件中的健康等级分为三级：

正常：该系统/设备工作正常，没有故障；

良好：该系统/设备运行故障，但可以正常工作；

告警：该系统/设备存在故障，无法正常工作。

同时在系统模型图上用红色、黄色和红色来标识不同的健康等级，健康程度是通过量化方式以百分比的形式来显示当前的健康状态。

* **定期维护**

定期维护图标用来提醒当前系统/设备是否需要进行维护保养工作，以图标（）形式提供操作人员。维护保养界面如下，分页分类型显示所有的维护项信息，操作人员可通过IETM链接查看对应的工作内容指导维护工作，工作完成后勾选“任务完成”保存当前数据到数据库。

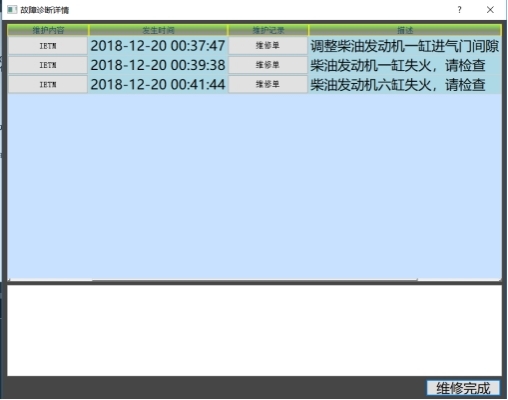


1. 定期维护界面示意图

* **维护建议**

维护建议用于提醒操作人员当前系统/设备是否存在故障，通过故障诊断和维护建议生成模块输出结果数据。

维护建议的详细信息显示当前故障的维护建议描述，故障发生时间，IETM链接和维修工作记录单链接。操作人员可查询IETM手册指导维修工作。



1. 维修详情界面示意图

#### 数据管理界面

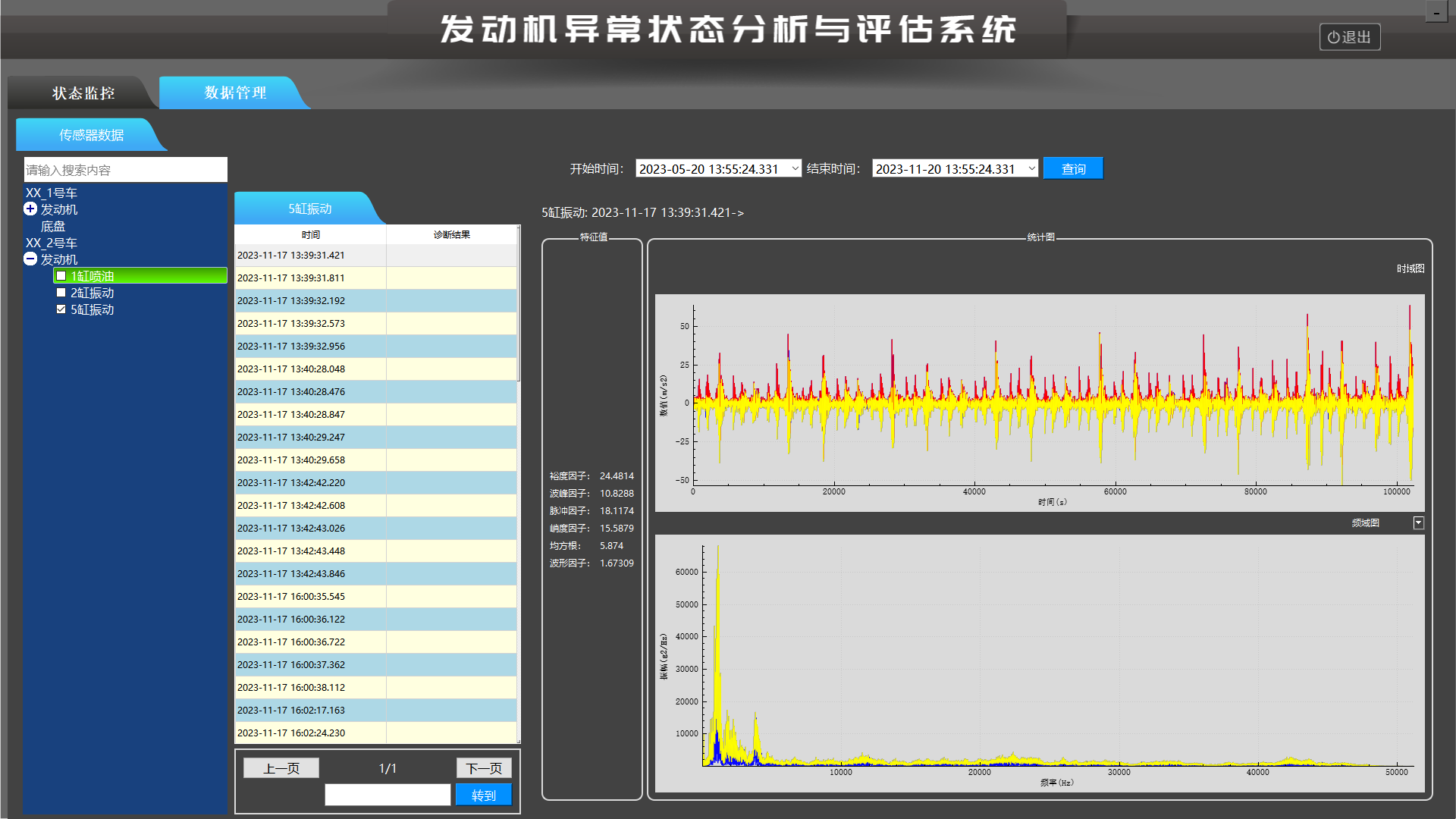
数据管理界面主要基于历史数据进行统计分析，采用列表的方式来显示历史数据，根据收集到的某个参数的数据集通过有效算法画出统计图（时频域特征图）以及趋势图等内容，主要用于相关技术人员对系统监控状态做进一步分析，达到对系统当前状态有效预测和评估的作用。

* 原始数据管理

原始数据管理主要实现以不同形式为索引，显示产品树上选中传感器的历史数据信息，包括频域图和时域图。

* 1. 首先在产品树结构上选定想要查看的传感器参数；
  2. 然后界面上会显示选中传感器参数的历史数据列表，选中其中一组数据查看对应的时域图和频域图。每组数据显示对应的特征值信息。

传感器波形数据的显示如下：



1. 传感器数据时、频域图

# 数据库设计

异常状态检测软件的数据库采用MySQL数据库设计实现，主要通过ODBC方式连接。数据库的表设计依据OSA-EAI-CRIS数据体系设计实现。

## 数据库接口

异常状态检测软件包括嵌入式运行软件和上位机运行软件，软件都通过调用qt自带的数据库驱动来实现数据库的连接和数据的增、删、改、查。连接数据库需要设置数据库类型、数据库名称、地址、端口号、用户名、密码等信息，如下所示。

m\_phmDB=QSqlDatabase::addDatabase(m\_connMode, m\_connDBName);

m\_phmDB.setHostName(m\_strHost);

m\_phmDB.setDatabaseName(m\_strName);

m\_phmDB.setPort(m\_nPort);

m\_phmDB.setUserName(m\_strUser);

m\_phmDB.setPassword(m\_strPassword);

