

Design and Development of Scheduling and Control Module in Smart Device Management System Based on EDDL and FDT Technology

Qingmin Han¹, Fangyun Zou²

¹National Computer System Engineering Research Institute of China, Beijing

²Beijing Consen Technologies Co., Ltd., Beijing

Email: hanqm@ncse.com.cn, zfangyun@163.com

Received: May 22nd, 2018; accepted: Jun. 6th, 2018; published: Jun. 14th, 2018

Abstract

The smart device management system is a universal system based on EDDL and FDT device integration technology for configuration and fault diagnosis and maintenance analysis of all kinds of field devices. The scheduling and control module is the key interface component of the system that can support two technologies of EDDL and FDT at the same time. In this paper, the generation of smart device management system, and device integration technology are introduced. Then, the solution of scheduling and control module is put forward based on the existing problems of the smart device management system. On the basis of functional decomposition, the functional interface design of the module, the interaction design between the sub modules and the realization of the integration of the two technologies of EDDL and FDT through the module are described in detail by the guidance of the design idea of interface and logic separation. Finally, the smart device management system implemented by the design scheme, and application effect are introduced. The design of the scheduling and control module can provide design reference for the development of similar systems.

Keywords

Smart Device, Scheduling and Control, EDDL, FDT/DTM

基于EDDL和FDT技术的智能仪表管理系统中调度与控制模块的设计与实现

韩庆敏¹, 邹芳云²

¹华北计算机系统工程研究所, 北京

²北京康吉森技术有限公司, 北京

Email: hanqm@ncse.com.cn, zfangyun@163.com

收稿日期: 2018年5月22日; 录用日期: 2018年6月6日; 发布日期: 2018年6月14日

摘要

智能仪表管理系统是基于EDDL和FDT设备集成技术对各种现场总线仪表进行组态配置、故障诊断和维护分析决策的通用系统, 而调度与控制模块是该系统能够同时支持EDDL和FDT两种技术的关键接口组件。本文首先介绍了智能仪表管理系统的产生及设备集成技术, 然后基于现有仪表管理系统存在的问题提出调度与控制模块的解决方案, 并对其进行功能分解的基础上, 通过采用界面和逻辑分离的设计思路, 详细描述了该模块的功能接口设计、子模块间交互设计以及如何实现EDDL和FDT两种技术的融合, 最后介绍了采用该设计方案实现后的软件系统及其达到的效果, 为类似系统的开发设计提供参考依据。

关键词

智能仪表, 调度与控制, EDDL, FDT/DTM

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着现场总线技术的发展及大量应用, 为工业自动化生产带来了巨大改变, 在石油化工、电力等行业的控制系统中, 往往存在着来自不同厂商、支持不同现场总线协议的智能仪表设备[1] [2]。这些现场总线设备和智能仪表在运行过程中会产生大量数据, 根据访问属性可以分为配置类数据和状态类数据, 配置类数据是可读可写的, 设备维护人员可以根据设备的使用工况进行远程设置和动态调节; 状态类数据是只读的, 由设备自身生成和管理, 反应设备的运行状态, 能够为设备维护人员实时提供设备的各种诊断和报警信息。根据轮询方式可以分为周期数据和非周期数据, 周期数据实时反馈仪表测量值, 非周期数据反馈设备的状态信息, 包括远程和就地故障、报警信息、电源状态、通道状态、维护信息、测量值单位、质量、量程等, 这些数据可以帮助用户及时诊断现场设备的实时状态, 对设备的管理、预测性维护提供了有效的数据支撑[3] [4] [5]。

但是现场智能仪表提高劳动生产力的同时也给用户造成了管理上的难度, 仪表种类和数量越多, 需要的维护人员和各种类型的维护管理设备越多, 因此用户希望能够通过一种统一的操作平台来获取现场智能仪表设备的监测信息及设备自身的运行状态信息, 并对设备进行配置、监视、操作、诊断等管理, 这便是智能仪表管理系统出现的根本原因[6] [7]。