连接数据库后，运行软件主要根据存储数据的类型将原始数据、告警数据、事件数据、诊断评估结果、维修决策数据分别存储到不同的数据表中。

人机交互界面主要通过访问不同数据表来查询获取对应的最新数据信息和历史数据，从而实现健康状态展示和数据管理功能。

所有数据的存储、查询、更新都通过标准sql语句来实现。对于不同数据类型的表设计后续详细介绍。

## 数据库表设计

异常状态检测软件的数据库采用MySQL数据库设计实现，针对不同组织层级的使用，软件数据库采用统一的表设计实现,方便不同层级间数据库的数据同步。根据异常状态检测软件的使用人员和存储的数据信息。数据库的表设计主要分为如下及部分信息：

1. 组织层级表
2. 系统设计表
3. 健康管理数据表
4. 人员备件表
5. 位置信息表

### 组织层级表

异常状态检测软件的组织层级表主要存储当前的系统组织层级管理，主要包括组织（enterprise）->装备（site）->分系统（segment）->设备（asset）以及监控参数(measurement location)等数据表，同时，还包括对应的类型（type）信息表。

在异常状态检测软件中，数据库中组织层级表是通过开发平台软件在构建过程中生成的数据信息。

#### 组织级

其中，组织信息数据表结构如下所示：

1. 组织信息数据表**（enterprise）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| enterprise\_id | 主键id，唯一标识。 | UINT NOT NULL |
| ent\_db\_site | 外键。用于关联type类型 | UINT NOT NULL |
| ent\_db\_id | UINT NOT NULL |
| ent\_type\_code | UINT NOT NULL |
| user\_tag\_ident | 描述 | STRING(1:254) NOT NULL |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

在组织信息表中，id来唯一标识当前组织名称；单个组织的相关信息包括系统类型、名称、描述信息、创建时间等。同时，通过如下组织的父子关系表来展示不同层级组织的树型结构。

例如对于当前异常状态检测软件中的数据库，组织级为装工院，包括名称和id等。以此类推，可存储所有组织单位的信息以及他们之间的层级关系。

1. 组织父子关系表**（enterprise\_child）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| enterprise\_id | 主键，唯一标识。 | HEXSTRING(16) NOT NULL |
| child\_ep\_id | 主键，对应的子组织 | HEXSTRING(16) NOT NULL |
| ordering\_seq |  | UINT |
| gmt\_last\_updated | 上次更新时间。 | DATETIME(19:29) |

#### 装备级

异常状态检测软件中的单个装备主要信息如下：

1. 装备信息表**（site）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| site\_code | 主键。唯一标识，长度与IP地址一致。 | HEXSTRING(16) NOT NULL |
| enterprise\_id | 外键。所属enterprise可以改变，此处id随着改变。但site\_code不变。 | UINT NOT NULL |
| site\_id | id标识。 | UINT NOT NULL |
| st\_db\_site | 外键 Site\_Type.st\_db\_id。唯一标识type。 | UINT NOT NULL |
| st\_db\_id | UINT NOT NULL |
| st\_type\_code | UINT NOT NULL |
| user\_tag\_ident | 描述 | STRING(1:254) NOT NULL |
| name | 名称 | STRING(0:254) |
| duns\_number | D-U-N-S编号。 | UINT |
| template\_yn | “Y”标识当前site为模板，抽象概念。 | STRING(1) |
| segment\_site | 外键。用于唯一关联一个segment。 | HEXSTRING(16) |
| segment\_id | UINT |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

其中，装备通过id来唯一标识，每个装备都定义了名称、描述信息、编号等信息。装备都从属于一个组织，在同一个组织下装备id唯一。不同组织下的装备id可以重复，但通过组织id+装备id（site\_code）来保证全局的唯一性。

#### 分系统级

每个装备都包含多个分系统和设备成员，而分系统本身也具有分层包含的关系。在异常状态检测软件的数据库设计中，主要通过分系统数据表和分系统父子关系数据表来描述。系统组成表示意如下：

1. 分系统信息表**（segment）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| segment\_site | 主键。表示所属site。 | HEXSTRING(16) NOT NULL |
| segment\_id | id。Site下唯一标识一个segment。 | UINT NOT NULL |
| sg\_db\_site | 外键。唯一标识一个type类型。 | HEXSTRING(16) NOT NULL |
| sg\_db\_id | UINT NOT NULL |
| sg\_type\_code | UINT NOT NULL |
| segment\_group\_yn |  | STRING(1) NOT NULL |
| criticality | 危害等级。 | INT |
| cs\_type\_db\_site | 外键唯一标识危害等级类型。 | HEXSTRING(16) |
| cs\_type\_db\_id | UINT |
| cs\_type\_code | UINT |
| user\_tag\_ident | 描述 | STRING(0:254) NOT NULL |
| name | 名称 | STRING(0:254) |
| long\_description | 长描述，定义长字节。 | STRING(0:4000) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

其中，在数据库中通过id来区别不同的分系统，同时每个分系统都从属于某个装备；每个装备内的分系统列表通过父子关系表来描述树形结构。

1. 分系统父子关系信息表**（segment\_child）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| segment\_site | 主键。对应外键Segment.segment\_site。父节点。 | HEXSTRING(16) NOT NULL |
| segment\_id | 主键。对应外键Segment.segment\_id。 | UINT NOT NULL |
| child\_sg\_site | 主键。对应外键Segment.segment\_site。子节点 | HEXSTRING(16) NOT NULL |
| child\_sg\_id | 主键。对应外键Segment.segment\_id。 | UINT NOT NULL |
| ordering\_seq |  | UINT NOT NULL |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

#### 监控参数

在异常状态检测软件中，每个分系统包含的监控数据信息在监控参数数据表中定义。监控参数数据表主要存储当前总线报文中发送的数据和传感器采集数据。监控参数数据表定义如下所示：

1. 监控数据信息表**（measurement location）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| meas\_loc\_site | 主键。标识所属site。 | HEXSTRING(16) NOT NULL |
| meas\_loc\_id | 主键。Id，site下唯一标识。 | UINT NOT NULL |
| ml\_db\_site | 外键，标识类型。 | HEXSTRING(16) NOT NULL |
| ml\_db\_id | UINT NOT NULL |
| ml\_type\_code | UINT NOT NULL |
| segment\_or\_asset | 属于segment或者asset | STRING(1) NOT NULL |
| segment\_site | 外键，关联segment。 | HEXSTRING(16) |
| segment\_id | UINT |
| asset\_org\_site | 外键，关联asset。 | HEXSTRING(16) |
| asset\_id | UINT |
| ds\_db\_site | 外键，关联datasource类型。 | HEXSTRING(16) |
| ds\_db\_id | UINT |
| ds\_type\_code | UINT |
| tr\_db\_site | 外键，关联传感器类型。 | HEXSTRING(16) |
| tr\_db\_id | UINT |
| tr\_type\_code | UINT |
| ta\_orient\_deg |  | UINT |
| ta\_db\_site | 外键，类型：X、y、r、a轴。 | HEXSTRING(16) |
| ta\_db\_id | UINT |
| ta\_type\_code | UINT |
| mim\_loc\_seq |  | UINT |
| motion\_direction | ‘N’ and ‘R’。 | STRING(1) |
| mim\_user\_prefix |  | STRING(4) |
| mc\_db\_site | 外键，校准方式。。 | HEXSTRING(16) |
| mc\_db\_id | UINT |
| mc\_type\_code | UINT |
| mc\_calc\_size |  | UINT |
| update\_interval | 采样频率。 | DOUBLE |
| int\_eu\_db\_site | 外键，采样频率单位。 | HEXSTRING(16) |
| int\_eu\_db\_id | UINT |
| int\_eu\_type\_code | UINT |
| collect\_duration | 采样周期 | DOUBLE |
| dur\_eu\_db\_site | 外键，采样周期单位。 | HEXSTRING(16) |
| dur\_eu\_db\_id | UINT |
| dur\_eu\_type\_code | UINT |
| ml\_eu\_db\_site | 外键，传感器单位。 | HEXSTRING(16) |
| ml\_eu\_db\_id | UINT |
| ml\_eu\_type\_code | UINT |
| xml\_data\_type | XmlDatasheet。 | STRING(254) |
| xml\_pattern\_regex | Xml datasheet格式。 | STRING(254) |
| user\_tag\_ident | 描述 | STRING(0:254) NOT NULL |
| barcode | 编号 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