目前国外许多不同的供应商均提供智能仪表管理系统和相关产品, 而 EDDL 技术和 FDT/DTM 技术是智能仪表管理系统常用的设备集成技术。由于两种技术的发布时间、发展方向和技术路线的不同, 各系统对两种技术的支持程度也有所不同, 通常情况下, 各系统最终实现方式为同一系统两种技术的功能

相互独立, 对于 FDT/DTM 的支持提供独立的 FDT 框架程序或接口; 而以 EDDL 为基础的系统框架下也不能添加相应的 DTM 设备。本文研究的调度与控制模块正是能够解决这两种技术在一个设备管理系统平台相互融合的问题, 从而实现在一个设备集成平台上对不同厂商设备进行配置和管理, 使智能设备发挥最大效益, 实现对设备调试、组态、标定和预测性维护等管理功能。

2. 设备集成技术

为了解决不同厂家采用不同协议的设备之间进行互操作的问题, 智能仪表管理系统通常支持 EDDL 或 FDT/DTM 两种设备集成技术。其中, EDDL 技术能够采用统一规范的描述语言来表示不同厂商不同类别设备的属性, 形成 EDD 文件, 而 EDD 解析器通过读取 EDD 文件为用户产生统一的输入输出界面, 该技术能够实现远程配置、调试、校准、诊断设备等功能, 具有易于更新、易于存储、支持多语言等优点, 但该技术对于复杂设备的描述、设备信息的可视化描述存在一定的局限性; 而近几年流行的 FDT/DTM 技术正好能够弥补 EDDL 技术的不足, 它是一种软件接口描述技术, 不仅提供与主机系统的接口, 还提供设备厂商开发设备描述 DTM 软件组件的接口, 使得按 FDT 标准开发的软件工具能够集成到基于 Windows 的主机系统中, 同时不同厂家按照标准接口开发的 DTM 能够集成到符合规范的 FDT 框架应用程序中, 从而实现对现场不同厂家不同协议的设备进行统一的配置和管理, 但这种技术需要增加设备开发的成本和集成的复杂性。

由于这两种设备集成技术各有优势和不足, 作为设备集成统一操作平台, 本文设计的智能仪表系统将支持 EDDL 和 FDT/DTM 两种设备集成技术, 而调度与控制模块能够同时对 EDD 设备和 DTM 设备进行管理, 从而实现对同时支持这两种技术的设备进行相互切换, 方便用户使用且能有效发挥两种技术各自优势, 弥补各自的不足。

3. 功能结构设计

智能仪表管理系统是基于 EDDL 和 FDT/DTM 设备集成技术对各种现场智能仪表进行组态配置、故障诊断和维护分析决策的通用系统[8], 采用基于功能块组件的设计方法, 通过各组件提供的接口进行数据的交互和共享, 其功能结构见图 1。

其中调度与控制模块是智能仪表管理系统设计中的重要组成部分, 是各应用模块与主框架集成的关键接口组件, 负责为各应用模块数据实体与主框架的交互提供统一的接口, 各上层应用模块获取数据后由主框架集中显示和操作, 这样能够降低主框架程序和各应用模块实现层的耦合程度, 提高开发效率和系统可维护性。