### 系统设计表

系统设计表主要用于存储在系统设计阶段的数据信息，如FMECA、RCM等。

#### 告警设计表

告警设计表主要定义了系统包括的所有数据告警配置信息，包括关联的测试点、告警名称、上下限阈值、单位、告警类型等信息。

告警设计表主要通过PHM开发平台软件在构建过程中生成。具体信息如下所示。

1. 告警配置信息（**num\_alarm\_reg**）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| alarm\_db\_site | 主键。关联一个测试点meas\_loc | HEXSTRING(16) NOT NULL |
| alarm\_db\_id | UINT NOT NULL |
| meas\_loc\_site | HEXSTRING(16) NOT NULL |
| meas\_loc\_ id | UINT NOT NULL |
| ordering\_seq | 区分同一meas\_loc不同告警。 | UINT NOT NULL |
| ml\_db\_site | 外键。测试点类型 | HEXSTRING(16) |
| ml\_db\_id | UINT |
| ml\_type\_code | UINT |
| al\_db\_site | 外键Alarm\_Type.al\_type\_code。告警类型。 | HEXSTRING(16) NOT NULL |
| al\_db\_id | UINT NOT NULL |
| al\_type\_code | UINT NOT NULL |
| eu\_db\_site | 外键Eng\_Unit\_Type.eu\_type\_code。数据单位。 | HEXSTRING(16) NOT NULL |
| eu\_db\_id | UINT NOT NULL |
| eu\_type\_code | UINT NOT NULL |
| gmt\_alarm\_start | 告警开始时间 | DATETIME(19:29) NOT NULL |
| st\_loc\_hr\_delta | 本地时间小时偏移。 | SHORT |
| st\_loc\_min\_delta | 本地时间分钟偏移。 | USHORT |
| min\_amplitude | 告警下限值 | DOUBLE [removed NOT NULL constraint] |
| max\_amplitude | 告警上限值。 | DOUBLE [removed NOT NULL constraint] |
| user\_tag\_ident | 描述信息 | STRING(0:254) |
| name | 告警名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

#### 故障设计表

在武器系统数据库中，故障设计表主要包括故障模糊组、故障模式以及关联信息等。故障设计表的内容同样是通过PHM开发平台软件在构建过程中生成的。

其中，故障模糊组信息如下所示。包括模糊组id、名称等信息，记录了当前系统包含的所有故障模糊组列表。

1. 故障模糊组信息表**（sg\_hyp\_ev\_amb\_set）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| sg\_hyp\_amb\_set\_db\_site | 模糊组id | HEXSTRING(16) NOT NULL |
| sg\_hyp\_amb\_set\_db\_id | UINT NOT NULL |
| sg\_hyp\_amb\_set\_id | UINT NOT NULL |
| segment\_site | 外键，关联segment。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| amb\_set\_db\_site | 外键，关联模糊组类型。 | HEXSTRING(16) NOT NULL |
| amb\_set\_db\_id | UINT NOT NULL |
| amb\_set\_type\_code | UINT NOT NULL |
| by\_agent\_site | 外键，关联算法名称。 | HEXSTRING(16) NOT NULL |
| by\_agent\_id | UINT NOT NULL |
| user\_tag\_ident | 描述 | STRING(0:254) NOT NULL |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

故障模式表主要记录了当前系统所有的故障模式列表，包括故障模式id、所属分系统/设备、故障模式名称、严酷度等级等信息。具体表信息如下所示。

同时，该表也存储系统包含的所有事件信息（故障模式就是一种特殊类型的事件），通过type来区分不同类型的事件。

1. 故障模式信息**（sg\_hyp\_event）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| sg\_hyp\_db\_site | 主键。Id。 | HEXSTRING(16) NOT NULL |
| sg\_hyp\_db\_id | UINT NOT NULL |
| sg\_hyp\_event\_id | UINT NOT NULL |
| segment\_site | 外键，关联segment。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| ev\_db\_site | 外键，类型信息。 | HEXSTRING(16) NOT NULL |
| ev\_db\_id | UINT NOT NULL |
| event\_type\_code | UINT NOT NULL |
| severity\_lev\_db\_site | 外键，表示严酷度。 | HEXSTRING(16) NOT NULL |
| severity\_lev\_db\_id | UINT NOT NULL |
| severity\_lev\_type\_code | UINT NOT NULL |
| gmt\_created | 产生时间。 | DATETIME(19:29) |
| cr\_loc\_hr\_delta | 与本地时间小时偏移量。 | SHORT |
| cr\_loc\_min\_delta | 与本地时间分钟偏移量。 | USHORT |
| by\_org\_asite | 外键，关联人或算法。 | HEXSTRING(16) |
| by\_agent\_id | UINT |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| assoc\_file\_name | 关联数据文件 | STRING(0:254) |
| associated\_blob | 关联数据块 | BASE64BINARY |
| gmt\_last\_updated | 上次更新时间 |  |

### 数据输出表

异常状态检测软件中数据处理模块的输出数据存储到本地数据库中。异常状态检测软件数据库的设计基于各层的输出数据信息，设计创建各层各类型的数据表用于存储数据。

所有数据输出表通过PHM运行软件在数据处理过程中将各自输出的结果存储到对应的数据表中。

#### BIT、传感器数据

异常状态检测软件中数据采集模块输出的数据存在原始数据表中，分别基于字符串（mevent\_chr\_data）、实数（mevent\_num\_data）和块数据（mevent\_blob\_data）分为三个表存储，每个原始数据表用于存储数据采集模块输出的数据值、数据单位、产生时间等信息。三个类型的表设计基本一致，只是存储数据的类型不同。

实数型原始数据表的存储信息如下所示：

1. 原始数据表的存储信息

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| meas\_loc\_site | 主键。id。 | HEXSTRING(16) NOT NULL |
| meas\_loc\_id | UINT NOT NULL |
| gmt\_event | 主键。对应一次Meas\_Event。 | DATETIME(19:29) NOT NULL |
| eu\_db\_site | 主键。关联实数数据的单位信息。 | HEXSTRING(16) NOT NULL |
| eu\_db\_id | UINT NOT NULL |
| eu\_type\_code | UINT NOT NULL |
| data\_value | 存储实数型数据数值 | DOUBLE |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

对于异常状态检测软件，可通过跨表查询mevent\_num\_data表里面的id、数值、时间戳以及measurement location表里对应id的参数名称来显示实时和历史数据信息。

#### 告警数据

告警事件表用于存储发生的告警事件，具体信息如下所示。在告警事件表中主要存储了对应meas\_num\_data发生的记录，表明在某一条meas\_num\_data数据产生时触发了告警，具体告警时对应的数值可以通过mevent\_num\_data关联查询。

同时，还可以通过meas\_loc\_id关联查询num\_alarm\_reg表对应的告警配置信息。用于异常状态检测软件展示告警详细信息。

1. 告警事件表**（mevent\_num\_alarm）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| meas\_loc\_site | 主键。对应外键 Mevent\_Num\_Data.meas\_loc\_site。关联一条Mevent\_Num\_Data记录。 | HEXSTRING(16) NOT NULL |
| meas\_loc\_id | 主键。对应外键 Mevent\_Num\_Data.meas\_loc\_id。 | UINT NOT NULL |
| gmt\_event | 主键。对应外键 Mevent\_Num\_Data.gmt\_event。 | DATETIME(19:29) NOT NULL |
| eu\_db\_site | 主键。对应外键 Mevent\_Num\_Data.eu\_db\_site。 | HEXSTRING(16) NOT NULL |
| eu\_db\_id | 主键。对应外键 Mevent\_Num\_Data.eu\_db\_id。 | UINT NOT NULL |
| eu\_type\_code | 主键。对应外键 Mevent\_Num\_Data.eu\_type\_code。 | UINT NOT NULL |
| al\_db\_site | 主键。对应外键 Alarm\_Type.al\_db\_site。 | HEXSTRING(16) NOT NULL |
| al\_db\_id | 主键。对应外键 Alarm\_Type.al\_db\_id。 | UINT NOT NULL |
| al\_type\_code | 主键。对应外键 Alarm\_Type.al\_type\_code。关联告警类型或告警等级。 | UINT NOT NULL |
| org\_agent\_site | Agent.org\_agent\_site。 | HEXSTRING(16) |
| agent\_id | Agent.agent\_id。关联获取告警的agent，可以是人员，也可以是算法。 | UINT |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 告警名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

#### 状态事件数据

状态事件表主要用于存储通过状态检测模块输出的事件信息，包括事件名称、事件开始时间、时间结束时间等信息；同时事件数据关联对应的设备id。状态事件表如下所示：

1. 状态事件表

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| segment\_site | 主键。关联一个segment。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| ev\_db\_site | 主键。事件类型。 | HEXSTRING(16) NOT NULL |
| ev\_db\_id | UINT NOT NULL |
| event\_type\_code | UINT NOT NULL |
| gmt\_event\_start | 事件开始时间。 | DATETIME(19:29) NOT NULL |
| gmt\_event\_stop | 事件结束时间。 | DATETIME(19:29) |
| ev\_loc\_hr\_delta | 与本地时间小时偏移量。 | SHORT |
| ev\_loc\_min\_delta | 与本地时间分钟偏移量。 | USHORT |
| gmt\_stored | 时间存储时间 | DATETIME(19:29) |
| st\_loc\_hr\_delta | 存储时间与本地时间小时偏移量。 | SHORT |
| st\_loc\_min\_delta | 存储时间与本地时间分钟偏移量 | USHORT |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

#### 诊断评估数据

故障诊断表主要存储诊断模块输出的模糊组id，用于表示可能出现故障的设备列表；设备健康表用于存储健康评估模块输出的分系统/设备的健康等级和健康度数据；其中，健康度数据为0-1之间的double类型数据，1标识完全正常，0表示健康度最差。数据表的主要字段定义如下：

1. 诊断数据表**（sg\_pr\_ev\_amb\_set\_sg\_ev）**

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| sg\_prop\_amb\_set\_db\_site | 主键。关联一个模糊组id。 | HEXSTRING(16) NOT NULL |
| sg\_prop\_amb\_set\_db\_id | UINT NOT NULL |
| sg\_prop\_amb\_set\_id | UINT NOT NULL |
| segment\_site | 主键。关联设备信息 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| ev\_db\_site | 主键。类型 | HEXSTRING(16) NOT NULL |
| ev\_db\_id | UINT NOT NULL |
| event\_type\_code | UINT NOT NULL |
| gmt\_event\_start | 主键。关联事件开始时间 | DATETIME(19:29) NOT NULL |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

1. 健康评估数据表

|  |  |  |
| --- | --- | --- |
| 字段 | 解释 | 大小/数据类型 |
| segment\_site | 主键。对应Segment。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| gmt\_assessment | 主键。产生数据时间。 | DATETIME(19:29) NOT NULL |
| by\_agent\_site | 主键。可以关联对应的评估算法。 | HEXSTRING(16) NOT NULL |
| by\_agent\_id | UINT NOT NULL |
| loc\_hr\_delta | 与本地时间小时偏移量。 | USHORT |
| loc\_min\_delta | 与本地时间分钟偏移量。 | USHORT |
| health\_lev\_db\_site | 外键，健康等级 | HEXSTRING(16) NOT NULL |
| health\_lev\_db\_id | UINT NOT NULL |
| health\_lev\_type\_code | UINT NOT NULL |
| health\_scale\_precise | 健康度0-1 | DOUBLE |
| likelihood\_prob |  | DOUBLE |
| gmt\_created | 建立或者应用时间 | DATETIME(19:29) NOT NULL |
| cr\_loc\_hr\_delta | 与本地时间小时偏移量。 | USHORT |
| cr\_loc\_min\_delta | 与本地时间分钟偏移量。 | USHORT |
| gmt\_audited | 评判时间 | DATETIME(19:29) |
| aud\_loc\_hr\_delta | 与本地时间小时偏移量。 | USHORT |
| aud\_loc\_min\_delta | 与本地时间分钟偏移量。 | USHORT |
| aud\_quality\_code | 数据可信度。 | UINT |
| aud\_by\_org\_asite | 外键，可以是算法。 | HEXSTRING(16) |
| aud\_by\_agent\_id | UINT |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

#### 维修决策数据

维修决策表用于存储分系统/设备的维修决策描述信息。在维修决策表中，主要存储了当前设备的维修决策生成时间和维修决策内容。其主要的字段定义如下：

1. 维修决策表

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| segment\_site | 主键，维修建议的对象。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| gmt\_recommendation | 主键，维修建议产生时间。 | DATETIME(19:29) NOT NULL |
| by\_agent\_site | 外键，关联算法。 | HEXSTRING(16) NOT NULL |
| by\_agent\_id | UINT NOT NULL |
| loc\_hr\_delta | 与本地时间小时偏移量。 | SHORT |
| loc\_min\_delta | 与本地时间分钟偏移量。 | USHORT |
| priority\_lev\_db\_site | 外键，优先级。 | HEXSTRING(16) |
| priority\_lev\_db\_id | UINT |
| priority\_lev\_type\_code | UINT |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