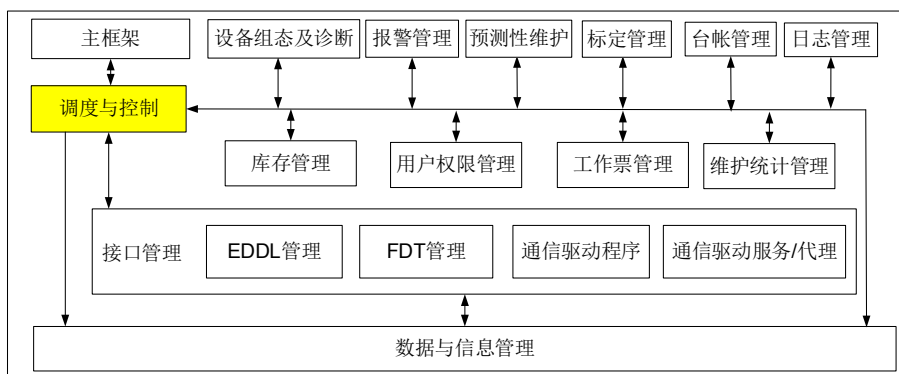


Figure 1. Functional structure design of smart device management system

图 1. 智能仪表管理系统功能结构设计图

智能仪表管理系统采用界面和逻辑分离的设计思路, 调度与控制模块作为主程序的逻辑控制与组织部分, 为上层的框架调用提供接口服务支持[9], 接收主框架传送的操作指令和消息, 并负责处理和分发给各应用模块, 从而降低主界面程序和各应用模块的耦合程度。

智能仪表管理系统由多个功能应用模块组成, 各应用模块之间功能相对独立, 由于调度与控制模块负责创建、销毁和管理各应用模块对象, 能够为模块间的交互提供沟通桥梁, 该模块提供对以下应用模块的管理和维护: 主框架、EDDL 管理模块、报警管理模块、数据与信息模块、标定管理模块、用户管理模块、接口管理模块、日志管理模块、预测性维护管理模块、台账管理模块、工作票管理模块、FDT 管理模块等, 该模块与其他各应用模块的关系见图 2, 其关系描述如下:

- ① 为主框架提供操作接口, 负责传递和响应主框架发送的指令或消息, 并且在自身有请求时通过与主框架的回调接口向上层发送请求。
- ② 创建 EDDL 管理组件对象, 通过控制 EDDL 管理模块, 实现对 EDDL 设备对象的所有操作, 并传递主框架对 EDDL 模块信息的请求。
- ③ 创建报警管理组件对象, 与报警模块交互, 传递主框架对报警管理模块信息的请求, 并实现报警管理模块与 EDDL 管理模块的交互任务。
- ④ 创建数据与信息组件对象, 调用数据与信息模块接口, 实现数据的读取和保存。
- ⑤ 创建标定管理组件对象, 传递主框架对该模块信息的请求。
- ⑥ 创建用户管理组件对象, 供其他模块调用。
- ⑦ 创建接口管理组件对象, 管理数据的导出功能。
- ⑧ 创建日志管理组件对象, 供其他模块调用。
- ⑨ 创建预测性维护管理组件对象, 供其他模块调用。
- ⑩ 创建台账管理组件对象, 传递主框架对该模块信息的请求, 并且响应台账模块的信息请求。
- ⑪ 创建工作票管理组件对象, 传递主框架对该模块信息的请求。
- ⑫ 创建 FDT 管理组件对象, 通过控制 FDT 管理模块, 实现对设备对象的所有操作, 并传递主框架对 FDT 模块信息的请求。