### 人员器件表

#### 人员信息表

异常状态检测软件的人员列表设计如下。主要包括人员id、人员类型、名称等信息。

1. 人员信息表（agent）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| org\_agent\_site | 主键。对应外键Site.site\_code。 | HEXSTRING(16) NOT NULL |
| agent\_id | 主键。Id，site下唯一标识一个agent。 | UINT NOT NULL |
| agent\_db\_site | 外键 Agent\_Type.agent\_type\_code。标识一个agent类型。 | HEXSTRING(16) NOT NULL |
| agent\_db\_id | UINT NOT NULL |
| agent\_type\_code | UINT NOT NULL |
| asset\_org\_site | 外键 Asset.asset\_org\_site。关联一个资产。 | HEXSTRING(16) |
| asset\_id | UINT |
| user\_tag\_ident | 描述 | STRING(0:254) NOT NULL |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

同时，人员角色表来标识角色类型及名称等，如根据角色类型来划分维修人员的等级：高级、中级、初级等；也可以通过角色类型定义不同类型的角色，如管理员、操作员等。

1. 角色信息表（agent\_role\_type）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| org\_agent\_site | 主键。对应外键外键Agent.org\_agent\_site。唯一关联一个agent。 | HEXSTRING(16) NOT NULL |
| agent\_id | 主键。对应外键 Agent.agent\_id。唯一关联一个agent。 | UINT NOT NULL |
| ag\_role\_db\_site | 主键。对应外键 Agent\_Role\_Type.ag\_role\_db\_site。唯一关联一个角色类型。 | HEXSTRING(16) NOT NULL |
| ag\_role\_db\_id | 主键。对应外键 Agent\_Role\_Type.ag\_role\_db\_id。唯一关联一个角色类型。 | UINT NOT NULL |
| ag\_role\_type\_code | 主键。对应外键Agent\_Role\_Type.ag\_role\_type\_code。唯一关联一个角色类型。 | UINT NOT NULL |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

在通过角色和人员的关联关系表，就可以把当前系统所有的维修维护人员以及他们对应的等级、类型关联起来。关联关系表如下所示：

1. 人员角色关联表（agent\_role）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| org\_agent\_site | 主键。对应外键Agent\_Role.org\_agent\_site。标识关联的agent。 | HEXSTRING(16) NOT NULL |
| agent\_id | UINT NOT NULL |
| ag\_role\_db\_site | 主键。对应外键Agent\_Role.ag\_role\_db\_site。标识关联的agent角色。 | HEXSTRING(16) NOT NULL |
| ag\_role\_db\_id | UINT NOT NULL |
| ag\_role\_type\_code | UINT NOT NULL |
| other\_org\_agent\_site | 主键。对应外键Agent.org\_agent\_site。标识关联的其他agent。 | HEXSTRING(16) NOT NULL |
| other\_agent\_id | UINT NOT NULL |
| user\_tag\_ident | 描述 | STRING(0:254) |
| name | 名称 | STRING(0:254) |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

#### 器件信息表

对于器件和备件管理，通过资产信息表来管理所有设备的备品备件信息，如下表所示。在资产信息表中，每一个资产都可以关联一个对应的分系统/设备id（工具类则不需要）。因此可以通过资产信息表来关系装备中所有分系统/设备的备件信息。资产表内容包括资产id、名称、描述信息、危害等级、厂家、类型、序列号等信息。

1. 备件信息表（asset）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| asset\_org\_site | 主键。对应外键Site.site\_code。 | HEXSTRING(16) NOT NULL |
| asset\_id | id，表示site下唯一asset。 | UINT NOT NULL |
| as\_db\_site | 外键Asset\_Type.as\_db\_site。唯一标识type类型 | HEXSTRING(16) NOT NULL |
| as\_db\_id | 外键Asset\_Type.as\_db\_id。唯一标识type类型 | UINT NOT NULL |
| as\_type\_code | 外键Asset\_Type.as\_type\_code。唯一标识type类型 | UINT NOT NULL |
| user\_tag\_ident | 描述 | STRING(0:254) NOT NULL |
| name | 名称 | STRING(0:254) |
| long\_description | 完整描述 | STRING(0:4096) |
| criticality | 外键Criticality\_Scale\_Type。危害等级。 | INT |
| cs\_type\_db\_site | 外键Criticality\_Scale\_Type.cs\_type\_db\_site。唯一标识危害等级。 | HEXSTRING(16) |
| cs\_type\_db\_id | 外键Criticality\_Scale\_Type.cs\_type\_db\_id。唯一标识危害等级。 | UINT |
| cs\_type\_code | 外键Criticality\_Scale\_Type.cs\_type\_code。唯一标识危害等级。 | UINT |
| mf\_db\_site | 外键Manufacture.mf\_db\_site。唯一标识一个Manufacture，与model二选一。可以关联一个型号或者一个工厂。 | HEXSTRING(16) |
| mf\_db\_id | 外键Manufacture.mf\_db\_id (Optional Manufacturer FK)。唯一标识一个Manufacture。 | UINT |
| manuf\_code | 外键Manufacture.mf\_manuf\_code (Optional Manufacturer FK) 。唯一标识一个Manufacture。 | UINT |
| model\_db\_site | 外键Model.model\_db\_site。。唯一标识一个Model。与Manufacture二选一。 | HEXSTRING(16) |
| model\_db\_id | 外键Model.model\_db\_id (Optional Model FK) | UINT |
| model\_id | 外键Model.model\_id (Optional Model FK) | UINT |
| serial\_number | 资产编号 | STRING(0:254) |
| asr\_db\_site | 外键As\_Readiness\_Type.asr\_db\_site。表示维护方式的类型。 | HEXSTRING(16) |
| asr\_db\_id | 外键As\_Readiness\_Type.asr\_db\_id。 | UINT |
| asr\_type\_code | 外键As\_Readiness\_Type.asr\_type\_code。 | UINT |
| segment\_site | 外键Segment.segment\_site 。关联一个segment。 | HEXSTRING(16) |
| segment\_id | 外键Segment.segment\_id 。 | UINT |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

对于备件的使用，通过与对应分系统/设备的关联关系来定义。如下表所示。在备件安装信息表中，定义了备件在关联分系统/设备的安装时间和卸载时间。可以通过录入维修维护履历信息时存储。通过该表就可以列出某个分系统/设备整个生命周期的所有备件的使用过程，同时还可以通过某个备件的使用周期来关联对应的采集数据信息。

1. 备件安装信息表（asset\_on\_segment）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| asset\_org\_site | 主键，资产id。 | HEXSTRING(16) NOT NULL |
| asset\_id | UINT NOT NULL |
| segment\_site | 外键，关联分系统/设备id。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| gmt\_installed | 安装时间 | DATETIME(19:29) NOT NULL |
| gmt\_removed | 卸载时间 | DATETIME(19:29) |
| installed\_by\_agent\_site | 关联安装操作人员 | HEXSTRING(16) |
| installed\_by\_agent\_id | UINT |
| removed\_by\_agent\_site | 关联卸载操作人员 | HEXSTRING(16) |
| removed\_by\_agent\_id | UINT |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

### 位置信息

在异常状态检测软件数据库中，还需要设计一些表来辅助实现一些特定的功能，如下设计位置信息的表。在异常状态检测软件运行时，能够根据从BIT中获取到的位置信息存储到数据库中，从而可以标识装备的移动曲线和当前位置信息。

其中，在gps\_location 表中，geo\_position\_id用于唯一标识一个位置信息，每一个id对应经度、纬度、高度等基本信息。

在segment\_geo\_position中，记录了当前设备（segment）对应的历史位置信息，包括geo\_position\_id、时间等。

1. 位置信息1（gps\_location）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| geo\_db\_site | 主键。关联一条geo\_position数据。 | HEXSTRING(16) NOT NULL |
| geo\_db\_id | UINT NOT NULL |
| geo\_position\_id | UINT NOT NULL |
| ordering\_sequence | 主键。用于存储一个geo\_position可能对应多个位置信息的情况。 | UINT NOT NULL |
| latitude | 纬度 | DOUBLE NOT NULL |
| longitude | 经度 | DOUBLE NOT NULL |
| elevation | 高度 | DOUBLE |
| datum\_db\_site | 外键，坐标系统和时间系统类型。 | HEXSTRING(16) |
| datum\_db\_id | UINT |
| datum\_type\_code | UINT |
| precision\_db\_site | 外键。精度类型。 | HEXSTRING(16) |
| precision\_db\_id | UINT |
| precision\_type\_code | UINT |
| elevation\_db\_site | 外键，高程类型。 | HEXSTRING(16) |
| elevation\_db\_id | UINT |
| elevation\_type\_code | UINT |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

1. 位置信息2（segment\_geo\_position）

|  |  |  |
| --- | --- | --- |
| **字段** | **解释** | **大小/数据类型** |
| segment\_site | 主键。关联的segment。 | HEXSTRING(16) NOT NULL |
| segment\_id | UINT NOT NULL |
| geo\_db\_site | 主键。关联的position id。 | HEXSTRING(16) NOT NULL |
| geo\_db\_id | UINT NOT NULL |
| geo\_position\_id | UINT NOT NULL |
| gmt\_at\_location | 主键。Segment在此位置的时间。 | DATETIME(19:29) NOT NULL |
| loc\_hr\_delta | 本地时间小时偏移量 | SHORT |
| loc\_min\_delta | 本地时间分钟偏移量 | USHORT |
| duration | Segment在此位置的持续时间 | DOUBLE |
| dur\_eu\_db\_site | 外键。持续时间的单位。 | HEXSTRING(16) |
| dur\_eu\_db\_id | UINT |
| dur\_eu\_type\_code | UINT |
| gmt\_last\_updated | 上次更新时间 | DATETIME(19:29) |

# 软件研制流程

对于本项目软件的研制过程，主要基于采集台架、实车数据，通过算法模块的开发和集成开发工具软件来实现整个软件的构建与部署。软件的研制过程如下图所示：



1. 软件研制流程

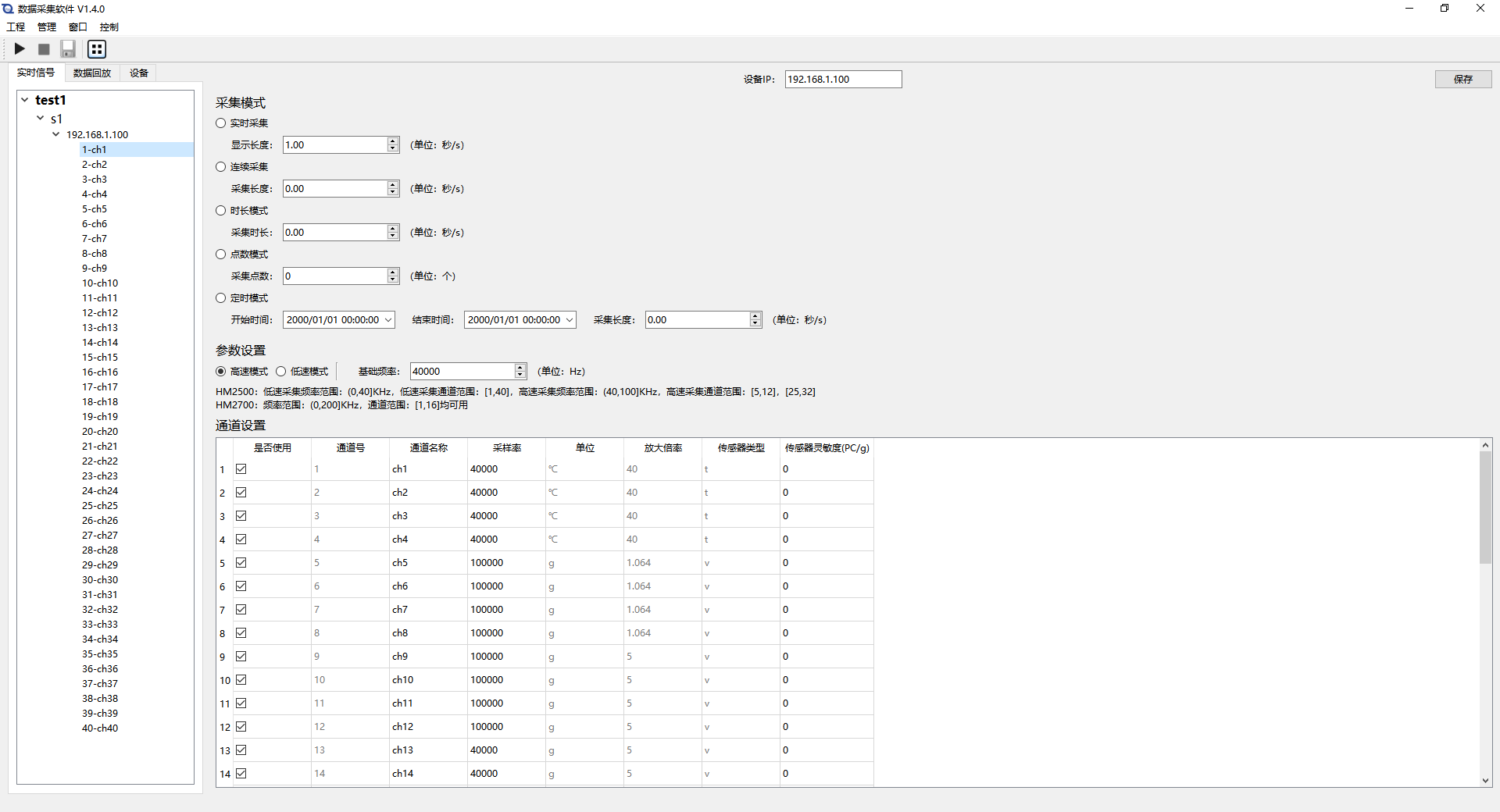
软件研制首先通过数据采集与分析软件现场采集台架和实车的传感器数据，将采集后的数据导入到算法工具箱软件。通过算法工具箱软件创建发送机系统的故障诊断与评估算法工具包，基于导入的训练数据完成算法的模型训练。同时，在算法工具箱软件还可以导入测试数据来验证模型的准确性。