由于调度与控制模块同时拥有 EDDL 管理组件对象和 FDT 管理组件对象, 因此能够方便快捷地实现两种技术的相互融合和切换。

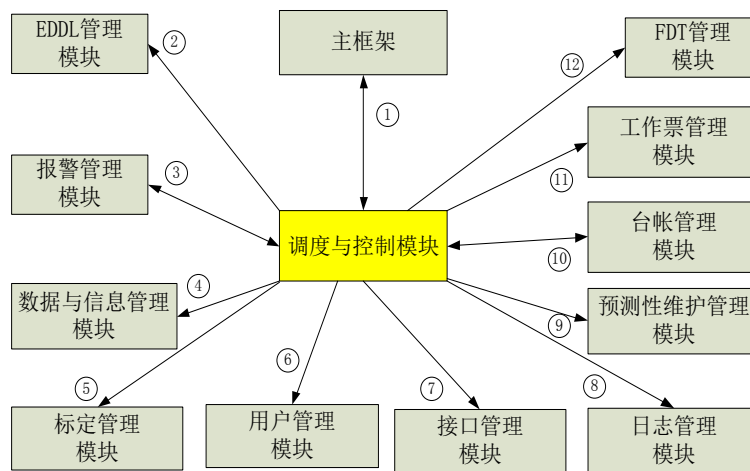


Figure 2. Relationship with other modules
图 2. 与其他模块的关系

4. 调度与控制模块的设计

按照功能点、处理逻辑和流程设计上的需要, 调度与控制模块可分为对象管理模块、数据管理模块、注册管理模块、回调管理模块四个子功能模块, 见图 3。

4.1. 对象管理模块

对象管理模块负责其他各模块的创建、初始化、销毁、释放资源的过程, 并且在整个软件运行期间, 对这些组件对象进行维护, 传递主程序对这些模块的调用请求, 对象管理模块与其他模块关系描述见表 1。

主框架创建 IUICtrlCallBack 回调接口对象, 并在系统初始化时传入, 由对象管理模块保存, 调度与控制模块通过该对象向主框架主动发送请求。对象管理模块保存了调度与控制内部所有被创建的对象指针, 并且提供了访问这些对象的接口函数, 作为输出数据可以被其他模块访问。

对象管理模块中的 CUICtrlManagerClass 类型对象在调度与控制模块的生存周期内有效且唯一, 负责创建外部组件对象以及模块内部的全生命周期对象。CTopologManager、CAAlertMonitorCallBack、CAccountInfoManageCallBack、CAreaManager 等类型的对象作为 CUICtrlManagerClass 类的成员数据存在, 通过 CUICtrlManagerClass 类的 InitUICtrlObject() 函数, 创建和初始化其他各外部模块, 并且负责建立起模块间的回调关系, 见图 4。

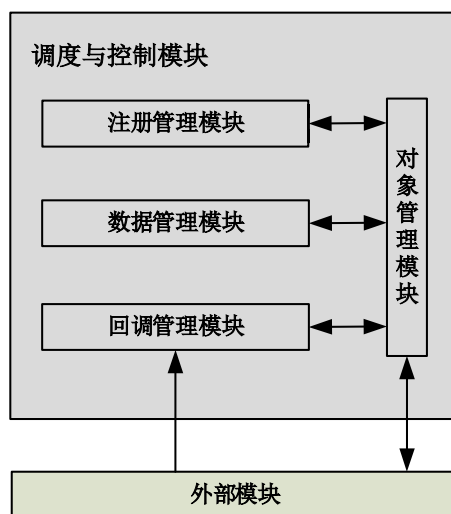


Figure 3. Functional decomposition of the module
图 3. 模块功能分解

Table 1. Object management module relationship description diagram

表 1. 对象管理模块关系描述图

编号	描述信息	属性/函数名/变量	类名/文件名
①	接收框架程序的调用以及向框架程序发送请求	调度与控制模块提供的所有外部接口及函数	调度与控制模块提供的所有外部接口及函数的相关文件
②	管理和维护注册管理模块对象, 传递框架程序的调用命令	注册管理模块相关类	注册管理模块相关类的实现文件
③	管理和维护数据管理模块对象, 传递框架程序的调用命令	数据管理模块相关类	数据管理模块相关类的实现文件
④	管理和维护回调管理模块对象, 传递框架程序的调用命令	回调管理模块相关类	回调管理模块相关类的实现文件

4.2. 数据管理模块

数据管理模块负责与数据库进行交互, 管理并维护在软件运行周期内的框架程序产生的所有数据, 包括区域信息、物理网络信息、设备对象、日志信息。数据管理模块对这些数据进行组织, 并负责实现管理这些对象的处理流程。同时数据管理模块是调度与控制接口的操作实现体, 框架通过对组件的调用, 通过该子模块实现。该模块的输入项包含从 EDDL 管理模块、报警管理模块、台账管理模块中获取的相关数据信息, 输出项是经过处理供其他各模块使用的相关数据等, 如区域信息、网络信息、设备对象信息等。数据管理模块关系描述见表 2。

4.2.1. 区域信息的管理

区域信息指由框架程序进行区域组态时产生的数据, 涉及的操作过程包括添加、删除、修改、保存和恢复等。为了实现界面和逻辑数据的分离, 上层框架不负责维护任何区域信息对象的数据, 只把相应的操作请求发送给调度与控制模块, 然后由该模块负责对区域节点对象的添加、删除、重修改、保存、读取等操作, 调度与控制模块会建立一个数据管理对象专门负责维护这些数据, 在必要时, 框架通过组件的接口函数获取这些数据的指针, 将来这些资源的释放也由调度与控制组件实现, 和上层无关。

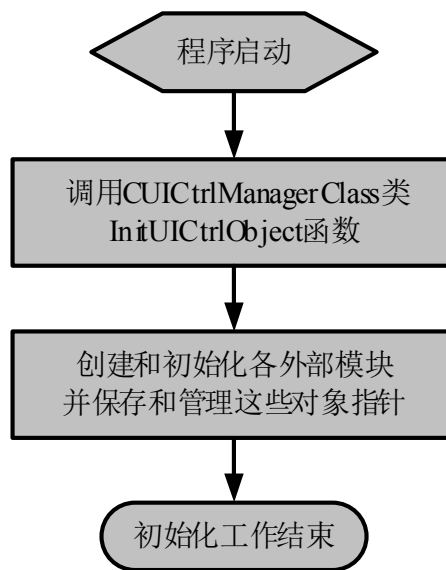


Figure 4. Module function decomposition
图 4. 模块功能分解

Table 2. Data management module relationship description diagram
表 2. 数据管理模块关系描述图

编号	描述信息	属性/函数名/变量	类名/文件名
①	调用对象管理模块提供的接口函数, 获取相应对象的指针	对象管理模块提供的 GetXXXObiPtr()函数	CAreaManager CEDDLDevice CEDDLDeviceExplor CEDDLDeviceStatus CEDDLNetWorkPage CNetWorkManagerDlg CTopologManager CXMLTree

4.2.2. 网络信息的管理

网络信息指由框架程序进行物理网络组态时产生的数据, 包括物理网络的拓扑结构信息、设备对象等。为了实现界面和逻辑数据的分离, 与区域信息管理实现的思路相同, 上层框架不负责维护网络信息的维护, 也是通过一系列的组件接口, 委托调度与控制组件执行相应的操作。组件内部建立了一个专门用来维护网络拓扑结构的类, 它和上层框架显示的网络结构是一致的, 在操作时二者保持一致, 通过接口函数, 上层框架可随时获取这些信息。将来这些资源的释放也由调度与控制组件实现, 和上层无关。

4.2.3. 设备对象的控制流程

调度与控制组件拥有所有设备对象的指针, 它通过调用 EDDL 管理对象提供的接口函数, 能够控制设备对象的行为, 如上线、上载参数、打开设备参数窗口等操作。调度控制对象在接收框架发送来的操作命令后, 向 EDDL 管理组件传递这些请求, 实现对设备操作的目的。

4.2.4. 数据存取的管理

软件运行过程中产生的数据要被保存到数据库中, 包括区域组态信息、网络组态信息、日志信息等。其中区域组态信息可直接调用数据库提供的操作接口进行保存或读取。日志信息则直接调用数据库提供的保存日志的接口, 写到数据库中。

4.3. 注册管理模块

注册管理模块用来控制和管理软件的注册授权, 提供了授权限制、软件注册验证和管理等功能。注册管理模块关系描述见表 3。

软件分为注册授权状态和非注册授权状态, 如果是非注册授权状态在软件启动时开始计算演示运行时间, 限制添加智能仪表数量。软件使用过程需要考虑智能仪表数量限制, 加密算法需要把用户名、授权点数、当前机器的机器码实现加密。在注册过程中只要有一个环节信息不正确, 生成的注册码是错误的, 注册授权就会失败。