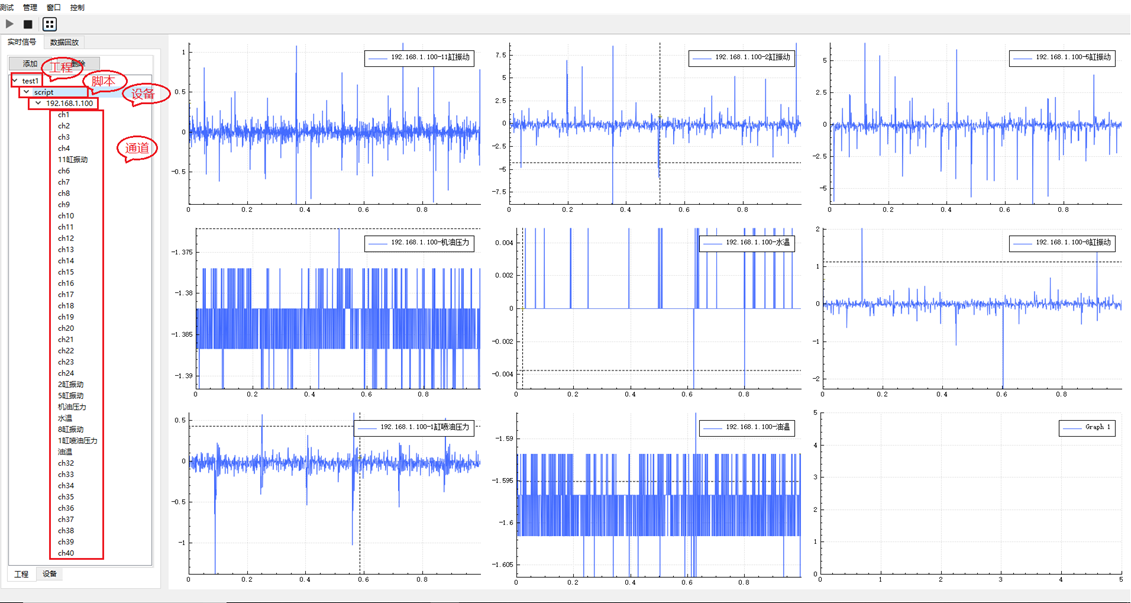
完成模型训练后，通过算法工具箱软件导出训练好的算法模块（包含算法模型），导入到开发平台软件中，在开发平台软件中创建项目工程，完成软件功能模块及相关的系统配置，生成可执行的运行软件。

然后将运行软件安装部署在对应的采集装置和终端电脑中，同时加载定制化开发的人机交互界面模块，完成整个项目软件的研制过程。

## 数据采集

本项目的数据采集主要通过数据采集与分析软件来实现，在采集数据时，通过以太网连接数据采集装置，可通过人机界面配置采集的通道、采样率、采集策略等。然后将采集的数据导出为csv格式的文件。