软件启动时, 首先判断软件是否在本机上被授权使用, 注册管理模块从本地注册表中寻找登记的注册信息, 包括机器码、授权码、用户名、授权点数。然后在程序中经过一定的加密算法动态生成机器码, 并与从数据表中获得的机器码比较是否相同, 不同表示注册信息非法并返回未注册。如果机器码正确, 则调用加密算法模块, 把由用户名、授权点数、机器码、授权码组合成的信息段进行加密, 动态生成正确的授权码, 然后和从数据库读到的授权码进行比较, 相同则返回已注册, 否则返回软件未注册。软件注册管理流程见图 5。

4.4. 回调管理模块

回调管理模块包含了调度与控制模块实现的所有回调接口类, 回调接口是外部模块与本模块交互的渠道, 通过回调接口, 外部模块可以向本模块发送请求, 涉及报警管理模块、台账管理模块。回调管理模块关系描述见表 4。

Table 3. Registration management module relationship description

表 3. 注册管理模块关系描述图

编号	描述信息	属性/函数名/变量	类名/文件名
①	对象管理模块创建并保存注册管理对象的指针, 并为其他模块提供访问该对象的调用接口	GetRegisterObjPtr CRegisterDlg m_RegisterDlg	CUICtrlManagerClass CRegisterDlg

Table 4. Callback management module relationship description diagram

表 4. 回调管理模块关系描述图

编号	描述信息	属性/函数名/变量	类名/文件名
①	与报警模块、台账模块和诊断与报警插件的回调过程	CaccountInfoManageCallBack 类中的函数实现了和报警管理模块间的回调。 CAlertMonitorCallBack 类中的函数实现了和台账管理模块间的回调。 CDeviceStatusManageCallBack 类中的函数实现了和诊断与报警插件的回调。	CAccountInfoManageCallBack; CAlertMonitorCallBack; CDeviceStatusManageCallBack;
②	回调接口对象被对象管理模块管理	作为CUICtrlManagerClass类中的 m_AccountInfoManageCallBack变量和 m_AlertMonitorCallBack变量 m_DeviceStatusManageCallBack变量	CUICtrlManagerClass

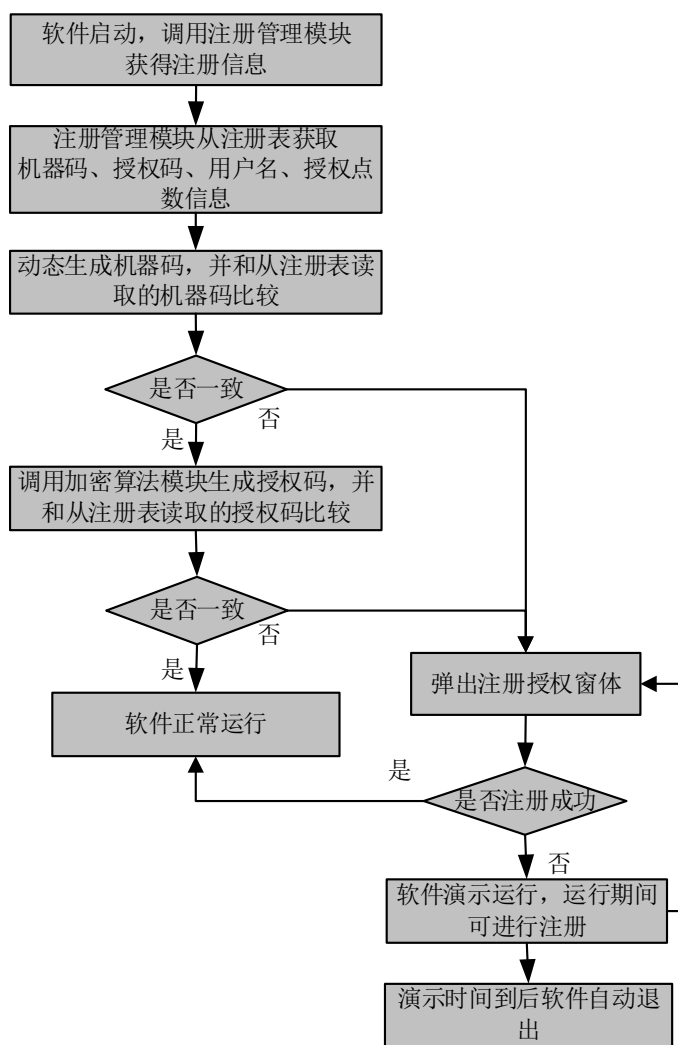


Figure 5. Registration module management process

图 5. 注册模块管理流程

回调接口为台账管理模块和报警管理模块定义的接口，调度与控制模块通过引入继承并实现。在本组件初始化时，分别定义这些接口类的对象，并通过台账和报警模块的初始化函数将这些对象指针分别