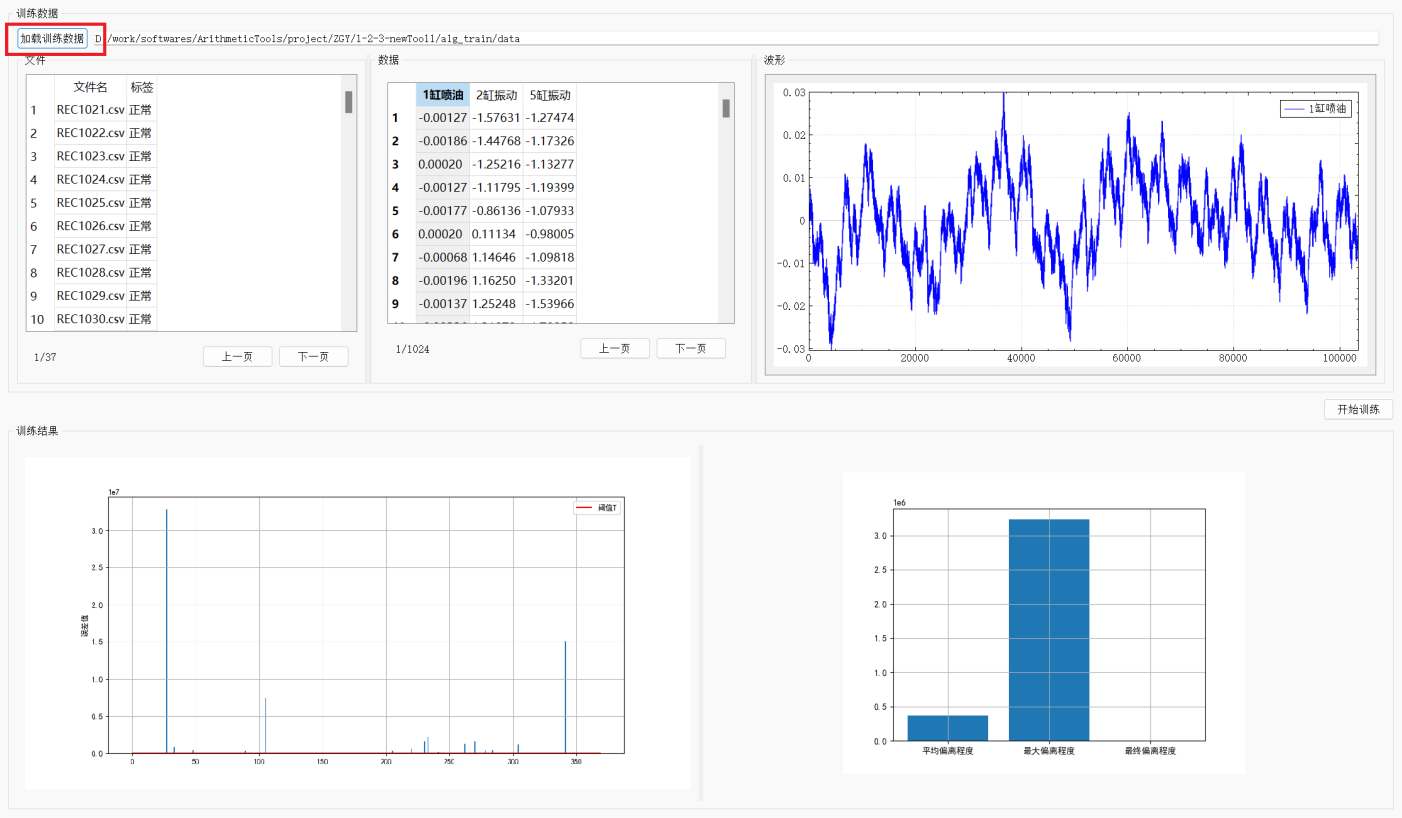


1. 数据采集

## 训练算法模型

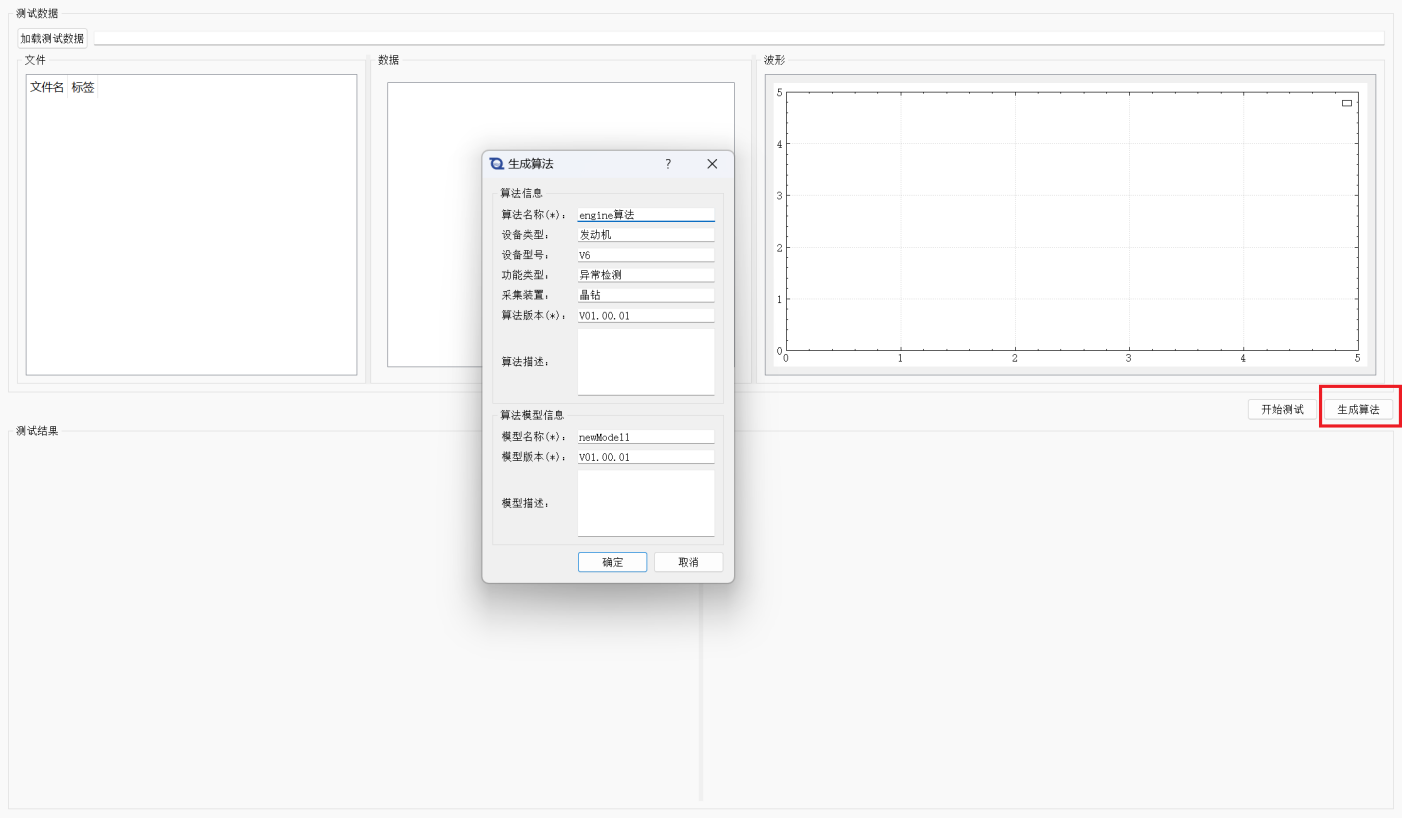
完成数据采集后，将采集的数据文件导入到算法工具箱软件，通过算法工具箱软件创建对应的发动机故障诊断算法和异常状态评估算法。

通过人机界面进行算法模型训练，可查询训练的结果。如下图所示：



1. 算法训练

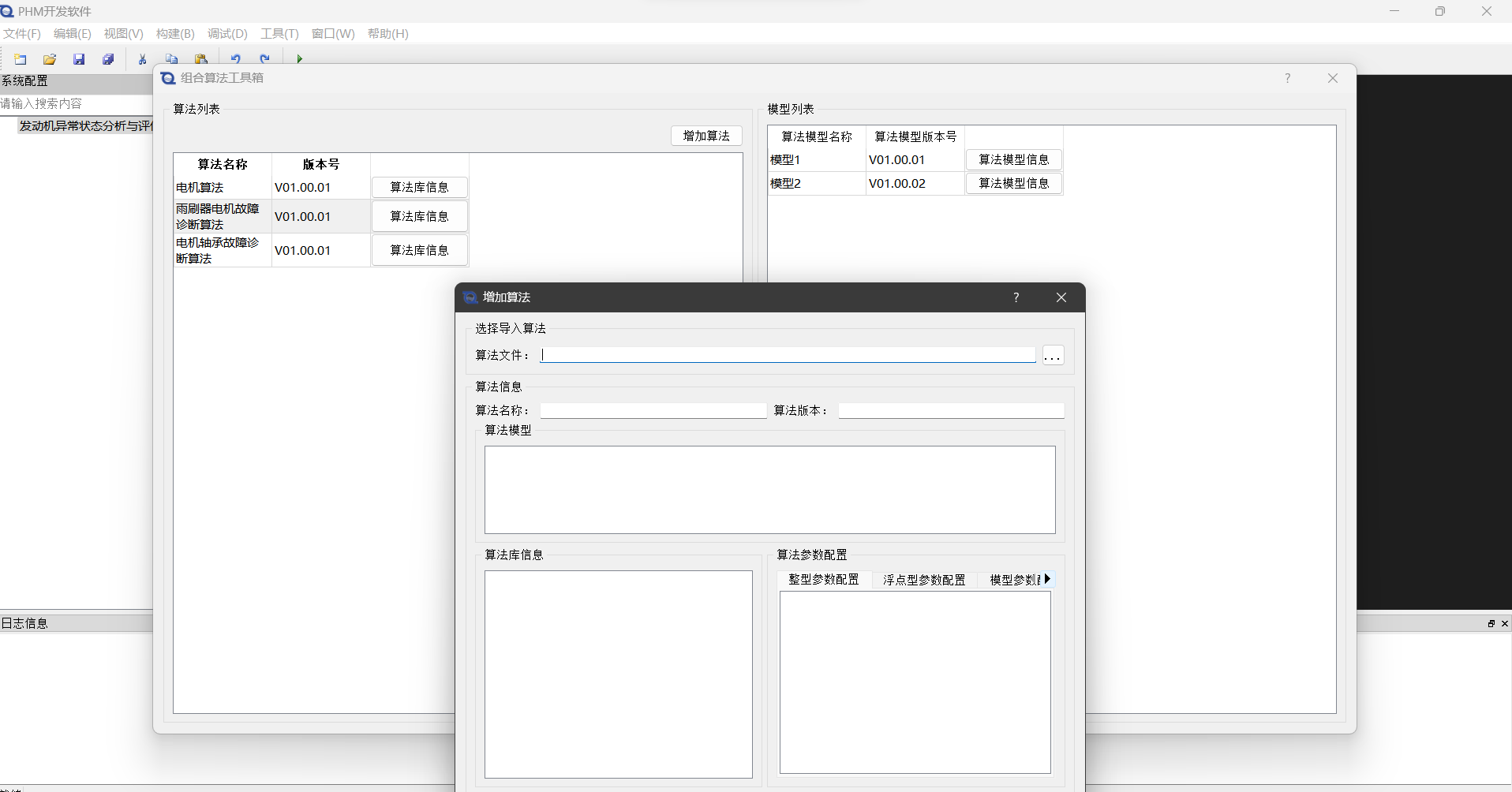
完成算法训练和测试后，可将算法打包导出。



1. 算法导出

## 开发平台构建

将导出的算法包导入到开发平台软件中。如下图所示：



1. 算法导入

然后在开发平台软件中，创建对应的运行软件工程，通过系统组成建模、功能模块及信号流建模等配置来构建出可运行的运行程序。如下图所示：

1. 工程构建

对于开发平台的工程构建，首先需要根据发动机系统组成构建产品树，在产品树上添加对应的PHM处理机（数据采集装置和终端设备）；然后在PHM处理机中构建对应的功能模块和算法模块，以及各模块间的输入输出关系。

在上图项目工程中，数据采集装置中的数据预处理模块输出的数据需要传递给终端设备中的诊断评估模块，通过webservice服务来实现快平台的数据传输功能。