传递给它们, 供其在需要时调用, 这样就建立起了关联关系, 见图 6。

5. 系统实现

采用该方案设计和实现的智能仪表管理系统的主界面见图 7, 对于与该系统平台连接的同一设备,

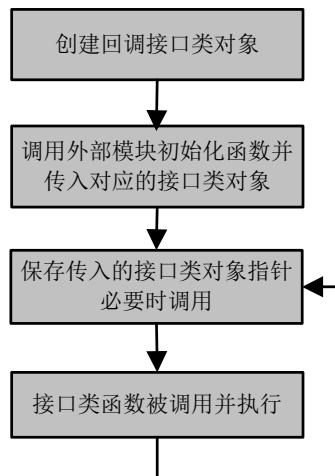


Figure 6. Callback establishment process

图 6. 回调建立流程

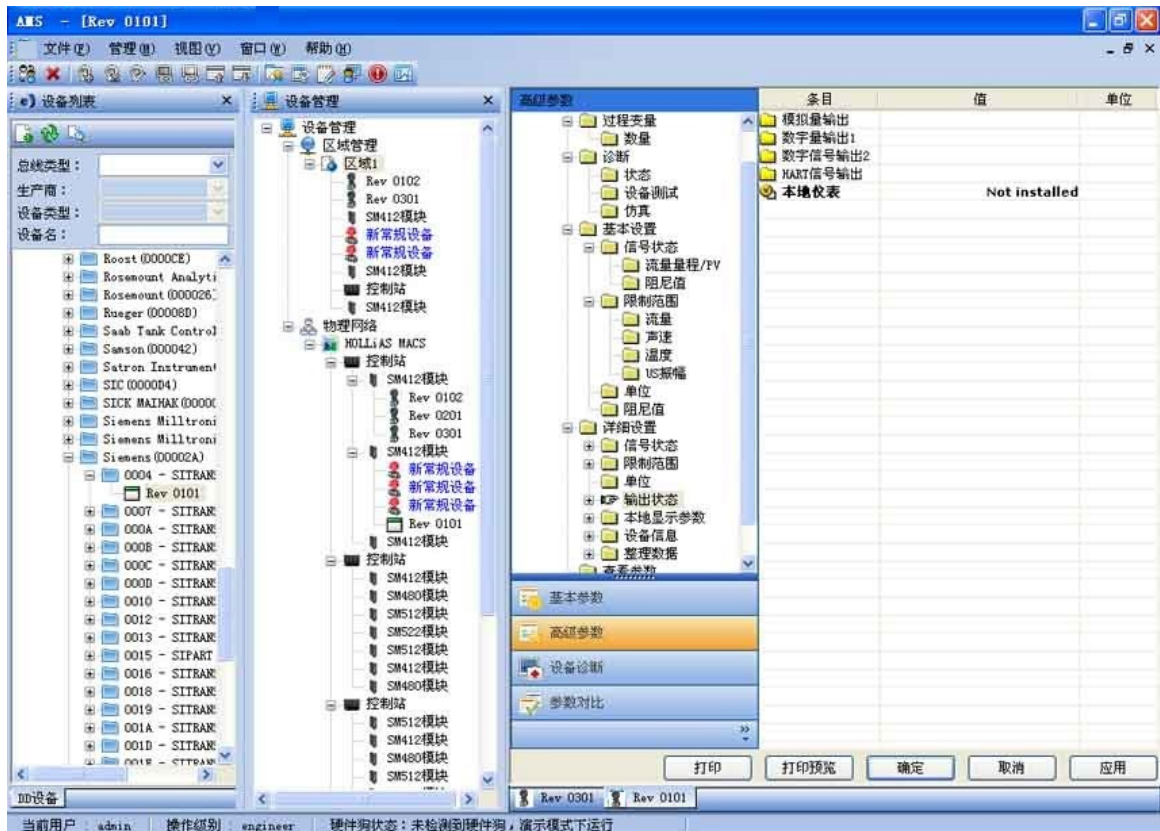


Figure 7. Main interface of smart meter management system

图 7. 智能仪表管理系统主界面

可以采用 EDD 方式和 FDT/DTM 方式打开配置界面, 用户可以方便地进行 EDD 和 DTM 两种集成方式的切换, 两种设备集成技术的支持能够为用户提供智能仪表管理与维护功能, 使智能设备自身能够发挥最大的生产效益, 减少智能仪表的损耗并降低维护成本, 为用户提供一个快速、方便、统一的管理平台, 可以大幅提高工厂使能设备的可用性、生产能力和产品质量, 改变了传统的维护操作方式, 有效的管理了工厂当前的资产, 优化了生产控制过程, 从而达到提升工厂整体维护管理水平、提高生产力、实现企业运行最优化的目标。

6. 结语

本文主要介绍了智能仪表管理系统中调度与控制模块的设计实现方法, 调度与控制模块作为逻辑部分, 在设计与实现过程中, 为智能仪表管理系统主框架提供操作接口, 为各模块间的交互实现了沟通的桥梁, 提高开发效率, 既方便了多个模块之间的交互, 又降低了主界面程序与实际功能实现层的耦合程度; 另外, 该模块将 EDDL 和 FDT/DTM 两种设备集成技术进行相互融合和切换, 能够有效地发挥两种技术各自的优势, 弥补各自的不足, 由于各自技术细节均在独立的模块中实现, 如何将两种技术进行深入融合, 尚待进一步研究。

参考文献

- [1] 刘丹. 自动控制系统的设备集成技术——EDDL 与 FDT/DTM [J]. 仪器仪表标准化与计量, 2008(3): 9-11.
- [2] 华镭. 把现场总线集成到系统中来[J]. 中国仪器仪表, 2007(12): 25-28.
- [3] 董永刚, 李昱, 等. 基于 EDDL、FDT 技术实现互联互通的研究与应用[J]. 自动化博览, 2017(11): 76-79.
- [4] 刘桐杰, 张树强, 等. 设备管理系统实现全厂智能仪表一体化管理的研究及应用[J]. 自动化博览, 2017(10): 36-40.
- [5] 史先亚, 高海东, 李晓博, 等. 基于 EDDL 的现场总线设备管理系统设计与实现[J]. 热力发电, 2017, 46(1): 121-123.
- [6] 包伟华, 陆妹. EDDL 和 FDT 技术的研究及实现[J]. 仪器仪表标准化与计量, 2010(1): 42-45.
- [7] 王松林. 基于 EDDL 的设备管理系统的研究与开发[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2010.
- [8] 邹芳云, 杨瑞峰. 智能仪表管理系统的设计与应用[J]. 自动化与仪表, 2013, 28(10): 5-10.
- [9] 王毓琦. 通用可编程测试平台的研究与开发[D]: [硕士学位论文]. 济南: 山东大学, 2015.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org